

8
ZEFJ



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**APLICACION DE LA TECNOLOGIA
ORIENTADA A OBJETOS PARA EL
DESARROLLO DE UN SISTEMA DE
PRECIOS UNITARIOS.**

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N :

AUSTRIA LÓPEZ ENRIQUE

CISNEROS ROJAS ROMÁN

ESLAVA SOTO GUADALUPE

GARCÍA MARTÍNEZ VIRIDIANA

SANDOVAL BARRAGÁN LAURA

Bajo la Dirección de: M.I. Juan Carlos Roa Beiza



MEXICO, D.F.

1995

FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A mis padres y hermanos, por
ser lo mejor que tengo en la
vida, porque todo lo que
alcanzamos lo logramos juntos.*



Laura T. Sandoval Barragán

A mis padres. Guadalupe
Solo y Tomás Estava. a
quienes debo lo que soy, por
haberme brindado su apoyo
y amor en todo momento.



A mis hermanos por que
siempre he recibido su apoyo.



Guadalupe Estava Solo.

A mis padres, pues gracias a su apoyo y enseñanza he logrado salir adelante y ser lo que ahora soy.

A ustedes todo mi amor y admiración.



A mis hermanos: Montserrat y Ramón los cuales han participado en este logro y con los que he pasado los mejores momentos de mi vida. Gracias!



Viridiana García Martínez.

*A mis padres, hermanos y
amigos, por el apoyo,
confianza, comprensión y
motivación que me brindaron
para la culminación de mis
estudios.*



Enrique Austria Lopez.

A mi madre Ernestina a quien debo tantos años de cariños y cuidados. A ti que me has ayudado a ser lo que hoy soy.

A mi padre Moises que con tanto esfuerzo has costado mis estudios, que me enseñaste con el ejemplo a luchar por lo que verdaderamente quieras, sabiendo que sólo con esfuerzo se consiguen las cosas que uno aprecia.



A mi hermana Judith a quien siempre he querido, aunque regularmente no se lo demuestro, pero que me ha enseñado que no solo con estudiar se consigue una educación, la vida es la que enseña y hay que estar dispuestos a aprender.

A ti Nora por compartir 16 años de tu vida, que me ayudaron a madurar y a enfrentarla con alegría. Gracias por demostrarme con tu ejemplo que se puede ser feliz pese a la adversidad.

Román Casneros Rojas

*A la U.N.A.M. y a la Facultad
de Ingeniería por habernos abierto
sus puertas y así permitirnos
desarrollarnos profesional e
intelectualmente dentro del campo
de la Ingeniería.*



*Nuestro agradecimiento muy
especialmente al M. I. Juan
Carlos Ros Beiza por su apoyo
para la culminación de esta tesis.*



*A la U.N.A.M. y a la Facultad
de Ingeniería por habernos abierto
sus puertas y así permitirnos
desarrollarnos profesional e
intelectualmente dentro del campo
de la Ingeniería.*



*Nuestro agradecimiento muy
especialmente al M. I. Juan
Carlos Rou Beiza por su apoyo
para la culminación de esta tesis.*



INDICE

OBJETIVOS

CAPITULO I: CONCEPTOS BÁSICOS EN LA INDUSTRIA DE LA CONSTRUCCIÓN

1.1	ANTECEDENTES	1
1.1.1	FUNCIONAMIENTO GENERAL DE UNA CONSTRUCTORA	1
1.1.2	CONCEPTOS BÁSICOS DE PRECIOS UNITARIOS	22
1.1.3	CONCEPTOS BÁSICOS DE CONTROL DE OBRA	31
1.1.4	CONCEPTOS BÁSICOS DE PLANEACIÓN DE PROYECTOS	41
1.2	MÓDULO DE PRECIOS UNITARIOS	51
1.2.1	INSUMOS	51
1.2.2	ANÁLISIS	62
1.2.3	PRESUPUESTOS	69
1.3	MÓDULO DE CONTROL DE OBRA	98
1.3.1	CÁLCULO DE ESTIMACIONES	98
1.4	MÓDULO DE PLANEACIÓN DE PROYECTOS	118
1.4.1	INTEGRACIÓN DE DATOS DEL PRESUPUESTO	118
1.4.2	MATRIZ DE SECUENCIAS	124
1.4.3	BARRAS DE GANTT	135
1.4.4	VOLUMEN DE OBRA	138
1.4.5	EGRESOS	142
1.4.6	EXPLOSIÓN DE INSUMOS	168

CAPITULO II: PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.1	RECOPIACIÓN Y ANÁLISIS DE INFORMACIÓN QUE SE DISPONE	81
2.2	REQUERIMIENTOS DEL USUARIO	190
2.3	DIAGRAMA DE DESCOMPOSICIÓN FUNCIONAL	203
2.4	OPCIONES DE SOLUCIÓN EN BASE A LOS REQUERIMIENTOS ANALIZADOS	219
2.5	ELECCIÓN DE SOLUCIÓN ÓPTIMA	235

CAPITULO III: BASES DE DATOS Y CARACTERÍSTICAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

3.1	CONCEPTOS GENERALES	255
3.2	TIPO DE BASES DE DATOS	268
3.3	MODELOS DE BASES DE DATOS, VENTAJAS Y DESVENTAJAS	272
3.4	NORMALIZACIÓN DE BASES DE DATOS	287
3.5	CARACTERÍSTICAS GENERALES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS	297
3.6	DISEÑO DE ARBOLES GENEALÓGICOS	312
3.7	FLUJO DE DATOS	319
3.8	ESTRUCTURAS JERÁRQUICAS TIPOS Y SUBTIPOS	335
3.9	ESTRATEGIA DE SOLUCIÓN	348
3.10	OLE UN NUEVO ENFOQUE DE LA ORIENTACIÓN A OBJETOS	362

CAPITULO IV: DESARROLLO DEL SISTEMA

4.1	ESPECIFICACIONES DEL DISEÑO	367
4.1.1	DIAGRAMA DE FLUJO DE DATOS	367
4.1.2	DICCIONARIO DE DATOS	382
4.1.3	DIAGRAMA DE ENTIDAD RELACIÓN	401
4.1.4	DISEÑO E IMPLEMENTACIÓN DE LOS MÓDULOS	418
4.1.5	INTERFASE AL USUARIO	448
4.1.5.1	DISEÑO DE FORMAS DE EDICIÓN DE DATOS	455
4.2	INTEGRACIÓN DEL SISTEMA BASADO EN LAS HERRAMIENTAS DE PROGRAMACIÓN ORIENTADA A OBJETOS	472
4.2.1	INTEGRACIÓN OLE EN EL SISTEMA DE PRECIOS UNITARIOS	472
4.2.2	INTEGRACIÓN DE APLICACIONES CON TECNOLOGÍA DE ORIENTACIÓN DE OBJETOS	609
4.2.2.1	PLANEACIÓN DE PROYECTOS CON MS-PROJECT	609
4.3	SISTEMAS DE AYUDAS	614
4.3.1	LA IMPORTANCIA DE LOS SISTEMAS DE AYUDAS ORIENTADAS AL USUARIO.	614
4.3.2	INTEGRACIÓN DEL SISTEMA DE AYUDAS AL SISTEMA DE PRECIOS UNITARIOS	616
4.3.3	MANUAL DEL USUARIO	626
4.4	DOCUMENTACIÓN TÉCNICA	789
CONCLUSIONES		809
BIBLIOGRAFÍA		843

OBJETIVOS

La industria de la construcción requiere de un proceso de modernización continuo y eficiente que le permita evolucionar más rápidamente que las demás ramas de la industria. Es por ello que contar con herramientas capaces de proveer la información de manera instantánea es de suma importancia. Una constructora realiza diferentes actividades para lograr un Presupuesto de Obra como son Área de Precios Unitarios, Planeación del Proyecto y Control de Obra, el flujo de datos entre estas diferentes actividades es de vital importancia. Así surge la necesidad de crear un esquema que nos permita la interacción de los datos entre las diferentes áreas mediante la creación de una Aplicación de Precios Unitarios desarrollada en un ambiente gráfico.

- Utilización de la tecnología orientada a objetos para la creación de un ambiente amigable y productivo para el usuario en la industria de la construcción.
- Diseño de un modelo para automatización de oficinas en la industria de la construcción.
- Elaboración de una estructura que permita aprovechar los beneficios de la tecnología OLE (Object Linking and Embedding), en la aplicación de Precios Unitarios.

- Implementación de utilerías (Add-On's) que permitan una interacción transparente y efectiva con otros productos para Windows, estas utilerías se presentan como un módulo adicional del Sistema de Precios Unitarios.
- Integración de OLE en aplicaciones de mercado vertical.



CAPITULO I

CONCEPTOS BASICOS EN LA INDUSTRIA DE LA CONSTRUCCION

1.1 ANTECEDENTES

1.1.1 FUNCIONAMIENTO GENERAL DE UNA EMPRESA CONSTRUCTORA

Una adecuada estructuración funcional y la correcta coordinación de servicios, son condiciones esenciales para obtener el más eficaz funcionamiento de todo el organismo empresarial y el aprovechamiento de los recursos disponibles de personal, equipo, técnica y medios financieros.

La forma y medida de la estructuración viene impuesta por el tamaño, el ámbito geográfico, la naturaleza y el volumen de las actividades a desarrollar y otros factores que dan la configuración exacta de la empresa.

No es posible establecer modelos exactos de estructuración para los diferentes tipos de empresa (grandes, medianas o pequeñas), en razón a la cantidad de trabajos contratados, el número de obreros y empleados adscritos a la empresa, y otros signos de catalogación y medida.

Una clasificación de la verdadera dimensión y capacidad de las empresas se obtiene valorando signos basados en cifras que miden y relacionan la propia potencia económica y financiera de estas empresas, su experiencia en la construcción de los diferentes grupos y tipos de obras, grados de mecanización y tecnicidad.

En términos generales, las estructuras básicas de las empresas constructoras están compuestas de los siguientes órganos y secciones centrales:

a) ÓRGANOS RECTORES

- Consejo de administración, (si se trata de sociedades anónimas).
- Gerencia. Presidencia.
- Dirección general. Subdirección.

b) ÓRGANOS DE DIRECCIÓN Y CONTROL TÉCNICOS

- Dirección técnica.
- Oficina técnica y de racionalización.
- Oficina de proyectos.
- Cálculos. Estudios de obras.
- Planificación y programación de trabajos.
- Previsiones. Controles. Valoraciones.
- Presupuestos. Reformados. Revisión de precios. Certificaciones.
Liquidaciones de obras.
- Control de costos. Análisis de resultados. Estadísticas.

c) ÓRGANOS DE ADMINISTRACIÓN, DIRECCIÓN COMERCIAL Y ASUNTOS GENERALES

- Dirección comercial.
- Administración general. Secretaría. Intervención.
- Contratación de obras. Financiamiento. Fianzas. Avals.

- Subcontratos.
- Tramitación de certificaciones y facturas de trabajos. Endosos. Cobros.

Retenciones.

- Gestiones. Relaciones públicas.
- Contabilidad general.
- Tesorería. Caja principal. Cuentas bancarias. Efectos.
- Proceso de datos. Mecanización de cuentas.
- Sección de personal. Departamento social.
- Compras. Ventas. Almacén general.
- Sección de maquinaria: Oficina. Parque central. Taller mecánico.

Otros servicios:

- Asesoría jurídica.
- Inspecciones.
- Servicios médicos.

d) ÓRGANOS DE EJECUCIÓN

- Dirección de obras, fabricaciones y talleres.
- Jefaturas de las obras públicas (de carreteras, puentes, presas, canales etc.).
- Jefaturas de obras de edificación (de construcción de viviendas, edificios industriales, urbanizaciones etc.).
- Laboratorios de ensayo.
- Fabricación: productos asfálticos, encofrados, andamiajes, estructuras

metálicas, elementos prefabricados, tuberías, losetas etc.

- Talleres: Carpintería, cerrajería, fontanería, electricidad etc.

Las principales funciones que corresponden a los servicios centrales de la empresa constructora son los siguientes.

SERVICIOS TÉCNICOS

Estos servicios desarrollan todas la actividades que afectan a la oficina central. Sus principales funciones son:

- Realización de proyectos completos de obras propiedad de la empresa, destinadas a su uso, venta o alquiler, especialmente cuando la empresa actúa también como inmobiliaria.
- Estudios de obras, cálculos de precios y presupuestos con vistas a la licitación en concursos y subastas. Estos estudios requieren de la colaboración de otros servicios técnicos y administrativos de la empresa: el control de costos, a través de los registros y análisis de resultados, que facilita valores reales de ejecución, técnicos y económicos, de los mismos o similares tipos de actividades realizadas por la empresa, tiempos, rendimientos, consumos, etc.; las secciones de planificación, maquinaria, compras, almacén, personal, subcontratos, que debe aportar los datos necesarios en cuanto a ofertas, precios, disponibilidades, formación de

equipos. Todo esto junto con el estudio técnico-económico del proyecto: planos, mediciones, emplazamiento de las obras, características geológicas y topográficas de los terrenos, fuentes de provisionamiento de materiales, distancias y medios de transporte, situación de la mano de obra y otros.

Finalmente se requiere de la intervención de los órganos directivos para ultimar las condiciones de licitaciones y fijar las cifras de remate de las propuestas económicas.

- **Planificación y programación de trabajos.**

Además de un empleo sistemático de las modernas técnicas de planificación, programación y control para seguir el desarrollo de los proyectos de obras, esta sección se ocupa también de la planificación y el control, a escala general, de todos los trabajos y recursos de la empresa.

La planificación de trabajos de cada obra, en particular, cuya finalidad es establecer las condiciones más favorables para el desarrollo óptimo de las actividades, abarca diversos problemas, como la elección de los procedimientos técnicos de ejecución más acertados, la preparación de los medios esenciales de trabajo para disponer de ellos en el momento y lugar adecuados; la coordinación de actividades; medida y reajuste de los tiempos, empleando las modernas técnicas de programación para llegar a cumplimiento o reducción de los plazos con la máxima seguridad y la consiguiente economía en recursos, tiempo y dinero.

Las previsiones y controles, fuentes necesarias de información general, aportan datos para orientar a los órganos directivos en la marcha actual y futura de toda la gestión de empresa. Recopila, extracta y presenta en forma estadística, cifras que se refieren a volúmenes totales de obras contratadas, presupuestos vivos de las obras en curso pendientes de ejecución; previsiones parciales y totalizadas, de los trabajos pendientes de realizar en periodos de tiempo definido.

La previsión y el control se extiende también a los consumos de materiales importantes, las instalaciones que hayan de ponerse en marcha, el empleo inmediato y necesidades futuras de la maquinaria y el transporte, previsión de gastos indirectos y de recursos financieros.

- Control de costos. Análisis de resultados.

Toda la información básica parte de las obras, y son los servicios centrales los que completan y analizan la información, llevándola a la síntesis o al grado de detalle necesario, dándole la forma estadística y gráfica, después de establecer para las actividades tratadas los costos unitarios absolutos y expresados en términos comparativos de costos reales-precios-previsión.

SERVICIOS DE ADMINISTRACIÓN, DIRECCIÓN COMERCIAL Y ASUNTOS GENERALES

Entre las funciones comunes a toda empresa constructora pueden citarse las siguientes:

- Administración e intervención en los asuntos relacionados con el patrimonio y las actividades industriales y mercantiles de la empresa.
- Gestión y contratación de obras y servicios con organismos oficiales y clientes particulares; obtención de informes y datos relacionados con los concursos y subastas; preparación de la documentación necesaria para la presentación de proposiciones; estudio e informes de la competencia; constitución y cancelación de fianzas y avales; planes y operaciones relacionados con la financiación de las obras.
- Tramitación y control de certificaciones de obras y facturas de trabajos, endoso, libramientos, cobros, retenciones.
- Gestiones. Relaciones públicas.
- Subcontratos. Debido a su importancia, los servicios son desempeñados por una sección con dedicación exclusiva a los asuntos que afectan el estudio y clasificación de ofertas; necesidades y conveniencias de la subcontratación, garantías técnicas, económicas o laborales de los posibles subcontratistas o destajistas; preparación de los subcontratos y documentos relacionados; vigilancia del desarrollo de los trabajos y el cumplimiento de todas las condiciones estipuladas; control y registro de mediciones, facturas, retenciones, pagos a cuenta, liquidaciones parciales y finales
- La contabilidad general centraliza las operaciones que afectan al movimiento normal contable de la empresa. Lleva cuentas individuales de explotación de cada obra, o grupo de obras similares, parque de maquinaria, fabricaciones y

talleres; prepara balances periódicos de los resultados económicos; interviene, registra y fiscaliza las operaciones bancarias, aceptación y cancelación de efectos, cobros, pagos y tributaciones.

- En las empresas se ha generalizado el uso de equipo electrónico para la contabilidad y otros procesos de datos: manejo de nóminas, control de costos, control de almacenes y maquinaria, operaciones de planificación y programación de trabajos, mediciones y valoraciones de obra, control presupuestario, cálculos técnicos, etc.
- La sección de personal se ocupa directa o indirectamente de los asuntos relacionados con los obreros y empleados al servicio de la empresa. De acuerdo con la dirección, dicta normas de régimen interior respecto a condiciones de admisión de personal, sistema de incentivos, premios, sanciones, medidas de seguridad e higiene del trabajo y otras.
- Tesorería y caja principal. En este departamento se agrupan los servicios de la caja principal y del movimiento de las cuentas bancarias, ocupándose de los cobros o pagos que se realizan en la oficina central y a través de los bancos, y de las demás operaciones de tesorería; custodia de títulos y valores; el registro de cuentas corrientes bancarias, de préstamos, créditos y libros de caja.
- Compras. Interviene directamente en la contratación de los suministros de materiales de gran consumo, como son el cemento, hierros, madera, productos asfálticos, prefabricados, cerámicas. La sección de maquinaria

tiene bajo su responsabilidad la tarea de atender a la reposición, renovación y actualización de máquinas y equipos. Controla y administra todo el parque de maquinaria y vehículos; autoriza y coordina sus desplazamientos; vigila el trabajo y el rendimiento de las máquinas y su buen estado de uso y conservación; cuida de los entretenimientos y reparaciones por talleres ajenos o de la empresa. **Proyecta y realiza las instalaciones de obras y de explotación de canteras.**

ÓRGANOS DE EJECUCIÓN

Estos órganos suelen estar representados por un departamento de dirección de obras, de laboratorios, fabricaciones y talleres.

Dirige y coordina con amplias facultades el funcionamiento de todos los órganos de ejecución de trabajos.

Los jefes de zona ejercen mando directo sobre el personal de los servicios de ejecución de las obras situadas en su demarcación. Sus inmediatos colaboradores son los jefes de estas obras. Su principal función es la de dirigir, controlar y coordinar las diferentes situaciones de los órganos y servicios que intervienen en la ejecución de las obras desde su preparación y desarrollo, hasta la recepción y liquidación definitivas.

Participan en la elaboración de planes y programas de trabajo, vigilando posteriormente su normal desarrollo.

Por el servicio de planificación y control de programas deben ser informados periódicamente de la marcha o cumplimiento de los objetivos y de los tiempos marcados, especialmente cuando se produzcan desviaciones que exijan medidas correctivas.

Por parte de la obras, los jefes de zona deben recibir amplia información sobre el curso de los trabajos, incidencias, rendimiento y producción de equipos, estado de mediciones y valoraciones etc. Autorizarán expresamente las principales operaciones que se realicen en las obras : contrataciones, gastos extraordinarios, sistemas de incentivos, etc.

EJEMPLOS DE ESTRUCTURACIÓN DE EMPRESAS

Los organigramas deben adaptarse a las necesidades particulares de organización y funcionamiento de cada empresa.

En empresas de volumen medio, prácticamente se tocan las mismas materias y se realizan - en menor escala - las mismas actividades de las empresas grandes. Pero la concentración de misiones suele ser proporcionalmente inversa al tamaño. Se van suprimiendo escalones intermedios, y en un mismo departamento se agrupan diversos cometidos, que en las grandes empresas son desempeñados con mayor grado de descentralización y de especialidad.

Las estructuras de los servicio centrales de las empresas de categoría media, pueden quedar limitadas a los siguientes órganos y secciones principales :

Dirección :

- **Secretaría.**
- **Oficina técnica:** Proyectos. Cálculos. Delineación.
- **Estudios de obras.** Licitaciones.
- **Planificación y control general de los trabajos.**
- **Control y análisis de costos.**
- **Administración general.** Intervención, Inspección.
- **Asesoría.** Trámites. Contratos. Subcontratos.
- **Tesorería :** Caja principal. Bancos. Letras.
- **Inspección médica y servicios sanitarios.**
- **Compras.** Almacén general.
- **Administración de maquinaria :** Parque. Talleres.
- **Personal.**
- **Contabilidad general.**

En una creciente simplificación de funciones y concentración de servicios, la pequeña empresa presenta una sencilla estructura de organización, ya que, proporcionalmente a su capacidad, las variadas actividades de dirección y administración que les corresponden, son desempeñadas por las pocas personas que llevan toda la gestión de la empresa, con participación personal, incluso, del propio director o empresario en muchas de estas funciones.

Un ejemplo de estructuración de una empresa pequeña es el siguiente .

Director empresario:

- Oficina técnica:

- Estudios. Cálculos.
- Delineación.

- Oficina administrativa.

- Administración. Compras. Personal.
- Contabilidad.
- Caja.
- Mecnografía. Archivo.

- Jefe o jefes de obras:

- Obra 1 - 2- ... 20.

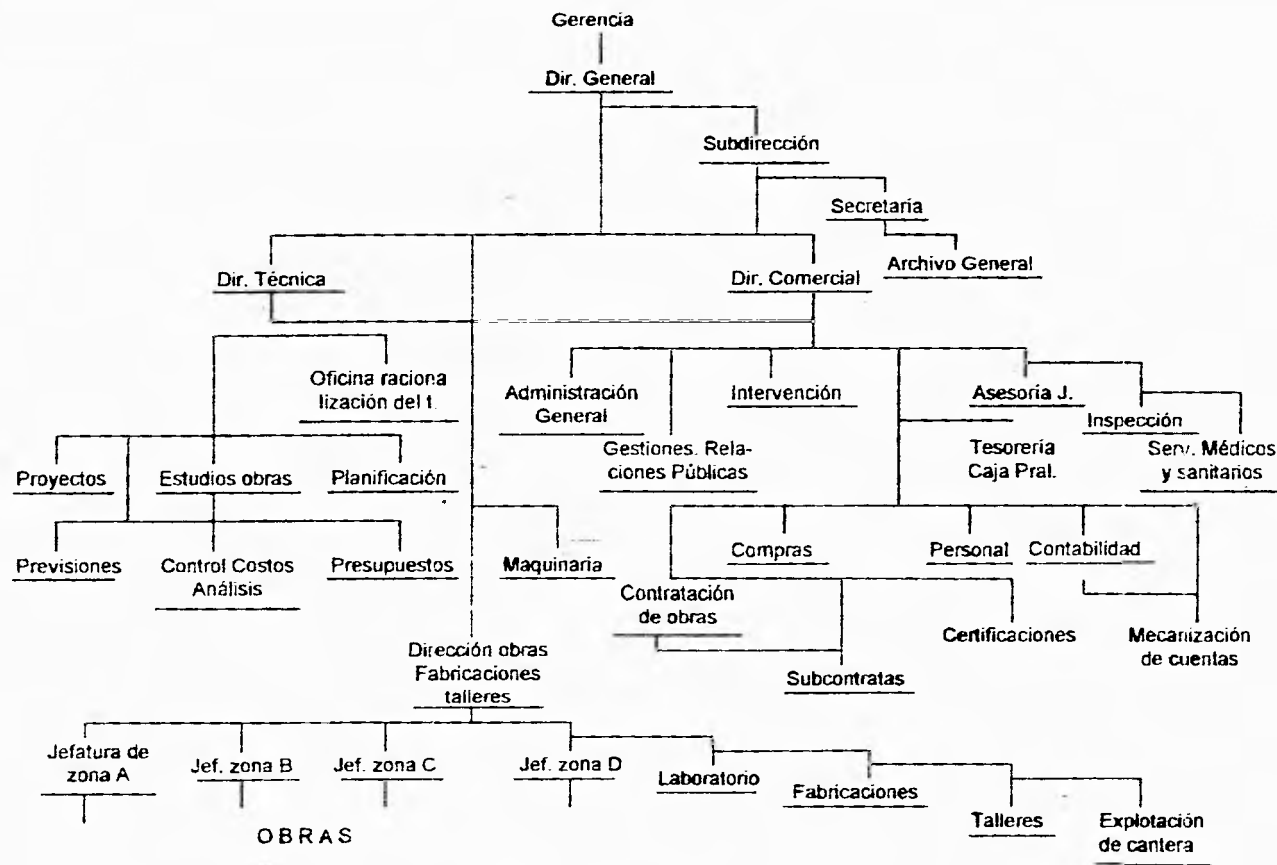


Fig. 1.1.1.1. Organigrama de una empresa grande.

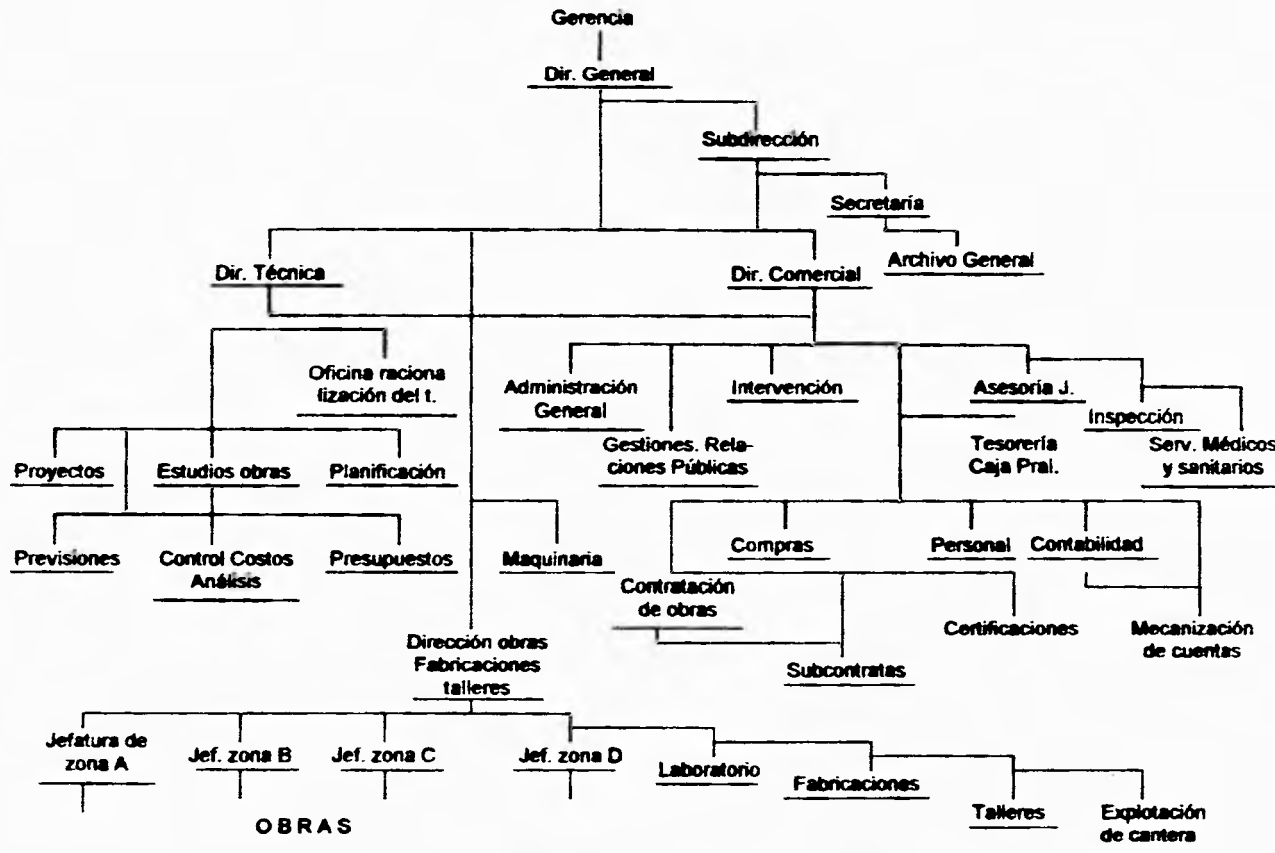


Fig. 1.1.1.1. Organigrama de una empresa grande.

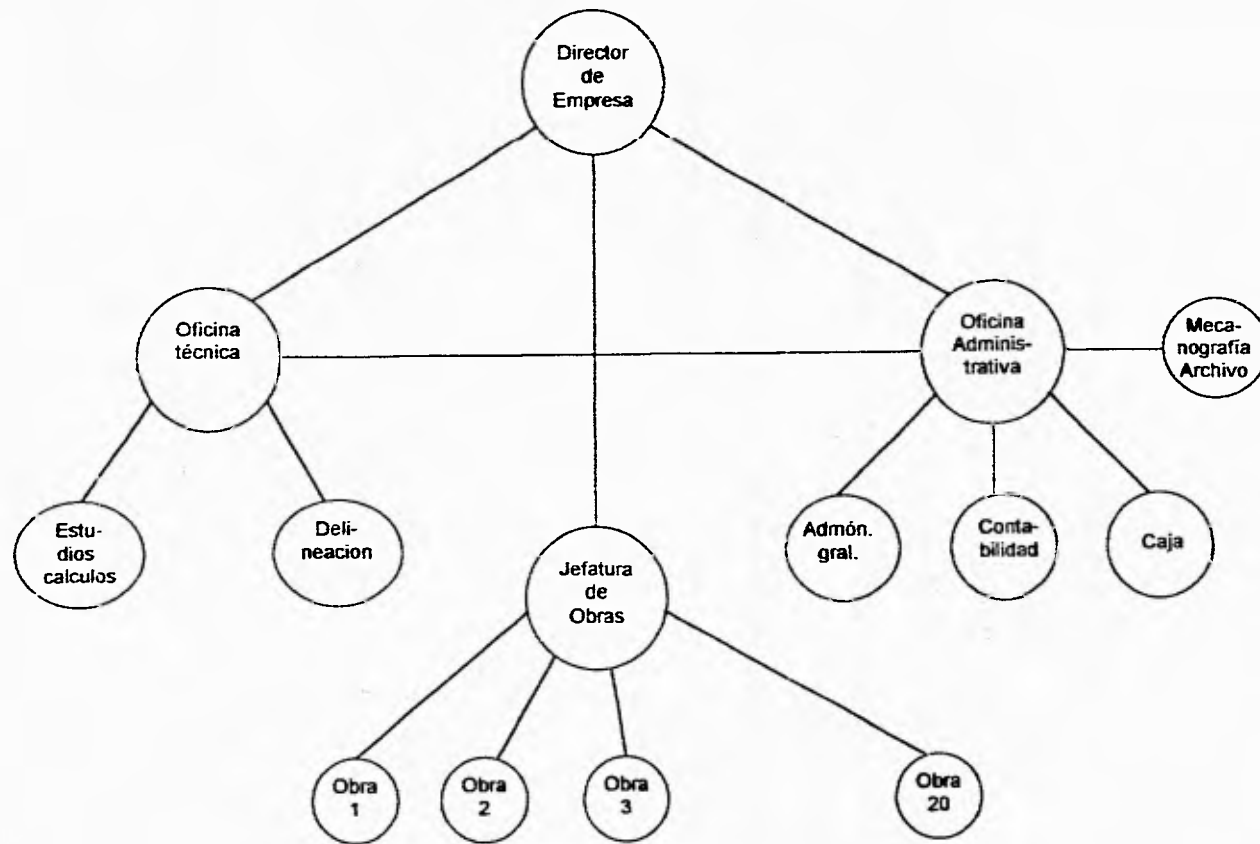


Fig. 1.1.1.3 Organigrama de una empresa pequeña.

ORGANIZACIÓN FUNCIONAL DE LAS OBRAS

El éxito de la gestión empresarial depende de la mejor organización y el desarrollo óptimo de actividades de los centros productores. Es por ello imprescindible atender a una perfecta estructuración de funciones de los servicios de obras y dotarlos de los medios y métodos de actuación más convenientes y eficaces, en consonancia con el volumen y ritmo de los trabajos, sus características propias, ubicación de las obras y demás circunstancias particulares.

Los nexos de unión jerárquica, están formados por los jefes de obras, responsables directos y principales de todo el desarrollo de los trabajos. A continuación se definirán las principales funciones que incumben a los servicios de obras.

JEFES DE OBRAS

Asumen la principal autoridad y responsabilidad en la dirección de las obras que les son asignadas. Su misión fundamental es velar por el mejor desarrollo y la perfecta ejecución técnica de los trabajos, dentro de los plazos establecidos y los costos más favorables.

Ejercen mando directo sobre los servicios ejecutivos de la obra, a cuyo frente se encuentra el encargado o encargados, seguidos de los capataces y jefes de equipo o de cuadrilla. Bajo sus ordenes funcionan también los servicios técnicos como: topografía, replanteos, mediciones, ensayos y dosificación de materiales.

Los restantes servicios, de maquinaria, transportes, talleres y los de administración y control, quedan subordinados a la máxima autoridad del jefe de obra.

Las principales obligaciones en colaboración con los servicios de obra son:

- Mantener la organización y disciplina de todos los servicios, cuidando de su buen funcionamiento.
- Colaborar activamente en la planificación y programación de los trabajos.
- Deben instruir al personal que tienen bajo sus órdenes directas en las diferentes misiones y trabajos a realizar.
- Dirigen y comprueban los trabajos de replanteos y las instalaciones de obra.
- Intervienen y controlan la subcontratación de partes de obra. Realizan o comprueban mediciones y autorizan la facturación de trabajos y suministros.
- Están facultados para conceder premios e imponer sanciones o, en ciertos casos, proponerlos a los órganos superiores.
- De acuerdo con una planificación de las necesidades de recursos y medios de trabajo, derivada de los programas de obra, realizarán oportunamente los pedidos detallados de materiales, maquinaria, transporte y elementos auxiliares.

ENCARGADOS

Tienen el mando directo sobre todo el personal operario de la obra o partes de obra en que ejercen sus funciones. Forman el enlace entre obreros y personal directivo.

Poseen los conocimientos técnicos, teóricos y prácticos, que les capacitan en sus variadas funciones de organización, dirección y vigilancia de los trabajos; replanteos,

nivelaciones, mediciones de obra, interpretación de planos; el empleo de la maquinaria, medios auxiliares; comportamiento y dosificación de materiales, formación y rendimiento de equipos; tareas y destajos; normas de seguridad e higiene del trabajo.

Reciben ordenes de los jefes de obra con quienes colaboran en todo momento.

Algunas veces, por la propia estructura de las empresas, especialmente las pequeñas, el distanciamiento de las obras u otras circunstancias, los encargados actúan al mismo tiempo como jefes de obra. Algunas de sus funciones son las siguientes:

- Dirigir de forma directa la realización de los trabajos bajo su control, ayudándose de los capataces, jefes de equipo, y auxiliares técnicos, a quienes instruyen en sus diversos cometidos. Imponen orden y disciplina en todas las tareas.
- Preparan y distribuyen el trabajo de hombres y máquinas; vigilan su perfecta ejecución, seguridad y garantía de los trabajos.
- Controlan y valoran los tiempos y rendimientos, con los que establecen las bases de estimación para fijar o proponer las primas, gratificaciones u otros incentivos.

CAPATACES, JEFES DE TALLER Y DE EQUIPO

Estos empleados constituyen la ramificación directiva extrema, el puente final entre los servicios de dirección ejecutiva y el personal obrero.

A veces realizan los mismos trabajos que los obreros de sus cuadrillas, especialmente los jefes de equipo, cuya misión de mando es ocasionalmente temporal.

SERVICIOS DE ADMINISTRACIÓN Y CONTROL

Estos servicios comprenden la dirección administrativa, contabilidad general y analítica de las obras, servicios de compras, almacén, personal, caja y pagaduría, correspondencia, mecanografía y archivo. Sus principales funciones son las siguientes:

- Se ocupan de contratación, encuadramiento y control de la mano de obra, su ordenamiento legal y disciplinario.
- Preparación y realización de pagos al personal, seguros sociales, gastos generales. Tramitación de los asuntos legales y sociales.
- Registro de clasificaciones de datos, operaciones de contabilidad, control de producción y de costos, ficheros, formalización de inventarios.
- A través de la oficina administrativa se formulan y tramita toda clase de documentos.

ORGANIGRAMAS

A continuación se presentan dos organigramas que describen la organización funcional de una obra.

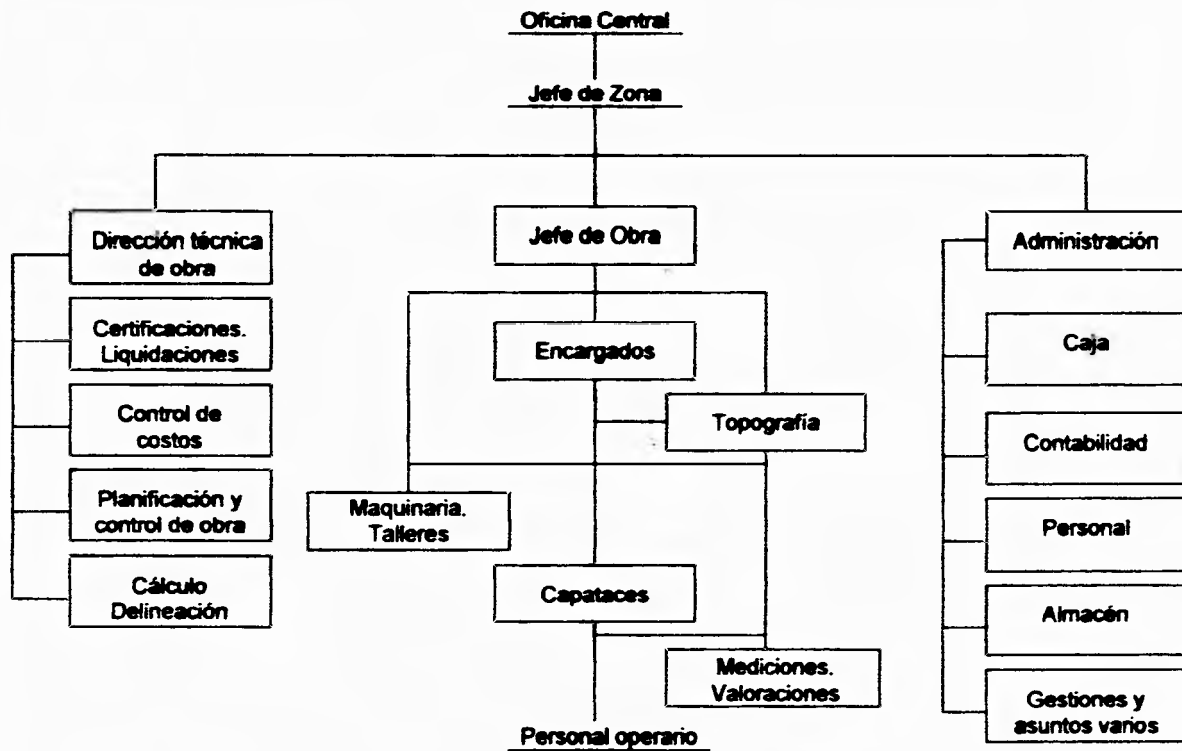


Fig. 1.1.1.4 Estructura básica y relaciones de interdependencia de los servicios correspondientes a un centro de trabajo de obras públicas.

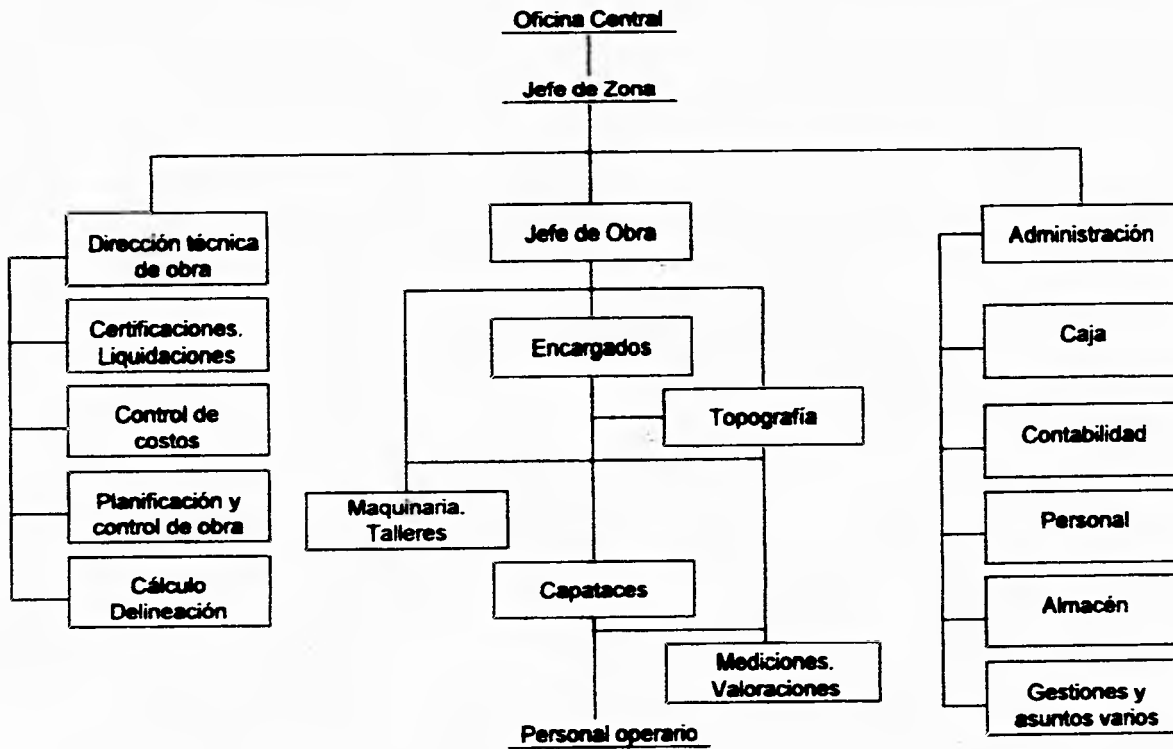


Fig. 1.1.1.4 Estructura básica y relaciones de interdependencia de los servicios correspondientes a un centro de trabajo de obras públicas.

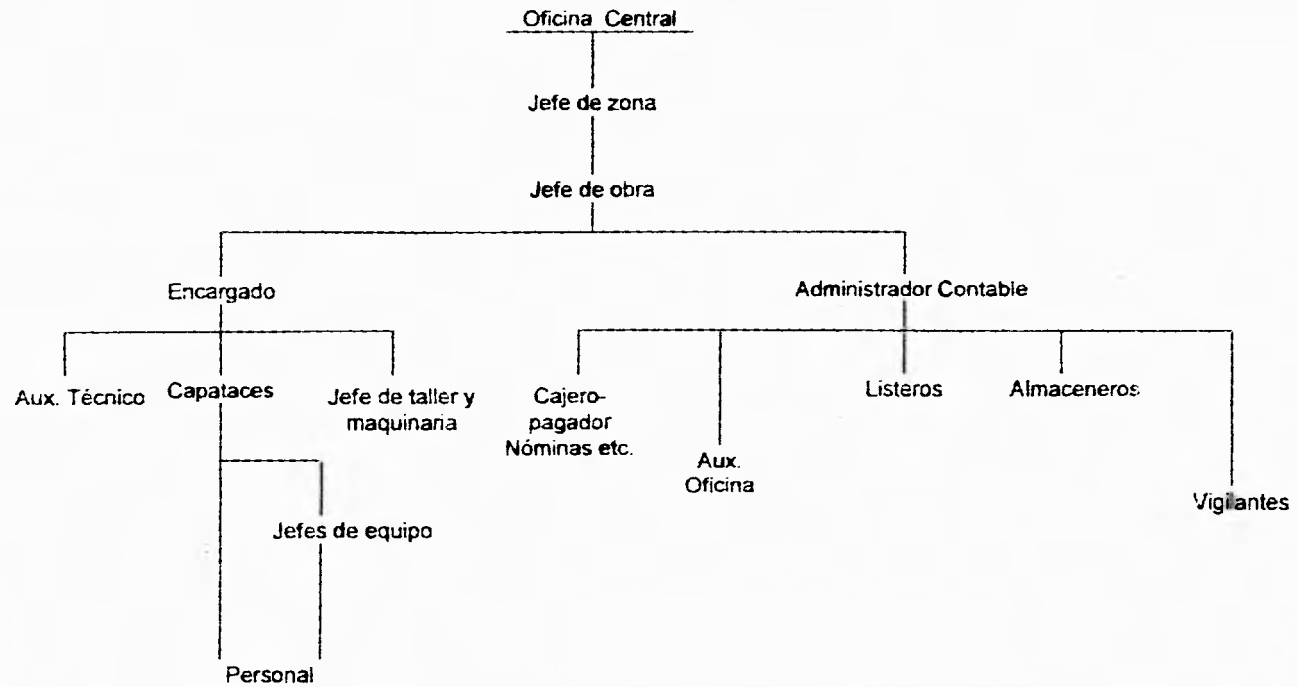


Fig. 1.1.1.5. Organigrama de una obra en general.

1.1.2. CONCEPTOS BÁSICOS DE PRECIOS UNITARIOS

La elaboración de los precios unitarios, es una etapa dentro del proceso constructivo general, que se inicia con la investigación o estudio de la factibilidad de realizar una obra, y que termina con la construcción, operación y mantenimiento de la misma.

Antes de la elaboración de los precios unitarios, es indispensable conocer a fondo, la naturaleza de los recursos, tanto humanos, como de maquinaria y materiales, así como la disponibilidad de los mismos.

Algunas definiciones utilizadas en el manejo de precios unitarios son las siguientes:

PRECIO UNITARIO

Es la remuneración o pago en moneda, que el contratante cubre al contratista, por unidad de obra y por concepto de trabajo que ejecute, de acuerdo a las especificaciones.

UNIDAD DE OBRA

Es la unidad de medición señalada en las especificaciones, para cuantificar el concepto de trabajo con fines de medición y pago.

CONCEPTO DE TRABAJO

Es el conjunto de operaciones manuales y mecánicas que el contratista realiza durante la ejecución de la obra, de acuerdo a planos y especificaciones, divididas convencionalmente para fines de medición y pago; incluyendo el suministro de los materiales correspondientes cuando éstos sean necesarios.

ESPECIFICACIONES

Son el conjunto de requerimientos exigidos en los proyectos y presupuestos para definir con precisión y claridad el alcance de los conceptos de trabajo. Las especificaciones de un concepto en particular deben contener las siguientes definiciones:

- a) Descripción del concepto.
- b) Materiales que intervienen y su calidad.
- c) Alcance de la ejecución del concepto.
- d) Mediciones para fines de pago.
- e) Cargos que incluyen los precios unitarios.

En términos generales, los elementos que componen un precio unitario son :



La utilidad será la ganancia que debe considerar cada empresa contratista, como resultado a sus esfuerzos técnicos, administrativos y económicos, para cumplir con la realización de un proyecto.

MATERIALES

El conocer ampliamente el material es de gran utilidad para seleccionar los más adecuados a las condiciones de trabajo, de servicio (calidad) y acordes con los límites económicos.

El costo del material que se toma como base para integrar el precio unitario de un concepto, está compuesto por el precio de adquisición, porcentaje de flete y porcentaje de descuento.

Existe una gran variedad de precios de adquisición de un mismo tipo de material debido a la calidad, cercanía del consumidor con respecto a la fuente de origen del material, el precio también varía con el volumen de compras del consumidor, ya que si es muy grande se obtendrán mejores precios y condiciones de pago.

La abundancia y escasez de los materiales depende directamente de la demanda en el mercado. El monto del costo de las operaciones de carga, descarga y flete dependen primordialmente de la distancia de la fuente de suministro a la fuente de consumo del material, y del procedimiento que se siga para la carga y descarga del mismo.

El costo debe integrarse al precio de adquisición para obtener el costo del material en obra.

MANO DE OBRA

La mano de obra es el costo del personal empleado directamente para la producción de una unidad de obra. El costo de la mano de obra es el resultado de prorratear el pago de los salarios al personal que interviene directamente en la ejecución de los trabajos necesarios del concepto por analizar .

Los sistemas que manejan en la industria de la construcción para cubrir el trabajo son los siguientes :

- a) Por Día: Cuando debe darse al trabajador una cantidad fija por jornada normal de trabajo.
- b) Por Destajo: Si la remuneración se define en base a las unidades de trabajo ejecutadas y afectadas en un precio previamente acordado.

- c) **Por Tarea:** Consiste en la asignación de un trabajo determinado por día, al ejecutar el trabajador las tareas asignadas, podrá retirarse, recibiendo su jornal diario completo.

Para determinar cuál es el más adecuado en cada caso se debe estudiar y analizar las condiciones y tipo de trabajo por realizar.

En algunos campos de la construcción la mano de obra representa un alto porcentaje del costo total de una obra; esto hace importante el estudio detallado y metódico de los factores que integran dicho costo, es decir, todas aquellas erogaciones que el constructor tiene que realizar para remunerar la fuerza de trabajo aportada por un obrero.

Equipo

Representa el conjunto de máquinas y equipo del que se sirven las industrias constructoras para realizar su trabajo.

En el mercado de la construcción existe una gran variedad de maquinaria y equipo de diferentes marcas, modelos, capacidades y especificaciones de calidad. Por lo tanto es necesario realizar estudios cuidadosos a fin de determinar cuál es la maquinaria más conveniente para la óptima ejecución de la obra.

ADMINISTRACIÓN DE OBRA

Los conceptos que constituyen este grupo son los siguientes:

- **Honorarios, sueldos y prestaciones:** Este aspecto cubre todas las erogaciones originadas por el personal técnico-administrativo que dirige y supervisa la ejecución de los trabajos.
- **Instalaciones y obras provisionales:** En este aspecto se incluyen todas las erogaciones relativas a la construcción de obras e instalaciones auxiliares, necesarias para el desarrollo de la obra misma, como pueden ser : Instalaciones eléctricas, hidráulicas, sanitarias, de gas, señalamientos, casetas de vigilancia, Instalaciones para servicios médicos.
- **Transporte, fletes y acarreos:** En este aspecto se agrupan los gastos originados por consumos y amortización de vehículos del servicio general de la obra. Fletes de materiales y equipo no incluidos en el costo directo.
- **Gastos de oficina:** Papelería, correo, situaciones bancarias, copias, suscripciones y cuotas, envíos, pasajes y transportes locales etc.
- **Varios:** Sindicatos, control de calidad, intercomunicación, desmantelamiento, rupturas y reposiciones.

ADMINISTRACIÓN CENTRAL

Los gastos totales de la administración se pueden agrupar de la siguiente manera:

- **Gastos técnicos y administrativos:** Cubren esencialmente los sueldos y honorarios del personal técnico administrativo, así como por servicios de asesoría contable, fiscal, laboral y otras.
- **Depreciaciones, alquileres y mantenimientos:** Los espacios físicos que ocupa la organización para operar, sean propios o rentados, generan gastos tanto para utilizarlos como para mantenerlos en buenas condiciones.
- **Obligaciones y seguros:** Son los gastos que se originan en cumplimiento de las disposiciones de la ley así como para proteger a los elementos de la organización de algunos riesgos a los que están sujetos.
- **Materiales de consumo:** Son los gastos para la adquisición y uso de materiales necesarios en la operación de la empresa.
- **Capacitación y promoción:** por ley, los empleados y trabajadores tienen derecho a capacitarse y, por otro lado, la organización requiere promocionarse para aumentar su mercado de trabajo.
- **Previsiones:** Con objeto de que la organización esté en posibilidad de hacer frente a situaciones de emergencia o planeadas se consideran como provisiones los períodos de inactividad, servicios médicos de emergencia, donativos e indemnizaciones.

FINANCIAMIENTO

Es un factor de costo de vital importancia, cuya imprevisión puede tener graves consecuencias en los resultados finales de una obra, y ocasionar serias pérdidas.

El monto de los financiamientos dependerá, en cada caso particular, de la relación que exista entre el programa previsto de erogaciones y el programa esperado de ingresos, dependiendo el primero del programa general de obra y el segundo de la forma de pago establecida en el contrato.

FIANZAS Y SEGUROS

En este grupo se involucran todas las erogaciones motivadas por los aspectos de fianzas, seguros, multas, recargos, regalías por el uso de patentes, etc.

IMPREVISTOS

Categoricamente, en todo trabajo de construcción, existen acusas o elementos de costo que no pueden ser expresados en números. No se pueden suprimir totalmente los errores, tanto en estimación como en el proceso de ejecución.

El criterio de estimación de imprevistos consistirá en tratar de presuponer con alguna base razonable, los cargos de previsión para el mayor número posible de contingencias, reduciendo a un mínimo aceptable el factor marginal que se supone servirá para cubrir en alguna proporción los riesgos verdaderamente imprevistos.

UTILIDAD

En la determinación de la utilidad, las empresas deben considerar el pago de impuestos a que está sujeta, así como el efecto de la inflación. La utilidad se expresa como un porcentaje de la suma del costo directo total y de los costos indirectos.

1.1.3. CONCEPTOS BÁSICOS DE CONTROL DE OBRA

La previsión y el control de la gestión son piezas esenciales en los sistemas de una eficiente organización de empresas.

Se apoyan en métodos ordenados de funcionamiento que permiten determinar y controlar los medios necesarios para que se cumplan objetivos concretos de los diversos planes de actuación.

En la ejecución de las obras, la previsión complementa y hace posible el normal desarrollo de los planes de trabajos, imprimiendo orden y coordinación a la aportación y distribución de los recursos en el lugar y tiempo apropiados.

El control es esencial en dos sentidos: para vigilar el cumplimiento de las previsiones y para ajustar éstas a las diferentes alteraciones que vayan produciendo las replanificaciones sucesivas.

PREVISIÓN DE OBRAS

Para lograr los objetivos definidos, la previsión debe ir unida a la planificación, a fin de que pueda hacerse en cada etapa o fase de obra la selección de los procedimientos de trabajo más convenientes y una adecuada previsión de los recursos a emplear:

- Equipos mecánicos. Instalaciones.
- Mano de obra.
- Materiales. Medios auxiliares. Herramienta.

- Transportes.
- Subcontratos.

Además de esta previsión de recursos, son necesarias las previsiones de tipo económico, unas que afectan globalmente al desarrollo de toda la obra, en el tiempo completo de ejecución, y otras relacionadas a plazos a ciclos de trabajo:

1. Previsión presupuestaria general:

- Costos directos.
- Costos indirectos.
- Gastos generales.
- Impuestos y otras cargas legales derivadas del contrato.

2. Previsión del volumen de obra a realizar. Inversiones.

3. Previsión de certificaciones o facturaciones (cobros).

4. Financiación de la obra.

Conjugando la previsión con una sistemática y bien ideada planificación de actividades, se consiguen resultados muy beneficiosos para la economía de las obras.

Los fallos en las previsiones originan muchas veces la desorganización de los trabajos, incrementan los costos de las obras y ponen en peligro el cumplimiento de los planes establecidos.

PREVISIÓN DE RECURSOS

La previsión de materiales principales de gran consumo - hierros, cementos, cerámicas y otros - debe hacerse siempre bajo los planes de obra, a largo plazo, para que el departamento de compras cuente con tiempo para recibir ofertas y seleccionar precios, calidades, condiciones de suministro y de pago.

La determinación de las cantidades de materiales se obtiene de los volúmenes de obra a realizar. El ritmo de suministro lo da la duración de las respectivas actividades, contando siempre con algún adelanto o almacenamiento para cubrir las posibles contingencias.

PREVISIÓN PRESUPUESTARIA

Comprende dos partes:

1. Presupuesto de inversiones o de costos: Es el presupuesto de estudio, que está formado por los costos de ejecución material - directos e indirectos - más los gastos generales previstos y los impuestos y tasas que gravan la ejecución de la obra.
2. Presupuesto de contrata: Debe entenderse en sus precios e importes líquidos aplicables a las certificaciones o facturaciones de trabajos.

PREVISIÓN FINANCIERA DE OBRA

En obras de contrata, el adelanto se origina en los gastos iniciales de gestión, preparación y puesta en marcha de la obra que, si son de cuantía se consideran inversiones a amortizar durante todo el curso de los trabajos. La financiación también es motivada por el capital inmovilizado en los depósitos de finanzas y, por el retraso en las certificaciones de obra, a causa de agotamientos de créditos, tramitación de reformados y presupuestos adicionales.

CONTROL DE OBRAS

El control de obras tiene como consecuencia vigilar en todo momento su cumplimiento, comprobar resultados, rectificar, donde sea preciso, los planteamientos iniciales y ajustarlos a las nuevas situaciones y, en lo posible corregir las desviaciones negativas que se originen.

Se debe mantener un control permanente de la situación en la que cada obra se encuentra. Estas situaciones pueden ser:

1. Obras adjudicadas provisionalmente, pendientes de :
 - 1.1. Constitución de fianza.
 - 1.2. Formalización de escritura de contrata.
 - 1.3. Presentación de planes de trabajo.

2. Obras adjudicadas definitivamente, no empezadas pendientes de:

- 2.1. Tramitación de crédito.
- 2.2. Replanteo general.
- 2.3. Expropiaciones y otras causas.

3. Obras en marcha:

- 3.1. En curso normal.
- 3.2. parcialmente paralizadas.

4. Obras paralizadas:

- 4.1. Por agotamiento de crédito.
- 4.2. Pendientes de reformado.
- 4.3. Orden de suspensión temporal.

5. Obras terminadas:

- 5.1. Resolución de contrato.
- 5.2. Terminación total de trabajos.
 - 5.2.1. Pendientes de recepción provisional.
 - 5.2.2. Pendientes de recepción definitiva.
 - 5.2.3. Pendientes de liquidación provisional.
 - 5.2.4. Pendientes de liquidación definitiva.
 - 5.2.5. Pendientes de cancelación de fianza.

En los aspectos técnico-económicos, el control de la ejecución de obra cumple los fines más importantes. Se basa en las mediciones y valoraciones periódicas de la producción, de donde se obtienen datos esenciales para certificaciones y liquidaciones de trabajos, facturaciones de contratistas y suministros, el control de provisiones, control de costos y el control contable.

DIAGRAMA DE CONTROL DE OBRA

Permite seguir la marcha de los trabajos en clara expresión gráfica, señalada en espacios-volúmenes. Al mismo tiempo relaciona la previsión con la ejecución parcial o total, reflejando en un orden de magnitud, el avance o retraso respecto al ritmo previsto para cada una de las citadas unidades de obra en los correspondientes ciclos de control.

CONTROL DE PRESUPUESTOS

Es necesario mantener un control permanente de los presupuestos activos de obras, de los importes realizados y los que queden pendientes de realizar, así como de los importes de certificaciones expedidas con cargos a los presupuestos y los saldos resultantes en cada situación. En la siguiente figura se muestra un ejemplo de control de presupuestos en el cual se obtienen: volúmenes de obra contratada, importe de la obra ejecutada a precios de proyecto o contrato, volumen de obra contratada pendiente de ejecutar, importe de la obra certificada, volumen de obra pendiente de certificar y saldos.

Obra (Designación o clave)	(a) Presupuest o total líquido	(b) Importe de obra ejecutada a precios líquidos de proyecto	(c) importe de obra pendiente de ejecutar (a - b) = ±	(d) Importe de certificaciones o facturaciones expedidas	(e) Obra pendiente de certificar (a - d)	(f) Saldo o diferencia entre importes obra ejecutada y certificada: (b - d) = ±
Sumas						

(a) Deberá recoger también los presupuestos adicionales de obra, compensación, revisión, etc.

(c) La diferencia puede ser negativa cuando el importe de la obra realizada sobrepase el presupuesto, lo que será motivo de aumento de presupuesto o de liquidación adicional.

(f) Saldo o diferencia que, excepcionalmente, puede ser negativa cuando se dé un adelanto de las certificaciones.

Nota : Este estado debe referirse a todas las obras en curso de ejecución, incluso a las terminadas pendientes de liquidación.

Fig. 1.1.3.2 Ejemplo de control de presupuestos.

PREVISIÓN Y CONTROL A ESCALA GENERAL

Como las actividades de las empresas constructoras se desarrollan en proyección horizontal, a través de sus diseminados centros de trabajo, siempre temporales, la previsión y el control, a escala general, se basa principalmente en la recopilación de datos emanados de todas las obras, lo mismo las que están en curso de ejecución que las que van incorporándose al cuadro general de contrataciones.

Es necesario llegar a esta previsión y control general, ya que la gestión de la empresa ha de ir planificada como un todo, en una política de visión y ordenación del conjunto.

Lo más importante de toda previsión económica y financiera, está en el volumen de actividades a desarrollar en un período dado, medido en inversiones probables y cobros previstos.

De ahí que sea importante establecer una previsión general de obras, dándole el detalle necesario para que el control pueda localizar en cada ciclo las posibles desviaciones, a la vez que se presenten resumidas, en cifras globales, toda la previsión y la realización.

1.1.4 CONCEPTOS BÁSICOS DE PLANEACIÓN DE PROYECTOS

Una vez terminados los cálculos de presupuestos y costos unitarios en el proceso constructivo, se requiere hacer la programación de actividades a realizar para llevar a buen término dicho proyecto, en esta parte del proceso es necesario asignar los tiempos en los cuales deben ejecutarse los volúmenes de obra y en base a ello es posible obtener información teórica que permitirá conocer información de suma relevancia para prever las cantidades de insumos a utilizar en un determinado periodo, así como el flujo de efectivo que tendrá el proyecto. Para conocer de manera más específica la parte de planeación es necesario ir de los conceptos generales a los particulares, es decir iniciar de un breve análisis de la administración como punto de partida para ubicar a la planeación, mencionando cuales son las técnicas utilizadas en esta parte del proceso administrativo y cuales son sus relaciones con una constructora en la planeación de proyectos.

ADMINISTRACIÓN

El proceso administrativo que empezó como una actividad empírica se ha ido desarrollando en base a una serie de acontecimientos que lo han transformado de manera tal que en la actualidad no es una simple actividad sino un conjunto de técnicas que permiten dividir cualquier organismo en áreas funcionales semi-autónomas que van a ser regidas por un poder central, con el fin de satisfacer ciertas necesidades.

Al desarrollarse el capitalismo la administración alcanza su máximo crecimiento. Esto no quiere decir que éticamente sea mejor si no que se ha buscado hacerla más eficiente.

Podemos decir que la administración es un conjunto de normas y técnicas que permiten determinar los pasos a seguir para alcanzar un objetivo.

Para conceptualizar de manera más precisa lo que representa un proceso administrativo tenemos en la siguiente figura de manera gráfica una parte del proceso administrativo ubicado en un proyecto de obra.

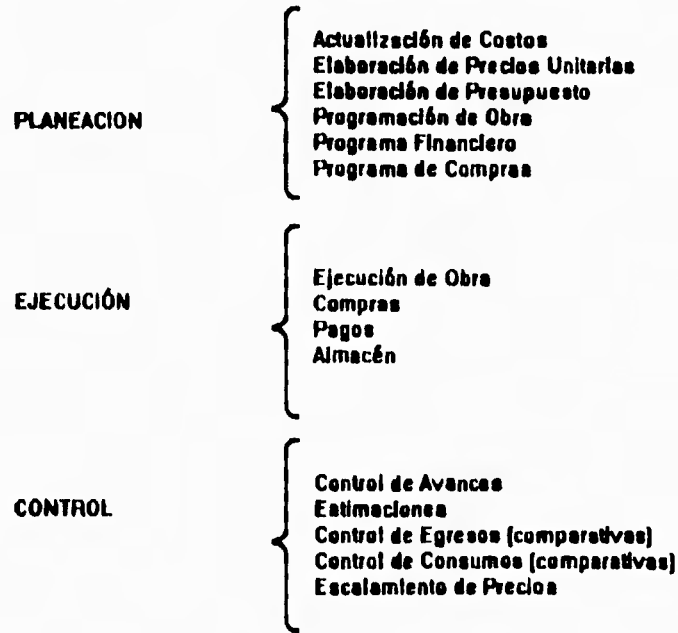


Fig. 1.1.4.1 Proceso administrativo aplicado a un proyecto constructivo.

Es importante hacer notar que en la figura , se señalan los puntos más importantes de dicho proceso, los cuales nos sirven de base para ubicar al proceso de planeación y así determinar sus objetivos y funcionamiento.

PLANEACIÓN

Una vez ubicada la planeación como un elemento del proceso administrativo es importante encontrar una definición objetiva, para lo cual después de tomar en cuenta algunas de las ya existentes, podemos tomar la proporcionada por Agustín Reyes Ponce (autor de libros de administración) que nos da una idea real de la importancia de ésta área en la Construcción, la cual queda definida como "Planeación es fijar el curso concreto de acción que ha de seguirse, estableciendo los principios que habrán de orientar la secuencia de operaciones para realizarla y las determinaciones de tiempos y números necesarios para su realización".

La planeación se encuentra basada en 3 principios, los cuales de alguna manera reflejan un sentido administrativo, pero que son aplicables a la planeación de proyectos. A continuación se mencionan estos principios de planeación dando una breve descripción de su significado con el objetivo de mostrar los factores que deben cuidarse al realizar la planeación de proyectos y para más adelante explicar las diferentes técnicas y procedimientos utilizables en una construcción.

Probabilidad: Deben establecerse aquellos objetivos que tengan un mayor grado de probabilidad con el fin de buscar una mejor aproximación de nuestra planeación a la realidad. Los grados de probabilidad se establecen en función a la experiencia y/o investigación previa.

Objetividad: Los objetivos deben ser determinados de manera concreta y clara, para así poder determinar los niveles de probabilidad.

Flexibilidad: Quizá sea este uno de los puntos más relevantes, debemos recordar que la información obtenida de la planeación es teórica y nos permitirá hacer previsión para el período de ejecución de una obra. Por lo tanto la planeación debe ser flexible para poder manejar secuencias, tiempos y costos con márgenes de variación, es decir la existencia de rangos en los que pueda moverse el proyecto asegurando el éxito de éste.

TÉCNICAS DE PLANEACIÓN

Anteriormente se ha dado la definición de planeación y se ha presentado el objetivo que ésta tiene en un proceso de construcción, ahora debemos señalar los procedimientos y técnicas que hacen posible la elaboración programas, presupuestos, y pronósticos.

Existen un número ilimitado de técnicas, cada proceso específico puede ir cubriendo sus necesidades desarrollando nuevas técnicas o modificando las ya existentes, en la siguiente figura se mencionan algunas de las más importantes clasificadas de acuerdo a su utilización.

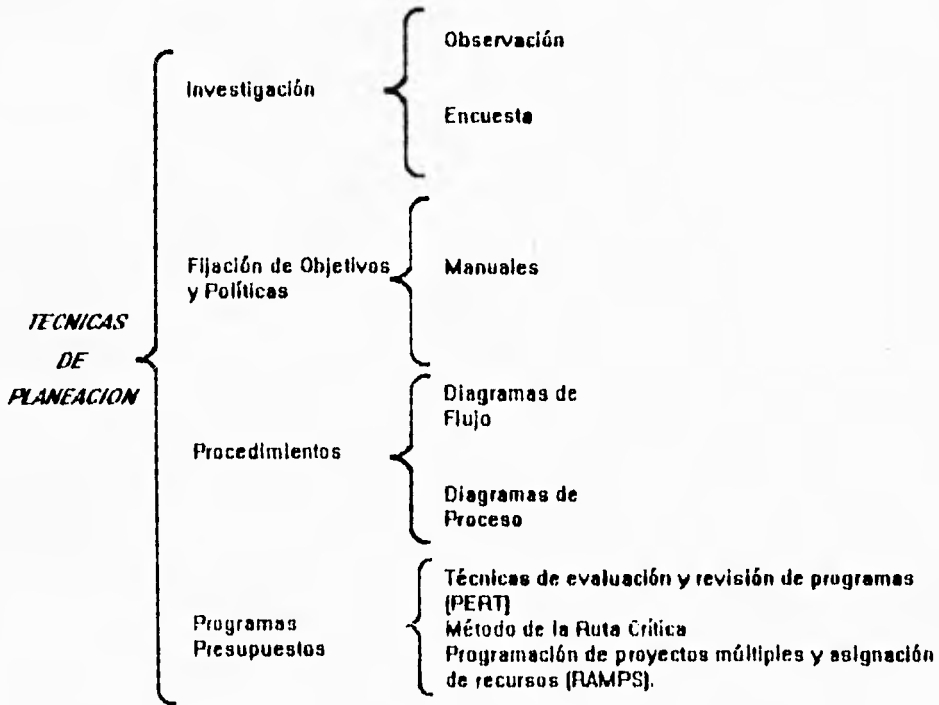


Fig. 1.1.4.2 Técnicas de planeación más relevantes.

Dentro del proceso constructivo las técnicas aplicables son las mostradas en la figura para programas y presupuestos, ya que precisamente el objetivo es programar los presupuestos elaborados. Para ello mencionaremos las características:

Estas técnicas son conocidas como de trayectoria crítica, son el instrumento básico de la planeación ya que permiten una descripción mucho más clara de los programas y presupuestos, además de permitir fijar y revisar costos, tiempos, secuencias y relaciones que son básicamente las necesidades de una constructora durante la ejecución de cada obra. Entre las técnicas más importantes se encuentran las siguientes:

PERT (Program Evaluation and Review Technique) Es decir, técnica de evaluación y revisión de programas. Esta técnica se encuentra basada en una red de eventos a los cuales se les asigna un orden de ejecución y una duración, de lo cual pueden obtenerse 3 tiempos: el óptimo, el normal programado y el pésimo.

CPM (Critical Path Method) Es decir, método de la ruta crítica, esta técnica es parecida a la anterior, la diferencia radica en que ahora se introducen los costos de actividades, buscando optimizar duraciones y costos, es decir, encontrar el punto medio de ambos.

RAMPS (Resource Allocations and Multiproject Shedulling) Es decir, programación de proyectos múltiples y sus recursos. Se apoya en las técnicas PERT y CPM y el objetivo principal es programar la asignación de recursos a distintos proyectos de manera simultánea por lo debe tenerse en cuenta un conjunto de redes que interactúan, comparten recursos, y se programan en

función a los recursos disponibles. Normalmente esta técnica es utilizada en empresas con diferentes proyectos simultáneos que tienen la necesidad de compartir recursos.

A continuación se presenta un ejemplo muy sencillo en el cual se aplican las 3 técnicas mencionadas anteriormente, debe notarse que la diferencia entre ellas consiste en la información que interviene y la información que proporcionan como resultado.

Consideremos el siguiente concepto, que para su realización requiere de cubrir con un número reducido (a manera de ejemplo) de actividades, las cuales se encuentran cuantificadas.

CIMENTACION DE EDIFICIO.

• EXCAVACION CEPA 2.0 M PROF. MATERIAL II.	53.7	M3
• ACARREO CARRETILLA MATERIALES A 20 M.	40	M3
• ZAPATA CORRIDA CIMENTACION 0.6 M ANCHO.	121.4	ML
• CONTRATRABE CIMENTACION 20 X 80 CM	19.8	ML
• DALA DE ENRASE 15 X 30CM 4 VARS 3/8"	121.4	ML
• IMPER. DALAS FRIO C/CARTON ASFALTICO.	54.64	M2

Aplicando la 1a técnica, únicamente debemos asignar el orden de ejecución de las actividades, así como su duración. En este caso el volumen de actividad a realizar nos ayuda a determinar el rendimiento con el cual se realiza cada una de las actividades.

CIMENTACION DE EDIFICIO.

Actividad	Volumen	Unidad	Duración Pésima	Duración Normal	Duración Óptima
• EXCAVACION CEPA 2.0 M PROF. MATERIAL II.	53.7	M3	10	8	7
• ACARREO CARRETILLA MATERIALES A 20 M.	40	M3	8	8	5
• ZAPATA CORRIDA CIMENTACION 0.6 M ANCHO.	121.4	ML	7	5	4
• CONTRATRABE CIMENTACION 20 X 80 CM	19.8	ML	9	7	6
• DALA DE ENRASE 15 X 30CM 4 VARS 3/8"	121.4	ML	6	4	3
• IMPER. DALAS FRIO C/CARTON ASFALTICO.	54.64	M2	5	3	2

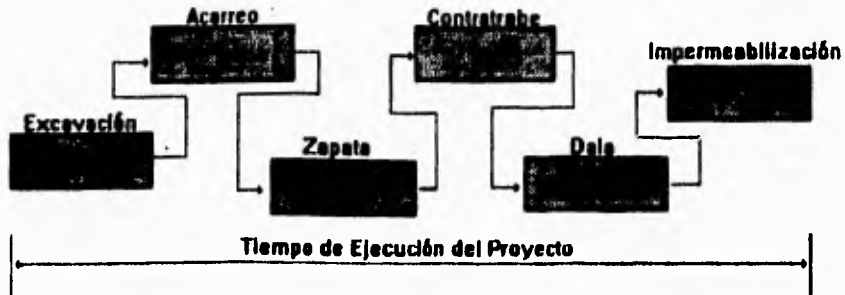


Fig. 1.1.4.3. Secuencia de Ejecución

Dependiendo de los traslapes que puedan manejarse entre estas actividades, será como podamos obtener los tiempos totales de ejecución del proyecto. Los cuales se calculan como la suma de las duraciones de las actividades menos la suma de los traslapes existentes, a manera de ejemplo , supongamos que entre la actividad 'Acarreo' y 'Zapata' existe un traslape de 1 día, así como también existe otro entre contratrabe y dala de 1 día también. Para calcular las 3 posibles duraciones de este proyecto tenemos lo siguiente:

Pesimista:

$$10 + 8 + 7 + 9 + 6 + 5 - \text{traslape 1} - \text{traslape 2} = 43 \text{ días}$$

Normal

$$8 + 6 + 5 + 7 + 4 + 3 - \text{traslape 1} - \text{traslape 2} = 31 \text{ días}$$

Optimista

$$7 + 5 + 4 + 6 + 3 + 2 - \text{traslape 1} - \text{traslape 2} = 25 \text{ días}$$

Si al ejemplo le agregamos ahora los costos por cada actividad, para la tabla de tiempos normales, podremos conocer cual es el comportamiento financiero del proyecto, así como las actividades que deben cuidarse para no elevar los costos.

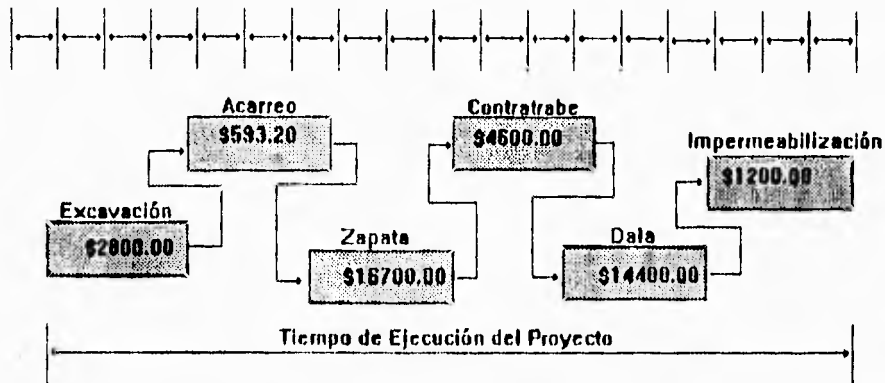


Fig. 1.1.4.4 Ejecución del proyecto en tiempo

Al aplicar la técnica *RAMPS* debemos considerar que se planea la distribución de recursos en funciones a diferentes proyectos y areas, es decir los recursos (mano de obra y materiales) del ejemplo dependen de su optimización en relación con la programación de otros sub-proyectos.

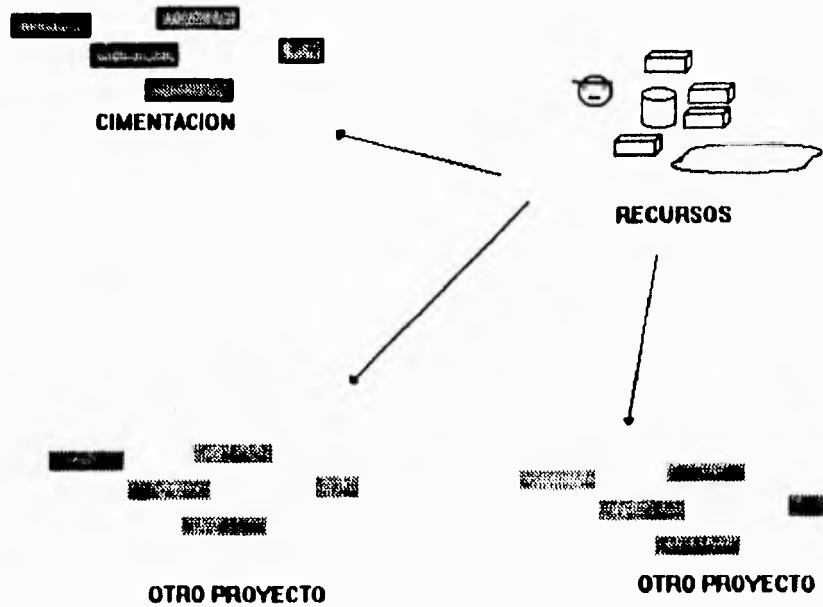


Fig. 1.1.4.5 Recursos para diferentes proyectos

Es muy importante señalar que se han definido las características de planeación de proyectos, su ubicación en el proceso administrativo, sus objetivos en un proyecto de construcción y sus características básicas. Esto nos sirve de antecedentes para determinar cuales son las necesidades específicas requeridas, es decir, cuál es la información a obtener de la realización de esta etapa de planeación los cuales se tratarán más adelante y se propondrá un modelo específico de automatización para dicho proceso.

1.2 MÓDULO DE PRECIOS UNITARIOS

1.2.1 INSUMOS

El concepto de insumos representa todos aquellos elementos que participan en un presupuesto y que regularmente no son elaborados por el constructor, son todos los bienes, servicios, mano de obra y herramientas que se obtienen de proveedores externos a una compañía constructora; cada unidad de ellos considerado en el presupuesto representa un costo para el constructor, es decir, deberá pasarse los precios de cada insumo directamente al cliente de la construcción para cubrir los gastos únicamente por adquirir estos insumos.

Los insumos son parte fundamental de un presupuesto, la variación en los costos de cada uno de ellos en conjunto con la cantidad requerida para una obra, alteran de manera directa el costo de los conceptos del presupuesto, lo cual trae como consecuencia variaciones en el importe total del mismo.

Sin contar con los insumos incluidos en el presupuesto, no es posible ejecutar ninguna de las actividades que puedan señalarse, de ahí la importancia de agruparlos y mantenerlos actualizados.

Se considera una primera agrupación de estos insumos, tomando en cuenta el tipo de insumo y la forma en que se determina su costo o unidad, por lo que se encuentran los siguientes:

Materiales

Mano de Obra

Maquinaria y Equipo

Subcontratos

A continuación se describen las principales características de cada uno de estos tipos de insumos y la forma en que debe calcularse los costos unitarios, es importante señalar que estos costos se determinan en base a los importes y factores proporcionados por los proveedores.

Es importante contar con un catálogo de proveedores para cada insumo que permita evaluar cuál de ellos ofrece el mejor servicio al mejor precio, para así evitar un crecimiento en los importes totales del presupuesto. Debe tomarse en cuenta que los costos de materiales dependen de la zona económica en que se encuentre el proveedor.

MATERIALES

Los materiales son aquellos bienes tangibles que son suministrados por los proveedores y que se utilizarán para la elaboración física de cada concepto considerado en el presupuesto, existe un número ilimitado de materiales en el mercado con características muy específicas para el tipo de obra en que se desean utilizar.

Los materiales son solicitados por el responsable de la obra, y el proveedor deberá entregarlos de acuerdo a las características de compra-venta, estas características hacen variar el costo de cada material, por ello a continuación se muestra la forma genérica en que debe calcularse dicho costo unitario, y cuales son los factores que pueden hacer variar dicho costo.

Existe un precio base determinado por el proveedor del material, el cual es el precio de lista, algunas veces se ofrecen descuentos adicionales por mayoreo y/o promociones, el factor principal para la entrega de materiales es el costo del flete, porque cada proveedor puede hacer un cargo adicional por entregar los materiales en el domicilio de la obra de acuerdo a las políticas que deban seguirse.

Por lo tanto el calculo de costo unitario de materiales queda de manera sumamente sencilla:

$$\begin{array}{rccccccc} \text{COSTO} & & \text{COSTO} & & & & & & \\ \text{UNITARIO} & = & \text{BASE} & - & \% \text{DESCUENTO} & + & \% \text{FLETE} & & \end{array}$$

Normalmente los descuentos en promociones son expresados como un porcentaje del costo base, por ello es que en la formula se considera de esta forma; de la misma forma los fletes son también representados como un porcentaje adicional.

Anteriormente se menciona que existen una gran variedad de materiales, cada material pertenece a una clasificación hecha en base a sus propiedades o a la forma en que fue manufacturado.

Con el objetivo de dejar claro el concepto de materiales, se mencionan a continuación algunas de las clasificaciones más comunes utilizadas en un presupuesto de obra.

Aglutinantes

Agregados

Aceros

Combustibles y lubricantes

Madera

Muros

Drenajes

Pisos

Recubrimientos

Diversos

A continuación se muestra una tabla con algunos materiales que nos sirven a manera de ejemplo:

MATERIAL	UNIDAD	COSTO UNITARIO	DESC	PRECIO UNITARIO
ARENA AZUL DE MINA	M3	47.73	0	47.73
GRAVA DE MINA DE 3/4"	M3	47.73	0	47.73
GRANO DE MARMOL	TON	190	0	190
PEDACERIA DE TABIQUE	M3	175	0	175
TEZONTLE	M3	45.32	0	45.32
CEMENTO CREST	KG	0.68	0	0.68
ACETATO DE POLIVINILO.	LT	5.5	0	5.5
MASTIQUE.	KG	3.5	10	3.15
RESISTOL 850, CUBETA 10 LTS	PZA	294.3	20	235.44
DIESEL	LT	0.73	0	0.73
KILOWATT / HORA	KWH	0.6	0	0.6

Capítulo I Conceptos Básicos en la Industria de la Construcción

AGUA	M3	10	0	10
ACERO REFUERZO Fy=4200, No. 2.5 (5/16")	TON	1,347.27	0	1,347.27
ALAMBRE RECOCIDO # 18.	KG	2.05	0	2.05
MALLA ELECTROSOLDADA 6-6/10x10.	M2	2.9	15	2.48
ALAMBRE GALVANIZADO CAL 16	KG	35.6	10	32.04
MADERA PINO 2a. DE 1"x4"x8.25'	PT	4.2	5	3.99
CARRETE DE HILO PLASTICO	PZA	9.85	0	9.85
CLAVO DE 1.5" CON CABEZA.	KG	2.08	0	2.08
CLAVO DE 4"	KG	2.08	0	2.08
PIEDRA DE DESBASTE No.24	PZA	36	0	38
CONCRETO PREMEZCLADO F'c=200 RN AM 3/4"A	M3	264	10	237.8
TABIQUE ROJO RECOCIDO 5.5 x 12.5 x 25 CM	MIL	385	0	385
LADRILLO COMUN BARRO RECOCIDO 12x23.5x1.	MIL	218.42	0	218.42
CELOSIA CONCRETO D-1500 VIBRO 10x30x30CM	MIL	3,504.00	10	3,153.80
BALDOSIN ROJO LAMOSA DE 20x20CM	M2	21.89	0	21.89
AZULEJO P-11 ARLEQUIN GRIS 15 X 20 cms.	M2	30.4	0	30.4
AGLOMERADO NATURAL 1.22 x 2.44 x 19 mm.	PZA	64.93	0	64.93
CRISTAL FLOTADO CLARO 4MM GRUPO 1	M2	43.37	15	36.87
VIDRIO ESPECIAL DE 3.5mm DE ESPESOR	M2	42.66	15	36.26
TORNILLO MADERA #10 4.8x45mm (CAJA 144).	PZA	12.64	0	12.64
TAQUETE DE FIBRA #14 x 38.1mm (CAJA 100)	PZA	6.47	0	6.47
LIJA MEDIANA PARA MADERA # 0.	PZA	1.67	0	1.67
PINTURA VINIL-ACRILICA VINIMEX 19 LITROS	PZA	206.9	0	206.9
SELLADOR P/MADERA 48 SOLIDOS COMEX 19 LT	PZA	188.5	0	188.5
AGUARRAS LATA DE 19 LT.	PZA	34.54	0	34.54
THINNER LATA DE 19 LT.	PZA	42	0	42
BROCHA DE PELO DE 6"	PZA	12	0	12
CERRADURA YALE A-40-S.	PZA	84.98	20	67.99
GLEIRO P/CLOSET CON 20 GANCHOS X 1.00 M.	PZA	52	0	52
GUIAS PISO PUERTA CORREDIZA CLOSET 1 M.	JGO	2.19	0	2.19
RIEL DOBLE PUERTA CORREDIZA CLOSET 1 M.	PZA	16.4	20	13.12
CARRETILLA PARA PUERTA CORREDIZA CLOSET	PZA	6.73	0	6.73
TUBO DE COBRE TIPO "M" DE 13 mm.	ML	4.75	0	4.75
CODO DE COBRE 13 MM X 90o ROSCA INTERIOR	PZA	2.5	0	2.5
TUBO PVC SANITARIO EXTRA LISO 10CM x6 M	PZA	39.28	0	39.28
COPELE DE DILATAION DE 50 mm.	PZA	3.2	0	3.2
VALVULA DE TANQUE BAJO STANDARD	PZA	19.8	0	19.8
ASIENTO I. S. REDONDO MOD. 138 C/TAPA.	PZA	38.24	10	34.42
LAVABO I. S. PROGRESO BLANCO.	PZA	283.66	0	283.66
LAVADERO DE CEMENTO CON PILETA.	PZA	35	0	35
REGADERA I. S. TLALOC 555 COMPLETA.	PZA	102.74	10	92.47
LLAVE EMPOTRAR URREA (ENSAMBLE BASICO).	JGO	18.1	0	18.1
LLAVES EMPOTRAR I.S. (ENSAMBLE BASICO).	JGO	102.52	0	102.52
TUBO CONDUIT P. DELGADA GAL. 13mm X 3.05M	PZA	8.06	0	8.06

MANO DE OBRA

Se consideran como mano de obra a las diferentes categorías de trabajadores que intervienen en un presupuesto de obra, es decir, se debe contar con los medios en que se procesarán los materiales para la construcción de cada concepto de la obra. Algunos ejemplos de mano de obra son: Cabo, Peón, Machetero, Chofer, Albañil, Plomero, etc.

La unidad base para el cálculo de costos de mano de obra es jornal, y dicho costo requiere de un análisis más detallado para su cálculo. De acuerdo a la Ley de Obra Pública existe un conjunto de parámetros que afectan estos costos. Los factores que intervienen en el cálculo de costo de mano de obra son los siguientes:

Salario Base , es la importe que el trabajador debe recibir por jornada trabajada y él es quien lo determina de acuerdo a su categoría y/o actividades.

Factor de Salario Real: Este factor representa los incrementos y/o decrementos que debe sufrir el salario base de una categoría de mano de obra y para su cálculo es indispensable cumplir con los requisitos solicitados por la ley, este factor depende de las prestaciones que el patrón proporcione a sus trabajadores, entre las más importantes se encuentran las siguientes:

Días trabajados por año

Días pagados

Prima vacacional

Aguinaldo

Seguro Social

Guarderías

Prima por antigüedad

Vacaciones

Además deben agregarse conceptos adicionales de acuerdo a las políticas del patrón, un punto importante es que las cuotas del Seguro Social dependen del salario base, es decir existen 2 niveles , para salarios mínimos, para salarios mayores al mínimo; lo cual arroja dos factores de salario real diferentes y de acuerdo al salario base debe decidirse cual es el correcto.

Por lo tanto nos queda la fórmula de la siguiente forma:

$$\text{COSTO DE MANO DE OBRA POR JORNADA} = \frac{\text{SALARIO BASE}}{\text{F.S.R.}}$$

Una forma alternativa de manejar la mano de obra, es el concepto de destajo únicamente se considera una categoría de mano de obra para cada actividad y no requiere de un análisis a detalle del factor de salario real.

A continuación se muestra una tabla con ejemplos de categorías utilizadas para mano de obra en un presupuesto,

CATEGORIA	UNIDAD	SALARIO BASE	FACTOR DE SALARIO REAL	IMPORTE DE MANO DE OBRA
AYUDANTE "A"	JOR	28.57	1.880618	48.02
AYUDANTE "B"	JOR	24.18	1.880618	40.83
AYUDANTE GENERAL.	JOR	26.37	1.880618	44.32
AYUDANTE CARPINTERO OBRA NEGRA	JOR	26.37	1.880618	44.32
AYUDANTE FIERRERO.	JOR	26.37	1.880618	44.32
AYUDANTE PLOMERO	JOR	26.37	1.880618	44.32
AYUDANTE ELECTRICISTA	JOR	26.37	1.880618	44.32
CADENERO O ESTADALERO.	EJO	24.17	1.880618	40.83
AYUDANTE ESPECIALIZADO	JOR	28.57	1.880618	48.02
AYUDANTE YESERO	JOR	26.37	1.880618	44.32
AYUDANTE PINTOR	JOR	26.37	1.880618	44.32
AYUDANTE ALUMINERO	JOR	28.57	1.880618	48.02
AYUDANTE VIDRIERO	JOR	26.37	1.880618	44.32
AYUDANTE AZULEJERO	JOR	26.37	1.880618	44.32
AYUDANTE LOSETERO	JOR	26.37	1.880618	44.32
AYUDANTE TUBERO	JOR	26.37	1.880618	44.32
AYUDANTE IMPERMEABILIZADOR	JOR	26.37	1.880618	44.32
AYUDANTE CARPINTERO OBRA BLANCA.	JOR	26.37	1.880618	44.32
AYUDANTE OPERADOR DE EQUIPO MENOR.	JOR	24.18	1.880618	40.83
AYUDANTE OPERADOR DE EQUIPO MEDIANO.	JOR	26.37	1.880618	44.32
MACHETERO DE CAMION VOLTEO O PIPA	JOR	24.18	1.880618	40.83
OFICIAL ALBANIL	JOR	47.14	1.880618	79.23
OFICIAL CARPINTERO DE OBRA NEGRA.	JOR	52.8	1.880618	88.74
OFICIAL FIERRERO.	JOR	52.8	1.880618	88.74
OFICIAL PLOMERO	JOR	52.8	1.880618	88.74
OFICIAL ELECTRICISTA	JOR	52.8	1.880618	88.74
TOPOGRAFO	JOR	75.43	1.880618	128.77
OFICIAL ESPECIALIZADO	JOR	51.66	1.880618	86.82
OFICIAL YESERO	JOR	50.91	1.880618	85.57
OFICIAL PINTOR.	JOR	49.03	1.880618	82.4
OFICIAL ALUMINERO	JOR	54.89	1.880618	91.91
OFICIAL VIDRIERO.	JOR	50.91	1.880618	85.57
OFICIAL AZULEJERO.	JOR	51.86	1.880618	87.15
OFICIAL LOSETERO.	JOR	51.86	1.880618	87.15
OFICIAL TUBERO.	JOR	49.03	1.880618	82.4
OFICIAL IMPERMEABILIZADOR.	JOR	52.8	1.880618	88.74
OFICIAL CARPINTERO OBRA BLANCA.	JOR	52.8	1.880618	88.74
CHOFER DE CAMION VOLTEO Y TORTON	JOR	51.43	1.880618	86.43
CABO PARA AYUDANTES "A".	JOR	56.57	1.880618	95.08
CABO PARA AYUDANTES "B".	JOR	24.18	1.880618	40.83
CABO PARA ALBANILES.	JOR	56.57	1.880618	95.08
CABO PARA CARPINTEROS OBRA NEGRA.	JOR	63.36	1.880618	106.48

Capítulo I Conceptos Básicos en la Industria de la Construcción

CABO PARA FIERREROS.	JOR	63.36	1.680616	106.48
CABO PARA PLOMEROS.	JOR	63.38	1.680616	106.48
CABO PARA ELECTRICISTAS.	JOR	63.36	1.680616	106.48
CABO PARA OFICIALES ESPECIALIZADOS.	JOR	61.99	1.680616	104.18
CABO PARA YESEROS.	JOR	61.1	1.680616	102.68
CABO PARA PINTORES.	JOR	58.84	1.680616	98.88
CABO PARA ALUMINEROS.	JOR	65.62	1.680616	110.29
CABO PARA VIDRIEROS.	JOR	61.1	1.680616	102.68
CABO PARA AZULEJEROS.	JOR	62.25	1.680616	104.62
CABO PARA LOSETEROS.	JOR	62.23	1.680616	104.58
CABO PARA TUBEROS.	JOR	58.83	1.680616	98.88
CABO PARA IMPERMEABILIZADORES.	JOR	63.36	1.680616	106.48
CABO PARA CARPINTEROS OBRA BLANCA.	JOR	63.36	1.680616	106.48

MAQUINARIA

La maquinaria es un Insumo, ya que sólo no es capaz de representar ningún concepto en el presupuesto, y además requiere de otros insumos para su funcionamiento y para el cálculo de sus costos por hora activa.

Existen dos formas de calcular el costo por hora de una maquina determinada, una de ellas es seguir el análisis especificado por la ley de obra pública o puede calcularse como un análisis.

A continuación se presenta el cálculo de acuerdo al análisis tradicional, la otra forma será explicada en la sección de análisis.

El costo por hora se componen de los siguientes elementos:

Costo base de la maquina: Es el importe necesario para la adquisición inicial del equipo.

Costo base de la máquina: Es el importe necesario para la adquisición inicial del equipo.

Cargos Fijos : Se determinan de acuerdo al tipo de maquinaria (factores de almacenamiento, primas de seguro anual, depreciación, inversión, mantenimiento)

Consumos: Son aquellos importes de combustibles y lubricantes necesarios para el funcionamiento de la máquina.

Cargos de Operación: Son los importes de los operadores que trabajarán con la máquina.

Un análisis de costo horario es muy específico y debe realizarse con mucho cuidado, sobre todo en el cálculo de las cantidades para consumos y operación ya que un error en alguna de ellas, puede generar grandes desviaciones en el importe del presupuesto.

A continuación se muestra una tabla con algunos ejemplos típicos de maquinaria y equipo.

EQUIPO	UNIDAD	PRECIO UNITARIO
MALACATE CON PLUMA DE 1 TON.MIPSA M-1000	HR	15.20
CORTADORA BLACK & DECKER MOD. 4076 1 HP	HR	0.62
NIVEL BASCULANTE ROSSBACH	HR	0.58
TRANSITO ROSSBACH	HR	1
CAMION DE VOLTEO FAMSA 1317/60 7 M3.	HR	65.41
REVOLVEDORA 1 SACO MIPSA R-10 P/CONCRETO	HR	11.59
VIBRADOR P/CONCRETO MECSA 4 HP KOHLER	HR	8.52
PULIDORA EQUIMAP PARA PISOS PETREOS.	HR	3.41

SUBCONTRATOS

Los Subcontratos son todos aquellos servicios que un proveedor puede ofrecer sin necesidad de hacer un desglose de los elementos que lo componen, y que a su vez son elementos activos en un presupuesto de obra. Normalmente estos Subcontratos son utilizados para la realización de actividades como : cancelería, colocación de alfombras, acabados en madera, jardinería, etc.

El costo ofrecido por un proveedor debe considerarse directamente en el presupuesto, es decir se debe indicar el costo por unidad del servicio o por el servicio completo, de acuerdo a como se defina en presupuesto, pero lo importante es que no se deben alterar por ningún factor adicional.

En algunas ocasiones, se llega a rentar el equipo necesario, el cual no es necesario actualizar y podrá considerarse como un subcontrato y únicamente debemos manejar el costo por hora.

SERVICIO	UNIDAD
COLOCACIÓN DE ALFOMBRAS	LOTE
COLOCACIÓN DE AZULEJOS	LOTE
COLOCACIÓN DE CRISTALES	LOTE
HERRERÍA	SERVICIO
CARPINTERÍA	SERVICIO
PLOMERÍA	SERVICIO
ELECTRICIDAD	SERVICIO

1.2.2 ANÁLISIS

Un análisis lo podemos definir como el conjunto de elementos necesarios para la elaboración de una unidad de concepto de obra, conociendo así el costo correspondiente.

Generalmente un análisis es conocido de diferentes formas, entre las que destacan *Integrado, Tarjeta, Matriz, Precio* y son los elementos base de un presupuesto. De acuerdo a su utilización podemos hacer una subdivisión de estos análisis de la siguiente forma:

Cuadrillas: Es un conjunto de personas que desarrollan una actividad común, es decir las categorías de mano de obra necesarias para llevar a cabo una unidad de concepto de obra. Las cuadrillas únicamente están formadas por mano de obra y herramienta y sirven de referencia para formar parte de un integrado. Por ejemplo: cuadrilla de acarreo, cuadrilla de despalme, cuadrilla de camión de volteo, etc.

Integrados: Contenido fundamental del presupuesto, en donde se encuentra la integración de diferentes insumos e incluso cuadrillas para formar una unidad de un concepto de obra. Se manejan conceptos de obra factibles de cuantificar en la elaboración de un presupuesto. Por ejemplo concreto, muro, cimbras, lozas, etc.

Conceptos de Presupuesto: Son conocidos como partidas del presupuesto y aquí se agrupan las cantidades de cada integrado necesarias para desempeñar conceptos presupuestados. Aquí los importes deben reflejar los volúmenes de obra presupuestados, dejan de ser unitarios. Ejemplos de partidas son : Instalación eléctrica, instalación hidráulica, preliminares (preparación de terreno), decoración, equipamiento, acabados, carpintería, etc.

Regularmente es necesario incluir integrados dentro de otros integrados para tener un buen análisis de precios unitarios, es por ello que una forma general de la composición de cada análisis podría ser de la siguiente forma:

Cuadrillas: Pueden estar formadas de mano de obra y/o otras cuadrillas.

Integrados: Pueden estar formados de insumos, cuadrillas y otros integrados.

Partidas: Pueden estar formados de insumos, cuadrillas, integrados y otras partidas.

Estas características pueden variar de proyecto a proyecto, pero en la mayoría de los casos se presenta una organización en la información de este tipo. Es decir, estas son las características de un caso típico sin que ello limite a un constructor a organizar sus presupuestos de manera diferente.

Para el caso de la maquinaria, esta se puede ver como un análisis (regularmente se hablaría de un nivel básico similar al de la cuadrilla), ya que contiene un conjunto de elementos que sumados representan el costo por hora, únicamente es necesario determinar las cantidades adecuadas de cada elemento que forme parte del costo-hora.

Para el cálculo del costo directo de un análisis se requiere del conocimiento de un Ingeniero Civil o Arquitecto que determine las cantidades de insumos requeridas para elaborar una unidad de concepto, es decir, que la elaboración de un concepto de obra tiene un fondo ingenieril, ya que el incluir una cantidad determinada de x material le este proporcionando diferentes características al precio unitario (presión, peso máximo, deformación, resistencia, humedad, etc) y deben apegarse a las normas de calidad de la industria de la construcción.

Durante la elaboración de un integrado se conocen las cantidades teóricas necesarias para la elaboración de los conceptos, pero en el proyecto real, siempre se consume más de lo que teóricamente se presupuestó, por ello es necesario manejar un factor que de alguna manera nos ayude a prever estas desviaciones, este factor es general y lo podemos definir de la siguiente forma:

$$\begin{array}{rcccl} \text{CANTIDAD} & & \text{CANTIDAD} & & \\ \text{REAL} & = & \text{TEÓRICA} & \cdot & \text{FACTOR REAL} \end{array}$$

Este factor real representa el desperdicio en el caso de almacén, y el rendimiento para los casos de mano de obra y equipo. En un subcontrato el rendimiento siempre debe ser el 100%.

Para las partidas el factor debe ser 1, porque se están presupuestando las cantidades de integrados, en este nivel las cantidades se conocen como volúmenes de obra a ejecutar.

En un análisis se maneja conceptos de herramienta y mando intermedio, estos funcionan igual que los insumos pero no tienen un costo unitario definido, porque dependen del importe de mano de obra existente en cada análisis y es por ello que no existe una categoría como tal que forme parte de los insumos.

El cálculo del costo directo de un análisis se calcula como la suma de todos los importes de sus componentes, y a su vez el importe de cada componente se calcula como la cantidad dividida por el rendimiento y multiplicada por el costo unitario.

A continuación se presentan 3 tablas, con algunos ejemplos representativos de cada nivel de análisis de los que se han mencionado anteriormente.

BASICOS

DESCRIPCION	UNIDAD	PRECIO UNITARIO
CUADRILLA 001 (1 AYUDANTE "A").	JOR	60.16
CUADRILLA 002 (1 AYUDANTE "B").	JOR	48.64
CUADRILLA 010 (1 OF. ALB. + 1 AYD.)	JOR	151.69
CUADRILLA 020 (1 CARP. O.N. + 1 AYD.)	JOR	163.83
CUADRILLA 030 (1 FIERRERO + 1 AYD.)	JOR	163.83
CUADRILLA 040 (1 PLOMERO + 1 AYD.)	JOR	163.83
CUADRILLA 050 (1 ELÉCT. + 1 AYD.)	JOR	163.83
CUADRILLA 080 (1 TOPOGRAFO + 2 CAD.)	JOR	216.35
CUADRILLA 090 (1 OF. ESPECIAL+1 AYD.)	JOR	165.6
CUADRILLA 200 (1 TUBERO + 1 AYD.)	JOR	155.73

CUADRILLA 210 (1 IMPERMEAB. + 1 AYD.).	JOR	163.83
CUADRILLA 230 (1 CARP. O.B. + 1 AYD.).	JOR	163.83
MORTERO CEMENTO-ARENA 1:3.	M3	225.61
MORTERO CEMENTO-ARENA 1:4.	M3	199.78
MORTERO CEMENTO-ARENA 1:5.	M3	177.77
MORTERO CEMENTO-CAL-ARENA 1:1:8.	M3	153.1
LECHADA CEMENTO BLANCO-AGUA.	M3	872.38
LECHADA CEMENTO GRIS-AGUA.	M3	496.93
PASTA CEMENTO GRIS-AGUA.	M3	713.60
PASTA YESO-CEMENTO-AGUA.	M3	203.53
PASTA CAL-CEMENTO B.-GRANO MARMOL 1:1:8.	M3	575.78
CONCRETO F'C=100 3/4" H. O. REVOLVEDORA.	M3	183.91
CIMBRA EN DALAS DE LIGA < 25 CM.	M2	26.0
CIMBRA EN CASTILLOS < 25 CM.	M2	25.0
TARIMA DE DUELA DE 0.50x.100 M.	PZA	20.04
RAMALEO HIDR. COBRE REGADERA G. ARIETE.	PZA	89.48
TUBO DE COBRE DE 13 mm. TRAMO DE 3 ML.	PZA	27.88
TUBO DE COBRE DE 13 mm. TRAMO DE 6 ML.	PZA	43.05
RAMALEO SANITARIO PVC INODORO.	PZA	75.98

INTEGRADOS O MATRICES

DESCRIPCION	UNIDAD	PRECIO UNITARIO
TRAZO/NIVELACION P/ESTRUCTURA SUP<300M2	M2	2.24
EXCAVACION CEPA 2.0 M PROF. MATERIAL I.	M3	16.21
EXCAVACION CEPA 2.0 M PROF. MATERIAL II.	M3	28.61
ACARREO CARRETILLA MATERIALES A 20 M.	M3	8.11
CARGA Y ACARREO MAT.PROD.EXC. TIRO LIBRE	M3	49.91
PLANTILLA CONCRETO F'C=100 KG/CM2 5CM.	M2	13.49
ZAPATA CORRIDA CIMENTACION 0.6 M ANCHO.	ML	73.21
CONTRATRABE CIMENTACION 20 X 80 CM	ML	125.42
CIMBRA FRONTERA ZAPATA CORRIDA <10CM 2C.	ML	6.78
CIMBRA CONTRATRABE 20x80 CM 2 CARAS.	ML	34.62
CIMBRA COMUN DUELA TRABE 25x40 CM.	ML	34.5
ACERO 5/16" CIMENTACION SUM., HAB. Y ARM	TON	2,955.24
CONCRETO H.O. F'C=200 TMA 3/4" CIMENT.	M3	279.25
CONCRETO PREM. FC=200 3/4" TRAB/LOSA "A"	M3	324.92
CONCRETO BOMBEADO FC=200 3/4" TRAB/LOSAS	M3	323.21
TRABE CONCRETO 25x40 CON 4 VAR. 5/8".	ML	108.12
LOSA CONCRETO 10 CM PERALTE	M2	83.79
LOSA MACIZA RAMPA ESCALERA 12CM PERALTE	M2	85.12
LOSA AUTOSUSTENTABLE VIG/BOV DE 20 CM.	M2	77.91
DALA DE ENRASE 15 X 30CM 4 VARS 3/8"	ML	64.24
CASTILLO 15 X 15CM 4 VARS 3/8"	ML	41.32
CELOSIA CONCRETO D-1500 CEMENT-ARENA 1:4	M2	49.11
ESCALON GRANITO PRECOLADO 17X30X100 CM.	ML	91.45

Capítulo I Conceptos Básicos en la Industria de la Construcción

SARDINEL FORJADO CONCRETO Fc=100 10x10CM	ML	15.84
SARDINEL AZULEJO 9 CUÁDROS 11X11 CM.	ML	47.15
PERFILAR ALEROS CON PASTA DE CEMENTO.	ML	13.55
ACCESORIOS BANO EMPOTRAR I.S. 500 BLANCO	JGO	52.51
LAVADERO DE CEMENTO C/PILETA SUM. Y COL.	PZA	89.14
BASE DE TABIQUE 50 X 50 CM P/CALENTADOR.	PZA	64.82
BASES DE TABIQUE PARA TINACO (2 BASES).	LOT	52.53
TUBERIA CONCRETO SIMPLE 15 CM SUM Y COL.	ML	17.29
REGISTRO 40x60x80 CM TABIQUE Y CONCRETO.	PZA	197.65
RESANES INSTALACION ELÉCTRICA EN MUROS.	ML	4.84
RESANES INSTALACION SANITARIA EN MUROS.	ML	5.84
COLADO DE TAPA DE REGISTRO 60 X 40 CM.	PZA	43.6
COLADO TAPA REGISTRO C/COLADERA CAMPANA.	PZA	55.05
RECIBIR Y AMACIZAR COLADERA DE BANO	PZA	25.88
MURO TABIQUE 5.5CM CEMENTO-ARENA 1:5.	M2	25.93
MURO TABIQUE 19.5CM PLASTOCÉM-ARENA 1:4.	M2	62.71
PRETIL DE CONCRETO ARMADO F'c=200 OBRA	ML	94.94
FIRME CONCRETO F'c=100 OBRA 1.5" DE 8CM	M2	23.67
PISO FINO ESCOBILLADO 3 CM CEM-ARE 1:3.	M2	12.69
PISO AZULEJO 9 CUADROS 11 x 11 cm.	M2	71.01
PISO BALDOSIN ROJO LAMOSA 20x20CM	M2	28.39
LAMBRIN DE AZULEJO ARLEQUIN 15x20 CM.	M2	59.26
APLANADO YESO 2CM REVENTON PLAFONES.	M2	13.04
APLANADO YESO 2CM PLOMO Y REGLA MUROS.	M2	13.27
TIROL RUSTICO EN PLAFONES.	M2	10.32
TIROL PLANCHADO EN MUROS.	M2	10.82
RÉPELLADO 2.5CM MOR-ARE 1:5 MUROS.	M2	19.18
APLANADO FINO 2.5CM CEM-CAL-AR 1:1:8 MUR	M2	19.67
CHAFLAN 10x10CM MORTERO CEM-CAL-AR 1:1:6	ML	4.78
ENTREPAÑO DE TRIPLAY DE PINO 6mm.(2 C)	ML	70.9
ENTREPAÑO DE AGLOMERADO 19 mm.	ML	37.84
PUERTA 90 x 210 CM TRIPLAY DE PINO.	PZA	233.79
CHAMBRANA PINO PARA PUERTA 90 x 213 CM	PZA	74.95
CAJON COMPLETO DE PINO P/PUERTA 70 x 213	PZA	52.34
CLOSET 210 x 230 CM PUERTAS CORREDIZAS	PZA	1,442.50
CERRADURA YALE A-40-S.	PZA	91.4
CERRADURA YALE A-79-PD.	PZA	118.31
CERRADURA YALE A-80-PD.	PZA	108.35
CANCELERIA ALUMINIO A.N. 2"x1 1/4" LUJO	M2	158.17
VENTANA ALUMINIO ANODIZADO NAT. 2".	M2	137.97
CRISTAL FLOTADO CLARO 4 MM GRUPO 1.	M2	57.83
CRISTAL FLOTADO CLARO 5MM GRUPO 1	M2	72.01
VIDRIO ESPECIAL DE 3.5 MM DE ESPESOR	M2	50.78
LUNA CRISTAL FLOTADO CLARO 4MM GRUPO 1	M2	103.94
SELLADO PERIMETRAL VENTANA ALUMINIO EXT.	ML	1.52
INTEGRADO DE PRUEBA	MTS	365.65
INST. HIDROSANITARIA W.C. COBRE/PVC.	SAL	144.82
LAVABO I.S. PROGRESO BLANCO C/LLAVES SYC	PZA	399.17
INODORO ORION ARGOS BLANCO SYC	PZA	360.52

REGADERA TLALOC 555 I.S. BRAZO Y CHAPET.	JGO	215.47
CALENTADOR AUTOM. 100 LT CALOREX PREMIER	PZA	672.13
TINACO ASBESTO 1100 LTS. CON BASES.	PZA	682.42
SALIDA PARA CENTRO ELECTRICO C/POLIDUCTO	SAL	65.18
SALIDA ALUMBRADO Y CONTACTO TUBO CONDUIT	SAL	106.81
CONTACTO C/POLIDUCTO SALIDA COMPLETA.	SAL	129.13
ARBOTANTE C/POLIDUCTO.SALIDA COMPLETA.	SAL	52.94
TIMBRE COMPLETO SUM. Y COL.	SAL	90.87
ACOMETIDA ENERGIA ELECTRICA TUBO P.V.C.	ML	21.29
INTERRUPTOR SEGURIDAD CUCHILLAS 2x30 AMP	PZA	62.45
ALIMENTACION A C.CARGA TUBO CONDUIT PVC	SAL	241.82
CENTRO DE CARGA QO-2 INTERRUPTOR 20 AMP.	PZA	101.03
TELEFONO C/POLINUCTO SALIDA SOLO GUIADA.	SAL	41.38
SALIDA TELEFONO Y TELEVISION T.CONDUIT.	SAL	33.32
PINTURA VINILICA MURO/PLAFON YESO	M2	8.69
BARNIZ NATURAL C/BROCHA MADERA TRATADA	M2	41.1
LIMPIEZA LAMBRIN AZULEJO.	M2	1.46
LIMPIEZA MUEBLE SANITARIO.	PZA	4.63
LIMPIEZA ACCESORIOS BANO.	PZA	2.2
LIMPIEZA VIDRIO DOS CARAS.	M2	2
LIMPIEZA DE CELOSIA DE BARRO.	M2	3.44
LIMPIEZA GENERAL DE LA OBRA	M2	3.04
LIMPIEZA GRUESA DE OBRA EN PROCESO.	M2	1.95
PULIDO Y BRILLADO DE ESCALONES GRANITO.	M2	11.04

CONCEPTOS DE PRESUPUESTO

DESCRIPCION	IMPORTE
PRELIMINARES EDIFICIO.	631.41
CIMENTACION DE EDIFICIO.	21,651.77
ALBANILERIA POR DEPARTAMENTO.	34,435.73
ESCALERAS EDIFICIO.	5,718.11
AZÓTEA EDIFICIO.	13,233.44
INSTALACION ELECTRICA POR DEPARTAMENTO.	3,434.59
INSTALACION HIDROSANITARIA POR DEPTO.	4,442.73
CANCELERIA DE ALUMINIO POR DEPARTAMENTO.	2,201.92
YESO Y PINTURA POR DEPARTAMENTO.	6,966.81
CARPINTERIA POR DEPARTAMENTO.	5,918.46
VIDRIOS Y LUNAS POR DEPARTAMENTO.	1,167.72
PULIDOS Y LIMPIEZAS POR DEPARTAMENTO.	730.55

1.2.3 PRESUPUESTO

El presupuesto de una obra es la determinación del monto total de las erogaciones o gastos necesarios para llevar a cabo la ejecución de la misma por parte del constructor, y si este es una empresa privada, deberá incluir su utilidad. Se obtiene de multiplicar el volumen de proyecto de cada concepto por su precio unitario y efectuando la suma de todos. Es decir:

$$\text{Costo} = (A_i \times P_{ui})$$

donde:

Costo	=	Monto de la obra ejecutada
A	=	Cantidad realizada por cada concepto (m ² , m ³ , etc..)
PU	=	Precio Unitario de cada concepto por unidad.
i	=	Conceptos

Este debe llenar las siguientes condiciones:

- Que cada parte de la obra corresponda a un concepto o grupo de conceptos de trabajo bien definidos.
- La descripción de estos conceptos debe permitir obtener una idea clara y precisa del trabajo al que se refiere.
- Los análisis de precios unitarios deben ser claros y sencillos.

Los presupuestos varían durante el desarrollo de la obra, debido a las siguientes causas:

- Trabajos extraordinarios
- Modificación de los volúmenes de proyecto
- Actualización de precios unitarios

La elaboración de un presupuesto es muy valiosa para poder llevar a cabo una correcta Planeación, Coordinación y Control de las operaciones de una empresa.

PLANEACIÓN

- Que se va a hacer. Establecer objetivos antes de comenzar las actividades.
- Con qué se va a hacer. Contar con los factores de producción que hagan factibles la obtención de los resultados planeados.
- Como se va a hacer. Organización de funciones y distribución de responsabilidades.

COORDINACIÓN

- Desarrollar y mantener las diversas actividades de cada departamento equilibradamente considerando sus relaciones recíprocas.
- La coordinación de esfuerzos y recursos , conduce a un objetivo común, para lo cuál es necesario la dirección o coordinación.

CONTROL

Es la acción de vigilancia para que las operaciones sigan el curso previsto, corrigiendo oportunamente las desviaciones en que se hubiese incurrido.

EXTENSIÓN DE LOS PRESUPUESTOS

La aplicación del sistema presupuestal en las empresas, se deriva de la práctica gubernamental de estimar los gastos e ingresos del Estado.

Por esto tradicionalmente, la palabra "presupuesto" implica en su aceptación primaria, la delimitación de un gasto.

Se puede decir que los objetivos que se persiguen con la presupuestación, son los siguientes:

- Consideración del futuro para que los planes trazados, permitan la obtención de la utilidad máxima, de acuerdo con las condiciones que se presenten.
- La coordinación de todas las actividades, para obtener ese fin.
- Asegurar la liquidez financiera de la empresa.
- Establecer un control para conocer si los planes son llevados a cabo y determinar la dirección que se lleva en relación a los objetivos establecidos.

EL PRESUPUESTO COMO UN INSTRUMENTO DE CONTROL

El presupuesto representa el curso a seguir para hacer posibles ciertos objetivos; pero por sí solo, no evita las desviaciones de ese rumbo o asegura la realización de las metas.

Podemos considerar que la dirección planteada por el presupuesto, se representa por una línea. Esta línea es la resultante de las fuerzas que actúan en sentido contrario; el predominio de una o de otras, trae consigo la desviación de la línea original. La función del presupuesto es trazar no solamente los resultados netos de la operación y su curso, sino que también considera el importe y la dirección de las fuerzas determinantes, de cuya actuación resulta la directriz.

Toda desviación debe indicarse por la comparación de las diversas fuerzas reales con aquellas que se presupuestaron.

La localización de las fuentes de variación señalará las posibles medidas correctivas.

La duración del período presupuestal depende esencialmente del tipo de empresa; sin embargo, su duración debe ser tal que comprenda los ciclos completos de:

- Fluctuación de volumen por temporadas.
- Producción
- Rotación de la mercancía
- Operaciones Financieras

Se aconseja que el presupuesto no cubra un período demasiado extenso ya que no pueden preverse todas las condiciones que puedan afectarlo.

CLASIFICACION DE PRESUPUESTOS

Heckert & Willson, clasifican los presupuestos en tres clases:

- Presupuestos de Apropiación
- Presupuestos Fijos
- Presupuestos Flexibles o Variables

El tipo de apropiación tiene una aplicación restringida en los negocios; se caracteriza por el establecimiento de ciertas sumas como límite máximo pueden gastarse en tal o cuál actividad . Se aplica en los negocios en relación a los gastos de publicidad o a los gastos de investigación y desarrollo.

Los presupuestos fijos representan un programa inalterable , al variar las ventas y el volumen de producción, conforme las condiciones que se presenten, las estimaciones presupuestadas "no se cambian" . Tanto en los gastos fijos como en los variables se deben limitar a las cantidades fijas que se presupuestaron.

El presupuesto flexible, permite la rectificación de las estimaciones, de los costos de operación y utilidades, como consecuencia de los cambios en la venta o en el volumen de la producción . Supera al presupuesto fijo, al permitir su ajuste con las variaciones

que se produzcan en la práctica . El programa de operaciones futuras, se basa en las condiciones previstas que se estima ocurrirán en el ejercicio presupuestal.

ANÁLISIS Y CONTROL DE COSTOS

Dentro de los propósitos más importantes del presupuesto se encuentran el análisis y control de costos.

El Análisis de costos lo entendemos de una manera general como el desglose de todos los factores que lo afectan directa o indirectamente y poder hacer una medición tanto cualitativa como cuantitativa de cada factor para así contar con las bases necesarias para la toma de decisiones adecuadas relacionadas con ellos.

El control de costos es el proceso mediante el cuál se comparan de una manera sistemática cada uno de los elementos constitutivos de los costos generados durante la construcción de una obra contra los parámetros previamente establecidos es fácil comprender que dicha comparación debe realizarse de una manera periódica , con el fin de poder implementar acciones correctivas en caso de que existan desviaciones importantes entre lo programado y lo real.

Los principales aspectos que se deben controlar en una obra en cuanto a costos son básicamente:

- Control de materiales
- Control de mano de obra

que se produzcan en la práctica . El programa de operaciones futuras, se basa en las condiciones previstas que se estima ocurrirán en el ejercicio presupuestal.

ANÁLISIS Y CONTROL DE COSTOS

Dentro de los propósitos más importantes del presupuesto se encuentran el análisis y control de costos.

El Análisis de costos lo entendemos de una manera general como el desglose de todos los factores que lo afectan directa o indirectamente y poder hacer una medición tanto cualitativa como cuantitativa de cada factor para así contar con las bases necesarias para la toma de decisiones adecuadas relacionadas con ellos.

El control de costos es el proceso mediante el cuál se comparan de una manera sistemática cada uno de los elementos constitutivos de los costos generados durante la construcción de una obra contra los parámetros previamente establecidos es fácil comprender que dicha comparación debe realizarse de una manera periódica , con el fin de poder implementar acciones correctivas en caso de que existan desviaciones importantes entre lo programado y lo real.

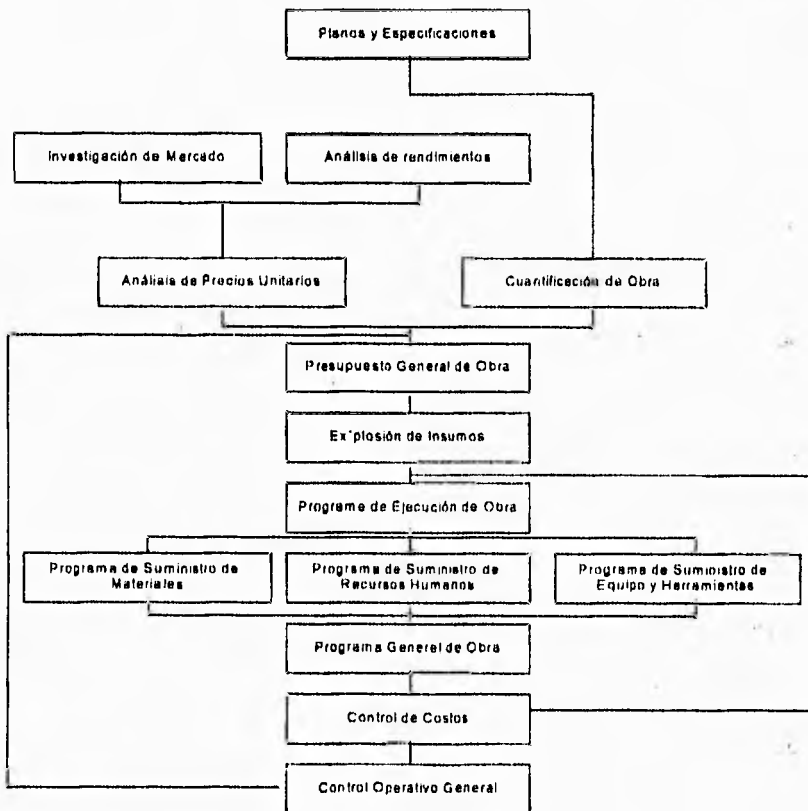
Los principales aspectos que se deben controlar en una obra en cuanto a costos son básicamente:

- Control de materiales
- Control de mano de obra

- Control de equipo y maquinaria
- Control de avances y estimaciones

Existe dentro de la industria de la construcción un proceso de análisis y control de costos el cuál se anexa a nivel flujograma.

PROCESO DE ANALISIS Y CONTROL DE COSTOS



Como se puede observar en el diagrama anterior los puntos principales son:

- Análisis de Precios Unitarios
- Presupuesto General de Obra
- Programa General de Obra
- Controles de Costos

Dentro de un presupuesto se verán reflejados los siguientes factores:

□ COSTOS DE MANO DE OBRA

Para obtener el costo real de cada trabajador se hace uso de tablas salarios-base vigente a la fecha del análisis de la obra. Esta tabla se encuentra en función de la zona económica de la obra.

El costo de mano de obra se clasifica de la sig. manera:

- 1.- Peón
- 2.- Ayudante
- 3.- Oficial de Albañilería
- 4.- Carpintero
- 5.- Etc...

Sin embargo para el análisis de costos no se usan dichos salarios individuales, sino que se consideran en base a cuadrillas.

Por ejemplo:

CUADRILLA	ELEMENTOS
1	Peón
2	Albañil-Peón
3	Carpintero-Ayudante
4	Azulejero-Ayudante

COSTO DE EQUIPO

Este varía de acuerdo al tipo de maquinaria que se va a utilizar en una obra y depende de la compañía constructora, pues es a ella, a la que va a afectar directamente en sus cargos directos de edificación.

COSTOS FIJOS

Se dividen en:

1.- Cargos por depreciación: Se denomina a la pérdida de valor que sufre la maquinaria o equipo por razón de utilización, desgaste y sistemas de trabajo a que está expuesta, así como a las condiciones de tiempo que la transforma en obsoleta.

Este cargo queda expresado de la siguiente manera:

$$D = (V_a - V_r) / V_e$$

en donde:

- Va** = Valor de adquisición, el cuál incluye, precio de compra, impuestos, impuestos causados por la compra, flete hasta su primer destino , costo de descarga, montaje, armado, para tener a la máquina en condiciones de trabajo.
- Vr** = Valor de rescate
- Ve** = Vida Económica de la máquina expresada en horas de trabajo.
- D** = Depreciación por hora efectiva de trabajo.

2.- Cargo por Inversión: Es el equivalente a los intereses correspondientes que causa el capital invertido en la compra de la maquinaria y se expresa:

$$I = ((Va - Vr) / (2 Ha)) \times i$$

en donde:

- I** = Cargo por inversión
- Va** = Valor de adquisición
- Vr** = Valor de rescate
- Ha** = Número de horas efectivas de trabajo durante el año.
- i** = Tasa de interés anual en vigor.

3.- Cargos por seguros: Es la erogación que se realiza para proteger y cubrir la maquinaria de los riesgos a que está expuesta durante su vida económica y se expresa:

$$S = ((Va + Vr) / (2 Ha)) \times s$$

en donde:

- S** = Cargo por seguro en hora efectiva de trabajo
- Va** = Valor de adquisición
- Vr** = Valor de rescate
- Ha** = Número de horas efectivas de trabajo
- s** = Prima anual promedio, expresada en % anual del valor de la maquinaria.

4.- Cargos por mantenimiento: Representa las erogaciones necesarias para conservar la maquinaria en buen estado y condiciones de trabajo óptimas durante su vida económica .

CARGOS POR CONSUMO

Es el derivado de todas las erogaciones originadas por todos los consumos de gasolina o diesel para que los motores produzcan energía que utilizan al desarrollar el trabajo.

$$E = e Pc$$

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

en donde:

- E** = Cargo por consumo de combustible
- e** = Cantidad de combustible necesario por hora efectiva de trabajo.
- Pc** = Precio del combustible que consume la máquina.

□ CARGOS POR OPERACIÓN

Se deriva de las erogaciones que hace el contratista por concepto de pago de salarios al personal, encargado de la operación de la máquina por hora efectiva de trabajo.

$$O = St / H$$

en donde:

- O** = Cargo de operación del equipo por hora efectiva de trabajo.
- St** = Representa los salarios por turno del personal necesario para operar la máquina.
- H** = Representa las horas efectivas de trabajo que se consideran para la máquina dentro del turno.

De lo anterior resumimos que el costo requerido para la elaboración de un presupuesto se puede determinar de la siguiente manera:

$$\text{COSTO DIRECTO} = (\text{C.M.})a + (\text{C.EQ.})b + (\text{C.F.})c + (\text{C.OP})d + (\text{C.C})e + \dots$$

donde:

a, b, c, d,... representan las cantidades consumidas por cada uno de los conceptos.

A continuación se muestra la relación que existe entre los datos del presupuesto. Ver

fig.1.2.3.1

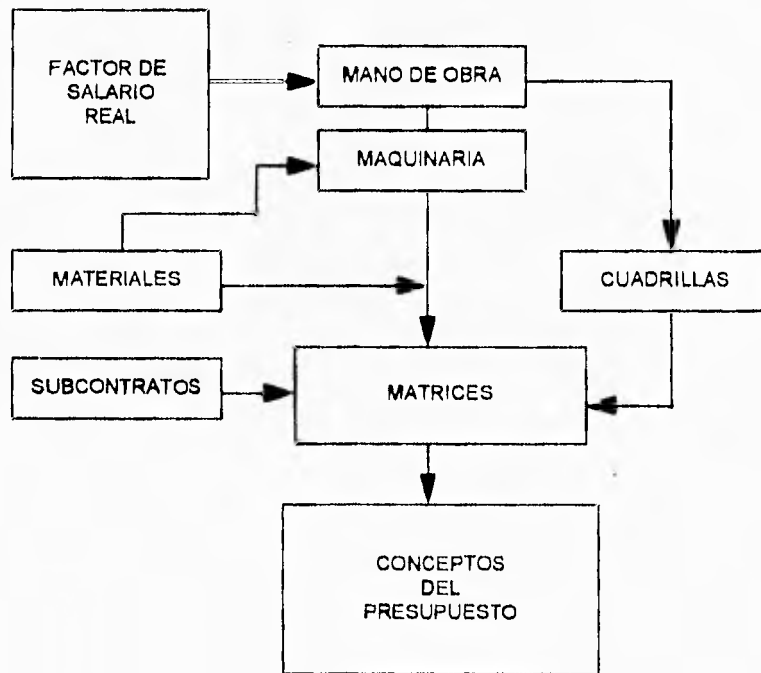


FIG.1.2.3.1 RELACION ENTRE LOS DATOS DEL PRESUPUESTO

Un ejemplo de un presupuesto podría ser:

Presupuesto de una Carretera

	IMPORTE
	827'889,251
	797'008,598
	1'398,133,074
	17'047,188
	233'051,354
	414'792,914
	87'661,125
	149'053,547
	3'924,637,053

Capítulo I Conceptos Básicos en la Industria de la Construcción

DESCRIPCION DEL CONCEPTO	UNIDAD	CANTIDAD	PRECIO UNITARIO	IMPORTE
TERRACERIAS				
Desmante por unidad de obre terminada	Ha	1.0000	698,826.21	698,826.21
Despalmes desperdiciando el material de cortes p.u.d.t.	M3	390.0000	4,051.94	1'580,256.60
Despalmes desperdiciando el material para desplante de terraplenes	M3	1,150 0000	4,051.94	4'659,731 00
Excavaciones en cortes y adicionales abajo de la subrasante cuando el material se desperdicie.	M3	690.0000	9,375.75	6'469,267.50
Excavaciones en rebajes de corono y/o terraplenes existentes cuando el material se desperdicie.	M3	690.0000	9,375.75	6'469,267.50
Excavaciones abriendo caja para desplante de terraplenes cuando el material se desperdicie.	M3	3,380.0000	9,375.75	31'690,035.00

Capítulo I Conceptos Básicos en la Industria de la Construcción

Excavaciones en escalones de liga en los taludes de los terraplenes existentes cuando el material se utilice para la formación de los terraplenes.	M3	150.0000	8,686.25	1'309,937.50
Excavaciones de prestamos del banco ubicado a 2,600 mts. a la izquierda de la estación 101+250	M3	48,740.0000	6,589.38	272'426,381.20
Compactación P.U.D.T. del terreno natural en el area de desplante de los terraplenes para 90%.	M3	2,250.0000	3,790.07	8'527,657.50
Compactación de las terracerías existentes construidas con anterioridad / para 95 % p.u.o.t.	M3	835.0000	3,816.33	3'186,635.55
Formación y compactación de terraplenes adicionados con sus cunas de	M3	25,470.0000	6,069.15	154'581,250.50

Capítulo I Conceptos Básicos en la Industria de la Construcción

sobreancho para 90%.				
Formación y compactación de terraplenes adicionados con sus cunas, de sobreancho para 95 %.	M3	9,890.0000	5,089.15	60'023,893.50
Formación y Compactación de terraplenes adicionados con sus cunas de sobreancho para 100%.	M3	5,740.0000	6,069.15	34'836,921.00
Formación de Compactación de ampliacion de la corona adicionada con sus cunas de sobreancho en terraplenes existentes para 100%.	M3	640.0000	6,317.68	4'043,315.20
Excavaciones para canales de entrada y salida a obras de drenaje.	M3	1,040.0000	35,125.13	36'530,135.20
Sobreacarreo para cualquier distancia de materiales de prestamo de	M3	48,740.0000	884.22	43'096,882.80

Capítulo I Conceptos Básicos en la Industria de la Construcción

banco para la construcción de la capa de subrasante y para completar la construcción de terraplen para el primer kilometro.				
Sobreacarreo para cualquier distancia de materiales de préstamo de banco para la construcción de la capa de subrasante y para completar la construcción de terraplen para kilometros subsecuentes.	M3	341,180.0000	389.64	132'937,375.20
			Subtotal	827'889,251.36

Capítulo I Conceptos Básicos en la Industria de la Construcción

OBRAS DE DRENAJE				
Excavación para estructuras cualesquiera que sea su clasificación y profundidad.	M3	480.0000	28,545.13	13'702,142.40
Rellenos para la protección de obras de drenaje, p.u.o.t.	M3	480.0000	6,317.47	3'032,385.60
Mampostería de tercera clase, para cualquier altura con mortero de cemento p.u.o.t.	M3	220.0000	114,598.69	25'211,711.80
Concreto hidráulico colado en seco p.u.o.t.	M3	20.0000	248,862.67	4'977,253.40
Acero de refuerzo, varilla de límite elástico igual o mayor de 2320 kg	Kg	1,720.0000	2,144.41	3'638,385.20
Tubería de concreto reforzado de F'C = 280 kg/cm ² de () cms de diámetro.	M1	120.0000	237,237.12	28'468,454.40
Guarniciones de concreto	M1	1,900.0000	23,940.98	45'487,862.00

Capítulo I Conceptos Básicos en la Industria de la Construcción

hidraulico F'C=200 kg/cm2 de 1,110 cm2 de sección para camellón central.				
Guarniciones de concreto hidraulico F'C= 100 kg/cm2 de 138 cm2 de sección (bordillos de 15 cm. de base mayor, 8 cm. de base menor y 12 cm de altura), con agregado máximo de 3/4"	M1	38,780.0000	5,861.15	227'295,897.00
Recubrimiento de cunetas con concreto hidraulico simple con agregado máximo de 3/4".	M3	1,510.0000	176,916.41	267'143,779.10
Lavaderos metálicos formados con tubo de lamina de acero con doble capa de cemento asfaltico de 60 cms. de diametro y calibre # 16.	M1	1,980.0000	89,899.61	178'001,227.80
			Subtotal	797'008,598.70

PAVIMENTOS				
Sub-base compactada al 100% con material del banco la ordena ubicado a 14,000 m a la derecha del kilometro 82+700 de la carretera Queretaro - Irapuato.	M3	6,630.0000	41,594.54	275'771,800.20
Base compactada al 100 % con material del banco la ordena ubicado a 14,000M a la derecha del kilometro 82+700 de la carretera Queretaro - Irapuato.	M3	6,340.0000	45,116.49	286'038,546.60
Material asfáltico, asfalto rebajado en riesgo de impregnación.	Lt	49,000.0000	45,116.49	286'038,546.60
Material asfáltico, asfalto rebajado en riesgo de liga.	Lt	18,000.0000	513.85	8'221,760.00
Material asfáltico, asfalto rebajado en riesgo de sello.	Lt	37,000.0000	513.85	19'012,820.00

Capítulo I Conceptos Básicos en la Industria de la Construcción

Material asfáltico, asfalto rebajado en carpeta asfáltica por el sistema de mezcla en el lugar.	Lt	539,000.0000	513.86	276'970,540.00
Aditivo Adiflex "go" para asfaltos rebajados.	Lt.	6,250.0000	10,090.09	63'063,052.50
Barrido de la superficie por tratar.	M3	3.0000	4'098,766.65	12'296,300.55
Carpeta asfáltica construida por el sistema de mezcla en el lugar con materiales ya producidos y tratados.	M3	4,316.0000	50,772.34	218'828,765.40
Riesgo de sello con materiales ya producidos y tratados utilizando material petreleo.	M3	300.0000	54,572.00	16'371,759.00
Acarreo de materiales petreos para pavimentación.	M3-Km	484,000.0000	389.64	188'585,760.00
Acarreo de materiales petreos para pavimentación cuando el volumen	M3-Km	20,000.0000	389.54	7'792,800.00

Capítulo I Conceptos Básicos en la Industria de la Construcción

acarreado para el riesgo de sello se determine.				
			Subtotal	1'398,133,074.25

SEÑALAMIENTOS				
Fantasma de Concreto hidráulico	Pza.	1,070.0000	15,931.95	17'047,186.50
			Subtotal	17'047,186.50

PASO A DESNIVEL SUBESTRUCTURA Y SUPERESTRUCTURA				
Excavación para estructuras cualesquiera que sea su clasificación y profundidad.	M3	1,563.0000	33,765.61	52'775,648.40
Mampostería de 3era. clase para cualquier altura con mortero de cemento.	M3	1,298.0000	116,836.91	150'615,909.18
Concreto hidráulico colado en seco.	M3	34.0000	233,990.78	7'955,686.12
Concreto hidráulico colado en seco para remodelación de coronas y diafragmas de estribo.	M3	12.0000	252,582.08	3'030,994.96

Capítulo I Conceptos Básicos en la Industria de la Construcción

Concreto hidráulico colado en seco en coronas de pilas.	M3	40.0000	252,119.09	10'084,763.60
Acero de refuerzo. varilla de límite elástico igual o mayor a 4000 Kg/cm ² .	Kg	4,005.0000	2,144.41	8'988,352.05
Concreto hidráulico colado en seco en diafragmas y losas sobre trabes prereforzadas.	M3	123.0000	248,823.81	30'605,328.63
Concreto hidráulico en guarniciones.	M3	28.0000	244,983.46	6'859,53.88
Juntas de dilatación mecánicas de acero estructural.	Kg	1,650.0000	3,091.58	5'101,107.00
Juntas de dilatación no metálicas de cartón asfaltado o material similar de 2 cms. de espesor.	M2	57.0000	4,042.22	230,406.54
Juntas de dilatación no metálicas de sikaflex 1-a o material similar.	M2	356.0000	9,631.28	3'428,735.68

Capítulo I Conceptos Básicos en la Industria de la Construcción

Trabes pretrenzadas de sección unicelular.	M3	206.0000	1'067,388.00	219'881,298.00
Apoyos de Neopreno.	M3	43.0000	51,550.40	2'216,667.20
Acero de refuerzo, varilla de limite elastico igual o mayor a 4000 Kg/cm2.	Kg	34,921.0000	2,144.41	74'884,941.61
Estructuras de concreto reforzado para petos de concreto para calzada.	M1	183.0000	42,142.36	7'712,051.88
Acero de preesfuerzo	Kg	10,270.0000	6,219.30	63'872,211.00
			Subtotal	647'844,269.16

Capítulo I Conceptos Básicos en la Industria de la Construcción

ACCESO				
Excavaciones en remoción de terraplenes existentescuando el material se utilice para la formación de terraplenes.	M3	1,650.0000	10,04.27	16'576,345.50
Formación y compactación de terraplenes.	M3	620.0000	14,661.49	9'090,123.80
Concreto hidraulico colado en seco en guarniciones sobre aleros.	M3	8.0000	348,490.07	2'787,90.56
Acero de refuerzo, varilla de limite elastico igual o mayor de 4000 Kg/cm2.	Kg	5,738.0000	2,144.41	12'304,624.58
Postes de concreto precolados.	M3	7.0000	808,029.13	5'656,203.91
Guarniciones coladas	M3	21.0000	244,530.40	5'135,138.40
Lavaderos de Mampostería de 3era clase.	M3	27.0000	305,482.20	8'248,019.40

Capítulo I Conceptos Básicos en la Industria de la Construcción

Defensas metálicas de lámina galvanizada.	M1	210.0000	84,993.58	17'848,651.80
			Subtotal	87'661,125.45

Capítulo I Conceptos Básicos en la Industria de la Construcción

OBRAS COMPLEMENTARIAS					
Demolición de muros de mampostería.	M3	268.0000	22,072.46	5'915,419.28	
Demolición de concreto reforzado en diafragmas y coronas.	M3	28.0000	227,156.86	6'360,392.08	
Demolición de guarniciones.	M3	5.0000	140,973.90	704,869.50	
Levantar las superestructuras existentes para incrementar el espacio libre, p.u.o.t, en el paso superior Irapuato II Bis.	Super	1.0000	136'072,866.8	136'072,866.84	
			Subtotal	149'053,547.70	
			TOTAL	3'924'637,053.1	

1.3 MODULO DE CONTROL DE OBRA

1.3.1 CALCULO DE ESTIMACIONES

La estimación de los costos de un proyecto de capital en todas las etapas del proceso de la construcción es de primordial importancia para el buen control administrativo. Sea que se administre para el cliente, el diseñador, constructor o usuario de la instalación, las estimaciones exactas y pertinentes reducen el desperdicio administrativo proporcionando una revisión constante de la viabilidad económica y lucratividad de una empresa. De esta manera se puede evitar realizar un esfuerzo en un proyecto que finalmente será "archivado", en una fase posterior del diseño o de la propuesta debido a que no tiene viabilidad económica. La estimación, o estimado proporciona la base para todos los pronósticos económicos y financieros así como para los presupuestos y el control.

Los propósitos de las estimaciones de la construcción se clasifican en general en tres formas:

- Estimaciones utilizadas para planeación y pronósticos con el objeto de que ayuden en las evaluaciones económicas y financieras de la inversión.
- Estimaciones de control que se hacen durante el diseño para asegurarse de que las evaluaciones económicas siguen siendo válidas conforme progresa el diseño.

- Estimaciones de la propuesta, que reflejan el costo que tiene para un constructor realizar el diseño terminado , permitiéndole de esta manera preparar una oferta para su presentación al cliente.

Para satisfacer las necesidades de las clasificaciones anteriores hay cuatro tipos de estimaciones:

- Estimaciones de planeación, que se utilizan en la primera clasificación.
- Estimaciones preliminares de ingeniería, que se usan en la segunda clasificación.
- Estimaciones detalladas de ingeniería, usadas en la segunda clasificación.
- Estimaciones en la fase de la construcción, que se emplean en la tercera clasificación.

AMBIENTE PARA LA ESTIMACION EN LA INDUSTRIA DE LA CONSTRUCCION

El propietario, diseñador, constructor y usuario consideran la estimación desde un punto de vista enteramente diferente. Los costos para el propietario incluyen más que los costos de la construcción física . Por supuesto, los costos de la construcción son el fundamento de esta información, pero para esta estimación base debe de añadir un conjunto completo de otros costos , dependiendo del tipo de instalación . Entre estos costos se incluyen los honorarios de diseño, seguros, honorarios establecidos, costos del terreno, costos de financiamiento, impuestos y otros semejantes. Por otro lado el diseñador considerará su estimación de los costos físicos del producto terminado para

pasárselos al cliente . Este costo muy bien puede incluir el amueblado completo , accesorios y equipo . También debe estar consciente de establecer "reservas" en su documentación para incluir cosas tales como sumas provisionales y costos de producción , que cubran productos especiales y condiciones desconocidas. Al constructor le preocupará preparar estimaciones del costo para su organización para la erección física de los elementos que contrate para construir e instalar , desde los cimientos hasta los acabados . Para esto debe añadir una utilidad que le proporcionará la organización con un rendimiento razonable por los riesgos que toma.

Por tanto, es importante que cada parte individual se interese por el costo de la estimación de la parte del proyecto de la cual es responsable , ya que cada una experimentará un nivel diferente de costos.

En cualquier etapa una estimación explica los costos pronosticados.

La exactitud de cada estimación varía con el grado de información que se tenga disponible sobre el proyecto a partir de la cuál se produzca. Según se desarrolla el diseño cada fase y especificaciones se vuelven más claras, se notan más detalles y progresivamente la estimación es más confiable.

Durante estas primeras etapas del diseño hay mucha incertidumbre, misma que debe ser compensada por una suma o una cantidad apropiada para contingencias con el objeto de cubrir conceptos que aún no están definidos. Conforme el diseño progresa y se especifican las áreas indefinidas , se reduce esta reserva para contingencias , y la estimación se detalla progresivamente.

Para cada estimación los detalles a los que se fijará precio habrán de dividirse en los componentes más pequeños posibles para evaluar y documentar en un marco de trabajo lógico. Este marco de trabajo que va desde la definición del alcance del proyecto hasta descripciones de los componentes del diseño totalmente detalladas, recibe un precio partiendo de los costos reales pertinentes y actuales analizados de una obra realizada con anterioridad de una naturaleza semejante.

En una etapa temprana de la concepción del proyecto se requiere planear la estimación para ayudar a determinar la factibilidad del proyecto y el desarrollo de las decisiones sobre política. Por tanto, ésta fase es probablemente la fase más importante de la estimación, pero, paradójicamente, es la fase en donde menos información se tiene disponible.

El concepto de los procedimientos de estimación anteriores varían bajo diferentes condiciones de adquisición del contrato de construcción que se establecen en los contratos de costos más honorarios fijos, costo a precio alzado, a precio unitario, o administración del proyecto. Bajo estas últimas condiciones habrán de considerarse refinamientos adicionales que modifican, amplían, o ajustan una estimación de construcción.

CONTRATOS DE COSTOS MAS HONORARIOS FIJOS

En aquellos proyectos en donde el alcance de la obra no está definido con claridad, tales como la renovación de unas instalaciones existentes, en donde está implicada

tecnología que no ha sido probada, o en proyectos confidenciales, se celebra un contrato de costos más honorarios fijos. Cuando el costo no es el interés prioritario, se le pueden dar instrucciones a un contratista para que proceda a realizar la obra, registrando todos sus costos conforme avanza. A los costos registrados se les añade la descripción por escrito de los gastos generales y utilidades cubiertos, y se le paga al contratista a intervalos convenidos. Sin embargo, en la mayoría de los casos es importante el costo final, aun cuando se desconozca la amplitud exacta de la obra. En este caso se negocian convenios de costo más honorarios fijos, que incorporan un gasto máximo posible o garantizado.

La estimación del costo máximo producido así incluye los gastos generales y la utilidad. Esta estimación refleja el precio máximo que tendrá que pagar el cliente por la obra contratada.

CONTRATOS A PRECIO ALZADO

De ordinario los contratos a precio alzado se adoptan para la construcción de instalaciones de una naturaleza "tipificada", proyectada y construida por la misma organización. En este caso el cliente sólo proporciona las especificaciones el tipo de rendimiento y los dibujos de la distribución preliminar. Entonces la organización especialista produce un diseño detallado y construye el proyecto.

Por consiguiente, las estimaciones incluyen el costo del diseño detallado y los planos de la obra así como la construcción. Debido a que los contratos de precio alzado son de una naturaleza común, se tienen disponibles costos "estándares", a partir de los cuáles se produce la estimación detallada en una etapa temprana debido a que los costos finales de una especificación de rendimiento relacionados con la construcción típica se conocen a partir de la experiencia pasada y de los procedimientos contables detallados.

CONTRATOS A PRECIO UNITARIO

Cuando se tienen disponibles planos y especificaciones y se define el alcance de la obra, pero sólo se pueden establecer cantidades aproximadas, se celebra un contrato a precio unitario. En este caso se requiere una estimación de precios unitarios, en donde toda la obra se divide en unidades mensurables para las cuáles se establece un costo, estimando analíticamente las constantes de mano de obra, uso y desperdicio de materiales, y requerimientos de planta-hora. A este costo unitario se le añade un porcentaje para cubrir los gastos generales y la utilidad. Se estima el reembolso para el constructor midiendo las unidades de obra conforme progresa el contrato y aplicando los precios unitarios estimados ofrecidos.

Este método de fijación de precio permite que el trabajo de construcción comience sin saber las cantidades exactas implicadas y resulta útil en las obras grandes de ingeniería civil que involucren grandes cantidades.

ADMINISTRACION DEL PROYECTO Y DE LA CONSTRUCCIÓN

Cuando el tiempo y la obra puede proseguir antes de que se termine el diseño final se tiene una situación contractual en la que se celebra un contrato de administración del proyecto o de la construcción.

Cuando esto sucede, los gerentes del proyecto o de construcción utilizan métodos de estimación típicos en la forma tradicional para hacer las estimaciones de la planeación, de ingeniería preliminar, y de ingeniería detallada.

Se lleva tiempo y, por tanto, cuesta dinero hacer la estimación. Mientras más detallada sea la estimación, más tiempo se requerirá para analizar e investigar, y en su preparación están involucrados más gastos. Fundamentalmente, mientras más exacta sea una estimación, será más costosa de preparar.

INFLACIÓN

La inflación es un proceso donde el poder adquisitivo del dinero disminuye al pasar el tiempo. Hay muchas razones económicas por las cuales ocurre la inflación, pero por sencillez consideraremos a este fenómeno como el resultado de dos influencias principales. La primera es que hay argumentos teóricos e históricos que respaldan el punto de vista de que la tasa de aumento en la existencia de dinero tenida por la comunidad, y que no viene emparejada por una tasa semejante de aumento en el nivel de productividad tarde o temprano dará a un aumento general en los precios. La

segunda es el aumento desigual en la existencia de dinero por la emisión adicional por parte del gobierno para financiar sus gastos.

Las causas inflacionarias anteriores ocurren debido a influencias de un mercado nacional. El mercado internacional también puede estar sujeto a inflación cuando la devaluación de la moneda de una nación en contra de la de otra aumenta el precio de compra de los bienes adquiridos en los mercados de cualquier otra nación.

El riesgo de que una estimación se vea afectada severamente por la inflación es mayor cuando el gasto estimado es a largo plazo. Invariablemente los proyectos de construcción implican compromisos a largo plazo, y de esta manera se exponen los costos estimados a los peores efectos de la inflación.

Cuando se preparan las estimaciones, habrán de considerarse y tomarse en cuenta los efectos de la inflación sobre los costos. Esto se hace de dos formas. Primero se expresa lo estimado en términos constantes, esto es, en un valor monetario constante que tenga un poder adquisitivo conocido. Así por ejemplo cuando se cotiza una cifra estimada, se dice que lo estimado para la construcción para una estación de energía es de N\$ 200 millones basándose en dinero de 1994, indicando así con claridad que cualquier inflación futura del dinero de 1994, afectará el valor estimado de la construcción.

Una segunda forma de tomar en cuenta los efectos de la inflación en los valores estimados de construcción consisten en anticiparse a la misma y estimar el costo evaluado de secciones de la obra en construcción en el momento en que se programan cronológicamente de manera que incluyan un factor inflacionario. Por ejemplo si el

costo estimado de la instalación de una planta de energía se programa que ocurra en "n" años en el futuro , la estimación actual del equipo será de N\$ 20 millones, y si la inflación sobre tal equipo es del 8% , entonces la estimación para esta parte de la obra será de N\$ 20 (1.08)ⁿ millones.

Sea cuál sea el método que se utilice, hay que recordar que el costo de una obra, esta compuesto fundamentalmente por salarios, materiales y uso de maquinaria, y que cada uno de estos conceptos se comporta de manera diferente en el fenómeno inflacionario, puesto que ninguno de los tres sufre el mismo porcentaje de incremento, ni entre ellos mismos existen las mismas variaciones. Esto quiere decir que en el caso de los salarios, el incremento correspondiente a un peón de la construcción no necesariamente es igual al incremento de salario para los operadores de maquinaria pesada o para el personal directivo de una obra . Así mismo, el incremento en el costos de los materiales, no necesariamente será igual para el acero, el cemento, los explosivos, etc.. como tampoco el incremento en el costo del equipo será igual para cualquier tipo de máquina.

Un sistema para hacer los ajustes de costos, es mediante el uso de la fórmula que se verá a continuación, cuyo lapso de aplicación (mensual, trimestral, semestral o anual) dependerá fundamentalmente de la magnitud de la inflación.

$$PF = P_i (.55 S_i/S_i + .30 M_i/M_i + .15 E_i/E_i)$$

en donde:

- PF** = El nuevo valor obtenido en el costo o precio unitario del concepto que se éste analizando.
- Pi** = Costo o precio unitario inicial o de la última revisión si ésta es periódica.
- Sf/Si** = La relación que existe entre el salario inicial al momento de la revisión (Sf) y el salario inicial (Si).
- Mf/Mi** = La relación que existe entre el costo actualizado de los materiales (Mf) y el costo inicial de los mismos (Mi).
- Ef/Ei** = El valor actualizado de la maquinaria considerada en forma de renta, depreciación, o valor de adquisición actualizado (Ef) al valor original (Ei).

ÍNDICES DE COSTOS

Otro sistema comúnmente aceptado para la corrección periódica de costos en épocas inflacionarias es el que se basa en la utilización de índices de costos, cuyo manejo debe ser cuidadoso, ya que cada uno de los insumos de la construcción pueden tener índices diferentes que no siempre es conveniente agrupar en un solo Índice General.

Se conoce como índice a un número cualquiera de una serie el cuál sirve de indicador de los cambios con respecto al tiempo que se producen en una variable con referencia

en una base arbitraria (comúnmente se toma como dicha base el 100), y representa el valor de la variable mencionada en un periodo específico previo.

Resumiendo podemos definir al número índice como una medida estadística utilizada para mostrar los cambios sufridos por una variable o grupo de ellas con respecto al tiempo, y en ocasiones con respecto a otros factores que afectan también a dicha variable.

Por lo tanto el índice de costo es la relación entre el costo de un determinado bien o servicio en un periodo dado y el costo del mismo bien pero en otro periodo tomado como base:

$$I_c = (C_n / C_1) \times 100$$

donde:

I_c = Índice de costo

C_n = Costo de un período "n" dado

C_1 = Costo de un período base.

Los índices de precios se dividen básicamente en dos clases:

- Índices de precios de insumos, que miden los cambios de precio que tienen lugar en los insumos de la construcción, tales como tasas de salarios, precios del material, o costos del equipo.
- Índices de producción, que miden los cambios de precios en los niveles de producción, tales como, el costo por kilowatt-hora por planta de

energía o el costo del acero estructural por tonelada que paga el consumidor de la construcción.

Cuando se prepara una estimación, las consideraciones anteriores tendrán un efecto considerable en el enfoque, manipulación y formato de las cifras resultantes, de manera que presenten una predicción exacta del costo y que sean un recurso administrativo útil.

Por ejemplo, tenemos que para octubre de 1994 la varilla del No. 3 (3/8"), tenía un costo de N\$ 12,200 dado que en octubre de 1993 su costo era de N\$ 5,215 el índice correspondiente en base 1986 será:

$$I_c = (12,200/5,215) \times 100 = 234.0 \%$$

De aquí podemos concluir que este producto tuvo un incremento de 234.0 % en el transcurso de un año, es decir casi se triplicó su valor.

A continuación se describen los cuatro tipos de estimaciones:

ESTIMACIÓN DURANTE LA PLANEACIÓN

En la fase de planeación del proceso de diseño hay poco material cuantitativo tangible para crear una estimación. Comúnmente sólo se tiene algo más que un breve diseño con los requerimientos de espacio o capacidad. Por consiguiente, la planeación de las

estimaciones normalmente se basa en costos de un solo parámetro. Estos costos de un solo parámetro se establecen a partir de datos sobre los costos históricos de proyectos anteriores de naturaleza semejante, analizados cuidadosamente para llegar a un costo base de la instalación.

Es importante utilizar los costos históricos de proyectos equivalentes recientes, en vez de basarse en su totalidad en los costos promedio publicados. Sin embargo, estos últimos se pueden utilizar para verificar o complementar los datos actuales analizados.

ESTIMACIÓN PRELIMINAR DE INGENIERÍA

Una vez tomada la decisión de pasar de la fase de planeación a la de diseño, se requiere elaborar una estimación más detallada. Se preparan planos preliminares para identificar los sistemas requeridos que formarán al proyecto. Se preparan planos preliminares y esbozos de las especificaciones en una forma que puede ser esquemática. Luego se calcula la estimación preliminar de ingeniería basándose en el análisis cuantitativo de los sistemas diseñados. Cualquier forma de estimación cuantitativa tiene tres constituyentes esenciales:

- El proyecto que se estimará
- Las unidades o sistemas que se considerarán
- La forma en que se expresará el resultado.

ESTIMACIÓN DETALLADA DE INGENIERÍA

No hay sustituto para las estimaciones que se hacen basándose en los diseños y especificaciones detallados de ingeniería. Un análisis cuantitativo de tales documentos junto con las tasas unitarias actuales obtenidas de las publicaciones de la industria y actualizadas producirá una estimación más exacta que las estimaciones de planeación o del diseño preliminar. Una estimación tal debe tener una exactitud dentro del 5%+-.

La estimación detallada de ingeniería está formada por estimaciones componentes, en las que las partes componentes son los conceptos detallados de un sistema de construcción. Cuando se diseñan y especifican estos componentes, se les puede cuantificar con exactitud mediante la medida de la cantidad necesaria. De nuevo, mientras, más abajo esté el nivel de identificación, medición y fijación de precios más exacto será la estimación resultante. Los documentos requeridos para este nivel de estimación son los planos detallados del diseño, los detalles normales, planos de taller y especificaciones detalladas. Las cantidades particularizadas junto con los insumos totalmente descritos de mano de obra y de materiales permitirán que se fije el costo de los componentes, con precisión.

ESTIMACIÓN EN LA FASE DE CONSTRUCCIÓN

Esta estimación se lleva acabo por el contratista que concurra en un proyecto. En esta etapa deben estar disponibles para el estimador los planos y las especificaciones detalladas , junto con un programa cronológico de la construcción para establecer la duración de la obra . Debe llevarse a cabo una inspección extensa del lugar para asegurarse de las implicaciones en términos de costo de cualesquiera condiciones únicas del mismo. Se documentará una estrategia previamente planeada para llevar el proyecto y a partir de esto se debe hacer un análisis completo de los recursos requeridos. El estimador o evaluador debe tener una buena comunicación con los proveedores y los subcontratistas.

Hasta aquí la estimación representa los costos en los que incurrirán al llevar a cabo un proyecto . Ahora es necesario añadir una utilidad pronosticada para establecer la cifra de oferta o la cifra que se va a ofrecer en el concurso.

La utilidad representa un rendimiento sobre el capital invertido junto con una contingencia por riesgo . La utilidad esperada en las operaciones de riesgo bajo debe ser menor que la que se aplica a las operaciones con riesgo alto.

A continuación se muestra un análisis de los tipos de estimaciones.

La magnitud de variabilidad inherente a cada nivel de estimación reflejan las curvas de contingencia que aparecen a la derecha de la figura 1.3.1.1. La variabilidad es, por supuesto, bastante alta al nivel de orden de magnitud y decrece al intervalo de 3 a 5 por ciento al tenerse disponibles los documentos para el nivel de cotización.

PREPARACIÓN DE ESTIMACIONES DETALLADAS

La preparación de una estimación detallada a nivel de cotización requiere que el estimador desglose el proyecto en sus centros de costos o subelementos de costo. Es decir, que se separe el proyecto en los subcomponentes que habrán de generar gastos.

Aunque el proceso de estimación es en parte arte y en parte ciencia, el estimador sigue generalmente ciertos pasos para hacer su estimación:

1. Divide al proyecto en centro de costos.
2. Estima las cantidades requeridas por centro de costos que representan conceptos físicos finales (por ejemplo yardas cúbicas de tierra, pies lineales de tubería, etc.) . Para los sistemas físicos se llama comúnmente a este procedimiento " toma de cantidades" . Para aquellos centros de costos que no se relacionan con ningún concepto físico , determina un parámetro apropiado para el calculo del costo (por ejemplo el nivel de seguro por riego del constructor que requiere el contrato , por sus términos o los montos de las fianzas que se pidan en el mismo).

3. Calcula los precios de las cantidades determinadas en el paso 2 utilizando datos históricos , cotizaciones de proveedores, catalogo de fabricantes y otra información que contenga precios . Esta asignación de precios puede basarse en un precio por unidad (costo unitario) o puede hacerse con base en sumas totales (por trabajo) . El desarrollo de precios por conceptos físicos de obra requiere de un análisis de las tasas de producción que habrán de lograrse, basado en el análisis de recursos. Si se utiliza este análisis el estimador debe:

- a.- Asumir una composición de cuadrillas que incluya el número de trabajadores (especializados, y no especializados) y el equipo requerido.
- b.- Con base en la composición de cuadrillas , estimar una tasa de producción por hora de acuerdo con la tecnología que se vaya a aplicar.
- c.- Hacer una estimación de la eficiencia que habrá de lograrse en la obra , considerando los factores del sitio y otros factores.
- d.- Calcular el precio unitario efectivo .

4. Calcula el precio total de cada centro de costo multiplicando la cantidad requerida por el precio unitario . A esta multiplicación se le llama comúnmente " extensión" , y a este proceso se le conoce comúnmente como "corrida de las extensiones".

Uno de los procedimientos de estimación más comunes es el llamado **Método de ENUMERACIÓN DE RECURSOS**.

En la figura 1.3.1.2 se muestran los pasos que intervienen para aplicar el enfoque de enumeración de recursos.

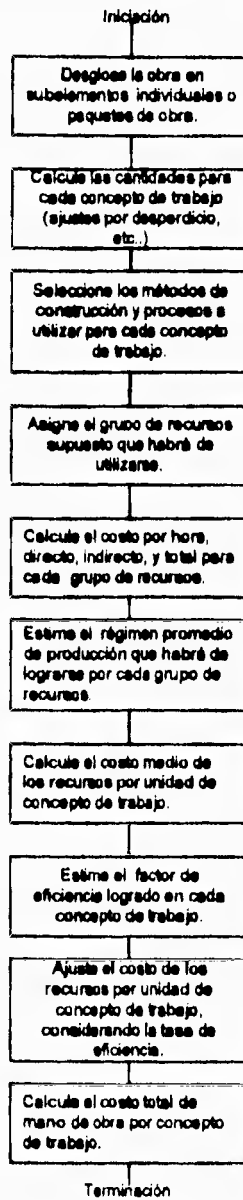


Fig. 1.3.1.2 Método de Estimación por enumeración de recursos.

En la figura 1.3.1.3 se presenta un ejemplo de enumeración de recursos a una operación de vaciado de concreto.

CUADRILLA DE COLOCACION DE CONCRETO

CANT.	MIEMBRO	TARIFA SALARIO	TOTAL/HORA
1	Sobrestante de Carpintería	9.42	9.42
2	Albañiles de Cemento	8.36	16.74
1	Encargado de bombas	8.45	8.45
7	Operarios	5.43	38.01
1	Bomba de Concreto	8.00	8.00
Salario de cuadrilla por hora :			80.60

Cimentación	53.2	15	0.9	\$ 7.47	\$ 398
Viga elevada	8.1	15	0.7	\$ 9.00	\$ 101
Muro interior	5.5	15	0.4	\$16.80	\$ 93
Piso elevados	8.9	10	0.9	\$ 7.47	\$ 2,028
Costo directo total de mano de obra por concreto:					\$ 2,620

1.4 PLANEACIÓN DE PROYECTOS

1.4.1 INTEGRACIÓN DE DATOS DEL PRESUPUESTO

Existe una multitud de modelos, muchos de ellos elaborados por los propios constructores. Sin embargo, se presentan los que por la práctica se consideran los más aceptables:

MODELO 1

Este modelo es usado solamente para presupuesto técnico y en él no se toman en cuenta los precios, tanto parciales como totales . Este formato pasa después a la segunda fase , que es la de aplicación de precios y que corresponde al modelo 2. Ver fig. 1.4.1.1

MODELO 2

Este otro formato es para precios parciales de obra.

Aquí como antes se decía, se aplican los precios en vigor y una vez cubiertos todos los ramos que entran en la obra se pasan al modelo numero 3 en el cuál van resumidas las distintas unidades de obra para llegar al presupuesto general de la obra . Ver fig. 1.4.1.2 y 1.4.1.3 respectivamente.

MODELO 4

Este formato es quizás el más completo, puesto que en el mismo se incluyen tanto el presupuesto técnico como el económico. Ver fig. 1.4.1.4

PRESUPUESTO TECNICO DE LA OBRA

Obra:

Propietario:

Observaciones:

No. Orden	Designación de la Clase de Obra	No. Partes Iguales	Largo	Ancho	Alto	Peso	Total	Unidad Base

Fig.1.4.1.1. Presupuesto Técnico

PRESUPUESTO PARCIAL DE LA OBRA

No. de Unidades	Designación de la Clase de Obra	Precio de la unidad	Importes Parciales	Importes Totales

Fig.1.4.1.2 Presupuesto Parcial

PRESUPUESTO GENERAL DE LA OBRA

	Importes Parciales	Importes Totales

Fig.1.4.1.3 Presupuesto General

PRESUPUESTO TECNICO Y ECONOMICO DE LA OBRA

Obra:

Propietario:

Observaciones

No. Orden	Designación de la Clase de Obra	No. Partes Iguales	Largo	Ancho	Alto	Peso	Total	Unidad Base	Precio	Total

Fig.1.4.1.4 Presupuesto Técnico y Económico

1.4.2 MATRIZ DE SECUENCIAS

La matriz de secuencia tiene por objetivo ordenar la secuencia de las actividades. Con la matriz de secuencia se puede conocer cuáles etapas son simultáneas, cuáles dependen de otras y cuáles coinciden en el mismo punto dentro de la ruta crítica.

Para realizar una matriz de secuencia, es necesario considerar que toda red debe comenzar en un evento cero y terminar en otro final y que a pesar de que no consumen tiempo, permiten detectar el origen y fin de la red. Por ejemplo si en un proyecto tenemos cinco actividades, este número corresponderá al número de columnas y renglones que debe tener dicha matriz, como se muestra en la siguiente figura:

	A	B	C	D	E
A		X			
B			X	X	X
C				X	
D					X
E					

Fig. 1.4.2.1 Muestra de Matriz de secuencias de 5 actividades

La matriz se divide en dos triángulos, esto es consecuencia de que la Actividad 'A' no puede seguir la 'A' ni de la 'B' puede seguir la 'B' y así sucesivamente; es decir, no pueden seguirse a sí mismas, por lo que eliminamos estos cuadros. Ahora bien, al

dividir en dos la matriz tenemos un lado superior derecho y uno inferior izquierdo. Cualquier marca realizada en el lado inferior izquierdo indicará que la secuencia que elegimos no es la adecuada; todas las marcas deben quedar en el lado superior derecho, porque de la actividad "B" no puede seguir a la actividad "A", ni de la "C" la "B", pero sí a la inversa.

En el eje vertical tendremos la actividad inicial y en el horizontal la final, es decir que si la actividad "A" sigue la "B" pondremos una marca en el primer renglón y la segunda columna por ejemplo:

		Actividad Final		
		A	B	C
Actividad Inicial	A		X	
	B			
	C			

Fig. 1.4.2.2 Muestra de actividades Inicial y Final

Por ejemplo en un proyecto de Obra podemos citar como actividades "Desmonte del terreno" es la actividad "A", "Despalme" es la actividad "B", "Trazo y Nivelación" es la actividad "C", "Excavación" es la actividad "D", "Rellenos" es la actividad "E", "Acarreos" es la actividad "F", "Plantillas" es la actividad "G", "Mampostería" es la actividad "H", "Cimbra y Descimbra en cimentación" es la actividad "I", "Acero de Refuerzo" es la actividad "J", "Concreto en cimentación" es la actividad "K", "Cadena de Desplante" es la actividad "L", "Cimbra y Descimbra en Estructura" es la actividad "M", "Acero de

Actividad	Actividades Anteriores	Actividades Posteriores
A	O	B
B	A	C
C	B	D,E,F
D	C	G
E	C	I
F	CD	H
G	F	I
H	E,G	G
I	H,G	FIN

Es necesario recalcar que la matriz de secuencias es indispensable para elaborar la red que permitirá obtener la ruta crítica, ya que facilita de tal manera el proceso que sólo restaría leerla paso por paso para obtener la red.

Una vez realizada la matriz secuencial, ya se cuenta con todos los elementos para elaborar la red, colocar las actividades ficticias y determinar "la ruta crítica". El método de la ruta crítica requiere de la determinación de diagramas y modelos de red que describan específica y únicamente las características del proyecto de construcción en consideración. Es muy importante desarrollar la red con suficiente detalle, para así mostrar con validez las características de los métodos de construcción que han de adoptarse, las necesidades de información que fijan la gerencia y el departamento de planeación, marcarán la magnitud y el detalle necesario de la red.

Un diagrama de red, o modelo, es determinístico, en virtud de que es la representación rígida de una forma específica de hacer un trabajo y, por lo mismo es esencial decidir los métodos de construcción, antes del trazo de la red. Esto no quiere decir que se niegue flexibilidad a la planeación, por el contrario, la flexibilidad es asegurada al

considerar tantas posibilidades como se desee, para la realización del proyecto; al revisar cada una de éstas se podrá escoger la mejor.

La ventaja sobresaliente del modelo de red es que obliga a una presentación completa y precisa de todas las actividades de un proyecto, desde su inicio, por medio de todas sus relaciones, hasta su terminación. Directores y planificadores deben pensar la forma de ejecutarlas, desde el principio hasta el fin, antes de que el trabajo se inicie. Al proceder en esta forma, y al mostrar la red relaciones insospechadas previamente, podrán surgir métodos y secuencias de construcción más eficientes.

Un proyecto de construcción es un conjunto de operaciones individuales o actividades. El orden en el que las actividades se inician y la relación de unas con otras, constituye el plan de construcción. El diagrama de red podrá terminarse, solamente después de decidir la colocación de actividades y su orden.

El primer paso en la elaboración de una red es la división del proyecto en sus actividades. La mejor manera de disponerlas es colocándolas en columnas. No es necesario un orden específico de procedencia, ya que dará mejor resultado un registro sistemático por oficios, o por forma de trabajo, localización, o equipo requerido.

El siguiente paso es formular una lógica de construcción o un orden específico de actividades. Esto implica un planteamiento preciso de las relaciones entre todas ellas. La ordenación general de las actividades de un proyecto no es difícil, ya que muchas veces su propia descripción implica una localización relativa dentro del trabajo; sin embargo, el orden es más difícil y requiere de consideraciones cuidadosas.

Una buena aproximación a la ordenación específica es determinar primero las restricciones físicas y de seguridad evidentes, después las limitaciones de mano de

obra y otros recursos, finalmente las limitaciones de administración. Las restricciones físicas nos conducen, al principio a cadenas de actividades fáciles de determinar y asociar.

La consideración de otras restricciones y la determinación detallada de las necesidades físicas, por lo general, nos lleva a una ramificación de las cadenas y a un cambio de éstas a redes. Es por esto que resulta ventajoso tabular las actividades sistemáticamente, para diferenciar aquellas que deben preceder de las que deben seguir a cada actividad, y las que pueden llevarse a cabo simultáneamente. El bosquejo de red queda, entonces, determinado por proceso de aproximaciones sucesivas primero satisfaciendo algunas condiciones y, después, refinando las porciones restantes de la red que resulten afectadas.

Para diseñar una red que satisfaga todas las restricciones, se requiere una gran habilidad. En algunos casos, las decisiones de la administración son difíciles para transcribirlas a un diagrama; sin embargo, es más o menos sencillo probar una red dada. En consecuencia, para determinar mejoras en un diagrama, es más fácil comenzar con un bosquejo rudimentario de red, incluyendo sucesivamente los detalles, que intentar un diagrama detallado desde el principio.

Por ejemplo, considérese una construcción simple de cimentación de concreto que incluya excavación, colocación del acero de refuerzo, cimbra y colado. Una lista de actividades podría ser la siguiente:

- A. Localización
- B. Excavación

- C. Cimbrado**
- D. Colado**
- E. Obtención del acero de refuerzo**
- F. Corte y doblado de varillas**
- G. Colocación del acero de refuerzo**
- H. Obtención del concreto**

La lista de actividades fue hecha en el orden que fueron pensadas, sin ninguna ordenación definida para un plan de construcción específico. La identificación de las actividades, en este caso, está dada por letras del alfabeto; pero podría darse, por medio de cualquier clave, por ejemplo, de acuerdo con la forma de trabajo, localización, etc. usando una notación de letras y números combinada.

Un examen detallado de la lista de actividades nos muestra que se pueden agrupar fácilmente. Así, si consideramos sólo las restricciones físicas, podremos desarrollar las siguientes cadenas físicas:

1. Bajo consideración la cimentación:
 - A - Localización.**
 - B - Excavación**
 - C - Cimbrado**
 - G - Colocación del acero de refuerzo**
 - D - Colado**
2. Bajo consideración del acero de refuerzo.
 - E - Obtención del acero de refuerzo**

F - Corte y doblado de varillas

G - Colocación del acero de refuerzo

D - Colado

3. Consideración únicamente del concreto

H - Obtención del concreto

D - Colado

Una vez que se ha observado el proyecto, desde estos diferentes puntos de vista, surgen cadenas individuales de actividades; pero al observar el trabajo como un todo, es obvio que existen relaciones entre ellas. Por ejemplo, es absurdo colar, antes de colocar los refuerzos e instalar la cimbra. Por lo tanto, todas las cadenas deben ser terminadas antes de colar. Si se deben tomar medidas para iniciar inmediatamente los trabajos de acero y concreto (ésta sería una decisión o restricción de la gerencia) entonces, todas las cadenas se iniciarán en el mismo punto o evento, con el trazo de los cimientos.

En el proceso de construcción la lógica se formula considerando las siguientes preguntas, que (aunque asociadas específicamente a cada una de las actividades) proporcionan las relaciones apropiadas que han de establecer entre los eventos y actividades.

1. ¿Qué actividades deben ser terminadas inmediatamente antes del inicio de ésta?
2. ¿Qué actividades son independientes de ésta y pueden realizarse simultáneamente con ella?

3. ¿Qué actividades deben iniciarse, inmediatamente después de la terminación de ésta?

Las respuestas a estas tres preguntas especifican las restricciones físicas del proyecto. Si se cambian las palabras "deben" y "pueden", por "serán"; entonces, las respuestas a las preguntas especifican que han aparecido restricciones de la gerencia. Además si la palabra "pueden", de la segunda pregunta, se reemplaza por "no pueden", las repuestas indicarán restricciones de seguridad, recursos de la mano de obra.

Toda red debe cumplir con las siguientes restricciones:

- Toda red debe iniciarse en un evento cero y terminar en uno final.
- Todos los eventos deben numerarse

Para armar una red sólo tenemos que seguir los pasos que nos marca la matriz secuencial.

También en la definición de una red es necesario considerar actividades ficticias, las cuales no consumen tiempo ni recursos, está permite unir dos o más actividades de una red.

La actividad ficticia tiene dos funciones básicas. La primera separa dos o más actividades que llegan o salen de un solo punto

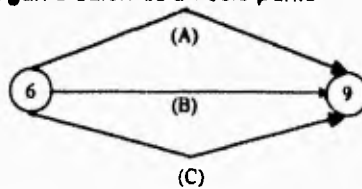


Fig. 1.4.2.3 Representación de una red gráfica de actividades

Este caso evidencia un grave error, dos o más actividades no pueden salir de un punto 6 y llegar al mismo punto 9 simultáneamente, porque no se sabría cuál es cuál, aparentemente existen tres actividades que tienen un inicio "6" y un fin "9", lo que resulta muy confuso, por esta razón se utilizan las actividades ficticias

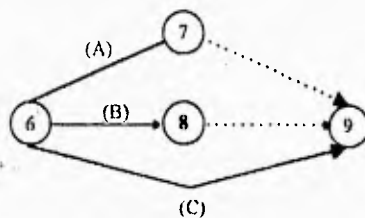


Fig. 1.4.2.4 Representación de actividades ficticias

La segunda función es evitar que existan actividades sueltas, que estén unidades a la red por un lado, con las actividades ficticias podemos unir actividades sueltas para unir las al principio y al fin de la red

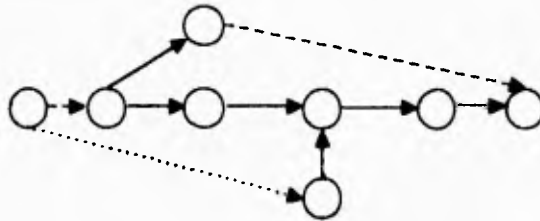


Fig. 1.4.2.5 Representación de actividades sueltas

Una vez elaborada la red, corregida, mejorada, anotamos sus tiempos y agregamos las actividades ficticias, el siguiente paso será determinar la ruta crítica.

La ruta crítica es el camino más largo que se puede seguir en tiempo desde el evento "cero" hasta el evento "fin". Se preguntarán ¿por qué el camino más largo? y la respuesta es simple, para poder terminar el proyecto requerimos que todas las actividades estén terminadas.

Todas las actividades que caigan dentro de la ruta crítica serán consideradas críticas, es decir que no se pueden atrasar ni un instante, porque todo el proyecto se retardaría. Aquellas actividades que no sean críticas tendrán un margen para iniciarse o terminar que se le llama holgura y que se determinan mediante la tabla de tiempos próximos y remotos.

Para determinar la ruta crítica, debemos encontrar el camino más largo entre la actividad cero y la actividad final, para detectar este camino se dividirá la red en todos los caminos posibles, marcando las secuencias de eventos, actividades y la suma de tiempos. Todas las actividades que se encuentren dentro de la ruta crítica serán consideradas como críticas.

1.4.3 BARRAS DE GANTT

El Sr. Henry L. Gantt desarrollo un método gráfico que representa las actividades a través de una escala de tiempo. Inicialmente se le llamaban diagramas de barras, estas representaciones gráficas han sido renombradas Barras de Gantt en honor al inventor del sistema. Las barras de gantt nos permiten determinar en cualquier momento, el estado y proyección de cada una de las actividades así como del proyecto mismo.

Dentro de las barras de Gantt la representación de las actividades se realiza por medio de barras de colores cuya longitud equivale a la duración en tiempo de la actividad.

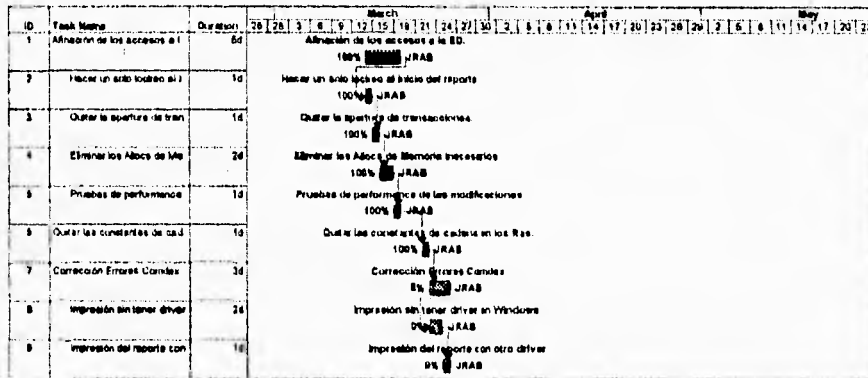


Fig. 1.4.2.6 Ejemplo de Barras de Gantt

El objetivo de las barras de Gantt es localizar fácilmente los tiempos en que una actividad determinada puede iniciarse o terminarse, alternativamente o iniciarse y terminarse simultáneamente.

Toda actividad tiene cuatro tiempos, dos de iniciación y dos de terminación. Al tiempo más cercano a cero se le toma tiempo próximo y al más lejano se le llama remoto; así nos encontramos con dos tiempos de iniciación -uno próximo y otro remoto-. Cuando en una actividad el tiempo próximo y el remoto son iguales, es decir que tienen que empezar o terminar en un único tiempo, se les llama actividades críticas.

Esto como ya se dijo significa que no tienen holgura; cuando la diferencia de los tiempos de iniciación y de terminación es igual a dos días, por ejemplo, significa que tienen dos días de holgura, es decir que pueden atrasarse hasta dos días sin retrasar el proyecto.

Por todo lo anterior es por lo que se considera importante realizar las barras de Gantt, ya que de esta forma se puede comprobar la ruta crítica, determinar holguras y obtener gran cantidad que se irá analizando.

Ya que se ha definido qué es la holgura; lo que todavía no se ha dicho es para qué sirve y cómo se optimiza.

Las holguras o actividades con holguras nos sirven para:

- Desahogar tensión de los involucrados en el proyecto, al saber que no existen límites para la realización de "x" actividad.
- Regular nuestros gastos, al poder mover alguna actividad dentro de su margen de iniciación y terminación.
- Manejar nuestros recursos humanos de manera más eficiente, al poder tener flexibilidad en horarios, fechas y tarifas.

Existen dos tipos de holgura: la independiente y la dependiente. La primera es cuando la holgura pertenece a la actividad, y la segunda cuando la holgura se la reparten entre varias actividades.

Para poder optimizar las holguras es necesario racionalizar su uso y no desperdiciar la flexibilidad que nos proporcionan en el manejo de los recursos humanos y materiales.

Dentro de todas las etapas de la técnica de la ruta crítica, una muy importante es el análisis de los costos, ya que el tiempo y los recursos humanos y materiales están íntimamente ligados. Veamos el procedimiento para la evaluación y optimización de los costos de un proyecto.

1. Calcular el número de recursos necesarios por unidad de tiempo.
2. Convertir estos recursos en unidades monetarias
3. Separar las actividades críticas de las no críticas
4. Convertir las flechas de las actividades en barras de tiempo
5. Colocar primero las actividades críticas en la gráfica
6. Colocar abajo de las críticas las no críticas
7. Anotar los recursos monetarios sobre las barras
8. Convertir las barras de actividades críticas en bloques cuyo volumen dependerá de los recursos monetarios necesarios
9. Sobreponer las actividades no críticas, también en formas de bloques sobre las críticas.
10. Equilibrar los recursos en forma ideal.

1.4.4 VOLUMEN DE OBRA

Los reportes de volúmenes de obra tienen como objetivo proporcionar un panorama general de la distribución del volumen de Obra a ejecutar durante el transcurso del proyecto.

El cálculo de los volúmenes se desglosa para cada actividad considerando una distribución uniforme, es decir, el volumen total de la actividad se dividirá entre el número de días asignados como duración, y de esta forma se calcula el volumen según el número de días que tenga el período. De esta forma una actividad que dispone de más días hábiles durante el mes tendrá un volumen a ejecutar mayor.

Cabe mencionar que el criterio para determinar que una actividad registre un comportamiento en su volumen de tipo uniforme, está íntimamente relacionado con el diseño de su Ruta Crítica, es decir, si una actividad no ha sido suficientemente desglosada en una serie de actividades más detalladas que muestren una evolución con más apego a la realidad, los resultados producidos en este reporte de Volúmenes será mucho más difícil de interpretar que los generados bajo una mejor Ruta Crítica.

El concepto de volumen de obra se refiere a la cantidad de obra que debe ser ejecutada para cada período de acuerdo con la planeación del proyecto. De la misma forma en que se tienen fechas de inicio y terminación para actividad, se tiene un volumen a ejecutar, lo cual representa costos, consumo de insumos pero también representa un avance dentro de la programación del proyecto.

1.4.4 VOLUMEN DE OBRA

Los reportes de volúmenes de obra tienen como objetivo proporcionar un panorama general de la distribución del volumen de Obra a ejecutar durante el transcurso del proyecto.

El cálculo de los volúmenes se desglosa para cada actividad considerando una distribución uniforme, es decir, el volumen total de la actividad se dividirá entre el número de días asignados como duración, y de esta forma se calcula el volumen según el número de días que tenga el período. De esta forma una actividad que dispone de más días hábiles durante el mes tendrá un volumen a ejecutar mayor.

Cabe mencionar que el criterio para determinar que una actividad registre un comportamiento en su volumen de tipo uniforme, está íntimamente relacionado con el diseño de su Ruta Crítica, es decir, si una actividad no ha sido suficientemente desglosada en una serie de actividades más detalladas que muestren una evolución con más apego a la realidad, los resultados producidos en este reporte de Volúmenes será mucho más difícil de interpretar que los generados bajo una mejor Ruta Crítica.

El concepto de volumen de obra se refiere a la cantidad de obra que debe ser ejecutada para cada período de acuerdo con la planeación del proyecto. De la misma forma en que se tienen fechas de inicio y terminación para actividad, se tiene un volumen a ejecutar, lo cual representa costos, consumo de insumos pero también representa un avance dentro de la programación del proyecto.

Es importante considerar que el volumen de obra programado es quien da origen a la explosión de insumos en el tiempo, ya que dependiendo de las cantidades a realizar de cada actividades puede calcular los insumos necesarios para su realización.

El consumo de insumos para las actividades representa los costos programados para el constructor, y el precio unitario de cada actividad multiplicado por el volumen a realizar en cada periodo proporciona una aproximación a la forma en que deberán cobrarse los avances del proyecto.

Dependiendo de la actividad existen 2 formas de distribuir el volumen a realizar en un determinado numero de días.

La primer forma se conoce como 'Distribución Lineal', en la cual se divide el volumen total de la actividad entre el número de días de su duración, arrojando como resultado el volúmen que debe ejecutarse cada día, cada semana o cada mes. Sin embargo no siempre se lleva a cabo de esta forma, existen actividades que por sus características pueden se distribuidas en forma 'variable' en el tiempo y que incluso pueden ser intermitentes, son aquellos casos que cada día se realiza un volumen diferente de obra, es decir el primer día se realizan 10 M2 de limpieza de terreno, al siguiente día se avanza sólo 5m2 y si existen días en los cuales el volumen programado deberá ser 0, con lo cual se convierte en una actividad intermitente, que no siempre se esta llevando a cabo (con interrupciones).

El volumen de obra también puede calcularse en función al rendimiento que tiene una cuadrilla dentro de un concepto, de forma que el avance planeado podrá definirse en base a las cuadrillas que se incluyan dentro de una actividad. A continuación se muestra un ejemplo para aclarar este concepto. Consideremos el siguiente análisis:

TRAZO Y NIVELACION DE TERRENO (60 M2)

Componente	Cantidad	Rendimiento
<i>Cuadrilla Básica.</i>	1	24 M2/Jor
<i>Cal Hidratada.</i>	0.003	1
<i>Madera Pino 3a. de 2"x4"x8.25"</i>	0.042	1
<i>Madera Pino 3a. de 4"x4"x8.25"</i>	0.051	1
<i>Madera Pino 3a. de 1"x4"x8.25"</i>	0.027	1
<i>Carrete de hilo plastico</i>	0.003	1
<i>Nivel Basculante Rossbach</i>	0.033	1
<i>Tránsito Rossbach</i>	0.033	1

Dependiendo del número de cuadrillas que se integren al proyecto, se calculará el volumen periódico aproximado, en este caso si sólo se considera una cuadrilla, el volumen diario sería de 24 M2 y la duración de la actividad sería de 2.5 días. En caso de querer integrar 2 cuadrillas, el volumen diario 48 y la duración de 1.25 días. Aquí puede apreciarse como del análisis en el presupuesto se pueden hacer precalculos de duraciones y volúmenes a manejar en la planeación.

Anteriormente se menciono que una vez contando con los volúmenes de obra a ejecutar en cada periodo del proyecto se puede obtener información que permite conocer cuales serán los gastos a considerar, esta información es de suma utilidad

para la elaboración de programas de compra, planeación de flujos de efectivo, políticas de pago y cobranza, planeación de almacenamiento.

Para cada precio unitario que se encuentre programado en un periodo debe calcularse los costos de acuerdo al volumen de obra calculado para ese periodo. Del ejemplo anterior, considerando el primer caso, de una sola cuadrilla tenemos que en el 1er día se tiene un volumen de 24 M2 de trazo y nivelación, de lo cual podemos obtener la siguiente información:

Información para el primer día

Insumo	Cantidad	Costo	Importe
<i>Cuadrilla Básica.</i>	1	231.00	231.00
<i>Cal Hidratada.</i>	0.792	227.68	180.32
<i>Madera Pino 3a. de 2"x4"x8.25"</i>	1.008	2.83	2.85
<i>Madera Pino 3a. de 4"x4"x8.25"</i>	1.22	2.71	3.31
<i>Madera Pino 3a. de 1"x4"x8.25"</i>	0.648	2.83	1.83
<i>Carrete de hilo plástico</i>	0.792	9.85	7.80
<i>Nivel Basculante Rossbach</i>	0.792	0.58	0.46
<i>Tránsito Rossbach</i>	0.792	1.00	0.792
			<hr/>
	Importe necesario para cubrir el 1er día		428.36

Debe notarse que teniendo los volúmenes de información calculados, es posible generar una gran cantidad de información y reportes útiles para tener una buena planeación del proyecto.

1.4.5 EGRESOS

Dentro de la Planeación de Proyectos es necesario estructurar un plan general a seguir para la implementación de un control de costos que permita supervisar y controlar los egresos (Costos) durante toda la vida del proyecto. Su función más importante será tratar de mantener los costos del diseño y los de construcción dentro de los límites de los estimados y presupuestos originales y anticiparse en una etapa temprana a cualquier sobregasto no previsto.

CONTROL DEL PROYECTO

Para la correcta administración de un proyecto se requiere de un control efectivo sobre parámetros que definan las variables de plazos, costos, y calidad, para ello la administración requiere de información que le muestre la situación del proyecto, esta será comparada con el presupuesto (modelo) del proyecto.

Si existen diferencias detectadas respecto al cumplimiento de plazos y presupuestos y estas son menores, se mantendrá la planificación inicial. Si por el contrario, comienzan a ser significativas, será necesario adoptar "medidas correctivas" tendientes a absorber las desviaciones. En su defecto se deberán modificar los objetivos entrando a un proceso de REACTUALIZACION DE PROGRAMAS Y PRESUPUESTOS. El efecto que tengan las nuevas políticas, deberá ser observado e informado. Las diferencias darán paso a nuevas medidas.

Para administrar un proyecto es necesario:

- Definir los límites del Proyecto y de los diferentes contratos y servicios que lo configuran.
- Formular un modelo en base al cual se desarrolla un programa y un presupuesto.
- Conocer lo que ocurre.
- Comparar las expectativas definidas por el modelo con la realidad.
- Analizar información, identificar desviaciones e informar a los niveles ejecutivos adecuados.
- Adoptar políticas correctivas.
- Proyectar la situación actual hasta el término de la obra.

La formulación del modelo debe hacerse en términos que posteriormente hagan posible una medida y comparación.

El conocer lo que ocurre implica medir elementos que están contenidos en la formulación del modelo.

La comparación de las expectativas con la realidad deben hacerse a intervalos cortos de manera que sus conclusiones sean oportunas para posibilitar la adopción de políticas correctivas que tengan efecto sobre el curso del proyecto.

El análisis de la información, la identificación de problemas y la adopción de políticas correctivas constituyen una rutina que tendrá validez a lo largo de toda la ejecución del proyecto.

Un control del proyecto consiste en:

- Establecer los mecanismos que permiten detectar qué ocurre en términos de cumplimiento de plazos, compromiso de presupuestos, obtención de la calidad. Más propiamente corresponde a un Monitoreo del Proyecto (Project Monitoring).
- Ejercer una presencia activa de Administración de Costo. Para ello define, implanta y controla políticas que a través de la eficiencia y la optimización en el uso de recursos mejora los resultados. Se le llama Control del Proyecto. (Project Control).

El control del proyecto comienza en un efectivo monitoreo del proyecto.

ESTABLECIMIENTO DE UN SISTEMA DE CONTROL

Los criterios básicos que deben ser asegurados al momento de establecer un sistema de control son:

El sistema de control contiene información de:	Las características de la información son que sea:
Costo	Relevante
Avance	Oportuna
Comportamiento	Confiable
	Flexible
	Económica

El sistema será útil sólo en la medida que la información que produzca sea de utilidad para quién está destinada. Para ello:

- Debe ser entendible para el usuario.
- Debe estar orientada hacia la toma de decisiones.
- Debe ser preparada y planificada de acuerdo a las necesidades específicas del proyecto, y particulares del usuario.
- Debe asociarse el grado de detalle de la información con el nivel ejecutivo que la recibe. Se le debe organizar para entregar a cada nivel el detalle que necesita. Debe, no obstante, ser consecuente y coherente en todos sus niveles ya que representa una realidad única.
- Debe ser compatible con las necesidades funcionales.
- Debe registrar las excepciones, identificar áreas, situaciones y condiciones críticas, ilustrando posibles acciones correctivas.

La confiabilidad en la fidelidad de la información proporcionada es condición necesaria para que ella sea utilizada por quienes deben adoptar las decisiones.

CAMPOS DE APLICACIÓN DE UN SISTEMA DE CONTROL

El control implica evitar el mal uso, la utilización descontrolada, y la mala utilización de tiempo, materiales, mano de obra y energía. La Administración de los recursos debe estar en mano de las personas que determinan su empleo. De ello derivan los principales campos de aplicación de un Sistema de Control.

- Diseño.
- Compras
- Contrataciones.
- Programación de la construcción.
- Costos.
- Calidad.
- Seguridad
- Puesta en marcha.

PROCEDIMIENTOS DE UN SISTEMA DE CONTROL

Debe estar compuesto por un conjunto de procedimientos, métodos y prácticas que incorporen a los siguientes elementos:

- Sistemas manuales integrados a sistemas computarizados.
- Formularios estandarizados
- Informes de avance estandarizados.
- Análisis de rendimientos.
- Flujos definidos de información.
- Registros actualizados de documentación del proyecto (planos, especificaciones, manuales, estándares, listados, etc.).
- Manual de Procedimientos.
- Organigrama a través del cual se definan funciones, responsabilidades y cargos.

CARACTERÍSTICAS DE UN SISTEMA DE CONTROL

Un sistema típico debe contener las siguientes características:

- Recoger información en forma periódica.
- Producir informes comprensivos.
- Informes coherentes pero con diferente nivel de detalle destinados a diferentes niveles de administración.
- La coherencia debe ser asegurada sobre la base de una fuente única de información.

La frecuencia con que se debe hacer verificaciones, depende de la naturaleza del trabajo y de la duración del proyecto. Para proyectos que durarán algunos meses una

frecuencia semanal o quincenal puede ser adecuada. Para los de más larga gestión, se recomendará un frecuencia quincenal o mensual.

TÁCTICAS DE CONTROL

El Control de lo que está ocurriendo en el proyecto debe hacerse mediante técnicas de fácil aplicación. Para que los resultados obtenidos a través del control sirvan como elemento de orientación de la administración, deben ser periódicos, oportunos y confiables.

Al respecto es recomendable seleccionar actividades entre las más representativas del proyecto. Estas pueden ser tan variadas como diferentes son los proyectos. Debe asociarse el avance de cada una de esas actividades a un indicador. Dicho indicador puede no ser exacto pero sí deberá ser fácil de conocer o apreciar.

El control periódico (diario, semanal) de estas actividades constituye una medida del trabajo hecho. El control debe ser hecho en periodos iguales con el objeto de que su representación sobre un mismo eje de abscisas tiempo pueda mostrar tendencias válidas. El error introducido al usar indicadores aproximados no resulta aditivo ya que en cada fecha de control se apreciará el trabajo realizado hasta entonces.

Si la información puntual es valiosa, la complementación de esta información mediante su representación gráfica respecto a una escala de tiempo muestra las "tendencias". El conocimiento de éstas es de mayor utilidad ya que la experiencia muestra que en la ejecución de todo proyecto existe una inercia con respecto a modificaciones de la productividad.

CONTROL DEL COSTO

Las principales acciones del control del costo son:

- Procurar que el proyecto se implemente con el menor costo posible.
- Procurar que el costo final del proyecto esté de acuerdo con los presupuestos.

El control de costos se divide en dos partes, el control de costos por el método convencional y el control de costos basado en la red.

CONTROL DE COSTO POR MEDIO DE MÉTODOS CONVENCIONALES

El objetivo principal del control del costo por medio de métodos convencionales es: Controlar los costos tanto durante las fases del diseño como de construcción. Deben informarse los costos reales con el objeto de que se puedan hacer comparaciones fácilmente. Debe presentar estimaciones y evaluar la evolución de los costos periódicos que se deben supervisar en contra de estimados con el fin de controlar efectivamente los costos del proyecto durante el diseño y la construcción.

MÉTODO

Para controlar los costos del diseño, es necesario programar cronológicamente las horas-hombre estimadas de diseño en cada etapa del mismo y llevar una hoja de

tiempo de las horas-hombre empleadas hasta la fecha y de las horas-hombre anticipadas para terminar las diversas secciones del trabajo.

El Ingeniero de costos controla los mismos en todo el proceso de diseño, asegurándose que el diseño final no sobrepase el estimado preliminar de ingeniería y este último no se sobrepase al estimado de planeación. Una vez conciliados los estimados de planeación y de ingeniería preliminares, el último se transforma en el instrumento de control de costos durante el diseño de detalle. Los criterios principales de un sistema de control de costos son : (1) debe proporcionar un marco de trabajo de referencia en contra del cual supervisar los costos comprometidos; (2) se debe suministrar a un centro de retroalimentación que advierta cuándo los gastos comprometidos se aproximan a los costos límite; (3) debe permitir establecer una acción de remedio para corregir los sobregastos en una etapa temprana antes de que sea demasiado tarde como para hacer algo acerca de ellos.

Durante la etapa de diseño, en donde es necesario controlar los costos comprometidos, la utilización de los estimados preliminares y detallados de ingeniería, codificados adecuadamente, es apropiada para lograr un marco de trabajo de referencia. Conforme se desarrolla el diseño de cada elemento, su costo se estima y se comprueba en contra del costo objetivo inicial del elemento. Este procedimiento proporciona la retroalimentación. Algunos elementos pueden necesitar solo un diseño mientras que otros necesitarán de varios diseños alternativos para permitir la selección de un elemento dentro del costo objetivo.

Durante el diseño se requiere una administración continua del costo no sólo para supervisar los costos, sino para ajustar el presupuesto y proporcionar informes económicos y de costos. Algunos temas a los que se dirigen tales informes son :

- 1.- Tendencias gubernamentales y locales. El conocimiento de las tendencias gubernamentales y locales industriales, comerciales, políticas, y sociales es esencial, ya que las decisiones pendientes en estas áreas pueden tener una influencia significativa sobre la factibilidad económica de un proyecto o pueden afectar en gran manera su desempeño.
2. Verdadero costo para el propietario. Aunque la estimación produce un costo predicho para el diseño de construcción, esta cifra no refleja el verdadero costo del proyecto para el propietario.
3. Factibilidad económica del proyecto. Una vez establecido el verdadero costo, es necesario evaluar el proyecto en sus términos económicos verdaderos o en términos económicos intangibles, tales como beneficios para la comunidad y sociales.
4. Disponibilidad de los recursos. Se necesita estudiar la disponibilidad de los recursos físicos para el proyecto con respecto a los fabricantes y proveedores para asegurarse de que se pueda ejecutar el proyecto propuesto.

Durante la fase de construcción es el constructor quien tiene la responsabilidad principal de controlar los costos, ya sea en su papel de administrador de la construcción, de administrador del proyecto, o contratista principal. El sistema de información puede diferir, pero la función principal del control de la construcción

consiste en controlar los costos de los recursos, mano de obra, materiales, instalaciones, y subcontratos. Para hacer esto, se aplican los mismos principios de control, esto es, un marco de trabajo para referencia, un sistema de retroalimentación, y un medio de tomar una acción de remedio cuando sea necesario.

La codificación de los costos es un elemento esencial en un sistema de control de costos. El fundamento del sistema de control de costos totalmente integrado está a nivel de recurso básico del registro del costo, y este ejercicio debe llevarse a cabo tan exactamente como sea posible en términos de tiempo y cantidad. Para lograr esto se pueden utilizar las siguientes formas:

1. Hoja de tiempos.
2. Registro de materiales.
3. Hoja de distribución del tiempo del equipo.

TERRANOVA CONSTRUCCION CO. LTD.											
HOJA DE TIEMPOS											
Nombre de la obra : <u>Hotel Plaza</u>						Fecha : _____					
Obra No. <u>134</u>						Informe No. <u>207</u>					
Condiciones meteorológicas : <u>Asoleado 5°C.</u>						Página No. <u>1</u>					
Número de mano de obra	Especialidad	Nombre	Concreto	03/00	03/00				Horas	Tarifa (\$)	Importe (\$)
4793	Carpintero	R. Jones								11.55	
4781	Carpintero	J. Kennedy								11.55	
4760	Albañil	B. Worth	7 1/2						7 1/2	9.10	68.25
4759	Carpintero	G. Moore	8						8	11.55	92.25
4792	Albañil	B. Maht	8						8	9.10	72.80
	Perifoneador de concreto	T. Yetman	8						8	11.30	90.40
4785	Carpintero	W. Wasmier								11.55	
4702	Carpintero	C. Martin	6						8	11.55	92.40
4933	Vanillero	B. Gillard								12.55	
4934	Vanillero	J. Cose								12.55	
	Operador de grúa	G. Tilley	8						8	11.55	
TOTAL											511.05
CANTIDAD			47 1/2						Capataz J. Wheton		

Fig. 1.4.5.1 Hoja de Tiempos.

TERRANOVA CONSTRUCTION CO. LTD
INFORME DE CONDICION DE MATERIALES

Contrato:
Hotel Plaza

Origen del Informe:
Diseño Arquitectónico, S.A.

No. de Contrato:
E 453

Periodo: 82 02 01 - 82 02 28
Fecha de Corte : 82 03 02
Fecha de Liberación : 82 03 05

Concepto Sumario : Material proporcionado por el propietario

82 02	LB16515	CONCRETO	m3	2,000	5,000	5,050	500
-------	---------	----------	----	-------	-------	-------	-----

Fig. 1.4.5.2. Registro de Materiales.

TERRANOVA CONSTRUCTION CO. LTD.										
HOJA DE DISTRIBUCION DEL TIEMPO DEL EQUIPO										
Mes 1983 Feb										
Equipo CRANE										
Equipo No. 61										
Tasa de uso por hora \$16.00										
Control de Obra										
Fecha										
	AA03100									
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
Dolares Hs.	128.00									

Fig. 1.4.5.3 Distribución del tiempo del equipo.

Luego se pueden recolectar y costear estos registros. Utilizando un código de costo adecuado, se asignan los gastos y se distribuyen a los componentes del proyecto. Basándose en el estimado de construcción que ha sido desarrollado debe prepararse un programa de trabajo en el que se listen todos los elementos que forman el proyecto con el objeto de obtener el presupuesto de construcción. Este proyecto cronológico del trabajo será semejante en su forma al programa cronológico del proyecto. Además el presupuesto se divide en costos de recursos con los cuáles se comparan los costos recolectados de las hojas de tiempos, registros de materiales, y hojas de instalaciones. Las actualizaciones regulares del porcentaje de obra programada terminada y el costo a la fecha de esta obra, determinada de los datos proporcionados en las tablas 1.4.5.1, 1.4.5.2, 1.4.5.3, se compara con el programa y el presupuesto original. La información actualizada se registra en una hoja de informe de costos. En esta hoja se resaltan lo excesos y los sobrantes en los costos. Esta última supervisión cumple con el tercer prerequisite de un buen sistema de control de costos, esto es, identificar las áreas que requieren se tome alguna acción para rectificar los excesos.

A nivel control de lugar las hojas de registro de recursos mostrarán:

- 1.- Cuantas horas-hombre se han gastado en una sección de la obra.
- 2.- Cuanto material se ha usado.
- 3.- Cuantas horas de planta se han usado.

Los registros de control extraídos de la estimación de construcción establecerán las horas-hombre presupuestadas, las cantidades de materiales, y las horas-instalación o planta anticipadas para terminar la obra. La utilización real de los recursos mostrará a

los supervisores del lugar en donde se necesita ejercitar un control más severo del costo, y de esta manera se estimula la administración. La utilización estimada de los recursos puede no ser totalmente exacta, y cuando los objetivos no se cumplen en una forma constante debe plantearse un programa de estudio de la obra para comprobar la viabilidad de los datos utilizados en la estimación.

Básicamente en la fase de diseño y en la de construcción se hace un estimado por el costo de las mismas, y luego se regulan los costos reales de la ejecución de cada una de estas fases tan estrechamente como sea posible con el estimado. Para comparar los costos reales con el estimado se lleva a cabo el control de rendimiento; y los dos métodos para determinar el rendimiento son:

- 1.- Costo real contra estimado para la obra correspondiente.
- 2.- Costo del pronóstico final contra el costo estimado.

COSTO REAL CONTRA ESTIMADO PARA OBRA CORRESPONDIENTE

Analizando el costo real a la fecha de la obra terminada a esa misma fecha en contra del costo estimado correspondiente, la gerencia determina la condición del proyecto. Para que este análisis sea exacto debe determinarse el costo real sobre una base acumulada; esto es, se considera que se incurrió en un costo tan pronto como la transacción tiene lugar, sin considerar si se hizo o no un pago. El Problema principal con este método consiste en determinar el costo estimado en contra del cuál se comparará el costo real de manera que ambos se relacionen con la misma cantidad de

trabajo. Un método para hacer esto consiste en estimar el porcentaje de obra física terminada y aplicar dicho porcentaje a la estimación para comparar la cifra resultante en contra del costo real.

También se hace referencia al costo estimado de la obra realizada como el "valor de obra". Este incluye todos los conceptos representados en el costo real. Si el costo abarca, además de los materiales instalados el costo de los materiales que se encuentren en el lugar o los compromisos de compra hechos, el valor de obra también debe incluir estos costos para que haya una confrontación significativa. La equiparación del costo real con el valor de obra da una comparación exacta de la condición del proyecto en esa fecha. Un método para determinar el valor de obra determinado requiere el pronóstico del costo final, en donde el cálculo del valor de obra es como sigue:

$$\text{valor} = \frac{\text{Estimación X Costo real de la obra realizada hasta la fecha}}{\text{Costo final pronosticado}}$$

Este método se aplica a todas las fases del proyecto. Sin embargo, se determina el valor de obra a partir del estimado de cada fase por separado.

Durante la construcción ocurren modificaciones. Por consiguiente el estimado se mantiene actualizado incluyendo el costo autorizado de todos los costos. Cuando las modificaciones no pueden cubrirse con recursos de la reserva para contingencias en el estimado, habrá de revisarse esta última. Esto se documenta en contra del estimado

original (esto es, el estimado cuando se fijó la magnitud de la obra). El estimado original se informa sin modificaciones durante toda la vida del proyecto, y se listan los cambios autorizados y el estimado revisado, con una división semejante al estimado original. De esta manera se toman en cuenta todas las revisiones.

A la diferencia entre el costo final pronosticado y el costo estimado, o el costo real a la fecha y el valor de obra, se le llama variación. Cualquier variación que se note debe analizarse y determinarse su causa de manera que pueda tomarse una acción correctiva apropiada. Una variación del costo final pronosticado obtenido del estimado puede ser causada por cambios en la magnitud de la obra, situaciones no previstas, y fluctuaciones en la tasa de inflación. Deben separarse estos tres tipos de variación. El primer tipo pertenece al estimado del costo base, el segundo a la reserva para contingencias, y el tercero a la reserva por incrementos.

El segundo criterio para medir el rendimiento es la comparación entre el costo final pronosticado y la estimación.

COSTO FINAL PRONOSTICADO CONTRA COSTO DE LA ESTIMACIÓN

El propósito del desarrollo de los pronósticos es que :

- 1.-** Ayudan a estimar y cuantificar costos futuros.
- 2.-** Identifican las dificultades potenciales.
- 3.-** Sugieren estrategias alternativas en cuanto a economía, tiempo y costo.

- 4.-Proporcionan indicadores para revisar la magnitud del proyecto, el diseño, las especificaciones, el programa y la estimación.

El costo final pronosticado es una predicción del costo del proyecto cuando este esté terminado. La viabilidad económica y financiera del propietario depende de este pronóstico. Es importante saber a tiempo que tanto se espera que cueste el proyecto. Difiere del costo de la estimación en que este último es un pronóstico de costo final antes de comenzar el proyecto, que solo se realiza para adaptar el costo a los cambios aprobados en la extensión de la obra. Sin embargo, el costo final pronosticado incluye el importe o el efecto que tendrán las fluctuaciones del mercado, los costos indirectos adicionales y los costos no previstos considerados necesarios para el proyecto basándose en la última información sobre el mismo. La selección de los componentes individuales debe reflejar sus valores representativos durante un período relativamente grande y debe ser altamente positivo. El costo final pronosticado cuando se compara en contra del costo total de la estimación da una evaluación del rendimiento del costo del proyecto a su terminación.

Los pronósticos son una herramienta vital para el ingeniero de costos que revelara sistemáticamente el costo del proyecto, permitiendo que se vuelva a realizar la planeación necesaria de las actividades del proyecto, y proporcionando una base para que se comprometa una parte significativa del costo material de un proyecto al advenimiento del mismo. Generalmente es menos costoso hacer alteraciones en la estrategia y financiamiento cuando el contratista conoce sus costos finales.

MÉTODO

Se usan tres métodos para pronosticar el costo final.

Un método es suponer excesos y sobrantes a la terminación que permanecerán siendo los mismos a una fecha dada de informe. En estas condiciones se supone que el rendimiento del equipo del proyecto durante el resto del trabajo estará cercano al presupuesto y de acuerdo con lo programado.

Un segundo método supone que el rendimiento pasado del equipo del proyecto continuará durante el resto del mismo. El pronóstico del costo final bajo esta suposición es sencillamente asunto de sumar el costo real a la fecha de los costos esperados para lo que resta de la obra y aplicar los costos unitarios de lo que se ha terminado hasta ese momento.

El tercer método usa información relativa al costo real a la fecha en que se incurrió en mano de obra, uso del equipo, materiales, subcontratos, etc. Se estiman los recursos requeridos para que se termine la obra a partir de los niveles de productividad más recientes, que consideran los efectos de la curva de aprendizaje y cuyo precio se fija a las tarifas o precios prevalecientes tomando en consideración las fluctuaciones del mercado.

CONTROL DE COSTOS BASADOS EN LA RED

Cuando se programa un proyecto mediante el método de redes, es posible cargar las actividades con datos de los recursos y costos integrada la programación y la ingeniería de costos. Una vez que los datos básicos están en su lugar, es posible con mucho menos esfuerzo producir informes de costo a partir de una entrada de porcentaje de terminación conforme se hace cada actualización a la red del proyecto. Para que funcionen adecuadamente los controles de costos basados en la red, debe aceptarse en toda la organización el concepto de planeación por redes.

Surgirán inevitablemente dificultades conforme la estimación de los recursos que normalmente forma el marco de trabajo del sistema de control de costos de una compañía, genera cantidades y costo de recursos por componentes convenientes para la estimación. Es difícil adaptar este marco de trabajo para que corresponda con las actividades de la red y que normalmente vienen dictadas por las secuencias tecnológicas en lugar de las secuencias de costo.

PRONÓSTICOS DE FLUJO DE EFECTIVO

Las cuentas de costos ayudan a la gerencia a determinar si el trabajo se realiza dentro del costo planeado. Sin embargo, el trabajo sólo puede desempeñarse si se tienen fondos disponibles para pagar durante toda la vida del proyecto. El flujo de los fondos dentro y fuera de una compañía generado por sus diversos proyectos, determinará el éxito o fracaso.

Es importante que siempre se este consciente de cuanto capital se necesita para asegurarse de que los fondos estarán disponibles oportunamente para satisfacer las necesidades del proyecto. Los pronósticos del flujo de efectivo requerirán de actualización constante, ya que las predicciones sobre el proyecto y las condiciones económicas cambian con frecuencia.

El pronóstico del flujo de efectivo proporciona información financiera firme de apoyo cuando se negocia servicios de crédito. También proporciona información acerca de extender o aceptar servicios de crédito de los deudores. Se puede ajustar la programación de cierta cantidad de contratos individuales de manera que, cuando se combinan, presenten colectivamente un flujo de efectivo adecuado para la compañía. Los flujos de efectivo de proyectos individuales se determinan para cada contrato por separado, y se aumentan los flujos adicionales de efectivo de la casa matriz (incluyendo todos los gastos generales no atribuibles al contrato) para hacer el pronóstico de flujo de efectivo total de la compañía.

Para proporcionar información sobre el estudio de factibilidad del proyecto, debe descontarse el flujo de efectivo del proyecto completo y usarse en las evaluaciones del mismo.

Es importante que siempre se este consciente de cuanto capital se necesita para asegurarse de que los fondos estarán disponibles oportunamente para satisfacer las necesidades del proyecto. Los pronósticos del flujo de efectivo requerirán de actualización constante, ya que las predicciones sobre el proyecto y las condiciones económicas cambian con frecuencia.

El pronóstico del flujo de efectivo proporciona información financiera firme de apoyo cuando se negocia servicios de crédito. También proporciona información acerca de extender o aceptar servicios de crédito de los deudores. Se puede ajustar la programación de cierta cantidad de contratos individuales de manera que, cuando se combinan, presenten colectivamente un flujo de efectivo adecuado para la compañía. Los flujos de efectivo de proyectos individuales se determinan para cada contrato por separado, y se aumentan los flujos adicionales de efectivo de la casa matriz (incluyendo todos los gastos generales no atribuibles al contrato) para hacer el pronóstico de flujo de efectivo total de la compañía.

Para proporcionar información sobre el estudio de factibilidad del proyecto, debe descontarse el flujo de efectivo del proyecto completo y usarse en las evaluaciones del mismo.

MÉTODO

Deben llevarse a cabo los pronósticos de flujo de efectivo para cada proyecto actualmente en proceso. Los pronósticos se toman desde el momento de la actualización del mismo hasta el final del proyecto.

Estos pronósticos del flujo de efectivo del proyecto forman la base de los flujos de efectivo de la compañía. El flujo de efectivo de la compañía consta de todos los movimientos de efectivo actuales sumados tales como renta, impuestos, teléfono, cargo de contratación, dividendos, y honorarios, todos los movimientos de efectivo del proyecto y de la compañía se suman para dar el pronóstico del flujo de efectivo total de la compañía.

Cuando se preparan los pronósticos del flujo de efectivo se calcula el valor de un proyecto durante la duración del contrato a partir del programa de la obra y se hace la gráfica para su interpretación.

Los costos del proyecto durante la duración del período del contrato se calculan a partir del contenido de obra indicado en el programa y se divide en costos periódicos de mano de obra, instalaciones, materiales, subcontratos, etc.

La tabla Tabla 1.4.6.3 muestra como se calcula el flujo de efectivo del proyecto a partir de un programa de obra. Obsérvese que se calcula el flujo de efectivo sobre una base mensual, tomando en cuenta el valor de toda la obra programada para que se termine durante ese mes. El flujo de efectivo acumulado también se registra y se hace la gráfica de la Figura 1.4.6.4.

Se grafican curvas para flujo de efectivo para cada uno de los proyectos semejantes, registrando el valor y el tiempo total expresado en porcentajes, y de esta manera estableciendo patrones a partir de los registros pasados. Suponiendo que se conoce el costo de cada operación de manera que puede establecerse el costo del proyecto, se calculan las utilidades como marcas porcentuales del costo. Por ejemplo, si se aumentarían uniformemente 12% todos los conceptos por utilidades y gastos generales, y el ingreso o entrada, total de efectivo se calculó a partir del plan del proyecto, el costo acumulado será de 0.89 multiplicado por la entrada total. Por tanto:

$$\text{Ingreso Total} = \text{Costo} + 12\% = 1.12 \text{ de Costo}$$

$$\text{Costo} = \text{Ingreso Total} / 1.12 = 0.89 \text{ de Ingreso Total}$$

En los proyectos en donde el margen de utilidad no se distribuye uniformemente, tales como proyectos con una elevada movilización o desmovilización de los costos o proyectos en donde los márgenes de utilidad se han ponderado por diversas razones, entonces habrá que calcularse y tabularse el costo del proyecto a partir del programa de obra. Siguiendo este procedimiento se registran los costos conforme se vencen para su pago. Los contratos por mano de obra, como instalaciones, materiales y subcontratistas se negociarán inevitablemente bajo condiciones diferentes, y la demora que hay entre el incurrir en los costos y hacer los pagos necesarios variarán entre ellos y por tanto se registrarán por separado. Para los pronósticos del flujo de efectivo a largo plazo en donde se necesita que se conozcan los compromisos grandes es

adecuado ponderar el período de pago demorado en contra de la contribución que cada pago hace a los gastos globales.

Actividad	Tiempo (meses)											
	1	2	3	4	5	6	7	8	9	10	11	12
Excevación	10	10										
Cimientos de concreto		5	20									
Losas sobre nivel			5	5								
Columnas al nivel del primer piso				20	5							
Losa del piso 1					20	5						
Columnas al nivel del piso 2						20	5					
Losa en el piso 2							20	5				
Columnas a nivel del techo								10	10			
Losa del techo y construcción del mismo									10	5		
Revestimiento de acero inoxidable									35	10	5	1
Servicios										20	5	1
Acabados											3	5
Flujo de efectivo mensual	10	15	25	25	25	25	25	50	50	15	5	5
Flujo de efectivo acumulado	10	25	50	75	100	125	150	200	250	265	270	275

Fig. 1.4.6.3. Flujo de efectivo calculado a partir de un programa de obra, en miles de dólares.

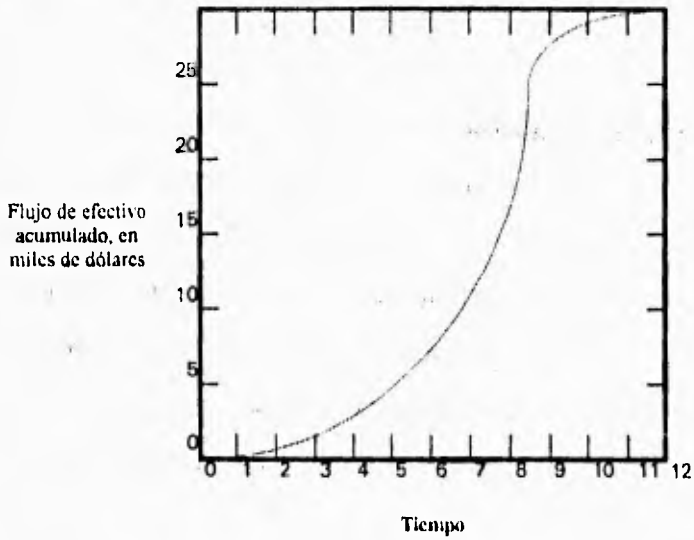


Fig. 1.4.6.4 Flujo de efectivo contra tiempo a partir de un diagrama de barras.

1.4.6 EXPLOSIÓN DE INSUMOS

La explosión de insumos es un reporte que permite conocer la cantidad utilizada de cada insumo en todo el presupuesto, así como el porcentaje de participación en el costo total de este.

Esta actividad se realiza dentro del diseño para estudiar la disponibilidad de los recursos físicos para el proyecto con respecto a los fabricantes y proveedores para asegurarse de que se pueda ejecutar el proyecto propuesto.

Dentro de la etapa de la construcción se lleva a cabo para realizar el plan de compras.

ADMINISTRACIÓN DE MATERIALES

Todo proyecto requiere disponer en mayor ó menor grado de materiales.

Deberá verificarse que los materiales comprometidos:

- Tengan la calidad definida en el diseño.
- Estén disponibles al pie de la obra en la fecha programada.
- Su costo no exceda al considerado en el presupuesto.

Para asegurar el cumplimiento de los requerimientos anteriores es necesario implantar una Administración de los Materiales. Esta abarca:

- La gestión de COMPRA.
- El control de existencias.

- El control de pérdidas y la prevención de faltas.
- El control presupuestario.

Los sistemas que se implanten deberán procurar:

- eliminar las faltas de material al pie de la obra que puedan provocar retrasos en los programas.
- manejar las compras en forma de producir favorables en cantidades, plazo de entrega y costos.
- trabajar con niveles de stock adecuados a las necesidades, evitando materiales en exceso que no son utilizados.
- procurar a través de la estandarización, que el número de items sea el menor posible.

LA GESTIÓN DE COMPRAS

La gestión de compras es fundamental para que el proyecto se realice dentro de los plazos y costos establecidos por su programa y presupuesto oficial.

El personal a cargo de las compras debe ser especializado. Además de tener un conocimiento técnico general de los equipos y materiales a adquirir, debe estar familiarizado con los mercados que tienen la capacidad de producirlos; y dentro de ellos, conocer nombres de fabricantes calificados para dar cumplimiento a una eventual orden de compra.

De acuerdo a la programación, especificaciones técnicas del proyecto y fechas en que se necesitan los materiales se elabora un requerimiento de compra.

El requerimiento de compra será verificado contra el presupuesto. En caso que la partida no haya sido considerada, o lo haya sido inadecuadamente, deberá tomarse una decisión. O se revisa la necesidad o debe modificarse el presupuesto.

Si existe conformidad entre el requerimiento y el presupuesto, la compra queda en condiciones de ser autorizada, en este momento se realiza la orden de compra.

Para órdenes importantes, se recomienda que el personal de compras se preocupe en forma adicional de:

- verificar que la orden de compra fue recibida por el proveedor.
- pedir el número que el proveedor asigna al pedido.
- establecer contacto con el proveedor para canalizar todas las comunicaciones y gestiones durante la preparación y despacho.
- establecer con el proveedor las instancias en que se recibirá la certificación.
- requerir del proveedor la programación interna que utilizará en la ejecución y preparación del pedido.
- establecer los programas de inspección.
- revisar periódicamente el avance efectivo alcanzado por la preparación del pedido.
- establecer rutinas que permitan informar sobre el avance del pedido.
- incluir información del estado de la gestión dentro de cuadros de control del proyecto.

Una vez emitida la orden de compra entran en actividad las funciones de inspección y de tráfico. Su acción se prolonga durante el período de fabricación o de preparación por parte del proveedor.

La recepción puede verificarse en la bodega central o en la bodega donde se realiza el trabajo. En el primer caso, existirá una gestión de posterior traspaso de los materiales desde la bodega central a la de trabajo.

Al momento de la Recepción, la bodega verifica el elemento físico contra la documentación de compra. La conformidad es certificada, con el objeto de dar curso al pago que se hará a través del Departamento Administrativo.

FUNCIÓN DEL GRUPO DE INSPECTORES

Las funciones de los inspectores serán las de verificar en los lugares de origen el cumplimiento del pedido con las condiciones definidas en la orden de compra. Esta revisión se basa en las especificaciones de los materiales, los procedimientos de fabricación, la calificación de la mano de obra, la realización de recepciones intermedias, el control de prearmados, y la fiscalización de embalajes y despacho de fábrica. Junto con el control de la calidad, se deberá ejercer el del cumplimiento del programa interno del proveedor. El objeto es conocer la posibilidad real que tiene el proveedor para cumplir con los plazos y condiciones establecidos por la orden de compra.

FUNCIÓN DEL GRUPO DE TRÁFICO

La función del Grupo de tráfico comenzará asesorando al grupo de compras con respecto a la selección del medio de transporte más adecuado desde el lugar de origen hasta el de trabajo. De esta selección, resultarán condiciones que serán incorporadas en la orden de compra.

Se compararán las estimaciones sobre costos de flete y transporte resultantes con respecto a las provisiones contempladas en el presupuesto oficial. En caso necesario se introducirán las correcciones correspondientes.

En seguida deberá coordinar a través del Grupo de Inspección el cumplimiento de los requerimientos de Codificación, Embalaje y Manipulación de los Equipos, Estructuras y/O Materiales.

Se coordinarán los programas de embarque con la reserva de los espacios de carga necesarios. La función es especialmente delicada cuando se ha planificado el despacho conjunto de materiales de varios proveedores diferentes. Esta situación se ha hecho más común con la generalización del uso de contenedores (containers).

Los cargas de gran peso (heavy lifts) o de gran volumen, necesitarán de una programación individual.

En caso que parte de los trayectos sean por tierra, deberán revisarse las limitaciones de tamaño (gálibos de túneles y puentes, radio de curvas) y carga por eje (carreteras y puentes). El traslado deberá coordinarse con autoridades de vialidad. El contratista de transporte será especializado, debiendo disponer de equipos adecuados para el manejo de carga excepcional. Normalmente es necesario el pago de derechos de vía

especiales. Eventualmente en el curso de estas operaciones hay que reforzar estructuras de puentes, construir desvíos (by-pass), obtener la suspensión ó proteger servicios aéreos (cables) que atraviesan por sobre la ruta.

El otro aspecto que debe ser manejado por el grupo de tráfico es el relacionado con la documentación necesaria para obtener la internación de la mercadería importada. Esta documentación debe ser obtenida y tramitada en lo posible con anterioridad a la llegada de los envíos, con el objeto de lograr un rápido desaduanamiento.

Junto a la revisión del contenido vs. documentación se debe establecer si ha habido daños ó mermas durante el transporte, que dan base a reclamaciones ante la compañía de seguros.

Se completa el trámite de llegada del equipo (desde el punto de vista físico como administrativo) cuando es recibido por la bodega del trabajo y la totalidad de la documentación es devuelta al grupo de compras. En base a ella podrá dar cumplimiento a la devolución de garantías y seguros, documentación de las recepciones y pago de facturas.

Como función adicional y en conjunto con el departamento de compras, el departamento de tráfico deberá asesorar al administrador del proyecto sobre la aplicación de los regímenes tributarios en lo que respecta al costo de los productos y bienes incorporados en las obras, sean de origen nacional o provengan de importación.

RECEPCIÓN, MANEJO Y CONTROL DEL MATERIAL

Al momento de la recepción deberá compararse la orden de compra con la factura y la guía de recepción. El material es inspeccionado, registrándose cualquier observación. En caso de generarse reclamos, deben cursarse, informando al departamento de compras y al de programación. En materiales de importación serán adicionales los trámites de introducción y reclamo al seguro en caso de mermas ó averías.

La salida del material desde la bodega a su vez requiere de un procedimiento de control. Este debe asegurar que:

- el material sea solicitado por una persona autorizada.
- el material solicitado corresponda por codificación al adquirido para dicho trabajo.
- el material entregado sea el solicitado.
- el material de salida sea descontado del cardex de existencia.

El control de pérdidas ó la detección de faltas es importante, en especial en el caso de materiales que se manejan en gran cantidad y que consecuentemente han sido depositados en lugares de menor control.

Existe una tendencia de parte de los supervisores a retirar más material que el necesario, con objeto de prevenir eventuales faltas. Esto adquiere importancia especial cuando los horarios de bodega no cubren la totalidad de los de trabajo (por ej. 1 turno de bodega y dos de trabajo, ó 1 de trabajo + sobretiempo). Los excedentes no serán

devueltos a bodega y como consecuencia se producirán faltas para abastecer otros pedidos.

Además del control administrativo en base a cardex, deberá establecerse uno de tipo físico. La bodega debe ser cerrada. Los patios bodega deben estar cercados. Deben tener iluminación perimetral suficiente, complementada con espacios que permitan detectar a distancia el movimiento de cargas que pueda producirse en horarios distintos a los de trabajo.

Los estacionamientos deben estar lejos de las bodegas, especialmente si éstas son semiabiertas. Debe limitarse y restringirse el acceso más allá de los estacionamientos a vehículos en los cuales puedan ocultarse cargas. Todo el personal que tenga contacto con materiales o con control de materiales debe estar avalado por antecedentes intachables de trabajos anteriores en una función semejante.

Es imprescindible hacer de tiempo en tiempo un inventario físico. Esto es una operación laboriosa que involucra cuantificar elementos que no son fácilmente contables. Requiere además, con el objeto de no entorpecer la entrega de materiales, ser realizado en jornadas en que el resto del trabajo esté paralizado. Pese a ello debe practicarse y es recomendable que todos los involucrados sepan que están sujetos a un control imprevisto de inventario físico.

Todo lo recomendado no evitará que se produzcan discrepancias entre el material recibido, el material incorporado a la obra y el excedente en bodega. Pero un cuidado permanente permitirá que estas discrepancias sean mantenidas dentro de límites tolerables.

CONTABILIDAD DE COSTOS

La contabilidad de costos del proyecto es una función importante de la ingeniería. El registro de los costos se hace de tal manera que pueda desarrollarse el control administrativo y determina el éxito de una organización. El costo en la construcción se clasifica ampliamente según los siguientes encabezados:

- 1.- Costos de material.
- 2.- Costos de mano de obra.
- 3.-Costo de las instalaciones.
- 4.-Costos de gastos generales y utilidades.

Cada proyecto deberá tener un conjunto independiente de cuentas en las que se registren los costos atribuibles a ese proyecto. Para una indicación verdadera del costo, se atribuirán al proyecto tantos costos indirectos como directos.

Los objetivos de llevar una contabilidad exacta del proyecto son:

- 1.-Planear, administrar y controlar el proyecto inmediato de existencia.
- 2.-Planear, administrar y controlar el progreso futuro proporcionando la retroalimentación esencial sobre los sucesos actuales.
- 3.-Proporcionar la retroalimentación exacta de los costos del proyecto actual que ayudará a hacer estimaciones futuras para otros proyectos.

MÉTODOS

Para planear, administrar y controlar el proyecto inmediato en existencia, se obtienen registros detallados de las horas-hombre, costos del material, y costo de las instalaciones a partir de las hojas de tiempo, hojas de registro de material, y hojas de instalación-horas. Los costos se distribuyen entre las diversas cuentas de costos por medio de una codificación para cada concepto de obra. Luego estas cuentas dan el costo real a la fecha de cada concepto de obra. Luego se mide la cantidad de obra y el costo planeado determinado por las tarifas estimadas originales de la mano de obra, el material, y las instalaciones. La comparación de este costo planeado en contra del costo real es el principio del control de costos. Cuando existen diferencias, se necesita una investigación administrativa para determinar si las ineficiencias causan excesos en el costo o se debe a la inexactitud de los registros originales. La exactitud de la contabilidad de costos reside en el correcto registro de los datos.

El análisis de los datos contables del costo proporciona datos históricos tan importantes como el costo de un tiempo no productivo, los costos promedio de costo extra, los costos del tiempo ocioso de las instalaciones, el costo de los desperdicios o desechos de materiales y finalmente la productividad horas-hombres, que es el concepto más difícil de estimar en la construcción. Esta información se extrae de las conciliaciones entre los costos, el tiempo y la cantidad de obra hecha, que es lo que finalmente determina el valor de un proyecto.

Capítulo I Conceptos Básicos en la Industria de la Construcción

La contabilidad de costos, registra los costos reales experimentados o tenidos, en cuanto a mano de obra, instalaciones, y materiales asignados a ciertas cantidades de obra. La acumulación de esta tasa unitaria de información proporciona un banco importante de datos de costo histórico para referencia futura al compilar precios unitarios para la estimación. La información y registro de la contabilidad de costos se lleva a cabo en formularios característicos que cumplen con el sistema típico de contabilidad adoptado por la organización. Las siguientes tablas muestran algunas de las formas utilizadas en los registros de la contabilidad de costos.

(1) Número del trabajo o tarea	(2) Código del costo de trabajo o tarea	(3) Horas	(4) Tarifa (\$)	(5) Clase
3004	1008	4	12.00	Reg.
3004	1107	4	12.00	Reg.
3004	9089	1	12.00	TT(a)
3004	1107	1	18.00	OT(b)

Los números que están entre paréntesis corresponden a lo siguiente: (1) Identifica el trabajo por número; (2) Identifica el concepto de obra y otros costos tales como el tiempo de viaje; (3) Identifica la distribución de las horas; (4) Indica las tasas por hora de los salarios; (5) Indica las tasas de tiempo tales como (a) tiempo de viaje, (b) tiempo extra, etc., que se muestran en (3) y (4).

Fig. 1.4.6.1 Tarjeta diaria de tiempo.

Nombre/Número del trabajo o tarea		Fecha								(4)	(5)
(1)	(2)	(3)								(4)	(5)
Empleados (nombre/número)	Tarifa (\$)	1	2	3	4	5	6	7	8	TE (horas)	Horas totales

Los números entre paréntesis corresponden a lo siguiente: (1) Nombre y número de los empleados; (2) Puede ser registrada por el personal de guardia o el empleado; (3) Cada hora (hasta 8 horas) se distribuye entre los conceptos de obra registrando números código en las columnas; (4) Aquí se registra el tiempo extra; (5) Se registran las horas totales registradas en el momento de marcar la tarjeta.

Fig. 1.4.6.2 Informe diario de distribución de mano de obra.

Para los costos de materiales, equipos, y de gastos generales, se utilizan formularios semejantes para registrar todos ellos en las cuentas del proyecto.



CAPITULO II

PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCION

2.1 ANÁLISIS DE REQUERIMIENTOS

Las constructoras en cada ocasión que tienen que elaborar un presupuesto es necesario que utilicen un programa de computadora, ya que las tareas realizadas durante este proceso es arduo y generalmente representa una gran inversión de tiempo. Sin lugar a dudas los programas de Precios Unitarios pueden representar una gran ventaja para la elaboración del presupuesto, sin embargo que tan competitiva puede ser una constructora si el programa de cómputo no reúne todas las características que el usuario necesita para elaborar sus Precios Unitarios. Hoy en día los analistas de costos han dejado atrás las reglas de cálculo, ya que se pueden utilizar desde programas de hojas de cálculo hasta programas especializados para lograr presupuestos óptimos, sin embargo de la elección de la herramienta depende el tiempo que se invertirá para la elaboración del presupuesto.

Aún cuando se cuenta con programas de computadora los constructores se ven ante una tarea ardua cada vez que necesitan elaborar un presupuesto, los pasos para la elaboración de los presupuestos pueden variar, a pesar de que existe un método homogéneo para ello, sin embargo muchos constructores al elaborar sus presupuestos hacen uso de su experiencia para obtener los mejores rendimientos lo que se reflejará en ganar concursos de obra, y posteriormente obtener las mejores utilidades posibles.

Las facilidades que el programa de Precios Unitarios ofrezca al constructor debe tener como meta el aumento de su productividad, como lograr esto con un manejo ágil y sencillo que le permita aplicar su experiencia de una forma casi similar a un proceso

manual. El programa de Precios Unitarios sin lugar a dudas es el equivalente a un programa CASE para computación, permite el análisis, diseño y ejecución de una obra, siguiendo con la analogía podemos decir que entre mejor sea la herramienta de CASE mejor será el sistema que logremos implementar, este es el objetivo de un programa de Precios Unitarios, entre mejor analizado sea la obra mejores rendimientos y utilidades serán obtenidas, lo más importante que el programa de Precios Unitarios permita al constructor aplicar el método que el considere más apropiado para elaborar su presupuesto, las herramientas CASE se basan en modelos y métodos establecidos, del usuario depende cual será el modelo a utilizar.

Podemos identificar las siguientes aspectos que deben ser cubiertos por un programa de Precios Unitarios:

- **Facilidad de Uso del programa de Precios Unitarios, buscando con esto un rápido aprendizaje por parte del usuario**
- **Qué se apegue a los estándares de la industria en cuanto a programas de cómputo, es decir aplicando una interface al usuario sencilla y similar a programas populares como hojas de cálculo y procesadores de texto.**
- **Permitir que el usuario pueda aplicar su experiencia en la elaboración de presupuestos de una forma sencilla, mejor aún extender sus conocimientos a partir del programa.**
- **Rapidez para elaborar los precios unitarios en el programa**
- **Que el programa cuente con herramientas de análisis que permitan tomar decisiones adecuadas y rápidas al momento de elaborar los precios unitarios**

- Que el programa cuente con reportes de calidad en su presentación y que además refleje los resultados de los precios unitarios adecuadamente.

Existe un problema básico que se tiene que resolver durante esta etapa de análisis, el programa de Precios Unitarios a diseñar debe ser lo suficientemente general para que pueda resolver las necesidades de los constructores, sin apegarse a un modelo en específico, ya que esto puede traducirse en problemas con el usuario final ya que puede dificultar el manejo del programa mismo, si se establece un modelo ágil en el cual el usuario se sienta confortable y seguro, donde él pueda aplicar su experiencia permitirá cumplir en principio uno de sus objetivos, productividad.

El programa de precios unitarios de acuerdo a los programas existentes en el mercado debe contar con los siguientes módulos:

- Edición y reportes de Insumos (Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos)
- Edición y reportes de Análisis (Básicos, Integrados, Partidas, Capítulos y Extraordinarios)
- Módulo de explosión de insumos
- Módulo para el cálculo del factor de salario real
- Manejo de múltiples monedas
- Control de Obra
- Ruta crítica para el control de proyectos

Estas son las necesidades básicas, sin embargo es necesario contemplar lo siguiente:

Para la elaboración de los precios unitarios se utilizan los análisis que son un aglutinador de insumos. Si los precios de los insumos varían esto producirá que los análisis se vean afectados en sus precios. Esto lo sabe el constructor, por lo que internamente el programa deberá contar con un esquema de actualización que permita reflejar los cambios que existan entre los elementos relacionados. El nuevo módulo de actualización deberá ser lo suficientemente rápido para que el usuario no se sienta incomodo con su uso.

Una constructora como cualquier otra compañía lleva una relación de la información sobre sus clientes. Y más aún se tiene que llevar información interna que permita controlar la obra, de lo anterior surge una nueva opción con la que debe contar el programa de Precios Unitarios, es necesario un módulo de Información de la Obra.

Otro problema que se tiene en la elaboración de un presupuesto por parte de un constructor es la consulta de datos de obras anteriores, lo que ellos denominan experiencia, es decir los muros se construyen siempre de cierta forma, se utiliza en gran medida el mismo número de obra de mano, se aplican las mismas cantidades de herramientas y materiales, además de los mismos rendimientos. Cuan importante es entonces el acceso a datos, que procesados en el programa de Precios Unitarios se convierten en información para la toma de decisiones por parte del usuario. E

programa de Precios Unitarios como consecuencia debe contar con módulo de consulta de obras y presupuestos anteriores.

Bien es tiempo de reflejar los procedimientos que el usuario lleva a cabo al momento de elaborar un presupuesto:

Método tradicional

- Se dan de alta los datos principales de una obra
- Se hace la captura de datos de Insumos
- Se Analizan las matrices de precios unitarios utilizando los datos de los insumos, a su vez una matriz puede contener a otra del mismo o de un nivel inferior. Los niveles utilizados en la mayoría de los casos son Básicos, Integrados, Partidas y Capítulos, que representan en ese orden los niveles que puede utilizar un usuario constructor para aglutinar sus insumos.
- Se hace una revisión de la información para que los datos chequeen
- Se aplican costos indirectos permitidos por la ley de Obra Pública
- Se elaboran los reportes de Insumos, de matrices y del Presupuesto obtenido
- Se presenta el presupuesto al cliente que ha licitado a concurso una obra

Si el usuario ha aceptado la participación de la constructora en la obra, la constructora deberá llevar un control de obra como estimaciones de avance, así como la explosión de insumos, que se traduce en la cantidad de insumos utilizadas en cierta fase de la obra de acuerdo a lo estimado, estos datos son utilizados por los constructores para cobro a los clientes basados en el avance en la obra. Las estimaciones se encuentran basadas en las matrices de precios unitarios elaboradas en el proceso de presupuestación.

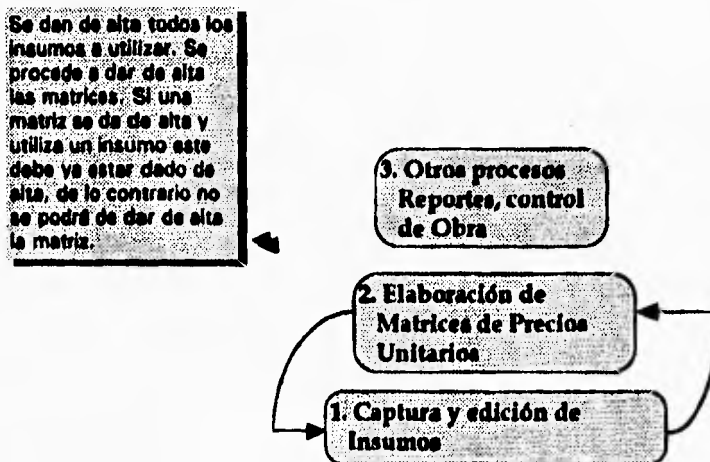


Figura 2.1.1 Método tradicional para elaboración de Presupuestos

MÉTODO RÁPIDO

- Se Analizan las matrices de Precios Unitarios utilizando los datos de Insumos que existan en ese momento, si el insumo o matriz que se desea agregar al precio unitario no existe se da de alta en ese momento, permitiendo agilidad al momento de elaboración de los precios
- Se hace una revisión de la información para que los datos chequeen
- Se aplican costos indirectos permitidos por la ley de Obra Pública
- Se elaboran los reportes de Insumos, de matrices y del Presupuesto obtenido

- Se presenta el presupuesto al cliente que ha licitado a concurso una obra
- Si el usuario ha aceptado la participación de la constructora en la obra, la constructora deberá llevar un control de obra como estimaciones de avance, así como la explosión de insumos, que se traduce en la cantidad de insumos utilizadas en cierta fase de la obra de acuerdo a lo estimado. Las estimaciones se encuentran basadas en las matrices de precios unitarios elaboradas en el proceso de presupuestación.

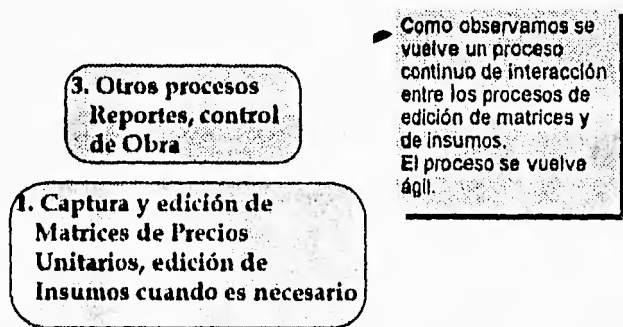


Figura 2.1.2 Método Rápido para elaboración de Presupuestos

Analizando los dos métodos descritos anteriormente el usuario se ve beneficiado por el segundo método ya que como su nombre lo dice es más rápido y eficiente, esto sin lugar a dudas representa un problema más que el programa de Precios Unitarios debe resolver para que el usuario no se vea limitado. Sin embargo no se puede cortar el módulo de edición de Insumos, ya que sigue siendo necesario, además no todos los

usuarios se sienten confortables con el método rápido. El programa entonces deberá contar con las ventajas de los dos métodos.

Surge también un nuevo elemento la aplicación de costos indirectos permiten al constructor obtener márgenes de ganancia en la ejecución de obras, la ley de obra pública exige al constructor que se indiquen cuales son sus utilidades y factores aplicables por la ley federal de trabajo, principalmente impuestos. Si bien el problema más grande a resolver es que las leyes de obra pública y las prestaciones cambian constantemente, por lo cuál el programa de Precios Unitarios deberá ofrecer un método simple para el manejo de estos factores, ofreciendo sencillez y un ajuste rápido a nuevas leyes.

De lo anterior surge la especificación de los objetos que deberán ser contemplados en la fase de diseño:

Objeto Insumos

- Objeto Materiales**
- Objeto Mano de Obra de Mano**
- Objeto Maquinaria y Equipo**
- Objeto Subcontratos**

Objeto Análisis

- Objeto Básicos**
- Objeto Integrados**
- Objeto Partidas**

Objeto Capítulos

Objeto Extraordinarios

Objeto de Actualización de Precios

Objeto actualización de Precios

Objeto Reportes

Objeto Factores de Sobre Costo

Objeto Explosión de datos

Objeto Estimaciones

Objeto Ruta Crítica

2.2. REQUERIMIENTOS DEL USUARIO.

El análisis de sistemas es una actividad que engloba la mayoría de las tareas que integran la ingeniería de sistemas en una computadora. Se centra en todos los elementos del sistema (hardware y software) y se realiza de acuerdo con los siguientes objetivos:

1. Identificación de las necesidades.
2. Estudio de viabilidad.
3. Realizar un estudio técnico y económico.
4. Asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema.
5. Establecer restricciones de costo y de tiempo.
6. Creación de una definición del sistema que sea la base para todo el trabajo de ingeniería subsecuente.

En el análisis de sistemas, la identificación de las necesidades es el punto de partida para el desarrollo, este proceso se lleva a cabo cuando el analista se entrevista con el cliente, definiendo los objetivos del sistema: por ejemplo, la información que se va a producir y suministrar, las funciones y el rendimiento requerido. El analista se asegura de distinguir entre lo que necesita el cliente (elementos críticos para la realización) y lo que el cliente quiere (elementos deseables pero no esenciales).

El estudio de viabilidad es necesario ya que con él se puede evaluar la factibilidad de un proyecto lo antes posible. En el estudio de viabilidad existen cuatro áreas de interés:

- **Viabilidad Económica :** Evaluación del costo de desarrollo frente al beneficio final producido por el sistema desarrollado.
- **Viabilidad Técnica :** Estudio de funcionalidad, rendimiento y restricciones que pueden afectar la posibilidad de realizar un sistema aceptable.
- **Viabilidad Legal :** Determinación de cualquier infracción, violación o ilegalidad que pudiera resultar de desarrollo del sistema.
- **Alternativas :** Evaluación de posibles alternativas al desarrollo del sistema.

Para realizar correctamente el desarrollo de software, es necesario realizar una especificación completa de los requerimientos del mismo.

El análisis de requerimientos es el primer paso técnico en el proceso de la Ingeniería de software, y es cuando se establece de forma general el ámbito del programa y este se da en una especificación concreta que se convierte en la base de la fase de desarrollo.

Para llegar a un óptimo establecimiento, el analista y el cliente tienen un papel activo en la especificación de requerimientos ya que existe una gran comunicación entre ellos y en ocasiones resulta difícil porque frecuentemente se presentan cambios por mala interpretación o falta de información.



Fig. 2.2.1 Debe existir comunicación entre el analista y el cliente para realizar la especificación de los requerimientos.

El análisis de requerimientos plantea la asignación de software a nivel sistema y el diseño de programas, además facilita al Ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interface con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa.

El análisis de requerimientos permite al Ingeniero refinar la asignación de software y representar el dominio de la información que será tratada por el programa. Da también al diseñador la "representación" de la información y las funciones que pueden ser traducidas en datos, arquitectura y procedimientos de diseño. Finalmente suministra al técnico y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido.

El análisis de requerimientos puede dividirse en cuatro áreas:

1. Reconocimiento del problema: Inicialmente, el analista estudia la especificación del sistema y el plan del proyecto. Es importante conocer el

contexto del sistema y el ámbito de los programas que se utilizaron en la generación de las estimaciones de la planificación. Posteriormente, se debe establecer la comunicación para el análisis, de forma que asegure el reconocimiento del problema. Las formas de comunicación requeridas para el análisis se muestra en la siguiente figura :

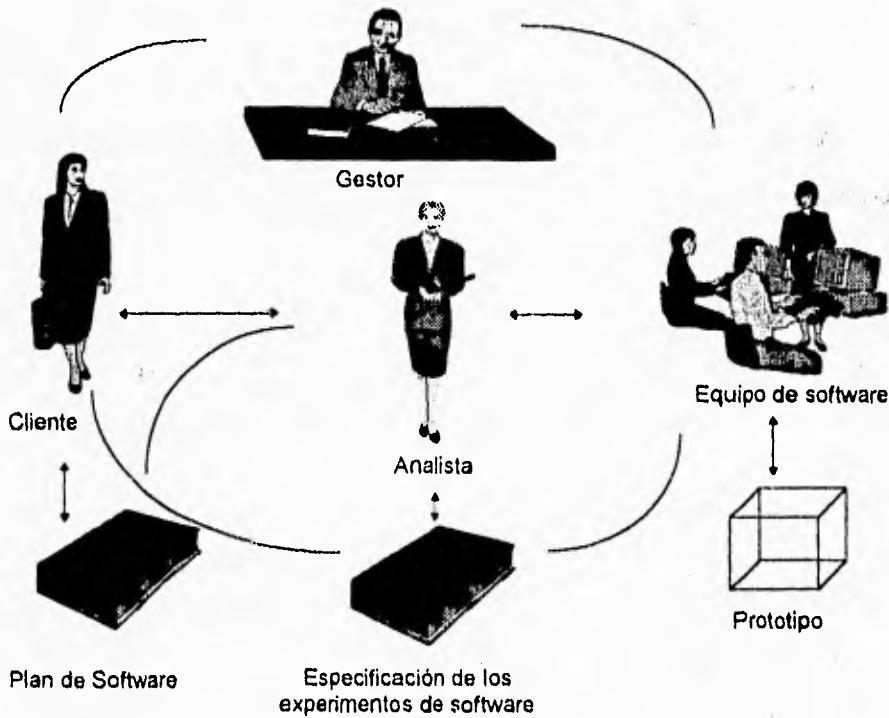


Fig. 2.2.2 Proceso de comunicación en el análisis.

2. **Evaluación y síntesis:** La evaluación del problema y la síntesis de la solución es la siguiente área principal de trabajo del análisis. En esta etapa, el analista debe evaluar el flujo y la estructura de la información, refinar minuciosamente las funciones del programa a detalle, establecer las características de la interface del sistema.
3. **Especificación:** Las tareas asociadas con el análisis y especificación existen para dar una representación del programa que pueda ser revisada y aprobada por el cliente. Una vez que se hayan descrito las funciones básicas, comportamiento, interface, información, se especifican los criterios de validación para demostrar una comprensión de la implementación de los programas. Estos criterios sirven como base para hacer la fase de pruebas durante el desarrollo de los programas.
4. **Revisión:** La revisión de los requerimientos sirve como base para una verificación conducida por el cliente y el técnico. Casi siempre produce modificaciones en la función, comportamiento, representación o criterios de validación.

Entre los problemas que pueden encontrarse durante el análisis de los requerimientos están las necesidades asociadas con la adquisición de la información pertinente, el manejo de la complejidad del problema y el acomodar los cambios que puedan ocurrir durante y después del análisis.

Los problemas que surgen durante el análisis de requerimientos son los siguientes:

1. Pobre comunicación que hace difícil la adquisición de la información..
2. Técnicas y herramientas inadecuadas que producen especificaciones imprecisas.
3. Tendencia a acortar el análisis de requerimientos, conduciendo a un análisis inestable.

Las técnicas de análisis de requerimientos combinan procedimientos sistemáticos con una notación única para analizar los dominios de información funcional de un problema de software.

En este módulo de requerimientos del usuario se plantean e identifican las necesidades que el usuario tiene y las cuales se pretende satisfacer mediante el desarrollo del presente sistema. A continuación se presentan algunos de los principales requerimientos para el sistema de precios unitarios, cuyo cumplimiento facilitará el manejo del sistema y provocará la aceptación entre los usuarios.

A) INTERFACE

- La interface hacia el usuario debe ser funcional, elegante y debe evitar la confusión. Para producir una interface funcional y visualmente confortable, se necesita entender los requerimientos del negocio, de la aplicación y una buena percepción de la estética.
- La aplicación debe ser fácil de leer, observar y de usar en comportamiento y organización.

- La aplicación debe ser fácil de leer, observar y de usar en comportamiento y organización.
- Accesar a cualquier opción del sistema desde cualquier ventana.
- Compartir los datos generados en el sistema con otras aplicaciones de Windows.
- Tener la facilidad de enviar los datos de la obra a : dBase, Excell, Access.
- Tener comunicación directa con Project de Microsoft para el cálculo de Ruta Crítica.
- Debe presentar la característica de ser ágil en la captura de información.
- Presentar catálogos y poder seleccionar la opción deseada sin tener que teclearla.
- Permitir copiar la información con solo seleccionar los datos, arrastrarlos y soltarlos sobre el destino.



Fig. 2.2.3 La aplicación debe ser fácil de leer, observar y usar.

B) OPERACIÓN DEL SISTEMA

REQUERIMIENTOS GENERALES

- El manejo del sistema debe ser rápido de aprender
- Manejar un número ilimitado de obras.
- Debe ser de fácil instalación, de manera que las personas que no se encuentren muy familiarizadas con el manejo de la computadora puedan instalar y utilizar el sistema con facilidad.
- Debe tener la facilidad de trabajar en forma individual o con varios usuarios dentro de la misma obra para explotar las facilidades de ambiente multiusuario.
- Tener la posibilidad de mantener varias obras abiertas en sus diferentes opciones al mismo tiempo (dependiendo de la memoria con la que cuente el equipo).
- Contar con un panel de control, desde el cual se pueda editar información diferente relacionada con la obra.
- Hacer más rápido el procedimiento de definición de una obra.
- Generar un catálogo maestro de precios.
- Manejar materiales por porcentajes de flete y descuento.
- Tener mano de obra con el factor de salario real integrado.
- Tener la maquinaria y equipo con cálculo de costo-hora máquina integrado.

- Realizar análisis o matrices de precios a cinco niveles:
 1. Básicos.
 2. Integrados.
 3. Partidas.
 4. Capítulos.
 5. Extraordinarios.
- Debe ser capaz de generar informes impresos con diseños mejorados y con formatos propios y adecuados a las necesidades de la empresa, de ahí la necesidad de contar con una herramienta que permita al usuario diseñar reportes personalizados y modificar los previamente definidos por el sistema.

El generador de reportes debe brindar las siguientes facilidades:

 - Seleccionar los datos que desee aparezcan en el reporte.
 - Definir títulos y tipo de letra en forma dinámica.
 - Incluir el logotipo de la empresa.
- Se requiere una carátula para los reportes de precios (básicos, análisis etc.) adecuada para la dependencia a la cual se envía el presupuesto.
- Presentar los reportes de presupuestos en 15" o en el tamaño que se requiera.
- Manejar una base de datos.
- Incluir seguridad (Manejar claves de acceso a análisis, reportes etc.) en el manejo del sistema.

REQUERIMIENTOS DE INSUMOS



Fig. 2.2.4 Insumos en la industria de la construcción.

- Manejar múltiples precios por insumo dependiendo de la zona económica que se haya seleccionado.
- Captura de precios en moneda nacional o extranjera.
- Presentar la edición de insumos en forma tabular.
- Permitir el acceso a cualquier insumo o matriz que se encuentre en otra obra.
- Permitir el manejo de descripciones para los elementos que se utilizan en la obra sin límite de longitud.
- Manejar familias para agrupación de insumos.

REQUERIMIENTOS DE ANÁLISIS

- Manejar familias para agrupación de análisis.
- Permitir la captura de insumos desde el módulo de análisis.

- Permitir acceder cualquier insumo o análisis que se encuentre en otra obra.

REQUERIMIENTOS DE PRESUPUESTOS

Dentro de la industria de la construcción, el análisis de costos juega un papel muy importante, por lo que es necesario contar con una herramienta para enfrentar la problemática de la ingeniería de costos. Algunos de los requerimientos de presupuestos del usuario son los siguientes:

- Permitir definir los propios formatos de Sobre Costo.
- Presentar el desglose de factor de sobrecostos como hoja de cálculo.
- Manejar presupuestos de mano de obra por destajo (Estimaciones de mano de obra).
- Manejar almacén, materiales y bodega.
- Registrar cantidades negativas (deductivas), descuentos (ejem. P. E. C. E.) • impuestos adicionales (ejem. I. V. A.).

DOCUMENTACIÓN Y AYUDAS

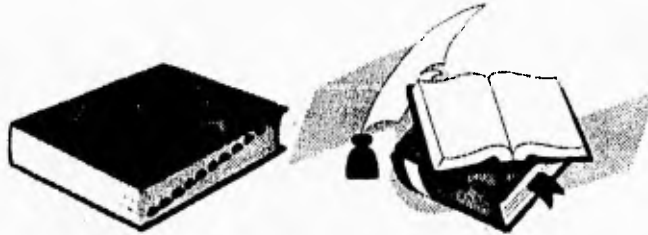


Fig. 2.2.5 Documentación y ayuda.

- Permitir el manejo de errores, los cuales al presentarse sean claros de tal forma que la solución sea identificada rápidamente.
- Incluir información en el manual sobre todos los temas y conceptos que involucra el sistema de precios unitarios.
- Los manuales deben ser realizados de tal forma que tengan una estructura que permita un localización fácil y rápida de la información que se requiere.
- Se debe tener acceso a información mientras se trabaja, mediante la ayuda en línea y el manual de usuario.

REQUERIMIENTOS DE HARDWARE

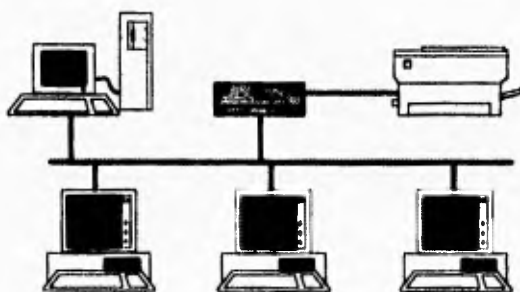


Fig. 2.2.6 Requerimientos de hardware.

El usuario necesita aprovechar los recursos de hardware que posee, por lo que el sistema debe estar diseñado de tal manera que sea compatible y adaptable a la mayor parte de hardware que existe en el mercado.

2.3 DIAGRAMA DE DESCOMPOSICIÓN FUNCIONAL

TÉCNICAS DE ANÁLISIS DE REQUERIMIENTOS

Las técnicas de análisis de requerimientos combinan procedimientos sistemáticos con una notación única para analizar los dominios de información funcional de un problema.

Se ha observado que el dominio de la información se caracteriza por los tres atributos siguientes:

- 1.- Flujo de datos.
- 2.- Contenido de los datos.
- 3.- Estructura de datos.

Los cuáles son comunes en todas las técnicas de análisis de requerimientos

CARACTERÍSTICAS COMUNES

Aunque cada técnica y/o herramienta introduce una nueva notación y heurística de análisis, todas las técnicas pueden ser evaluadas en el contexto de las siguientes características comunes:

- 1.- Mecanismo para el análisis del dominio de la información.

- 2.- Método de representación funcional.
- 3.- Definición de interfaces.
- 4.- Mecanismos para subdividir el problema.
- 5.- Soporte de la abstracción.
- 6.- Representación de las visiones físicas y lógicas.

Las diferentes técnicas y/o herramientas que se utilizan para el análisis del dominio de la información se conduce de forma diferente en cada uno de los mismos siendo factible, reconocer las siguientes guías comunes:

- Todas la técnicas se enfocan (directa o indirectamente) hacia el flujo de dato y al contenido o estructura de estos.
- El flujo se caracteriza generalmente en el contexto de las transformaciones (funciones) que se aplican para cambiar la entrada en la salida.
- El contenido de los datos puede representarse explícitamente utilizando un mecanismo de diccionarios.
- Las funciones se describen normalmente como procesos o transformaciones de la información. Cada función puede ser representada usando una notación específica (ejem. El Diagrama de Flujo de Datos [D.F.D.]).
- La mayoría de la técnicas de análisis permiten al analista evaluar la representación física de un problema antes de derivar a la solución lógica. En general, la notación se utiliza para representar ambos aspectos.

TÉCNICAS ORIENTADAS AL FLUJO DE DATOS

Para estudiar un sistema de información es recomendable que los analistas se cuestionen por lo general, para determinar los requerimientos de los sistemas, lo siguiente:

- 1.- Qué procesos integran al sistema.
- 2.- Qué datos se utilizan en cada proceso.
- 3.- Qué datos se almacenan.
- 4.- Cuáles son los datos que entran y cuáles los que salen del sistema.

La jerarquía de cada una de las preguntas se determina con las técnicas orientadas al flujo de datos.

Con estas técnicas, el analista de sistemas reconoce el papel central e importancia de los datos en una organización con los cuáles, al realizar un seguimiento de flujo de los mismos en un sistema, obtiene una gran cantidad de información con la que se puede identificar, como se están llevando a cabo los objetivos de una determinada organización o empresa, siendo este el propósito del análisis por flujo de datos.

Lo anterior proporciona que al manejar las transacciones y completar las tareas los datos de entrada sean procesados, almacenados, consultados, utilizados, modificados y emitidos, realizándose así procesos específicos, que forman parte del objetivo de una investigación de sistemas.

El análisis por flujo de datos es el encargado de estudiar el uso de éstos en cada actividad, documentando esta información en las dos herramientas siguientes:

DIAGRAMAS DE FLUJOS DE DATOS

Muestran la relación entre los procesos y los datos.

El trabajo básico que hacen los sistemas es la transformación de entrada a salidas, como se muestra en la siguiente figura:

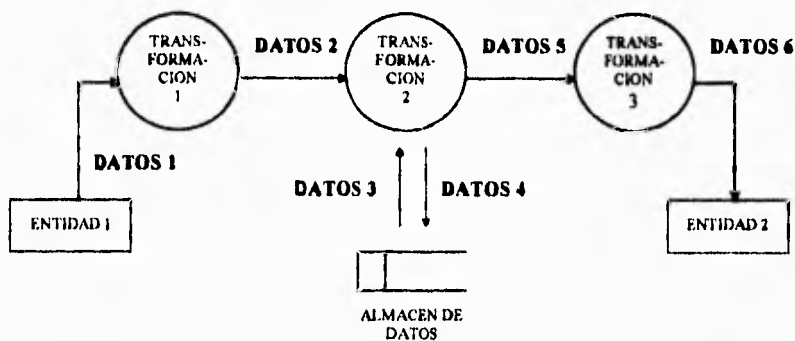


Fig. 2.3.1 Representación del Diagrama de Flujo de Datos (D.F.D.)

El D.F.D. es una técnica gráfica que muestra a un sistema como una red de procesos conectados a cualquier otro por la trayectoria de datos, puede describir:

- Como se relaciona el sistema con el medio ambiente.
- Como las partes de un sistema se relacionan unas con otras.

- El flujo de información y las transformaciones que se aplican a los datos conforme se mueven de la entrada a la salida, obteniéndose de ellos el conocimiento claro para saber como entran, fluyen y salen del sistema.

La forma básica de un D.F.D. se muestra en la siguiente figura:



Fig. 2.3.2 Flujo de Información

Los D.F.D. son usados principalmente para:

- Representar un sistema o software a cualquier nivel de abstracción.
- Subdividir un sistema.
- Analizar interfaces.
- Desarrollar una descomposición funcional.
- Comunicar a usuarios y diseñadores.
- Desarrollar especificaciones.
- Particionarse en niveles que representan el flujo incremental de información y detalle funcional.
- Consistencia entre procesos.

Su funcionamiento radica en no introducir en el "nivel inferior" nuevas entradas o salidas en el siguiente proceso, que no se hayan identificado en los diagramas de mayor nivel, sin embargo, si es posible identificar nuevos flujos de datos y almacenamiento de los mismos.

VENTAJAS

- a) Las actividades paralelas son útiles en diversos organismos, cuyas actividades sean múltiples y concurrentes; mediante el uso de actividades paralelas el D.F.D. permite que los analistas puedan representar diversas actividades que se dan en forma simultánea, mostrándolas con mayor exactitud, según vayan apareciendo, por lo anterior - el D.F.D. - aporta una mayor visión del sistema para lograr un entendimiento entre el usuario y el analista.
- b) La concentración de los datos en los D.F.D.'s, permite que los analistas tengan un gran conocimiento, de todo el sistema, permitiéndoles la plena identificación y descripción de los datos del área de aplicación, aparte de definir cuáles datos son útiles para que sean almacenados.
- c) El análisis del flujo de datos permite a los analistas aislar las áreas de interés, estudiarlas y examinar los datos que entran al sistema. así como observar como se modifican cuando salen del mismo.

- d)** Utilizan una notación bastante clara, sencilla y accesible para las personas que no están familiarizadas con la terminología de computación.
- e)** Permite utilizar una notación sencilla para describir en forma más exacta la actividad de los procesos del sistema de información de manera que al examinar los diagramas puedan señalarse los problemas más rápidamente de manera que se puedan corregir, antes que se inicie otro trabajo de diseño.
- f)** Los D.F.D.'s se desarrollan y utilizan de manera progresiva, que va de lo general a lo particular para el sistema de interés; esto es, generar D.F.D.'s generales (procesos a un nivel de mayor detalle), a partir de los cuales se generarán D.F.D.'s particulares (procesos de un nivel de menor detalle) para tener la posibilidad de encontrar más información sobre los procesos internos.
- g)** Muestran las características lógicas de las aplicaciones, es decir, señalan que ocurre y cuando.
- h)** Proporciona elementos para el análisis y desarrollo de especificaciones.
- i)** Al descomponer un proceso en diversos niveles hasta llegar a las funciones simples y sus interconexiones nos permite ver que conjunto de procesos tiene menor número de conexiones.
- j)** La meta es crear un modelo del sistema en el cual todos los procesos, a todos los niveles, sean funciones simples.

DICCIONARIO DE DATOS

El diccionario de datos D.D. es un segundo componente o técnica que consiste en describir el contenido de los elementos de información de los D.F.D.

El D.D. contiene las definiciones de todos los datos mencionados en el D.F.D. en una especificación del proceso y en el propio diccionario de datos.

El D.D. proporciona:

- a) Información adicional sobre el sistema.
- b) Auxilia el analista involucrado en la determinación de los requerimientos dentro del sistema.
- c) Describe la actividad o entidad que se está estudiando.
- d) Almacena los detalles adicionales y las descripciones de todos los elementos incluidos en el conjunto de los D.F.D.'s, su flujo dentro del sistema, su estructura, así como los procesos de los sistemas.
- e) Define los elementos de información sin ambigüedad.
- f) Ilustra tanto los datos, como flujos a través de un sistema y las transformaciones que se dan en los datos del mismo.

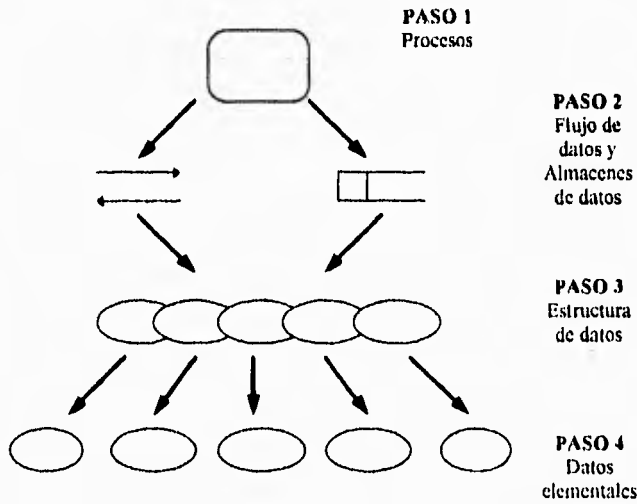


Fig. 2.3.3 Pasos para construir un diccionario de datos.

VENTAJAS

- 1) Proporciona al usuario el tener un conjunto conciso de definiciones para todas las entidades de (los) proceso(s) que conforman al sistema de información que se está estudiando.
- 2) Frecuentemente es la única fuente común de definiciones para los investigadores y usuarios, para ser utilizado como única fuente de respuestas

- a todas las preguntas que se relacionan con el formato y el contenido de los conjuntos de datos utilizados en el sistema.
- 3) Permite encontrar cuales son los flujos de datos que faltan, detectar definiciones duplicadas así como descubrir datos que no se utilicen en ningún proceso durante la investigación, así también evitará problemas posteriores en la determinación de los requerimientos en el diseño de sistemas.
- 4) El D.D. por sí mismo, puede procesarse para revelar información adicional en:
- Listados de elementos datos/estructura de datos. Conjunto completo de todos los datos utilizados en el sistema que se está investigando y que incluyen nombres, descripción, longitud de campo y nombres alternos (alias).

El diagrama muestra una tarjeta de integración de diccionario de datos con los siguientes elementos:

- Una barra superior con una escala de 10 divisiones.
- Un botón circular etiquetado como "Buscar".
- Campos de texto para "Descripción breve" y "Sinónimos (contextos)".
- Una lista de tipos de datos con guiones: "Alfabético", "Alfanumérico" y "Numérico".
- Una sección "Si es continuo:" con campos para "Longitud" y "Rango".
- Un espacio rectangular para valores, con la instrucción: "Si son más de cinco valores, utilice el reverso de la tarjeta o tarjetas adicionales."

Fig. 2.3.4 Tipo de forma que se utiliza para integrar el diccionario de datos.

DIAGRAMA DE DESCOMPOSICIÓN FUNCIONAL

A partir del análisis de requerimientos se identificaron los principales módulos que el usuario desea manejar, en donde los módulos de mayor nivel se fueron desglosando en módulos de menor nivel.

Dentro del módulo principal se definieron los siguientes módulos:

- 1) Insumos
- 2) Factor de Salario Real
- 3) Análisis
- 4) Presupuestos
- 5) Relaciones

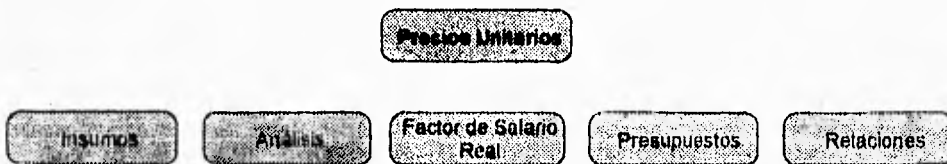


Fig. 2.3.5 Diagrama de descomposición funcional (Módulo Principal).

INSUMOS

Este módulo controlará los Insumos para el presupuesto de obra en cuestión. En el análisis se identificaron cuatro tipos diferentes de insumos: Materiales, Mano de Obra, Maquinaria y Equipo y Subcontratos.

Dentro de Materiales se identificó la necesidad de clasificarlos por Zonas, Precio de Lista asociado o no a una Zona, Porcentaje de Flete aplicado y Porcentaje de Descuento aplicado.

Dentro de Mano de Obra se consideró el Salario Base y el Cálculo del Factor de Salario Real (Mínimo, Mayor al mínimo y definición propia).

En Subcontratos se encontró la necesidad de asignar un subcontrato a diversos contratistas.

FACTOR DE SALARIO REAL

Dentro de este módulo se calculará el Factor de Salario Real, para salarios mínimos, o para mayores al mínimo, el cálculo podrá ser para Obra Pública o Privada.

El sistema deberá ofrecer la ventaja de modificar los factores relacionados con la Ley de Obra Pública o con el Seguro Social en caso de que existiera algún cambio, también debe permitir borrar un factor si éste es cancelado.

Para el cálculo del Factor de Salario Real es necesario tomar en cuenta los siguientes parámetros:

Dentro de la sección dedicada al Factor de Salario Real según la Ley Federal del Trabajo:

- Días Pagados
- Días no Laborados

Dentro de la sección dedicada al Factor de Salario Real por Seguro Social

- Conceptos del Factor por Seguro Social

Dentro de la sección dedicada al Factor de Salario Real por Otros Conceptos

- Otros Factores a considerar

Dentro de la sección que presenta los factores ya calculados con los valores proporcionados.

- Factor Total
- Factor por el Seguro Social
- Factor por la Ley Federal del Trabajo
- Total de Otros Factores

ANÁLISIS

Este módulo se enfocará a manejar cinco tipos de Análisis o matrices:

- Básicos
- Integrados
- Partidas
- Capítulos y
- Extraordinarios.

El módulo presentará el cálculo del importe de cada uno de los elementos que componen el Análisis, así como el costo total del mismo.

PRESUPUESTOS

Este módulo permitirá diseñar la portada del presupuesto, establecer los factores de sobre costo definiendo sus propios formatos de cálculo, actualizar precios, así como realizar la explosión de Insumos.

Dentro de la definición de factores de sobre costo podrán definirse formatos propios, los factores de sobre costo, que se aplicarán al presupuesto y a los análisis al momento de emitir sus reportes.

El módulo debe presentar las siguientes opciones:

- Actualización de Precios
- Carátula del Presupuesto
- Factores de Sobre Costo
- Leyendas y Alias
- Explosión
- Modelo de Costo

RELACIONES

Este módulo debe permitir capturar la información de las zonas económicas, las familias con las que se agruparán los insumos o análisis y capturar los diferentes tipos de monedas que se utilizarán dentro de la obra.

Dentro de zonas se debe permitir capturar la información más relevante de ellas, como: la descripción de la zona, las regiones asociadas con ésta y visualizar los insumos que contiene dicha Zona.

Dentro de familias es importante considerar la descripción de la familia y visualizar los insumos o análisis que están contenidos en dicha Familia.

En un catálogo de monedas se podrá capturar el nombre de la moneda, el símbolo con el cual se representa, el tipo de cambio, así como definir la moneda base para el cálculo del Presupuesto.



Fig. 2.3.6 Diagrama de descomposición funcional.

.4 OPCIÓN DE SOLUCIÓN EN BASE A LOS REQUERIMIENTOS ANALIZADOS

Para la definición del ambiente del Sistema de Precios Unitarios para la Industria de la Construcción se deben de tomar en cuenta varios aspectos para su desarrollo como sistema de procesamiento de datos o información.

Un sistema de procesamiento de datos tiene cuatro componentes básicamente : La máquina donde corre , los programas , los datos y la gente que utiliza el sistema . Las máquinas y los programas son llamados " Hardware & Software" respectivamente.

La invención y desarrollo de las computadoras a partir de la década de los 40's produjo que las empresas respondieran a esta revolución de la computación, así adquirieron e instalaron equipo, entrenaron programadores , analistas y automatizaron los procesos de la oficina. Además se desarrollaron sofisticados algoritmos para extender las capacidades de las nuevas máquinas , se generaron nuevos lenguajes para optimizar los recursos humanos y se optimizaron los sistemas operativos para mejorar el uso de estas. De esta manera la industria de la computación y las empresas usuarias han trabajado muy duro para desarrollar mejores técnicas de programación . Los lenguajes de alto nivel han sido desarrollados a tal grado de que un proceso pueda ser ejecutado sin importar la máquina en que se encuentre.

El sistema a desarrollar, manejará una gran cantidad de información que está dividida en varios conceptos, por lo que su conjunción para facilitar su manejo es a través de una base de datos.

Está base de datos contendrá toda la información requerida para armar un presupuesto, pues éste necesita ejecutar una serie de actividades para llevar a su término la obra. Además de los costos de éstas actividades es necesario saber a detalle como están formadas o armadas, para poder contabilizar globalmente cada material utilizado, cuánta mano de obra necesita, y el equipo a utilizar .

Los papeles de los usuarios son divididos en programadores y no-programadores. Una vez que un DBMS (Sistema Manejador de Bases de Datos) es hecho, su habilidad para crecer y desarrollarse depende de la habilidad de un programador para extender o modificar sus facilidades.

- **Usuarios no programadores:** Usuario Casual
Usuario Paramétrico
Usuario General

- **Usuarios Programadores:** Programador de la Base de Datos.
Programador de Aplicaciones Convencionales.

El usuario casual interactua con el DBMS irregularmente y por lo tanto el sistema debe de emplear un diálogo para ayudar al usuario y minimizar los errores de uso. El usuario paramétrico interactua con el sistema en caminos preestablecidos y restringidos, llamando solicitudes ya almacenadas y dando los parámetros necesarios para

completar esta solicitud. El usuario general interactúa en caminos libres y por lo tanto necesita un lenguaje de alto nivel que sea flexible.

En este caso nuestro sistema debe de dar caminos preestablecidos para que el usuario no tenga que ser todo un experto en la creación de presupuestos, además de llevarlo en forma amigable y en caso de necesitar una ayuda, que ésta sea sencilla de consultar.

Es así que con las facilidades que nos brinde el DBMS que se seleccione, se tendrán que suministrar las ayudas necesarias para que el usuario no tenga problemas en el manejo del sistema y de la información que necesita explotar. Por lo anterior se puede decir que la mejor forma de organizar nuestra información es a través de una Base de Datos relacional, para poder tener en archivos independientes la información, sin mezclar campos o repetir información en cada archivo y teniendo relaciones entre éstos a través de campos claves, para poder accederlos y armar nuestro presupuesto de la manera más sencilla posible teniendo el acceso a todos los archivos que se estén manejando, optimizando el uso de la computadora.

Ahora el software que se está desarrollando, es mucho más fácil de operar y está más integrado y puede en algunos casos utilizarse sin haber leído el manual de operación. Ya que estos sistemas utilizan ventanas y menús para dar además una ayuda en línea en forma interactiva y amigable.

En base a los requerimientos del usuario se llevo a cabo un análisis de algunas herramientas de desarrollo.

HERRAMIENTAS DE DESARROLLO

Dentro de la industria del software existen dos tipos de lenguajes de programación, éstos son :

- Constructores de Soluciones
- Constructores de Componentes

CONSTRUCTORES DE SOLUCIONES

Los constructores de soluciones son programadores que se caracterizan por estar más cerca de los usuarios que utilizarán el software , por lo que conocen más sus necesidades . Éstos usualmente construyen soluciones de negocios, se caracterizan por su habilidad para trabajar con los usuarios hasta refinar sus requerimientos.

CONSTRUCTORES DE COMPONENTES

Los constructores de componentes se caracterizan por no tener ningún contacto con los usuarios para tratar de refinar los requerimientos del mismo, se enfocan básicamente al hardware y no así a las consideraciones de negocios.

Un tercer grupo que juega un papel muy importante son los proveedores de software y se les podría considerar como una extensión de los constructores de componentes. La siguiente tabla resume el papel de cada uno de los grupos. Ver fig. 2.4.1

GRUPO	ENFOQUE	CONJUNTO DE APTITUDES	HERRAMIENTAS REQUERIDAS	HERRAMIENTAS TÍPICAS
Constructores de Soluciones.	Proyecto: Definición de soluciones y explosión de componentes.	Orientado a soluciones de negocios: Relación con el usuario final; comprensión profunda del problema.	Iterativo, bueno como constructor de interfaces y unión de componentes.	Visual Basic Visual C++ Microsoft Access Microsoft FoxPro Microsoft Excel Microsoft Word Microsoft Project.
Constructores de Componentes.	Organización: Construcción de componentes reutilizables.	Orientado a la máquina: Métodos de desarrollo tradicionales; dominio del análisis.	Producción rápida y compacta de código; pruebas rigurosas.	Visual C++ Cobol Fortran MASM
Proveedores Externos.	Vendedor de Componentes.	Desarrollo y mercadeo de componentes genericas de alta calidad y de industrias específicas.	Producción de código de alta calidad.	

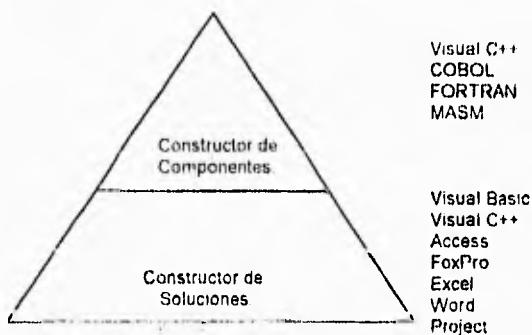


Fig.2.4 1 Constructores de Componentes & Constructores de Soluciones

VISUAL BASIC

Este lenguaje esta disponible para programadores de MS-DOS y Windows. Visual Basic es la herramienta más rápida para crear aplicaciones generales de negocios. Cuenta con una corta curva de aprendizaje, requiere de pocos conocimientos de programación y tiene una gran facilidad para una rápida integración y rehuso de componentes .

Tiene una gran flexibilidad para llevar a cabo llamadas a archivos .DLL y agregar una gran cantidad de controles al gusto del cliente.

Visual Basic constituye un camino fácil para crear interfaces de usuario utilizando componentes preconstruidas , por lo que regresamos a un lenguaje flexible que cuenta con un editor verificador de sintaxis y herramientas de debugging (Perfeccionamiento).

VENTAJAS :

- Es un sistema productivo para crear soluciones en Windows.
- Controles Visuales preconstruidos por terceros.
- Acceso a Bases de Datos.
- Manejo de Multimedia
- Permite ensamblar fácil y rápidamente una interface de usuario con componentes prefabricadas.

- Ofrece una gran capacidad y velocidad en su sofisticado debugger.
- Permite manipular otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excell, Project, etc..), siempre y cuando dichas aplicaciones soporten OLE automation.
- Soporta diversos tipos de Manejador de Bases de Datos.
- Interface al usuario amigable
- Manejo de ayudas en línea robustas.
- Interprete poderoso que permite la detección de errores al momento de edición del programa.
- Aplicaciones escritas que toman ventaja de grandes arreglos y un espacio ilimitado en **strings** (cadenas).
- Maneja aplicaciones Cliente-Servidor
- Visual Basic incluye la base de datos Access, la cuál provee acceso simultáneo con FoxPro, Dbase, Paradox, SQL, etc..
- Un usuario inexperto puede ser productivo con algunas semanas de utilización de Visual Basic.
- Puede ser utilizado como un servidor de OLE Automation.

DESVENTAJAS:

- Visual Basic es poco flexible es decir, se tiene que auxiliar de lenguajes más robustos como el "C" ó Pascal para hacer por ejemplo manipulación de memoria, accesos directos al Sistema Operativo, etc..

- No cuenta con herencia, característica muy importante dentro de la programación orientada a objetos
- Su interprete es lento en la creación de ejecutables.
- Su código no es compatibles con paltformas "no" Windows.

ESPECIFICACIONES TÉCNICAS DE DESARROLLO:

- Requiere de una computadora personal que contenga Windows versión 3.1 o mayor.
- 4 MB de memoria disponible (8 MB o más es la recomendada) Fig.2.4.2
- Disco duro con capacidad mínima de 30 MB.Fig.2.4.3.
- Monitor VGA o de una mayor resolución
- Mouse

ESPECIFICACIONES TÉCNICAS DEL USUARIO:

- Requiere de una computadora personal que contenga Windows versión 3.1. o mayor.
- 4 MB de memoria RAM minima disponible . Fig.2.4.2.
- Disco duro con capacidad mínima de 10 MB. Fig.2.4.3.
- Configuración mínima 486.
- Mouse

A continuación se muestran algunas de las ventajas anteriormente mencionadas.

MEMORIA "RAM" REQUERIDA EN AMBIENTE DE DESARROLLO.

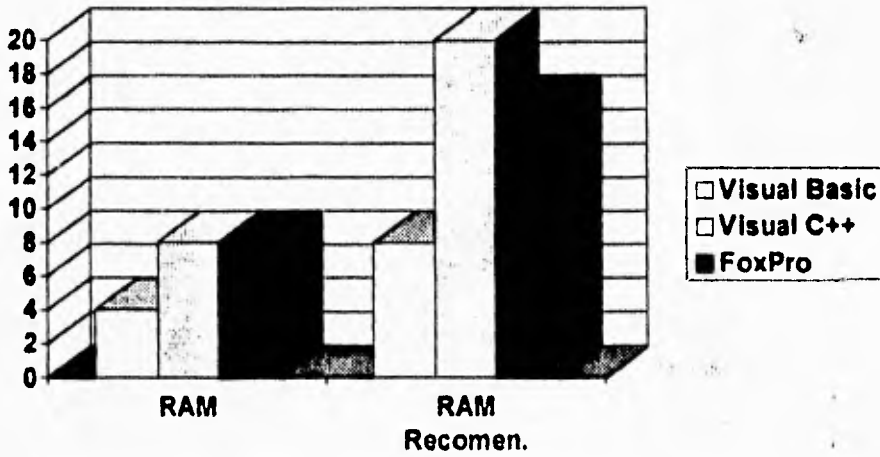


Fig.2.4.2. Requerimientos de Memoria RAM.

VISUAL C++

Es una herramienta flexible y poderosa para generar soluciones de negocios, además de ser una herramienta rápida para programadores que utilizan C y C++.

VENTAJAS:

- Incluye librerías preconstruidas
- Controles Visuales
- Permite la reutilización de Código total
- Manejo completo del esquema de la programación orientada a objetos. (Herencia, Polimorfismo, etc..).
- Maneja controles VBX (Visual Basic eXtension) que se refiere básicamente a componentes preconstruidas por lo que aceleran el desarrollo del software, compatibles con la Ver.1.0. de Visual Basic.
- Todo el manejo de dispositivos de Entrada/Salida se hacen a través de librerías.
- Cuenta con las características de un Visual Workbench (Integra accesos a herramientas como librerías , compiladores, editores ,etc..; Control de proyectos , etc..).
- Cuenta con compilador "no" con interprete.
- A través de el se pueden crear clientes de OLE.

- Es el único lenguaje que al salir una nueva versión de Windows, ya cuenta con las herramientas necesarias para trabajar en él.
 - Manejo de Multitarea.
 - Lenguaje poderoso y flexible ya que permite crear aplicaciones núcleo.
 - Rapidez en ejecución de programas. Ver Fig.2.4.4.
 - Manejo de variables locales y globales.
 - Permite obtener funcionalidades de otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excell , Project, etc..)
 - Manejo de Controles OLE.
 - Total libertad para manejo de memoria y accesos a Sistema Operativo.
- Permite la creación de servidores de OLE.

DESVENTAJAS:

- Curva de aprendizaje alta.
- Tiempos de compilación largos (superable en nuevas versiones).

ESPECIFICACIONES TÉCNICAS DE DESARROLLO:

- Requiere de una computadora personal que contenga Windows versión 3.1 o mayor. (Ver 1.5,16 bits)
- Requiere de una computadora personal que contenga Windows NT o Windows 95 (Ver 2.0,32 bits).

- 8 MB de memoria disponible (20 MB o más es la recomendada) Fig.2.4.2
- Disco duro con capacidad mínima de 60 MB. (Ver 2.0, 90 MB).Fig 2.4.3
- Monitor VGA o de una mayor resolución
- Mouse

ESPECIFICACIONES TÉCNICAS DEL USUARIO:

- Requiere de una computadora personal que contenga Windows versión 3.1. o mayor.
- 4 MB de memoria RAM mínima disponible . Fig 2.4.2
- Disco duro con capacidad mínima de 10 MB sin contar con el programa generado. Fig 2.4.3.
- Configuración mínima 486.
- Mouse

A continuación de muestran algunas de las ventajas anteriormente mencionadas.

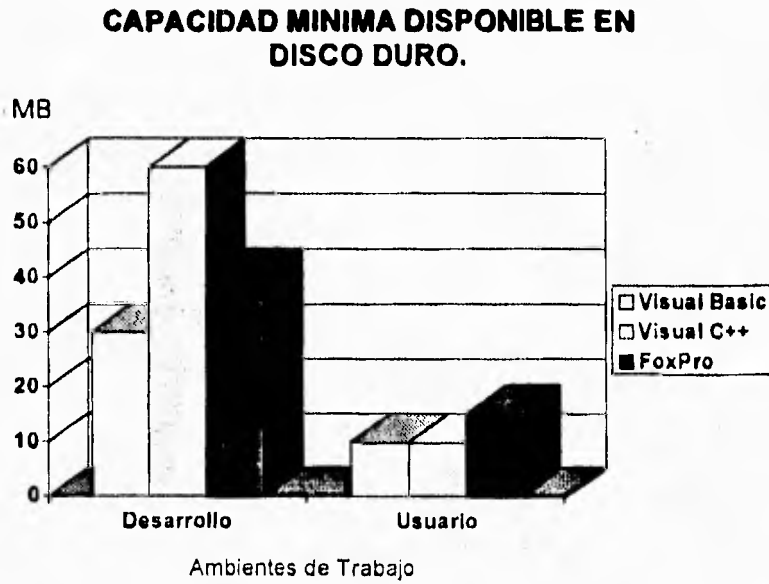


Fig 2.4.3. Requerimientos de Capacidad en Disco Duro

FOX PRO

FoxPro es el administrador de bases de datos más rápido para computadora personales , permite hacer consultas para manejar grandes cantidades de datos.

VENTAJAS:

- Maneja comandos de Xbase
- El modelo de la programación orientada a objetos está completamente implementado es decir, maneja Polimorfismo, Herencia, etc..
- Soporta el manejo de Multiusuarios
- Alta velocidad en queries
- Las aplicaciones hechas en DOS son fácilmente transportables a Windows.
- Los datos de FoxPro pueden ser exportados a ASCII, DBaseIII, WK1, WR1 y Excell.
- Permite explotar la tecnología OLE .
- FoxPro es una herramienta poderosa para el manejo de datos masivos.
- Cualquier objeto de interface al usuario como formas y controles pueden utilizarse como base para crear una nueva clase.
- Cuenta con un ambiente altamente integrado.
- Es una herramienta Multiplataformas
- Permite la conexión a diferentes Bases de Datos.
- Permite la utilización de herramientas OLE.

DESVENTAJAS:

- No puede ser utilizado como un servidor de OLE Automation, es decir no puede manipular aplicaciones.
- Utilización de Interprete en vez de Compilador.
- FoxPro es poco flexible es decir, se tiene que auxiliar de lenguajes más robustos como el "C" ó Pascal para hacer por ejemplo manipulación de memoria, accesos directos al Sistema Operativo, etc..

ESPECIFICACIONES TÉCNICAS DE DESARROLLO:

- Requiere de una computadora personal que contenga Windows versión 3.1 o mayor.
- 8 MB de memoria disponible (16 MB recomendada). Fig 2.4.2
- Disco duro con capacidad minima de 40 MB. Fig 2.4.3.
- Monitor VGA o de una mayor resolución.
- Mouse

ESPECIFICACIONES TÉCNICAS DEL USUARIO:

- Requiere de una computadora personal que contenga Windows versión 3.1. o mayor.
- 8 MB de memoria RAM minima disponible . Fig.2.4.2

- Disco duro con capacidad mínima de 15 MB. Fig 2.4.3
- Configuración mínima 386.
- Mouse

A continuación se muestra una gráfica comparativa entre velocidades de ejecución.

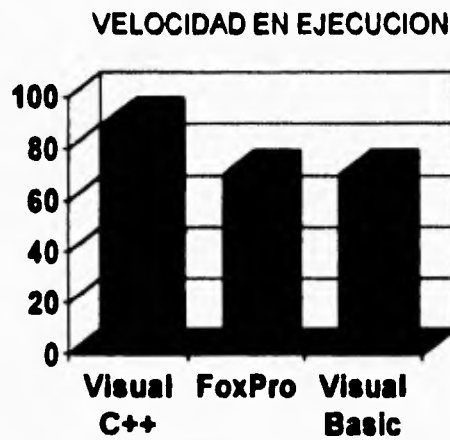


Fig.2.4.4. Velocidad en ejecución de programas.

2.5 ELECCIÓN DE LA SOLUCIÓN OPTIMA.

En función al análisis elaborado anteriormente debe seleccionarse de entre las opciones disponibles aquella que nos permita cubrir los objetivos planteados y que a su vez responda a las necesidades que el usuario tiene y a las cuales el mercado no ha dado solución.

Debe asegurarse que las necesidades del usuario sean cubiertas de manera general, es decir, no es posible elegir alternativas que únicamente proporcionen solución específica a un grupo determinado de usuarios, ni debemos considerar una solución con características particulares que sigan procedimientos no aplicables a la mayoría de los usuarios porque esto dificultaría el alcance de uno de los objetivos más importantes, aquel de proporcionar a todos los constructores de herramientas que les permitan realizar su trabajo en la misma forma que lo venían haciendo pero ahora con el apoyo de las computadoras para así disminuir sus costos y tiempos de trabajo. De otra forma se estaría imponiendo a los usuarios un nuevo mecanismo de trabajo, que no siempre será aceptado por ellos y que además puede no dar los resultados esperados perdiendo así el sentido de un sistema. Es muy importante hacer notar que el objetivo es automatizar los procesos que ellos llevan, únicamente debe presentarse un nuevo esquema de hacer lo mismo de manera más eficiente y no cambiar de fondo los procedimientos que deben realizarse.

También deben considerarse los siguientes puntos relevantes en la selección de la mejor alternativa, ya que ellos permiten definir de manera más precisa los alcances de una solución:

TECNOLOGÍA

Desde un punto de vista técnico se debe buscar que la solución a la necesidad de automatizar las constructoras este montada en una infraestructura de nueva tecnología, es decir, es absurdo pensar en ofrecer un sistema desarrollado en ambientes obsoletos o fuera de tiempo, porque al intentar dar mantenimiento al sistema o cuando sea necesario la generación de un nuevo módulo debemos asegurar la existencia de herramientas adecuadas, herramientas nuevas que de alguna manera permitan mantener de forma relativamente fácil al módulo principal del sistema. Es decir, al realizar un proyecto de este tipo se deben evaluar las nuevas opciones disponibles en el mercado para el desarrollo del sistema planteado.

HERRAMIENTAS DE APOYO

Al definir la base en la cual se desarrollará el sistema, es muy importante pensar en el tipo de rutinas, librerías, utilerías que existen para la herramienta base a utilizar, porque no se trata de empezar de cero, debe buscarse la integración de estas ayudas disponibles en el mercado para disminuir los tiempos de desarrollo y a su vez el monto de la inversión a realizar, es mejor utilizar funciones ya elaboradas para el manejo de

pantallas, de menús, de reportes que intentar desarrollarlos desde sus inicios. Debemos asegurar que dichas herramientas de apoyo permitan dar al sistema los niveles de seguridad y calidad que el usuario requiere.

INNOVACIÓN

Las herramientas que se utilicen en el desarrollo de la aplicación deben permitir dar al sistema aspectos innovadores que llamen la atención del usuario, estas innovaciones no deben requerir de mucho tiempo de trabajo o de mucha inversión, lo que se requiere es ofrecer a los usuarios las características del sistema como ventajas ante los demás sistemas que puedan existir en el mercado, es necesario captar la atención de los usuarios con el sistema, y la mejor forma de hacerlo es ofreciendo características que ningún otro producto tenga en el mercado y además que representen un ahorro en tiempo y costo para los constructores.

INTERFACE AMIGABLE

La industria de la construcción en nuestro país se distingue por lo ordenado de sus procesos, los procedimientos se encuentra claramente definidos, pero no se encuentra en un nivel de conocimiento informático deseable, es decir, las computadoras son subutilizadas por considerarlas muchas veces como una herramienta difícil de aprender a usar, uno de los objetivos al ofrecer este sistema es precisamente darle al usuario una herramienta que le facilite el trabajo y que no requiera de un largo proceso de

aprendizaje, es por ello que debe tomarse en cuenta para selección de la mejor alternativa a aquellas herramientas y plataformas de desarrollo que cumplan con este objetivo. Desde el inicio de la relación entre el usuario y la aplicación deberá existir una intuición natural en el uso del programa de forma que el usuario sienta que realmente la aplicación lo ayuda en la realización de su trabajo.

RELACIÓN COSTO-BENEFICIO

La plataforma de desarrollo y los recursos mínimos de operación para la aplicación muchas veces dependen de las herramientas en que ha sido desarrollada, es por ello que debemos tomar en cuenta que existen usuarios con una infraestructura de equipos bastante aceptable y para los cuales se podría desarrollar una aplicación eficiente que consuma bastantes recursos, sin que ello represente un problema para estos usuarios, pero no debemos olvidar a aquellos usuarios que también realizan los mismo procesos y que no cuentan con equipos tan sofisticados. Tomando en cuenta estos puntos debe definirse una plataforma de desarrollo y herramientas en un punto intermedio, es decir que no requieran de gran cantidad de recursos pero si con un mínimo que permita ofrecer una aplicación eficiente.

TIEMPO DE RESPUESTA

Los tiempos de respuesta de una aplicación dependen de los lenguajes y herramientas utilizados en su desarrollo, así como de la forma en que se lleve a cabo la programación y las características del equipo en que sea ejecutado. Lo importante aquí es tomar en cuenta que la herramienta para el desarrollo debe permitir dar un tiempo óptimo de respuesta a los usuarios de acuerdo a los equipos con los que cuente cada uno de ellos.

COMPATIBILIDAD

Las aplicaciones deben permitir al usuario la interacción con las más comunes dentro del mismo ambiente de operación. Esto se debe a que existen aplicaciones específicas para cada proceso a realizar dentro de una constructora, por ejemplo, un sistema de precios unitarios deberá poder comunicarse de alguna forma con un programa de inventarios, de compras e incluso de control de pagos de mano de obra.

A continuación se muestra una tabla con las características, plataforma, herramientas y lenguaje de desarrollo que se considera cumplen con los puntos anteriores. Más adelante se presenta la justificación de porqué usar cada una de ellas:

TIEMPO DE RESPUESTA

Los tiempos de respuesta de una aplicación dependen de los lenguajes y herramientas utilizados en su desarrollo, así como de la forma en que se lleve a cabo la programación y las características del equipo en que sea ejecutado. Lo importante aquí es tomar en cuenta que la herramienta para el desarrollo debe permitir dar un tiempo óptimo de respuesta a los usuarios de acuerdo a los equipos con los que cuente cada uno de ellos.

COMPATIBILIDAD

Las aplicaciones deben permitir al usuario la interacción con las más comunes dentro del mismo ambiente de operación. Esto se debe a que existen aplicaciones específicas para cada proceso a realizar dentro de una constructora, por ejemplo, un sistema de precios unitarios deberá poder comunicarse de alguna forma con un programa de inventarios, de compras e incluso de control de pagos de mano de obra.

A continuación se muestra una tabla con las características, plataforma, herramientas y lenguaje de desarrollo que se considera cumplen con los puntos anteriores. Más adelante se presenta la justificación de porqué usar cada una de ellas :

CARACTERÍSTICA	SELECCIÓN
Plataforma de Desarrollo	Microcomputadoras IBM-PC Compatibles
Ambiente Operativo para la Aplicación	Microsoft Windows
Lenguaje de Desarrollo	MS Visual C++ V1.5
Manejador de Base de Datos	Raima Data Manager
Herramientas de Apoyo	Microsoft Foundation Clases ToolBox

PLATAFORMA DE DESARROLLO

Existen en México un gran número de empresas dedicadas al ramo de la construcción regularmente son compañías chicas que subcontratan servicios de otras compañías por lo que sus necesidades de cómputo pueden cubrirse con Microcomputadoras, e por ello que se ha decidido desarrollar el producto en esta plataforma. Para la constructoras relativamente grandes que desean compartir información entre le diferentes áreas que las componen, la solución más atractiva es la instalación d Redes de Área Local, la cual se forma en base a PC's . Tomando en cuenta lc puntos anteriores podemos observar que la opción adecuada para la construcción di sistema son las PC's. Ahora bien, existen constructoras con mucho mayor capacidad que llegan a utilizar redes o sistemas multiusuarios con diferentes sistemas operativo a Windows y/o MS-DOS, tal es el caso de redes Unix. Para estas constructoras lo má

adecuado es trabajar sus presupuestos en Microcomputadoras y después con un módulo de interface pueden transportar su información a sus sistema multiusuario e integrarlo a sus aplicaciones específicas.

Buscando cumplir el objetivo de proporcionar un producto capaz de satisfacer las necesidades de la mayoría de las empresas constructoras, consideramos que esta es la plataforma más adecuada, desde el punto de vista técnico y desde la perspectiva comercial que se pretende dar al producto.

AMBIENTE OPERATIVO PARA LA APLICACIÓN

Hasta hace algunos años, el sistema operativo MS-DOS venía siendo el más común instalado en las Microcomputadoras, de hecho, actualmente se encuentra instalado en la mayoría de ellas, pero ahora existen tendencias a desaparecerlo lo cual ha sido originado por diferentes causas. Las aplicaciones desarrolladas en DOS fueron muy exitosas y el usuario aprendió a utilizarlas ya que le permitían realizar su trabajo de una manera más rápida, si embargo la misma necesidad de contar con aplicaciones extensas para mercados verticales muy específicos como la construcción, administración, contabilidad, etc. trajo consigo que dichas aplicaciones se hicieran más difíciles de entender, por lo que el usuario debía tomarse un tiempo de entrenamiento para saber utilizar cada aplicación, estos entrenamientos representaban un tiempo sin productividad en las empresas así como también implicaban un costo. Con las aparición en el mercado del ambiente operativo MS-WINDOWS en un principio no se lograron resultados que pudiesen solucionar estos problemas, fue hasta

la versión 3.0 donde Microsoft dio un giro completo a la interface buscando que el usuario pudiese utilizar las aplicaciones casi de inmediato, la interface gráfica de MS-WINDOWS considerando la utilización de los iconos (pequeñas imágenes representativas de la aplicación), comandos en forma de botones hicieron más fácil el aprendizaje para los usuarios. Si a esto agregamos que a partir de Windows V3.0 se estableció un estándar para todas las aplicaciones que quisieran ejecutarse bajo este ambiente, buscando así que el usuario al saber operar Windows ya conociera de manera general el funcionamiento de cualquier aplicación Windows.

Actualmente se cuenta en el mercado con la versión 3.1 de Windows, se espera la nueva versión, en la cual se ofrecen aún nuevas características de interface que permitirán al usuario utilizar al máximo los recursos de su máquina de una manera mucho más fácil. Debemos tomar en cuenta también que el mismo mercado ha estado desplazando el uso de aplicaciones para MS-DOS por lo cual no sería adecuado a estas alturas pensar en desarrollar un aplicación en modo texto. Con la nueva versión de Windows se espera una sustitución total del sistema operativo MS-DOS, porque Windows ya será una sistema operativo completo; actualmente Windows requiere del sistema operativo MS-DOS y se ejecuta después de que este ha sido cargado.

Se seleccionó desarrollar la aplicación en MS-Windows porque nos permitirá desarrollar un producto amigable, con una buena interface al usuario y sobre todo con innovación tecnológica. Deberá cumplir los estándares y además podrá ser utilizado en la siguiente versión de Windows.

Sin importar las herramientas de desarrollo que se utilicen, la programación en Windows es diferente de la forma en que tradicionalmente se venía programando.

Antes de comenzar a hacer programas bajo Windows, es necesario conocer el funcionamiento fundamental de Windows desde el punto de vista técnico:

Normalmente los programas en MS-DOS escriben directamente en la memoria del vídeo y en la memoria de los puertos de impresión, la desventaja de esta técnica es que era necesario sustituir los **drivers** (archivos de control) utilizados para cada tipo de vídeo y para cada tipo de impresora, lo cual de alguna manera hace no muy eficiente a los programas. Windows introduce un nuevo concepto **GDI (Graphics Device Interface)** (dispositivos de interface gráfica) los cuales son manejados por Windows sin necesidad de que cada programa deba configurar las características de vídeo y/o de impresora.

Para los programas en MS-DOS es necesario codificar constantes, inicializaciones y datos o debían manejarse archivos de datos separados que se leían al momento de ejecución del programa, ahora Windows permite almacenar este tipo de datos en un archivo de recursos con un formato específico, los cuales no se compilan junto con el demás código, sino que únicamente son incluidos en los programas al momento de hacer el ligado.

El uso de memoria por los programas para MS-DOS fue siempre limitada a 640 KB, incluso el tamaño de los programas se encontraba igualmente limitado, se requerían varios manejadores de memoria extendida/expandida (memoria superior a los 640 KB), ahora con Windows y Visual C++ se cuentan con herramientas y características que

nos permiten obtener un mejor aprovechamiento de esta memoria de una manera confiable y fácil.

En el ambiente de MS-DOS todos los módulos de cada programa son ligados o unidos para generar un solo archivo que cuente con todas las funciones necesarias para su ejecución, en Windows aparece el concepto de Librerías Dinámicas (DLL) las cuales ayudan a ahorrar memoria y espacio en disco, porque muchas de ellas son utilizadas por diferentes aplicaciones, además permiten dar una mayor modularidad a la programación porque las librerías se compilan y se prueban por separado.

Algunas ventajas nuevas que ofrece Windows son el concepto de OLE y TrueType, la primera le permite al usuario integrar objetos de diferentes aplicaciones en una sola, es decir generar documentos o archivos que incluyan características de diferentes aplicaciones. El TrueType es un tipo especial de letras o caracteres que siempre tendrán la misma forma independientemente de la escala en la que se impriman sobre pantalla o impresora.

LENGUAJE DE DESARROLLO

Al pensar en el desarrollo de un producto para miles de usuarios debe buscarse un lenguaje que permita cubrir todas las necesidades del usuario. Actualmente existen en el mercado una gran variedad de herramientas para la construcción de aplicaciones Windows, tal es el caso de Visual Basic, FoxPro, C-Borland, Turbo Pascal para Windows y Microsoft Visual C++. Siempre se ha reconocido que C es un lenguaje de alto poder por la simplicidad de sus instrucciones básicas y por el tiempo de respuesta que proporciona. Por lo tanto en una primera instancia se pretende desarrollar el producto en C, algunas de las herramientas anteriormente mencionadas ofrecen gran facilidad para el desarrollo pero no pueden llegar a moldearse de acuerdo a las necesidades de los usuarios de un mercado de tipo general. Visual Basic es utilizado para desarrollos a la medida, que no deben tomar mucho tiempo en construirse y que solo estarán enfocados a satisfacer las necesidades de un grupo reducido de usuarios.

Visual C++ se ha encontrado como una de las mejores alternativas, no implica que sea la mejor ya que el tiempo que toma desarrollar una aplicación en este lenguaje requiere de mayor capacitación de los desarrolladores y toma mayor tiempo, sin embargo de acuerdo a las características que proporciona, consideramos que bien vale la pena el desarrollo con esta herramienta. A continuación se mencionan algunas de las características más relevantes de Visual C++ :

En realidad Visual C++ contiene 2 aplicaciones completas para el desarrollo de sistemas, una de ellas es Lenguaje C, con el cual se desarrollan aplicaciones para Windows utilizando el un conjunto de librerías llamado **SDK (Software Development Kit)** y contiene además el compilador de C++ con la librería MFC que hacen la labor del desarrollador más fácil.

El ambiente de trabajo en Visual C++ es un ambiente interactivo para los desarrolladores, lo cual rompe con el esquema de la compilación en línea que se venía utilizando anteriormente. el ambiente de trabajo en Visual C++ contiene un editor que cumple con las características de Windows con opciones que ayudan al desarrollador a ordenar su código.

AppStudio Resource Editor.- Visual C++ cuenta con un herramienta que permite la edición de cajas de dialogo, bitmaps, botones, tipos de caracteres, etc. Anteriormente existían aplicaciones por separado que ayudaban al desarrollador con estas tareas.

Linker.- Para generar programas ejecutables cuenta con un **Linker (programa para integrar el código de los módulos de la aplicación)** al cual se le especifica si se utilizarán las librerías de MFC y el linker se encargará de integrar los modelos de memoria adecuados.

El **Compilador de Recursos**.- incluye la utilerías para compilar los archivos de recursos, independientemente si estos fueron generados con el AppStudio o con algún editor diferente.

Debugger.- utileria que permite seguir paso a paso la ejecución del código de un programa, de forma que permite encontrar los errores de una manera más rápida y sencilla.

AppWizard.- Es una utileria que permite generar código base de una aplicación, con las características, nombres y campos que el desarrollador especifique. Esto permite ahorrar tiempo en el desarrollo.

ClassWizard.- Es una utileria que permite dar mantenimiento al código de **clases** (unidades de programación) y es muy útil para el desarrollador.

Ayuda en Línea.- Todos los manuales de referencia y guías de usuarios son incluidos como una ayuda en línea, de forma que le ahorran al usuario a ahorrar tiempo en consulta a manuales.

Apoyo Externo. - Cuenta con utilerías de apoyo en la revisión de la memoria utilizada por un programa desarrollado en Windows y un compilador de ayudas.

MANEJADOR DE BASE DE DATOS

El hecho de pensar en desarrollar un producto con posibilidades de migrar a otras plataformas en un futuro partiendo de que funciona actualmente en Microcomputadoras, nos define un camino a seguir cuanto a la utilización de las librerías de un manejador de base de datos (engine) , por tanto es lógico pensar en hacer uso de las rutinas de una manera "indirecta", esto es, agregar código en una función de la aplicación que haga uso de las funciones de un manejador de base de datos determinado, para así tener la facilidad de migrar en un futuro la aplicación a otro ambiente sin tener una dependencia.

Algunas de las características deseables más relevantes en el manejador de base de datos, son las siguientes:

- Manejo en Red.
- Velocidad.
- Sencillez de uso.
- Portabilidad.
- Volumen de Información que se pueda manejar.
- Documentación.
- Soporte Técnico.
- Disponibilidad del Código Fuente.
- Facilidad para migrar a otros ambientes.

MANEJO EN RED

Raima Data Manager tiene la posibilidad de funcionar en un ambiente de red, para que en un futuro sea posible implementar su funcionamiento. Este manejador mantiene el control, haciendo su uso transparente y sencillo para el desarrollador.

VELOCIDAD

Se obtuvo información de la revista C-GAZETTE de mayo de 1991, en la cual se presentaban los resultados de algunas pruebas realizadas entre algunos de los manejadores considerados como los mejores y más rápidos. Dicha prueba consistió en observar los tiempos de duración para cada una de las pruebas, los tiempos se midieron en **clock-ticks (pulsaciones de reloj)** de 18.2 ticks/seg. Se realizaron en una AT a 10 Mhz, las entradas y salidas se hicieron desde un disco virtual. (RAM DISK), se genero una base de datos de 5000 registros obteniendo los siguientes resultados:

MANEJADOR	TIEMPO FINAL
C-TREE	6312
RAIMA	4764
PARADOX	8622

con lo cual podemos deducir que el tiempo de respuesta de Raima es bastante aceptable y nos ayuda a cubrir este requerimiento.

SENCILLEZ DE USO

En el programa de demostración se pueda apreciar de una manera muy sencilla el uso de las librerías de acceso a datos, ya que la base de datos se maneja como una sola entidad y el manejador se encarga automáticamente de intercalar las llaves en una forma muy sencilla, disminuyendo así los tiempos de desarrollo.

PORTABILIDAD

Existe la posibilidad de trabajar bajo los siguientes ambientes DOS, WINDOWS, VMS, UNIX, OS/2 Y BSD 4.2. Lo cual nos representa una fácil migración en un futuro.

VOLUMEN DE INFORMACIÓN

Raima permite manejar un límite de 16 millones de registros por tabla, donde la longitud del registro puede ser de hasta 32 KB. Se considera que de acuerdo a las características requeridas para el desarrollo del producto, cubre las necesidades aún previniendo el posible crecimiento.

DOCUMENTACIÓN

El manual de referencia se encuentra bastante claro y entendible, se encuentra la documentación para códigos de error, acceso a datos, búsquedas, ingreso de registros nuevos y se encuentran ejemplos de cómo utilizarlos.

SOPORTE TÉCNICO

Raima proporciona un servicio de soporte técnico durante 60 días a partir de la fecha de compra sin costo, en caso de ser requerido después de este período deberá pagarse una cuota que consideramos es adecuada.

DISPONIBILIDAD DE FUENTES

Se puede solicitar al distribuidor los costos de adquisición del código fuente del manejador, con la previsión de que pudiese necesitarse un cambio interno de uno o más rutinas.

LENGUAJES SOPORTADOS

Raima soporta ser utilizado por C y C++, lo cual corresponde al lenguaje en el que será desarrollada la aplicación.

ADICIONALES

El manejador cuenta con sus propias rutinas para la reconstrucción de datos en caso de que estos hayan sido dañados por alguna interrupción de energía o u otro factor. En la revista **BYTE** se encontró que Raima es el manejador que ofrece un mejor performance

HERRAMIENTAS DE APOYO

MFC

Originalmente se crearon DLL's para ser usadas con lenguaje C, y C++ ha creado algunas complicaciones en el uso de ellas , es por ello que MFC provee librerías dinámicas que ayudan a generar un aplicación en Windows, y de cualquier forma cada desarrollador puede generar sus propias librerías dinámicas y si decide utilizar librerías estáticas también pueden integrarse en cualquier aplicación.

No debe olvidarse que este conjunto de librerías trabajan de la mejor forma en desarrollo con Visual C++, lo cual nos proporciona grandes ventajas durante la

construcción de la aplicación. Ayudan a proporcionar al usuario una interface estándar con las aplicaciones Windows y permite obtener el producto en un tiempo razonable.

TOOLBOX

Estas librerías ayudarán en el manejo de estructuras necesarias en la construcción de la aplicación. Además de que contienen un control de errores que ayuda a evitar generar errores graves en el producto. Estas librerías son de las más utilizadas actualmente y su costo no es alto.

EN CONCLUSIÓN

Considerando todas las características anteriores para las herramientas en evaluación se decide utilizar las siguientes, que de alguna manera permiten obtener grandes beneficios y que fueron consideradas desde el inicio de nuestro análisis de herramientas.

SELECCIÓN
Microsoft Windows
MS Visual C++ V1.5
Raima Data Manager

El trabajar bajo estas herramientas y con sus características nos permiten obtener la calidad que es necesaria para el producto a desarrollar, de entre los que destacan:

Ambiente amigable para el usuario, fácil aprendizaje y operación intuitiva que cumple con el estándar de aplicaciones en su tipo en el mercado. (aplicaciones para Windows).

Utilización de herramientas de gran poder para mejorar el funcionamiento de la aplicación, con la seguridad de que dichas herramientas son actualmente mejoradas para contar en el futuro con versiones más adelantadas de las mismas, que permitan proporcionar un fácil y adecuado mantenimiento a las aplicaciones, además de hacerlo a un costo bajo lo cual representa uno de los puntos más importantes que deben cubrirse.

Se considera que Raima DataBase Manager, es un manejador de base de datos poderoso, que cumple con las características de velocidad y manejo en RED

SELECCIÓN
Microsoft Windows
MS Visual C++ V1.5
Raima Data Manager

El trabajar bajo estas herramientas y con sus características nos permiten obtener la calidad que es necesaria para el producto a desarrollar, de entre los que destacan:

Ambiente amigable para el usuario, fácil aprendizaje y operación intuitiva que cumple con el estándar de aplicaciones en su tipo en el mercado. (aplicaciones para Windows).

Utilización de herramientas de gran poder para mejorar el funcionamiento de la aplicación, con la seguridad de que dichas herramientas son actualmente mejoradas para contar en el futuro con versiones más adelantadas de las mismas, que permitan proporcionar un fácil y adecuado mantenimiento a las aplicaciones, además de hacerlo a un costo bajo lo cual representa uno de los puntos más importantes que deben cubrirse.

Se considera que Raima DataBase Manager, es un manejador de base de datos poderoso, que cumple con las características de velocidad y manejo en RED

planteadas para así proporcionar a la aplicación en construcción estas características que puedan diferenciarse de la competencia.



CAPITULO III

***BASES DE DATOS Y
CARACTERISTICAS DE
LA PROGRAMACION
ORIENTADA A OBJETOS***

3.1 ANTECEDENTES

Los sistemas administradores de bases de datos emergieron en el mercado al final de la década de los 60s, hoy son lo suficientemente maduros y poderosos siendo la capa principal en la cual se sustentan los sistemas de información. El objetivo de los DBMS son el proporcionar herramientas poderosas para manejar base de datos, donde una base de datos esta definida como una colección integrada de datos compartidos. Un DBMS típico proporciona un número específico de funciones:

1. Mantenimiento de estructuras de datos físicos
2. Lenguajes para almacenar y recuperar los datos: estos pueden ser lenguajes de tercera generación (3gls) como COBOL, o lenguajes orientados a queries como SQL y más recientemente los 4GLs como FOCUS, NOMAD y RAMIS
3. Facilidades para asegurar la integridad de los datos
4. Soporte para acceso multiusuario, incluyendo usuarios quienes simultáneamente están actualizando la base de datos
5. Medidas de respaldo y recuperación de los datos
6. Facilidades para mantenimiento de meta-data, por ejemplo a través de un diccionario de datos o sistema de catálogo
7. Independencia de datos -El aislamiento de datos físicos desde la forma en la cual son vistos por los usuarios y los programadores de aplicaciones (vista lógica)

8. Herramientas para administración de bases de datos; estas pueden incluir herramientas tanto para el diseño como para el monitoreo.

La ventaja los sistemas de bases de datos sobre los sistemas de archivos convencionales fue su habilidad para integrar datos pertenecientes a diferentes aplicaciones (contabilidad, libro mayor, personal, control de inventarios, etc.) dentro de una organización, esto permitió compartir los recursos corporativos. Vino además a solucionar los problemas de incompatibilidad, redundancia y dependencia de datos, los cuales estaban ligados a los tradicionales sistemas de archivos. Los sistemas de información computarizados han sido regularmente utilizados en muchas organizaciones por un número de años antes de que la administración reconociera el rol vital que el manejo de datos por este tipo de sistemas jugaban en la eficiencia y productividad de la organización. Los sistemas de información no son simplemente utilizados como herramientas de manejo de datos en el cual se pueden hacer los balances, proceso de ordenes, impresión de pagos de cheques, etc., ellos proporcionan soporte para la toma de decisiones para lograr una estrategia bien planeada.

La mayoría d los DBMS comerciales disponibles hoy en día están basados en la arquitectura ANSI-SPARC, la cual divide el sistema en tres niveles - interno, conceptual y externo.

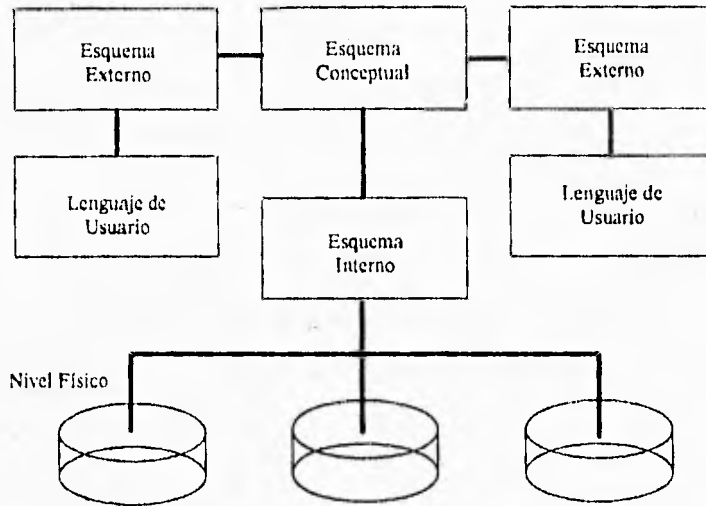


Figura 3.1.1 Arquitectura de tres niveles ANSI-SPARC

COMPONENTES DE UN DBMS

Los DBMSs son altamente complejos, son piezas sofisticadas de software las cuales proporcionan un manejo confiable de datos compartidos. Las bases de datos pueden manipular datos que van desde kilobytes hasta gigabytes. No es posible generalizar la estructura de componentes de una DBMS ya que estas pueden variar de sistema en sistema. De cualquier forma, es posible identificar el número de funciones claves las cuales son comunes para todos los DBMSs.

En el corazón de una DBMSs se encuentra el catálogo de sistema o diccionario de datos, con el cual virtualmente todos los otros componentes del sistema interactúan. El

diccionario contiene todos los meta-datos del sistema, es en sí una descripción de todo el sistema (tipos, longitudes, valores permitidos, etc.), permisos de acceso, vistas entre otras. Es a través del diccionario que los tres niveles de la arquitectura ANSI-SPARC están implementados como diccionarios, el cual contiene los esquemas y las relaciones entre ellos.

EL Interpretador de queries es responsable de la aceptación de comandos desde usuarios finales como de procedimientos almacenados, obtener y actualizar datos en la base de datos. Usando la información almacenada en el diccionario, este interpreta los requerimientos y los traduce hacia peticiones de accesos físico de datos lo cual puede ser comunicado al sistema operativo para su procesamiento. Los programas de aplicación escritos en 3GLs usando adiciones al lenguaje a través de comandos especiales para manejo de bases de datos, dichos comandos son traducidos para después convertirlos en sentencias del lenguaje 3GL.

El proceso de obtener y actualizar datos por el usuario (o programa de aplicación) es llamado **transacción**. Un DBMS convencional generalmente tiene la facilidad para soportar muchos usuarios simultáneamente para acceso a la base de datos (muchas transacciones activas al mismo tiempo). Es un hecho que en sistemas multiusuarios de un solo procesador en estricto sentido de la palabra no es posible hacer algo simultáneo. De cualquier forma, ambiente multiusuarios permiten a varios procesos estar activos juntos dentro del sistema, cuya responsabilidad se encuentra en la CPU para intercambiar estos procesos. Es responsabilidad del **manejador de transacciones** del DBMS asegurar que las transacciones no interfieran una con la otra y corrompan la base de datos.

Uno de los principales objetivos de la tecnología de bases de datos fue la de proporcionar mejores facilidades de respaldo y recuperación que eran dadas por los sistemas tradicionales de procesamiento de archivos a los cuales reemplazaron. Nada es 100% confiable, los sistemas pueden estar diseñados para tolerancia a fallas y recuperar correctamente y en forma confiable. Este es el rol del **manejador de recuperación** de una DBMS para minimizar los efectos de fallas en las bases de datos. Existen diferentes causas de fallas en los ambientes de bases de datos, desde errores en los programas (ejemplo: divisiones por cero) pasando por fallas en discos, fallas de energía eléctrica y errores de operación. El objetivo del manejador de recuperación es la de recuperar la base de datos a su estado conocido como consistente. Debido a las numerosas causas de las fallas y la complejidad del procesamiento de datos, el trabajo de recuperación y la complejidad de procesamiento, el trabajo del manejador de recuperación es bastante difícil y puede representar el 10% o más del código de un DBMS.

Los DBMSs son generalmente clasificados de acuerdo a como los datos son representados en el esquema conceptual (por ejemplo: esquema conceptual de datos.)

De ahí que hay tres principales accesos:

1. Relacional
2. De Red
3. Jerárquico

De estos modelos las bases de datos relacionales han sido las más exitosas en su expansión, debido a que se basan en modelos matemáticos y por el amplio respaldo que le ha dado la industria. No por esto el modelo de red deja de tener ventajas que pueden ser utilizadas al máximo al momento de implementar una base de datos dentro de una aplicación.

El desarrollo de la tecnología de bases de datos ha resultado en el desarrollo de técnicas para un buen diseño de bases de datos, este es el diseño del esquema conceptual. Los diseños lógicos de las BD toma en cuenta todos los atributos que son almacenados dentro de una BD, examinando las relaciones entre ellas con una vista para agruparlas dentro de relaciones como una forma de eliminar la redundancia de datos no necesarios evitando así características no deseadas. Este proceso es conocido como normalización.

El diseño de una base de datos siempre debe ser precedido por una fase de requerimientos, la cual puede identificar entre otras cosas los datos (atributos) que deberían estar almacenados en la base de datos, esto es mediante el establecimiento de relaciones del uno con el otro. Toda la información que interviene en el proceso de creación de una base de datos puede ser representada en forma de reglas, frecuentemente llamadas **reglas de negocio** porque especifican la forma en la cual los datos son utilizados por el negocio siendo modelado dentro de la base de datos. Una parte vital dentro del diseño de una base de datos es el de identificar las **dependencias funcionales**, por ejemplo:

1. Un insumo está identificado por una descripción, un costo, unidad, fecha de costo y fecha de cálculo.
2. Cada insumo tiene solamente una clave, unidad y fecha de costo.

De la primera regla, si nosotros obtenemos una clave de insumo está esta asociada con un insumo que tiene una clave única, descripción, costo, unidad fechas de calculo y costo. En este caso, decimos que el insumo determina o es determinante de la clave, descripción costo, unidad fechas de costo y cálculo; alternativamente descripción, costo, unidad, fechas de costo y cálculo son dependientes del Insumo. La notación estándar es la siguiente:

Insumo#->Descripción

Insumo#->Costo

Insumo#->Unidad

Insumo#->Fecha de Costo

Insumo#->Fecha de Cálculo

Las dependencias funcionales son ejemplos de restricciones de integridad que los atributos pueden tomar. Cuando un nuevo renglón o instancia está siendo agregado a la base de datos, cada valor de los atributos debe ser checado contra cualquiera de las restricciones que involucran al atributo. Un módulo de un DBMS es llamado **checador**

de restricciones y es responsable de la validación de estas. Por ejemplo si alguien trata de agregar un segunda clave para un insumo con la dependencia funcional

Insumo#->Descripción

el checador de integridad debe mostrar el error.

El proceso de normalización involucra la identificación de dependencias funcionales dentro de las reglas de negocio, produciendo así un conjunto completo de relaciones normalizadas. Existen varias formas de lograr lo anterior, una técnica muy recurrida es la de determinancia o diagramas de dependencia. Si A determina B, esto es $A \rightarrow B$, entonces se tiene la siguiente representación

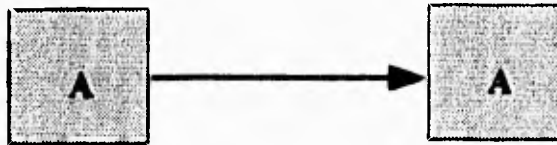


Figura 3.1.2 Ejemplo de diagrama de determinación

El diagrama de determinación para nuestro BD de insumos es la siguiente

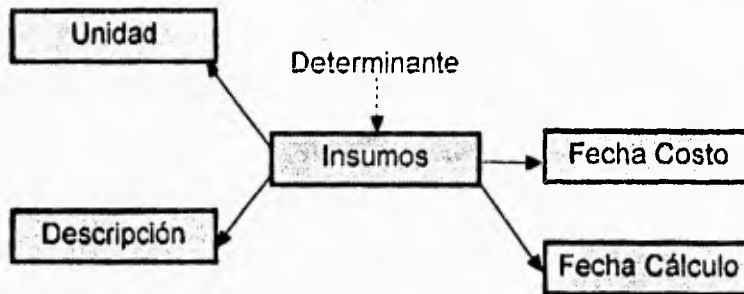


Figura 3.1.3 Diagrama de determinación de Base de datos de Insumos

Toda base de datos debe cumplir con las siguientes reglas de oro:

- Independencia de los datos
- Habilidad de compartir datos
- Irredundancia
- Habilidad de relacionar
- Integridad
- Flexibilidad de acceso
- Seguridad
- Rendimiento y eficiencia
- Administración y control

El concepto de *independencia de los datos* es fundamental en los DBMS's. Denota la independencia o el aislamiento de un programa de aplicación respecto a las

organizaciones lógicas o físicas específicas, y respecto a las consideraciones del almacenamiento de las estructuras de datos y archivos que ese programa use.

El concepto de la *habilidad para compartir datos* es la idea de poseer la capacidad de que varios programas de aplicación compartan en forma independiente una base de datos.

La manera tradicional de proceder, específicamente el recopilar y codificar datos para programas específicos y por tanto "amalgamar" los datos en forma más o menos permanente y exclusiva a esos programas, ha ocasionado la costosa redundancia de registros y archivos en los sistemas de aplicación implicados. La eliminación de la redundancia conduce a la posibilidad de compartir. La redundancia resulta cuando dos aplicaciones diferentes requieren el mismo dato elemental y no lo comparten. Sin embargo, puede ser necesaria cierta redundancia para evitar un tiempo de acceso excesivo a los registros. Es verdad que las redundancias conllevan los problemas de mantenimiento para asegurar que la información redundante sea consistente.

La *habilidad para relacionar* es un término que se usa con mucha frecuencia, pero desafortunadamente, sin el mismo significado para todos los que la usan. En la mayoría de los casos se utiliza para denotar la propiedad de la existencia de relaciones entre diferentes registros lógicos. La habilidad de definir relaciones entre registros a nivel lógico y en forma tan conveniente como se definen los registros, es un objetivo principal de la tecnología de bases de datos.

Cuando muchos usuarios diferentes comparten los mismos datos y además están involucradas todas las relaciones, es imposible que cada usuario sea responsable de la consistencia de los valores en la base de datos, y de mantener las relaciones de los

organizaciones lógicas o físicas específicas, y respecto a las consideraciones del almacenamiento de las estructuras de datos y archivos que ese programa use.

El concepto de la *habilidad para compartir datos* es la idea de poseer la capacidad de que varios programas de aplicación compartan en forma independiente una base de datos.

La manera tradicional de proceder, específicamente el recopilar y codificar datos para programas específicos y por tanto "amalgamar" los datos en forma más o menos permanente y exclusiva a esos programas, ha ocasionado la costosa redundancia de registros y archivos en los sistemas de aplicación implicados. La eliminación de la redundancia conduce a la posibilidad de compartir. La redundancia resulta cuando dos aplicaciones diferentes requieren el mismo dato elemental y no lo comparten. Sin embargo, puede ser necesaria cierta redundancia para evitar un tiempo de acceso excesivo a los registros. Es verdad que las redundancias conllevan los problemas de mantenimiento para asegurar que la información redundante sea consistente.

La *habilidad para relacionar* es un término que se usa con mucha frecuencia, pero desafortunadamente, sin el mismo significado para todos los que la usan. En la mayoría de los casos se utiliza para denotar la propiedad de la existencia de relaciones entre diferentes registros lógicos. La habilidad de definir relaciones entre registros a nivel lógico y en forma tan conveniente como se definen los registros, es un objetivo principal de la tecnología de bases de datos.

Cuando muchos usuarios diferentes comparten los mismos datos y además están involucradas todas las relaciones, es imposible que cada usuario sea responsable de la consistencia de los valores en la base de datos, y de mantener las relaciones de los

datos elementales que los usuarios visualizan respecto a todos los demás datos elementales, algunos de los cuales pueden serles desconocidos o estarles vedados para acceso. Un DBMS como tarea principal tiene el de mantener el control y preservar la integridad de la base de datos. El término integridad ha evolucionado para referirse a la coordinación del acceso de datos por programas distintos, a la propagación de valores actualizados a otras copias y valores dependientes, y asegurar la validez de los datos. La conservación de la integridad también incluye el mantener una guía de auditoría, esto es, una bitácora o registro de todos los accesos y cambios que afecten a cada dato elemental en el momento de tiempo en el que tenga lugar un acceso o un cambio, así como de la interacción de programas y datos, de manera que pueda recuperarse la integridad si se detecta un error posteriormente.

La *Flexibilidad de acceso* se refiere a la propiedad de la capacidad de tener acceso a los datos fácil y eficientemente de diversas maneras. Es un término que incluye: la habilidad de tener acceso a los datos.

Un sistema DBMS debe tener los mecanismos adecuados para la asignación, control y revocación de derechos de acceso (leer, cambiar, insertar, eliminar) a cualquier usuario de cualquier dato elemental. El sistema debe garantizar la seguridad de los datos. Ciertos artículos y combinaciones o selecciones de ellos, pueden ser sensibles y requerir de altos niveles de autorización para que se permita su acceso.

EL buen rendimiento y la eficiencia de un sistema soportado para un DBMS son los requisitos principales debido al gran tamaño de estos. Cada facilidad que deba implementarse para dar soporte a los objetivos que se han fijado, es una carga adicional para el diseño, organización, implementación y rendimiento del DBMS.

3.2 TIPOS DE BASES DE DATOS

La arquitectura de tres niveles del ANSI-SPARC para bases de datos centralizadas ha ganado gran aceptación y muchos DBMS comerciales se han unido a esta arquitectura. una de las principales razones para la confusión en la clasificación de los DBMS es que se agrupan de acuerdo a diferentes criterios.

Heterogeneidad en los DBMSs pueden surgir en diferentes niveles en el sistema, incluyendo hardware diferente, sistemas operativos diferentes y diferentes tipos de protocolos de red. La heterogeneidad en las bases de datos puede surgir cuando existen diferentes DBMS locales basados en el mismo modelo de datos (ejemplo INGRES o Oracle) o basados en diferentes modelos (relacional, jerárquico o de red). Las bases de datos centralizadas son acosadas con problemas que no pueden ser resueltos sin una revisión básica de conceptos de compartición de datos.

Los diferentes tipos de bases de datos ofrecen una variedad de interfaces a través de las cuales se puede manejar la base de datos. Además proporcionan un lenguaje de definición (DDL) para describir los datos que estarán almacenados en la base de datos. Los lenguajes de Query son sumamente importantes para las bases de datos. Los lenguajes vienen en muchas formas, desde llamadas especiales a procedimientos para manejo de la base de datos desde lenguajes 3GL, poderosos 4gls, lenguajes de propósito general y el SQL de propósito específico. Los diferentes tipos de bases de datos más utilizados hoy en día utilizan SQL como interface para acceso a los datos. Sql es un poderoso lenguaje que ofrece muchas características que permiten no solo

almacenar sino actualizar datos, también permiten la creación de vistas entre otras. Existe un standard internacional para el SQL especificado por ANSI. Como muchos estándares, el estándar de 1986 ha sido sujeto a muchas críticas y muchos productos relacionales ofrecen características adicionales que no son parte del estándar. Algunos de las críticas más utilizadas al estándar SQL son :

- No soporte a integridad referencial (Hoy en día ya soportado por muchos fabricantes de DBMS);
- Un método poco consistente para manipulación de actualizaciones a través de vistas;
- No existe una forma fácil de almacenar resultados de consultas para un procesamiento subsecuente.

Para que una base de datos distribuida multiusuario cumpla con los objetivos de su creación, debe cumplir con las siguientes reglas:

- Autonomía local
- No debe depender de un nodo centralizado
- Debe operar continuamente
- Debe ser independiente de su localización
- Debe ser independiente de la fragmentación
- Debe ser independiente en la generación de réplicas
- Debe soportar el procesamiento de queries distribuidos

- Debe soportar el manejo de transacciones distribuido
- Debe ser independiente del hardware en que corra
- Debe ser independiente del sistema operativo
- Debe ser independiente de la red

ARQUITECTURAS DE BASES DE DATOS

El manejo y organización de la información en una base de datos puede ser vista en 4 niveles de Arquitectura que van desde el nivel de Organización Lógico hasta su correspondiente en el nivel de Organización Físico, desde el más independiente del equipo hasta el más dependiente.

En el nivel 4, el más alto, es donde se hacen sólo las consideraciones lógicas de Datos, donde se hace el análisis de necesidades del usuario, de los datos con que cuenta y los que necesita, se especifica el flujo de la información y sus transformaciones. Posteriormente se definen entidades u objetos y sus relaciones funcionales.

A partir de éstas entidades y relaciones funcionales se construye o modela la base utilizando los modelos lógicos de organización. Cabe mencionar que éste nivel es el más independiente del computador donde se va a implementar la base de datos. El siguiente nivel, se orienta a la implementación de los modelos de organización, se abordan acciones orientadas al desempeño, se empiezan a tomar en cuenta las limitaciones de los dispositivos físicos. Aquí se definen las estrategias de acceso para mejorar el desempeño en el manejo de datos (consulta, actualización). Por ejemplo, en

este nivel se decide qué organizaciones de archivo se utilizarán, si son de acceso secuencial, random ó por organizaciones secundarias (listas invertidas, multilistas, etc.) y sus posibles variaciones.

A los niveles 3 y 4 se les llama interface física lógica. El nivel 2 refleja alternativas de organización física y formatos de almacenamiento de datos en dispositivos de almacenamiento lineal para cada una de las organizaciones definidas en el nivel 3.

Algunos autores sólo reconocen 3 niveles y manejan como uno solo al 3 y al 2.

El nivel 1, es más independiente que los anteriores del equipo donde se implementa la Base de Datos, se refiere a las características y aspectos particulares de los dispositivos físicos de almacenamiento. En este nivel se ven aspectos como: la organización de los dispositivos en subdivisiones tales como bloques y sectores; el mapeo de datos dentro de estas subdivisiones; el almacenamiento de áreas de sobreflujo y características dependientes de cada máquina, así como la localización de datos de control, verificación de integridad, etc.

3.3 MODELOS DE BASES DE DATOS, VENTAJAS Y DESVENTAJAS

Una entidad (bases de datos) es llamada un elemento de datos (Data Element), a un conjunto de entidades de datos se les llama registros y a la colección de registros semejantes se les nombra archivos.

Para hacer posible la administración de archivos de registros, un sistema administrador de base de datos asume la siguiente regla: Todos los registros semejantes toman una forma común.

Dado un archivo de registros, los cuales presentan características semejantes, un sistema administrador de base de datos puede adicionar registros al archivo, recuperar registros del archivo, modificar registros del archivo, y eliminar registros del archivo.

Un registro consiste de elementos de datos, un elemento de dato es una entidad particular de datos que pueden tomar un valor que tiene un significado funcional para un registro en particular.

Un elemento de datos puede ser el nombre, dirección, fechas, promedios, etc., característicos de una entidad. Un elemento de datos puede consistir de uno o más tipos de datos dentro de una estructura.

Al conjunto de archivos que soportan una aplicación en particular se le llama Base de Datos.

El destino de la información de una organización consiste en un número independiente de base de datos. Algunas bases de datos tienen puntos de interface con otras. El

software que administra una base de datos es llamado un sistema administrador de base de datos.

El propósito de un sistema administrador de base de datos es almacenar, mantener y recuperar registros de la base de datos en archivos. Un archivo contiene registros con el mismo formato que sirven para un propósito en común.

Un sistema administrador de base de datos puede optimizar el código de una aplicación, administrando el acceso a la base de datos.

Tomando en cuenta el diseño de las bases de datos existentes y considerando las características comunes de la mayoría de las aplicaciones, se creó software administrador de base de datos de propósito general.

El propósito de los sistemas administradores de las bases de datos de propósito general se encuentra bajo la definición de ciertos modelos de base de datos.

Los sistemas administradores de bases de datos de propósito general utilizan cuatro lenguajes de interface entre el lenguaje de programación de la aplicación y el administrador de la base de datos: El lenguaje de definición de datos (The data definition language), el lenguaje de administración de datos (The data Manipulation Language), el lenguaje de consultas (The query language), y el lenguaje generador de reportes (The report writer language).

LENGUAJE DE DEFINICIÓN DE DATOS

El Lenguaje de definición de datos (DDL) define el formato (o esquema) de la base de datos. Identifica a los archivos, da formato a los registros y las relaciones entre los archivos. El DDL es comunmente un archivo externo de definiciones que genera la utileria constructora de la base de datos utilizada para definirla. Así mismo, el administrador de la base de datos utiliza al DDL para especificar el formato de los archivos y registros para soportar llamadas del administrador de datos y el lenguaje de consultas.

LENGUAJE DE ADMINISTRACIÓN DE DATOS

El lenguaje de administración de datos (DML) es una interface entre los programas de aplicación y el sistema administrador de la base de datos. Este proporciona el programa de interface para abrir y cerrar la base de datos, realizar búsquedas de registros en archivos, navegar a través de los registros de un archivo, agregar nuevos registros, y modificar o eliminar registros existentes.

EL LENGUAJE DE CONSULTA

El lenguaje de consulta es una técnica de propósito general para realizar búsquedas dentro de la base de datos que cumplan con un criterio de selección. Algunos lenguajes de consulta son procesados por programas de utilerias, y otros son

implementadas como extensiones de lenguajes de programación de aplicaciones. Una consulta retorna generalmente un conjunto temporal de registros de la base de datos. El conjunto de datos puede después tener su propio formato independiente del formato de los registros definidos en la base de datos, y algunas veces los registros consultados pueden consistir de registros formados por la combinación lógica de registros relacionados en la base de datos.

EL LENGUAJE GENERADOR DE REPORTES

Algunos sistemas administradores ofrecen un programa de utilería generador de reportes que se utiliza para generar la salida de una consulta generada por un reporte. El lenguaje generador de reportes especifica el formato de la presentación de los valores de los datos en un reporte.

MODELOS DE BASES DE DATOS

MODELO DE DATOS JERÁRQUICO

En este modelo se encuentran las bases de datos comprendidas por archivos con una relación jerárquica. Por ejemplo:

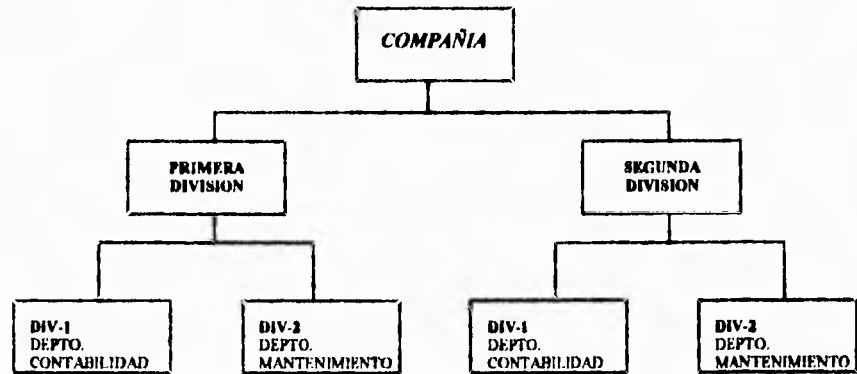


Fig. 3.3.1 Modelo Jerárquico.

Un archivo principal en la jerárquica es el archivo padre que contiene registros padres. Los archivos subordinados contienen registros hijos. Un registro hijo puede tener sus propios hijos, estos pueden ser hijos de un nivel, o bien, padres de otro. En este modelo un registro tipo hijo puede tener sólo un registro tipo padre, y cada instancia de registro hijo puede tener sólo una instancia de registro padre.

Puede considerarse al modelo como un árbol de segmentos con un nivel de tipos en el que un tipo raíz posee n hijos, que a su vez, poseen cada uno, varios tipos hijos. De estos últimos, algunos son de un primer tipo, otros de un segundo, ...etc. A su vez, las ocurrencias de los hijos pueden tener otros hijos.

Los sistemas administradores de base de datos jerárquicos soportan la relación entre los archivos que la forman mediante apuntadores físicos a los registros de la base de datos. Cada registro de división podría tener la dirección en disco de su padre (compañía), así como la dirección de la siguiente división que forma parte de la compañía.

implementadas como extensiones de lenguajes de programación de aplicaciones. Una consulta retorna generalmente un conjunto temporal de registros de la base de datos. El conjunto de datos puede después tener su propio formato independiente del formato de los registros definidos en la base de datos, y algunas veces los registros consultados pueden consistir de registros formados por la combinación lógica de registros relacionados en la base de datos.

EL LENGUAJE GENERADOR DE REPORTES

Algunos sistemas administradores ofrecen un programa de utilería generador de reportes que se utiliza para generar la salida de una consulta generada por un reporte. El lenguaje generador de reportes especifica el formato de la presentación de los valores de los datos en un reporte.

MODELOS DE BASES DE DATOS

MODELO DE DATOS JERÁRQUICO

En este modelo se encuentran las bases de datos comprendidas por archivos con una relación jerárquica. Por ejemplo:

CARACTERÍSTICAS

- Se representa a los datos como estructuras de árbol.
- El árbol representa una jerarquía de registros de datos.
- Procesamiento Top-Down, navegacional.
- Términos Jerárquicos:
 - SET (DATA SET): registro.
 - PARENT : registro padre.
 - CHILD: registro hijo.
- Relaciones entre registros: 1 padre, múltiples hijos.

DESVENTAJAS

- No permite el modelado sencillo de las relaciones **N a M**.
- Presenta anomalías de inserción.
- Presenta anomalías de borrado.
- Presenta Anomalías de actualización.
- Asimetría en la consulta.

MODELO DE DATOS EN RED

La desventaja del modelo de datos jerárquico de no soportar las relaciones entre los archivos cuando un registro hijo tiene más de un padre dió origen al modelo de datos en red. Si dentro de la organización de la figura 3.3.1 se determinara crear un departamento de contabilidad que soportara más de una división, por ejemplo, el modelo de datos jerárquico no podría soportar ésta relación. Mediante el modelo de datos jerárquico, no se pueden manejar eficientemente aplicaciones que requieren múltiples registros padres para un registro hijo. El modelo de red cubre esta desventaja.

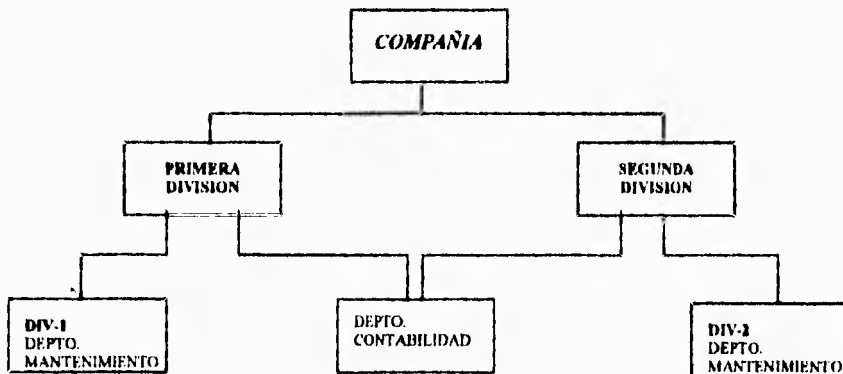


Fig. 3.3.2 Modelo de Base de Datos en Red

Las bases de datos en red fueron principalmente implementadas con apuntadores a las direcciones de los registros, tal como en el modelo de datos jerárquico, excepto que más de un registro padre puede apuntar a un registro hijo en particular, y el registro hijo apunta a una lista de padres.

CARACTERÍSTICAS

- Se representan los datos como registros ligados formando un conjunto de datos intersectados.
- Procesamiento multidireccional, navegacional.
- Términos de red:
 - Esquema (schema)
 - Subesquema (subscheme)
- Cualquier tipo de relación entre registros puede ser modelada.

DESVENTAJAS

- Se presenta dificultad al definir nuevas relaciones.
- Difícil proporcionar mantenimiento (cualquier cambio requiere una descarga de los datos).
- Gran cantidad de overhead (desperdicio de recursos).

MODELO DE DATOS RELACIONAL

Dentro del modelo de datos jerárquico y el modelo de datos de red se maneja apuntadores físicos ocultos en los registros de los datos, lo que originaba que no se asegurara la integridad de los datos, por lo tanto resultaban fácilmente inestable: Errores dentro del sistema originaban modificaciones en los apuntadores, esto ocasionaba que las relaciones entre los registros se perdieran.

Dentro del modelo de datos relacional, las relaciones entre los archivos se representan por valores de datos dentro de los registros a diferencia de los apuntadores físicos.

Los archivos dentro de una base de datos relacional presentan una longitud fija, y el formato de los registros es fijo, los que forman tablas de renglones y columnas. Cada registro en un archivo tiene un elemento de datos llamado llave primaria, que representa un identificador único del registro.

Las relaciones dentro del modelo de datos relacional se representan mediante valores de los datos dentro de los registros. Un registro de departamento contiene un identificador de la división a la cual pertenece. Al elemento de dato que funciona como un identificador de la división dentro de un registro de departamento es el elemento de datos denominado llave secundaria, mediante la cual es posible obtener el conjunto de registros de departamentos que se encuentran identificados en un registro de división en particular.

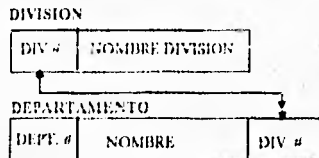


Fig. 3.3.3 : Base de Datos Relacional

En la figura 3.3.3 se muestra la relación establecida por el número de división dentro del registro de departamento con el registro de división. Ya que como el número de división es la llave primaria del registro de división, diferentes registros de departamento pueden ser relacionados a un registro de división en particular, y sólo un registro de división puede ser relacionada a cualquier registro de departamento. A ésta característica se le llama relación uno a muchos, dentro de la figura ésta relación se muestra mediante una flecha y la doble punta de flecha al final de las entidades.

La relación uno a muchos entre las divisiones y los departamentos no soportan el problema donde un departamento pueda ser asignado a varias divisiones, este problema se resuelve utilizando archivos que funcionan como conectores. Esto se muestra en la siguiente figura, es la relación muchos a muchos entre los registros de datos.

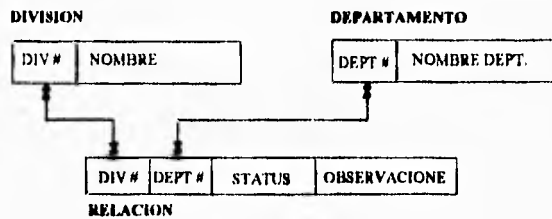


Fig. 3.3.4 Relación muchos a muchos.

La concatenación del número de división y el número de departamento forma la llave primaria del archivo que funciona como intersección. Cada uno por separado forman llaves secundarias.

El modelo relacional tiene asociada una teoría que no puede ser separada del modelo: la teoría de la normalización de las relaciones. Esta teoría tiene por objeto eliminar los comportamientos anormales de las relaciones durante las actualizaciones. También permite eliminar los datos redundantes y facilita la comprensión de las relaciones semánticas entre los datos.

CARACTERÍSTICAS

- **Tabla:**

- Arreglo de dos dimensiones.

- Consiste de renglones y columnas.

- Colección de registros.

- **Columna:**

Contiene un mismo tipo de información.

Renglón:

Contiene un mismo conjunto de información.

- **Campo:**

Contiene un valor en la intersección del renglón y la columna.

- **La información en una tabla puede estar relacionada con la información de otra tabla.**

VENTAJAS

- Simplicidad
- Fácil de usar
- Fácil obtener respuestas
- Fácil insertar y actualizar datos
- Fácil cambiar la estructura de los datos
- Todas las consultas son posibles.

MODELO DE DATOS ORIENTADO A OBJETOS

Una base de datos tradicional sólo almacena datos, sin procesarlos, de modo que resulten independientes de los procedimientos. Los datos son accesibles a diferentes usuarios, con diversos propósitos. Por lo contrario, una base de datos orientada a

objetos almacena objetos. Los datos se almacenan junto con los métodos que procesa dichos datos. Dentro de las bases de datos orientadas a objetos no se tiene acceso a ningún dato si no es a través de los métodos almacenados en la base de datos. Estos métodos están listos para ejecutarse al momento en que reciben una solicitud. Los datos de todos los objetos quedan entonces encapsulados. En general los datos son activos más que pasivos.

Varias de las necesidades no cubiertas por las bases de datos relacionales convergieron para formar una nueva generación de sistemas para la administración de los datos. Las bases con conocimiento se orientaron a los marcos, donde el marco es un objeto con un conjunto de reglas asociadas con él. Los tipos de datos abstractos (tipos de datos definidos por el usuario), con una gran variedad, necesitaban un soporte que incluyera sonido, imágenes y video. Los datos complejos necesitaban técnicas de acceso que mejorasen el desempeño sobre las bases de datos por relación. Estas necesidades convergieron a las BDOO (Bases de Datos Orientadas a Objetos).

Las BDOO se desarrollan al describir en primer lugar los tipos de objetos importantes del dominio de aplicación y los comportamientos asociados con aquellos tipos de objetos. Estos tipos de objetos determinan las clases que conformarán la definición de la BDOO. Por ejemplo, una base de datos diseñada para almacenar la geometría de ciertas partes mecánicas incluiría clases como cylinder (cilindro), sphere (esfera) y cube (cubo). El comportamiento de cylinder podría incluir información relativa a sus dimensiones, volumen y área superficial:


```
CLASS CYLINDER {  
    FLOAT HEIGHT();  
    FLOAT RADIUS();  
    FLOAT VOLUME();  
    FLOAT SURFACEAREA();  
};
```

En el ejemplo se observa la ausencia de detalles de la implantación, aún si se almacena o calcula la información.

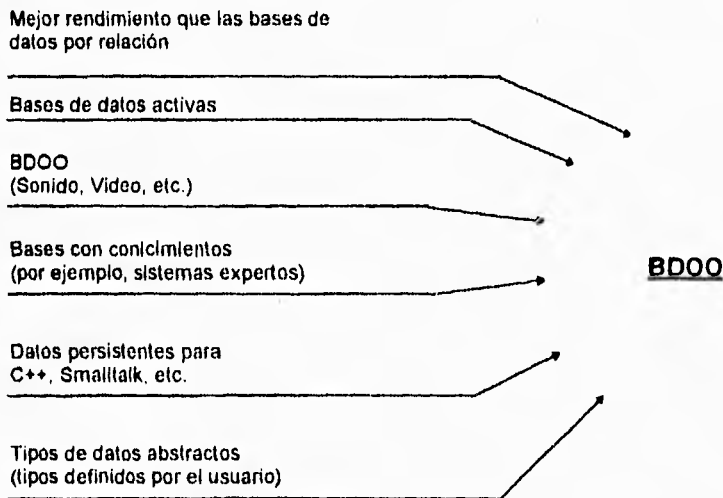


Fig. 3.3.5 Una gama de necesidades convergió en la tecnología de las BDOO.

Este enfoque da flexibilidad a la BDOO. Puesto que toda la aplicación se escribe mediante el envío de solicitudes, las implantaciones se pueden alterar en cualquier grado sin afectar a las aplicaciones. En consecuencia, la aplicación es más sencilla y

Este enfoque da flexibilidad a la BDOO. Puesto que toda la aplicación se escribe mediante el envío de solicitudes, las implantaciones se pueden alterar en cualquier grado sin afectar a las aplicaciones. En consecuencia, la aplicación es más sencilla y más fácil de mantener y mejorar. En este sentido, las BDOO ofrecen una separación aún mayor entre la especificación de un sistema y su implantación.

Uno de los principales objetivos de la tecnología orientada a objetos es el encapsulado. Esto significa que los datos sólo pueden ser utilizados con los métodos que forman parte de una clase. Se pretende que las clases orientadas a objetos sean reutilizables. Por lo tanto, otro objetivo de la tecnología OO (Orientada a Objetos) es lograr la máxima reutilización. Debido a esto, la clase debe estar libre de errores y sólo debe modificarse en un caso absolutamente necesario.

La tecnología OO utiliza la herencia para reducir el desarrollo redundante de los métodos. También crea clases diseñadas para utilizarse otra vez en muchas aplicaciones. El encapsulado y la herencia reducen la cantidad de código y datos redundantes, de dos maneras: mediante la propagación y la reutilización de las clases. Esto también reduce el costo del desarrollo y el mantenimiento.

3.4 NORMALIZACIÓN DE BASES DE DATOS

El proceso de normalización es utilizado para verificar las relaciones de los archivos de a base de datos y reconocer los elementos de datos diseñados en los diferentes pasos para eliminar la redundancia, apuntadores internos del archivo y grupos repetidos de elementos de datos en los registros.

Las tres primeras formas normales tienen por objetivo permitir la descomposición de relaciones sin pérdida de información, a partir del concepto de dependencia funcional.

PRIMERA FORMA NORMAL.- La entidad debe ser expresada como un conjunto no ordenado de tablas de dos dimensiones. La tabla no debe contener grupos repetidos.

ORDER ID	DATE	CUSTOMER ID	CUSTOMER ID	STATE	ITEM NUM	ITEM DESCRIP	QUANTITY	PRICE
2301	6/23	101	Volleyrite	IL	3786	net	3	35.00
					4011	racket	6	65.00
					9132	3-pack	8	4.75
2302	6/25	107	Herman's	WI	5794	6-pack	4	5.00
2303	6/26	110	We-R-Sports	MI	4011	racket	2	65.00
					3141	cover	2	10.00

Fig. 3.4.1 Tabla no normalizada

Si se considera la tabla anterior de valores, se puede observar que existen grupos de acuerdo al número de producto (ITEM NUM), a la descripción del producto (ITEM

DESCRIPTION), a la cantidad QUANTITY, y al precio (PRICE). La primera forma normal no permite los grupos repetidos.

Para normalizar en la primera forma, es necesario:

- 1) Eliminar los grupos repetidos.
- 2) Crear una nueva tabla con la llave primaria de la tabla base y el grupo repetido.

Se elimina el grupo repetido de ITEM NUM, ITEM DESCRIPTION, QUANTITY, y PRICE. Se crea una nueva tabla con el identificador único de la tabla base y con el grupo repetido.

Tabla **ORDER**

ORDER ID	DATE	CUSTOMER ID	CUSTOMER NAME	STATE
PK				
2301	6/23	101	Volleyrite	IL
2302	6/25	107	Herman's	WI
2303	6/26	110	We-R-Sport	MI

Tabla **ORDER ITEM**

ORDER ID	ITEM NUM	ITEM DESCRIP	QUANTITY	PRICE
PK,FK		PK		
2301	3786	net	3	35.00
2301	4011	racket	6	65.00
2301	9132	3-pack	8	4.75
2302	5794	6-pack	4	5.00
2303	4011	racket	2	65.00
2303	3141	cover	2	10.00

Fig 3.4.2 Tablas normalizadas.

SEGUNDA FORMA NORMAL .- La tabla debe encontrarse en la primera forma normal.

Todas las columnas (atributos) que no formen parte de la llave, sólo deben depender de la llave primaria.

Para pasar una tabla a la segunda forma normal se debe:

- 1) Determinar cuales columnas que no forman parte de la llave, no dependen de la llave primaria de la tabla completamente.
- 2) Eliminar estas columnas de la tabla.
- 3) Crear una segunda tabla con estas columnas y las columnas que forman parte de la llave primaria de las cuales depende.

Ejemplo:

Tabla **ORDER ITEM**

ORDER ID	ITEM NUM	ITEM DESCRIP	QUANTITY	PRICE
PK,FK	PK			
2301	3786	net	3	35.00
2301	4011	racket	6	65.00
2301	9132	3-pack	8	4.75
2302	5794	6-pack	4	5.00
2303	4011	recket	2	65.00
2303	3141	cover	2	10.00

Fig. 3.4.3 Tabla sin normalizar.

Al observar la tabla mostrada es posible notar que las columnas PRICE y DESCRIPTION no dependen de la llave primaria de la tabla. Para colocar la tabla en la segunda forma normal, se eliminan parcialmente las columnas no dependientes. Se

crea una tabla con estas columnas y la columna de la llave primaria de la cuál dependen.

ORDER ID	ITEM NUM	QUANTITY
PK,FK	PK,FK	
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

ITEM NUM	ITEM DESCRIP	PRICE
PK		
3786	net	35.00
4011	racket	65.00
9132	3-pack	4.75
5794	6-pack	5.00
3141	cover	10.00

Fig. 3.4.4 Tablas en la segunda forma normal.

TERCERA FORMA NORMAL .- La tabla debe estar en la segunda forma normal. Ninguna columna que no forme parte de la llave primaria debe ser funcionalmente dependiente de cualquier otra columna que no forme parte de la llave primaria.

Para colocar una tabla en la tercera forma normal debe realizarse lo siguiente:

- 1) Determinar cuales columnas dependen de otras que no forman parte de la llave primaria.
- 2) Eliminar estas columnas de la base de datos.
- 3) Crear una segunda tabla con estas columnas y con las que depende.

Ejemplo:

Tabla **ORDER**

ORDER ID	DATE	CUSTOMER ID	CUSTOMER NAME	STATE
PK				
2301	6/23	101	Volleyrite	IL
2302	6/25	107	Herman's	WI
2303	6/26	110	We-R-Sports	MI

Fig. 3.4.5 Tabla sin normalizar

Las columnas CUSTOMER NAME y STATE dependen de la columna CUSTOMER ID. CUSTOMER ID no es la llave primaria. Se coloca la columna que no es llave primaria como las columnas que dependen de ella dentro de una nueva tabla:

Tabla **ORDER**

ORDER ID	DATE	CUSTOMER ID
PK		FK
2301	6/23	101
2302	6/25	107
2303	6/26	110

Tabla **CUSTOMER**

CUSTOMER ID	CUSTOMER NAME	STATE
PK		
101	Volleyrite	IL
107	Herman's	WI
110	We-R-Sports	MI

Fig. 3.4.6 Tablas en la tercera forma normal.

FORMA NORMAL DE BOYCE-CODD.- Una relación está en BCNF (Boyce y Codd) si y solo si las únicas dependencias funcionales elementales son aquellas en las que una llave determina un atributo.

Ejemplo

Considerando la siguiente relación:

VINOS(caldo,pais,región)

donde la llave primaria es la combinación de los atributos <caldo y región>, esta relación está en tercera forma normal porque ningún atributo no clave depende de una parte de la clave o de un atributo no clave. No obstante presenta numerosas redundancias.

Tabla VINOS

CALDO	PAIS	REGION
Chenas	Francia	Beaujolais
Rioja	España	La Rioja
Vaidepeñas	España	Ciudad Real
Jerez	España	Málaga
Chablis	Estados Unidos	California

Fig. 3.4.7 Tabla correspondiente a la relación VINOS.

La relación VINOS se puede descomponer en dos relaciones CALDOS(caldo,región) REGIONES(región,pais).

DEPENDENCIAS MULTIVALUADAS Y CUARTA FORMA NORMAL.- El concepto de dependencia funcional lleva a descomponer las relaciones en tercera forma normal y en forma normal de Boyce-Codd. Esto es insuficiente para eliminar las redundancias y las anomalías de las actualizaciones. Si se considera la siguiente relación:

ESTUDIANTE(ne,curso,deporte)

donde "ne" es el número de estudiante, "curso" los cursos a los que asiste y "deporte" los deportes que practica.

Tabla ESTUDIANTE

NE	CURSO	DEPORTE
100	Bases de datos	Tenis
100	Bases de datos	Fútbol
200	Bases de datos	Vela
200	Matemáticas	Vela

Fig. 3.4.8 Relación en tercera forma normal con redundancias.

Se puede observar que aparecen redundancias en esta relación y, debido a la ausencia de dependencias funcionales, no se le puede descomponer.

Con este ejemplo es posible observar que el concepto de dependencia funcional no es suficiente y que no permite captar la independencia que existe entre atributos tales como curso tomado y deporte practicado. Por este motivo se genera el concepto de dependencia funcional introduciendo el de dependencia multivaluada.

DEPENDENCIA MULTIVALUADA .- Sea $R(A_1, A_2, \dots, A_n)$ un esquema de relación y X e Y unos subconjuntos de A_1, A_2, \dots, A_n . Se dice que $X \twoheadrightarrow Y$ (X multidetermina a Y , o existe una dependencia multivaluada de Y sobre X) si dados unos valores de X existe un conjunto de valores de Y asociados y este conjunto es independiente de los otros atributos, $Z = R - X - Y$ de la relación R .

Capítulo III Bases de Datos y Características de la Programación Orientada a Objetos.

DEPENDENCIA MULTIVALUADA ELEMENTAL .- Una dependencia multivaluada elemental tiene, a la vez un lado derecho y un lado izquierdo mínimo, como una dependencia funcional elemental. Ejemplo: Si se presenta la relación:

PERSONA(No.SS,nombre-niño,No. vehículo)

se pueden observar las dependencias multivaluadas elementales:

No.SS ->> nombre-niño

No.SS ->> No.vehículo

La cuarta forma normal es una generalización de la forma normal de Boyce-Codd para descomponer las relaciones que tengan Dependencia Multivaluada elementales.

CUARTA FORMA NORMAL .- Una relación está en cuarta forma normal sí y sólo si las únicas dependencias multivaluadas son aquéllas en las que una clave determina un atributo.

Debido a que una dependencia funcional es un caso particular de dependencia multivaluada, resulta que una relación en cuarta forma normal está en forma normal de Boyce-Codd y, por consiguiente, en tercera forma normal.

Como ejemplo, la relación ESTUDIANTE(ne,curso,deporte) no se encuentra en cuarta forma normal: la clave es el conjunto de atributos y existen las DM elementales:

ne->>curso

ne->>deporte

entre atributos que forman parte de la clave.

Como toda relación tiene una descomposición (no forzosamente única) en cuarta forma normal que se realiza sin pérdida, la relación ESTUDIANTE se puede descomponer en dos relaciones (ne, curso) y (ne, deporte) que está en cuarta forma normal.

DEPENDENCIAS DE REUNIONES .- Si se considera la siguiente relación:

Tabla VINOS

BEBEDOR	CALDO	PRODUCTOR
Ramón	Chablis	Claudio.
Ramón	Chablis	Nicolás
Ramón	Montilla	Nicolás
Jesús	Chablis	Nicolás

Fig. 3.4.9 Relación en cuarta forma normal con redundancia.

En la cual se modela unos vinos consumidos por unos bebedores, de un caldo dado y servidos por un productor encargado de producir caldo. Esta relación está en cuarta forma normal. No obstante la relación presenta bastantes redundancias: indica dos veces que Ramón bebe Chablis y que Nicolás produce Chablis. Es una relación que no se puede descomponer en dos relaciones.

El ejemplo anterior muestra que el concepto de dependencia multivaluada no es suficiente para eliminar las redundancias. El problema es debido a que siempre se intenta descomponer una relación en solamente dos relaciones: así el concepto de dependencia multivaluada incorpora la posibilidad de descomponer una relación en dos; la relación (XYZ) en la que $X \twoheadrightarrow Y$ está de hecho, descompuesta en (XY) y (XZ) ya que Y y Z son independientes con respecto a X. Existen relaciones que no se

pueden descomponer en dos, pero sí en tres, cuatro ó, más generalmente, en n relaciones.

BEBEDOR	CALDO
Ramón	Chablis
Ramón	Chablis
Jesús	Chablis

BEBEDOR	PRODUCTOR
Ramón	Claudio
Ramón	Nicolás
Jesús	Nicolás

CALDO	PRODUCTOR
Chablis	Claudio
Chablis	Nicolás
Montilla	Nicolás

Figura 3.4.10 Tres Proyecciones de la relación VINOS

QUINTA FORMA NORMAL .- La quinta forma normal es una generalización de la cuarta a partir del concepto de dependencia de reunión. Una relación está en quinta forma normal si y sólo si toda dependencia de reunión viene implicada por las claves candidatas de R. Así la relación VINOS no está en quinta forma normal porque la única clave candidata (bebedor,caldo,productor) no implica la DR * {bebedor caldo, caldo productor, bebedor productor}. Así pues, debe estar descompuesta en estas tres relaciones para evitar las anomalías de las actualizaciones.

3.5 CARACTERÍSTICAS GENERALES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos, es una nueva forma de pensar cómo resolver algunos de los problemas planteados en el mundo de la informática. En lugar de tratar de adaptar el problema a algún aspecto familiar a la computadora, la computadora se adapta al problema. En la programación orientada a objetos se examina el problema en entidades independientes que se relacionan con otras partes del problema. Las entidades son representadas como objetos dentro del programa para la computadora. El objetivo es tener una correspondencia, una a una, entre entidades en el problema físico y objetos en el programa.

LA EVOLUCIÓN DE LOS LENGUAJES ORIENTADOS A OBJETOS

Se ha sugerido que la programación orientada a objetos, surgió como resultado del caos que ocurrió cuando los lenguajes tradicionales trataron de enfrentarse a los grandes problemas. A medida que se van desarrollando los lenguajes, se va desarrollando también la facilidad de resolver problemas cada vez más complejos. En la evolución de cada lenguaje, llega un momento en que los programadores comienzan a tener dificultades a la hora de manejar programas que sean de un cierto tamaño y sofisticación.

Los primeros lenguajes de programación, se diseñaron para minimizar los errores generados en la traducción de la programación orientada a usuario con su

representación en código de máquina. Estos lenguajes utilizan palabras similares al inglés , y permiten la sustitución de posiciones de memoria con nombres elegidos por el programador . Este soporte para los dispositivos mnemotécnicos conduce a una asociación mental , que hace que se pierda más tiempo en la semántica (que significa el código) y menos tiempo en la sintaxis (cómo se ensamblan las palabras y las frases).

LOS LÍMITES DE LOS LENGUAJES FUNCIONALES

La complejidad aumenta a medida que se dispone de más recursos. Los primeros programadores pronto se dieron cuenta de la dificultad de crear programas más grandes y elaborados que funcionaran correctamente . Los lenguajes de procedimientos fueron la siguiente etapa en la evolución de lenguajes . El modelo utilizado en estos lenguajes es la caja negra. Cada parte de un programa se cambia completamente para evitar la complejidad de ésta. Estas "cajas" son funciones . En el nivel superior, el programa finalizado es un conjunto de llamadas de función.

El concepto de caja negra no modifica los datos fuera de sus límites, es decir, no genera efectos laterales, en la práctica este imperativo no es demasiado severo; las cajas negras utilizan datos fuera de sus límites lo que provoca una desventaja ya que los datos no están completamente bajo el control de la caja.

Los primeros programas escritos, no basados en el lenguaje de procedimientos, tendían a ser pequeños. De esta forma , cuando se pasó a los lenguajes de procedimientos , el código no fue muy difícil de reescribir . Estos lenguajes, sin embargo, son lo

eficientemente eficaces como para escribir programas mayores y más complicados por lo que es más difícil abandonarlos. Desgraciadamente, los diseñadores de los lenguajes de procedimientos tradicionales partieron de la base, cuando utilizaron el modelo de la caja negra, de que siempre se iba a seguir este esquema: "el mantenimiento no sería necesario. Las necesidades no cambian. Una vez que el código está corregido y funciona adecuadamente, deberá ser guardado en su caja negra". Esta suposición fue válida mientras el tamaño del proyecto fue limitado, pero empezó a crear dificultades no esperadas, a medida que aumentaba la complejidad.

Al finales de los años 60 y principios de los 70, los profesionales vieron que el límite de los lenguajes de procedimientos estaba en su complejidad de manejo. Comenzó así la llamada "Crisis del Software" que, como toda crisis, no surgió de repente. El software a medida cada vez más difícil de cambiar y modificar, y se ha ido reduciendo el número de programadores dispuestos a realizar el trabajo de mantenimiento. Los directores están preocupados porque saben que los programas solo se resolverán si se les dedica muchísimo tiempo. Sin embargo, al resultar muy caro abandonar el viejo código, los programadores se aferran a él en lugar de buscar nuevas formas de programar.

DESARROLLO ESTRUCTURADO

Los programas, realizados con lenguajes procedurales tradicionales, pueden ser difíciles de modificar y mantener. La reacción a este problema, fue forzar la estructura de los programas a través de una metodología denominada desarrollo estructurado

Las técnicas salvaron los procedimientos de lenguajes tradicionales para su empleo en grandes proyectos. Sin embargo requiere de un alto grado de prudencia y planificación para que el proyecto se ensamble rápida y correctamente, que quede libre de errores y sea fácil de mantener. El desarrollo estructurado no funciona también como parece, ya que en un intento de salvar a un grupo concreto de lenguajes sin ajustarse realmente a las verdaderas necesidades de los programadores.

Los programadores al igual que el resto de las personas, aprenden y crean a través de la experimentación y no de la planificación. Necesitan un sistema que apoye la experimentación y que salve tantos experimentos como sea posible y no uno que requiera una planificación total del proyecto antes de escribirse cualquier código.

Aunque se pudieran aprovechar algunas técnicas útiles de lenguajes estructurados no fue una 'solución mágica'. Las técnicas estructuradas fueron parte de la revolución científica. Cada vez que el sistema tradicional se iba destruyendo se intentaba por todos los medios salvarle. La siguiente etapa, tenía que ser, por tanto, la creación de un nuevo sistema que aceptará las limitaciones que el viejo sistema intentaba negar. La programación orientada a objetos es uno de esos sistemas, aunque por supuesto no es la solución definitiva. Las técnicas orientadas a objetos tienden a enfatizar el diseño incluso más que las técnicas estructuradas.

Al utilizar el lenguaje ensamblador el programador podía dejar de decodificar números y en su lugar tratar con palabras. Los lenguajes de procedimientos ocultaron la complejidad de las operaciones en los datos. A su vez, los lenguajes orientados a

objetos encubren la complejidad del programa en si mismo, así como el ocultamiento de los datos.

Un lenguaje orientado a objetos, enfatiza los tipos de datos y las operaciones intrínsecas que pueden desarrollarse en aquellos tipos de datos. En la programación orientada a objetos, los datos no fluyen libremente por el sistema, ya que están protegidos de alguna modificación accidental. Son los *mensajes*, más que los datos, los que se pueden mover en el sistema. A diferencia del enfoque de procedimientos que "invocaba una función en un dato", en el lenguaje orientado a objetos se "envía un mensaje a un objeto".

Un lenguaje orientado a objetos apoya la experimentación de dos formas:

- Los objetos son paquetes compactos que no se rompen simplemente porque se añadan otros objetos. El programador puede introducir nuevos objetos y codificarlos, sabiendo que los errores del nuevo código serán aislados del resto del sistema.
- Los tipos de objetos nuevos, pueden derivarse de los viejos. Esto ahorra al programador tiempo y permite exploraciones más rápidas. También localiza errores en el código para el tipo de objeto derivado, ya que se supone que el tipo original funciona correctamente.

QUE ES UN OBJETO ?

En la programación orientada a objetos las plantas, árboles y libros, se conocen en general como objetos. Un objeto contiene en una sola entidad y con un único nombre

las estructuras de datos y las acciones (procedimientos y las funciones) que actúan sobre este. Ver Fig.3.5.1.

Una clase (objeto en general) es una plantilla o modelo que define los datos y las funciones que actúan sobre esos datos (llamados métodos). Por ejemplo, una clase puede describir las características fundamentales de un empleado de una empresa de automóviles (nombre, título, salario, cargo, departamento al que pertenece, etc..), mientras que un objeto representará un ejecutivo específico (Jaime Ortega, Ingeniero, 6500.00, Supervisor, etc..).

Cuando un elemento dato es miembro de una clase se llama objeto. Se pueden asemejar las clases a los tipos de datos definidos por el usuario en el lenguaje de programación tradicional, ya que los tipos definidos por las clases se comportan como los tipos incorporados a un lenguaje de programación.

La programación orientada a objetos encierra la reutilización de código en vez de la reinención del mismo lo cual permite que el desarrollo del software y el mantenimiento del mismo sea más rápido.

La programación orientada a objetos permite crear software que puede ser fácilmente comprendido y compartido con otros. Las funciones contenidas en un objeto son usualmente llamadas *métodos*. En una programación orientada a objetos los datos y procedimientos son agrupados para formar objetos que se pueden unir para crear más objetos.

En general un objeto tiene asociado:

Un conjunto de variables que contienen los datos del objeto. El valor de cada variable es un objeto.

Un conjunto de mensajes a los que el objeto responde.

Un método es una porción de código para implementar cada mensaje. Un método devuelve un valor como respuesta al mensaje.

El término mensaje dentro de la programación orientada a objetos no implica el uso de un mensaje físico en una red de computadoras, sino que se refiere al paso de solicitudes entre objetos.

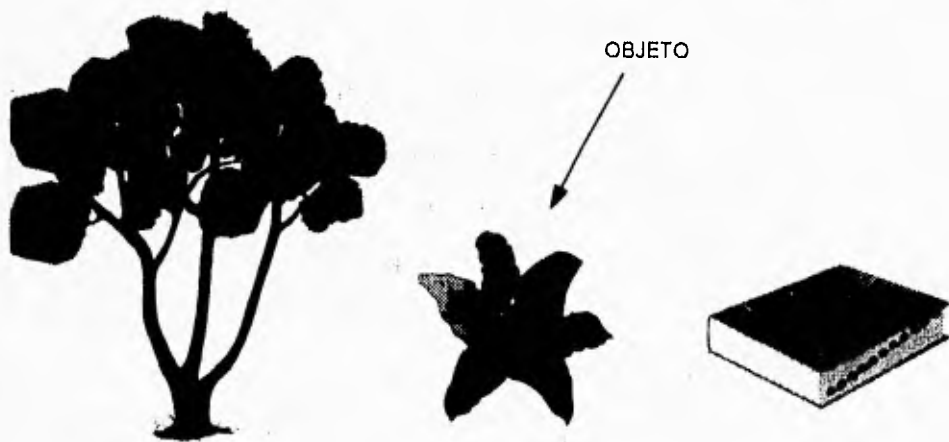


Fig 3.5.1. Objeto

Los cinco conceptos más importantes dentro de la programación orientada a objetos son:

- Abstracción de datos
- Encapsulación
- Herencia
- Polimorfismo
- Reutilización

ABSTRACCIÓN DE DATOS

La abstracción fue un concepto introducido por la programación estructurada. Se define como la capacidad para examinar algo sin preocuparse de sus detalles internos. En un programa estructurado, es suficiente conocer la tarea específica que hace un procedimiento dado, y no es tan importante cómo se realiza esa tarea. Esta propiedad se conoce como abstracción funcional. Cuando este fenómeno ocurre con los datos se le conoce como abstracción de datos.

La abstracción de datos libera al programador de preocuparse sobre detalles no esenciales.

Algunos de los lenguajes modernos tales como Smalltalk, Ada, etc., permiten definir sus propios tipos abstractos de datos.

Tipos abstractos de datos (TAD) es un tipo de dato definido por el programador que se puede manipular de un modo similar a los tipos de datos definidos o incorporados al sistema.

Un objeto es simplemente un tipo abstracto de datos y, por ello, las nociones de programación orientada a objetos se construye sobre la idea de dicho tipos de datos, añadiéndoles otras propiedades, como puede ser la ocultación y reutilización de código.

ENCAPSULACIÓN

La abstracción y el encapsulamiento son conceptos complementarios: La abstracción se enfoca a la vista exterior de un objeto y el encapsulamiento de datos es la habilidad que tiene un objeto de ocultar información de otras partes de un programa.

Por ejemplo cuando se observa una radiografía se pueden ver los órganos internos de una persona, el corazón, los huesos, etc... Sin los rayos X no se pueden observar los organos ya que la piel los esconde este fenómeno es comúnmente conocido como encapsulación de datos.

Cuando se observa una licencia de conducir se pueden aprender ciertas características acerca de una persona como su peso, estatura, dirección ,etc.. pero existen muchos factores de una persona que no se conocen a simple vista, la licencia representa una parte de la información a esto se le conoce como abstracción de datos en la fig. 3.5.2 se muestra la comparación entre encapsulación y abstracción de datos.

La encapsulación permite que los miembros de una clase no se modifiquen sin permitirlo el programador por funciones definidas dentro de una clase.

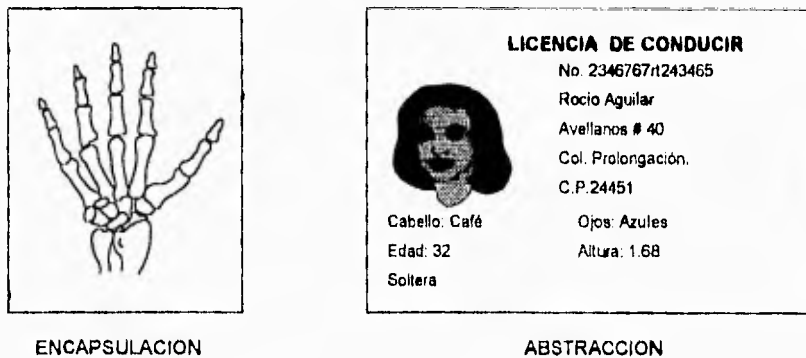


Fig.3.5.2 Comparación entre Encapsulación y Abstracción.

HERENCIA

En nuestra vida diaria el concepto de clases se divide en subclases. Así, las clases animales se dividen en: mamíferos, anfibios, insectos, etc. El principio en que se basan las sucesivas subclases es que cada subclase comparte características comunes con la clase con la que se deriva o desciende, por ejemplo, todas las clases de vehículos tienen un motor, ruedas, volante, etc. Además de las características comunes, cada subclase tiene sus propias características particulares, por ejemplo, número de ruedas, número de asientos, etc.

Herencia es una de las más importantes características de la programación orientada a objetos que diferencia a ésta de los sistemas comunes de programación. Cada programación orientada a objetos provee una herencia de una manera o otra. La herencia promueve compartir código permitiendo al lenguaje que el programador reutilice el código de una clase a otra. Un sistema del área de personal contiene información acerca de los empleados de la misma, éstos son personas y todas las personas tienen características en común como: nombre, dirección, fecha de nacimiento, pero además el sistema debe registrar diferentes tipos de empleados diferentes, Encargados de proyecto y personal de soporte, éstos deben ser registrados de diferente manera, aún así se puede diseñar una clase base de empleados que contenga dirección, número de seguro social, etc. y derivar clases separadas para gerentes, Encargados de proyectos, etc. cada una de éstas clases derivadas pueden heredar las características de la clase empleado y además tener características propias. Por ejemplo la clase gerente debe incluir un aumento anual de salario que las otras clases no poseen y el encargado de proyecto una lista de proyectos asignados etc. La figura 3.5.3. muestra un ejemplo de herencia.



Fig.3.5.3. Herencia

En una programación orientada a objetos se pueden crear nuevas clases de clases ya existentes pero modificando algunas propiedades de éstas. Una clase de la cuál otra es derivada se le conoce como *clase base*, y todas las clases que son derivadas de ella (a través de herencia) son conocidas como *clases derivadas* y las clases que son derivadas de ellas se conocen como *clases parientes*. Además, la nueva clase puede añadir nuevos elementos datos y métodos a los que hereda de su clase base.

Cualquier clase base (incluyendo un descendiente) puede tener cualquier número de clases descendientes.

Mediante el mecanismo de la herencia se puede construir una jerarquía de clases que adoptará la forma típica de árbol, donde la clase base se llamará padre, y la clase descendiente o derivada, clase hija.

Existen dos tipos de herencia: Simple y Múltiple. La herencia simple, es aquella en la que una clase solo puede tener un ascendiente. Herencia múltiple se refiere a la capacidad de las clases para heredar propiedades de más de una clase. Un caso típico de herencia múltiple es el popular término multimedia, supongamos dos tipos de clases, sonido y gráfico; si se combinan ambos pueden dar lugar a una nueva clase multimedia con propiedades de ambas clases, otro ejemplo podría ser el hecho de que un hijo herede algunas facciones de su padre, otras de su madre y tal vez otras de su abuelo. Ver Fig.3.5.4.

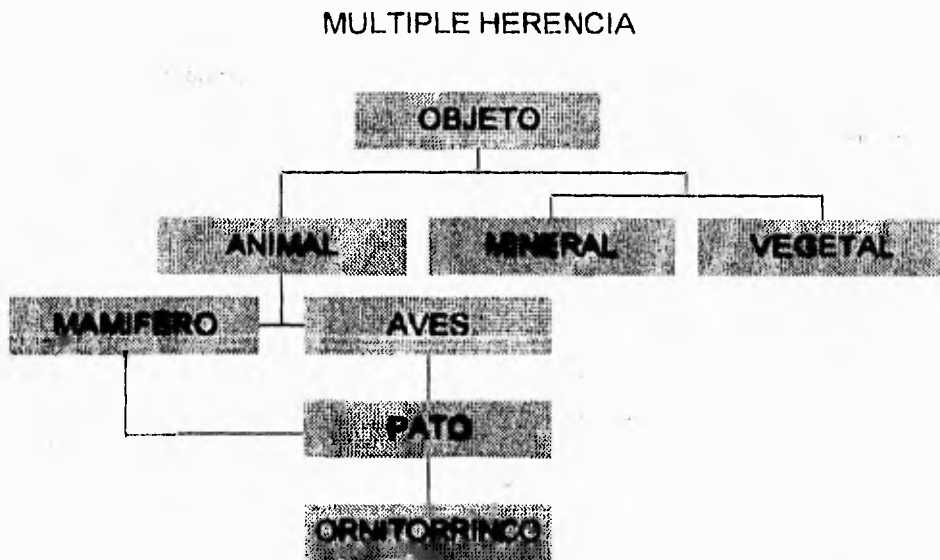


Fig.3.5.4 Herencia Múltiple

POLIMORFISMO

En biología la ocurrencia de diferentes propiedades en un organismo individual o en organismos de la misma especie es llamado polimorfismo. El polimorfismo permite crear una familia de clases con conductas que varían de acuerdo a lo que es apropiado para cada clase. En otras palabras, objetos diferentes reaccionan de modo diferente al mismo mensaje. El polimorfismo permite que una función llame a múltiples funciones. El polimorfismo requiere de herencia, por lo que solo se puede aplicar a métodos heredados de una clase base.

El polimorfismo depende de una ligadura, que es el proceso por el cual un método se asocia con una función real. Cuando los métodos polimórficos se utilizan, el compilador no puede determinar a qué función llamar; la función específica llamada depende de la clase de elemento al cual se envía el mensaje; en consecuencia, la función a llamar se determina en tiempo de ejecución. Esta operación se conoce como **ligadura tardía** (late binding) ya que ocurre cuando el programa se está ejecutando. La **ligadura temprana o previa** ocurre para métodos no polimórficos; el compilador conoce exactamente cuál es el método que se invoca, de modo que se puede construir una llamada directa en tiempo de compilación. Aunque la ligadura previa es muy eficiente la mayoría de los lenguajes orientados a objetos utilizan ligaduras tardías para todos los métodos. La fig. 3.5.5. muestra un ejemplo de polimorfismo.

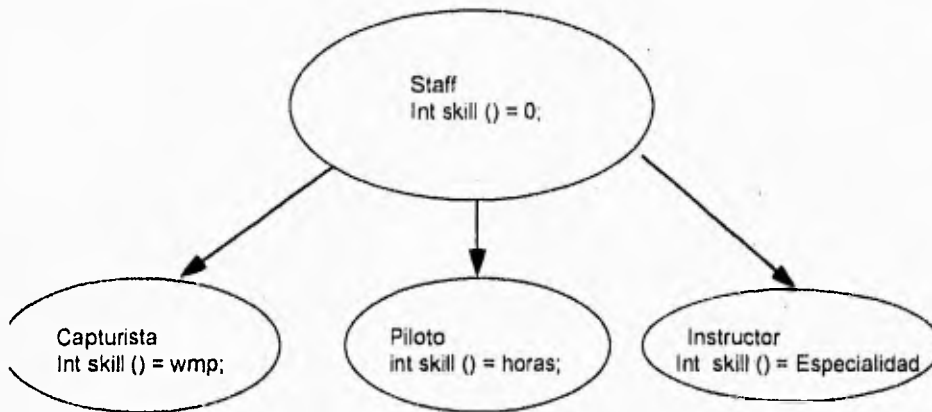


Fig.3.5.5 Polimorfismo

REUTILIZACIÓN

Una vez que una clase se ha escrito, creado, depurado y comprobado, se puede distribuir a otros programadores para utilizarla en sus propios programas. Es decir, se puede utilizar como bloque básico para construir un programa. Esta operación se conoce con el nombre de reutilización.

El concepto de herencia en programación orientada a objetos proporciona una extensión importante a la idea de reutilización. Un programador puede tomar una clase existente, perteneciente a una biblioteca de clases y, sin modificarla, le puede añadir características y propiedades adicionales.

3.6 DISEÑO DE ARBOLES GENEALÓGICOS

Un árbol se puede considerar como la extensión lógica de una lista enlazada, y se utiliza principalmente para facilitar la búsqueda de datos clasificados. Los nodos hacen referencia a datos y estos nodos se organizan en una estructura de árbol. Las listas tienen una estructura unidimensional, en tanto que los árboles tienen una estructura bidimensional. La figura 3.6.1. ilustra un árbol genérico.

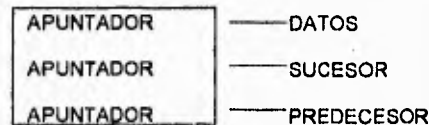


FIG.3.6.1. Árbol Genérico

El número de descendientes que puede tener un nodo se conoce como orden de un árbol. En un árbol binario, los nodos tienen dos descendientes. En un árbol n -ario, los nodos tienen n descendientes.

Todos los árboles emplean el concepto de comparación para localizar elementos. La comparación requiere una forma de determinar si los elementos son iguales, menores o mayores que cada uno de los otros. En el caso de tipos de datos escalares, como los enteros y los caracteres, no hay problema. Para utilizar los objetos de clase de un árbol se debe disponer de operadores de ordenación apropiados.

ÁRBOLES BINARIOS

Los árboles binarios se cuentan entre las formas más simples de árboles. Cada nodo almacena un elemento, además de dos apuntadores a descendientes: un descendiente izquierdo y uno derecho. Cuando un nodo no tiene descendiente, el apuntador correspondiente es nulo. Un nodo sin descendientes se denomina nodo terminal.

RECORRIDO DE ÁRBOLES BINARIOS

Para hacer el barrido de un árbol binario se utilizan tres tipos de métodos. Todos los métodos serán definidos en forma recursiva de tal manera que el recorrer un árbol binario comprende la visita a la raíz y el recorrido de los subárboles izquierdo y derecho. La única diferencia entre los métodos es el orden en que se realizan estas tres operaciones.

Para recorrer un árbol binario en **preorden**, realizamos las siguientes tres operaciones:

1. Visitar la raíz
2. Recorrer el subárbol de la izquierda en preorden.
3. Recorrer el subárbol de la derecha en preorden

Para recorrer un árbol binario no vacío en una forma de **entreorden**:

1. Recorrer el subárbol de la izquierda en entreorden.
2. Visitar la raíz

3. Recorrer el subárbol de la derecha en entreorden.

Para recorrer un árbol binario que no está vacío en **posorden**:

1. Recorrer el subárbol de la izquierda en posorden.
2. Recorrer el subárbol de la derecha en posorden.
3. Visitar la raíz.

La Fig.3.6.2. presenta dos árboles binarios y sus recorridos de preorden, entreorden y posorden.

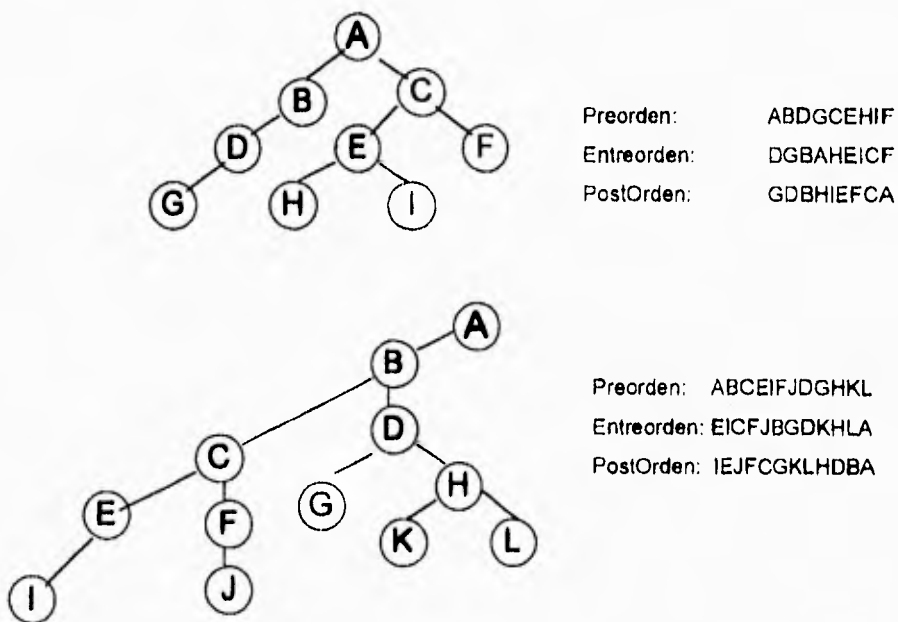


Fig.3.6.2 Árboles Binarios y sus Recorridos.

INSERCIÓN DE ELEMENTOS

Cuando se agrega un elemento a un árbol binario, el elemento se compara con el del nodo raíz. Si el valor es igual al elemento de la raíz, el elemento ya está contenido en el árbol. A menudo los árboles binarios deshabilitan inserciones múltiples del mismo elemento, pero es posible utilizar una variable contadora en cada nodo para contar el número de inserciones de un elemento dato. Si el elemento es menor que el elemento raíz, se elige el apuntador del descendiente izquierdo. Se repite el proceso de pasar de un nodo a su descendiente hasta que se encuentra un nodo que no tenga descendiente o hasta que se localice el elemento. Si no se localiza el elemento, este se inserta en el árbol como descendiente de último nodo visitado.

BÚSQUEDA DE ELEMENTOS

Los procesos de agregar y buscar elementos son mutuamente dependientes. No se pueden buscar elementos a menos que haya un árbol con elementos ya contenidos, y no se pueden agregar elementos a menos que se pueda buscar en un árbol. La búsqueda se realiza pasando de un nodo a uno de sus descendientes, hasta que encuentre el elemento o el final del árbol. Si el elemento destino es menor que el elemento de un nodo dado, se selecciona el descendiente izquierdo. De lo contrario se utiliza el descendiente derecho.

ELIMINACIÓN DE ELEMENTOS

En los árboles binarios, la eliminación de elementos puede ser, en general, más complicada que la adición de elementos. Existen tres tipos de nodos y cada uno de ellos afecta un árbol binario de manera diferente.

Nodos terminales (ningún descendiente). Tan sólo se inicia el apuntador del nodo de progenitor al valor nulo.

Nodos internos (un descendiente). El apuntador del nodo progenitor se define para que apunte al único descendiente del nodo.

Nodos internos (dos descendientes). Se tienen que ordenar los descendientes izquierdos y derechos del nodo en el árbol restante. Si los descendientes son nodos terminales, el proceso es simple. Si los descendientes son las raíces de subárboles, tiene que combinar los dos subárboles de un subárbol y luego reemplazar el nodo interno borrado con el subárbol combinado.

ÁRBOLES - B

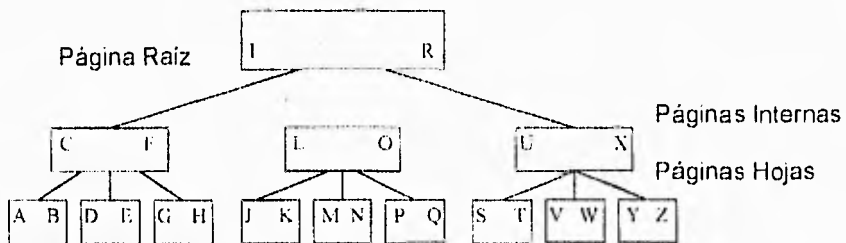
Los árboles-B son árboles balanceados especializados. Los árboles-B no son árboles binarios y se crearon como una solución a las deficiencias de los árboles desbalanceados.

Aunque es posible balancear un árbol binario después de cualquier inserción en la práctica se incurre en un gasto excesivo si se insertan elementos con frecuencia. Lo

árboles-B se utilizan cuando se desconoce la cantidad de datos que se deben manejar y se espera sea bastante grande. Los árboles- B son árboles multivía, lo que significa que cada nodo puede tener no sólo dos, sino muchos descendientes. El número de descendientes de un nodo define el orden de un árbol-B.

La principal diferencia entre un árbol - B y un árbol binario es que los primeros pueden tener más de 2 hijos en cada nivel.

Las páginas en los arboles-B son analogas a los nodos en los arboles binarios: hay una página raíz, páginas internas y paginas hojas. Cada página puede contener muchos miembros o items, cada item guarda una abreviación del dato registrado. Las páginas hojas no tienen hijos. Un ejemplo de un árbol B se muestra en la sig. figura:



ESTRUCTURA DE LOS ARBOLES - B

Con los árboles -B, el desempeño está garantizado porque existe la garantía de que la estructura del árbol permanezca balanceada. Lo que hace que los árboles-B sean superiores a los árboles binarios y a otros tipos de árboles es un conjunto de reglas

que se utilizan para expandir y contraer el árbol. Los árboles-B crecen de las hojas hacia arriba a diferencia de los binarios que crecen de la raíz hacia abajo.

REGLAS DE LOS ÁRBOLES - B

- Los nodos no terminales pueden tener a los sumo N claves y $N+1$ descendientes.
- Los nodos terminales pueden tener un máximo de $2^{(N+1)}$ claves. Por definición , los nodos terminales no tienen descendientes.
- Todos los nodos (menos el raíz) deben tener un número mínimo de claves, determinado por las variables `InnerLowWater` y `LeafLowWaterMark`.
- Todos los nodos terminales están a la misma distancia que la raíz.
- El orden de un árbol-B es mayor que o igual a 3.
- Sólo los objetos derivados de la clase `sorteable` se pueden manejar como claves.

.7 FLUJO DE DATOS

En el análisis del comportamiento de objetos se realizan esquemas de eventos que muestran eventos, la secuencia en que ocurren y cómo éstos modifican el estado de un objeto.

ESTADO DE UN OBJETO

Un objeto puede existir en varios estados. Por ejemplo, un objeto reservación aérea puede ser una instancia de alguno de los siguientes tipos de objetos:

- Reservación solicitada
- Reservación en lista de espera
- Reservación confirmada
- Reservación cancelada
- Reservación satisfecha (una vez que el avión ha despegado)
- Reservación archivada

Estos tipos de objetos se pueden percibir como estados posibles del ciclo vital de un objeto.

EVENTOS

En el análisis orientado a objetos el mundo se describe en términos de los objetos sus estados, así como de los eventos que modifican esos estados. El analista orientad a objetos no desea conocer cada evento que ocurra en una organización, sólo los tipos de eventos. Así como hablamos de tipos de objetos, podemos hablar de tipos d eventos. Por ejemplo, en la Fig. 3.7.1 se muestra el tipo de evento *reservación en lista de espera confirmada* es la colección de eventos donde un objeto cambia de un *reservación en lista de espera* a una *reservación confirmada*.

Evento en el que el objeto cambia de ser una instancia de *reservación en lista de espera* a una *reservación confirmada*

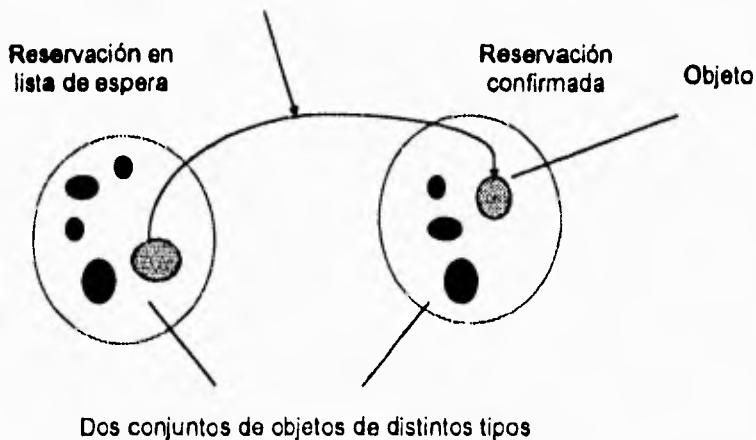


Fig. 3.7.1 Estado de un objeto que cambia de un tipo de objeto a otro.

Los tipos de eventos indican los cambios sencillos en el estado de un objeto: por ejemplo, cuando se deposita dinero en una cuenta bancaria o se actualiza el salario de un empleado. Básicamente, los tipos de eventos describen las siguientes formas de cambios de estado:

- Un objeto se crea. Por ejemplo, se crea una reservación aérea.
- Un objeto se termina. Por ejemplo, un producto se destruye o un contrato se termina.
- Un objeto se clasifica como una instancia de un tipo de objeto. Por ejemplo, una empresa se convierte en cliente, un empleado se convierte en gerente.
- Un objeto se desclasifica como una instancia de un tipo de objeto. Por ejemplo, una firma deja de ser cliente; un producto sale del catálogo de ventas.
- Un objeto cambia de clasificación. Por ejemplo, un abogado pasa de ayudante a socio; una cuenta cambia de normal a atrasada.
- El atributo de un objeto se cambia.

Los eventos pueden asociar un objeto con otro. Por ejemplo, en la mayoría de las organizaciones, cuando un objeto se clasifica como empleado, debe estar asociado con un departamento. Un evento clasificará al objeto como empleado. Otro evento creará una asociación entre el objeto empleado, y un objeto departamento. (Las asociaciones son objetos como los demás. Este tipo de asociación se puede llamar el tipo de objeto asignación empleado-departamento.) Si el empleado, cambia de departamento, se crea una nueva asignación empleado-departamento y se termina la anterior.

asignación empleado-departamento.) Si el empleado, cambia de departamento, se crea una nueva asignación empleado-departamento y se termina la anterior.

OPERACIONES

En el análisis orientado a objetos, el término operación se refiere a una unidad de procesamiento que puede ser solicitada. El procedimiento se implanta mediante un método. El método es la especificación de cómo llevar a cabo la operación. Es el guión de la operación. A nivel programa, el método es el código que implanta la operación. Una operación puede o no cambiar el estado de un objeto, si lo cambiara ocurriría un evento. Las operaciones se representan mediante cuadros con esquinas redondeadas. Los tipos de eventos se representan mediante triángulos sólidos negros conectados a la caja como se muestra en la siguiente figura :

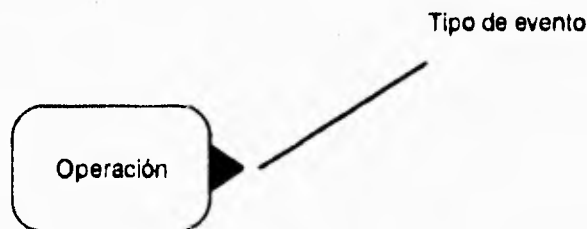


Fig. 3.7.2 Representación de operaciones y eventos.

FUENTES EXTERNAS DE ACTIVACIÓN

Los eventos son cambios de estado en los que un sistema debe reaccionar ante ellos de algún modo. Muchas de las operaciones que producen estos eventos suelen ser externas al sistema. En este caso, el símbolo de operación se dibuja como una caja sombreada con esquinas redondeadas.

Un reloj externo es una forma particular de fuente externa. Indica que un proceso externo emitirá señales de reloj con cierta frecuencia determinada con anterioridad; como cada segundo, al final de cada día, el principio de cada mes. Así, el evento se da cuando aparece la señal del reloj externo.

REGLAS DE ACTIVACIÓN

Cuando ocurre un evento, lo usual es que el cambio de estado active el llamado a una o más operaciones. Por ejemplo, si se retiran biones de un almacén y la cantidad conservada en él baja de cierto nivel, ello puede activar una operación para realizar un pedido.

Las reglas de activación definen la relación entre la causa y el efecto. Siempre que ocurra un evento de cierto tipo, la regla de activación invoca a una operación ya definida. El tipo de evento pedido procesado tiene dos reglas de activación que reaccionan ante cada ocurrencia del evento y activan sus operaciones asociadas. Un tipo de evento puede tener varias reglas de activación, cada una de las cuales invoca a su operación en paralelo. Las operaciones paralelas pueden producir diferentes

cambios de estado en forma simultánea. Además, una operación puede ser invocada por varias reglas de activación. Por ejemplo, en la siguiente figura se activa la operación genera cheque cuando ocurre un evento cheque solicitado o fin-de-mes

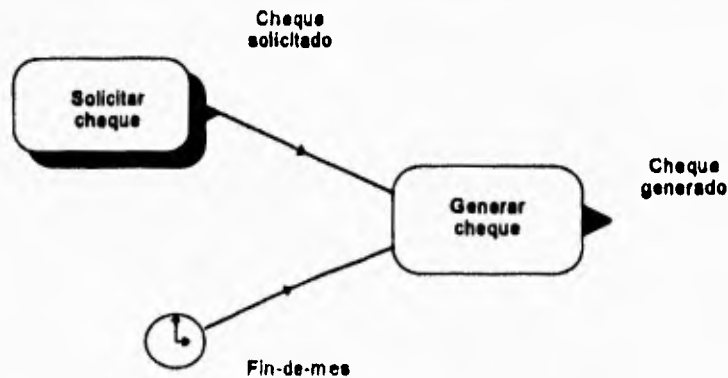


Fig. 3.7.3 La operación producir cheque se activa cuando ocurre cualquiera de los eventos cheque solicitado o fin-de-mes.

DIAGRAMAS DE FLUJO DE OBJETOS

Los esquemas de eventos son adecuados para la descripción de procesos en términos de eventos, de reglas de activación, de condiciones y de operación.

Sin embargo, podría no ser adecuado expresar de esta forma los procesos grandes complejos. En estos casos son útiles los diagramas de flujo de objetos.

Los diagramas de flujo de objetos (DFO) son parecidos a los diagramas de flujo de datos (DFD), puesto que muestran las actividades que interactúan con otras. En la

DFD, una interfaz transfiere datos. En las técnicas orientadas a objetos el diagrama debe representar cualquier tipo de cosa que se transfiera de una actividad a otra, ya sean pedidos, partes, artículos terminados o datos. En resumen, el DFO indica los objetos que se producen y las actividades que los producen e intercambian. La siguiente figura presenta un ejemplo de diagrama de flujo de objetos.

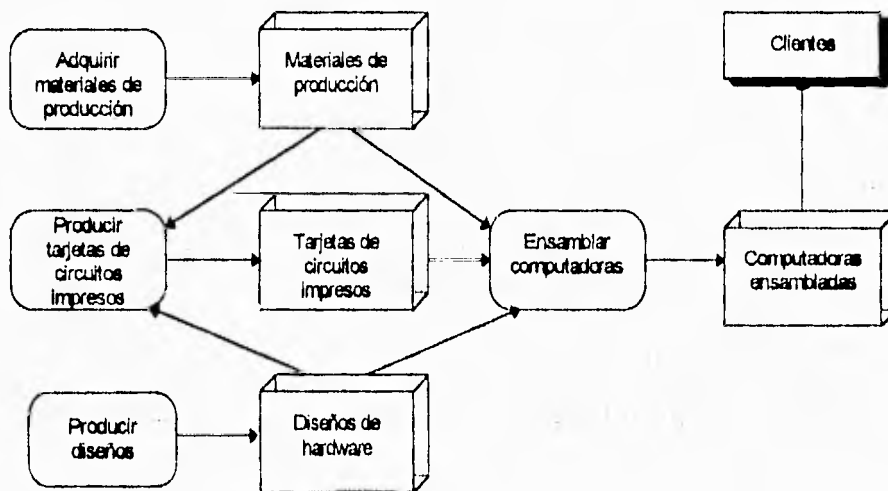


Fig. 3.7.4 Diagrama de flujo de objetos. El símbolo de almacenamiento de objetos es una caja tridimensional, lo cual indica que el DFO representa el hecho de que los objetos de la vida real influyen entre las actividades.

Los DFO describen los objetos , como se producen y se consumen

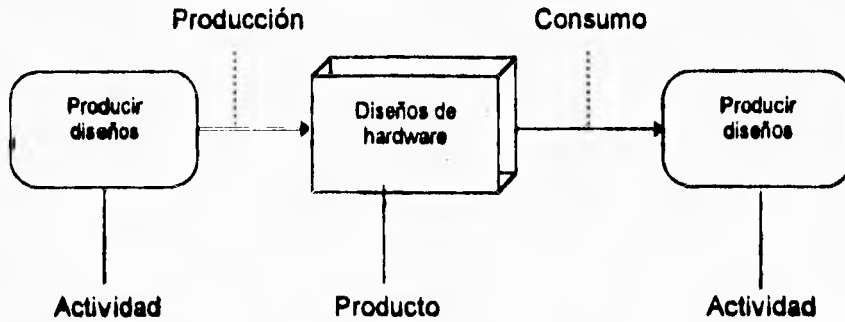


Fig. 3.7.5 Diagrama de flujo de objetos.

El producto es el resultado final que satisface el propósito de la actividad. Los productos se genera para el consumo por parte de otras actividades que le añaden valor al producto consumido (para generar un producto más complejo). A cada paso se crean cualidades nuevas, más complejas y sutiles. Así, los diagramas de flujo de objetos pueden representar el modelo en forma ascendente o descendente.

MODELO FUNCIONAL

El modelo funcional especifica los resultados de un cálculo sin importar quién o cuándo fueron calculados. El modelo funcional consiste de múltiples diagramas de flujo de datos los cuales especifican el significado de las operaciones y restricciones. Un diagrama de flujo de datos (DFD) muestra las relaciones funcionales de los valores calculados por un sistema, incluyendo valores de entrada, de salida y datos almacenados internamente. Un DFD muestra el flujo de los valores desde sus fuentes a través de procesos que los transforman hasta su destino.

Un DFD contiene procesos, objetos demandantes y objetos almacenadores que transforman, mueven, consumen y almacenan datos.

PROCESOS

Un proceso transforma los datos y puede tener varios efectos si éste tiene componentes no funcionales, tales como almacenadores de datos u objetos externos. El modelo funcional indica las rutas funcionales. Ejemplos de procedimientos no funcionales son la lectura y escritura en archivos y el despliegue de imágenes en un sistema de ventanas.

Un proceso se dibuja como una elipse que contiene el nombre de la transformación. Cada proceso tiene un número fijo de flechas de entrada y salida cada uno de los cuales lleva un valor de cierto tipo. Las entradas y salidas pueden ser etiquetadas para mostrar su función.

La siguiente figura muestra dos procesos, nótese que un proceso puede tener más de una salida.

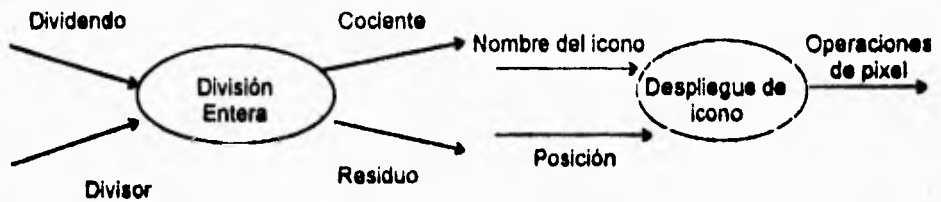


Fig. 3.7.6 Procesos.

La figura sólo muestra el patrón de entradas y salidas. El cálculo de valores de entrada y salida también puede ser especificado. Los procesos de alto nivel pueden ser incluidos en un diagrama de flujo de datos completo, muchas subrutinas pueden ser expandidas en subrutinas de bajo nivel. Los procesos atómicos deben describirse directamente, en lenguaje natural, ecuaciones matemáticas, o por algún otro medio. Por ejemplo, "División entera" puede definirse matemáticamente. Frecuentemente los procesos atómicos son tan simples como acceder un valor de un objeto.

Los procesos se implementan como métodos (o fragmentos de métodos) de operaciones en clases de objetos. El objeto destino es usualmente uno de los flujos de entrada, especialmente si la misma clase de objetos es también un flujo de salida.

FLUJO DE DATOS

Un flujo de datos conecta la salida de un objeto o proceso a la entrada de otro. Un flujo de datos es dibujado como una flecha entre el proceso que produce los datos y el proceso que los consume. La flecha se etiqueta con una descripción de los datos, usualmente el nombre o el tipo. El mismo valor puede ser enviado a varios lugares; esto se indica por una bifurcación con varias flechas que salen de ella. Algunas de las flechas de salida no se etiquetan porque representan el mismo valor que la entrada. La siguiente figura representa algunos flujos de datos: Algunas veces un dato es descompuesto en sus componentes, cada uno de los cuales va a un proceso diferente. Esto se representa con una bifurcación en la cual cada una de las flechas es etiquetada con el nombre del componente.

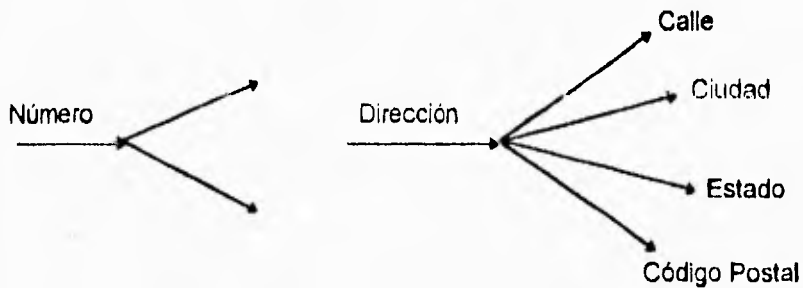


Fig. 3.7.7 Flujo de datos para copiar un valor y descomposición de un valor.

DEMANDANTES

Un demandante es un objeto activo que maneja el diagrama de flujo de datos produciendo o utilizando valores. Los demandantes son asociados a las entradas y salidas de un diagrama de flujo de datos. En un sentido, los demandantes radican en los bordes del diagrama de flujo de datos como fuentes y consumidores de datos, por lo que algunas veces son llamados terminadores. Ejemplos de demandantes utilizados en un programa son un termostato, y un motor controlado por computadora.

Un demandante se representa con un rectángulo para mostrar que es un objeto. Las flechas entre el demandante y el diagrama son entradas y salidas del diagrama.

ALMACENADORES DE DATOS

Un almacenador de datos es un objeto pasivo dentro de un diagrama de flujo de datos que almacena datos para un acceso posterior. No genera ninguna operación pero responde a demandas de almacenamiento, se dibuja como un par de líneas paralelas que contienen el nombre del almacenador. Las flechas de entrada indican la información u operaciones que modifican los datos almacenados; esto incluye adición de elementos, modificación de valores y borrado de elementos. Las flechas hacia afuera representan la información obtenida del almacenador; esto incluye la obtención de toda la información o sólo una parte. En la siguiente figura se muestra un almacenador de datos de lecturas de temperatura. Al final del día, las lecturas máximas y mínimas son consultadas.

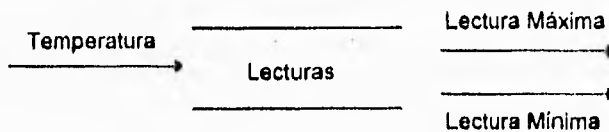


Fig. 3.7.8 Almacenadores de datos.

Tanto los demandantes como los almacenadores de datos son objetos. Se diferencian por su funcionamiento y uso, aunque en un lenguaje orientado a objetos ambos pueden ser implementados como objetos. Por otra parte, un almacenador de datos puede ser implementado como un archivo y un demandante como un dispositivo externo.

Existe una diferencia entre ver un objeto como un valor simple o como un almacenador de datos que contiene muchos valores.

Un diagrama de flujo de datos es utilizado particularmente para mostrar un alto nivel de funcionalidad de un sistema. Un proceso puede expandirse a otro diagrama de flujo. Cada entrada y salida del proceso es una entrada o salida del nuevo diagrama.

CONTROL DE FLUJO

Un diagrama de flujo de datos muestra todas las rutas posibles de los datos, pero no muestra las rutas que se van a ejecutar y el orden. Un control de flujo es un valor booleano que se afecta cuando se evalúa un proceso. El control de flujo no es un valor de entrada a un proceso en sí. Se representa con una línea punteada que va desde un proceso que genera un valor booleano al proceso que será controlado.

En la siguiente figura se muestra un diagrama de flujo de datos para un retiro de una cuenta bancaria. El cliente proporciona una clave y una cantidad. El retiro se efectúa solamente si la clave es correcta.

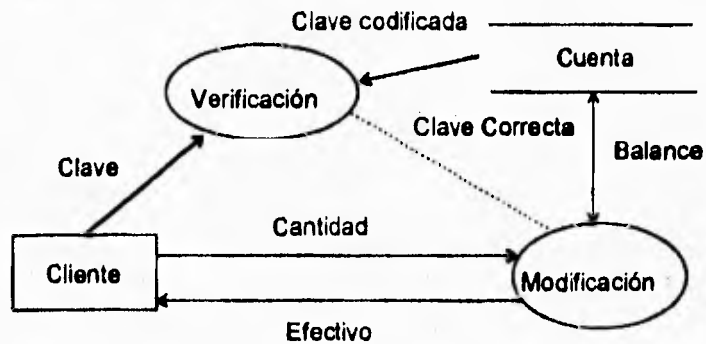


Fig. 3.7.9 Control de flujo.

ESPECIFICACIÓN DE OPERACIONES

En un diagrama de flujo de datos, las operaciones pueden representarse de la siguientes formas:

- Funciones matemáticas, tales como funciones trigonométricas.
- Tabla de valores de entrada y salida (enumeración).
- Ecuaciones, especificando la salida en términos de la entrada.
- Tablas de decisión.
- Seudocódigo.
- Lenguaje natural.

La especificación de una operación incluye una señal y una transformación. La señal define la interface de la operación : el argumento (número, orden y tipo) y el valor que regresa (número, orden y tipo). La transformación define el efecto de una operación : los valores de salida como función de los valores de entrada.

La especificación externa de una operación solamente describe los cambios visibles fuera de la operación. Durante la implementación de una operación, los valores internos pueden ser creados por conveniencia u optimización. Algunos valores pueden ser parte del estado interno de un objeto. Por ejemplo, una lista ordenada de valores puede ser implementada utilizando varias estructuras de datos como listas lineales o arboles balanceados, cuya organización interna puede ser libremente modificada y no presenta cambios externos en la lista ordenada. Tales detalles internos no aparecen en la especificación externa. el propósito de la especificación es indicar lo que hace la operación lógicamente, no cómo será implementado.

Las *operaciones de acceso* son operaciones que leen o escriben atributos o ligas de un objeto. Durante el diseño es necesario indicar las operaciones de acceso que serán públicas y las que serán privadas a la clase de objetos.

RESTRICCIONES

Una restricción muestra la relación entre dos objetos al mismo tiempo (tales como frecuencia y longitud de onda), o entre diferentes valores de un mismo objeto en diferente tiempo. Una restricción puede ser expresado como una función total (un valor es completamente especificado por otro). Por ejemplo, una transformación de coordenadas puede especificar que el factor de escala de la coordenada x y la coordenada y será igual; esta restricción define totalmente un valor en términos de otro.

3.8 ESTRUCTURAS JERÁRQUICAS, TIPOS Y SUBTIPOS

CONCEPTOS

CLASE

Una clase es una especificación genérica para un número arbitrario de objetos similares.

Se pueden derivar nuevas clases de las ya existentes mediante mecanismos de herencia. La clase de la cual se deriva otra clase es llamada *clase base*.

Cuando una clase se deriva de otra, la clase derivada hereda características de la clase base, específicamente, una clase derivada hereda todas las variables públicas, protegidas y funciones miembro de la clase base. Sin embargo, una clase derivada no puede derivar las variables privadas.

Además de las variables miembro y funciones miembro que hereda, un clase derivada, puede definir sus propias variables y funciones miembro.

Cuando una clase se deriva de otra, muchas clases más pueden derivarse de ella, de esta forma una clase derivada puede ser una clase base. Los múltiples niveles de clases que se derivan de cada una forman una *jerarquía de clases*. En una jerarquía de clases, las clases del nivel más bajo heredan todas las variables miembro públicas y protegidas (pero no las privadas) y funciones miembro de su clase base.

RAZONES PARA UTILIZAR CLASES DERIVADAS:

- Cuando una clase se deriva de una clase base, el resultado es un nuevo tipo de datos que hereda todas las cualidades de la clase base, sin que esto afecte al resto del programa.
- Cuando se desee derivar una clase de una base, no se necesita tener acceso al código fuente de la base. Si se tiene acceso al código fuente que define la clase base (el cual es almacenado usualmente en un archivo de encabezados), se tiene todo lo necesario para derivar más clases de ésta.
- Una jerarquía de clases puede hacer uso de *Clases Abstractas*; estas son clases de propósito general que no hacen nada por si mismas, pero se diseñadas específicamente para ser utilizadas como clases base. El único propósito de una clase abstracta es definir el comportamiento de una estructura de datos genérica.
- Con las jerarquías de clases compuesta de clases base y clases derivadas, se puede hacer uso de otras propiedades de los lenguajes orientados a objetos: tales como polimorfismo - esto es la capacidad de las clases para modificar su propio comportamiento basándose en las características de las subclases.

VENTAJAS DE UTILIZAR CLASES DERIVADAS

Una de las ventajas es que las clases derivadas proporcionan un medio para construir una jerarquía de clases orientada a objetos bien organizada, en la cual el usuario define

tipos de datos que se generan de una clase raíz.

Mediante las clases derivadas que heredan comportamientos de otras clases, se puede crear nuevas clases que son similares - pero no idénticas - a otras clases. De ésta forma, la herencia proporciona mecanismos sistemáticos, lógicos y fáciles de usar para crear nuevas clases con nuevos comportamientos.

SUBCLASE

Es una clase que hereda el funcionamiento de otra clase.

SUPERCLASE

Es una clase cuyo funcionamiento puede ser heredado. Una clase puede solamente tener una superclase.

INSTANCIA

Objetos que han sido, de alguna manera especificados por una clase. Todos los objetos son instancias de alguna clase.

ESTRUCTURAS JERÁRQUICAS

Algunas de las herramientas que se utilizan para obtener una perspectiva global en las relaciones de herencia son :

- Gráficas jerárquicas.
- Diagramas de Venn.

GRÁFICAS JERÁRQUICAS

Una gráfica jerárquica es una herramienta que muestra una representación gráfica de las relaciones heredadas entre las clases. Las clases se representan mediante rectángulos, etiquetados con el nombre de la clase.

La herencia se indica mediante una línea de la superclase a la subclase, las superclases se encuentran arriba de las subclases. La siguiente figura indica que la *herramienta de selección* y la *herramienta de creación* son subclases de *herramienta*.

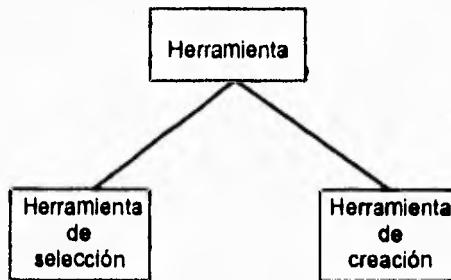


Fig. 3.8.1. Gráfica Jerárquica Simple.

La siguiente figura indica que *Arreglo* es una subclase de *Colección Ordenada* y de *Colección Indexada*. *Magnitud* no es subclase de ninguna. *Matriz* es una subclase de *Arreglo*. *Fecha* es una subclase de *Magnitud*. Finalmente *Cadena* es una subclase de *Arreglo* y de *Magnitud*.

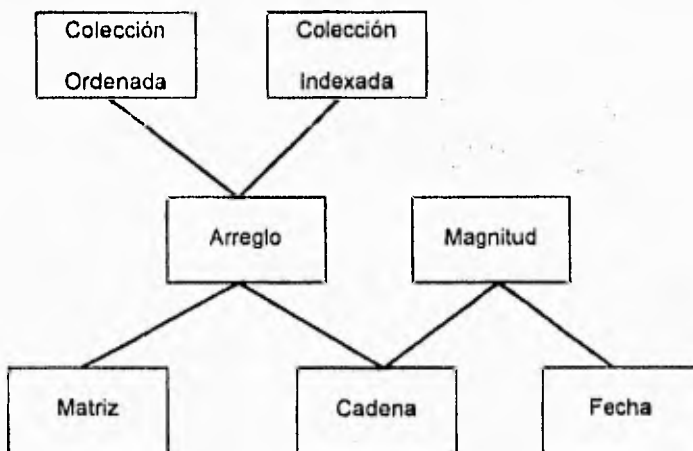


Fig. 3.8.2 Gráfica Jerárquica compleja.

En el análisis, es necesario distinguir entre clases abstractas y clases concretas:

Las Clases Abstractas se diseñan solamente para ser heredadas. Se representan con rectángulos con la esquina superior izquierda rellena.

DIAGRAMAS DE VENN

Si vemos las clases como conjuntos de responsabilidades, podemos utilizar los diagramas de Venn para mostrar la distribución de responsabilidades entre clases. Por ejemplo, la jerarquía de clase de la figura 3.8.1 se puede representar con diagramas de Venn de la siguiente manera :



Fig. 3.8.3 Diagrama de Venn de una jerarquía simple.

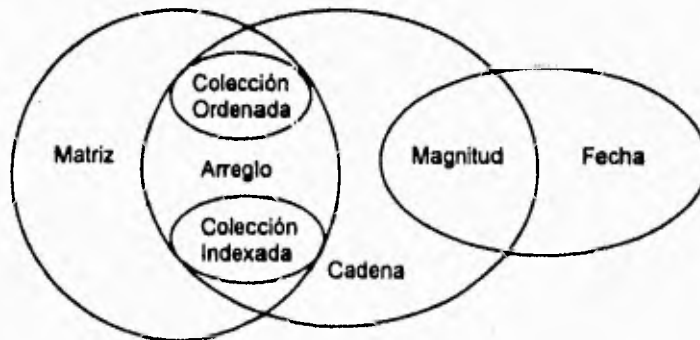


Figura 3.8.4 Diagrama de Venn Complejo.

Una vez que se realizaron los diagramas jerárquicos se analiza las relaciones de herencia entre clases para identificar y resolver los problemas del diseño. El análisis se basa en lo siguiente :

- Modelo de una jerarquía.
- Factor de responsabilidades, tan alto como sea posible.
- Asegurarse de que las clases abstractas no hereden de clases concretas.
- Eliminar las clases que no agreguen funcionalidad.

MODELO DE UNA JERARQUÍA

Una clase puede heredar de otra clase solamente si soporta todas las responsabilidades definidas para esa clase. Las subclasses pueden soportar todas las responsabilidades definidas por sus superclases, y posiblemente mas.

Los diagramas de Venn pueden ayudar a visualizar cuando una subclase no es capaz de soportar todas las responsabilidades de su superclase.

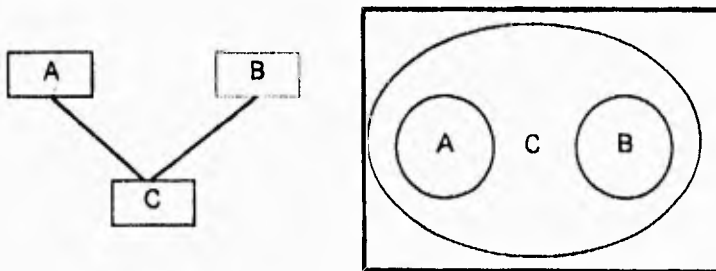


Fig. 3.8.5 Responsabilidades correctamente formadas de una Subclase.

Si una subclase soporta solamente parte de las responsabilidades de la superclase, su diagrama de Venn se representará como sigue :

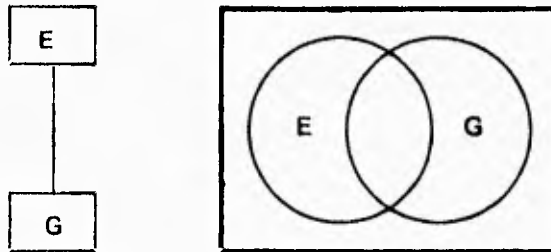


Fig. 3.8.6. Relación incorrecta de subclase/superclase.

Quando el comportamiento de una subclase incluye solamente parte de las responsabilidades definidas por su superclase, se crea una clase abstracta con todas las responsabilidades comunes a la clase y superclase, a las cuales serán heredadas como se muestra en la siguiente figura :

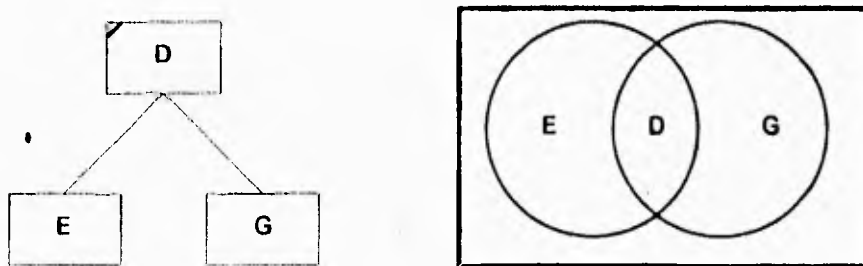


Fig. 3.8.7. Relaciones de Herencia Revisada.

FACTOR COMÚN DE RESPONSABILIDADES TAN ALTO COMO SEA POSIBLE.

Como corolario de lo anterior, se puede establecer que si en un conjunto de clases, todas soportan responsabilidades comunes, las pueden heredar de una superclase común.

Si la superclase común no existe, se crea y se mueven las responsabilidades comunes a ésta.

Una forma de tener un factor de responsabilidades alto en la jerarquía es diseñando tantas clases abstractas como sea posible. Cuando se haya determinado cuántas clases abstractas existirán en el sistema, se especula en las clases abstractas que puede encapsular el funcionamiento que será utilizado por subclases existentes o futuras. Observando los atributos comunes y duplicando las responsabilidades.

A continuación se presenta un ejemplo de la jerarquía del dibujo de elementos, primero se identifican cuales son las clases abstractas y cuales las concretas. En el ejemplo solo existe una clase abstracta : Elemento de dibujo. Todos los demás representan un elemento de un dibujo que puede ser creado. La clase jerárquica para elementos de dibujo se muestra en la siguiente figura :

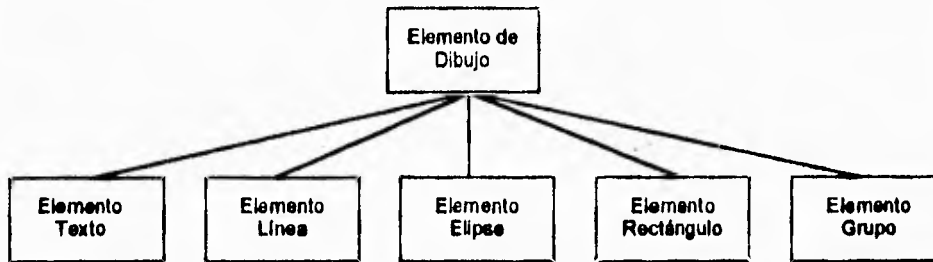


Fig. 3.8.8 Jerarquía de un elemento de dibujo.

Posteriormente se observan las responsabilidades distribuidas entre las clases. Si observa que Elipses, Rectángulos y Líneas son responsables de mantener dos atributos: el ancho y el color de la línea con la cual son dibujados o delimitados. Aunque los Elementos Línea, Elipse y Rectángulo provienen de la misma superclase se encuentran junto con otras subclases que son Elemento de Grupo y Elemento de Texto, los cuales no soportan estas responsabilidades. Por lo tanto no es apropiado agregar las dos subclases a la superclase Elemento de Dibujo.

Podemos, por lo tanto, crear una clase abstracta que represente un elemento dibujado con una o más líneas: Elemento Lineal. El Elemento Lineal es una clase abstracta con tres subclases concretas: Elemento Línea, Elemento Elipse y Elemento Rectángulo. Otra posible subclase del Elemento Lineal podría ser el Elemento Polígono. Los cambios mencionados se presentan en la siguiente figura :

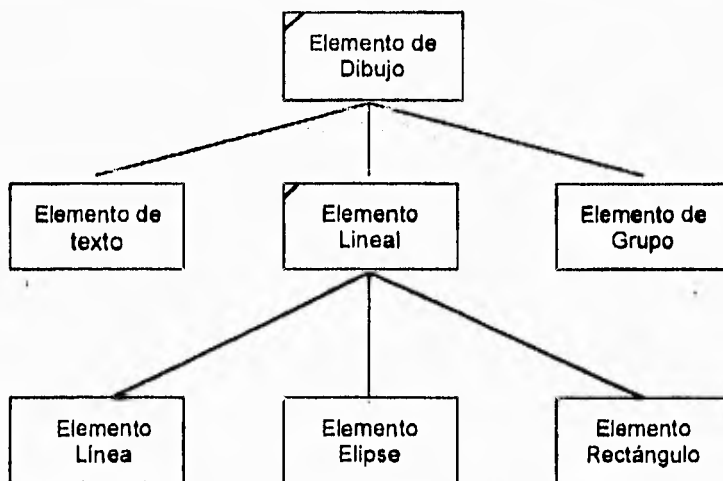


Fig. 3.8.9 Jerarquía de elementos revisada.

Pueden existir más responsabilidades, las elipses y los rectángulos tienen adicionalmente, asociada la característica de ser llenados con un color. Por lo tanto podemos crear una clase abstracta llamada Elemento Lleno, para representar todos los elementos que pueden tener algún color. Elemento lleno se convierte en una superclase de Elemento Elipse, Elemento Rectángulo y una subclase de Elemento lineal. La jerarquía de Elemento de Dibujo queda como sigue:

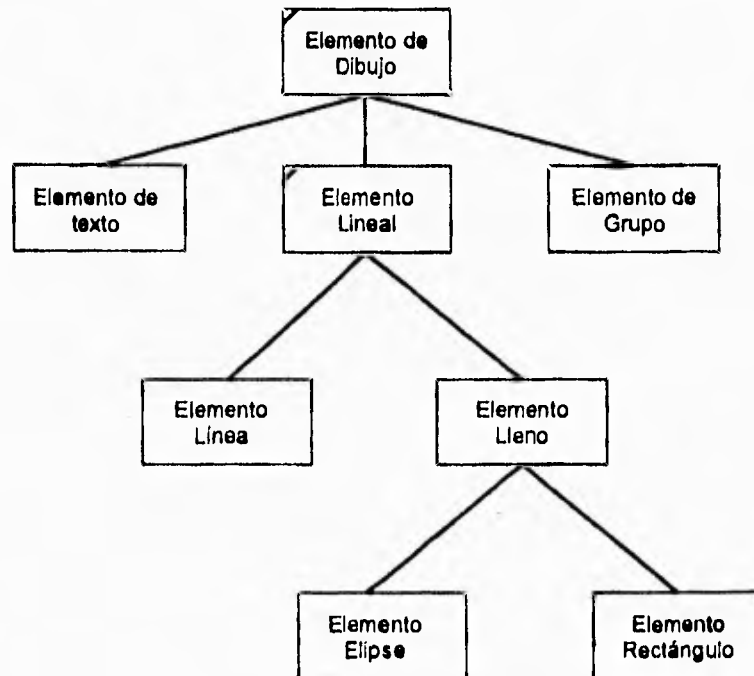


Fig. 3.8.10 Jerarquía final de Elemento de Dibujo.

El principal beneficio de diseñar cuidadosamente las jerarquías con clases abstractas se presenta cuando se desea agregar una nueva funcionalidad a la aplicación.

Una gran parte del funcionamiento puede ser reutilizado si tiene el mayor número posible de clases abstractas.

ASEGURARSE DE QUE LAS CLASES ABSTRACTAS NO HEREDEN DE CLASES CONCRETAS

Las clases abstractas, por naturaleza soportan sus propias responsabilidades en forma independiente. Por lo tanto nunca podrán heredar de clases concretas. Si el diseño tiene tal relación, se puede resolver el problema creando una clase abstracta de la cual heredarán el funcionamiento la clase concreta y la clase abstracta.

ELIMINAR CLASES QUE NO AGREGUEN FUNCIONALIDAD

Las clases que no tengan responsabilidades pueden ser eliminadas.

La jerarquía de clases muestra todas las clases dentro del sistema, de tal forma que las relaciones, superclase y clases derivadas sean fácilmente trazadas. La definición de la jerarquía es muy importante para determinar los elementos y los métodos que una clase hereda de una superclase.

3.9 OLE UN NUEVO ENFOQUE DE LA ORIENTACIÓN A OBJETOS

En un principio las aplicaciones eran totalmente independientes, interfaz gráfica esquemas de almacenamiento propietario. Así para generar un reporte en *WordStar* otro procesador de palabras con información cualquier hoja de cálculo era necesario volver a teclear toda el texto en cada aplicación.

Posteriormente los programas empezaron a contar con capacidad de exportar importar información a diferentes formatos preferidos por el usuario, o que de alguna manera representaban un formato de los más comunes. Con ello se solucionaba problema de transferir información, pero la forma de hacerlo era difícil además de contar con muchas opciones.

Con la llegada de la interfaz gráfica se puso en énfasis en la inter-operatividad. Así con Microsoft Windows y otras interfaces nació el concepto de 'Cortar y Pegar' lo que se hacía con el uso del portapapeles sin necesidad de grabar a disco, esto representó una mejora significativa en la transferencia de datos entre aplicaciones.

Pero desafortunadamente no es muy eficiente en algunos casos. Si se tiene un programa que almacena las cotizaciones de acciones y se actualiza muchas veces al día, cada vez que cambie el dato original será necesario actualizar la información dependiente de esos datos, ahora si lo ubicamos dentro de una hoja de cálculo el problema se extiende al número de celdas con información que se tengan, lo que representa mucho trabajo y poca eficiencia.

Para dar solución a ello se origina el DDE (Dynamic Data Exchange - Intercambio dinámico de datos), este esquema mejoró mucho y dio solución a algunos de los problemas que se tenían pero es frágil y dependiente de la estructura en disco de archivos y aplicaciones.

Así surge el concepto de OLE en su primera versión, que permite la inserción de objetos foráneos en diversas aplicaciones, si se cambia la estructura del disco, el proceso actualizará todo de forma automática.

Actualmente OLE en su segunda versión es el más reciente método para interoperatividad de aplicaciones y en el futuro de explotación que se requiera de las mismas. El concepto es mucho más amplio que todo lo existente a la fecha en cuanto a interoperatividad se refiere.

Sin lugar a dudas la tarea de construir aplicaciones hoy en día no es una tarea fácil de realizar. Los cambios que a diario ocurren en el marco tecnológico nos hacen reflexionar sobre la elección de herramientas cada vez más productivas, que nos permitan realizar de una forma sencilla nuestras tareas cotidianas.

OLE es una tecnología desarrollada por Microsoft que engloba servicios de intercambio de información, también ofrece extensiones al sistema operativo Windows a través de diferentes componentes. Object Linking and Embedding (OLE) Versión 2 es una tecnología que habilita a los desarrolladores para crear aplicaciones sofisticadas, así como también crear **extensiones** para otras aplicaciones a través de múltiples plataformas. Esta tecnología conforma el puntal en forma de modelo de programación para futuras versiones de la familia de sistemas operativos Windows. OLE es el primer

componentes independientes (Por ejemplo Excel es un objeto que expone 128 métodos que pueden ser explotados por otras aplicaciones o componentes).

Uno de los objetivos hacia el usuario que pretende OLE es que pueda utilizar aplicaciones de una forma sencilla e intuitiva. El centro de atención del usuario es la manipulación de documentos, de esto parte la creación de una metáfora que permita al usuario utilizar aplicaciones centradas al documento. El usuario desea obtener documentos y es ahí donde deben ayudar las aplicaciones de hoy en día. Se ha comprobado en laboratorios de diferentes compañías y en grandes corporativos que los usuarios tardan más tratando de aprender cierta herramienta que en realizar su trabajo cotidiano (absurdo, la computadora debe ser una herramienta no un objeto de distracción). Con OLE los usuarios podrán crear documentos compuestos de información que se encuentran en diferentes formatos, y centrar su atención directamente en la información, y ya no en la aplicación que obtiene sus datos. Los datos pueden ser mezclados o añadidos (embedded) y ligados (Linking) dentro del documento, así sólo una referencia a los datos son almacenados dentro del documento.

OLE facilita la integración de aplicaciones a través de la definición de un estándar de interfaces (Conjunto de funciones), agrupando semánticamente las funciones a través de las cuales una aplicación accesa servicios de otra. El concepto de exponer funcionalidad a través de interfaces hace a OLE un sistema abierto y extensible. Es abierto en el sentido de que cualquier aplicación puede proporcionar una implementación definida en una interface y cualquier otra aplicación puede utilizarla

para sus propios fines. Es extensible ya que nuevas interfaces pueden ser definidas o integradas dentro de las aplicaciones sin requerir cambios al código existente.

OLE es un conjunto de protocolos que tienen los programas Microsoft para permitir la interacción de unos con otros, todavía es considerado como una opción alternativa en la creación de aplicaciones, existen diferentes herramientas que pueden utilizarse en la realización de una aplicación que tenga estas características. El integrar a un sistema los conceptos de OLE es de suma importancia ya que los usuarios conocen de alguna manera estas características que se encuentran disponibles en las utilerías que vienen con Windows (Paintbrush, Write, ...) así como en las aplicaciones más comunes (Excel, Word, ...).

Con OLE debe cambiarse la forma en que se programa y ello no implica que deba desecharse el conocimiento acumulado con el que se cuenta, lo importante es encontrar el punto de equilibrio en la combinación de la tecnología OLE y tecnología API para el desarrollo de aplicaciones orientadas a objetos dentro de un ambiente operativo como lo es Windows.

OLE provee de un conjunto de características para el sistema independientemente del lenguaje en el cual se este trabajando, dichas propiedades son denominadas 'Objetos Windows' en lugar de utilizar las funciones tradicionales de programación (API). Básicamente un Objeto Windows es una parte de código que contiene sus propias funciones a través de uno o más grupos distintos de funciones, cada grupo es llamado interface y cada uno de ellos provee de diferentes niveles de operación, es decir, en lugar de trabajar con un conjunto de funciones con apuntadores, se estará programando con diferentes objetos, cada uno diseñado para funciones muy

específicas con sus propias funciones integradas. El primer paso que debe hacerse para adoptar esta nueva tecnología es entender que uno no solo debe aprender a utilizar una forma instantánea sino que es necesario familiarizarse con algunos conceptos para después poder integrarlos en una aplicación.

OLE está compuesto de un conjunto de servicios que pueden ser vistos en dos grupos. En el primer grupo el de menor nivel se contienen los servicios de infraestructura, los servicios básicos que proporcionan los medios por los cuales las características pueden ser implementadas o utilizadas. La infraestructura de servicios incluye: *negociación de interfaces, manejo de memoria, reporte de errores y estado, comunicación entre procesos, almacenamiento estructurado y transferencia de datos*. Las características de las aplicaciones son aquellos servicios que benefician al usuario final, las características son colocadas en el nivel superior de la jerarquía de OLE. Algunas de estas características que se encuentran disponibles hasta el momento incluyen *documentos compuestos, in place activation, programmability, drag and drop*. OLE se encuentra basado en tecnología orientada a objetos, sin embargo no cumple con todos los lineamientos que esta exige, OLE es un buen intento inicial hacia sistemas operativos totalmente orientados a objetos. Existen otras especificaciones que si se apegan totalmente al estándar OOP, sin embargo o no se encuentran en el mercado o son implementaciones aun pobres y sin aplicaciones verdaderamente útiles disponibles.

Sin lugar a dudas el éxito de un producto de computo en la actualidad se mide por el nivel de ventas que este produzca a sus creadores, OLE es una tecnología que permite integrarse a un nuevo nicho que en lo futuro tendrá diferentes ramas. Hoy en día

nuestra debemos pensar en los beneficios que se obtienen al ir desarrollando con OLE es decir "Construir aplicaciones a través de componentes". Hoy en día se presenta esta oportunidad de negocio, tendiente hacia la automatización de oficinas.

Office 4.3 de Microsoft integra los primeros productos del gigante que sacan provecho de OLE. Son productos altamente integrados que hoy contienen una nueva herramienta de desarrollo VBA (Visual BASIC for Applications), disponible al momento dentro de Excel y Project. El cual es altamente re-utilizable. Pensemos en la idea de colocar proyectos en los cuales integremos información a través de una aplicación desarrollada con un objetivo específico y Excel, Word, Project, Mail y muchos otros productos, sería de impacto ante un cliente. Durante un tiempo esto se volverá un nicho interesante, antes de que se presente un abaratamiento de los presupuestos de los proyectos (Por abaratamiento de costos de especialistas) sin embargo mientras esto sucede se genera un oportunidad de negocio.

Una anotación importante: algunas tareas se facilitan (Programación), sin embargo la capacidad creativa es un factor importante para la venta de negocios, esto es un factor que no debe ser perdido de vista. En algún comentario escuchamos que los costos de la fuerza de trabajo se abaratan (Hoy en día más gente conoce Visual BASIC), pero aún así existen y existirán factores que hacen la diferencia, como son Calidad, Disponibilidad y premura con que se realice un proyecto.

Office y los productos que la integran se vuelven una herramienta de apoyo, en lo futuro Microsoft pretende que se vuelvan una herramienta esencial para realizar programas de computo. Estos productos significan la mayor cantidad de ingresos del gigante verde, y de ahí su interés en hacerlos crecer.

Bien se abre una nueva generación de aplicaciones, por lo cual debemos entrar a la creación de una nueva área de consultaría que este basada en aplicaciones OLE y además proporcionar las ventajas de esta tecnología en las aplicaciones desarrolladas para el uso normal en un mercado específico como lo es la construcción ya que Windows centra su futuro en OLE.

PROPÓSITO DE OLE

La idea de este concepto es que las aplicaciones no deben existir. Actualmente se pueden generar documentos con tablas y gráficas en cualquiera de las aplicaciones que forman *MS.OFFICE*. El problema es que las aplicaciones han crecido mucho por la necesidad de soportar muchos tipos de datos y distintos. En el caso de *EXCEL* tiene implementaciones mucho más complejas de lo que necesita una simple hoja de cálculo, sabe de palabras y de gráficas, lo cual lleva a notar que las aplicaciones cuentan con funcionalidad redundante, y que opera de manera distinta para el usuario.

El propósito es utilizar una hoja en blanco y dentro de él se arrastrará un rectángulo y se indicará con que propiedades, gráficas, vídeo, etc. Las aplicaciones pasarán a ser componentes altamente especializados, y no sólo las aplicaciones Microsoft sino también aquellas de terceros que tengan el soporte necesario.

Este concepto esta denominado como interface, porque es la manera en que el usuario cubrirá sus necesidades, esta interface recibe el nombre de 'interface centrada en documentos'.

CARACTERÍSTICAS GENERALES DE OLE

El objetivo de OLE es permitir al usuario la interacción de diferentes aplicaciones de una forma más poderosa y sencilla. Un documento puede contener diferentes 'objetos', es decir, datos que pertenecen a otras aplicaciones y el usuario podrá editar estos datos desde el documento activo.

Las aplicaciones que soportan OLE se dividen en dos categorías diferentes:

Las aplicaciones cuyos objetos pueden ser ligados desde otra aplicación son llamadas 'servidores' o 'servers'. *Paintbrush* y *Sound Recorder* utilerías que forman parte de Windows se encuentran en esta clasificación.

Las aplicaciones que aceptan contener objetos de otras aplicaciones son llamados 'clientes' o 'clients'. *Write* y *CardFile* que son algunos de los programas que forman parte de Windows son clasificados en este tipo.

Para cambiar la información que se encuentra ligada dentro de un documento no es necesario borrar y copiar nuevamente. Ahora es posible abrir la aplicación a la que pertenece la información desde el documento que la contenga, editar, modificar y automáticamente la información será modificada en el objeto del documento.

Existen un conjunto de funciones en las librerías de Windows que proveen las funciones fundamentales que el programador puede utilizar, los objetos componen es un 'Objeto Windows' especial que es identificado por todas las aplicaciones. Cuenta con una tabla de apuntadores que permite hacer llamadas de funciones entre dos aplicaciones distintas. Este mecanismo crea una reacción de los objetos que reciben los mensajes, independientemente del lenguaje en que este desarrollada cada una de las aplicaciones. Un objeto Windows no siempre requiere ser estar estructurado como un objeto componente para compartir las funciones de comunicación.

Otro punto de suma importancia en la implementación de Objetos Componente y es el uso de las funciones llamadas "marshaling" que son aquellas que controlan el paso de parámetros y las llamadas de funciones entre dos objetos de diferentes aplicaciones.

SERVICIOS QUE OFRECE OLE

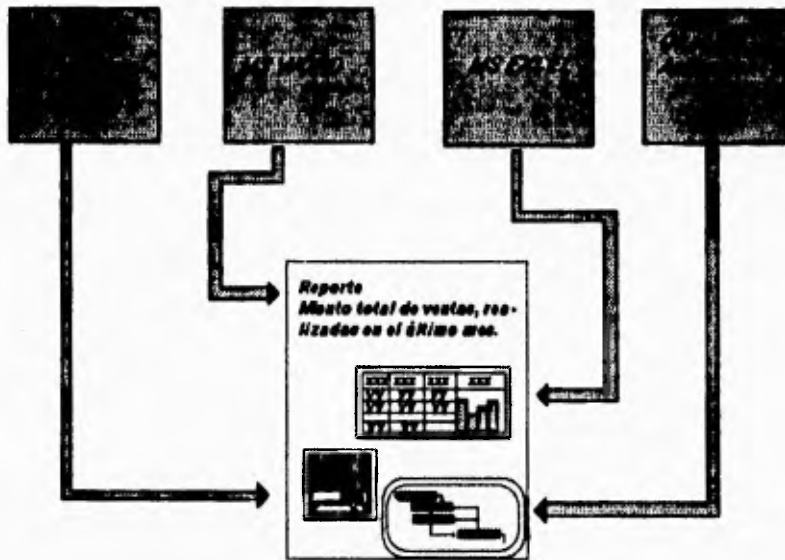
DOCUMENTOS COMPUESTOS

Se define un documento compuesto como aquel que contiene al menos un objeto que pertenece a otra aplicación. Es importante recordar que los documentos compuestos pueden contener n objetos de n aplicaciones distintas. En la segunda versión de OLE se cuenta con ventanas de ayuda al usuario que permiten entender mejor los conceptos y trabajar con ellos de forma más sencilla.

ALMACENAMIENTO ESTRUCTURADO

Las especificaciones de OLE proveen un número relacionado de interface que permite n de manera colectiva un mejor almacenamiento información, estas funciones son llamadas Almacenamiento Estructurado. Por la definición de interface, no se trata de una implementación, sino que se describe la forma en que debemos crear un archivo de sistema dentro de otro archivo, esto nos da gran poder desde el punto de vista técnico y proporciona características muy particulares a nuestra aplicación. En lugar de requerir de un continuo almacenamiento de secuencias en un archivo nos permite manejarlo como un apuntador que puede moverse de manera muy sencilla y obtener o grabar la información necesaria sin que ello requiera de algoritmos complejos. Estos archivos son llamados también archivos compuestos, ya que almacenan diferentes estructuras de datos

DOCUMENTO COMPUESTO



DRAG & DROP ENTRE APLICACIONES

El concepto de Drag-Drop para mover la información de forma visual ha sido muy bien aceptado por los usuarios, para quien es más sencillo hacerlo así que a través de menús. Esta ventaja ofrecida por OLE hará que el concepto 'cortar-pegar' tienda a desaparecer.

AUTOMATIZACIÓN

OLE proporciona la infraestructura para la Automatización en un nuevo concepto para construcción de aplicaciones. Para entenderlo es necesario pensar en Automatización OLE como un concepto completamente aislado de los elementos descritos anteriormente de OLE.

La automatización provee el soporte para explotar la funcionalidad de todas las aplicaciones, es posible considerarlo como 'pegamento de aplicaciones' o como un Macro Lenguaje muy evolucionado.

La situación actual es la siguiente: si suponemos que existe un programa procesador de palabras y otra de nuestras aplicaciones requiere de una revisión ortográfica tenemos tan solo 3 opciones muy claras de hacerlo, las cuales son:

- *Desarrollar nuestro diccionario y rutinas de verificación.*
- *Comprar una librería y 'ligarla' a nuestro programa.*
- *Utilizar DDE para controlar el procesador de palabras con esa funcionalidad.*

La primera solución sería muy tardada, tediosa y por lo tanto costosa. La segunda opción es buena pero dependemos de un producto no muy utilizado en el mercado, además de que no sabemos si se encuentra disponible para el lenguaje con el que estamos trabajando, tomando en cuenta que deberá existir un tiempo para conocer dichas funciones y librerías. La última de las opciones requiere de una manipulación muy indirecta del procesador de palabras, por lo que no se tiene forma de controlar

completamente los errores y puede ser un poco tardado, incluyendo a un programador con mucha experiencia.

De esto deducimos que se requiere de un esquema flexible que ofrezca una solución flexible, robusta y sencilla para ser implementada por programadores sin mucha experiencia.

La solución de Automatización OLE es que cada aplicación expone objetos hacia exterior. Estos objetos pueden ser manipulados fácilmente por un lenguaje como Visual Base para Aplicaciones.

Automatización OLE tiene otro elemento de gran importancia en lo referente a tareas muy repetitivas, las aplicaciones también vivieron una evolución. Por ejemplo se inició con un lenguaje de 'Macros' que algunas aplicaciones consideraban y que muchas veces únicamente servía para esas aplicaciones. Ahora será posible utilizar objetos con diversos administradores simultáneamente lo cual representa un cambio de paradigma en la manera de crear aplicaciones.

EDICIÓN VISUAL

Cuando se tiene un objeto foráneo insertado en una aplicación, bastará con seleccionarlo con el Mouse para que sea ejecutada en ese momento la aplicación que maneja ese tipo de objetos. Esta es una de las grandes ventajas que ofrece OLE dependiendo de las características que tenga la máquina utilizada será el tiempo de respuesta de este cambio continuo de aplicaciones.

TRANSFERENCIA DE DATOS Y NOTIFICACIÓN

Este concepto nos da la funcionalidad para representar toda la información necesaria en una transferencia (Drag & Drop, DDE, OLE) a través de una simple parte de código llamada Objeto Dato. Estos objetos no están restringidos para el paso de información a través de espacios de memoria globales, es posible hacer este intercambio de información a través de archivos compuestos, es decir, el programador podrá escoger los métodos apropiados para dar a la aplicación el mejor performance posible.

La notificación se presenta al cambiar un conjunto de datos pertenecientes a un documento compuesto, es decir con solo cambiar los datos originales OLE indicará a las demás aplicaciones que se ha presentado una modificación y es necesario considerarla dentro de las características del documento compuesto.

Al hacer un análisis de las características mencionadas anteriormente, nos damos de cuenta de la importancia que representa tanto para los usuarios finales como para los desarrolladores. Básicamente nos permite centralizar las características de un documento y no de una aplicación lo cual es más eficiente para ambos. Dentro de las ventajas que nos proporciona se encuentra que cada aplicación es responsable de los datos en los objetos de su tipo, por lo que el desarrollador no deberá manejarlos.

El uso de OLE y las interfaces que se proporcionan a los desarrolladores permiten que las aplicaciones sean flexibles, fácil de mantener y de mejorar.

3.10 ESTRATEGIA DE SOLUCIÓN

Una vez presentadas las características básicas de la programación orientada a objetos, es necesario definir cual será la estrategia de solución que debemos tomar por ello se han definido algunos puntos básicos de las características que el sistema debe cubrir, en cuanto a las ventajas tecnológicas.

El sistema de precios unitarios a desarrollar deberá estar orientado a cumplir con las características de un objeto Windows, deberá cumplir con los estándares de interfaz de funcionalidad y sobre todo administración de recursos de los objetos en windows, de manera tal que pueda ser accedido desde los demás programas y/o módulos.

Un sistema de precios unitarios consta de diferentes módulos los cuales cumplen con diferentes objetivos específicos, a su vez, cada uno de ellos debe compartir información con los demás módulos.

A continuación se mencionan los módulos del sistema de precios unitarios que deben manejarse de forma independiente y que deben compartir recursos y los datos proporcionados por los usuarios para el presupuesto:

MÓDULO DE PRECIOS UNITARIOS Y PRESUPUESTOS

En este módulo se contemplan básicamente las opciones para la elaboración de los presupuestos, tales opciones serán la sección de Insumos (Materiales, Mano de Obra, Equipo y Maquinaria, Subcontratos) así como también las ventanas de captura de

análisis (Básicos, Integrados, Partidas, Capítulos, Extraordinarios). Se encuentran también las opciones de configuración para el presupuesto como son : Actualización de Precios, Explosión de Insumos, Formatos de Sobre-costos, Catálogo de Monedas, Catálogo de Zonas.

Al estar considerando el desarrollo del sistema en un ambiente Windows se ofrece al usuario todas las ventajas de dicho ambiente como lo son *Drag and Drop* , *Cut and Paste* y sobre todo el manejo de varias ventanas.

El usuario tendrá la facilidad de abrir cuantas ventanas desee aún siendo del mismo tipo de insumo o análisis, esto es aprovechando las características de la programación orientada a objetos, donde cada ventana de captura representa un objeto-windows y tendrá su espacio de operación en la memoria de la computadora, permitiendo así la existencia de otros objetos del mismo tipo sin que interfieran unos con otros. Al hacer una comparación con el funcionamiento tradicional en un ambiente MS-DOS podemos notar que esta es una de las características más importantes que ayudan en la elaboración de un presupuesto, porque los usuarios podrán estar en la captura de análisis y automáticamente abrir una ventana para consulta o edición de otro análisis del mismo tipo o de algún insumo sin necesidad de cerrar la ventana que se encuentre activa, se buscará que el funcionamiento de estas utilerías no sea complicado de manera que el usuario pueda acceder la información de manera instantánea, estos son algunos de los puntos técnicos que debemos cubrir como una estrategia de

competitividad, pero el diseño de las ventanas, de los campos y su interacción con el usuario serán detalladas en el capítulo de diseño del sistema.

Los datos contenidos en estas ventanas de captura podrán ser copiados a través del portapapeles de windows hacia otras aplicaciones windows, esta es otra de las ventajas importantes, que probablemente le ayudarán al usuario a minimizar los tiempos de elaboración de un presupuesto.

MÓDULO DE REPORTES

Aprovechando las ventajas que ofrece Windows en el manejo de dispositivos de hardware (fax-modems, impresoras, CD-ROMS, Tarjetas de Audio) la generación de reportes se vuelve sumamente sencilla para el usuario y deberá estar enfocada a :

Preparación de Reportes acorde con las necesidades del usuarios.

Generación de reportes de forma inmediata y eficiente.

Ahorro en consumos.

Contando con un reporteador que nos permita conocer el reporte antes de ser impreso es de suma importancia, ya que ello nos ahorrará gastos de papel innecesarios, así como diseñar reportes que contengan toda la información requerida por la dependencia o la empresa a quien se entregará dicho presupuesto.

En el ambiente windows es posible enviar el reporte a impresión y continuar haciendo modificaciones o consultando otra información en el presupuesto, esto gracias a otra de las características de dicho ambiente (multitareas).

MÓDULO DE INTERCAMBIO DE INFORMACIÓN

En este módulo se consideran facilidades adicionales que representan la exportación / importación de datos con otras aplicaciones, cuyas estructuras de base de datos son diferentes a la que se van a utilizar. Lo ideal en este punto es intercambiar toda la información o solo la que el usuario desee, los mecanismos de copiado en el ambiente windows son de gran utilidad cuando el volumen de información es relativamente pequeño, pero si deseamos hacer una conversión de un número mayor de registros es necesario contar con una herramienta fácil de usar, práctica , eficiente y sobretodo rápida y este es el objetivo que el módulo de intercambio de información debe cubrir.



CAPITULO IV

DESARROLLO DEL SISTEMA

4.1 ESPECIFICACIONES DEL DISEÑO

4.1.1 DIAGRAMA DE FLUJO DE DATOS

Es importante considerar el flujo de información que se presenta al elaborar un presupuesto, dentro de un sistema de precios unitarios existen diferentes sub-módulos que se especializan en la validación de diferente información y que una vez aceptada se registra en la base de datos para así poder utilizarse en los demás sub-módulos.

Los sistemas de precios unitarios deben ser capaces de organizar, almacenar, separar y agrupar la información recibida, es necesario manejar tipos de control interno.

Al ofrecer velocidad de captura de información al usuario, se alteran los esquemas de orden tradicionales, sin que ello represente una desorganización en el almacenamiento de la información. Para el caso del sistema se manejan diferentes catálogos de insumos, análisis, monedas, familias que aún cuando comparten archivos físicos e incluso estructuras de datos, es muy sencillo identificar el tipo de cada registro a través de una clasificación de uso interno que el usuario no siempre conoce.

A continuación se presenta un análisis de la estructura global del sistema, que permite conocer la interrelación entre los datos de un módulo y otro. Se presenta por sub-módulo de forma que puedan identificarse los datos centrales de cada utilería y además se distinguen aquellos que son el resultado de otros procedimientos dentro del sistema.

Como base debemos recordar que el módulo de precios unitarios consta de los siguientes sub-módulos.

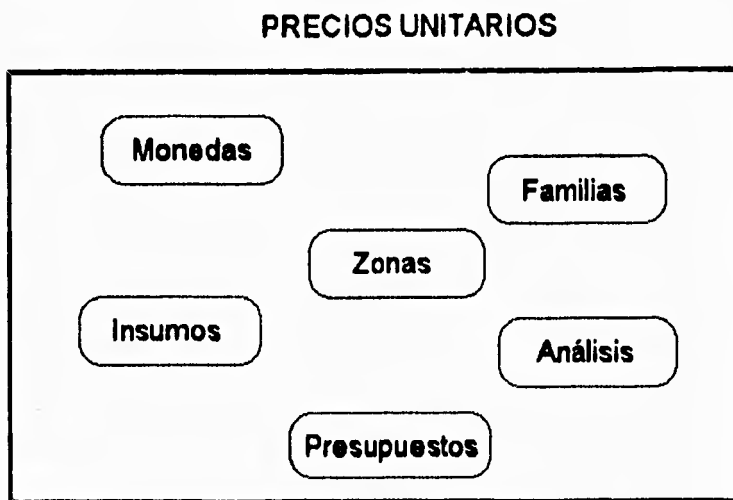


Fig. 4.1.1.1 Módulo de Precios Unitarios

MONEDAS

Existe un catálogo de monedas como una herramienta al usuario, que le permite manejar los costos de sus insumos en la moneda que lo desee, es necesario indicar al sistema cual será la moneda base para el presupuesto y de ahí dependerá la conversión a realizar. Este sub-módulo le brinda una amplia versatilidad al sistema, ya que le permite al usuario proporcionar un tipo de cambio de moneda diferente y en

base a ello actualizar los análisis elaborados. En muchas ocasiones, sobre todo para el caso de Maquinaria, se manejan costos en diferentes monedas y es muy práctico para el usuario realizar la conversiones al tipo de cambio actual en el momento de hacer los cálculos de actualización de precios.

Desde el punto de vista de la información, el catálogo de monedas es una tabla de referencia que guarda factores de conversión aplicables al generar reportes o presupuestos completos.

La información contenida en este catálogo es Descripción de Moneda, El símbolo que debe utilizarse, el factor de cambio respecto a la moneda base y una fecha que sirve de referencia para conocer cuando fue la última vez que dicho factor ha sido alterado.

FAMILIAS

Se presenta como un catalogo de referencia, en donde la única información registrada es una clave de identificación y la descripción, que muchas veces contiene las características principales de todos los insumos que pertenezcan a esta clasificación o categoría.

Las familias permiten a un usuario agrupar o identificar a un conjunto de insumos con características similares, es decir, es una relación de 1 a muchos con el catálogo de insumos, porque cada registro de familia puede estar asociado a n insumos. Esta relación es de suma importancia, ya que más adelante permitirá al usuario trabajar con sub-conjuntos de insumos dentro de su catálogo global.

Debe mencionarse que cada familiar puede tener dentro de sus elementos a insumos de diferentes tipos, es decir, si se estableciera un criterio de agrupación de insumos sin importar el tipo de estos, el sistema podría cubrirlo. Por otro lado es importante notar, que al ser únicamente una agrupación de elementos no se requiere de cálculo alguno por lo que no existen procedimientos que calculen resultados, únicamente existirán rutinas para la selección de registros de características particulares.

ZONAS

El concepto de zonas dentro del módulo de Insumos es sumamente amplio, las zonas pueden considerarse tal y como lo indica su nombre, para asignar diferentes precios a un insumo y cada precio asociado a una región geográfica, por ejemplo si hablamos de México, una zona puede ser Guadalajara, Monterrey, Sonora. Al asignar zonas se va armando el catálogo de zonas, en donde únicamente tenemos una descripción detallada del área que representa esa zona.

De las mayores ventajas que se pueden ofrecer es la flexibilidad del concepto de zonas, ya que si un usuario prefiere utilizar el concepto para aplicarlo a proveedores, lo cual puede hacerse de manera muy sencilla, en lugar de que cada precio se encuentre asociado a una zona, se asocia a un proveedor, de forma que ahora el usuario podrá almacenar todos los proveedores, para cada uno de sus insumos, e incluso agruparlos por proveedor, o utilizar los mismos proveedores de diferentes insumos y como resultado estará generando un catálogo de proveedores en lugar de zonas. Cabe

mencionar que en el caso de que el usuario desee manejar ambos conceptos lo puede hacer, la recomendación es organizar la información en base a la claves.

Al asignar los diferentes precios es necesario que exista una relación entre el precio y la zona, pero además cada precio estará asociado también a un Insumo, es decir un Insumo contiene n precios asociados pero sólo uno de ellos podrá estar activo. Esto obliga a una relación de un registro de insumos con múltiples registros de precios.

INSUMOS

En los párrafos anteriores se han mencionado los catálogos auxiliares para la generación del presupuesto, ahora debemos detallar el funcionamiento para los registros de insumos. Puesto que los insumos se clasifican de acuerdo a su tipo, debemos analizarlos por separado para lo cual se explica a continuación la forma en que interactúa un registro de insumo con los demás catálogos.

MATERIALES

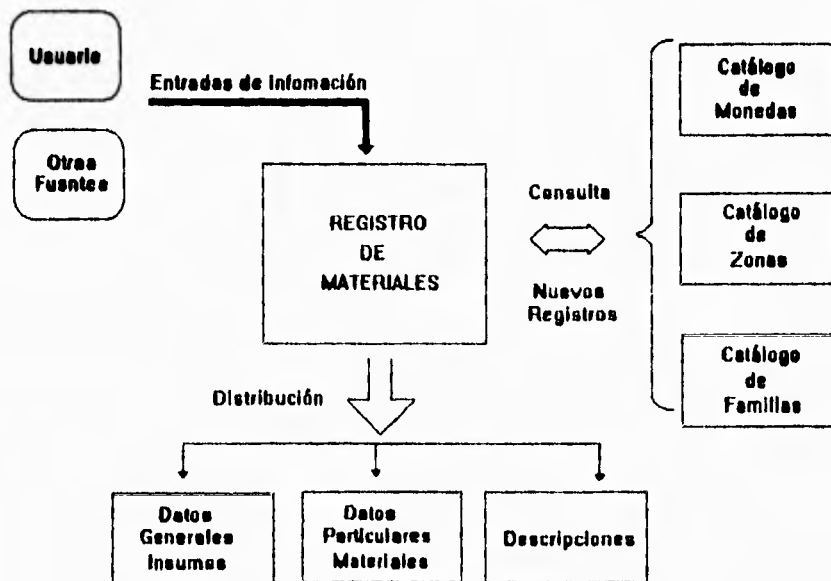


Fig. 4.1.1.2 Información en Materiales

Se definen las entradas de información por parte del usuario, que será una de las principales, pero no debe dejarse a un lado aquellas que se reciben a través de compañías dedicadas a la investigación de mercados, para ello es importante siempre considerar algunos mecanismos de captura de esta información.

De cualquier forma, el flujo de información debe darse de acuerdo a como la figura lo muestra, tomando en cuenta que la primera parte es similar para todos los registros de insumos y funciona de la siguiente forma:

El usuario debe proporcionar una clave que será el principal identificador del registro en todo el catálogo, así como también deberá proporcionar la descripción del elemento, la fecha de registro, fecha de cálculo, y podrá asignar una familia, para ello se basará en el catálogo existente o podrá generar un nuevo registro de familia.

El conjunto de información anteriormente explicado se refiere a los datos generales de los insumos, y son los mismos para todos los tipos de insumos, y existe una segunda categoría de datos la cual se conoce como los datos particulares y son aquellos que son específicos para un tipo de insumos, a continuación veremos los correspondientes a los materiales:

Existe un campo de Zona, en el cual el usuario podrá utilizar la clave de un registro existente en el catálogo y/o podrá agregar uno nuevo. Un registro de materiales solo necesita ahora un costo base, un porcentaje a aplicar que corresponde al flete de material y además se permite capturar un porcentaje de descuento para aquellos casos en que así se requiera. El usuario puede proporcionar todos los diferentes precios que desee y un flete y un descuento diferentes para cada uno de estos precios, pero no debe olvidarse que sólo será posible manejar un precio como el activo.

MANO DE OBRA

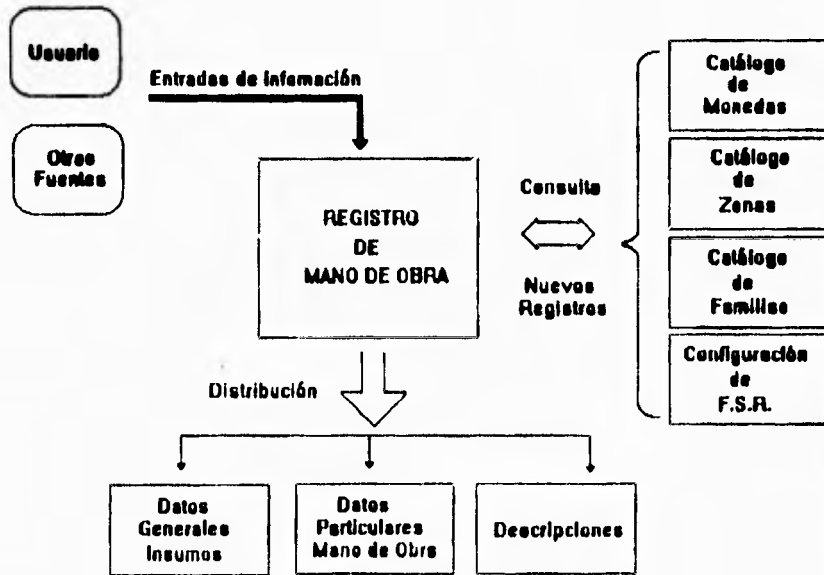


Fig. 4.1.1.3 Información en Mano de Obra

Los datos generales funcionan de manera similar a los materiales. Aquí mencionarán las características para los datos particulares.

Existe un tabla en la cual se almacenan todos los datos particulares para mano obra, los datos que se almacenan son: el salario base, el factor de salario real; c estos datos es posible calcular el costo unitario para cada categoría de mano de ob ahora bien , el F.S.R. que debe aplicarse puede ser alguno de los siguientes:

F.S.R. para salarios mínimos.

F.S.R. para salarios mayores al mínimo.

F.S.R. otro factor.

Para el caso de los dos primeros el usuario podrá elegir cual de ellos desea utilizar, estos factores son los calculados y analizados en el módulo de F.S.R. dentro del sistema, para el caso de categorías especiales que requieren un factor diferente, el sistema permitirá al usuario que lo proporcione y ese dato será almacenado en la tabla de datos particulares para mano de obra.

MAQUINARIA

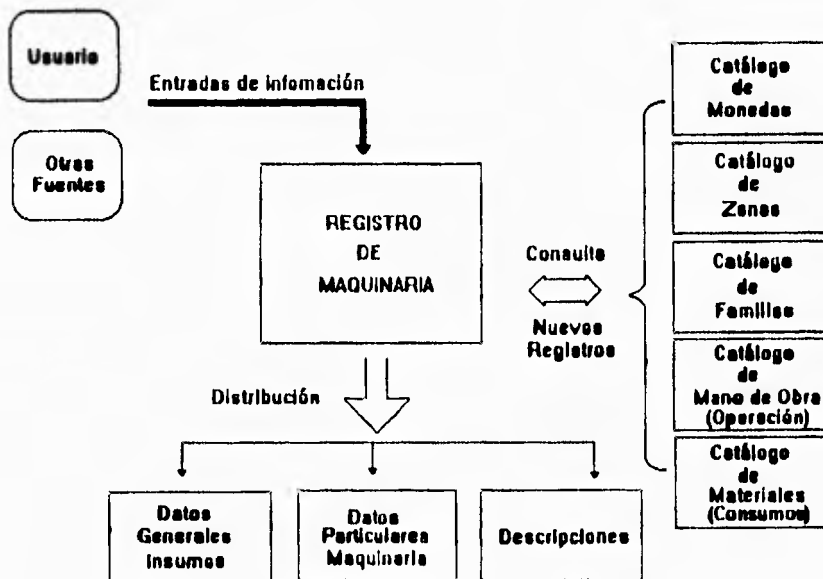


Fig. 4.1.1.4 Información en Maquinaria

En el procedimiento de captura de maquinaria y equipo se tiene demasiada información, ya que aquí debe realizarse un análisis completamente detallado del costo por hora de cada equipo a utilizar. Los datos generales se capturan y almacenan de la misma forma que para materiales.

Para los datos particulares existe un archivo especial para almacenar únicamente registros de maquinaria, un análisis de maquinaria requiere de los siguientes datos:

Datos Generales.

Cargos Fijos (almacenamiento, depreciación, seguros).

Consumos (combustibles y lubricantes).

Gastos de Operación (mano de obra para operación).

Los datos y factores para el cálculo de cargos fijos deben ser proporcionados por el usuario, en el caso de los consumos es necesario consultar el catálogo de materiales e indicar cuales son los combustibles y en que cantidades se utilizan, en ocasiones es importante que si un registro no existe en el catálogo de materiales deberá darse de alta en ese momento. Para el conjunto de datos correspondientes a gastos de operación sucede lo mismo, sólo que en este caso debe solicitarse al usuario las cantidades de mano de obra, las cuales incluso podrán manejarse como cuadrillas, es necesario hacer análisis preliminares.

SUBCONTRATOS

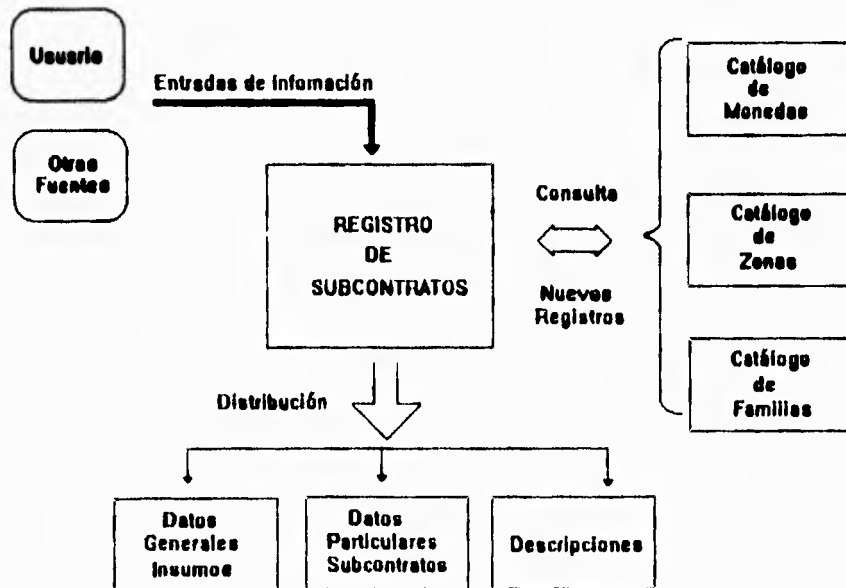


Fig. 4.1.1.5 Información en Subcontratos

El caso de Subcontratos es muy similar a materiales, existen los datos generales que se consultan de la misma forma que los demás y existen los datos particulares. Los datos correspondientes exclusivamente a subcontratos son el nombre del proveedor del servicio y el costo, al igual que en los demás tipos de insumos se pueden asignar precios diferentes asociados a una determinada zona e independientemente de moneda en que se quiera cotizar.

ANÁLISIS

Existen diferentes niveles de análisis, cuyo objetivo es diferente, algunos se utilizan específicamente para un tipo de datos, pero su funcionamiento interno es similar, en todos los niveles se relacionan con la misma información.

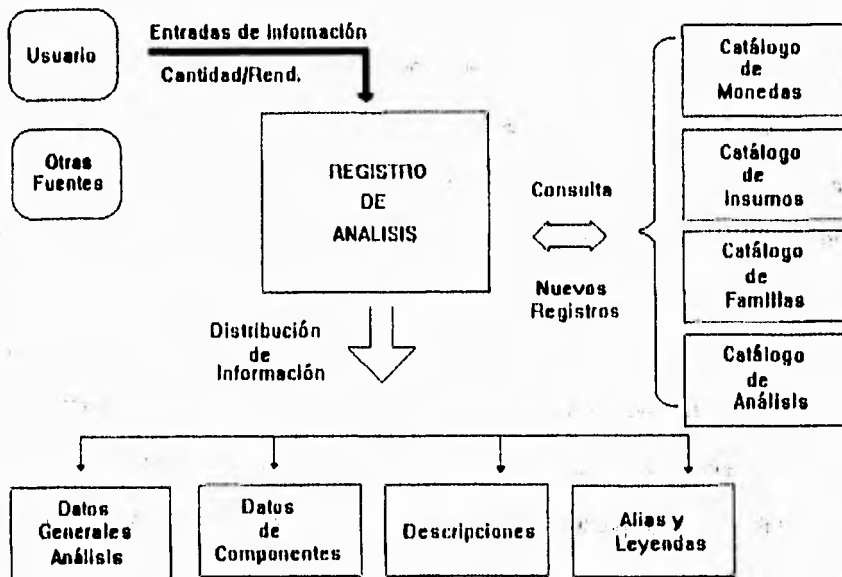


Fig. 4.1.1.6 Información en Análisis

Partiendo de que ya se tiene un catálogo armado de insumos, el usuario puede comenzar a generar un catalogo de matrices o análisis, y la información que debe agregar es la siguiente:

Datos Generales

Contenido del Análisis

Información Adicional.

Al igual que en insumos, existen datos generales que son la principal fuente de identificación de los mismos, se tiene una clave, una descripción, una asociación con familia, fecha de registro, fecha de cálculo, moneda en la cual se elaboró el análisis.

La información particular para cualquier registro de análisis (X, C, P, I, B) consisten en una tabla de relación entre el identificador del análisis y el identificador del componente, a lo cual sólo debe asignarse una cantidad y el rendimiento del precio, cada registro de insumo o análisis puede estar como elemento de uno o más registros de análisis. Además cada registro de análisis podrá tener n elementos hijos.

Para la elaboración de presupuestos se utiliza el nivel de análisis correspondientes a partidas, el funcionamiento es similar y sólo cambian algunos formatos de impresión.

Con el objetivo de ilustrar los conceptos anteriores, a continuación se presenta un diagrama con el seguimiento que debe hacerse en la elaboración de un presupuesto.

En el ejemplo se aplica la idea a un análisis en particular pero el funcionamiento debe ser el mismo para todos.

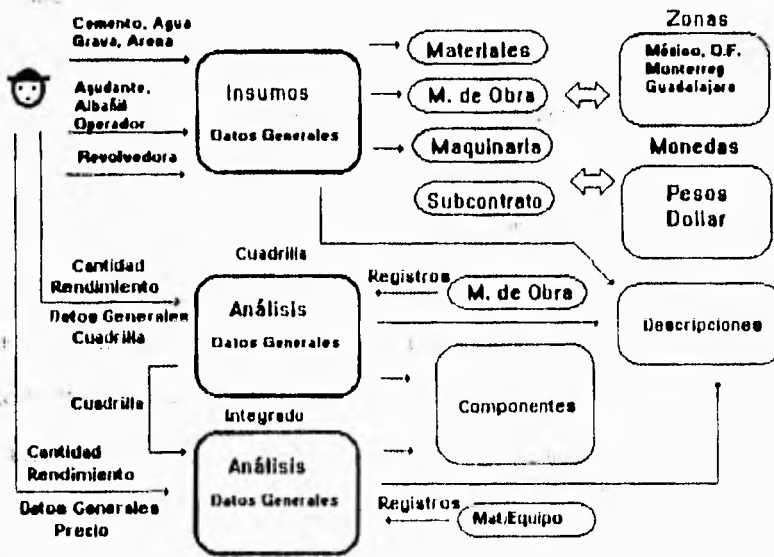


Fig. 4.1.1.7 Flujo de Información Global

En este ejemplo se puede observar de forma clara que un análisis puede ser elemento de otro, estos niveles de integración pueden presentarse al gusto del usuario, ya que el sistema no pone ninguna restricción. Aquí se muestra de alguna manera el flujo de información que siguen los datos proporcionados por el usuario.

4.1.2 DICCIONARIO DE DATOS

El diccionario de datos del sistema de precios unitarios es el elemento básico de referencia para localizar los nombres y atributos de los datos utilizados en todo el sistema. Deberá actualizarse cada vez que se hagan cambios.

MANEJADOR DE BASE DE DATOS RAIMA

El manejador de base de datos Raima ha sido diseñado mediante la combinación del modelo de red y el modelo relacional. Permite organizar y acceder información de forma eficiente independientemente de la complejidad de los datos. La combinación de ambas tecnologías proporciona rapidez y permite que se minimice la redundancia de los datos.

El manejador está escrito en lenguaje C, de tal forma que las aplicaciones son portables a una variedad de sistemas operativos como VMS, UNIX, Xenix, MS-DOS/2 y Microsoft Windows.

VENTAJAS DE LA COMBINACIÓN DE MODELOS

Los sistemas de bases de datos construidos mediante el modelo relacional han sido muy populares, principalmente por la facilidad de uso. Con el modelo de red,

4.1.2 DICCIONARIO DE DATOS

El diccionario de datos del sistema de precios unitarios es el elemento básico de referencia para localizar los nombres y atributos de los datos utilizados en todo el sistema. Deberá actualizarse cada vez que se hagan cambios.

MANEJADOR DE BASE DE DATOS RAIMA

El manejador de base de datos Raima ha sido diseñado mediante la combinación del modelo de red y el modelo relacional. Permite organizar y acceder información de forma eficiente independientemente de la complejidad de los datos. La combinación de ambas tecnologías proporciona rapidez y permite que se minimice la redundancia de los datos.

El manejador está escrito en lenguaje C, de tal forma que las aplicaciones son portables a una variedad de sistemas operativos como VMS, UNIX, Xenix, MS-DOS/2 y Microsoft Windows.

VENTAJAS DE LA COMBINACIÓN DE MODELOS

Los sistemas de bases de datos construidos mediante el modelo relacional han sido muy populares, principalmente por la facilidad de uso. Con el modelo de red,

principales beneficios son mejor funcionamiento, reducción de los requerimientos de almacenamiento, y tiene seguridad en la integridad de los datos.

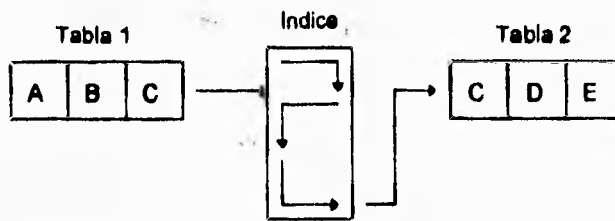


Fig. 4.1.2.1 El diagrama muestra dos tablas relacionadas mediante un campo en común (C). Notese que C debe ser definido en ambas tablas y que debe existir un índice para acceder las ocurrencias de las dos tablas relacionadas.

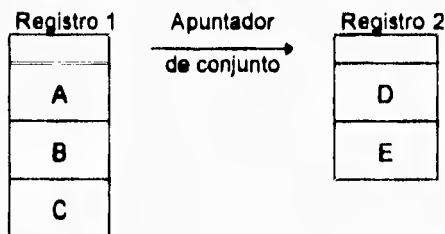


Fig 4.2.1.2 El modelo de red elimina la redundancia mediante la relación de dos tipos de registros, sin requerir de campos duplicados y archivos de índices. Además, el registro relacionado se accesa directamente con una operación de lectura de la base de datos.

MÉTODOS DE ACCESO DEL MANEJADOR DE BASE DE DATOS RAIMA

Indexado : Permite buscar una ocurrencia proporcionando la llave. La llave puede corresponder exactamente, en este caso se posicionará directamente en el registro, cuando es una correspondencia cercana, se posicionará antes del registro que contiene el valor mayor de la llave en la comparación.

Conjunto : Permite moverse a través de conexiones en varias direcciones. Se puede mover desde el dueño de un conjunto de instrucciones al primero o último miembro del conjunto.

Secuencial : Permite buscar la primera, última, siguiente o anterior instancia física de un tipo de registro. El manejador no permite insertar registros en un posición física específica de un archivo. El método secuencial es utilizado cuando se buscan los registros de un cierto tipo pero sin importar el orden.

ELEMENTOS DEL MANEJADOR DE BASE DE DATOS RAIMA

- **Diccionario** : Almacena información describiendo el contenido y la organización de la base de datos.
- **Archivos de datos (data files)** : Contiene ocurrencias de uno o más tipos de registros.

- **Archivos de índices (key files)** : Contiene un índice para uno o más campos llave (key fields). Los registros contienen campos de datos (data fields), campos llave (key fields) y conjuntos de conexiones (linkages) las cuales son transparentes al usuario.

DECLARACIÓN DE LA BASE DE DATOS

Sintaxis :

```
database dbname [ [pgsize] ] {  
    ddl statements ...}
```

La sentencia **database** es utilizada para especificar el nombre de la base de datos y opcionalmente el default del tamaño de página.

DECLARACION DE ARCHIVOS

Sintaxis :

```
data file [ [ pgsize ] ] [fileid] = filename contains rename [,rename] ...;
```

```
key file [ [ pgsize ] ] [fileid] = filename contains [rename. keyfld [,rename. keyfld] ] ...;
```

DECLARACIONES DE REGISTROS

[Static] record *recname*

{field_stmt}

 ...

{comkey_stmt}

La sentencia **record** define un grupo o campos relacionados llamado *recname* que serán accedidos y almacenados en la base de datos como una sola unidad. La declaración de registros consiste de cero o más campos seguidos de cero o más sentencias **compound key**.

El atributo *static* especifica que el registro nunca será modificado cuando la base de datos se encuentre abierta en modo protegido (*shared*).

Compound [optional] [unique] **key** *keyname* |

fldname [asc[ending] | desc[ending]];

Es utilizado para definir llaves compuestas. Las llaves compuestas se crean con una combinación de campos de un registro. Cada subcampo de la llave puede especificarse ordenado en forma ascendente o descendente.

DECLARACIÓN DE CONJUNTOS

```
Set setname {  
    Order {asc[ending] | desc[ending] | first | last | next }  
    Owner rename  
    Member rename [ by fldname [,fldname] ];
```

Define en forma explícita una relación de uno a muchos entre tipos de registros. Los conjuntos se implementan como listas ligadas de registros de instancias conectadas a instancias simples de un registro dueño, el cual sirve como raíz o cabeza de la lista. El orden en el cual se insertan los miembros se especifica en la cláusula **order** y sus posibles valores son :

first El nuevo miembro del registro es insertado al principio de la lista.

last El nuevo miembro del registro es insertado al final de la lista.

ascending El nuevo miembro del registro es insertado en forma ascendente basándose en el contenido de los campos especificados en la parte del **by** de la cláusula **member**.

descending El nuevo miembro del registro es insertado en forma descendente basándose en el contenido de los campos especificados en la parte del **by** de la cláusula **member**.

Next El nuevo miembro del registro es insertado inmediatamente después c miembro actual del conjunto, si el miembro actual es nulo, se coloca al principio de lista.

La definición de la base de datos del sistema de precios unitarios es la siguiente :

ARCHIVOS DE DATOS E ÍNDICES UTILIZADOS

Nombre y contenido	Descripción
data file[1536] INSUMOS_DAT InsuGen	Archivo para insumos
key file[1536] INSUMOS_KEY InsuGen.szClave	Archivo para llave de insumos
data file[1024] PART_MAT_DAT Materiales	Archivo para materiales
data file[1024] PART_OBR_DAT ObradeMano	Archivo para Mano de obra
data file[1024] PART_SUB_DAT SubContrato	Archivo para subcontratos
data file[2048] DESC_INSU_DAT DescInsumos	Archivo para descripción de insumos
data file[1536] EQUIPO_DAT MaquinaEquipo	Archivo para Maquinaria y equipo

key file[1536] INSDADD_KEY Materiales.DB_ADDRSZona ObradeMano.DB_ADDRSZona SubContrato.DB_ADDRSZona MaquinaEquipo.DB_ADDRSZona MaquinaEquipo.DB_ADDSCveOp1 MaquinaEquipo.DB_ADDSCveOp2 MaquinaEquipo.DB_ADDSCveOp3 MaquinaEquipo.DB_ADDSCveComb MaquinaEquipo.DB_ADDSCveLub MaquinaEquipo.DB_ADDSCveLubHidra MaquinaEquipo.DB_ADDSCveGasArr MaquinaEquipo.DB_ADDSClaveLlanEqAdi MaquinaEquipo.DB_ADDSClaveLlanMaqBase	Base de datos de insumos
data file[2024] LEY_DAT DescFam DescZona LeyInsu LeyAnali NotasAnalis DescEstima	Archivo de leyendas
data file[2048] ALIAS_DAT AliasInsumos AliasAnalisis	Archivo para alias de los datos
key file[2560] DESC_INSU_KEY DescInsumos.szDesc	Archivo para llaves de descripción de insumos

key file[3072] DESC_TIPO_KEY DescInsumos.TipDescInsumos DescFam.TipDescFam DescZona.TipDescZona DescEstima.TipDescEst	Archivo para llaves de descripción de tipos
data file[2048] ANALIS_DAT AnalisGen	Archivo para datos de análisis
data file[4096] COMPON_DAT Componentes	Archivo para datos de componentes
data file[2048] DIM_DAT AnalisDim	
key file[1536] CMPELEM_KEY Componentes.EleAnali	Archivo para llaves de componentes
key file[1536] ANALISIS_KEY AnalisGen.szClave	Archivo para llaves de análisis
data file[2560] FAMUNI_DAT Familia	Archivo para datos de familias
data file[2560] ZONA_DAT ZonaEco	Archivo para datos de zonas
key file[2560] ZONA_KEY Familia.szSeudonimo ZonaEco.szZona	Archivo para datos llaves de zonas
data file[2560] REGION_DAT Region Usuarios	Archivo para datos de regiones

key file[1024] REGION_KEY Usuarios.sNombreUsuario	Archivo para llaves de regiones
data file[2048] ESTIMS_DAT Estimaciones	Archivo para datos de estimaciones
data file[2048] ESTIMS_ELE_DAT ElementoEstimacion	Archivo para elementos de estimaciones
key file[2560] ESTIMS_KEY Estimaciones.szClave	Archivo de llaves de estimaciones
key file[2048] ELESTIMS1_KEY ElementoEstimacion.ElementoEst	Archivo para elementos de estimaciones
keyfile[2048] ELESTIMS2_KEY ElementoEstimacion.ElementoEst	Archivo para elementos de estimaciones

DEFINICIÓN DE REGISTROS Y CONJUNTOS

REGISTROS

Registro	Elemento	Tipo de Dato	Función
InsuGEN	dCostUnitario	Double	Costo Unitario
	IFechCosto	Long	Fecha de costo, está como Juliano
	IFechCosto	Long	Fecha de cálculo, está como Juliano
	ImiembroSelec	DB_ADDR	Número secuencial que le corresponde en el conjunto al insumo donde se tomará el precio para los cálculos
	IAliasActivo	DB_ADDR	Alias activo de este insumo

	iCurrency	int	Indica el currency que se utiliza como activo de este insumo. Si el currency no existe se toma el de default.
	ucBanderas	Unsigned char	Campo de banderas: 0 Se utiliza para saber si un elemento de este registro está capturado. Si está en uno el registro contiene toda su información, si está a cero el registro no está completo en toda su información.
	szUnidad	Char[11]	0 - 9 : Datos de la unidad 10 : Nulo
	szClave	Unique Char[28] key	0 : Tipo de insumo 1 : Tipo de base de cálculo para la herramienta, si el byte 0 es diferente a 'H', este byte debe contener un blanco y no un nulo. 2 - 26 : Clave alfanumérica 27 : Carácter nulo.
Materiales	dPrecio	Double	Datos particulares de materiales.
	fPorcFlet	Float	Porcentaje de flete Zona Eco.
	fPorcDesct	Float	Porcentaje de desc. Zona Eco. A
	icurrency	Int	Indica el currency que se utiliza como activo
	DB_ADDRZona	Key DB_ADDR	Dirección del registro de zona a la que pertenece el insumo.
	IFechaCotizacion	Long	Fecha de cotización del insumo.
ObradeMano			Datos particulares de Obra de Mano
	dFacSalReal	Double	Factor de salario real Elegido
	dSalaBase	Double	Salario base del trabajador zona eco. A.
	iCurrency	Int	Currency que se utiliza como activo.
	cSelecFSR	Char	Selección de Factor de salario real.
	DB_ADDRZona	Key DB_ADDR	Dirección del registro de zona a la que pertenece el insumo.

	IFechaCotizacion	Long	Fecha de cotización del insumo.
ibContrato	dCostoUnitario	Double	Costo unitario
	iCurrency	Int	Currency que se utiliza como activo.
	szProveedor	Char(80)	Proveedor sin blancos
	DB_ADDRSZona	Key DB_ADDR	Dirección del registro de zona a la que pertenece el insumo.
	IFechaCotizacion	Long	Fecha de cotización del insumo.
asclnsumos	iTipo	Int	Identifica que tipo de insumo o análisis es propietario de esta descripción.
	szDesc	Optional Key Char(51)	Descripción de los insumos.
	Compound Optional key TipDescInsumos (iTipo szDesc)		
maquinaEquipo	fValResct	Float	Valor de rescate
	dCostMaqBase	Double	Costo máquina base
	dCostEqAdi	Double	Costo Equipo Adicional
	fSeguro	Double	Prima de seguro anual
	fFlacAlmacj	Float	Factor de almacenaje
	fFacManten	Float	Factor de mantenimiento
	fInteres	Float	Tasa de interés anual
	fFacOpera	Float	Factor de operación
	dOtrasFuentes	Double	Otras fuentes de energía
	fFacComb	Float	Factor de combustible
	fFacLub	Float	Factor de lubricante
	fFacLubHidra	Float	Factor de lubricante hidraulico
	fFacGasArr	Float	Factor de gasolina para arranque cuando el motor es eléctrico, en este campo se guarda el factor de motor eléctrico.
	dFacObraMano	Double	Factor de operación de obra de mano.
	uTipoMotor	Unsigned Char	Tipo de motor : 0 - Diesel 1 - Gasolina 2 - Eléctrico

Capítulo IV Desarrollo del Sistema

	iCantLlaMaqBase	Int	Cantidad de llantas de máquina base.
	iCantLlaEqAdi	Int	Cantidad de llantas del equipo adicional
	fCantOp1	Float	Cantidad para operador 1
	fCantOp2	Float	Cantidad para operador 2
	fCantOp3	Float	Cantidad para operador 3
	fHorasDiarias	Float	Horas por turno
	fVidaLlant	Float	Vida de las llantas
	fHorasAnua	Float	Horas anuales
	fPotencia	Float	Potencia nominal
	fCapaCarter	Float	Capacidad del carter
	fTimeCambio	Float	Tiempo de cambio
	fVidaEcono	Float	Vida económica
	iCurrency	Int	Currency que se utiliza como activo.
	iTipoMonedaMaqBase	Int	Tipo de moneda de máquina base.
	iTipoMonedaEqAdicional	Int	Tipo de moneda del equipo adicional.
	DB_ADDSClaveLlanEqAdi	Key DB_ADDR	DB_ADDS de las llantas de equipo adicional.
	DB_ADDSClaveLlanMaqBase	Key DB_ADDR	DB_ADDS de las llantas de máquina base.
	DB_ADDSCveComb	Key DB_ADDR	DB_ADDR del combustible
	DB_ADDSCveLub	Key DB_ADDR	DB_ADDR del lubricante
	DB_ADDSCveLubHidra	Key DB_ADDR	DB_ADDR del lubricante hidráulico
	DB_ADDSCveGasArr	Key DB_ADDR	DB_ADDR de gasolina para arranque.
	DB_ADDSCveOp1	Key DB_ADDR	DB_ADDR del operador 1
	DB_ADDSCveOp2	Key DB_ADDR	DB_ADDR del operador 2
	DB_ADDSCveOp3	Key DB_ADDR	DB_ADDR del operador 3
	DB_ADDRSZona	Key DB_ADDR	Dirección del registro de la zona a la que pertenece el insumo.
	dCostoAnalizado	Double	Define el costo analizado de este insumo.
	iFechaCotizacion	Long	Fecha de cotización de insumo.

AnalísGen	dCostoUnitario	Double	Costo unitario
	IFechCosto	Long	Fecha de costo, está como Juliano.
	IFechCalculo	Long	Fecha de cálculo, está como Juliano.
	IMiembroSelec	DB_ADDR	Número secuencial que le corresponde en el conjunto al insumo donde se tomará el precio para los cálculos.
	IAliasActivo	DB_ADDR	Alias activo del análisis.
	iCurrency	Int	Currency que se utiliza como activo.
	ucBanderas	Unsigned Char	Campo de banderas: 0 Se utiliza para saber si un elemento de este registro está capturado. Si está en uno el registro contiene toda su información, si está a cero el registro no está completo en toda su información.
	szunidad	Char[11]	0 - 9: Datos de la Unidad 10 : Nulo
	szClave	Unique Key Char[28]	0 : Tipo de insumo 1 : Indica si está formado únicamente por mano de obra. Si el byte 0 es igual a 'B' (básico) y contiene únicamente mano de obra se coloca una 'O' (esto indica una cuadrilla). De lo contrario contendrá un blanco y no un nulo. 2 - 26 : Clave alfanumérica 27 : Caracter nulo.
AnalísDim	dPreliminares	Double	Total de preliminares.
	dMateriales	Double	Total de materiales.
	dObraMano	Double	Total de Obra de Mano.
	dSubcontratos	Double	Total de subcontratos.
	dMaquinaria	Double	Total de maquinaria
	dRendimiento	Double	Total de rendimiento de análisis.
	iFormatoSobrecostoAsig	Int	Formato de sobrecosto asignado.
	iDimension	Unsigned Int	Dimensión a la que pertenece el análisis.

Capítulo IV Desarrollo del Sistema

Componentes	dCantidad	Double	Cantidad del elemento.
	dRendimiento	Double	Rendimiento del elemento.
	DB_ADDRAnálisis	DB_ADDR	Dirección del registro de datos generales del análisis propietario de éste componente.
	szElemento	Char[28]	Elemento.
	DB_ADDRSElemento	DB_ADDR	Dirección del registro de datos generales del elemento.
	cBanderas	Unsigned Char	
	Compound Key EleAnali { szElemento DB_ADDRAnálisis}		
Familia	szSeudonimo	Unique Key Char[28]	Seudónimo de la familia
DescFam	iTipo	int	
	szDesc	Char[51]	Descripción de familias.
	Compound Optional key TipDescFam { iTipo szDesc}		
Leyinsu	iTipo	int	
	szDesc	Char[51]	Leyendas.
AliasInsumos	szAlias	Char[28]	Alias de insumos.
AliasAnálisis	szAlias	Char[28]	Alias de análisis.
Usuarios	HoraAcceso	Long	Hora en la cual se acceso el sistema.
	szNombreUsuario	Unique key Char[33]	Nombre del usuario que accesa la base de datos.
ZonaEco	szZona	Unique key Char[28]	0 : Tipo de insumo 1 : Indica si está formado únicamente por mano de obra. Si el byte 0 es igual a 'B' (básico) y contiene únicamente mano de obra se coloca una 'O' (esto indica una cuadrilla). De lo contrario contendrá un blanco y no un nulo. 2 - 26 : Clave alfanumérica 27 : Caracter nulo

DescZona	iTipo	int	
	szDesc	Char[51]	Descripción de zonas.
	Compound Optional key TipDescZona { iTipo szDesc}		
Region	szRegion	Char[26]	Región inmersa dentro de la zona.
NotasAlis	iTipo	Int	Nota : El primer renglón de la descripción contendrá en su primer byte el tipo de insumo al que pertenece. De esta forma se puede diferenciar el tipo de elemento al cual se hace referencia
	szDesc	Char[51]	Descripción de los Análisis.
Estimaciones	dCostoUnitario	Double	Costo unitario.
	IFechCosto	Long	Fecha de inicio de la estimación, está en juliano.
	IFechCalculo	Long	Fecha de termino de la estimación, está en juliano.
	IAliasActivo	DB_ADDR	Alias activo de este insumo
	iCurrency	int	Indica el tipo de moneda que se utiliza como activo de la estimación. Si el tipo de moneda no existe se toma la moneda base en ese momento.
	ucBanderas	Unsigned char	Campo de banderas: 0 Se utiliza para saber si un elemento de este registro está capturado. Si está en uno el registro contiene toda su información, si está a cero el registro no está completo en toda su información.
	szUnidad	Char[11]	
	szClave	Unique key char[28]	
ElementoEstimacion	dbAddrElemento	DB_ADDR	
	dbAddrAnalisis	DB_ADDR	
	dbAddrEstimacion	DB_ADDR	
	cCantoPorc	Char	

Capítulo IV Desarrollo del Sistema

	dCantidad	Double	
	Compound Optional key ElementoEst { dbAddrElemento dbAddrEstimacion }		
	Compound Optional key EstClaveElemento { dbAddrEstimacion dbAddrAnálisis dbAddrElemento }		
DescEstima	iTipo	int	
	szDesc	Char(51)	Descripción de familias.
	Compound Optional key TipDescEst { iTipo szDesc }		

DEFINICIÓN DE CONJUNTOS PARA INSUMOS

Conjunto	Order	Owner	Member	Función
InsuMaterialDat	Last	InsuGen	Materiales	Conjunto pa Materiales
InsuObraMano Dat	Last	InsuGen	Obrademano	Conjunto pa Obra de mano
InsumoEcMaDa t	Last	InsuGen	MaquinaEquipo	Conjunto pa Maquinaria equipo
InsumoSubC Dat	Last	InsuGen	Subcontrato	Conjunto pa Subcontrato

InsuDesc	Last	InsuGen	DescInsumos	Conjunto para descripción de insumos
FamilInsu	Last	Familia	InsuGen	Conjunto para Familia de insumos
LeyendInsu	Last	InsuGen	LeyInsu	Conjunto para leyendas de insumos
AliasInsu	Last	InsuGen	AliasInsumos	Conjunto para alias de Insumos

DEFINICIÓN DE CONJUNTOS PARA ANÁLISIS

Conjunto	Order	Owner	Member	Función
BasicoDat	Last	AnalisisGen	Componentes	Conjunto para formar básicos
SetAnalisisDim	Last	AnalisisGen	AnalisisDim	
AnalisisDesc	Last	AnalisisGen	DescInsumos	Conjunto para descripción de básicos
LeyenAnalisis	Last	AnalisisGen	LeyAnali	Conjunto para leyendas del análisis
FamiAnalisis	Last	Familia	AnalisisGen	Conjunto para familias de análisis
AliasAnalisis	Last	AnalisisGen	AliasAnalisis	Conjunto para alias de análisis
AnalisisNotas	Last	AnalisisGen	NotasAnalisis	Conjunto para notas de análisis

DEFINICIÓN DE CONJUNTOS PARA FAMILIAS

Conjunto	Order	Owner	Member	Función
FamDesc	Last	Falimia	DescFam	Conjunto para descripción de familias
FamFam	Last	Familia	Familia	Conjunto de familias

DEFINICIÓN DE CONJUNTOS PARA ZONA ECONÓMICA

Conjunto	Order	Owner	Member	Función
DescripZona	Last	ZonaEco	DescZona	Conjunto para descripción de zona económica
ZonaRegion	Last	ZonaEco	Region	Conjunto de regiones

DEFINICIÓN DE CONJUNTOS PARA ESTIMACIONES

Conjunto	Order	Owner	Member	Función
ConjuntoEstimaciones	Last	Estimaciones	ElementoEstimacion	Conjunto para estimaciones
DescEstimaciones	Last	Estimaciones	DescEstima	Conjunto para descripción de estimaciones

1.3 DIAGRAMA DE ENTIDAD-RELACIÓN

MODELO ENTIDAD-RELACIÓN

Modelo Entidad-Relación es una técnica para definir las necesidades de información en cualquier empresa. Esta técnica involucra conceptos que se identifican con varios objetos de importancia para la empresa, a los cuales se les denomina Entidades, a las características de dichos objetos se les denomina Atributos y a cómo se relacionan estos objetos entre sí se le denomina Relaciones.

Todos estos conceptos se modelan a través de cierto tipo de esquemas gráficos, los cuales muestran a los usuarios una manera más sencilla y práctica de visualizar sus necesidades de información.



Fig. 4.1.3.1 Modelo de Entidad-Relación que representa un problema real.

ENTIDAD

Una entidad es una persona, cosa o lugar que cae dentro del alcance del sistema acerca de la cual el sistema debe mantener, correlacionar y desplegar información.

La entidad se representa por medio de una caja con las esquinas redondeadas dentro de ésta se escribe el nombre de la entidad el cual debe estar en singular.

Cada entidad debe tener un nombre único dentro del sistema, lo que indica que no pueden existir dos entidades del mismo sistema con el mismo nombre.

Ejemplo:

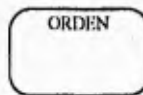


Fig. 4.1.3.2 Representación de una entidad.

RELACIÓN

Una relación requiere de una a más entidades, la cual debe caer dentro del alcance del sistema, acerca de la cual el sistema debe mantener, correlacionar y desplegar información.

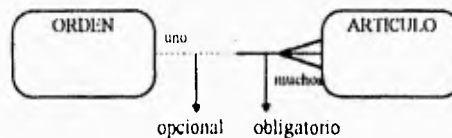


Fig. 4.1.3.3 Representación de una entidad.

Las relaciones se presentan en varias modalidades:

- Relaciones uno a uno con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.



Fig. 4.1.3.4 Relación uno a uno.

- Relaciones uno a muchos con las combinaciones de obligatorio a opcional, opcional a opcional, obligatorio a obligatorio y opcional a obligatorio.



Fig. 4.1.3.5 Relación uno a muchos.

- Relaciones muchos a muchos con las combinaciones de opcional a opcional, obligatorio a opcional y obligatorio a obligatorio.

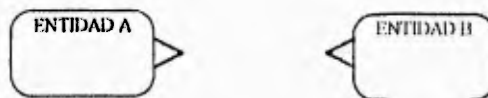


Fig. 4.1.3.6 Relación muchos a muchos.

La manera de leer las relaciones es:

Cada elemento de la Entidad A debe o puede relacionarse con uno y sólo uno (muchos) elemento (s) de la Entidad B.

Ejemplo:

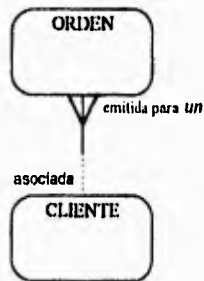


Fig. 4.1.3.7 Relación uno a muchos.

ATRIBUTO

Un atributo es una característica o cualidad de una Entidad o Relación, que cae dentro del alcance del sistema, acerca del cual el sistema debe mantener, correlacionar o desplegar información.

Para representar uno o varios atributos, se escribe el nombre del atributo dentro de la entidad.

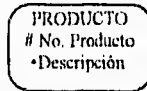
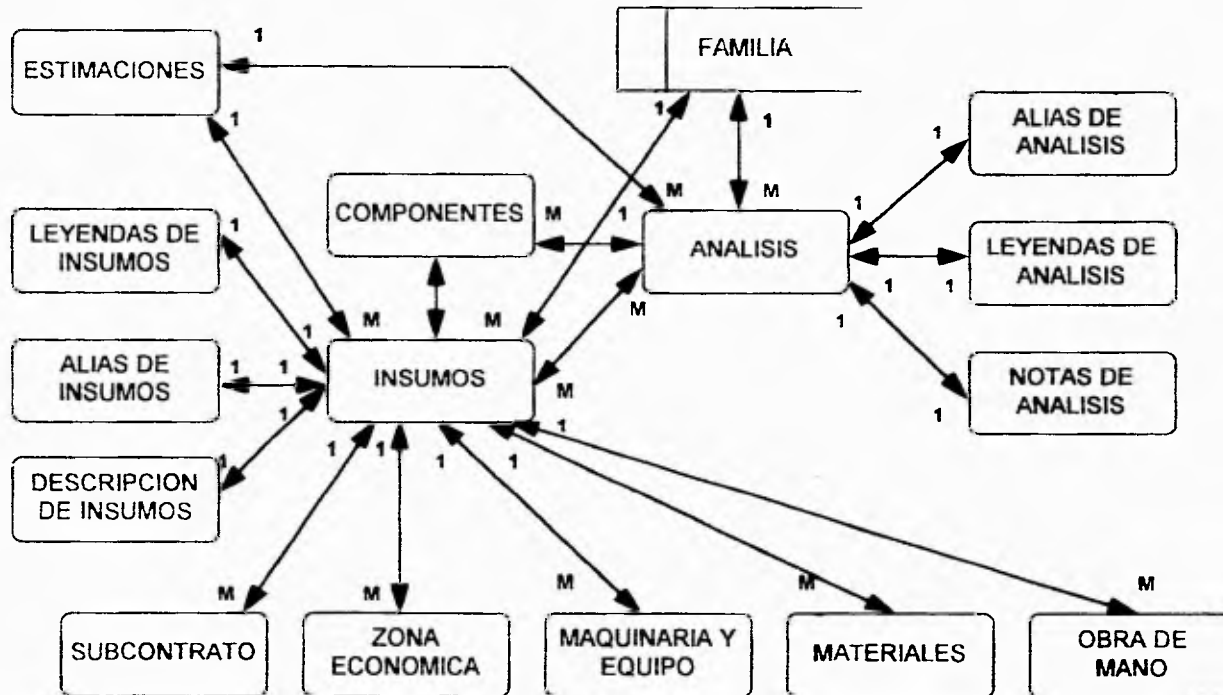


Fig. 4.1.3.8 Relación uno a muchos

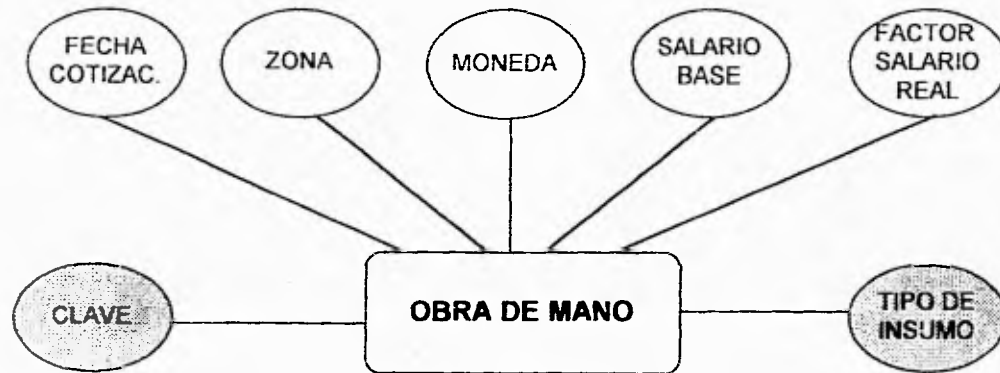
Indica que el atributo es llave primaria (PK).

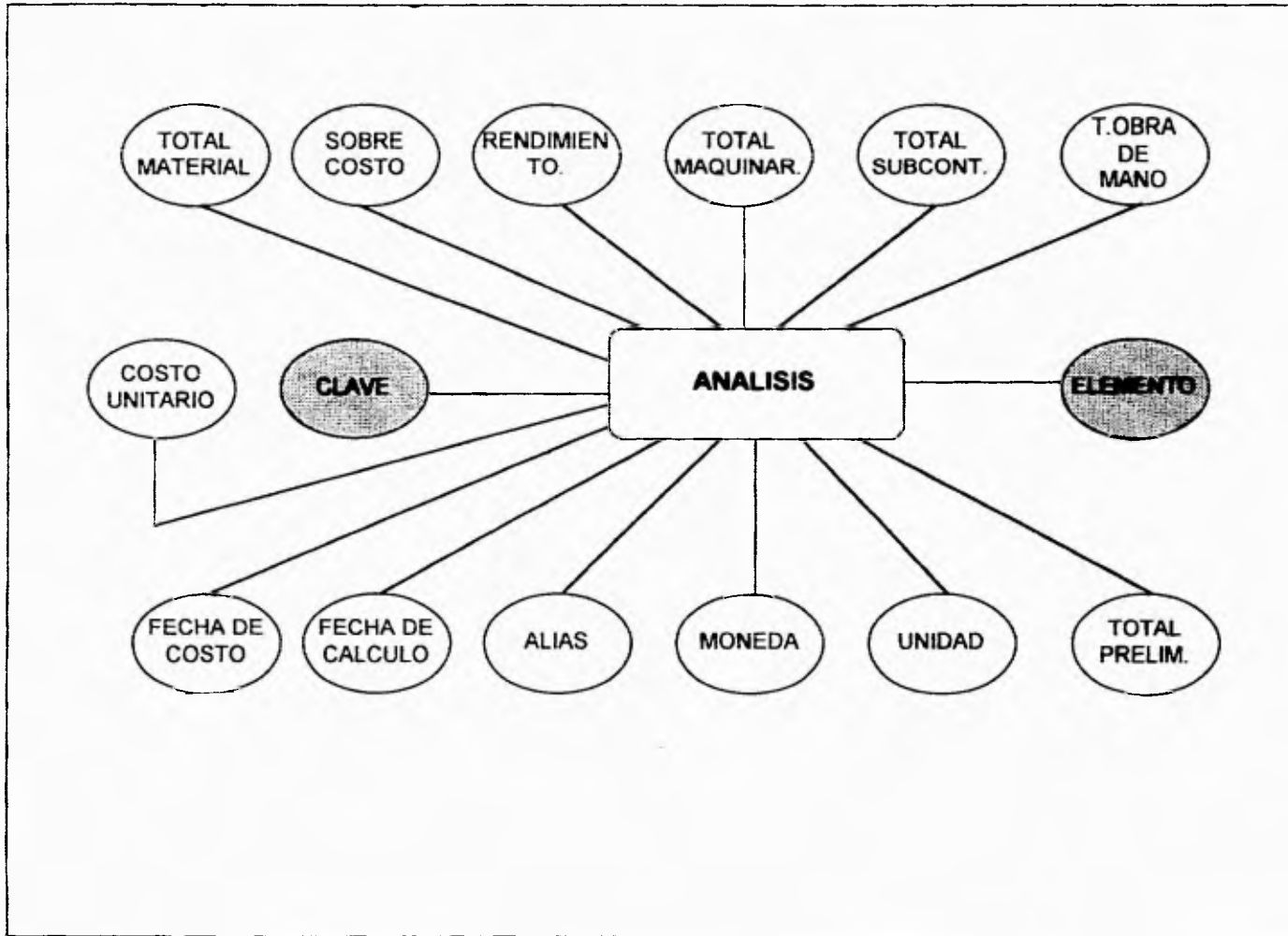
Indica que el atributo es obligatorio.

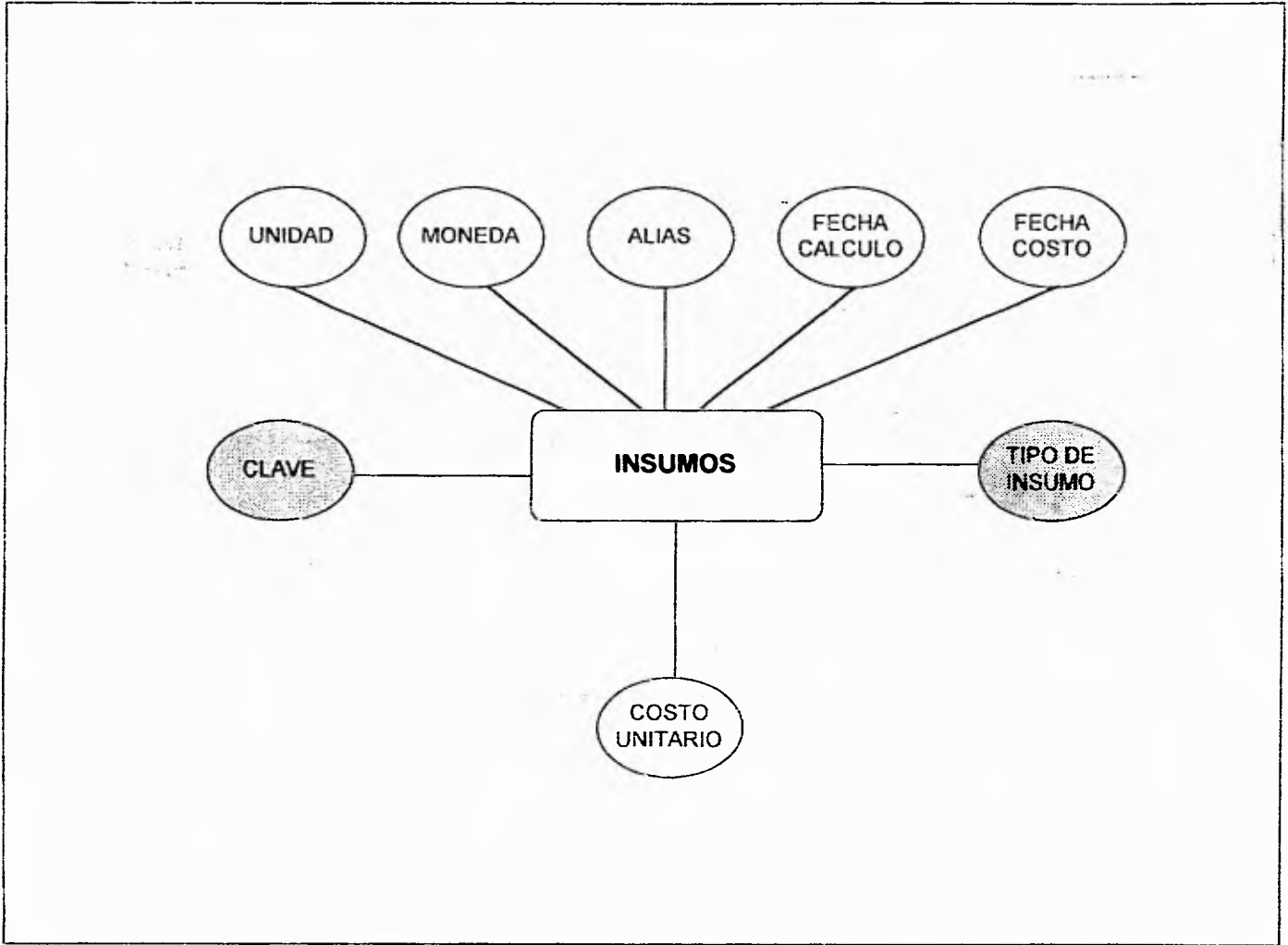
DIAGRAMA ENTIDAD - RELACION

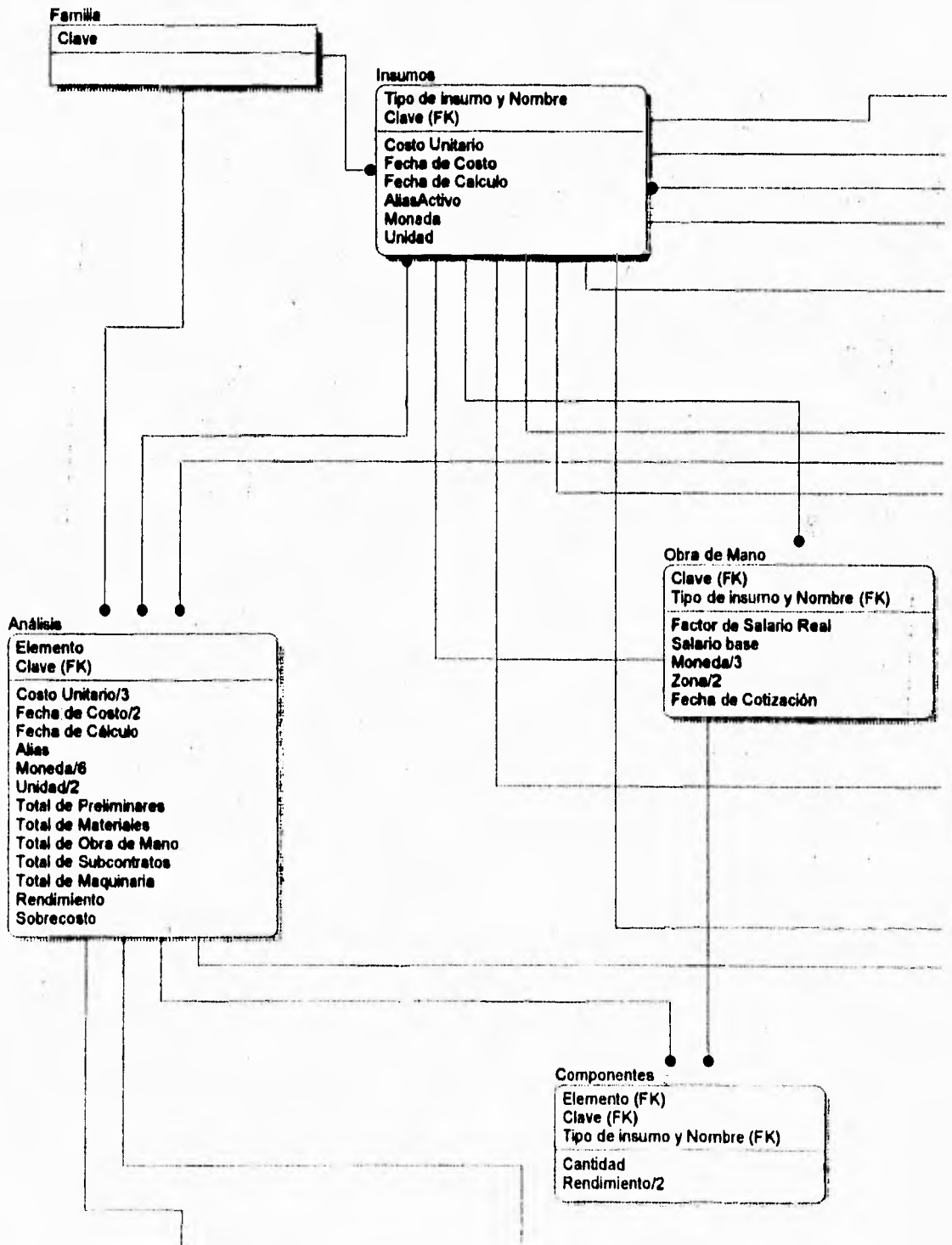


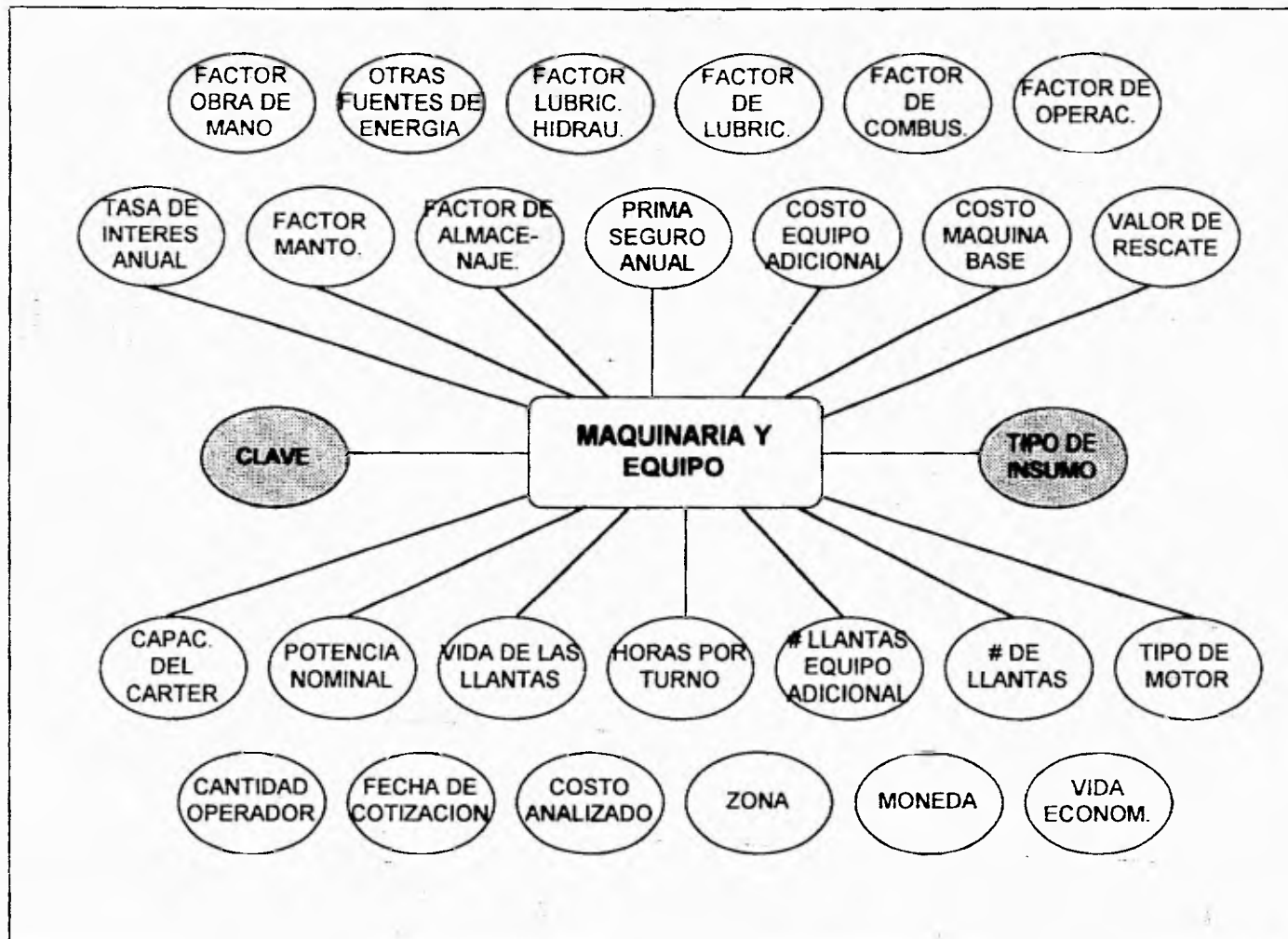


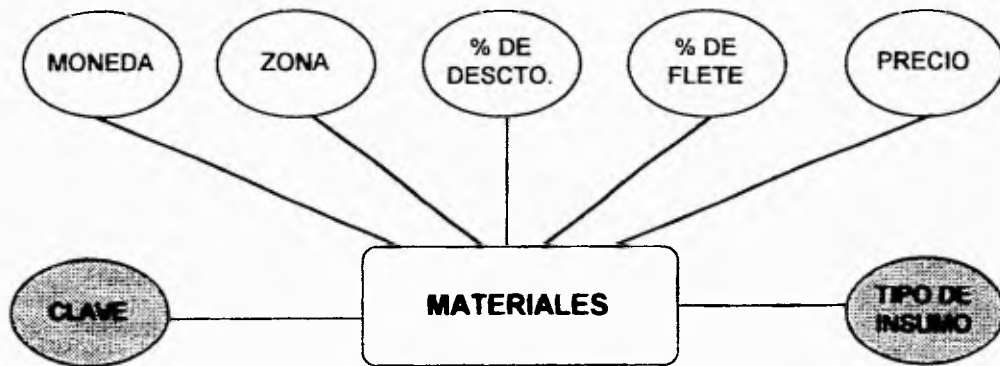


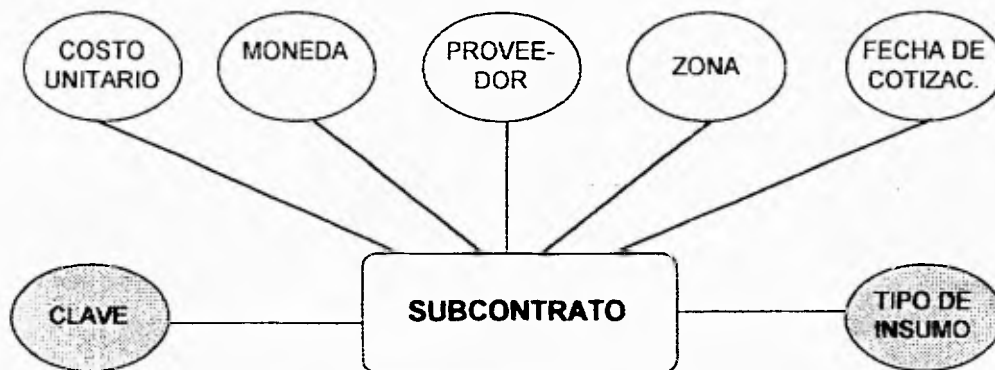


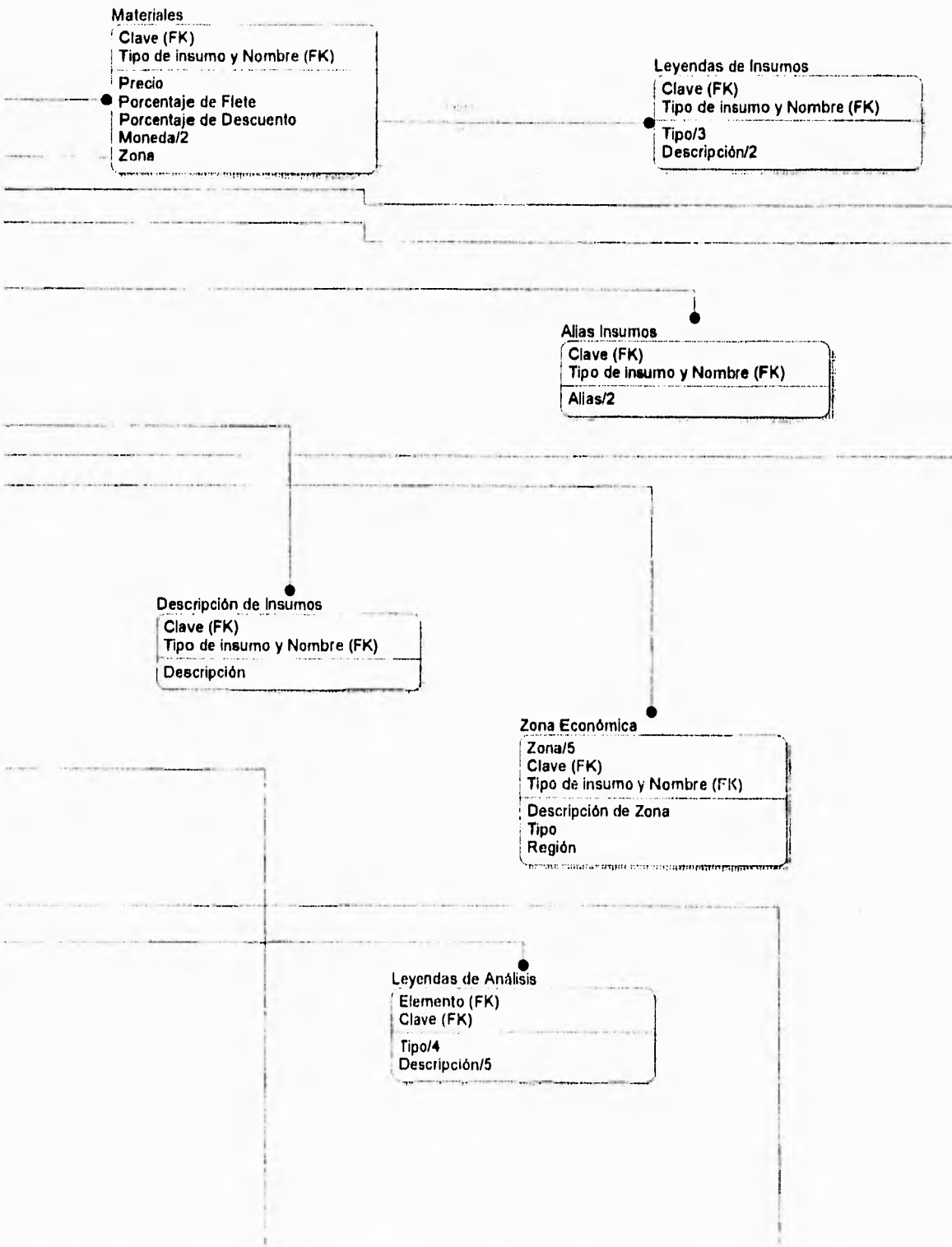












Alias de Análisis

Elemento (FK)
Clave (FK)

Alias/3

Notas de Análisis

Elemento (FK)
Clave (FK)

Tipo/5
Descripción/3

Maquinaria y Equipo

Clave (FK)
Tipo de insumo y Nombre (FK)
Valor de Rescate
Costo de Máquina Base
Costo Equipo Adicional
Prima Seguro Anual
Factor de Almacenaje
Factor de Mantenimiento
Tasa de Interés Anual
Factor de Operación
Factor de Combustible
Factor de Lubricante
Factor de Lubricante Hidráulico
Otras Fuentes de Energía
Factor de Operación de Obra de Ma
Tipo de Motor
Cantidad de Llantas para Maquinaria
Cantidad de Llantas Equipo Adicional
Horas por Turno
Vida de las Llantas
Potencia Nominal
Capacidad del Carter
Vida Económica
Moneda/5
Zona/4
Costo Analizado
Fecha de Cotización/3
Cantidad Operador 1
Cantidad Operador 2
Cantidad Operador 3

Subcontrato

Clave (FK)
Tipo de insumo y Nombre (FK)
Costo Unitario/2
Moneda/4
Proveedor
Zona/3
Fecha de Cotización/2

4.1.4 DISEÑO E IMPLEMENTACIÓN DE LOS MÓDULOS

El sistema estará compuesto por diferentes módulos que realizarán diversas funciones de apoyo a la construcción. Estos serán:

- Precios Unitarios
- Control de Obra
- Manejo de Obras
- Opciones

I.- PRECIOS UNITARIOS

El módulo de precios unitarios realizará el manejo de presupuestos de una obra, permitiendo los siguientes análisis:

- Insumos
- Análisis de Concepto de Obras
- Presupuestos de Obra

Los submódulos que compondrán a precios unitarios serán los siguientes :

Insumos

- Materiales
- Mano de Obra
- Maquinaria y Equipo

Subcontratos

Reportes

 Materiales

 Mano de Obra

 Maquinaria y Equipo

 Subcontratos

Factor de Salario Real

 Cálculo del Factor de salario Real

 Reporte del factor de salario Real

Análisis

 Básicos

 Integrados

 Partidas

 Capítulos

 Reporte de Conceptos

 Básicos

 Integrados

 Partidas

 Capítulos

Presupuestos

Las actividades que se llevarán acabo en cada uno de estos submódulos serán:

INSUMOS

Dentro de este módulo se permitirá crear el catálogo de Insumos para el presupuesto de obra; el sistema manejará cuatro tipos diferentes de Insumos: Materiales, Mano de Obra, Maquinaria y Equipo y Subcontratos.

Para la edición de Insumos y Análisis dentro del módulo se utilizará la Edición en Modo Registro, las principales características de este modo son:

CARACTERÍSTICAS

- Permite visualizar un registro a la vez y la información asociada a éste.
- La captura de los insumos es más ágil y más sencilla.
- Ofrece una funcionalidad semejante a la captura de un registro dentro de una base de datos.
- Los datos comunes en todos los insumos podrán capturarse mediante campos, mientras que los datos particulares de éstos se capturarán mediante un conjunto de celdas.

El sistema contará con una parte de información general común a todos los insumos como son su Clave, Descripción, Unidad, Fechas de Costo y Cálculo, Familia (Mano de agrupaciones) y Costo Unitario.

- 1.- En el campo de edición de Clave, se registrará la clave del insumo, ésta clave podrá ser de hasta 25 caracteres y podrá contener caracteres

alfanuméricos. Al proporcionar la clave del insumo el sistema adicionará automáticamente la letra correspondiente al tipo de insumo:

Letra	Tipo Análisis
M	Materiales
O	Mano de Obra
Q	Maquinaria y Equipo
T	Subcontratos

El tipo de insumo servirá para identificar si el insumo es un Material, Mano de Obra, Maquinaria y Equipo o Subcontrato.

- 2.- Dentro del campo de edición de Descripción, se registrará la descripción del insumo, podrá ser tan grande como el usuario lo desee y podrá contener dígitos, letras, espacios y cualquier signo del teclado estándar.
- 3.- En el campo de Familia, se proporcionará la Familia a la que pertenece el insumo, la captura de éste dato podrá ser opcional y podrá ser de hasta 11 caracteres alfanuméricos. Si la Familia que se capture no ha sido dada de alta, con sólo teclearla dentro de éste campo se agregará al Catálogo de Familias.
- 4.- Dentro del campo de Unidad, se proporcionará la unidad de medida del insumo, el nombre de la unidad de medida debe ser de hasta 11 caracteres alfanuméricos.
- 5.- En el campo de Fecha de Costo, se capturará la fecha en la cual se le asigna el costo al insumo.

- 6.- Dentro del campo de Fecha de Cálculo, se capturará la fecha en la cual se actualiza el costo del insumo.

La segunda parte del módulo consistirá en el registro de los datos particulares de los insumos.

EDICIÓN DE LOS DATOS PARTICULARES DE MATERIALES

Para editar los datos particulares de materiales, se utilizará el módulo de Materiales para generar y mantener actualizados los datos de materiales para la obra.

Dentro de los datos particulares de Materiales, se podrá asignar diferentes precios a un material, estos precios podrán ser asociados o no a una Zona Económica, también contará con campos para la captura de información del Precio de Lista el cuál puede ser capturado con diferentes tipos de monedas, Porcentaje de Flete aplicado y Porcentaje de Descuento aplicado.

- 1.- Dentro del campo correspondiente a la Zona Económica, se proporcionará la zona a la cuál pertenece el Precio de Lista capturado, podrá ser opcional la captura de este dato.
- 2.- Dentro del campo de Moneda se proporcionará la Moneda (el símbolo) con la cuál se calculará el precio del material.

Para registrar el símbolo de la Moneda en los datos particulares del insumo, éste deberá estar registrado dentro del catálogo de monedas.

- 6.- Dentro del campo de Fecha de Cálculo, se capturará la fecha en la cual se actualiza el costo del insumo.

La segunda parte del módulo consistirá en el registro de los datos particulares de los insumos.

EDICIÓN DE LOS DATOS PARTICULARES DE MATERIALES

Para editar los datos particulares de materiales, se utilizará el módulo de Materiales para generar y mantener actualizados los datos de materiales para la obra.

Dentro de los datos particulares de Materiales, se podrá asignar diferentes precios a un material, estos precios podrán ser asociados o no a una Zona Económica, también contará con campos para la captura de información del Precio de Lista el cuál puede ser capturado con diferentes tipos de monedas, Porcentaje de Flete aplicado y Porcentaje de Descuento aplicado.

- 1.- Dentro del campo correspondiente a la Zona Económica, se proporcionará la zona a la cuál pertenece el Precio de Lista capturado, podrá ser opcional la captura de este dato.
- 2.- Dentro del campo de Moneda se proporcionará la Moneda (el símbolo) con la cuál se calculará el precio del material.

Para registrar el símbolo de la Moneda en los datos particulares del insumo, éste deberá estar registrado dentro del catálogo de monedas.

- 3.- Se proporcionará el precio de lista de material dentro del campo de Precio de Lista.
- 4.- Dentro del campo de %Flete se capturará el Porcentaje de Flete del material, será opcional la captura de este campo.
- 5.- Dentro del campo de %Desct se proporcionará el Porcentaje de Descuento del material, será opcional la captura de éste campo.
- 6.- Una vez proporcionados los precios, se deberá elegir el precio con el que se realizará el cálculo del costo unitario, es decir, el precio de lista activo.

El Costo Unitario del Material será calculado de la siguiente manera:

$$\text{Costo Unitario} = \text{Precio de Lista} + \% \text{ de Flete} - \% \text{ de Descuento}$$

EDICIÓN DE LOS DATOS PARTICULARES DE UNA MANO DE OBRA

Dentro de los datos particulares de la Mano de Obra, se podrán asignar diferentes salarios a una Mano de Obra, estos salarios podrán ser asociados o no a una zona económica, también se contará con campos para la captura del Salario Base el cual podrá ser capturado con diferentes tipos de moneda y el resultado del Cálculo del Factor de Salario Real (Mínimo, Mayor al mínimo y definición propia).

- 1.- Contará con un campo correspondiente a la Zona Económica, que indicará la zona a la cuál pertenecerá el salario base, la captura de este campo podrá ser opcional.

- 2.- Dentro del campo de Moneda, se capturará el símbolo de la moneda bajo la cuál estará el salario base de la mano de obra.
- 3.- Se proporcionará el Salario Base de la mano de obra dentro de la celda de Salario Base.
- 4.- Dentro del campo Tipo de Factor, se seleccionará un tipo de factor a aplicar.

Letra	Tipo de Factor
M	Factor Mínimo
A	Factor Mayor al Mínimo
P	Factor Propio

- 5.- Dentro del campo Factor a Aplicar, se proporcionará el valor de dicho factor en caso de haber seleccionado como tipo Factor Propio, en caso de haber elegido Factor Mínimo o Mayor al mínimo, éste será proporcionado de acuerdo a lo calculado en el módulo del Factor de Salario Real.
- 6.- Una vez proporcionados los Salarios Base relacionados con la Mano de Obra, se deberá elegir el salario base de la mano de obra con el que se realizará el cálculo del Costo Unitario, es decir, el salario base activo.

EDICIÓN DE LOS DATOS PARTICULARES DE LA MAQUINARIA Y EQUIPO Y EL COSTO HORARIO

- 1.- Dentro del campo correspondiente a la Zona Económica, se proporcionará la zona a la cuál pertenecerá el costo de la maquinaria, la captura de éste campo será opcional.
- 2.- Dentro del campo de Moneda, se capturará el símbolo de la moneda bajo la cuál estará el Costo Unitario de la Maquinaria.
- 3.- En caso de no desearse el análisis del costo horario de la maquinaria, dentro del campo de Costo Rápido, se proporcionará el costo unitario de la maquinaria, esto podrá ser de utilidad si se renta el equipo.

CÁLCULO DEL COSTO HORARIO DE LA MAQUINARIA Y EQUIPO

El Cálculo Costo Horario estará formado por las siguientes opciones: Gastos Equipo, Factores, Motor y Gastos Operación, los cuales se deberán permitir editar para realizar el cálculo.

GASTOS EQUIPO

Dentro de esta opción se podrán capturar los costos de la maquinaria base y del equipo adicional, el precio de las llantas y el valor de rescate de la maquinaria y equipo, para ello se requerirá lo siguiente:

Dentro de la tabla Costo de la Maquinaria y Equipo Adicional:

- 1.-Dentro del campo Moneda se registrará el símbolo de la moneda bajo la cual estará el costo de la maquinaria base.**
- 2.-Dentro del campo de Costo, se registrará el costo de adquisición de máquina base.**

Dentro de la Tabla Precio de las Llantas Máquina Base y Equipo Adicional:

- 1.-En el campo de Clave, se registrará la clave de las llantas de la máquina base.**
- 2.- Dentro del campo de No., se registrará el número de llantas.**

El Valor Total de las Llantas, se calculará como:

$$\begin{array}{l} \text{Importe de las llantas} \\ \text{de máquina base} \end{array} + \begin{array}{l} \text{Importe de las llantas de equipo} \\ \text{adicional} \end{array}$$

El Valor de Adquisición (Va) se calculará de la siguiente manera:

$$\text{Costo Máquina Base} + \text{Costo de Equipo Adicional} - \text{Valor Total de las llantas}$$

- 3.- Dentro del campo de Valor de Rescate% (Vr) se registrará el porcentaje a aplicar al valor de adquisición de la Maquinaria y Equipo, éste se calculará como:**

$$\text{Valor de Adquisición} * \text{Porcentaje}/100$$

FACTORES

En esta parte del módulo se permitirá registrar la vida económica en horas, la vida de las llantas, las horas anuales, la prima de seguro anual, los factores de almacenaje, mantenimiento y operación, la potencia nominal y la tasa de interés anual, para esto será necesario:

- 1.- Dentro del campo de Vida Económica en Horas (Ve) se registrará la vida económica en horas de trabajo de la Maquinaria y Equipo.
- 2.- Dentro del campo de Vida de las Llantas en Horas (HVLL), se registrará la vida económica de las llantas de la maquinaria y Equipo, expresada en horas de trabajo.
- 3.- Dentro del campo de Horas Anuales (Ha), se registrará el número de horas efectivas que el equipo trabaja durante el año.
- 4.- Dentro del campo de Prima de Seguro Anual (s), se registrará la prima anual de la Maquinaria y Equipo, expresada en porcentaje.
- 5.- Dentro del campo de Factor de Almacenaje (Fa), se registrará el factor de almacenaje para la Maquinaria y Equipo.
- 6.- Dentro del campo de Factor de Mantenimiento (Q), se deberá registrar el coeficiente que incluya tanto el mantenimiento mayor como el menor de la maquinaria.
- 7.- En el campo de Factor de Operación, se registrará el factor de operación de la maquinaria y equipo.

8.- Dentro del campo de Potencia Nominal, se registrará la potencia nominal del motor de la maquinaria

9.-En el campo de Tasa de Interés Anual (i), se registrará la tasa de interés anual, la cuál estará expresada como un porcentaje.

MOTOR

Dentro de esta opción se podrá editar la información relacionada con el tipo de motor de la maquinaria, para esto será necesario lo siguiente:

- a) El tipo de motor podrá ser Diesel, Gasolina, Eléctrico.**
- b) Una vez proporcionada la descripción del combustible utilizado se deberá proporcionar la moneda bajo la cual se encontrará el costo unitario.**
- c) En seguida se deberá registrar el costo del combustible utilizado.**
- d) Se deberá registrar como Factor, es decir, la cantidad de combustible por hora efectiva de trabajo.**

GASTOS OPERACIÓN

Dentro de esta sección se podrá editar la información necesaria para calcular el importe de los cargos por operación de la maquinaria, para esto será necesario:

- 1) Se deberá proporcionar el Factor de Operación de la Mano de Obra por hora efectiva de trabajo.**

2) Se deberán registrar las Horas Diarias de trabajo de la Mano de Obra.

3) Mano de Obra utilizada para esto se requerirá :

- a) Clave.
- b) Categoría, descripción o categoría de la Mano de Obra.
- c) Moneda, la moneda bajo la cual se encuentra el Salario de la Mano de Obra.
- d) Salario, el salario de la Mano de Obra.

4) Cantidad de operadores para el tipo de categoría seleccionada. El importe de los salarios de la Mano de Obra se calculará como:

$$\text{salario} * \text{cantidad}$$

El cargo por operación de la maquinaria y equipo por hora efectiva de trabajo, se calculará como:

$$\text{suma de salarios} / (\text{factor de operación} * \text{horas diarias})$$

El cálculo final del Costo Unitario de la Maquinaria y equipo se realizará de la siguiente manera:

$$\text{VALOR DE RESCATE} = \text{VALOR DE ADQUISICIÓN} * \frac{(\% \text{ VALOR DE RESCATE})}{100}$$

$$\text{POTENCIA DE OPERACIÓN} = \text{POTENCIA} * \text{FACTOR DE OPERACIÓN}$$

$$1) = \frac{(\text{VALOR DE ADQUISICIÓN} - \text{VALOR DE RESCATE})}{\text{VIDA ECONÓMICA EN HORAS}}$$

$$2) = \frac{(\text{VALOR DE ADQUISICIÓN} + \text{VALOR DE RESCATE}) * \text{TASA DE INTERÉS ANUAL}}{(2 * \text{HORAS ANUALES}) * 100}$$

$$3) = \frac{(\text{VALOR DE ADQUISICIÓN} + \text{VALOR DE RESCATE}) * \text{PRIMA DE SEGURO ANUAL}}{(2 * \text{HORAS ANUALES}) * 100}$$

$$\text{ALMACENAJE} = D \cdot \text{FACTOR DE ALMACENAJE}$$

$$M = \text{FACTOR DE MANTENIMIENTO} \cdot D$$

Para Motor Diesel:

$$\text{GASOLINA DE ARRANQUE} = \left[\frac{\text{FACTOR DE GASOLINA DE ARRANQUE}}{\text{DE ARRANQUE}} \right] \cdot \left[\text{POTENCIA DE OPERACIÓN} \right]$$

$$\text{COMBUSTIBLE} = \left[\frac{\text{FACTOR DE COMBUSTIBLE}}{\text{DE COMBUSTIBLE}} \right] \cdot \left[\text{POTENCIA DE OPERACIÓN} \right] \cdot \left[\text{COSTO COMBUSTIBLE} \right]$$

$$\text{LUBRICANTE} = \left(\left(\frac{\text{CAPACIDAD CARTER}}{\text{TIEMPO CAMBIO}} \right) + \text{FACTOR LUBRICANTE} \right) \cdot \text{POTENCIA OPERACION} \cdot \text{COSTO LUBRICANTE}$$

$$\text{LUBRICANTE HIDRÁULICO} = \left[\frac{\text{FACTOR LUBRICANTE HIDRÁULICO}}{\text{HIDRÁULICO}} \right] \cdot \left[\frac{\text{COSTO LUBRICANTE HIDRÁULICO}}{\text{HIDRÁULICO}} \right]$$

$$\text{CARGOS POR CONSUMO} = \text{COMBUSTIBLE} + \text{LUBRICANTE} + \text{LUBRICANTE HIDRÁULICO} + \text{CONSUMO DE LLANTAS} + \text{GASOLINA ARRANQUE} + \text{OTRAS FUENTES ENERGÍA}$$

Para Motor Gasolina:

$$\text{COMBUSTIBLE} = \left[\frac{\text{FACTOR COMBUSTIBLE}}{\text{COMBUSTIBLE}} \right] \cdot \left[\frac{\text{POTENCIA DE OPERACIÓN}}{\text{OPERACIÓN}} \right] \cdot \left[\frac{\text{COSTO COMBUSTIBLE}}{\text{COMBUSTIBLE}} \right]$$

$$\text{LUBRICANTE} = \left(\left(\frac{\text{CAPACIDAD CARTER}}{\text{TIEMPO DE CAMBIO}} \right) + \text{FACTOR LUBRICANTE} \right) \cdot \text{POTENCIA OPERACIÓN} \cdot \text{COSTO LUBRICANTE}$$

$$\text{LUBRICANTE HIDRÁULICO} = \left[\frac{\text{FACTOR LUBRICANTE HIDRÁULICO}}{\text{HIDRÁULICO}} \right] \cdot \left[\frac{\text{COSTO LUBRICANTE HIDRÁULICO}}{\text{HIDRÁULICO}} \right]$$

$$\text{CARGOS POR CONSUMO} = \text{COMBUSTIBLE} + \text{LUBRICANTE} + \text{LUBRICANTE HIDRÁULICO} + \text{CONSUMO DE LLANTAS} + \text{OTRAS FUENTES ENERGÍA}$$

Para Motor Eléctrico

$$\text{CARGOS POR CONSUMO} = \text{FACTOR MOTOR ELÉCTRICO} \cdot \text{POTENCIA OPERACIÓN} \cdot \text{COSTO DEL KILOWATT}$$

$$\text{COSTO HORA MÁQUINA SIN OPERACIÓN} = D + I + S + M + \text{CARGOS POR CONSUMO}$$

**COSTO HORA MÁQUINA = COSTO HORA MÁQUINA SIN OPERACIÓN + CARGOS POR OPERACIÓN
CON OPERACIÓN**

FACTOR DE SALARIO REAL

Dentro de este módulo del sistema, se calculará el Factor de Salario Real, para salarios mínimos, ó para mayores al mínimo, el cálculo deberá ser para Obra Pública ó Privada.

Deberá permitir modificar los factores relacionados con la Ley de Obra Pública o con el Seguro Social y agregar nuevos conceptos para considerarlos dentro del cálculo final.

Este módulo deberá presentar las siguientes características:

- 1.-El módulo deberá presentar el resultado final del Factor de Salario Real, así como sus principales cálculos.
- 2.-Permitirá capturar los diferentes factores según la Ley Federal del Trabajo, por Seguro Social y por Otros Conceptos.

El usuario elegirá el factor de salario real que se presentará calculado con valores por default, ó bien, definir los factores que intervendrán dentro de su cálculo.

Para el cálculo del Factor de Salario Real se registrarán:

- 1) **Días Pagados** .- Dentro de esta parte se registrará el número de días pagados por año para cada concepto (Prima Vacacional, Aguinaldo, Otros, etc.). El total de días pagados se calculará como:

Días por año + Prima Vacacional + Aguinaldo + Otros1 + Otros2 + Otros3

2) Días no laborados .- Aquí se registrará el título y el dato de todos los conceptos relacionados con los días no laborados (Séptimo Día, Días Festivos, Vacaciones, Fiestas de Costumbre, Enfermedad y Mal Tiempo).

El total de días no laborados se calculará como:

Séptimo Día + Días Festivos + Vacaciones + Fiestas de Costumbre + Enfermedad + Mal Tiempo + Otros1 + Otros2 + Otros3

3) Seguro Social .- En esta parte se registrará el Factor de Salario Real por Seguro Social.

Se proporcionarán los porcentajes para Salarios Mínimos relacionados con cada uno de los conceptos por Seguro Social (Riesgos del Trabajo, Enfermedad y Maternidad e invalidez y Vejez).

Se registrará el porcentaje para Salarios Mayores al mínimo para cada uno de los conceptos por Seguro Social.

El total del Seguro Social para Salarios Mínimos y para Mayores al Mínimo se calculará como:

Suma de los conceptos mínimos = Riesgos del Trabajo + Enfermedad y Maternidad + Invalidez + Otros1 + Otros2 + Otros3

$$\text{Factor para Salarios mínimos} = \frac{(\text{Suma de los Conceptos mínimos}/100) * \text{Días Pagados}}{\text{Días Laborados.}}$$

Suma de los Conceptos may. al min. = Riesgos del Trabajo + Enfermedad + Maternidad + Invalidez + Otros1 + Otros2 + Otros3

$$\text{Factor para Salarios} = \frac{(\text{Suma de los Conceptos may. al min.}/100) * \text{Días Pagados}}{\text{Días Laborados.}}$$

4) Otros Factores .- Aquí se registrará el porcentaje relacionado con los conceptos de Guarderías, Infonavit, ISRP, y SAR. Se deberá indicar si el cálculo del factor se realizará con los Días por Año o con los Días Pagados.

Los Factores se calcularán como:

Factor de Guarderías = ((% Guarderías /100) * Días por Año) / Días Laborados

Factor de Infonavit = ((% INFONAVIT /100) * Días por Año) / Días Laborados

Factor ISRP = ((% ISRP /100) * Días Pagados) / Días Laborados

Factor del SAR = ((% SAR /100) * Días Pagados) / Días Laborados

Otros 1,2, o 3 por días por año = $\frac{(\% \text{ de Otros } 1,2 \text{ o } 3/100) * \text{Días por año}}{\text{Días Laborados}}$

Otros 1,2 o 3 por días pagados = $\frac{(\% \text{ de Otros } 1,2 \text{ o } 3/100) * \text{Días pagados}}{\text{Días Laborados}}$

Otros = Otros1 + Otros2 + Otros3 por días por año + Otros 1 + Otros 2 + Otros 3 para días pagados

ANÁLISIS

NIVELES DE ANÁLISIS

Este módulo permitirá manejar cinco tipos de Análisis o Matrices: Básicos, Integrados, Partidas, Capítulos, y Extraordinarios, deberá presentar el importe de cada uno de los elementos que componen el análisis, así como el costo total del mismo. Por medio de estos niveles, se podrán elaborar los Precios Unitarios y el Presupuesto.

Capítulo IV Desarrollo del Sistema

El sistema permitirá hacer las agrupaciones que se deseen y formar cada uno de niveles de Análisis que se seleccionen, no deberá existir restricción en cuanto al número de elementos o componentes que puedan estar dentro de un análisis, así como número de éstos. Un análisis podrá contener otro de su mismo nivel, por ejemplo, básico podrá contener un insumo u otro básico.

Para realizar el agrupamiento de los Análisis se tomará en cuenta lo siguiente:

- a) **Básicos:** Formados de Básicos e Insumos
- Integrados o Precios Unitarios:** Formados de Integrados, Básicos e Insumos
- b) **Partidas:** Formadas por Integrados y Básicos
- Capítulos:** Formados de Partidas e Integrados

En estos dos niveles se encontrará el presupuesto de la Obra o catálogo base. Aquí manejarán los volúmenes de obra.

- c) **Extraordinarios:** Formados de Capítulos y Partidas

En este nivel se pueden analizar todos los Básicos, Integrados, Partidas y/o Capítulos que están fuera del Presupuesto.

DATOS GENERALES DEL ANÁLISIS

El módulo de Análisis contendrá una sección de datos generales para todos los niveles existentes, ejemplo:

Para el nivel de:	Ejemplo:
Básicos	Morteros, concretos, cuadrillas, andamios, etc.
Integrados	Muros, zapatas, castillos, instalaciones, etc.
Partidas	Albañilería, cimentación, instalación, hidráulica, instalación sanitaria, etc.
Capítulos	Frentes de Obra, edificio uno, edificio dos, conjunto habitacional A, etc.
Extraordinarios	Son todos aquellos conceptos fuera de obra

- 1.-Deberá registrarse la clave del análisis. Al proporcionar la clave del Análisis, se adicionará a ésta el Tipo de Análisis.

Letra	Tipo Análisis
B	Básicos
I	Integrados
P	Partidas
C	Capítulos
X	Extraordinarios

El tipo de análisis servirá para identificar a que nivel pertenece el análisis.

- 2.-Se proporcionará la descripción del Análisis los caracteres podrán ser dígitos, letras, espacios y cualquier signo del teclado estándar.
- 3.-Se registrará la Familia a la que pertenece el Análisis.
- 4.-La Unidad de medida del Análisis.

- 5.-La Fecha de Costo; fecha en la cual se recalcula el Costo del Análisis.
- 6.-La Fecha de Cálculo; fecha en la cual se recalcula el Costo del Análisis.
- 7.-Moneda; moneda con la cual será calculado el Costo Total del Análisis.

COMPONENTE DE UN ANÁLISIS

La información que deberá proporcionarse para un componente de un análisis será:

- 1.-Tipo; tipo de Insumo o Análisis que se desea, esto tipos pueden ser:

Tipo	Descripción
M	Material
O	Mano de Obra
Q	Maquinaria y Equipo
T	Subcontrato
B	Básico
P	Partida
C	Capítulo
X	Extraordinario
H	Herramienta
HI	Mandos Intermedios
HM	Porcentaje sobre Materiales
HQ	Porcentaje sobre Maquinaria y Equipo
HT	Porcentaje sobre Subcontratos
HP	Porcentaje Preliminares

- 2.-Clave; clave del elemento.
- 3.-Cantidad; cantidad del componente a utilizar dentro del Análisis.
- 4.-Rendimiento, rendimiento del elemento; el importe para el elemento será calculado de la siguiente manera:

$$\text{Importe} = (\text{Cantidad} / \text{Rendimiento}) * \text{Costo Unitario}$$

El concepto de Rendimiento, deberá entenderse como desperdicio en el caso de Materiales y como rendimiento en caso de una Mano de Obra.

HERRAMIENTAS

Las herramientas son porcentajes que se podrán aplicar sobre los componentes al elaborar un Análisis. Se considerarán los siguientes porcentajes de herramientas:

Tipo	Descripción	Se calcula sobre:
H	Herramientas	El total de la Mano de Obra y Cuadrillas.
HI	Mandos Intermedios	El total de la Mano de Obra y Cuadrillas.
HM	Porcentaje sobre Materiales	El total de Materiales.
HQ	Porcentaje sobre Maquinaria y Equipo	El total de la Maquinaria y Equipo.
HT	Porcentaje sobre Subcontratos	El total de Subcontratos.
HP	Porcentaje sobre Preliminares	El total de Preliminares: Básicos, Integrados, Partidas, Capítulos y Obra Extraordinaria como elementos de un Análisis.

La información que se proporcionará para una herramienta que se incluirá dentro de un Análisis será:

- 1.-Tipo; tipo de herramienta que se desee agregar.
- 2.-Clave; clave de la herramienta.
- 3.-Descripción; descripción de la herramienta podrá contener dígitos, letras, espacios y cualquier signo del teclado estándar.

4.-Unidad; unidad de cálculo para la herramienta.

UNA HERRAMIENTA COMO UN ELEMENTO DEL ANÁLISIS

Para adicionar una herramienta como un elemento del Análisis se proporcionará:

1.-Tipo, tipo de herramienta que se desee adicionar:

Tipo	Descripción
H	Herramienta
HI	Mandos Intermedios
HM	Porcentaje sobre Materiales
HQ	Porcentaje sobre Maquinaria y Equipo
HT	Porcentaje sobre Subcontratos
HP	Porcentaje sobre Preliminares

2.-Clave; clave de la herramienta que se desee adicionar.

3.-Cantidad; porcentaje a aplicar de la herramienta proporcionada.

4.-Rendimiento, Rendimiento de la Herramienta, el importe para la herramienta será calculado de la siguiente manera:

$$\text{Importe} = (\text{Total de la Herramienta} / \text{Rendimiento}) * \text{Costo Unitario}$$

PRESUPUESTO

Este módulo permitirá actualizar los precios de los Análisis y de sus elementos.

El concepto de Rendimiento, deberá entenderse como desperdicio en el caso de Materiales y como rendimiento en caso de una Mano de Obra.

HERRAMIENTAS

Las herramientas son porcentajes que se podrán aplicar sobre los componentes al elaborar un Análisis. Se considerarán los siguientes porcentajes de herramientas:

Tipo	Descripción	Se calcula sobre:
H	Herramientas	El total de la Mano de Obra y Cuadrillas.
HI	Mandos Intermedios	El total de la Mano de Obra y Cuadrillas.
HM	Porcentaje sobre Materiales	El total de Materiales.
HQ	Porcentaje sobre Maquinaria y Equipo	El total de la Maquinaria y Equipo.
HT	Porcentaje sobre Subcontratos	El total de Subcontratos.
HP	Porcentaje sobre Preliminares	El total de Preliminares; Básicos, Integrados, Partidas, Capítulos y Obra Extraordinaria como elementos de un Análisis.

La información que se proporcionará para una herramienta que se incluirá dentro de un Análisis será:

- 1.-Tipo; tipo de herramienta que se desee agregar.
- 2.-Clave; clave de la herramienta.
- 3.-Descripción; descripción de la herramienta podrá contener dígitos, letras, espacios y cualquier signo del teclado estándar.

Si se lleva a cabo una modificación al costo de algún insumo que intervenga en el presupuesto, será necesario recalcular los precios para obtener el costo real de la obra.

Así mismo, se deberá poder elaborar cálculos para definir cualquier factor de sobre costo que se requiera aplicar a los conceptos del presupuesto y a los análisis.

ESPECIFICACIÓN DE LOS FORMATOS DE SOBRE COSTO

Para definir un formato de sobre costo se deberá proporcionar la siguiente información:

- 1.-Nombre; nombre con el cual se identificará el formato del factor de sobre costo que se desee definir.
- 2.-Información de los Conceptos; este campo contendrá variables establecidas para definir el formato del factor de sobre costo, estas serán:

Variable:	Que es el:
Costo	Costo Unitario del concepto en cuestión, sobre él se aplicará el sobre costo.
TotMat	Total de Materiales del concepto en cuestión.
TotManOb	Total de Mano de Obra del concepto en cuestión.
TotMaqEq	Total de Maquinaria y Equipo del concepto en cuestión.
TotSubCont	Total de Subcontratos.

Se deberá permitir crear nuevas variables proporcionando el nombre, la descripción, así como, la fórmula de la cual se obtendrá el valor de la nueva variable, se podrán utilizar las variables definidas, ó bien, las definidas por el usuario.

Ejemplo. Si se desea definir la variable de "Indirectos" la cual se define como:

$$\text{Indirectos} = \text{Costo Unitario} * 0.25$$

Se deberá proporcionar la siguiente fórmula de acuerdo con las variables antes mencionadas:

$$\text{Costo} * 0.25$$

EXPLOSIÓN DE INSUMOS

Dentro de esta opción se podrá realizar la Explosión de Materiales o Cubicación para la obra presupuestada, así como para Capítulos, Partidas, Integrados y Extraordinarios en forma independiente.

Los datos que representarán a cada insumo resultante de la Explosión (Materiales, Mano de Obra, Maquinaria y Equipo, Subcontratos y Herramientas), serán la clave, descripción, unidad, costo unitario, cantidad, importe y porcentaje de incidencia con respecto al importe total de la obra, capítulo, partida o integrado explotado.

Para realizar la Explosión de Insumos se deberá proporcionar:

- 1.-Nivel de Explosión; indicará para que elemento se realizará la Explosión de Insumos, la explosión podrá ser:**

Por:
Obra
Integrado
Partida
Capítulo
Extraordinario

2.-El nombre del atributo del insumo por el cual se desee ordenar la explosión.

Al realizar la Explosión de Insumos, se deberán mostrar los datos asociados a los insumos resultantes de esta explosión (Clave, Descripción, Unidad, Costo Unitario, Cantidad, Importe y el Porcentaje de Incidencia con respecto al Importe Total de la Obra, Capítulo, Partida o integrado explotado). Así como, el Importe Total de la Obra, Capítulo, Partida o Integrado, la suma total de los insumos de cada grupo de insumos incluyendo la Herramienta, así como el porcentaje de incidencia total.

RELACIONES

Dentro de este módulo se permitirá registrar la información de las Zonas Económicas, las Familias en las que se agruparán los Insumos o Análisis y capturar los diferentes tipos de Monedas que se utilizarán dentro de la obra.

- Dentro de Zonas se registrará la descripción de la zona, las regiones asociadas con ésta y presentar los insumos que contienen a dicha zona.
- Dentro de Familias se registrará la descripción de la familia y presentará los insumos o análisis que están contenidos en dicha familia.
- En el Catálogo de Monedas se registrará el nombre de la moneda, el símbolo con el cual se representa, el tipo de cambio, así como definir la moneda base para el cálculo del presupuesto.

EDICIÓN DE UNA ZONA

Se deberá permitir asociar el precio de un insumo a una zona económica y tener una cantidad ilimitada de zonas asociadas a un insumo.

Una zona deberá tener los siguientes atributos:

- 1.-El nombre de la zona económica que se desee capturar.
- 2.-Descripción; descripción de la zona.
- 3.-Las entidades relacionadas con la zona económica, para una zona se podrán tener las entidades que se deseen.

EDICIÓN DE UNA FAMILIA

Una familia es una agrupación de insumos. Cada familia podrá tener dentro de sus elementos a insumos de diferentes tipos.

Una familia estará representada por los siguientes atributos:

- 1.-El nombre de la familia que se desee registrar.
- 2.-La Descripción de la familia, que será tan grande como se desee y podrá contener dígitos, letras, espacios y cualquier signo del teclado estándar.

CATÁLOGO DE MONEDAS

Dentro del sistema se podrán manejar diferentes tipos de moneda para definir los costos de los insumos, y combinar estos tipos para la elaboración del presupuesto, en éste módulo se podrán definir las monedas que se manejarán dentro de la obra. Permitiendo al usuario proporcionar un tipo de cambio de moneda diferente y en base a ello actualizar los análisis elaborados.

Para definir un tipo de moneda será necesario:

- 1.-La descripción de la moneda.
- 2.-El símbolo con el cual se representará la moneda.
- 3.-El tipo de cambio con respecto a la moneda base.
- 4.-La fecha en la cual se capturó el tipo de cambio.

II. CONTROL DE OBRA

En este módulo, se permitirá llevar a cabo el control de las obras presupuestadas previamente con el módulo de Precios Unitarios, los submódulos que lo integrarán serán los siguientes:

Cálculo de Escalatorias

- Captura de Incrementos
- Cálculo manual de Incrementos
- Cálculo en forma automática
- Reportes de Incrementos

Proyección de Incrementos

Estimaciones Normales

- Cálculo de Estimaciones
- Explosión de Estimaciones
- Reportes de Estimaciones

Estimaciones Escaladas

- Cálculo de Estimaciones Escaladas
- Reportes de Estimaciones Escaladas

ESTIMACIONES

Este módulo permitirá elaborar las estimaciones periódicas de todas las obras que se hayan presupuestado.

Para la elaboración de estimaciones se requerirá la clave del concepto y la cantidad elaborada. Deberá proporcionarse:

- 1.-Clave; clave o número de estimación.
- 2.-Descripción de la estimación; la descripción puede ser tan grande como se desee y podrá contener cualquier signo del teclado estándar.
- 3.-Fecha de Inicio del período de la estimación.
- 4.-Fecha de Terminación del período de estimación.
- 5.-Tipo de Análisis; tipo al que pertenece al análisis del cual se van a estimar sus conceptos.
- 6.-Clave Análisis; clave del análisis del cual se van a estimar sus conceptos.
- 7.-Tipo Elemento; tipo al que pertenece el concepto a estimar.
- 8.-Clave Elemento; clave del concepto a estimar.
- 9.-Cantidad Estimada; el volumen ejecutado del concepto en esta estimación, el total acumulado se calculará después de haber proporcionado la cantidad estimada como:

Acumulado Anterior + Cantidad Estimada

y el importe para el concepto como:

Cantidad Estimada * Precio Unitario

ó

Cantidad Estimada * Costo Unitario

El Total de toda la estimación se calculará como:

Suma de los Importes de cada uno de los conceptos estimados

III. MANEJO DE OBRAS

El módulo de manejo de obras permitirá administrar las diferentes obras, este estará compuesto por:

Selección de Obras

Copiado de Obras

Borrado de Obras

Reconstrucción de Obras

IV. OPCIONES

En este módulo se encontrarán diversas utilerías que permitirán la selección de monitor, impresora, colores e intensidades de los elementos que se despliegan en la pantalla. Este estará compuesto por:

Configuración

Monitor

Default

Color

Blanco y Negro

Monocromático

Colores

Impresoras

Rango de redondeo

Formato de Reportes

4.1.5 INTERFACE AL USUARIO

Es importante recordar que uno de los objetivos a cubrir es proporcionar al usuario una interface fácil de entender y que no requiera de una larga etapa de aprendizaje. Considerando este punto como uno de los más importantes en el diseño de la interface del sistema.

Como un siguiente punto de importancia debemos tomar en cuenta que al ofrecer una aplicación en Ambiente Windows debemos cumplir con los estándares que ya están definidos para las aplicaciones de este tipo, a continuación se presentan los criterios a utilizar en el diseño de la interface del programa de precios unitarios.

En el ambiente MS-Windows, el control de un proyecto está en manos del desarrollador. La interface al usuario debe ser funcional, elegante y evitar confusiones. Los resultados de una interface pobre en el diseño puede traer como consecuencia una mala impresión del producto al usuario lo cual a su vez lleva a un nivel de aceptación mucho menor a lo esperado. Microsoft ofrece una amplia variedad de guías de diseño de interfaces para aplicaciones Windows que cumplen con sus estándares. El que una aplicación siga los patrones presentados en una guía no implica que sea una buena interface, ya que existen otros aspectos a considerar. La aplicación debe ser fácil de leer, de observar y de usar en comportamiento y organización. Para diseñar una interface funcional y visualmente confortable se requiere conocer las necesidades del mercado, los requerimientos de negocio de la aplicación y el nivel informático del mercado en el que se venderá el producto.

Los usuarios no compran o utilizan productos que se perciben irresponsables o difíciles de usar.

Debe recordarse que no todas las personas piensan como nosotros (iniquidad de percepción), por lo tanto no debemos actuar como si así fuera, además de que siempre existirán opiniones a favor y en contra de la funcionalidad de una interface y que podría ser mejor y que puntos no son útiles, lo más recomendable en estos casos es buscar una interface que cumpla con características generales y que tenga ventajas o funcionamiento adicional para aquellos usuarios que lo prefieran. La complejidad no es negativa, sólo que debe ser ingeniada de tal forma que permita a los usuarios tomar el control de las acciones y que además las acciones más comunes se lleven a cabo de manera rápida con un simple pulsar de teclas o algún botón.

El objetivo a seguir es automatizar operaciones que lleva a la reducción de tiempos de captura que a fin de cuentas representa un ahorro de dinero. Es muy importante agrupar las acciones de acuerdo al objetivo que persiguen y que vaya enfocado al fácil entendimiento por parte del usuario.

El comportamiento de mucho controles en Windows ya se encuentra definido, no debe cambiarse este comportamiento a menos que la acción del control no sea similar a la utilizada en los demás programas ya que en caso contrario estaríamos alterando los estándares.

A continuación se muestra una lista de puntos a seguir en el diseño de la interface del programa de precios unitarios:

además de generar confusión cada vez que se requiera dar de alta nuevos datos.

- Otro aspecto que debe considerarse es el orden en que la información será proporcionada por el usuario debe seguir una secuencia lógica, es decir, el inicio de un grupo de datos debe ser continuo en el orden de captura, sin necesidad de solicitar un dato que no pertenezca al grupo, además de que los grupos deberán seguir una secuencia de acuerdo a sus importancia, para que en caso de que el usuario no desee dar la información completa de un registro, la información sea faltante que no es importante se encontrará al final de la secuencia. Para la realización de este objetivo, es posible ayudarse mediante el uso de diagramas de secuencia.
- Es importante familiarizarse con alguna técnica de diseño gráfico, que pueda auxiliarnos en el diseño de pantallas atractivas para el usuario a primera vista, debe olvidarse el concepto de 'Detalles de interface' o 'Error de Cosmético' ya que ello representa un punto clave para la satisfacción del usuario.
- Es posible apoyarse en consultas a revistas y artículos sobre aplicaciones exitosas que tengan una interface agradable al usuario y determinar cuales son los puntos de la interface de esas aplicaciones que más agradan al usuario.

IDEAS PARA LA PREVENCIÓN DE ERRORES

Deshabilitar las opciones que puedan afectar el funcionamiento de una parte de la aplicación, es decir, cuando las condiciones de una pantalla no permitan la realización de alguna otra actividad deberá deshabilitarse esa otra opción, por ejemplo, cuando el usuario abre una pantalla de captura y no ha proporcionado ninguna información

Cuando se requiere la captura de algún dato localizado en otro catálogo dado de alta previamente, es mucho más eficiente dar al usuario una herramienta de consulta selectiva que solicitarle teclee una clave, esto ofrece dos ventajas primero asegura que el dato proporcionado es válido, y segunda permite al usuario consultar los demás elementos asegurando así que el seleccionado es el correcto.

Antes de realizar una acción destructiva deberá forzarse al usuario a confirmar la acción, muchas veces se presionan teclas por error ocasionando así pérdidas de información y tiempo, en esta confirmación deberá indicarse la acción, por ejemplo, si el usuario activó el mecanismo de borrado de un registro o de un conjunto de registros, el sistema deberá un mensaje parecido a los siguiente *"Desea Borrar la información ?"*, la respuesta sugerida por el sistema debe ser negativa para que en caso de que el usuario presione teclas adicionales no se proceda con tal acción destructiva. Este es uno de los puntos que evitarán al usuario ejecutar acciones no deseadas y ayudará al sistema a proteger la información, consiguiendo así un punto más de satisfacción al usuario.

MENSAJES DE ERROR

Uno de los problemas más comunes es que cuando se llega a presentar un mensaje de error en una aplicación, el usuario no tiene ni la más remota idea de cual es la causa que ha generado dicho error, además que regularmente el mensaje no indica cuales son las consecuencias que dicho error puede acarrear y lo más importante no indica al usuario las formas en que dicho error puede ser eliminado. Por estas razones es indispensable que al enviar un mensaje de error el sistema le proporcione al usuario la siguiente información:

Posibles causas del error: Es decir, presentar una breve descripción del error que sugiera cuales son las causas que lo originan, si falta algún dato, si la acción anterior no fue terminada satisfactoriamente, no existe espacio suficiente en disco, se ha terminado la memoria disponible.

Consecuencias del error: Debe indicar también que al haberse presentado el error serán deshabilitadas algunas funciones, que no podrá consultar la información, que los datos no quedarán registrados en el sistema o que los resultados de los cálculos realizados no serán correctos.

Posibles Soluciones: Es importante decir al usuario cuales son las acciones que deba tomar para corregir dicho error, dependiendo de la gravedad del mismo habrá ocasiones en que la recomendación adecuada será comunicarse al

ocasiones en que la recomendación adecuada será comunicarse al departamento de soporte técnico, pero en aquellas ocasiones en que el problema sea relativamente sencillo deberá detallarse al usuario los diferentes procedimientos que pueden seguirse para eliminar el error. Lo peor que puede hacerse como desarrollador es enviar el mensaje "Este Error No Debe Ocurrir" ya que ello causa un impacto en el usuario que lo hace pensar en dejar de utilizar el sistema además de

PUNTOS ADICIONALES

En ambiente Windows podemos utilizar el mouse y el teclado, una aplicación debe proporcionar al usuario la facilidad de ejecutar acciones o procesos con el mouse con el teclado de forma que sea el usuario quien decida como prefiere trabajar. No deben implementarse mecanismos en los cuales el usuario no pueda ejecutar alguna utilidad solo con el teclado o sólo con el mouse. Se debe utilizar la característica de los aceleradores, que son una forma muy sencilla de ejecutar acciones de manera rápida sin forzar al usuario a utilizar el mouse o recorrer los demás campos hasta llegar al deseado. La combinación de teclas se prepara utilizando la tecla <ALT> más una letra del nombre del campo o botón al cual se quiere llegar .

FONTS, COLORES Y CONFIGURACIÓN.

Si recordamos, anteriormente se mencionó que cada usuario piensa diferente por lo tanto sus gustos son diferentes, debemos señalar que las opciones de colores y el tipo de letra con el cual se presentan las ventanas deberá ser de acuerdo a la configuración que el usuario defina, sin olvidar la opción de una configuración predeterminada por el sistema. En el ambiente Windows existen utilerías que determinan las preferencias de colores y fonts ha utilizarse, por lo tanto todas las aplicaciones en deberán respetar y utilizar dichas configuraciones.

Al elaborar la configuración predeterminada por el usuario deben seleccionarse colores que permitan ofrecer una buena impresión y no una combinación de colores que parezca circo.

4.1.5.1 DISEÑO DE FORMAS DE EDICIÓN DE DATOS

A continuación se presentarán las principales características de los controles utilizados en las pantallas de edición y captura de datos. Se ha decidido utilizar los controles más comunes para que el usuario no tenga ningún problema en identificar su funcionamiento, ya que en diferentes aplicaciones los controles funcionan de manera similar, en el sistema de precios unitarios además de ofrecer ventajas adicionales.

Se tomará como base la pantalla principal del sistema, en donde se mencionarán las ventajas de contar con los controles ahí localizados.

Se tomará como base la pantalla principal del sistema, en donde se mencionarán las ventajas de contar con los controles ahí localizados.

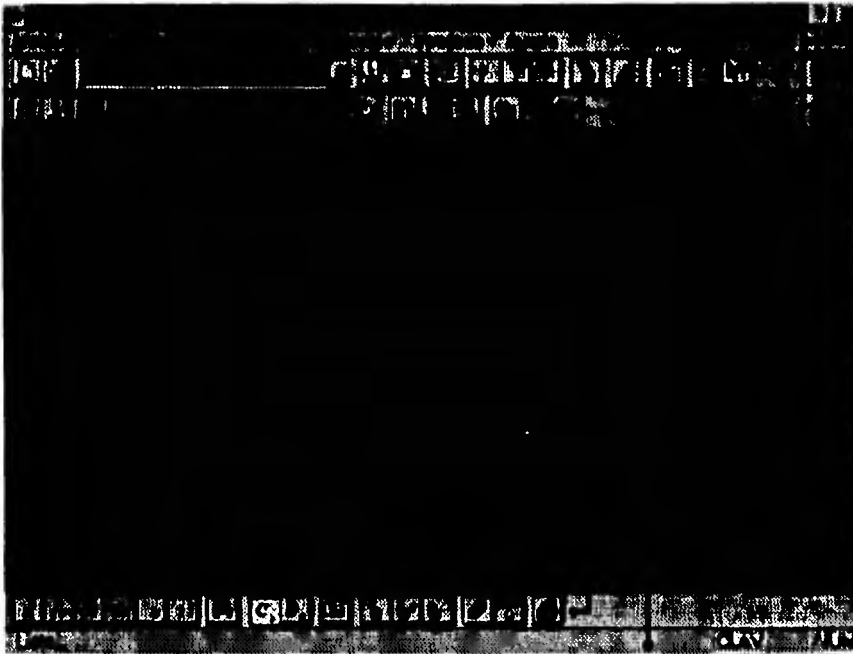


Fig. 4.1.5.1.1 Ventana Principal

MENÚS DESCOLGANTES

Este tipo de menús es muy importante, en primer lugar cumple con los estándares de las aplicaciones de ambiente Windows, además permite trabajar cómodamente a todos aquellos usuarios que no gustan de utilizar el mouse en exceso, los menús pueden ser activados presionando sus teclas aceleradoras, que es la combinación de la tecla

Dentro de éstos menús se deben encontrar todas las funciones del sistema, es decir, el usuario debe encontrar aquí la entrada a todos los procesos que el sistema es capaz de realizar. Muchas veces es más rápido el acceso a través de menús que con algún otro mecanismo, si el usuario es nuevo en el sistema, a través de los menús podrá conocer el sistema.

PANEL DE OBRAS

Este control no es estándar en todas las aplicaciones, en base a las necesidades de cada aplicación podrá existir o no, además de que se presenta de diferentes formas y con diferentes nombres dependiendo de la función que realice. Para este caso el panel de obras consiste en una lista de los nombres de las obras que el usuario tiene abiertas en un momento determinado, mediante este panel el usuario podrá activar una obra de las ya abiertas diferente de la actual, es muy importante mencionar que el nombre de la obra activa siempre estará a la vista del usuario a través de este control y que es independiente de las ventanas abiertas que se tengan, es decir, el usuario podrá tener abiertas varias ventanas que no necesariamente pertenecen a la misma obra, pero conforme se mueva en cada una de ellas, la obra activa se cambiará por la correspondiente a la ventana en consulta.

BARRA DE HERRAMIENTAS

La barra de herramientas es un conjunto de botones que permiten al usuario ejecución de funciones de una forma más rápida e ilustrativa, ya que cada botón contiene un icono referente a la función que realizan. Básicamente la barra de herramientas realiza las mismas funciones que el menú y aquí se encuentran disponibles las principales funciones del sistema. Estas opciones se ejecutan colocando el apuntador del mouse en la función deseada y presionando el botón izquierdo se activa el procedimiento. El usuario debe saber cual es el botón correspondiente a cada acción, pero como una ventaja adicional el sistema ofrece una ayuda para el manejo de estos botones, dicha ayuda consiste en que al colocar el apuntador del mouse sobre un botón, automáticamente aparecerá en la parte inferior de dicho botón el nombre de la función que se ejecutará al presionarlo. Otra ventaja es que este conjunto de botones podrán cambiarse de su lugar original en la pantalla, el usuario podrá definir en que parte lo quiere visualizar, ya sea en la parte inferior de la ventana, en el costado izquierdo o en el derecho, con ello se brinda la facilidad al usuario a definir las características y configuración que el desea para el sistema, si por alguna razón no desea que aparezca esta barra de herramientas también podrá indicarlo al sistema.

CONTROLES WINDOWS

Al ser una aplicación bajo ambiente se deben cubrir los estándares, es por ello que todas las aplicaciones Windows tiene características básicas comunes, por lo tanto el sistema de precios unitarios aquí planteado también debe cumplirlos y por lo tanto cuenta con los controles para Mover la Ventana, Ajustar Tamaño, Minimizar, Maximizar y Cerrar, así como también las incluye una área especial donde se coloca el nombre asociado a cada pantalla. Del lado derecho cuenta con dos botones para minimizar y maximizar el tamaño de la ventana independientemente de poder ejecutar estas funciones con el control que aparece en el lado izquierdo.

CONTROLES DE EDICIÓN

En todas la ventanas en las cuales el usuario ingrese datos, se activará la barra de edición en la cual encontramos la forma rápida de acceso a funciones que sólo son válidas cuando se esta dando de alta registros o cuando se están modificando algunos de ellos. Estas funciones tienen también su correspondiente opción dentro de los menús descolgantes y dichas opciones también se activarán únicamente cuando las ventanas de edición se encuentren en uso.

Primer Registro: Obtiene la información del primer registro correspondiente al tipo activo (material, mano de obra, equipo, subcontrato, análisis) y la muestra en los campos correspondientes.

Registro Anterior: Del último registro consultado en pantalla, se obtiene el anterior de acuerdo al orden por clave o por descripción y se presenta la información en pantalla.

Registro Siguiete: Del registro actual permite avanzar al siguiente, si es que existe en orden correspondiente a clave o descripción, obtiene los datos y los muestra en pantalla.

Ultimo Registro: Permite acceder el último dato del catálogo de registros, si es que existe al menos uno, en tal caso obtiene los datos y los muestra en pantalla.

Grabar: Independientemente de que exista la tecla para grabar un registro, mediante el uso de este botón podremos ejecutar esta acción de forma fácil y sencilla.

Navega por Clave o Descripción: Este botón funciona como un selector, es decir, el usuario deberá decidir si quiere hacer recorridos en función a la descripción de los registros en el catálogo o en función a las claves.

Consulta a Catálogo: Permitirá abrir la ventana de catálogo que ofrece todos los registros dados de alta en el sistema, con el objetivo de que el usuario consulte los que necesite o incluso los copie a su ventana actual.

Panel de Reportes: Cuando el usuario ha modificado información contenida en los registros, es muy probable que vaya a imprimirla, por ello deberá estar disponible la opción de generar algún reporte.

BARRA DE ESTATUS

Esta sección de la pantalla del sistema es de suma importancia porque es aquí donde deberá enviarse información adicional del sistema, en cuanto a las operaciones que realiza el usuario. Tal información puede ser de utilidad para el usuario permitiéndole así un mejor uso del sistema.

A continuación se presentan pantallas principales del sistema, en donde se han aplicado los conceptos explicados anteriormente, en ellas se muestra una interface atractiva al usuario y sobre todo que invita a una operación sumamente sencilla. Existen algunos controles adicionales a los mencionados cuya función varía dependiendo del módulo al que pertenece la ventana y por ello se explican al presentar cada ventana.

La ventana de captura, edición y consulta para materiales se define de la siguiente forma:

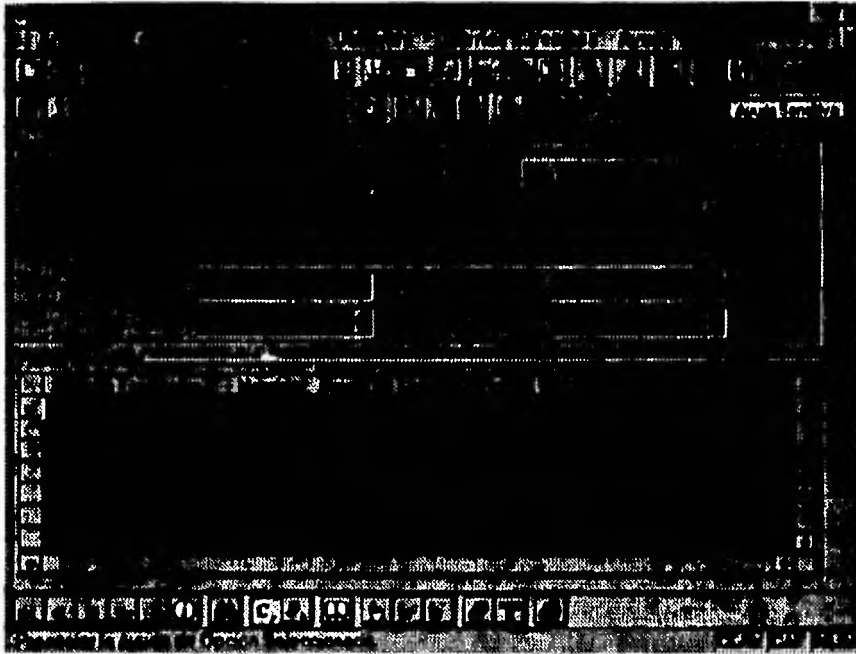


Fig. 4.1.5.1.2 Ventana Materiales

El uso de controles tipo hoja de cálculo llamados "Spread" es de suma importancia; regularmente los usuarios están acostumbrados a este tipo de controles porque cualquier hoja de cálculo disponible en el mercado los contiene y su uso es muy sencillo. En las ventanas de insumos se han diseñado los spreads para contener los precios correspondientes a diferentes zonas para un insumo determinado, de forma tal que el usuario puede consultar todos los precios disponibles y asignar con cual de ellos desea trabajar. Estos controles se encuentran en las ventanas de captura por Materiales, Equipo, Obra de Mano y Subcontratos.

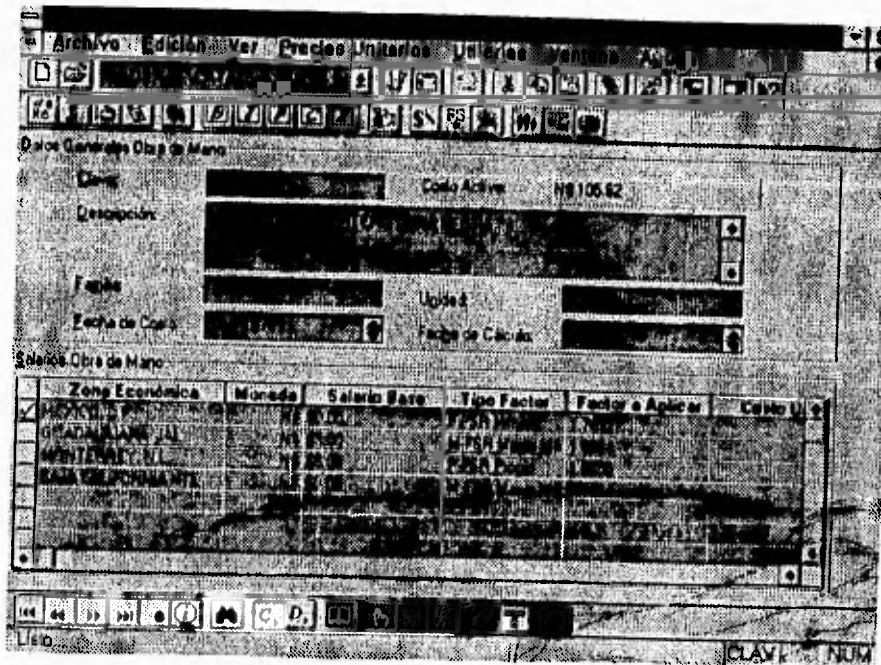


Fig. 4.1.5.1.3 Ventana de Mano de Odra

En este caso tenemos una interfaz muy similar a la ventana de materiales, el objetivo es que el usuario se familiarice con un mismo funcionamiento y no le tome más tiempo aprender a usar cada ventana. Las columnas dentro del spread representan algunos datos correspondientes a los registros de mano de obra. El tipo de factor de salario real será presentado en una pequeña lista donde el usuario podrá seleccionar cual es el adecuado para esa categoría de mano de obra, si desea proporcionar un factor diferente podrá hacerlos en la columna donde se indica el factor.

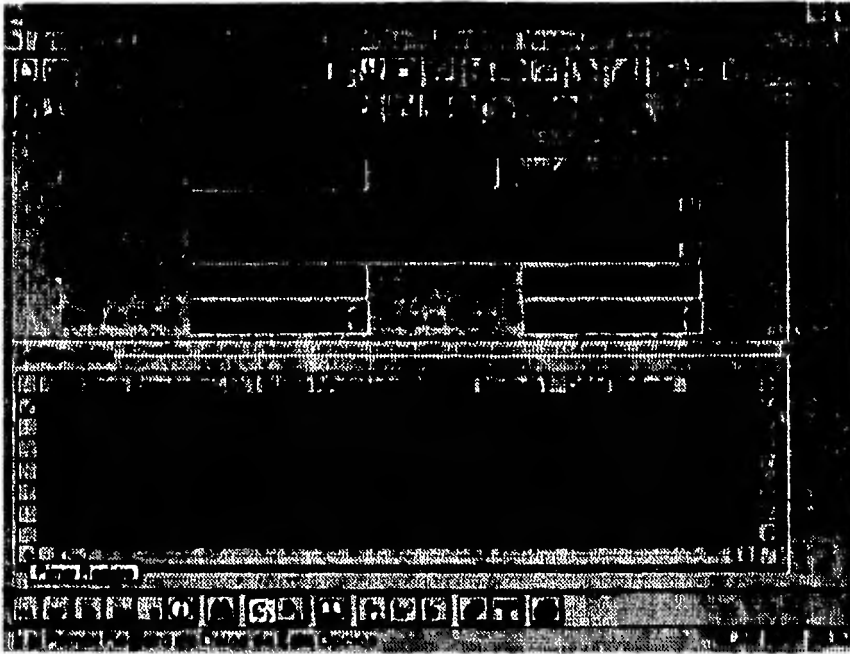


Fig. 4.1.5.1.4 Ventana Subcontratos

Para el caso de Subcontratos se utiliza el spread para dar de alta a los proveedores del servicio, los cuales pueden ir asociados a diferentes zonas económicas, el funcionamiento de la ventana es similar al de los demás insumos.

Es muy importante recordar que es posible tener diferentes ventanas abiertas al mismo tiempo, con el objetivo de consultar o editar información en cualquier parte dentro del programa.

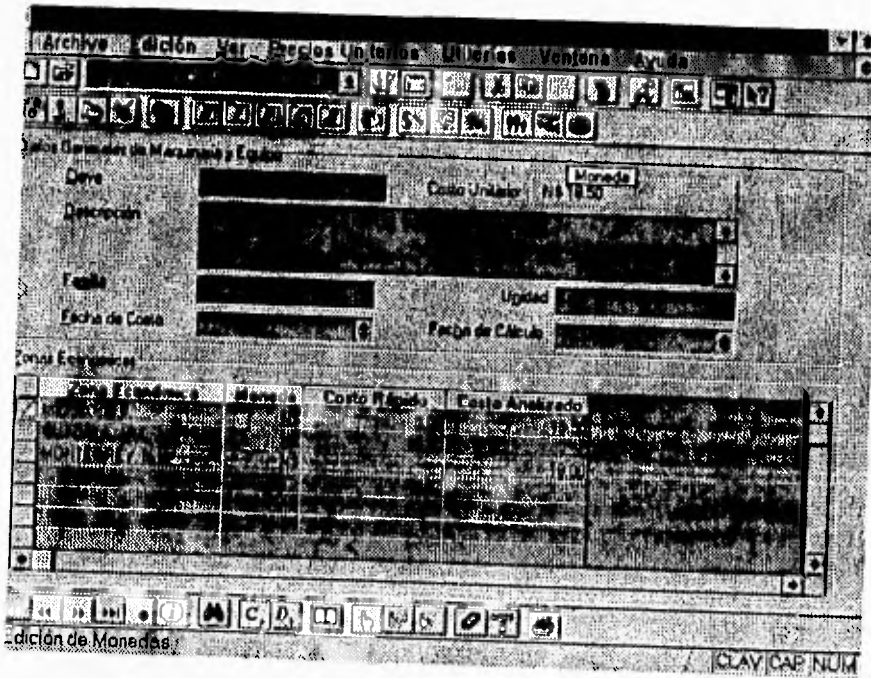


Fig. 4.1.5.1.5 Ventana de Maquinaria y Equipo

Para la ventana de Maquinaria existen diferentes alternativas de asignar un costo, al igual que en los demás insumos contamos con el spread de consulta, pero en este caso el costo por hora puede ser asignado directamente o en caso contrario debe analizarse, como para el análisis se requiere de más datos y cálculos, es necesario contar con ventanas adicionales, las cuales tendrán características muy similares y permitirán al usuario trabajar de una manera organizada y agrupando los conceptos que corresponden a un mismo cálculo, a continuación se presentan estas ventanas como un ejemplo de la interfaz que debe ofrecerse al usuario.

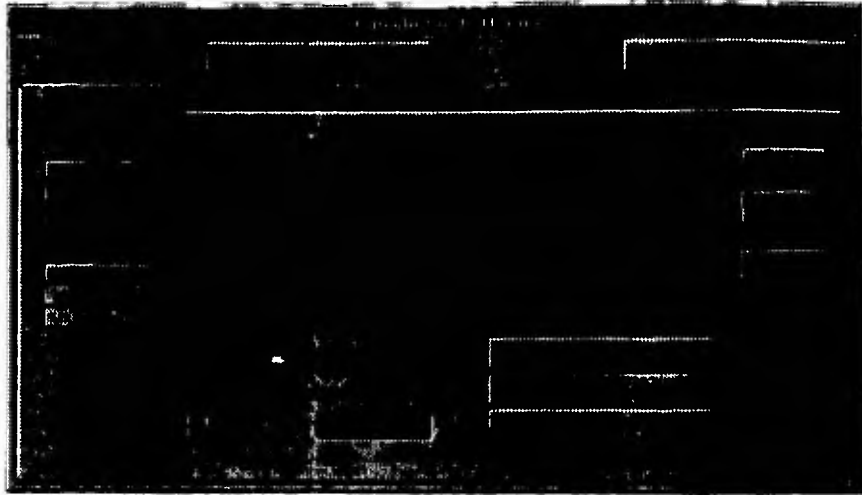


Fig. 4.1.5.1.6 Ventana de Maquinaria y Equipo- Gastos Equipo

En esta ventana se encuentran agrupados los costos base de la maquinaria y el equipo adicional necesario, también se incluyen los registros de los costos de llantas, estos son seleccionados del catálogo de materiales, en esta ventana el objetivo es obtener el valor de adquisición y el valor de rescate.

Cálculo Costo Horario			
Costo Sin Operación		Costo con Operación	
NS 59 10		NS 70 90	
Gastos Equipo	Factores	Mólv	Gerenc Operación
Vida Económica en Horas (Ve):	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/> Borrar
Vida de los Lléntes en Horas (HVLL)	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/> Calcular
Horas Anuales (Ha):	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Ayuda
Prima de Seguro Anual (%):	<input type="text"/>	<input type="text"/>	
Factor de Almacenaje (Fa):	<input type="text"/>	<input type="text"/>	
Factor de Mantenimiento (M):	<input type="text"/>	<input type="text"/>	
Factor de Operación:	<input type="text"/>	<input type="text"/>	
Potencia Nominal:	<input type="text"/>	<input type="text"/>	
Tasa de Interés Anual (%):	<input type="text"/>	<input type="text"/>	

Fig. 4.1.5.1.7 Ventana de Maquinaria y Equipo- Cargos Fijos

En esta segunda ventana se encuentran todos los factores y porcentajes que participan en el cálculo del costo horario.

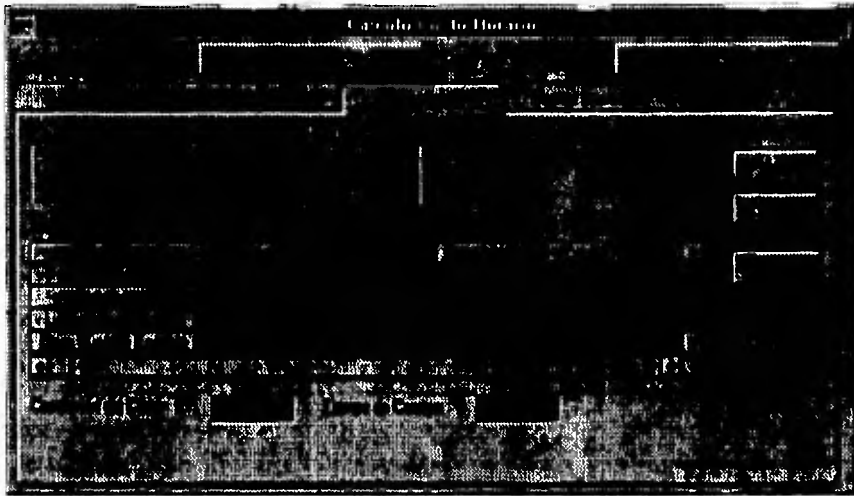


Fig. 4.1.5.1.8 Ventana de Maquinaria y Equipo- Combustibles

En esta ventana se incluyen los datos necesarios para cálculos de consumos del motor como son: gasolina, aceites, lubricantes, aditivos.

Capítulo IV Desarrollo del Sistema

extraordinarios son un concepto adicional que debe utilizarse cuando una vez finalizado un presupuesto se requiere el manejo de conceptos extra.

Ahora bien, en todos estos niveles de análisis la información es similar y por lo tanto las ventanas de edición son iguales, únicamente nos indican el nivel de integración activo, con ello el usuario podrá compartir información a diferentes niveles.

Un punto importante, al aprender a elaborar un presupuesto básico utilizando el sistema automáticamente estará listo para elaborar todo un presupuesto al nivel que desee.

En estas ventanas podrá utilizar el spread para almacenar la información de todos los componentes de un análisis determinado, asignar su cantidad, rendimiento, incluso si el componente no ha sido dado de alta y se trata de un insumo podrá crearlo en esta sección únicamente proporcionando sus datos.

The screenshot shows a software window titled "Factores Sobre Costo". At the top, there is a menu bar with options: Archivo, Edición, Ver, Precios, Unitarios, Utilidades, Ventana, Ayuda. Below the menu is a toolbar with various icons. The main area contains a data entry form with fields for "Clave", "Descripción", "Familia", "Unidad", "Fecha de Costo", and "Tipo de Trabajo". Below the form, there is a table with columns: Tipo, Clave, Descripción, Unidad, Cantidad, Minuto, and Costo Un. The table contains 10 rows of data. At the bottom of the window, there is a status bar with the text "Eleccion de los Factores de Sobre Costo" and "CLAVI CAP NUM".

Tipo	Clave	Descripción	Unidad	Cantidad	Minuto	Costo Un
1
2
3
4
5
6
7
8
9
10

Fig. 4.1.5.1.6 Ventana General de Análisis

4.2 INTEGRACIÓN DEL SISTEMA BASADO EN LAS HERRAMIENTAS DE PROGRAMACIÓN ORIENTADA A OBJETOS

4.2.1 INTEGRACIÓN OLE EN EL SISTEMA DE PRECIOS UNITARIOS

El desarrollo bajo Windows es sumamente desgastante, gracias a avances en el desarrollo de frameworks el esfuerzo de programación para Windows se ve acelerado. Hoy en día contamos con diferentes frameworks como el MFC Versión 2.5 (Microsoft foundation Classes) de Microsoft incluido este en el producto Visual C++, el OFC Borland (Object Components Framework), que sin lugar a dudas ayudan a acelerar el desarrollo de aplicaciones para Windows. Sin embargo a pesar de que ofrecen un mecanismo sencillo para los desarrollos se deben emplear un adecuado análisis y diseño para que el desarrollo sea correcto. Crear una interface orientada al usuario no es sencillo, y es en este punto donde se aprecia el mayor esfuerzo dentro de una aplicación para Windows. Debido a su estructura y diseño el MFC 2.5 tiene una mejor integración con Windows que otros frameworks dentro del mercado. Nuestro producto utiliza los mecanismos ofrecidos por MFC para su creación. Nuestro rol en el desarrollo de la aplicación con MFC es proporcionar el código fuente específico a esta y conectar los componentes definiendo mensajes y comandos que le respondan al usuario. Utilizamos el lenguaje C++ y sus técnicas para derivar nuestras propias clases de las proporcionadas por el framework, con esto logramos una adaptación de dichas clases a nuestra aplicación. A continuación describimos cuáles son los elementos necesarios para la creación de un programa basado en la librerías de MFC.

Área	Lo que nosotros hacemos	Lo que el Framework realiza
Crear un esqueleto de aplicación	Ejecutar al AppWizard, especificar las opciones que deseamos.	Appwizard crea los archivos esenciales para el esqueleto, incluyendo código fuente, documentos, vistas y ventanas Frame; archivos de recursos; un archivo de proyecto.
Un framework que ofrece el framework y AppWizard sin adicionar una línea de código	Contruir el esqueleto, lo que significa compilar y ligarlo. Una vez realizado esto ejecutar la aplicación dentro de Visual C++	El esqueleto ya contiene la funcionalidad básica de toda aplicación. Tal funcionalidad posee comandos estándares para edición, creación y vista de archivos.
Construir nuestra interface al usuario	Utilizar el editor de recursos de Visual C++ para visualmente editar la interface al usuario. <ul style="list-style-type: none"> • Crear Menúes • Definir aceleradores • Crear cajas de diálogo • Crear y editar bitmaps, iconos y cursores • Editar la barra de herramientas • Crear y editar otros recursos 	Los recursos creados por default dentro del editor de recursos proporcionan un punto de partida adecuado para la personalización de la aplicación.
Asociar menús a funciones controladoras.	Utilizar ClassWizard para conectar menús y aceleradores a funciones controladoras en nuestro código.	ClassWizard inserta mapas de mensajes y plantillas de funciones en nuestro código fuente, manejando muchas tareas manuales de codificación.
Escribir nuestro código controlador	Utilizar ClassWizard para saltar directamente al código a modificar. Llenar las funciones de control para completar la interacción con el usuario	Classwizard nos coloca en el editor, y nos posiciona en el lugar exacto en donde se encuentra la plantilla que debemos editar
Asociar botones de la barra de herramientas a comandos	Mapear cada uno de los botones del toolbar a un comando de menú o acelerador, asignándole a dicho botón el ID de comando apropiado.	El framework controla el dibujo, habilitación, deshabilitación, chequeo y otros aspectos visuales de los botones de las barras de herramientas.

Basados en este esquema tenemos que nuestro programa de Precios Unitarios centra su desarrollo en el framework de MFC. Nuestro punto de partida es el módulo AICIC.CPP, el cuál cuenta con la función de inicialización de una aplicación.

Capítulo IV Desarrollo del Sistema

```
//CODIGO: MASTERPRO
//ARCHIVO: SAICIC.CPP
//AUTOR: Román Holvín Cisneros Rojas
//COMENTARIOS: Este archivo contiene las rutinas que manipulan a la aplicacion
// contiene las funciones de inicialización y procesamiento de comandos
//

#define DECLARA_GLOBALES

#include "stdafx.h" //Soporte de HEC
#include "saicic.h"
#include "auxdata.h" //Util para el manejo de diálogos
#include "mainfrm.h" //Manejo del Frame de SAICIC Windows
#include "ipframe.h" //Manejo de In-place activation
#include <obras.h> //Información para manejo de diálogos de creación
//y apertura de una obra
#include <mdigenesa.h> //Manejo del MDI General
#include <farcalcu.h> //Vista del Factor de Salario Real
#include <mdifsr.h> //Manejo del MDI de Factor de Salario Real
#include <mdifamil.h> //Manejo del MDI de Familias
#include <mdlanal.h> //Manejo de MDI de Analisis
#include <analis.h> //Manejo de Analisis
#include <nomarch.h> //Nombre de archivos de SAICIC Windows
#include "borrarar.h" //Rutinas de borrado de SAICIC Windows
#include "saicidoc.h" //Documento general
#include "analidoc.h" //documento particular de análisis
#include "insumosf.h" //Forma general de insumos
#include "siguebus.h"
#include "clavespo.h"
#include "inicia.h"
#include <seccion.h>
#include <twprotoc.h>

#ifdef _DEBUG
#ifdef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
#endif

// =====
// CSaIcicApp
INSTANCE hLibFunc; //Libreria de Funciones SAICIC
INSTANCE hLibRecurs; //Libreria de recursos de SAICIC
char szClave[ sizeof_DESCFAM_SZDESC + sizeof_DESCFAM_ITIPO ];
CConfiguracion oConfigura;
BOOL bglCopiando = FALSE;
BOOL bglNoEditar = FALSE;
TAGEXPORTINFORMACION stglCopiaInfo;

// =====
// The one and only CSaIcicApp object
CSaIcicApp NEAR theApp;

// =====
// CSaIcicApp initialization
static const CLSID BASED_CODE clsid =
( 0x4453cbe1, 0xc82e, 0x101b, ( 0x92, 0xb0, 0x0, 0x0, 0xc0, 0xde, 0x1f, 0x4d ) );

BEGIN_MESSAGE_MAP(CSaIcicApp, CWinApp)
// {AFX_MSG_MAP(CSaIcicApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
ON_COMMAND(ID_UTILERIAS_RECOBRA, OnUtilRecobra)
ON_COMMAND(ID_UTILERIAS BORRAOBRA, OnUtilBorraobra)
ON_COMMAND(ID_WINDOW_NEW, OnWindowNew)
ON_COMMAND(ID_VEN_PANEL, OnVenPanel)
ON_COMMAND(ID_SAICIC_TIPS, OnSaIcicTips)
// }AFX_MSG_MAP
// Standard file based document commands
//ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
//ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
// Standard print setup command
```



```

// COMMAND ID FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
// Global help commands
// COMMAND ID_HELP_INDEX, CWinApp::OnHelpIndex)
// COMMAND ID_HELP_USING, CWinApp::OnHelpUsing)
// COMMAND ID_HELP, CWinApp::OnHelp)
// COMMAND ID_CONTEXT_HELP, CWinApp::OnContextHelp)
// COMMAND ID_DEFAULT_HELP, CWinApp::OnHelpIndex)
END_MESSAGE_MAP()

// Construction
CWinApp::CWinApp()
{
    // TODO: add construction code here.
    // Place all significant initialization here.
    // Lleva a cabo la inicialización de la aplicación
    // Inicializa globales
    m_nCopiaInfo = 1;
    m_nPaquetes = 0;
    m_nMensajes = ARCH_OBRA_INIT;
    m_nPopUpClass = 0;
    m_nMsgBox = NULL;
    m_nSignon = NULL;
    m_nTipScreen = 0;

    m_nNumInTransac = 0;
}

BOOL CWinApp::InitInstance()
{
    // Inicializamos las librerías de OLE 2.0
    if (!AfxOLEInit()) {
        AfxMessageBox( IDS_OLE_INIT_FAILED );
        return FALSE;
    }

    // Configura.fuConfigura ();

    // Permitimos el manejo de VRX
    CMakeJater( AfxGetInstanceHandler() ); // Habilitamos la librería de Efectos 3-D
    C3dViewSubclass( AfxGetInstanceHandler() );

    LoadStandardFileSettings(); // Load standard DDI file options (including MRU)
    // ...

    // Sustituimos la función SetDialogBKColor ya que la librería STL3D.DDI
    // realiza el mismo trabajo
    // set dialog background color to gray

    if (AfxMessageBox( IDS_DIALOG_BACKGROUND_COLOR ) == IDYES ) {
        AfxMessageBox( IDS_DIALOG_BACKGROUND_COLOR );
        return FALSE;
    }

    // ...

    SetDialogBKColor( RGB_WHITEFIELD, VK_F3 );
    SetDialogBKColor( RGB_CURRENTDATE, VK_F11 );
    SetDialogBKColor( RGB_POPUPMENU, VK_F12 );
    // ...

    // ...

    // ...

    return FALSE;
}

```

Constructor de la aplicación, inicialización de propiedades

El método InitInstance es construido dentro de la clase CWinApp, que se deriva de la clase CWinApp

FALLA DE ORIGEN


```

endif
return FALSE;
endif
m_pMainWnd = pMainFrame;

// VerificarUser ("SAICIC", theApp.m_szUsuario, SAICIC_INI,
theApp.m_pMainWnd->m_hWnd );

ColSetNames ( DDInTraeStringLib( m_hInstance, IDST_DIAS_CALENDARIO ),
DDInTraeStringLib( m_hInstance, IDST_MESES_CALENDARIO ) );

if ( !theApp.m_pPanelCentral ) {
    TRACE ("No se logro crear el Panel Central\n");
    return -1; //No se logro crear el dialogo de Panel Central
}

endif

// CheckInterfase ( SAICICN_PRECIOS_UNITARIOS, TW_SAICICN_V40,
m_hInstance, IDS_STRING61216 );

// Buscamos en la línea de comandos si la aplicación fue ejecutada como
// un servidor de OLE
if ( RunEmbedded() || RunAutomated() ) {

    // La aplicación fue ejecutada con /Embedding o /Automation. No se
    // puede presentar la ventana principal en este caso
    return -1;
}

endif

// Registramos información sobre la clase ToolBarSet y sus tips
// Se tiene que llamar a la función de registro antes de crear el toolbar.
// Esto se hace dentro de la función de creación de la ventana principal
// Cuando una aplicación es ejecutada en stand-alone, es una buena práctica
// actualizar el system registry en caso de que este haya sido dañado
m_serverAnalysis.UpdateRegistry ( CAT_REPLACE_SERVER );
CStdAfxFactory::UpdateRegistryAll ();

// Inicializo registroReportes ( hlibRecur, hlibFunc, theApp.m_szUsuario );

if ( TRUE == GetProfileInt ( ID_SECC_PRESENTA, ID_CAMPO_MAXIMO, 0 ) ) {

    m_nCmdShow = SW_SHOWNORMALIZED;
}

endif

// MainFrame->ShowWindow ( m_nCmdShow );
// MainFrame->UpdateWindow ();

// Cuando no existe información para ser procesada, tenemos que abrir
// las carpetas necesarias
if ( !m_pCmdLine [0] != "" ) {

    // Realizar el análisis de una carpeta desde la línea de comandos
    // la aplicación es
    m_pMainFrame->OpenFolderFrom ();
}

else

m_pMainFrame->PostMessage ( WM_INIT_MENU, 0, 0 );

endif

```

Registro de trayectorias para manejo de ayudas. También registramos al usuario SAICIC.

Capítulo IV Desarrollo del Sistema

```
        return TRUE;
    }

    // In this override of OnIdle we are doing UI for our app.
    // Since this needs to be as fast as possible to give the user
    // the best result we do our updates first when lCount is zero
    // then we call the library to do its work.
    //BOOL CMainWinApp::OnIdle(LONG lCount)
    {
        if ( lCount == 0 && NULL != m_pMainWnd )
        {
            ASSERT(m_pMainWnd != NULL);

            // look for any top-level windows owned by us
            // we use 'HWND's to avoid generation of too many temporary CWnds
            for (HWND hWnd = ::GetWindow(m_pMainWnd->m_hWnd, GW_HWNDFIRST);
                hWnd != NULL; hWnd = ::GetNextWindow(hWnd, GW_HWNDNEXT))
            {
                if (::GetParent(hWnd) == m_pMainWnd->m_hWnd)
                {
                    // If owned window is active, move the activation to the
                    // application window
                    //if (GetActiveWindow() == hWnd && (::GetCapture() != NULL))
                    //    m_pMainWnd->SetActiveWindow();

                    // also update the buttons for the top-level window
                    SendMessage(hWnd, WM_IDLEUPDATECMDUI, (WPARAM)TRUE, 0);
                }
            }
        }
        return CWinApp::OnIdle(lCount);
    }

    ////////////////////////////////////////////////////
    // CAboutDlg dialog used for App About

    class CAboutDlg : public CDialog
    {
    public:
        CAboutDlg();
        CBitmapButton m_logo, m_demo;
        // Dialog Data
        //{{AFX_DATA(CAboutDlg)
        enum { IDD = IDD_ABOUTBOX };
        //}}AFX_DATA

        // Implementation
    protected:
        //{{AFX_MSG(CAboutDlg)
        virtual BOOL OnInitDialog();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
    };

    CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
    {
        //{{AFX_DATA_INIT(CAboutDlg)
        //}}AFX_DATA_INIT
    }

    BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
        //{{AFX_MSG_MAP(CAboutDlg)
        //}}AFX_MSG_MAP
    END_MESSAGE_MAP()

    // App command to run the dialog
    void CMainWinApp::OnAppAbout()
    {
        CAboutDlg aboutDlg;
        aboutDlg.DoModal();
    }
}
```


Capítulo IV Desarrollo del Sistema

```
/*
FUNCION: OnFileNew ( iPosicion )
PARAMETROS: int iPosicion Indica la opción que se desea sea abierta para
              crear un nuevo documento

DESCRIPCION: Esta función es muy importante ya que se encarga de crear un
nuevo documento dentro de una obra, de esta forma podemos evitar que
al usuario se le pregunte cual será la opción inicial en la que desea
trabajar
*/
void CSaicicApp::OnFileNew ( int iPosicion )
{
    CString strTypeName;

    if (m_templateList.IsEmpty())
    {
        TRACE("Error : No se han registrado documento con SAICIC\n");
        AfxMessageBox(AFX_IDP_FAILED_TO_CREATE_DOC);
        return;
    }

    CDocTemplate* pTemplate = (CDocTemplate*)m_templateList.GetAt(
        m_templateList.FindIndex ( iPosicion ) );

    ASSERT(pTemplate != NULL);
    ASSERT(!pTemplate->IsKindOf(RUNTIME_CLASS(CDocTemplate)));

    pTemplate->OpenDocumentFile( NULL );
    // if returns NULL, the user has already been alerted

}

void CSaicicApp::OnVenPanel()
{
    if ( m_pPanelCentral->m_plInfoObras.IsEmpty() ) {
        ::MessageBeep ( MB_ICONHAND ); //No puede presentarlo
    }
    else {
        m_pPanelCentral->DoModal ();
    }
}

} // function

CDocument* CSaicicApp::OpenDocumentFile(LPCSTR lpszFileName)
{
    return ( ( CMainFrame * )m_pMainWnd )->OpenDocumentFile( lpszFileName );
}

} // function

void CSaicicApp::OnSaicicTips()
{
    CString strTips = "JI ( \"%saicic.hlp>Tarjeta\",
        \"%HID_TIPS_SAICIC\" )";

    // Mandamos a Invocar a los SAICIC Tips
    AfxHelp ( 0L, HELP_INDEX );
    AfxHelp ( (DWORD)(LPCSTR) strTips, HELP_COMMAND );
}

} // function
```

En esta parte evitamos el comportamiento que ofrece el Win-C ya que no es muy útil para una buena interfaz con el usuario

Punto de Inicio para los SAICIC Tips

Ahora bien, otra clase que es importante para lograr una aplicación en Windows es crear una derivada de la clase CMainFrameWnd. En nuestro caso hacemos la derivación y la llamamos CMainFrame. A continuación mostramos el código de dicha clase

```
// Proyecto: SAICTC Windows 4.0
// Archivo: Mainfrm.cpp
// Fecha de Modificación:
// Modificadores:
// Comentarios:
//   mainfrm.cpp : Implementación de la clase CMainFrame
//   Control del Status Bar
//   Control de los comando de los menús y Toolbar
//

#include <stdafx.h>
#include "saicic.h"
#include <auxdata.h>
#include <mainfrm.h>
#include <clavespo.h>
#include "nomarch.h"
#include "formagra.h"
#include "reporte.h"
#include "wte-sin.h"
#include <mdipensa.h>
#include <scotion.h>
#include <swprotot.h>
#include <sigabus.h>
#include <reporter\repestru.h>

#ifdef _DEBUG
#define THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif

// CMainFrame

IMPLEMENT_DYNAMIC(CMainFrame, CMDIFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CMDIFrameWnd)
    //[[AFX_MSG_MAP(CMainFrame)
    ON_WM_CREATE()
    ON_WM_SYSCOLORCHANGE()
    ON_WM_CLOSE()
    ON_COMMAND(ID_FILE_EXIT, OnFileExit)
    ON_COMMAND(ID_UTILERIAS_XIF, OnUtileriasExporta)
    ON_COMMAND(ID_ARCHIVO_CREA_OBR, OnArchivoCreaObr)
    ON_COMMAND(ID_ARCHIVO_ABRIR, OnArchivoAbrir)
    ON_UPDATE_COMMAND_UI(ID_VEN_PANEL, OnUpdateVenPanel)
    ON_WM_TIMER()
    ON_COMMAND(ID_VIEW_TOOLBAR, OnViewToolbar)
    ON_UPDATE_COMMAND_UI(ID_VIEW_TOOLBAR, OnUpdateViewToolbar)
    ON_COMMAND(ID_CERRAR_OBRA, OnCerrarObra)
    ON_UPDATE_COMMAND_UI(ID_EDIT_UNDO, OnUpdateEditUndo)
    ON_COMMAND(ID_REPORTES_PRESU, OnReportesPresu)
    ON_UPDATE_COMMAND_UI(ID_LISTA_CVE_RECUERDO, OnUpdateListaCveRecuerdo)
    ON_COMMAND(ID_LISTA_CVE_RECUERDO, OnListaCveRecuerdo)
    ON_COMMAND(ID_BUSCAR_CLAVE, OnBuscarClave)
    ON_UPDATE_COMMAND_UI(ID_BUSCAR_CLAVE, OnUpdateBuscarClave)
    ON_COMMAND(ID_BUSCAR_DESCRIP, OnBuscarDescript)
    ON_UPDATE_COMMAND_UI(ID_BUSCAR_DESCRIP, OnUpdateBuscarDescript)
    ON_COMMAND(ID_EXPLOSION, OnExplosion)
    ON_COMMAND(ID_HERRAMIENTA, OnHerramienta)
    ]]]

```

Todo vez que tiene un grupo de mensajes que puede procesar, este mecanismo es la base fundamental de la implantación y procesamiento de los mensajes enviados a una ventana por Windows


```

ON COMMAND(ID_UTILERIAS_CONFIGSAICIC, OnUtileriasConfigsaicic)
ON WM_ACTIVATEAPP()
ON COMMAND(ID_ESTIMACIONES, OnEstimaciones)
ON COMMAND(ID_ACTUALIZA_PRECIOS, OnActualizaPrecios)
ON WM_QUERYENDSESSION()
ON COMMAND(ID_VER OPCIONES, OnVerOpciones)
ON UPDATE_COMMAND_UI (ID_VER OPCIONES, OnUpdateVerOpciones)
ON COMMAND(ID_VER_EDICION, OnVerEdicion)
ON UPDATE_COMMAND_UI (ID_VER_EDICION, OnUpdateVerEdicion)
ON MESSAGE ( WM_INIT_MENU, OnAbreObrasRecientes )
ON COMMAND(ID_AYUDA_ERROR, OnAyudaError)
ON COMMAND(ID_AYUDA_SOPORTE, OnAyudaSoporte)
//]]AFX_MSG_MAP
ON CBN_SELCHANGE ( IDC_COMBO_ACTIVAS, OnChangeObra )
ON UPDATE_COMMAND_UI ( ID_INDICADOR_CLAVE, OnUpdateTBus)
ON COMMAND ( IDC_BANNER, OnCierraBanner )
ON COMMAND ( ID_LIMPIAR_LISTA, OnLimpiarLista )
ON UPDATE_COMMAND_UI ( ID_LIMPIAR_LISTA, OnUpdateLimpiarLista )
ON UPDATE_COMMAND_UI ( ID_ACTUALIZA_PRECIOS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_REPORTES_PRESU, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_CERRAR_OBRA, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_MATERIALES, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_OBRA_DE_MANO, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_SUBCONTRATOS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_BASICOS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_INTEGRADOS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_PARTIDAS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_CAPITULOS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_EXTRAORDINARIOS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_FACTOR SOBRE COSTO, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_FSR, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_FAMILIAS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_ZONAS, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_CAT MONEDA, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID MAQUINA EQUIPO, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_EXPLOSION, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_HERRAMIENTA, OnUpdateVenPanel )
ON UPDATE_COMMAND_UI ( ID_ESTIMACIONES, OnUpdateVenPanel )
ON COMMAND(ID_MATERIALES, OnMateriales)
ON COMMAND(ID_OBRA_DE_MANO, OnObraDeMano)
ON COMMAND(ID_SUBCONTRATOS, OnSubcontratos)
ON COMMAND(ID_BASICOS, OnBasicos)
ON COMMAND(ID_INTEGRADOS, OnIntegrados)
ON COMMAND(ID_PARTIDAS, OnPartidas)
ON COMMAND(ID_CAPITULOS, OnCapitulos)
ON COMMAND(ID_EXTRAORDINARIOS, OnExtraordinarios)
ON COMMAND(ID_FACTOR SOBRE COSTO, OnFactorSobreCosto)
ON COMMAND(ID_FSR, OnFsr)
ON COMMAND(ID_FAMILIAS, OnFamilias)
ON COMMAND(ID_ZONAS, OnZonas)
ON COMMAND(ID_CAT MONEDA, OnCatMoneda)
ON COMMAND(ID MAQUINA EQUIPO, OnMaquinaEquipo)
ON MESSAGE ( REP_SAI_NOTIFICA, OnRepNotifica )
END_MESSAGE_MAP()

```

Este tipo de manejador se encarga de habilitar gráficamente cierto elemento dentro de nuestra interfaz. Se encarga de habilitar o deshabilitar un botón de la barra de herramientas, un menú, un panel del status bar, etc.

```

////////////////////////////////////
// arrays of IDs used to initialize control bars
UINT capture[] = {
    ID_PRIMER_REG,
    ID_REG_ANTERIOR,
    ID_REG_POSTERIOR,
    ID_ULTIMO_REG,
    ID_GRABAR,
    ID_INFORMACION,
    ID_SEPARATOR,
    ID_BUSCAR,
    ID_SEPARATOR,
    ID_BUSCAR_CLAVE,
    ID_BUSCAR_DESCRIP,
    ID_SEPARATOR,
    ID_CATALOGO,
    ID_SEPARATOR,
    ID_RECUERDA_CLAVE,
    ID_LISTA_CVE_RECUERDO,

```

Estos ID's nos permiten obtener comandos desde la barra de herramientas

Capítulo IV Desarrollo del Sistema

```

    ID_LIMPIAR_LISTA,
    ID_SEPARATOR,
    ID_HERRAR,
    ID_LIMPIAR,
    ID_SEPARATOR,
    ID_REPORTES_PRESU
};

static int BASED_CODE buttonsTextCapID[] =
{
    // Colocamos los indices a los que hacen referencia los elementos
    // dentro del toolbar
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8,
    9,
    0,
    4,
    3,
    6,
    10,
    5,
    11,
    12,
    13,
    8,
    14,
    15,
    0,
    16
};

static LPCSTR BASED_CODE szBarTipsCap [] =
{
    "", //0
    "Primer Registro", //1
    "Registro Anterior", //2
    "Siguiente Registro", //3
    "Último Registro", //4
    "Borrar", //5
    "Información", //6
    "Iniciar", //7
    "Navegar por Clave", //8
    "Navegar por Descripción", //9
    "Presentar Catálogo", //10
    "Recordar Clave", //11
    "Lista de Recordatorio", //12
    "Limpiar Lista", //13
    "Herrar Registro", //14
    "Limpiar Forma", //15
    "Panel de Reportes" //16
};

// toolbar buttons - IDs are command buttons
UINT buttons[] =
{
    // same order as in the bitmap 'toolbar.bmp'
    ID_ARCHIVO_CREA_OBR, //
    ID_ARCHIVO_ABRIR,
    ID_SEPARATOR,
    ID_SEPARATOR, //ES EL COMBOBOX
    ID_SEPARATOR,
    ID_CERRAR_OBRNA,
    ID_VEN_PANEL,
    ID_SEPARATOR,
    ID_EDIT_UNDO, //ID Para cambio a obra anterior
    ID_SEPARATOR,
    ID_EDIT_CUT,
    ID_EDIT_COPY,
    ID_EDIT_PASTE,
    ID_SEPARATOR,
};
```

Estos ids son posiciones a una tabla de cadenas que son los Bar-Tips que se presionan en cada uno de los botones del Toolbar.

```

ID_UTILERIAS BORRAOBRA,          //ID de borrar una obra
    ID_SEPARATOR,
ID_UTILERIAS XIE,                //ID de exportación de una obra
    ID_SEPARATOR,
ID_UTILERIAS CONFIGSAICIC,      //ID de configuración de una obra
    ID_SEPARATOR,               //Separador
ID_SAICIC_TIPS,                 //Tips de SAICIC para Windows
ID_CONTEXT_HELP                 //Caja de Diálogo de ayuda por Contexto
static int BASED_CODE buttonsTextID[] =

// Colocamos los indices a los que hacen referencia los elementos
// dentro del toolbar
1,
2,
0,
0,
0,
3,
4,
0,
5,
0,
6,
7,
8,
0,
11,
0,
10,
0,
12,
0,
13,
14
;

// COLOCAMOS LA INFORMACION SOBRE LOS BARTIPS DE SAICIC WINDOWS
static LPCSTR BASED_CODE szBarTips [] =

"", //0
"Crear Obra", //1
"Abrir Obra", //2
"Cerrar Obra", //3
"Panel de Control", //4
"Desahacer", //5
"Cortar", //6
"Copiar", //7
"Pegar", //8
"Copiar Obra", //9
"Operación con Obras", //10
"Borrar Obra", //11
"Configurar SAICIC", //12
"SAICIC Tips", //13
"Ayuda Sensitiva" //14
;

static UINT BASED_CODE indicators[] =

ID_SEPARATOR, // status line indicator
ID_INDICATOR_CLAVE, //Búsqueda por clave o descripción
ID_INDICATOR_CAPS,
ID_INDICATOR_NUM,
;

////////////////////////////////////
/ MainForm construction/destruction

MainForm::MainForm()

    m :Banderas = 0;

endfunction

```

Capítulo IV Desarrollo del Sistema

```
CMainFrame::CMainFrame()
{
}

BOOL CMainFrame::PreCreateWindow ( CREATESTRUCT & crs )
{
    CString strPosicion = theApp.GetProfileString ( ID_SECC_PRESENTA,
                                                    ID_CAMPO_POSICION,
                                                    "" );

    LPCSTR lpszNum;
    int nCont = 0;
    int *pNumero;

    if ( !strPosicion.IsEmpty () )
    {
        // Si tiene valores
        lpszNum = strPosicion;
        pNumero = &crs.cy;

        while ( *lpszNum )
        {
            if ( *lpszNum != ' ' )
            {
                szClave [ nCont ] = *lpszNum;
                nCont++;
            }
            else
            {
                szClave [ nCont ] = '\0';
                *pNumero = StringToInt ( szClave );
                pNumero++;
                nCont = 0;
            }
            endiff

            lpszNum++;
        }
        while

        szClave [ nCont ] = '\0';
        *pNumero = StringToInt ( szClave );

    }endif

    return CMDIFrameWnd::PreCreateWindow ( crs );
}endifunction

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CMDIFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    int iTamTool;

    //Ya tenemos el usuario en momento de Regi.
    theApp.m_dlgBanner = new CBanner;
    theApp.m_dlgBanner->ShowWindow ( SW_SHOW );
    theApp.m_dlgBanner->UpdateWindow();

    CString strEstiloBarra;
    DWORD dwEstilo;

    dwEstilo = WS_CHILD | CBS_BOTTOM;

    // Ahora tenemos que saber que pasa con las barras
```

Una vez creada la ventana principal pasamos a la inicialización de nuestro ambiente. Creamos Toolbars, estamos Bar, registramos también nuestro banner de Proclamación.

```

strEstiloBarra = theApp.GetProfileString ( ID_SECC_PRESENTA,
                                           ID_CAMPO_BARRA_STATUS,
                                           "1" );

if ( '1' == strEstiloBarra[0] )
{
    dwEstilo |= WS_VISIBLE;
}
endif

if (!m_wndStatusBar.Create( this, dwEstilo )
    ||
    !m_wndStatusBar.SetIndicators(Indicators,
                                  sizeof(Indicators)/sizeof(UINT)))
{
    TRACE("Failed to create status bar\n");
    return -1; // fail to create
}
endif

theApp.m_pPanelCentral = new CPreciosDlg ( this );

iTamTool = sizeof(buttons)/sizeof(UINT);

dwEstilo = WS_CHILD;
// Ahora tenemos que saber que pasa con las barras
strEstiloBarra = theApp.GetProfileString ( ID_SECC_PRESENTA,
                                           ID_CAMPO_BARRA_GENERAL,
                                           "11" ); // VISIBLE Y ARRIBA

if ( '1' == strEstiloBarra[0] )
{
    dwEstilo |= WS_VISIBLE;
}
endif

if ( '1' == strEstiloBarra[1] )
{
    dwEstilo |= CBRS_TOP;
}
else
{
    dwEstilo |= CBRS_BOTTOM;
}
endif

if (!m_wndToolBar.Create( this, "General", dwEstilo ) ||
    !m_wndToolBar.LoadBitmap(IDK_MAINFRAME) ||
    !m_wndToolBar.SetButtons(buttons,
                             iTamTool,
                             7,
                             s_ParTips,
                             buttonsTextID ) )
{
    TRACE("Failed to create toolbar\n");
    return -1; // fail to create
}

dwEstilo = WS_CHILD;
// Ahora tenemos que saber que pasa con las barras
strEstiloBarra = theApp.GetProfileString ( ID_SECC_PRESENTA,
                                           ID_CAMPO_BARRA OPCIONES,
                                           "11" ); // VISIBLE Y ARRIBA

if ( '1' == strEstiloBarra[0] )
{
    dwEstilo |= WS_VISIBLE;
}
endif

```

Capítulo IV Desarrollo del Sistema

```
if ( '1' == strEstiloBarra(1) )
{
    dwEstilo |= CBRS_TOP;
}
else
{
    dwEstilo |= CBRS_BOTTOM;
}
endif

if ( !m_wndToolBarPart.Create( this, "Opciones",
                               dwEstilo,
                               ID_TOOL_PARTICULAR)
    ||
    !mOpcionesPrecios ( ) )
{
    TRACE("No crea el toolbar secundario\n");
    return -1;    // fail to create
}
endif

dwEstilo = WS_CHILD;
// Ahora tenemos que saber que pasa con las barras
strEstiloBarra = theApp.GetProfileString ( ID_SECC_PRESENTA,
                                           ID_CAMPO_BARRA_EDICION,
                                           "10" ); // VISIBLE Y ARRIBA

if ( '1' == strEstiloBarra(0) )
{
    dwEstilo |= WS_VISIBLE;
}
endif

if ( '1' == strEstiloBarra(1) )
{
    dwEstilo |= CBRS_TOP;
}
else
{
    dwEstilo |= CBRS_BOTTOM;
}
endif

if ( !m_wndToolBarCap.Create( this, "Edición",
                              dwEstilo,
                              ID_TOOL_CAPTURA )
    ||
    !m_wndToolBarCap.LoadBitmap( IDR_CAPTURA)
    ||
    !m_wndToolBarCap.SetButtons( captura,
                                sizeof( captura ) / sizeof( UINT ),
                                12,
                                szBarTipsCap,
                                buttonsTextCapID ) )
{
    TRACE("Failed to create toolbar\n");
    return -1;    // fail to create
}

iItemTool = sizeof( buttons ) / sizeof( UINT );

m_wndToolBar.OnToolPrincipalCambio ( iItemTool, -1 );

return 0;
}

//////////////////////////////////////
// WinFrame diagnostics
```

```

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CMDIFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CMDIFrameWnd::Dump(dc);
}

#endif // _DEBUG

void CMainFrame::OnSysColorChange()
{
    // CMDIFrameWnd::OnSysColorChange();

    //Cambio de Colores del Sistema windows
    CUI::ColorChange();
}

void CMainFrame::OnClose()
{
    if ( theApp.fnGetBool ( REP_ACTIVADO ) )
    {
        // No se puede salir de la aplicación hasta
        // estudio
        DD::Msg::Error ( m_hWnd,
            theApp.m_hInstance,
            theApp.m_szUsuario,
            IDST_ERR_REP_SAI );

        return;
    }

    // Ahora tenemos que saber que pasa con las barras
    CString strEstiloBarra;

    // Salva info de barras de estatus
    strEstiloBarra = (m_wndStatusBar.GetStyle () & WS_VISIBLE ) ?
        '1' : '0';
    theApp.WriteProfileString ( ID_SECC_PRESENTA,
        ID_CAMPO_BARRA_STATUS,
        strEstiloBarra );

    // Salva info de barra General
    strEstiloBarra = (m_wndToolBar.GetStyle () & WS_VISIBLE ) ?
        '1' : '0';
    strEstiloBarra += (m_wndToolBar.GetStyle () & CBRS_TOP ) ?
        '1' : '0';
    theApp.WriteProfileString ( ID_SECC_PRESENTA,
        ID_CAMPO_BARRA_GENERAL,
        strEstiloBarra );

    // Salva info de barra de Opciones
    strEstiloBarra = (m_wndToolBarPart.GetStyle () & WS_VISIBLE ) ?
        '1' : '0';
    strEstiloBarra += (m_wndToolBarPart.GetStyle () & CBRS_TOP ) ?
        '1' : '0';
    theApp.WriteProfileString ( ID_SECC_PRESENTA,
        ID_CAMPO_BARRA OPCIONES,
        strEstiloBarra );

    // Salva info de barra de Captura
    strEstiloBarra = (m_wndToolBarCap.GetStyle () & WS_VISIBLE ) ?
        '1' : '0';
    strEstiloBarra += (m_wndToolBarCap.GetStyle () & CBRS_TOP ) ?
        '1' : '0';
}

```

Al cerrar la ventana principal es necesario guardar información referente al ambiente, para que en una nueva sesión se puedan restablecer todas las obras y sus opciones que el usuario tenía abiertas

```

theApp.WriteProfileString ( ID_SECC_PRESENTA,
                           ID_CAMPO_BARRA_EDICION,
                           strEstiloBarra );

// Ahora obten las dimensiones de la ventana
CRect rect;

if ( pwndCatClaves )
    pwndCatClaves->DestroyWindow ();
endif

GetWindowRect ( rect );

if ( IsZoomed () )
    (
        theApp.WriteProfileInt ( ID_SECC_PRESENTA,
                                ID_CAMPO_MAXIMO,
                                1 );
    )
else
    (
        theApp.WriteProfileInt ( ID_SECC_PRESENTA,
                                ID_CAMPO_MAXIMO,
                                0 );
    )
endif

KillTimer(m_uiTimer);

theApp.m_pPanelCentral->CierraObras();

LPMDIVentStruct lpMDIInfo;
POSITION pos;

for ( pos = theApp.m_pPanelCentral->m_pInfoObras.GetTailPosition();
      pos != NULL;
      )
    (
        lpMDIInfo = (LPMDIVentStruct)
                    theApp.m_pPanelCentral->m_pInfoObras.GetPrev (pos);
        ASSERT ( lpMDIInfo );

        delete lpMDIInfo;
    )
endifor

theApp.m_pPanelCentral->m_pInfoObras.RemoveAll();

CMDIFrameWnd::OnClose();

delete (CPreciosDlg *) theApp.m_pPanelCentral;

// Es momento de ver que información tenemos que salvar,
// posición de la ventana principal y su estado
wsprintf ( szClave, "ti ti ti ti",
          rect.bottom,
          rect.right,
          rect.top,
          rect.left );

theApp.WriteProfileString ( ID_SECC_PRESENTA,
                           ID_CAMPO_POSICION,
                           szClave );

}

}

void CMainFrame::OnCerrarObras()

```



```

{
    theApp.m_pPanelCentral->fnCierzoObra ();
}

void CMainFrame::OnUpdateEditUndo(CCmdUI* pCmdUI)
{
    pCmdUI->Enable ( FALSE );
}

void CMainFrame::WinHelp(DWORD dwData, UINT nCmd /* = HELP_CONTEXT */)
{
    CMDIGenSAI * pMDIActivo;
    ASSERT_VALID(AfxGetApp());
    ASSERT(AfxGetApp()->m_pszHelpFilePath != NULL);
    BeginWaitCursor();
    if (IsFrameWnd())
    {
        // CFrameWnd windows should be allowed to exit help mode first
        CFrameWnd* pFrameWnd = (CFrameWnd*)this;
        pFrameWnd->ExitHelpMode();
    }

    // cancel any tracking modes
    SendMessage(WM_CANCELMODE);
    SendMessageToDescendants(WM_CANCELMODE, 0, 0, TRUE, TRUE);

    // need to use top level parent (for the case where m_hWnd is in DLL)
    CWnd* pWnd = GetTopLevelParent();
    pWnd->SendMessage(WM_CANCELMODE);
    pWnd->SendMessageToDescendants(WM_CANCELMODE, 0, 0, TRUE, TRUE);

    // attempt to cancel capture
    HWND hWndCapture = ::GetCapture();

    if (hWndCapture != NULL)
        ::SendMessage(hWndCapture, WM_CANCELMODE, 0, 0);

    TRACE3("WinHelp: pszHelpFile = '%s', dwData: $%lx, fuCommand: %d\n",
        AfxGetApp()->m_pszHelpFilePath, dwData, nCmd);

    pMDIActivo = ( CMDIGenSAI * ) MDIGetActive ();

    // Bueno chequea si tiene el focus
    if ( pMDIActivo )
    {
        switch ( dwData - HTD_BASE_RESOURCE )
        {
            case IDR_MATERIALES:
            case IDR_ANALISIS:
            case IDR_FSR:
                // SI hay MDI activo, mandale el context help
                pMDIActivo->WinHelp ( dwData );
        }
    }

}

if ( HTD_BASE_NCAREAS <= dwData
    && HTD_BASE_CONTROL - 1 >= dwData )
{

```

Capítulo IV Desarrollo del Sistema

```
nCmd = HELP_INDEX;
}
endif

if (!::WinHelp(pWnd->m_hWnd, AfxGetApp()->m_pszHelpFilePath, nCmd, dwData))
    AfxMessageBox(AFX_IDP_FAILED_TO_LAUNCH_HELP);
EndWaitCursor();
}

int CToolBarCombo::OnToolPrincipalCambio ( int nCount, int nEleSel )
{
    CRect rect;
    CString strPathName = "";
    MINIVentStruct lpMiniInfo;
    CMainFrame *pMainFrame = ( CMainFrame *) theApp.m_pMainWnd;
    CSalcicApp *pSalcicApp = ( CSalcicApp* ) AfxGetApp ();
    CPtrList *pInfoObras;

    pInfoObras = pSalcicApp->m_pPanelCentral->m_pInfoObras;

    SetButtonInfo ( 3,
        IDC_COMBO_ACTIVAS,
        TBBS_SEPARATOR, 200 );

    GetItemRect ( 3, &rect);
    rect.top = 3;
    rect.bottom = rect.top + 100;

    if ( !m_Obras.Create( CBS_DROPDOWNLIST | WS_VISIBLE | WS_TABSTOP,
        rect,
        this,
        IDC_COMBO_ACTIVAS ) )
    {
        TRACE("Fallo la creacion del combobox de obras\n");
        return 1;
    }

    endif

    // Creamos el font del combobox
    LOGFONT logFont;
    Zero ( logFont );

    // Since design guide says toolbars are fixed height so is the font.
    logFont.lfHeight = -12;
    logFont.lfWeight = FW_BOLD;
    logFont.lfPitchAndFamily = VARIABLE_PITCH | FF_SWISS;
    strcpy(logFont.lfFaceName, "MS Sans Serif");

    if ( !m_Font.GetSafeHandle () )
    {
        if (!m_Font.CreateFontIndirect(&logFont))
            TRACE("No puede crear el font del combobox de obras\n");
        endif

        m_Obras.SetFont( &m_Font );
    }
    else
    {
        m_Obras.SetFont( &m_Font );
    }
    endif

    CToolBarSubclassCtl ( m_Obras,m_hWnd );

    if ( -1 != nEleSel )
    {

```

Capítulo IV Desarrollo del Sistema

```
        nCmd = HELP_INDEX;
    }
    }

    if (!::WinHelp(pWnd->hWnd, AfxGetApp()->m_pszHelpFilePath, nCmd, dwData))
        AfxMessageBox(AFX_IDP_FAILED_TO_LAUNCH_HELP);
    EndWaitCursor();
}

int CToolBarCombo::OnToolPrincipalCambio ( int nCount, int nEleSel )
{
    CRect      rect;
    CString   strPathName = "";
    IMenuVer::Struct  lpMenuInfo;
    CMainFrame *pMainFrame = ( CMainFrame * ) theApp.m_pMainWnd;
    CSaicicApp *pSaicicApp = ( CSaicicApp * ) AfxGetApp ();
    CToolBarList *pInfoObras;

    pInfoObras = pSaicicApp->m_pPanelCentral->m_pInfoObras;

    SetButtonInfo ( 3,
                   IDC_COMBO_ACTIVAS,
                   TBBS_SEPARATOR, 200 );

    GetItemRect ( 3, &rect);
    rect.top = 3;
    rect.bottom = rect.top + 100;

    if ( !m_Obras.Create( CBS_DROPDOWNLIST | WS_VISIBLE | WS_TABSTOP,
                        rect,
                        this,
                        IDC_COMBO_ACTIVAS ) )
    {
        TRACE("Fallo la creacion del combobox de obras\n");
        return 1;
    }

    // Creamos el font del combobox
    LOGFONT logFont;
    Zero ( logFont );

    // Since design guide says toolbars are fixed height so is the font.
    logFont.lfHeight = -12;
    logFont.lfWeight = FW_BOLD;
    logFont.lfPitchAndFamily = VARIABLE_PITCH | FF_SWISS;
    strcpy(logFont.lfFaceName, "MS Sans Serif");

    if ( !m_Font.GetSafeHandle () )
    {
        if ( !m_Font.CreateFontIndirect (&logFont) )
            TRACE("No puede crear el font del combobox de obras\n");
        endif

        m_Obras.SetFont ( &m_Font );
    }
    else
    {
        m_Obras.SetFont ( &m_Font );
    }
    endif

    Ctl3dSubclassCtl ( m_Obras.m_hWnd );

    if ( -1 != nEleSel )
    {

```

```

// Si hay elemento seleccionado agrega los elementos seleccionados
// agréjalos al combobox
for ( POSITION pos = pInfoObras->GetHeadPosition(); pos != NULL; ) {

    lpMDIInfo = (LPMDIventStruct) pInfoObras->GetNext (pos); //Obtenemos Información de las
    //borramos el elemento
    ASSERT ( lpMDIInfo ); //Checar el Valor

    // OK obten el drive, directorio y colocalos en el combobox
    strPathName = lpMDIInfo->szDrive;
    strPathName += lpMDIInfo->szDir;
    strPathName += '\\';
    strPathName += lpMDIInfo->szNomDiccData;
    strPathName += EXT_CON_PUNTO_DHD;
    m_Obras.AddString ( strPathName );

}

m_Obras.SetCurSel ( nEleSel );

}

return TRUE;

}

LRESULT CToolBarCombo::WindowProc(UINT nMsg, WPARAM wParam, LPARAM lParam)
{
    ASSERT_VALID(this);
    switch (nMsg)
    {
        case WM_COMMAND:
            switch ( wParam )
            {
                case ID_ARRIBA:
                    if ( m_bEsPopup )
                    {
                        CWnd* pParent = GetOwner ();
                        HBITMAP hBitmapAux = m_hbmImageWell;
                        INT nEleSel;

                        m_hbmImageWell = NULL;
                        nEleSel = m_Obras.GetCurSel ();

                        // Tenemos que destruir a la ventana y volverla una
                        // ventana normal
                        if ( !DestroyWindow ( ) )
                        {
                            return FALSE;
                        }
                    }
                    m_bEsPopup = FALSE;
                    if ( CToolBar::Create ( m_pOwnerWnd,
                        WS_CHILD | WS_VISIBLE | CBS_TOP,
                        m_nID ) )
                    {
                        m_hbmImageWell = hBitmapAux;
                    }
            }
    }
}

```

Capítulo IV Desarrollo del Sistema

```
    }
    else
    {
        SetWindowLong ( m_hWnd, GWL_STYLE,
            WS_CHILD | WS_VISIBLE | CURS_TOP );
    }
endfunction

{ (CFrameWnd *) GetOwner() }->RecalcLayout ();
m_ArribaAbajoFlotante = BOOL_ARRIBA;
Invalidate ();
return FALSE;

case ID_ABAJO:

if ( m_bEsPopup )
{
    CWnd* pParent = GetOwner ();
    HBITMAP hBitmapAux = m_hbmImageWell;
    INT nEleSel;

    m_hbmImageWell = NULL;
    nEleSel = m_Obras.GetCurSel ();

    // Tenemos que destruir a la ventana y volverla una
    // ventana normal
    if ( !DestroyWindow () )
    {
        return FALSE;
    }
}endif

m_bEsPopup = FALSE;

if ( CToolBar::Create ( m_OwnerWnd,
    WS_CHILD | WS_VISIBLE | CURS_BOTTOM,
    m_ID ) )
{
    m_hbmImageWell = hBitmapAux;
}endif

}
else
{
    SetWindowLong ( m_hWnd, GWL_STYLE,
        WS_CHILD | WS_VISIBLE | CURS_BOTTOM );
}

endfunction

{ (CFrameWnd *) GetOwner() }->RecalcLayout ();
m_ArribaAbajoFlotante = BOOL_ABAJO;
Invalidate ();
return FALSE;

case ID_FLOTANTE:

if ( !m_bEsPopup )
{
    CString strTexto;
    CWnd* pParent;
    LPCTSTR lpszUbica;
    CString strPos;
    INT nEleSel;

```

```

HBITMAP hBitmapAux = m_hbmImageWell;
UINT nID = GetDlgCtrlID ();

//m_hbmImageWell = NULL;

GetWindowText ( strTexto );
CString strUbica = AfxGetApp ()->GetProfileString (
    "Toolbars",
    strTexto,
    "1,20,30,0,0" );

CRect rect;

pParent = GetOwner ();
nEleSel = m_Obras.GetCurSel ();

if ( DestroyWindow () )
{
    // Dentro de strUbica se encuentran las coordenadas de posición
    // inicial, posición superior, ancho y altura.
    lpszUbica = strUbica;
    m_hbmImageWell = hBitmapAux;

    fnObtenToken ( strPos, lpszUbica ); // Necesario
    // Se trata de un Toolbar popup, seguimos con la extracción de
    // dimensiones
    // Tenemos que colocar las acciones de dicha
    m_bEsPopup = TRUE;

    // Procedemos a obtener la posición en x del Toolbar
    fnObtenToken ( strPos, lpszUbica );
    rect.left = StringToInt ( (LPGTR) (LPCSTR) strPos );
    // Procedemos a obtener la posición en y del Toolbar
    fnObtenToken ( strPos, lpszUbica );
    rect.top = StringToInt ( (LPSTR) (LPCSTR) strPos );
    m_ArribaAbajoFlotante = BOOL_FLOTANTE;

    // OK ya tenemos las posiciones, Mandamos crear el Toolbar
    if ( CreateEx(0,
        "AfxControlBar",
        strTexto,
        WS_POPUP |
        WS_VISIBLE,
        rect.left,
        rect.top,
        0,
        0,
        pParent->GetSafeHwnd(),
        0,
        NULL) )
    {
        SetButtons(buttons,
            m_nCount,
            7,
            szBarTips,
            buttonsTextID );
        m_hbmImageWell = hBitmapAux;
        ((CFrameWnd *) m_pOwnerWnd )->RecalcLayout ();
    }
}
endif // If CreateWindow
}
else
{
    m_hbmImageWell = hBitmapAux;
}
endif //if DestroyWindow
endif

m_ArribaAbajoFlotante = BOOL_FLOTANTE;

```

Capítulo IV Desarrollo del Sistema

```
( (CFrameWnd *) GetOwner() )->RecalcLayout ();
return FALSE;

case ID_DESAPARECE:
ShowWindow ( ( IWindowVisible () ) ? SW_HIDE : SW_SHOWNORMAL );
( (CFrameWnd *) GetOwner() )->RecalcLayout ();
Invalidate ();
return FALSE;

}endswitch

}endswitch

return CToolBarSAI::WindowProc(nMsg, wParam, lParam);
}endfunction

void CMainFrame::OnActivateApp(BOOL bActive, HTASK hTask)
{
    if ( !bActive )
    {
        // Oculta al famoso tip
        fnDesapareceTip ();
        bJcl.Editar = FALSE;
    }
    else
    {
        bJcl.NoEditar = TRUE;
    }
}endif

CMainFrame::OnActivateApp(bActive, hTask);
}endfunction

BOOL CMainFrame::OnQueryEndSession()
{
    if ( !CMainFrame::OnQueryEndSession() )
        return FALSE;

    OnClose ();
    return TRUE;
}

void CMainFrame::OnVerOpciones()
{
    m_wndToolBarPart.ShowWindow (
        m_wndToolBarPart.GetStyle() & WS_VISIBLE ) == 0 ?
        SW_SHOW : SW_HIDE );

    if ( !m_wndToolBarPart.m_bEsPopup )
    {
        RecalcLayout();
    }
}

void CMainFrame::OnUpdateVerOpciones(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck( ( m_wndToolBarPart.GetStyle() & WS_VISIBLE ) != 0 );
}
}
```

Punto muy fino, ya que cuando Windows termina su sesión es necesario también guardar nuestras opciones para restablecerlas posteriormente.

```

id CMainFrame::OnVerEdicion()

    m_wndToolBarCap.ShowWindow (
        ( m_wndToolBarCap.GetStyle () & WS_VISIBLE ) == 0 ?
        SW_SHOW : SW_HIDE );

    if ( !m_wndToolBarCap.m_bEsPopup )
    {
        RepalcLayout();
    }
}

id CMainFrame::OnUpdateVerEdicion(CCmdUI* pCmdUI)

    pCmdUI->SetCheck( ( m_wndToolBarCap.GetStyle() & WS_VISIBLE) != 0 );

id CMainFrame::OnAyudaError()

    // TODO: Add your command handler code here:
    CString strPath = theApp.m_pszHelpFilePath;
    LPSTR lpszCambiaPath;

    // Ya tenemos el path
    lpszCambiaPath = strPath.GetBuffer ( LONG_DIR );

    if ( lpszCambiaPath = StrChr ( lpszCambiaPath, '\\' ) )
    {
        // Bueno pues cambia el path
        lpszCambiaPath++;
        lstrcpy ( lpszCambiaPath, ERRORES );
    }

    strPath.ReleaseBuffer ();

    ::WinHelp ( m_hWnd, strPath, HELP_CONTENTS, 0L );
}

id CMainFrame::OnAyudaSoporte()

    // TODO: Add your command handler code here
    CString strPath = theApp.m_pszHelpFilePath;
    LPSTR lpszCambiaPath;

    // Ya tenemos el path
    lpszCambiaPath = strPath.GetBuffer ( LONG_DIR );

    if ( lpszCambiaPath = StrChr ( lpszCambiaPath, '\\' ) )
    {
        // Bueno pues cambia el path
        lpszCambiaPath++;
        lstrcpy ( lpszCambiaPath, ARCH_SOPORTE );
    }

    strPath.ReleaseBuffer ();

    ::WinHelp ( m_hWnd, strPath, HELP_CONTENTS, 0L );
}

RESULT CMainFrame::OnRupNotifica ( WPARAM, LPARAM )

```


Capítulo IV Desarrollo del Sistema

```
if ( !theApp.OnIdle ( 0 ) )
{
    theApp.PumpMessage ();
}

return TRUE;
}

int * pIDsText = NULL;

// COLOCAMOS LA INFORMACION SOBRE LOS BARTIPS DE SAICIC WINDOWS
static LPCSTR BASED_CODE scharTips [] =
{
    "", //0
    "Materiales", //1
    "Obra de Mano", //2
    "Maquinaria y Equipo", //3
    "Subcontratos", //4
    "Básicos", //5
    "Integrados", //6
    "Partidas", //7
    "Capítulos", //8
    "Extraordinarios", //9
    "E.S.R.", //10
    "Actualizar Precios", //11
    "Formatos Sobre Costo", //12
    "Explosión de Insumos", //13
    "Familias", //14
    "Zonas", //15
    "Moneda", //16
    "Herramienta" //17
    //"Estimaciones" //18
};

int CMainFrame::InOpcionesPrecios ()
{
    UINT iOpPrecios [] =
    {
        ID_MATERIALES, //Opción de Materiales
        ID_OBRA_DE_MANO, //Opción de Obra de Mano
        ID_MAQUINA_EQUIPO, //Opción de Maquinaria y Equipo
        ID_SUBCONTRATOS, //Opción de Subcontratos
        ID_SEPARATOR, //Separador
        ID_HERRAMIENTA,
        ID_SEPARATOR, //Separador
        ID_BÁSICOS, //Opciones de Básicos
        ID_INTEGRADOS, //Opciones de Integrados
        ID_PARTIDAS, //Opciones de Partidas
        ID_CAPITULOS, //Opciones de Capítulos
        ID_EXTRAORDINARIOS, //Opciones de Extraordinarios
        ID_SEPARATOR,
        ID_ESR,
        ID_SEPARATOR,
        ID_ACTUALIZA_PRECIOS, //Información de presupuestos
        ID_FACTOR SOBRE_COSTO, //Factores de Sobre Costo
        ID_EXPLOSION,
        ID_SEPARATOR,
        ID_FAMILIAS, //Opciones de actualización de Precios
        ID_ZONAS, //Caratula de presupuestos
        ID_CAY_MONEDA //CATALOGO DE MONEDA
        // ID_SEPARATOR,
        //ID_ESTIMACIONES
    };

    static int pIDsText [] = { 1, 2, 3, 4, 0,
        17, 0,
        5, 6, 7, 8, 9, 0,
        14, 0,
        11, 12, 13, 0,
        14, 15, 16,
        17
    };
};
```

```

if ( !m_wndToolBarPart.LoadBitmap( IDE_TODAS OPCIONES ) ||
    !m_wndToolBarPart.SetButtons( iOpPrecios,
                                sizeof( iOpPrecios ) / sizeof ( UINT ),
                                12,
                                szBarTips,
                                pIDsText ) )
{
    TRACE("No se logran cargar las imagenes del Toolbar");
    return 0; // fail to create
}
endif
return 1;
}endifunction

```

```

/*void CMainFrame::fnObtenIndicesToolBar ( )
{

```

```

    int iIndiceBitmap = 0,
        iIndiceArreglo = 0;

    if ( (RAN_POR_DESC & m_lBanderas ) { (
        switch ( HIBYTE ( m_lBanderas ) ) {
            case FORMA_OP_PRECIOS:
                iIndiceBitmap = INDEX_IMAG_PRECIOS;
                iIndiceArreglo = INDEX_ARRAY_PRECIOS;
                break;

            case FORMA_OP_INSUMOS:
                iIndiceBitmap = INDEX_IMAG_INSUMOS;
                iIndiceArreglo = INDEX_ARRAY_INSUMOS;
                break;

            case FORMA_OP_ANALISIS:
                iIndiceBitmap = INDEX_IMAG_ANALISIS;
                iIndiceArreglo = INDEX_ARRAY_ANALISIS;
                break;

            case FORMA_OP_PRESUPUE:
                iIndiceBitmap = INDEX_IMAG_PRESUPUE;
                iIndiceArreglo = INDEX_ARRAY_PRESUPUE;
                break;

            case FORMA_OP_RELACION:
                iIndiceBitmap = INDEX_IMAG_RELACION;
                iIndiceArreglo = INDEX_ARRAY_RELACION;
                break;
        }endswitch
    )
    else {
        switch ( HIBYTE ( m_lBanderas ) ) {
            case FORMA_OP_PRECIOS:
                iIndiceBitmap = INDEX_IMAG_ORIGEN_PRECIOS;
                iIndiceArreglo = INDEX_ARRAY_PRECIOS;
                break;

            case FORMA_OP_INSUMOS:
                iIndiceBitmap = INDEX_IMAG_ORIGEN_INSUMOS;
                iIndiceArreglo = INDEX_ARRAY_INSUMOS;
                break;

            case FORMA_OP_ANALISIS:
                iIndiceBitmap = INDEX_IMAG_ORIGEN_ANALISIS;
                iIndiceArreglo = INDEX_ARRAY_ANALISIS;

```

Capítulo IV Desarrollo del Sistema

```
        break;

    case FORMA_OP_PRESUPUE:
        iIndiceBitmap = INDEX_IMAG_ORIGEN_PRESUPUE;
        iIndiceArreglo = INDEX_ARRAY_PRESUPUE;
        break;

    case FORMA_OP_RELACION:
        iIndiceBitmap = INDEX_IMAG_ORIGEN_RELACION;
        iIndiceArreglo = INDEX_ARRAY_RELACION;
        break;

    }endswitch

}endif

m_wndToolBarPart.SetButtonInfo( iIndiceArreglo,
                                ID_PRESENTA_DATA,
                                TBBS_BUTTON, iIndiceBitmap );

}endifunction
/**
void CMainFrame::OnUpdateVenPanel(CCmdUI* pCmdUI)
{
    pCmdUI->Enable( (BOOL) !theApp.m_pPanelCentral->m_pInfoObras.GetCount() );
}

void CMainFrame::OnFileExit()
{
    // TODO: Add your command handler code here
    OnClose();
}

void CMainFrame::OnUpdateMenus(CCmdUI* pCmdUI)
{
    pCmdUI->Enable( TRUE );
}endifunction

void CMainFrame::OnUtileriasExporta()
{
    CString strComando = "";
    strComando = DDfnTraeStringLib( AfxGetInstanceHandle(), IDS_COMMAND_EXPORTAR );

    BeginWaitCursor ();
    WinExec ( strComando, SW_SHOW );
    EndWaitCursor ();
}

void CMainFrame::OnCerrarBanner()
{
    delete theApp.m_dlgBanner;
}endifunction
```



```

// FUNCION: OnAbreObrasRecientes ()
// DESCRIPCION: Esta función se encarga de abrir todas las obras que
// han sido utilizadas en sesiones anteriores de SAICIC para
// Windows, con mayor precisión en la última sesión de SAICIC
// Se encarga de abrir la obra, una vez que la ha abierto
// intenta crear las ventanas que el usuario estaba utilizando
// antes de salir de la obra
//
LRESULT CMainFrame::OnAbreObrasRecientes ( WPARAM, LPARAM )
{
    TRY
    {
        BOOL        bContinua = TRUE,
                   bContinuaClave,
                   bAnalisis;
        CString     strDirectorio;
        int         nCont = 1,
                   nClave;
        LPMDIVentStruct lpInfoObra;
        CMDIAnalysis *pMDIAnalysis;
        CFormaGral *pForma;
        (int nTipoBusqueda = BUSC_APROX,
         nSubTipoBusqueda = 0;
        POSITION      posMDI;
        CInfoMDI *pInfoDoc;

        // Tenemos que verificar si existe el nombre del usuario en el
        // archivo de inicialización
        /*fnVerificarUser ( theApp.m_szUsuario, IDD_USUARIO );
        fnIniLib ( theApp.m_szUsuario, theApp.m_pszHelpFilePath, ERRORES );*/

        do
        {
            // Intentamos saber si existe información sobre una obra reciente
            wprintf ( szClave, "%s", ID_CAMPO_OBRAS, nCont );
            strDirectorio = theApp.GetProfileString (
                           ID_SECCION_OBRAS_ABIERTAS,
                           szClave,
                           "" );

            if ( strDirectorio.IsEmpty () )
            {
                bContinua = FALSE;
            }
            else
            {
                // Bueno es momento de abrir la obra,
                lpInfoObra = (LPMDIVentStruct) new MDIVentStruct;

                // Se acepta la creación de una obra
                lstrcpy ( lpInfoObra->szDir, (LPCSTR) strDirectorio + 2 );
                lpInfoObra->szDrive[0] = strDirectorio[0];
                lpInfoObra->szDrive[1] = ':';
                lpInfoObra->szDrive[2] = '\\';

                if ( DDfnCamDrvDir( m_hWnd,
                                   HLibFunc,
                                   lpInfoObra->szDrive[0],
                                   lpInfoObra->szDir,
                                   TRUE ) )
                {
                    // Ya se cambio de directorio, ahora intentamos abrir la obra
                    lstrcpy ( lpInfoObra->szNomDicoData,
                             theApp.m_pszExeName );

                    if ( theApp.m_pPanelCentral->fnBuscaObra (
                               m_hWnd,
                               lpInfoObra,

```

Capítulo IV Desarrollo del Sistema

```
IDS_ERR_SAL_OBRA_YA_ABIERTA | )
{
    if ( !fnAbreObra ( lpInfoObra ) )
    {
        // Excelente, ahora abre las ventanas que sean
        // necesarias
        nClave = 1;
        bContinuaClave = TRUE;

        do
        {
            wsprintf ( szClave,
                "%s%1Clave%",
                ID_CAMPO_OBRAS,
                nCont,
                nClave );

            strDirectorio = theApp.GetProfileString (
                ID_SECCION_OBRAS_ABIERTAS,
                szClave,
                "" );

            if ( strDirectorio.IsEmpty () )
            {
                bContinuaClave = FALSE;
            }
            else
            {
                // Ahora por favor averigua quien es la clave que
                // tenemos
                if ( strlen ( strDirectorio ) > TAM_TIPO )
                {
                    theApp.m_strClaveMostrar =
                        strDirectorio + TAM_TIPO;
                }
                else
                {
                    theApp.m_strClaveMostrar = "";
                }
            }
        } while ( bContinuaClave );

        theApp.m_pPanelCentral->
            fnOtraObraSeleccionada ();

        switch ( strDirectorio[0] )
        {
            case TIPO_BASICO:
                theApp.m_TipoForma = ANALISIS_BASICOS;
                theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_ANALISIS,
                    theApp.m_lpMDIGen );
                // Los analisis siempre tienen dos claves
                bAnalisis = TRUE;
                break;

            case TIPO_CAPITULO:
                theApp.m_TipoForma = ANALISIS_CAPITULOS;
                theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_ANALISIS,
                    theApp.m_lpMDIGen );
                bAnalisis = TRUE;
                break;

            case TIPO_ESTIMACION:
                theApp.m_TipoForma = IDD_ESTIMACIONES;
                theApp.m_pPanelCentral->
                    fnAgrRelOpciones ( TEMPLATE_INSUMOS,
```

```

        theApp.m_lpMDIGen );
        bAnálisis = FALSE;
        break;

case TIPO_FAMILIA:
    bAnálisis = FALSE;
    theApp.m_iTipoForma = IDD_FAMILIAS;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones ( TEMPLATE_INSUMOS,
            theApp.m_lpMDIGen );

    break;

case TIPO_HERRAMIENTA:
    bAnálisis = FALSE;
    theApp.m_iTipoForma = IDD_HERRAMIENTA;
    theApp.m_strClaveMostrar =
        (LPCSTR) strDirectorio;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones ( TEMPLATE_INSUMOS,
            theApp.m_lpMDIGen );

    break;

case TIPO_INTEGRADO:
    theApp.m_iTipoForma = ANALISIS_INTEGRADOS;
    theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_ANALISIS,
        theApp.m_lpMDIGen );

    bAnálisis = TRUE;
    break;

case TIPO_MAT:
    bAnálisis = FALSE;
    theApp.m_iTipoForma = IDD_MATERIALES;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones( TEMPLATE_INSUMOS,
            theApp.m_lpMDIGen );

    break;

case TIPO_EXPLOSION:
    theApp.m_iTipoForma = IDD_EXPLOSION;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones ( TEMPLATE_EXPLOSION,
            theApp.m_lpMDIGen );
    theApp.m_strClaveMostrar = "";
    bAnálisis = FALSE;
    break;

case TIPO_OBRAMANO:
    bAnálisis = FALSE;
    theApp.m_iTipoForma = IDD_OBRA_MANO;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones( TEMPLATE_INSUMOS,
            theApp.m_lpMDIGen );

    break;

case TIPO_PARTIDA:
    theApp.m_iTipoForma = ANALISIS_PARTIDAS;
    theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_ANALISIS,
        theApp.m_lpMDIGen );

    bAnálisis = TRUE;
    break;

case TIPO_MAQUINARIA:
    bAnálisis = FALSE;
    theApp.m_iTipoForma = IDD_MAQUINARIA;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones( TEMPLATE_INSUMOS,
            theApp.m_lpMDIGen );

    break;

case TIPO_FSR:
    theApp.m_strClaveMostrar = "";
    theApp.m_iTipoForma = IDD_FSR;
    theApp.m_pPanelCentral->
        fnAgrRelOpciones ( TEMPLATE_FSR,
            theApp.m_lpMDIGen );

```


Capítulo IV Desarrollo del Sistema

```
        bAnálisis = FALSE;
        break;

    case TIPO_SUBCONTRATO:
        bAnálisis = FALSE;
        theApp.m_iTipoForma = IDD_SUBCONTRATO;
        theApp.m_pPanelCentral->
            InAjrRelOpciones( TEMPLATE_INSUMOS,
                theApp.m_lpMDIGen );
        break;

    case TIPO_EXTRA:
        theApp.m_iTipoForma = ANALISIS_EXTRA;
        theApp.m_pPanelCentral->InAjrRelOpciones ( TEMPLATE_ANALISIS,
            theApp.m_lpMDIGen );
        bAnálisis = TRUE;
        break;

    case TIPO_ZONA :
        bAnálisis = FALSE;
        theApp.m_iTipoForma = IDD_ZONAS;
        theApp.m_pPanelCentral->
            InAjrRelOpciones ( TEMPLATE_INSUMOS,
                theApp.m_lpMDIGen );
        break;
}

endswitch

if ( bAnálisis )
{
    // Tenemos que leer otra clave para análisis
    // Un poco complejo pero
    // obten la vista actual de acuerdo al mdi
    // actual
    // Obten la forma

    wsprintf ( szClave,
        "%s%04d",
        ID_CAMPO_OBRAS,
        nCont,
        nClave );

    strDirectorio = theApp.GetProfileString (
        ID_SECCION_OBRAS_ABIERTAS,
        szClave,
        "" );

    if ( !strDirectorio.IsEmpty () )
    {
        // Ahora por favor averigua quien es la clave que
        // tenemos
        pMDIAnálisis =
            ( CMDIAnálisis * ) MDIGetActive ();

        if ( pMDIAnálisis )
        {
            pForma = ( CFormaGraf * )
                pMDIAnálisis->m_wndSplitter.
                    GetPane ( 0, 1 );

            if ( pForma )
            {
                if ( !strlen( strDirectorio ) > TAM_TIPO )
                {
                    pForma->InCargarDatos(
                        (LPCSTR) strDirectorio +
                            TAM_TIPO,
                        nTipoBusqueda,
                        nSubTipoBusqueda );
                }
            }
        }
    }
}
```

```

        }endif
    }endif
}endif

}endif

nClave++;

} while ( bContinuaClave );

// Ahora tenemos que recorrer la lista para que así podamos
// dejar la información en el orden solicitado
theApp.m_pPanelCentral->fnObtenObraSelecc ();
nClave = theApp.m_lpMDIGen->ObjListaVent.GetCount () - 1;

while ( nClave >= 0 )
{
    posMDI = theApp.m_lpMDIGen->ObjListaVent.
        FindIndex ( nClave );
    pInfoDocto = ( CInfoMDI * )
        theApp.m_lpMDIGen->ObjListaVent.
        GetAt ( posMDI );

    nClave--;

}endwhile

}endif

}
else
{
    delete lpInfoObra;
}endif

}
else
{
    delete lpInfoObra;
}endif

}endif

nCont++;

} while ( bContinua );

//MDICascade ();

//Crea un timer para muestrear la llave del sistema.
oConfigura.m_bDemo = FALSE;
m_ulTimer = SetTimer( 1, //id del Timer
    2000, //en milisegundos
    NULL );
//TimerCaratula ); //Procesa el Timer.

}
CATCH ( CMemoryException, e )
{
    DBfnDappError ( m_hWnd,
        HIDEFINDER,
        theApp.m_szUsuario,

```

Capítulo IV Desarrollo del Sistema

```
        IDST_ERR_NO_MEM
    );

END_CATCH

return TRUE;
}

function

void CMainFrame::OnArchivoAbrir()
{
    CAbreObraSai dlgAbreObra ( this,
        IDD_ABRE_OBRA,
        FMT_ARCH_DBD,
        EDInTraeStringLib (
            theApp.m_hInstance,
            IDS_EXT_SAICIC) );

    if ( dlgAbreObra.DoModal() == IDOK ) {
        TRY {
            LPMDIVentStruct lpInfoObra;

            lpInfoObra = (LPMDIVentStruct) new MDIVentStruct;

            // Se acepto la creación de una obra
            lpInfoObra->szDir[0] = '\\';
            DfFr.MoObtenDirActual ( &lpInfoObra->szDir[1] );
            DfFr.MoObtenDriveAct ( lpInfoObra->szDrive );
            lstrcpy ( lpInfoObra->szNomDicData,
                theApp.m_ps2.exeName );

            if ( theApp.m_pPanelCentral->fnBuscaObra (
                m_hWnd,
                lpInfoObra,
                IDS_ERR_SAI_OBRA_YA_ABIERTA
            ) ) {

                fnAbreObra ( lpInfoObra );
            }
            else {
                delete lpInfoObra;
            }
        }
        CATCH (CMemoryException, e) {

            DfFr.DspError ( m_hWnd,
                hLibFunc,
                theApp.m_szUsuario,
                IDST_ERR_NO_MEM
            );
        }
    }

    END_CATCH

    // El usuario acepto la apertura, con éxito
}

function

/*
FUNCION: BOOL fnAbreObra ( LPMDIVentStruct lpInfoObra )
PARAMS: BOOL: LPMDIVentStruct lpInfoObra apuntador al Area de memoria que
contiene la información acerca de la obra.
*/

```

DESCRIPCION: Esta función se encarga de intentar abrir una obra SAICIC Windows PRO, para posteriormente si tuvo éxito cargar el nombre de la obra dentro del Control Panel de SAICIC, sino se logra abrir la obra se libera el apuntador de memoria referenciado por lpInfoObra

```
OL CMainFrame::fnAbreObra ( LPMDIVentStruct lpInfoObra )
```

```
LPSTR pszNomObra = NULL;
BOOL bRegreso = TRUE;
CString strDirObra;

//Logro crear la Memoria. Tenemos que crear archi
//la base de datos
lpInfoObra->ulBanderas = 0;
DDfnIniBanderasObra( m_hWnd, (LPOBRASTRUCT) lpInfoObra );

BeginWaitCursor ( ); //Aparece el HourGlass

strcpy ( lpInfoObra->szNombreData, theApp.m_pszExeName );
OutputDebugString ( lpInfoObra->szDir );
```

Esta función es fundamental para la apertura de una nueva obra.

```
If ( DDfnAbreObra ( m_hWnd, (LPOBRASTRUCT) lpInfoObra,
theApp.m_szUsuario, TRUE ) ) {
```

```
strDirObra = lpInfoObra->szDrive;
strDirObra += lpInfoObra->szDir;
```

```
theApp.m_pPanelCentral->fnAgregaObra ( lpInfoObra );
```

```
}
else {
```

```
//Tenemos que borrar el elemento de la lista
delete lpInfoObra;
bRegreso = FALSE;
```

```
} endif
```

```
EndWaitCursor ( ); //Desaparece el hourglass
```

```
return bRegreso;
```

```
endfunction
```

FUNCION: OpenDocumentFile (LPCSTR lpszFileName)

PARAMETROS: LPCSTR lpszFileName Nombre del archivo a abrir

DESCRIPCION: Esta función se encarga de abrir un archivo desde el menú de obras recientes, sin embargo se modifica para que se cargen la obra que se encuentra en el directorio especificado

```
Document* CMainFrame::OpenDocumentFile(LPCSTR lpszFileName)
```

```
LPMDIVentStruct lpInfoObra; //Información sobre la obra
LPSTR pszNomObra; //Nombre de la obra chat *
```

```
// find the highest confidence
char szPath[ MAX_PATH];
```

```
If ( AfxFullPath ( szPath, lpszFileName ) ) {
```

```
If ( lpInfoObra = (LPMDIVentStruct) new MDIVentStruct ) {
```

```
DDfnMaLimpiaBuffer ( lpInfoObra->szDrive );
DDfnMaLimpiaBuffer ( lpInfoObra->szDir );
```

```
//Tenemos que llenar la estructura de información que nos es solicitada
//para lpMDIInfoObra
strcpy ( lpInfoObra->szDrive, (LPSTR) szPath, 2 );
```



```

}endfunction

void CMainFrame::OnCapitulos()
{
    theApp.m_pPanelCentral->fnObtenObraSelecc ();
    theApp.m_iTipoForma = ANALISIS_CAPITULOS;
    theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_ANALISIS,
        theApp.m_lpMDIGen );
}endfunction

void CMainFrame::OnExtraordinarios()
{
    theApp.m_pPanelCentral->fnObtenObraSelecc ();
    theApp.m_iTipoForma = ANALISIS_EXTRA;
    theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_ANALISIS,
        theApp.m_lpMDIGen );
}endfunction

void CMainFrame::OnFactorSobreCosto()
{
    //Se ha invocado el llamado a los factores de Sobre Costo
    CFactorSobreCosto dlgFactores;

    theApp.m_pPanelCentral->fnObtenObraSelecc ();
    dlgFactores.DcModal();
}

void CMainFrame::OnFsr()
{
    //Creamos nueva ventana con la opción del Factor de Salario Real
    theApp.m_pPanelCentral->fnObtenObraSelecc ();
    theApp.m_iTipoForma = IDD_FSR;
    theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_FSR, theApp.m_lpMDIGen );
}

void CMainFrame::OnViewTool()
{
    m_wndToolBar.ShowWindow( ( m_wndToolBar.GetStyle() & WS_VISIBLE ) == 0 );
    RefreshLayout();
}

void CMainFrame::OnUpdateViewTool(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck( ( m_wndToolBar.GetStyle() & WS_VISIBLE ) != 0 );
}

void CMainFrame::OnMateriales()
{
    //Tenemos que crear una nueva ventana de Materiales, es sencillo, lo que tenemos que hacer
    //es llamar a la función del control panel que lleva a cabo este procedimiento
    theApp.m_pPanelCentral->fnObtenObraSelecc ();
    theApp.m_iTipoForma = IDD_MATERIALES;
    theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS, theApp.m_lpMDIGen );
}

void CMainFrame::OnObraDeMano()
{
    //Tenemos que crear una nueva ventana de Materiales, es sencillo, lo que tenemos que hacer

```

La opción de formatos de sobre costo se encuentra implementado a través de un diálogo y no una ventana de control.

Capítulo IV Desarrollo del Sistema

```
//es llamar a la función del control panel que lleva a cabo este procedimiento
theApp.m_pPanelCentral->fnObtenObraSelec ();
theApp.m_iTipoForma = IDD_OBRA_MANO;
theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS, theApp.m_ipMDIGen );
}

void CMainFrame::OnSubcontratos()
{
    //Tenemos que crear una nueva ventana de Materiales, es sencillo, lo que tenemos que hacer
    //es llamar a la función del control panel que lleva a cabo este procedimiento
    theApp.m_pPanelCentral->fnObtenObraSelec ();
    theApp.m_iTipoForma = IDD_SUBCONTRATO;
    theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS,
        theApp.m_ipMDIGen ); //Obtener el apudador del MDI
}

void CMainFrame::OnMaquinaEquipo()
{
    //Tenemos que crear una nueva ventana de Materiales, es sencillo, lo que tenemos que hacer
    //es llamar a la función del control panel que lleva a cabo este procedimiento
    theApp.m_pPanelCentral->fnObtenObraSelec ();
    theApp.m_iTipoForma = IDD_MAQUINARIA;
    theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS,
        theApp.m_ipMDIGen ); //Obtener el apudador del MDI
}

void CMainFrame::OnFamilias()
{
    //Creación de Nueva Ventana de Familias
    theApp.m_pPanelCentral->fnObtenObraSelec ();
    theApp.m_iTipoForma = IDD_FAMILIAS;
    theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS,
        theApp.m_ipMDIGen );
}

int function
void CMainFrame::OnZonas()
{
    //Creación de Nueva Ventana de Zonas
    theApp.m_pPanelCentral->fnObtenObraSelec ();
    theApp.m_iTipoForma = IDD_ZONAS;
    theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS,
        theApp.m_ipMDIGen );
}

int function
void CMainFrame::OnCatMoneda()
{
    CCatMoneda dlgCatMoneda;

    theApp.m_pPanelCentral->fnObtenObraSelec ();
    dlgCatMoneda.DoModal();
}

int function
void CMainFrame::OnUpdateViewToolBar(CCmdUI* pCmdUI)
{
    // Aja, aja aja
    pCmdUI->SetCheck( ( m_wndToolBar.GetStyle() & WS_VISIBLE ) != 0 );
}

int function
void CMainFrame::OnViewToolBar()
{
    m_wndToolBar.ShowWindow( ( m_wndToolBar.GetStyle() & WS_VISIBLE ) != 0 ?
```

La opción de monedas también se encuentra controlado por un diálogo como tal lo habilitamos.

```

        SW_SHOW | SW_HIDE );

if ( !m_wndToolBar.m_bEsPopup )
{
    RecalcLayout();
}
endif

id CMainFrame::OnChangeObra ()

// Ha cambiado la selección de obra dentro del combobox, tenemos
// que cambiarnos al directorio que se especifica
theApp.m_pPanelCentral->fnObtenObraSelec ();

idfunction

id CMainFrame::OnReportesPrevu()

CReporte dlgReporte;

theApp.m_pPanelCentral->fnObtenObraSelec ();
dlgReporte.m_lpInfoObra = theApp.m_lpMDIGen;
dlgReporte.DoModal();

idfunction

id CMainFrame::OnUpdateListaCveReuerdo(CCmdUI* pCmdUI)

if ( theApp.m_lpMDIGen )
{
    pCmdUI->Enable ( (BOOL) theApp.m_lpMDIGen->m_StrClavesRem.GetSize () );
}
else
{
    pCmdUI->Enable ( FALSE );
}
endif

id CMainFrame::OnListaCveReuerdo()

CBookmark  dlgBookmark;
dlgBookmark.m_lpMDIInfo = theApp.m_lpMDIGen;
if ( IDOK == dlgBookmark.DoModal () )
{
    theApp.m_iTipoForma = dlgBookmark.m_nTipoForma;

    if ( (int) dlgBookmark.m_nElemento
        <
        theApp.m_lpMDIGen->m_StrClavesRem.GetSize () )
    {
        theApp.m_strClaveMostrar =
            LPCTSTR(theApp.m_lpMDIGen->m_StrClavesRem[ dlgBookmark.m_nElemento ] + 2);

        if ( !theApp.m_strClaveMostrar.IsEmpty () )
        {

```

Esta opción es útil para cuando un usuario desea traer una clave de la lista de claves a recordar.


```

if ( pwndBusqueda && IsWindow ( pwndBusqueda->GetSafeHwnd () ) )
{
    pwndBusqueda->ShowWindow ( SW_HIDE );
    pwndBusqueda->UpdateWindow ();
}
endif

if ( pwndCatClaves && IsWindow ( pwndCatClaves->GetSafeHwnd () ) )
{
    pwndCatClaves->ShowWindow ( SW_HIDE );
    pwndCatClaves->UpdateWindow ();
}
endif
}
endif
}

void CMainFrame::OnBuscarDescrip()
{
    fnCambiaPane ( DDFnTraeStringLib (
        theApp.m_hInstance, ID_INDICATOR_DESC ), 1 );
    m_IBanderas = BAN_POR_DESC;
    CMDIGenSAI *pMDIGenSAT = ( CMDIGenSAI * ) MDIGetActive ( FALSE );
    if ( pMDIGenSAI )
    {
        pMDIGenSAI->m_IBanderas = BAN_POR_DESC;
    }
}

void CMainFrame::OnUpdateBuscarDescrip(CCmdUI* pCmdUI)
{
    CMDIGenSAI *pMDIGenSAT = ( CMDIGenSAI * ) MDIGetActive ( FALSE );
    if ( pMDIGenSAT )
    {
        pCmdUI->SetCheck ( ( pMDIGenSAT->m_IBanderas & BAN_POR_DESC ) ?
            1 : 0 );
    }
    else
    {
        pCmdUI->SetCheck ( ( m_IBanderas & BAN_POR_DESC ) ?
            1 : 0 );
    }
}

void CMainFrame::OnExplosion()
{
    theApp.m_pPanelCentral->fnObtenObraSelec (1);
    theApp.m_iTipoForma = IDD_EXPLOSION;
    theApp.m_pPanelCentral->fnAyarRelOpciones ( TEMPLATE_EXPLOSION,
        theApp.m_lpMDIGen );
}

void CMainFrame::OnHerramienta()
{
}

```

Capítulo IV Desarrollo del Sistema

```
theApp.m_pPanelCentral->fnObtenObraSelec ();
theApp.m_iTipoForma = IDD_HERRAMIENTA;
theApp.m_pPanelCentral->fnAbrRelOpciones ( TEMPLATE INSUMOS,
theApp.m_lpMDIgen );

}endfunction

void CMainFrame::OnUtileriasConfigaalcic()
{
    // OK mandamos a ejecutar el comando de Configuración de SAICIC para
    // Windows

    CConfigura dlgConfigura;

    theApp.m_pPanelCentral->fnObtenObraSelec ();
    dlgConfigura.DoModal ();

}endfunction

void CMainFrame::OnEstimaciones()
{
    theApp.m_pPanelCentral->fnObtenObraSelec ();
    theApp.m_iTipoForma = IDD_ESTIMACIONES;
    theApp.m_pPanelCentral->fnAbrRelOpciones ( TEMPLATE INSUMOS,
theApp.m_lpMDIgen );

}

void CMainFrame::OnActualizarPrecios()
{
    CActualiza dlgActualiza;

    theApp.m_pPanelCentral->fnObtenObraSelec ();
    dlgActualiza.m_lpInfoObra = theApp.m_lpMDIgen;
    dlgActualiza.DoModal ();

}endfunction
```

El manejo de mapas de mensajes dentro del MFC es la pieza fundamental para manejar los comandos que se le ofrecen al usuario. El funcionamiento de los mapas de mensajes es de la siguiente forma:

La ventana principal busca una ventana hija activa, Si existe dicha ventana se envía el mensaje a su forma, que es el objeto que permite la presentación de los datos en la ventana, si la forma no procesa el mensaje está lo envía al documento que es el medio de control de los datos desde la base de datos o documento hacia la vista, si este no lo procesa se envía de regreso a la ventana principal, si este tampoco lo

procesa se prueba si la aplicación o CWinApp puede procesarlo, sino es así el mensaje se pasa por alto.

este mecanismo permite que de acuerdo al objeto activo nosotros podamos habilitar cierto funcionamiento de los comandos, es decir como si aplicáramos el poliformismo a cada uno de los elementos visuales. Así por ejemplo la creación de una nueva opción depende desde que objeto fue invocada su creación.

En el corazón del Framework se encuentran los conceptos de documento y vista. un documento es un objeto de datos con el cuál interactuamos en una sesión de edición. El documento es creado por los comandos de Nuevo y Abrir y generalmente es salvado mediante el comando de Salvar. Una vista es un objeto del tipo Window a través del cual el usuario interactúa con el documento. El MFC define que un elemento clave dentro de la aplicación son los documentos. Nuestra clase de documentos CSaioicDoc es derivada desde la clase Cdocument. Sin embargo en la edición de datos a través de formas el manejo de documentos no es muy útil, sin embargo tenemos que crearlo ya que sin el no se cumple con las reglas del framework para crear una vista de datos.

Ahora dentro de nuestro programa la vista tiene una mayor importancia para el control de datos, y es ahí donde se controla la mayor parte de la aplicación. La clase vista utilizada es CFormaGral. La clase vista especifica como el usuario ve los datos e interactúa con ella. En algunos casos, se desea que un documento tenga múltiples vistas, como es el caso de Análisis que contiene dos vistas. Si se necesita hacer

Pagineo en la vista para ver más datos de los permitidos se hace uso de la clase CScrollView. Sin embargo para nuestro caso es necesario una vista en la cuál puedan mostrarse los campos y botones para que el usuario haga uso de la misma, la clase que se encarga de este comportamiento es CFormView, CFormaGral se deriva de este tipo de clase. CFormaGral es la base para todo tipo de vistas que se presentan dentro de SAICIC de aquí se derivan las clases de control de vistas de:

- Materiales CMatForm
- Obra de Mano CObraManForm
- Maquinaria y Equipo CMaqForm
- Subcontratos CSubcontForm
- Análisis CAnalisisView
- Factor de Salario Real CFSRForm
- Explosión CExplosion
- Herramientas CHerraForm

A continuación mostramos como se encuentra codificado la clase CFormaGral

```
//PROYECTO: SAICIC Windows PRO
//ARCHIVO: FORMAGRA.CPP
//AUTOR: RHCR
//FECHA DE MODIFICACION:
//NOTIFICACION:
//COMENTARIOS:
//      Contiene las rutinas que se encargan de manipular las formas
//      tanto de analisis como de insumos
//

#include "stdafx.h"
#include "saicic.h"
#include "mainfrm.h"
#include "formagra.h"
#include "odotbas.h"
#include "matform.h"           // Información de materiales

#ifdef _DEBUG
#undef THIS_FILE
```

```

static char BASED_CODE THIS_FILE[] = __FILE__ ;
#endif

// Nuestras famosas variables globales
HWND CFormaGral::m_hwndHijo = NULL;
Margar MarcaTexto;

////////////////////////////////////
// CFormaGral

IMPLEMENT_DYNCREATE(CFormaGral, CFormView)

CFormaGral::CFormaGral()
    : CFormView(CFormaGral::IDD)
{
}

CFormaGral::CFormaGral( UINT IDD)
    : CFormView( IDD )
{
    //((AFX_DATA_INIT(CFormaGral)
    // NOTE: The ClassWizard will add member initializations here.
    m_Descrip = "";
    m_Clave = "";
    m_Familia = "";
    m_Unidad = "";

    //))AFX_DATA_INIT
    m_CostoUnitario = 0.0;
    m_ClaveAnterior = ""; //Limpiamos el campo

    //GENINS : 12/EHE/94 : RCR : 0001
    m_ClaveBusqueda = ""; //Ultima clave buscada
    m_s2TipoDato = "";
    m_lpObraInfo = TheApp.m_lpMDIGen;

    m_iVecesChecaHabilitar = 0;
    m_uiRunderas = 0;
    m_uiIDControlChecar = 0;
    m_addrActual = NULL_DBA;
    m_ColorGrid = oConfigura.m_ColorGrid;
    m_bCambios = FALSE;

    m_ColorExiste = oConfigura.m_ColorExiste;
    m_ColorNoExiste = oConfigura.m_ColorNoExiste;
    m_ColorNoAnali = oConfigura.m_ColorNoAnali;
    m_hwndFocus = NULL;
}

void CFormaGral::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    //((AFX_DATA_MAP(CMatForm)
    DDX_Text(pDX, IDC_DESCRIP, m_Descrip);
    DDX_Text(pDX, IDC_INSUMOS, m_Clave);
    DDX_Text(pDX, IDC_FAMILIA, m_Familia);
    DDX_Text(pDX, IDC_UNIDAD, m_Unidad);
    //))AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CFormaGral, CFormView)
    //((AFX_MSG_MAP(CFormaGral)
    ON_WM_CREATE()
    ON_WM_LBUTTONDOWN()
    ON_WM_PAINT()
    ON_EN_KILLFOCUS(IDC_INSUMOS, OnKillFocusInsumos)
    //))AFX_MSG_MAP
    ON_COMMAND(IDOK, OnOK)

```

Inicialización de la clase, inicializamos las variables miembro o propiedades de la clase.

Inicialización de variables que habilitan la comunicación con los campos comunes de todas las vistas.

Habilitamos comunicación DDX (Dialog Data Exchange) con campos comunes de la vista.

Capítulo IV Desarrollo del Sistema

```

ON COMMAND(ID_LIMPIAR, OnLimpiar)
ON COMMAND(ID_EDIT_COPY, OnCopy)
ON COMMAND(ID_EDIT_PASTE, OnPaste)
ON COMMAND(ID_EDIT_CUT, OnCut)
ON UPDATE_COMMAND_UI ( ID_EDIT_CUT, OnUpdateCut )
ON COMMAND(ID_PRIMER_REG, OnPrimerRegistro)
ON COMMAND(ID_ULTIMO_REG, OnUltimoRegistro)
ON COMMAND(ID_REG_ANTERIOR, OnRegistroAnterior)
ON COMMAND(ID_REG_POSTERIOR, OnRegistroPosterior)
ON COMMAND(ID_BUSCAR, OnBuscarClave)
ON COMMAND(ID_RECUERDA_CLAVE, OnRecuerdaClave)
ON UPDATE_COMMAND_UI (IDOK, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_CATALOGO, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_RECUERDA_CLAVE, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_BUSCAR, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_REG_ANTERIOR, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_PRIMER_REG, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_REG_POSTERIOR, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_ULTIMO_REG, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_HORRAR, OnUpdateBuscaClave)
ON UPDATE_COMMAND_UI (ID_FILE_PRINT, OnUpdateImpression)
ON MESSAGE ( SSM_DRAG_DROP, OnDragSpread )
ON UPDATE_COMMAND_UI ( ID_GAABAR, OnUpdateGrabar )
ON UPDATE_COMMAND_UI ( ID_INFORMACION, OnUpdateGralar )
ON UPDATE_COMMAND_UI ( ID_EDIT_PASTE, OnUpdatePaste )
ON CONTROL ( SSM_SETFOCUS, IDCSP_PRECIOS, OnSetFocusSpread )
ON EN_CHANGE(IDC_INSUMOS, OnChangeData)
ON EN_CHANGE(IDC_DESCRIP, OnChangeData)
ON EN_CHANGE(IDC_FAMILIA, OnChangeData)
ON EN_CHANGE(IDC_UNIDAD, OnChangeData)
ON EN_CHANGE(IDC_F_ECHA_COSTO, OnChangeData)
ON EN_CHANGE(IDC_F_ECHA_CAL, OnChangeData)
ON WM_CLOSE ( )
ON COMMAND(ID_CATALOGO, OnCatalogo)
ON COMMAND ( IDCANCEL, OnCancel )
ON COMMAND ( ID_SPREAD_CUT, OnSpreadCut )
ON COMMAND ( ID_SPREAD_COPY, OnSpreadCopy )
ON COMMAND ( ID_SPREAD_PASTE, OnSpreadPaste )
ON COMMAND( ID_LISTA_OVE_RECUERDO, OnListaOveRecuerdos )
END_MESSAGE_MAP()

```

```

////////////////////////////////////
// CFormaGral message handlers

```

```

int CFormaGral::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFormView::OnCreate(lpCreateStruct) == -1)
        return -1;

    // TODO: Add your specialized creation code here

    // Se ha logrado crear la ventana, procedemos a Registrar a esta forma
    // como target de un drop
    m_DropTarget.Register( this );
}

```

```

return 0;
}

```

```

void CFormaGral::OnInitialUpdate ()
{
    CWnd *pWndInsu;

    pWndInsu = GetDlgItem( IDC_INSUMOS );
    ( (CEdit *) pWndInsu )->LimitText ( sizeof INSUMOS * sizeof CHAR );

    if ( pWndInsu = GetDlgItem ( IDC_FAMILIA ) )
    {
        ( (CEdit *) pWndInsu )->LimitText ( sizeof INSUGEN * sizeof CHAR );
    }

    if ( pWndInsu = GetDlgItem ( IDC_UNIDAD ) )

```

Base fundamental para inicialización de datos dentro de las vistas. Las vistas a diferencia de los diálogos no procesan el mensaje WM_INITDIALOG, pero utilizan OnInitialUpdate para tal efecto

```

(
    ( ( CEDIT * ) pWndInsu )->LimitText (sizeof_INSUGEN_SZUNIDAD - 1);
)endif

m_Spread.fnSetBool ( SS_BOOL_CTRL_DEL, TRUE );
m_Spread.fnSetBool ( SS_BOOL_INS_LINEA, TRUE );
fnCambiaFont ( m_hWnd );

Ctl3dSubclassDlg(m_hWnd, CTL3D_ALL);

ndfunction

id CFormaGral::OnInitialUpdate ( CString & Clave )

CFormaGral::OnInitialUpdate ();

if ( theApp.m_strClaveMostrar.GetLength() ) {
    INT nTipoDeBusqueda = BUSCAR,
        nSubTipoBusqueda = 0;

    // Existen clave que debe ser presentada en el análisis
    if ( !fnCargaDatos ( theApp.m_strClaveMostrar,
                        nTipoDeBusqueda,
                        nSubTipoBusqueda,
                        FALSE )
        != S_OKAY ) {

        Clave = theApp.m_strClaveMostrar;
        UpdateData ( FALSE );

    }endif
}endif

theApp.m_strClaveMostrar = "";

ndfunction

*void CFormaGral::fnColocaOpcionesClipboard ( CMenu &Menu )

// Colocamos las opciones del clipboard dentro del menu
Menu.AppendMenu ( MF_STRING | MF_ENABLED,
                 ID_CAT_MONEDA,
                 DDInTraeStringLib (
                     theApp.m_hlp_Lance,
                     IDST_CAT_MONEDA ) );

ndfunction
/

id CFormaGral::OnLButtonDown(UINT nFlags, CPoint point)

CObjDataSource *pDataInsu; // Creamos nuestro data source para llevar
CRect rect ( 0, 0, 0, 0); // Rectangulo para inicio del drag
DROPEFFECT dropResult; // Resultado de arrastrar información
CDC *pDC;

TRY {

    if ( !fnGetBool ( FM_ELIGE_INFO ) ) {

        // El usuario desea seleccionar la información que desea copiar
        // Ahora lo que tenemos que hacer es llevar a cabo los siguientes
        // pasos:

        // 1. Empezar a pintar un rectangulo que indique que el usuario

```

Toda forma derivada de CFormaGral cuenta con soporte de cortar, copiar y pegar, además de poderse arrastrar la información estilo drag and drop


```

// esta seleccionando información a copiar OK.
// Prendemos bandera recibirá información el mouse move
CRectTracker rectTracker ( rect,
                          CRectTracker::hatchInside );

rectTracker.TrackRubberBand ( this, point );

if ( rectTracker.m_rect.left != rectTracker.m_rect.right
    || rectTracker.m_rect.top != rectTracker.m_rect.bottom ) {

    fnSetBool ( FM_ELIGE_INFO, TRUE );

    // 2. Ahora colocamos los rectangulos que indican que el control
    // se encuentra seleccionado para ser copiado
    CRect rectIntersect;
    UINT nEleSel = 0; //Elementos seleccionados para el copiado

    pDC = GetDC ();

    for ( m_hwndHijo = ::GetWindow ( m_hwnd, GW_CHILD );
          m_hwndHijo;
          m_hwndHijo = ::GetWindow ( m_hwndHijo, GW_HWNDNEXT ) ) {

        // 1. Verifiquemos si la ventana puede ser seleccionada para
        // ser copiada
        if ( fnEsControlCopiar ( m_hwndHijo ) ) {

            // efectivamente la ventana puede ser seleccionada
            // 2. Obtuvimos una ventana chequeemos si esta dentro de nuestra
            // lista de ventanas hijas candidatas a ser seleccionadas
            // La siguiente función siempre regresa las coordenadas de
            // pantalla
            ::GetWindowRect ( m_hwndHijo, (LPRECT)rect );

            // 3. Pasamos el rectangulo a coordenadas de cliente de esta
            // vista
            ScreenToClient ( rect );

            rectTracker.m_rect.NormalizeRect ();
            rectIntersect = rect & rectTracker.m_rect;

            if ( rect == rectIntersect ) {

                // 4. Como dicen los quingos bingo, ahora tenemos ya el area
                // de intersección y resulta que es igual a la región
                // que ocupa nuestro rectangulo perteneciente a nuestro
                // control, ya sabes que hacer, marcarlo
                // Cambiamos el tamaño del rectangulo de esta forma
                // marcamos la parte externa del control
                CreaRectSeleccion ( pDC, rectIntersect );
                m_arrIDEAIs ( ::GetDlgItemID ( m_hwndHijo ) );
                nEleSel++;

            }

        }

    }

}

}

}

if ( nEleSel ) {

    // Si hay elementos seleccionados, tenemos que deshabilitar los
    // controles que pertenecen a esta forma

    for ( m_hwndHijo = ::GetWindow ( m_hwnd, GW_CHILD );
          m_hwndHijo;
          m_hwndHijo = ::GetWindow ( m_hwndHijo, GW_HWNDNEXT ) ) {

        ::EnableWindow ( m_hwndHijo, FALSE );

    }

}

// Lo quitamos y lo volvemos a poner
CreaRectSeleccion ( pDC, rectIntersect );

```

```

        CreaRectSelecion ( pDC, rectIntersect );
    }
    else
        fnSetBool ( FM_ELIGE_INFO, FALSE );
    endif

    // Adios al device context
    ReleaseDC ( pDC );

}endif

}
else {
    int nIndice;
    BOOL bIniciaDrag = FALSE;

    fnSetBool ( FM_ELIGE_INFO, FALSE );

    // 1. Tenemos que verificar si efectivamente se elige una región dentro
    // de un control que se desea copiar
    nIndice = m_arrICS.GetUpperBound ();

    while ( nIndice > -1 ) {
        // 2. Obtenemos la ventana asociada al ID
        m_hwndHijo = ::GetDlgItem ( m_hWnd, m_arrIDS[ nIndice ] );
        ::GetWindowRect ( m_hwndHijo, (LPRECT)rect );
        // 3. Pasamos el rectangulo a coordenadas de cliente de esta
        // vista
        ScreenToClient ( rect );

        if ( rect.PtInRect ( point ) ) {
            bIniciaDrag = TRUE;
            break;
        }

    }endif

    nIndice--;
} endwhile

if ( TRUE == bIniciaDrag ) {
    // Es necesario llevar a cabo una acción de drag and drop
    // Obtenemos información de la forma
    pDataInsu = fnReallyCopy ();

    GetClientRect ( &rect );
    ClientToScreen(&rect);

    dropResult = pDataInsu->DoDragDrop(
        DROPEFFECT_COPY,
        &rect );

}endif

fnValidaForma ();

}endif
}
CATCH_ALL (e)
{
    // No hay suficiente memoria o algo por el estilo
}
END_CATCH_ALL.
}

```

Capítulo IV Desarrollo del Sistema

```
void CFormaGral::CreaRectSeleccion ( CDC *pDC, CRect & rectSeleccion )
{
    // Esquina superior Izquierda
    pDC->PatBlt ( rectSeleccion.left - ANCHO_RECT_SEL,
                 rectSeleccion.top - ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Parte media superior
    pDC->PatBlt ( rectSeleccion.left + ( rectSeleccion.right - rectSeleccion.left ) / 2,
                 rectSeleccion.top - ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Esquina superior derecha
    pDC->PatBlt ( rectSeleccion.right,
                 rectSeleccion.top - ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Parte Media extremo derecho
    pDC->PatBlt ( rectSeleccion.right,
                 rectSeleccion.top + ( rectSeleccion.bottom - rectSeleccion.top ) / 2,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Esquina inferior derecha
    pDC->PatBlt ( rectSeleccion.right,
                 rectSeleccion.bottom,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Parte Media inferior
    pDC->PatBlt ( rectSeleccion.left + ( rectSeleccion.right - rectSeleccion.left ) / 2,
                 rectSeleccion.bottom,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Esquina inferior izquierda
    pDC->PatBlt ( rectSeleccion.left - ANCHO_RECT_SEL,
                 rectSeleccion.bottom,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );

    // Parte media lado izquierdo
    pDC->PatBlt ( rectSeleccion.left - ANCHO_RECT_SEL,
                 rectSeleccion.top + ( rectSeleccion.bottom - rectSeleccion.top ) / 2,
                 ANCHO_RECT_SEL,
                 ANCHO_RECT_SEL,
                 DSTINVERT );
}

void CFormaGral::OnPaint()
{
    CPaintDC dc(this); // device context for painting
    int m_nIndex = m_arrayDS.GetUpperBound ();
    CRect rect;

    // Tenemos que verificar si es necesario llevar a cabo el pintado de los
    // controles seleccionados
}
```

Para soportar la retroalimentación de selección de campos a copiar por el usuario tenemos que manejar el mensaje WM_PAINT.

```

if ( fnGetBool ( EM_ELIGE_INFO ) ) {
    // Tenemos que colocar los rectangulos de selección
    while ( nIndex > -1 ) {
        // 2. Obtenemos la ventana asociada al ID
        m_hwndHijo = ::GetDlgItem ( m_hWnd, m_arrIDS[ nIndex ] );
        ::GetWindowRect ( m_hwndHijo, (LPRECT)rect );
        // 3. Pasamos el rectangulo a coordenadas de cliente de esta
        // vista
        ScreenToClient ( rect );
        // 4. Hasta la vista baby, rectangulo de selección
        CreaRectSelección ( &dc, rect );
        nIndex--;
    } endwhile
} endif

// Do not call CFormView::OnPaint() for painting messages

/ Limpiamos la ventana de cualquier rebaba que aparezca en la
/ pantalla
void CFormaGral::fnValidaForma ()

CDC *pDC;
COLORREF colorFondo, colorFrente;

pDC = GetDC ();

// Habilitamos todas las ventanas del diálogo
for ( m_hwndHijo = ::GetWindow ( m_hWnd, GW_CHILD );
      m_hwndHijo;
      m_hwndHijo = ::GetWindow ( m_hwndHijo, GW_HWNDNEXT ) ) {
    // 1.a Habilitamos todas las ventanas hijas
    ::EnableWindow ( m_hwndHijo, TRUE );
} endif

ReleaseDC ( pDC );

// El spread ya se encuentra deshabilitado pero tenemos que
// cambiarle los colores para que se vea mejor el efecto, OK
m_Spread.GetColor ( 2, 1, &colorFondo, &colorFrente );
m_Spread.SetColor ( SS_HEADER,
                    SS_HEADER,
                    colorFondo,
                    RGBCOLOR_BLACK );

m_Spread.SetColor ( SS_ALLCOLS,
                    SS_ALLROWS,
                    colorFondo,
                    RGBCOLOR_BLACK );

InvalidateRect ( NULL, TRUE );

// Liberamos memoria que estamos ocupando
m_arrIDS.FreeExtra ();
m_arrIDS.RemoveAll ();
}endifunction

void CFormaGral::fnLlenaGenerales (
                                DBSINSUGEN & dbaDataGen,
                                LPCSTR lpcarPseudonimo
                                )

```

Capítulo IV Desarrollo del Sistema

```
CString strClaveAnt = m_ClaveAnterior;

OnLimpiar();
m_ClaveAnterior = strClaveAnt;
m_Clave = dbsDataGen.szClave + TAM_TIPO;
m_Familia = lpszSeudonimo + TAM_TIPO;
m_Unidad = dbsDataGen.szUnidad;

DDfnPonFechaenControl ( m_hWnd, IDCE_FECHA_COSTO, dbsDataGen.lpszFechaCosto );
DDfnPonFechaenControl ( m_hWnd, IDCE_FECHA_CAL, dbsDataGen.lpszFechaCalculo );
}endifunction

void CFormaGral::fnLimpiaControles ()
{
    UINT nIDSspread;
    int nindice;

    nIDSspread = ::GetDlgCtrlID ( m_Spread.m_hWnd );
    nindice = m_arrIDS.GetUpperBound ();

    while ( nindice > -1 ) {

        // Obtenemos la ventana asociada al ID, como de costumbre
        // tenemos que chequear si se trata del spread
        if ( m_arrIDS[ nindice ] == nIDSspread ) {

            m_Spread.fnLimpiaSpread();

        }
        else {

            m_hwndHijo = ::GetDlgItem ( m_hWnd, m_arrIDS[ nindice ] );

            ::SetWindowText ( m_hwndHijo, "" );

        }endif

        nindice--;

    } endwhile
}endifunction

void CFormaGral::fnCambiaStatusBar( LPCSTR lpsztext )
{
    CWinApp* pApp = AfxGetApp();
    CMainFrame* pFrame = (CMainFrame*)App->m_pMainWnd;

    pFrame->fnCambiaPane ( lpsztext );
}endifunction

void CFormaGral::OnActivateView ( BOOL bActive, CView *, CView * )
{
    if ( bActive )
    {
        if ( IsWindow ( m_hwndFocus ) )
        {
            ::SetFocus( m_hwndFocus );
        }
        else
        {
            SetFocus ();
        }
    }
}
```

Nuestro programa cuenta con autoconfiguración cuando el usuario ha cambiado parámetros de configuración, es en este momento que damos soporte a esta opción

```
CString strClaveAnt = m_ClaveAnterior;

OnLimpiar();
m_ClaveAnterior = strClaveAnt;
m_Clave = dbsDataGen.szClave + TAM_TIPO;
m_Familia = lpszSeudonimo + TAM_TIPO;
m_Unidad = dbsDataGen.szUnidad;

DDfnPonFechaenControl ( m_hWnd, IDC_F_ECHA_COSTO, dbsDataGen.lFechaCosto );
DDfnPonFechaenControl ( m_hWnd, IDC_F_ECHA_CAI, dbsDataGen.lFechaCalculo );

}endfunction

void CFormaGral::fnLimpiaControles ()
{
    UINT nIDSspread;
    int nIndice;

    nIDSspread = ::GetDlgCtrlID ( m_Spread.m_hWnd );
    nIndice = m_arrIDS.GetUpperBound ( );

    while ( nIndice > -1 ) {
        // Obtenemos la ventana asociada al ID, como de costumbre
        // tenemos que chequear si se trata del spread
        if ( m_arrIDS[ nIndice ] == nIDSspread ) {
            m_Spread.fnLimpiaSpread();
        }
        else {
            m_hwndHijo = ::GetDlgItem ( m_hWnd, m_arrIDS[ nIndice ] );
            ::SetWindowText ( m_hwndHijo, "" );
        }
        nIndice--;
    } endwhile
}endfunction

void CFormaGral::fnCambiaStatusBar( LPCSTR lpsztext )
{
    CWinApp* pApp = AfxGetApp();
    CMainFrame * pFrame = (CMainFrame*)pApp->m_pMainWnd;
    pFrame->fnCambiaPane ( lpsztext );
}endfunction

void CFormaGral::OnActivateView ( BOOL bActive, CView *, CView * )
{
    if ( bActive )
    {
        if ( !IsWindow ( m_hwndFocus ) )
        {
            ::SetFocus( m_hwndFocus );
        }
        else
        {
            SetFocus ( );
        }
    }
}
```

Nuestro programa cuenta con autoconfiguración cuando el usuario ha cambiado parámetros de configuración, es en este momento que damos soporte a esta opción

```

}
endif

m_Spread.SetBool ( SSB_ARROWSEXITFEDITMODE,
                  oConfigura.m_bUtilizaFlechas );
m_Spread.SetEditEnterAction ( oConfigura.m_wMueveCursorSpread );
m_Spread.SetBool ( SSB_EDITMODEREPLACE,
                  oConfigura.m_bReemplazarTexto );

// Ve si cambio el color del Grid
if ( m_ColorGrid != oConfigura.m_ColorGrid )
{
    // Cambio el color del Grid
    m_Spread.SetBool ( SSB_REDRAW, FALSE );
    m_Spread.SetGridColor ( oConfigura.m_ColorGrid );
    m_ColorGrid = oConfigura.m_ColorGrid;
    m_Spread.SetBool ( SSB_REDRAW, TRUE );
}

}endif

if ( m_ColorExiste != oConfigura.m_ColorExiste
    || m_ColorNoExiste != oConfigura.m_ColorNoExiste
    || m_ColorNoAnali != oConfigura.m_ColorNoAnali
    )
{
    // Bueno ha habido un pequeño cambio en la configuración
    // tenemos que cambiar los colores para ajustar esto a lo que el
    // usuario previamente definió
    SS_COORD ssRen = 1;
    COLORREF ColorFrente, ColorFondo;

    m_Spread.GetDataCnt ( );
    m_Spread.SetBool ( SSB_REDRAW, FALSE );

    while ( ssRen <= m_Spread.m_lNumRenglonas )
    {
        m_Spread.GetColor ( oConfigura.m_IDCOL_ANA_CLAVE,
                          ssRen,
                          &ColorFondo,
                          &ColorFrente );

        if ( ColorFrente == m_ColorExiste )
        {
            // Es el color de no Existe
            m_Spread.SetColorRange ( oConfigura.m_IDCOL_ANA_TIPO,
                                   ssRen,
                                   m_Spread.m_iLastCol,
                                   ssRen,
                                   ColorFondo,
                                   oConfigura.m_ColorExiste );
        }
        else
        {
            if ( ColorFrente == m_ColorNoExiste )
            {
                // Es el color de no Existe
                m_Spread.SetColorRange ( oConfigura.m_IDCOL_ANA_TIPO,
                                         ssRen,
                                         m_Spread.m_iLastCol,
                                         ssRen,
                                         ColorFondo,
                                         oConfigura.m_ColorNoExiste );
            }
        }
    }
}
else

```

```
{
    m_Spread.SetColorRange ( oConfigura.m_IDCOL_ANA_TIPO,
                            sken,
                            m_Spread.m_IllustCol,
                            sken,
                            ColorFondo,
                            oConfigura.m_ColorNoAnali );

}endif
}endif
ssRent++;
}endwhile
// OK ahora copia los colores que el usuario elige
m_ColorExiste = oConfigura.m_ColorExiste;
m_ColorNoExiste = oConfigura.m_ColorNoExiste;
m_ColorNoAnali = oConfigura.m_ColorNoAnali;
m_Spread.SetBool ( SSB_REDRAW, TRUE );
}endif
}
else
{
    m_hwndFocus = ::GetFocus ();
    m_Spread.InChecaEstatusKey ();
}
}endif
}endifunction

void CFormaGrat::OnChangeData()
{
    // TODO: Add your control notification handler code here
    m_bCambios = TRUE;
}endifunction

void CFormaGrat::OnClose()
{
    if ( m_bCambios )
    {
        // Debe haber clave de lo contrario no podemos grabar al insumo
        if ( ::GetWindowTextLength ( ::GetDlgItem ( m_hWnd, IDC_INSUMOS ) ) )
        {
            if ( IDYES == DfFr.DspyErrorUsuario (
                m_hWnd, theApp.m_hInstance,
                theApp.m_szUsuario, "",
                IDS_STRING446, MB_YESNO ) )
            {
                // Ahora tenemos que salvar la informacion solicitada
                OnGrabar();
            }endif
        }
    }
    else
    {
        if ( IDYES == DfFr.DspyErrorUsuario (
            m_hWnd, theApp.m_hInstance,
            theApp.m_szUsuario, "",
            IDS_STRING447, MB_YESNO ) )
        {

```



```

    {
        return;
    }
}

endif
endif
endif

ndfunction
id CFormaGrnl::OnKillFocusInsumos()

CWnd *pEditInsumos = GetDlgItem ( IDC_INSUMOS );
int nNumero;

//nBanderaas = pMDIActiva->m_lBanderas;
pEditInsumos->GetWindowText ( m_Clave );

// Tenemos que ver si se trata de un numero
nNumero = StringToInt ( (LPSTR) (LPCSTR) m_Clave );

if ( !oConfigura.m_bClavesNumericas && nNumero <= 9999 && nNumero > 0 )
{
    // Se trata de un numero por favor rellenalo con zeros
    wprintf ( szClave, "%04i", nNumero );
    m_Clave = szClave;

    pEditInsumos->SetWindowText ( m_Clave );
}

endif

```

```

id CFormaGrnl::fnInfoDbGeneral (
    CSATDataBase * pdbaData,
    DBSTNSUGEN & dbInsu,
    DBSFAMILIA & dbFamilia,
    dbRecordPadre & dbPadre,
    dbRecordHijo & dbHijo,
    LPDBSZONAECO pZonas,
    *DWordArray * dwArrayZonas
)

```

```

long lCurrency;
int nStatus = S_OKAY;

UpdateData (); // Traemos datos generales

lstrcpy ( dbInsu.szClave, m_szTipoDato );
lstrcat ( dbInsu.szClave, m_Clave );
StrLTrim ( dbInsu.szClave + TAM_TIPO );
StrRTrim ( dbInsu.szClave + TAM_TIPO );

```

```

//Obtener información sobre la familia es necesaria para establecer como se liga el
//material a la familia indicada
lstrcpy ( dbFamilia.szSecDenimo, IDENT_FAMILIAS );
lstrcat ( dbFamilia.szSecDenimo, m_Familia );
StrLTrim ( dbFamilia.szSecDenimo + TAM_TIPO );
StrRTrim ( dbFamilia.szSecDenimo + TAM_TIPO );

//Obten datos sobre la unidad
lstrcpy ( dbInsu.szUnidad, m_Unidad );

// Obtenemos las fechas de costos y calculo
dbInsu.lFechCosto = DBfnObtenJuliano ( m_hWnd, TDC_Fecha_Costo );
dbInsu.lFechCalculo = DBfnObtenJuliano ( m_hWnd, TDC_Fecha_Cal );
dbInsu.lMiembroSelec = BULL_DNA;

dbPadre.lTask = DBfnMaGetTarea ( m_lpObrainfo );

```

Este método o función miembro permite la parametrización adecuada de todos los campo comunes dentro de una vista derivada de CFormaGrnl

Capítulo IV Desarrollo del Sistema

```
if Padre.dBloqueo = BLOQUEA_ESCRI_COMINTO;
putsData->fnSetBufferHijo ( &dbPadre );

// Dependiente de Insumos
switch ( m_nTemplate ) {

case IDD_ANALISIS:
// Llenamos información sobre análisis
// obtenemos el costo seleccionado
// Ojo Anguitos tenemos que ver si el usuario puede o no grabar
// más información de un análisis
if ( oConfigura.m bDemo )
{
    int     nFile;
    long    lsl;
    DB_ADDR dbAddr;

}endif

fnObtenCosto ( 0 );

dbInsu.dCostUnitario = m_CostoUnitario;

// llenamos información sobre el buffer hijo para manejo de la base
// de datos
dbHijo.lRecordMiembro = ANALISGEN;
dbHijo.lCampoRehijo = ANALISGEN_SZCLAVE;
dbHijo.pszBufferMiembro = (LPSTR) &dbInsu;
dbHijo.lIndicaOwner = -1;
dbHijo.lSet = FAMIANALIS;
lCurrency = ANALISGEN_LALIASACTIVO;

break;

case IDD_MATERIALES:
case IDD_OBRA_MANO:
case IDD_MAQUINARIA:
case IDD_SUBCONTRATO:

if ( oConfigura.m bDemo )
{
    int     nFile;
    long    lsl;
    DB_ADDR dbAddr;

}endif

nStatus = fnGrabamosInsu ( m_Spread,
                          pdsData,
                          pZonas,
                          &dbPadre,
                          dwArrayZonas );

m_Spread.GetRowUserData ( m_Spread.m lRowBotonSel,
                          lCurrency );
dbInsu.lCurrency = lCurrency;

// Obtenemos el costo seleccionado
fnObtenCosto ( m_Spread.m lRowBotonSel );
dbInsu.dCostUnitario = m_CostoUnitario;

// Llenamos información sobre el buffer hijo para manejo de la base
// de datos
dbHijo.lRecordMiembro = INSUGEN;
dbHijo.lCampoRehijo = INSUGEN_SZCLAVE;
dbHijo.pszBufferMiembro = (LPSTR) &dbInsu;
dbHijo.lIndicaOwner = -1;
dbHijo.lSet = FAMIINSU;
lCurrency = INSUGEN_LALIASACTIVO;
break;

case IDD_HERRAMIENTA:
```

```

        dbHijo.iRecordMiembro = INSUGEN;
        dbHijo.lCampoRegHijo = INSUGEN_SZCLAVE;
        dbHijo.pszBufferMiembro = (LPSTR) &dbInsu;
        dbHijo.iIndicaOwner = -1;
        dbHijo.iSet = FAMILIASU;
        break;
    }
}endswitch

// Inicializamos la información sobre la base de datos SAJCIC
// Primero la información del registro Padre
dbPadre.lCampo = FAMILIA_SZSEUDONIMO;
dbPadre.iRegActual = FAMILIA;
dbPadre.pszBuffer = (LPSTR) &dbFamilia;

// OJO tenemos que hacer esto para encontrar si tiene campo de Alias
// el insumo o análisis y respetarlo
if ( S_OKAY == dt_keyfind ( dbHijo.lCampoRegHijo,
                            dbInsu.szClave,
                            dbPadre.dbTask,
                            dbPadre.iNumDataBase ) )
{
    // Si encuentro al elemento ahora obten del campo de alias
    // el dato que este contiene
    dt_crread ( lCurrency,
                &dbInsu.lAliasActivo,
                dbPadre.dbTask,
                dbPadre.iNumDataBase );
}

}endif

return nStatus;
}endfunction

BOOL CForma1::fnHayElementosSpread ( CSAIDataBase &dbData,
                                     long iNumEleDataBase
    )
{
    if ( m_Spread.GetEditMode() )
    {
        m_Spread.SendMessage ( WM_KEYDOWN, VK_RETURN, 1 );
        m_Spread.SendMessage ( WM_KEYUP, VK_RETURN, 1 );
    }

}endif

m_Spread.GetDataCnt ();

if ( m_Spread.m_lNumRegiones == 0 ) {

    // No hay datos en el spread de este elemento, tenemos que borrar
    // los elementos miembros que se encuentran relacionados a él,
    // pero antes para mayor velocidad verifiquemos si existen
    // elementos en la base de datos, sino no hacemos absolutamente
    // nada
    if ( iNumEleDataBase > 0 ) {

        //Si hay elementos, se borran
        dbData.fnBorraMiembros (); //Listo

    }endif //No hay elementos en la base de datos

    return FALSE;

}endif

return TRUE;

}endfunction

```

Capítulo IV Desarrollo del Sistema

```
void CFormaGral::fnBorraRestantes ( CSAIDataBase &dbaData,
                                   LONG lNumFileDataBase )
{
    int iDiferenciaEle;
    iDiferenciaEle = (int) ( lNumFileDataBase -
                             m_Spread.m_lNumRenglonos );

    for ( int iStatus = S_OKAY;
          iDiferenciaEle > 0;
          iStatus == S_OKAY;
          iDiferenciaEle-- ) {

        dbaData.fnBorraMember ();
        iStatus = dbaData.fnFncNextMember(); //Obtenemos el siguiente miembro
    } //Borra elementos Miembros
} //endfunction

void CFormaGral::fnGrabaDescrip (
    CSAIDataBase &dbaData,
    LPSTR pszDescrip,
    dbkRecordHijo & dRHijo,
    INT iSetDescrip )
{
    ASSERT ( pszDescrip );

    // Llenamos información sobre la descripción
    dRHijo.lRecordMiembro = DESCINSUMOS;
    dRHijo.lCampoReqHijo = DESCINSUMOS.szDESC;
    dRHijo.pszBufferMiembro = pszDescrip;
    dRHijo.lIndicaOwner = 00;
    dRHijo.lISet = INSUDESC;
    dbaData.fnSetBufferHijo ( &dRHijo,
                             iSetDescrip );

    // Grabamos las descripciones
    dbaData.fnGrabaDescripcion( m_Descrip,
                                m_szTipoData,
                                DESCINSUMOS | PDESCINSUMOS );
} //endfunction
```

```

FUNCION: fnCargaZonas ( CSAIDataBase & dbsData,
                      DBSZONAECO & Zonas,
                      dbRecordPadre & dbPadre
DESCRIPCION: fnCargaZonas se encarga de extraer la clave correcta de una
zona, ya que la clave que se encuentra del lado de los insumos
solo hace referencia a la clave sin blancos
d CFormaGral::fnCargaZonas (
    CSAIDataBase & dbsData,
    DBSZONAECO & Zonas,
    dbRecordPadre & dbPadre
)

```

Todos los insumos dentro de SAJIC soportan el manejo de zonas

```

// El registro padre se convertira en el registro de Zonas
dbPadre.lRegActual = ZONAECO;
dbPadre.lCampo = ZONAECO_SZZONA;
dbPadre.pszBuffer = (LPTSTR) &Zonas;
dbsData.fnSetPadreDirect ( &dbPadre );

lstncpy ( Zonas.szZona, IDENT_ZONAS );

for ( SS_COORD iNumEle = m_Spread.m_lNumRenglonas;
      iNumEle > 0;
      iNumEle--
    ) {
    // En el spread se encuentra la zona sin blancos
    lstncpy ( Zonas.szZona, IDENT_ZONAS );
    m_Spread.GetData ( IDCOL_ZONA, iNumEle,
                      Zonas.szZona + TAM_TIPO );
    // volvemos a obtener la clave de la Zona, pero ahora sin blancos
    // y la colocamos dentro del spread
    if ( S_OKAY == dbsData.fnKeyFind ( ZONAECO_SZZONA,
                                      Zonas.szZona,
                                      TRUE ) )
    {
        // OK si lo encontro, sino lo encuentra creamos que RCR esta en
        // problemas.
        // Atentamente RCR
        dbsData.fnLeeCampoRegActual ( ZONAECO_SZZONA,
                                     Zonas.szZona );

        m_Spread.SetData ( IDCOL_ZONA, iNumEle,
                          Zonas.szZona );
    }
}

}endif
}endfor
endifunction

```

```

FUNCION: fnBorrarElementodb ()
DESCRIPCION: Esta función se encargara de borrar la información de un
elemento como Insumo o analisis de la base de datos, es importante
hacer notar que la rutina pueda ser algo lenta ya que es necesario
chocar en donde se encuentra el elemento deseado. Un Insumo o un
analisis se puede encontrar en analisis o un estimaciones
Si el usuario decide solo marcar como borrado el elemento o no hacerlo
fisicamente será necesario colocar DBA_NULL a todos los elementos
que así sean considerados

```

```

d CFormaGral::fnBorrarElementodb (
    CSAIDataBase & dbsData,
    dbRecordPadre & dbPadre,
    dbRecordHijo & dbHijo
)

```

```

int iConjunto;
int iDim = 0;
DBSELEANALI dbsEleAnali, dbsEleContrasta;
CString strString;

```

La mayoría de las formas permiten el borrado de elementos, agruparlos el funcionamiento en un método general

Capítulo IV Desarrollo del Sistema

```
        strString2,
        strStringMsg;
int nStatus = S_OKAY;
DB_ADDR dbAddr = NULL;
BOOL bYaPregunto = FALSE;
int nAccion;

// En la versión 4.0 de SATCIC la dimensión siempre es cero, en futuras
// versiones, la dimensión variara, cuidado con eso
Zero ( dbxFieContraata );

UpdateData (); //Obtenemos información de los campos

istrncpy ( szClave, m_Clave );
StrlTrim ( szClave );
m_Clave = szClave;
wprintf ( szClave, "%s",
          (LPCSTR) m_szT.p_Dato,
          (LPCSTR) m_Clave );

istrncpy ( dbxFieAnali.szElemento, szClave );
dbxFieAnali.DB_ADDR Analisis = NULL;
// Inicializamos la información sobre la base de datos SATCIC
// Primero la información del registro Padre
if Padre.dbTask = DDFnMaObjTarea ( m_ipOperaInfo );
dbxFieAnali.uiBloqueo = BLOQUEA_ESCRI_CONJUNTO;

strStringMsg.LoadString ( IDST_SB_BUSCANDO );
fnCambiaStatusBar( strStringMsg );

// Primero tenemos que verificar si el componente a borrar no se encuentra
// dentro de cierto análisis, si lo esta tenemos la prioridad de avisarle
// al usuario
if Padre.iCampo = COMPONENTES_ELEANALI;
dbxFieAnali.iKeyActual = COMPONENTES;
if Padre.ps.iBuffer = (LPSTR) &dbxFieAnali;
dbxFieAnali.fnGetBufferPadre ( &dbxFieAnali );

// Ahora chequea si el elemento no se encuentra en maquinaria y equipo
int nRecord;
long iCampo;
int nConjuntoDescrip;

switch ( m_nTemplate )
{
    case IDD_ANALISIS:
        nRecord = ANALISGEN;
        iCampo = ANALISGEN_SLAVE;
        nConjuntoDescrip = ANALISGENDESC;
        break;

    default:
        nRecord = INSUGEN;
        iCampo = INSUGEN_SLAVE;
        nConjuntoDescrip = INSUGENDESC;
}

endswitch

if ( S_OKAY != ( nStatus = dbxFieAnali.fnKeyFind ( iCampo,
          szClave,
          TRUE ) ) )
{
    // La clave no existe desde el usuario
    return;
}

endif

if ( S_OKAY == dbxFieAnali.fnSearchkey ( COMPONENTES_ELEANALI,
          (LPSTR) &dbxFieAnali ) )
{
    // Se hizo el movimiento hacia algún elemento dentro de la base de datos
    // sin embargo hay que chequear si se trata de lo que queremos
}
```

```

if ( S_OKAY == dt_keyread ( (LPSTR) &dbaEleContraata, dbPadre.dbTask )
    *
    StriEqu ( dbaEleContraata.szElemento, szClave ) )
{
    // El componente se encuentra en uno o más análisis, tenemos que
    // solicitar al usuario su autorización para borrarlo
    MensajePrecio dlgAccion;

    strString.LoadString ( IDST_MSG_BORRAR );
    strString2.LoadString ( IDST_MSG_BORRAR2 );
    strString += ' ';
    strString += m_Clave;
    strString += ' ';
    strString += strString2;

    // Tenemos que liberar el bloqueo
    dbData.fnLiberaBloqueo ();

    // Mandamos a desplegar el mensaje especial preguntando al usuario
    // acerca del borrado
    dlgAccion.m_nTipoMensaje = IDS_STRING115;
    dlgAccion.m_strMensaje.LoadString ( IDS_STRING115 );
    dlgAccion.m_strRadio1.LoadString ( IDS_STRING116 );
    dlgAccion.m_strRadio2.LoadString ( IDS_STRING117 );
    bYaPregunto = TRUE;

    switch ( nAccion = dlgAccion.DoModal () )
    {
        case IDOK:
            // Lo anterior es por algo que no conozco, no sabría dar la
            // razón de porque se pierde parcialmente el valor de szClave,
            // pero de esta forma lo estamos solucionando
            wprintf ( szClave, "%s",
                (LPCSTR) m_szTipoDato,
                (LPCSTR) m_Clave );

            // OK acepta borrar al elemento, sin embargo pregunta cual
            // será el movimiento que haga, borrarlo definitivamente o
            // borrarlo de sus componentes

            // Tenemos que activar nuevamente el bloqueo
            dbData.fnSetbufferPadre ( &dbPadre );
            nStatus = dbData.fnStatus ();
            strString2.LoadString ( IDST_SB_BORRANDO );

            if ( IDC_RBT1 == dlgAccion.m_nTipoAccion )
            {
                // El usuario ha decidido borrar al elemento, y colocar en
                // DBA_NULL todos los componentes en los que aparece
                nStatus |= dbData.dt_searchkey ( COMPONENTES_ELEANALI,
                    (LPSTR) &dbaEleAnali );

                while ( S_OKAY == nStatus )
                {
                    // El elemento se encuentra dentro de los componentes, tenemos
                    // que desligarlo de su padre y después fuera.
                    fnCambiaStatusBar( strStringMsg );

                    dbaEleContraata.szElemento[0] = '\0';

                    dt_keyread ( (LPSTR) &dbaEleContraata, dbPadre.dbTask );

                    if ( StriEqu ( dbaEleContraata.szElemento, szClave ) )
                    {
                        // Coloca en el campo de dbADDR el DBA_NULL
                        nStatus |= dt_cwrite ( COMPONENTES_DB_ADDRSELEMENTO,
                            (LESTR) &dbAddr,
                            dbPadre.dbTask,
                            dbPadre.nNumDataBase );
                    }
                }
            }
        }
    }

```

Capítulo IV Desarrollo del Sistema

```
        fnCambiaStatusBar( strString2 );
    }
    else
        break;
    endif

    // Tenemos que pasarnos al siguiente elemento
    nStatus := dt_keynext ( COMPONENTES_ELEANALI,
                        dbPadre.dtTask,
                        dbPadre.lNumDataBase );
}endwhile
}
else
(
// El usuario ha decidido borrar definitivamente al
// elemento de la base de datos con todo y componentes
while ( S_OKAY == nStatus
      &&
      dbData.dt_searchkey ( COMPONENTES_ELEANALI,
                          {LPSTR} szEleAnali )
      ==
      S_OKAY )
(
// El elemento se encuentra dentro de los componentes, tenemos
// que desligarlo de su padre y después fuera.
fnCambiaStatusBar( strStringMsg );

dbEleContrastrata.szElemento[0] = '\0';

dt_keyread ( {LPSTR} szEleContrastrata, dbPadre.dtTask );

if ( StriEq ( dbEleContrastrata.szElemento, szClave ) )
(
// Procedemos a borrar el elemento de la base de datos, previamente
// tenemos que desconectarlo del análisis al que pertenece
dt_netmr ( BASICOFAT,
          dbPadre.dtTask,
          dbPadre.lNumDataBase );

dt_discon ( BASICOFAT,
          dbPadre.dtTask,
          dbPadre.lNumDataBase );

fnCambiaStatusBar( strString2 );
nStatus = dt_delete ( dbPadre.dtTask,
                    dbPadre.lNumDataBase );
)
)
else
    break;
endif

}endwhile
}endif
break;
}endswitch
}endif
```



```

}endif

if ( S_OKAY == ( nStatus = dbData.fnKeyFind ( lCampo,
                                             szClave,
                                             TRUE ) ) )
{
    DB_ADDR dbAddrAux = NULL_DBA,
            dbAddrMaq;
    BOOL    bFueEncontrado = FALSE;
    long    lCampoMaq;

    lCampoMaq = MAQUINA\EQUIPO_DB_ADESCLAVELLANEQAD1;
    dt_getet ( &dbAddr, dbPadre.dbTask, dbPadre.lNumDataBase );

    for ( int iCont = 0; iCont < 9; iCont++ )
    {
        if ( S_OKAY == dbData.dt_searchkey ( lCampoMaq, (LPSTR) &dbAddr ) )
        {
            dt_keyread ( (LPSTR) &dbAddrMaq, dbPadre.dbTask );

            if ( dbAddrMaq == dbAddr )
            {
                bFueEncontrado = TRUE;
                break;
            }
        }
    }

    }endif

    lCampoMaq++;
}endifor

if ( bFueEncontrado )
{
    CMensajePrecio dlgAccion;

    if ( !bYaPregunto )
    {
        // Es necesario preguntar al usuario si desea borrar
        // Al usuario no se le ha preguntado acerca de si borra o no los
        // elementos
        dlgAccion.m_nTipoMensaje = IDST_MSG_BORRAR;

        strString.LoadString ( IDST_MSG_BORRAR );
        strString2.LoadString ( IDST_MSG_BORRAR2 );
        strString += ' ';
        strString += m_Clave;
        strString += ' ';
        strString += strString2;
        bYaPregunto = TRUE;
        // Tenemos que liberar el bloqueo
        dbData.fnLiberaRiqueo ();

        // Mandamos a desplegar el mensaje especial preguntando al usuario
        // acerca del borrado
        dlgAccion.m_strMensaje.LoadString ( IDS_STRING115 );
        dlgAccion.m_strRadio1.LoadString ( IDS_STRING116 );
        dlgAccion.m_strRadio2.LoadString ( IDS_STRING117 );

        nAccion = dlgAccion.DoModal ();
    }

    }endif

    switch ( nAccion )
    {
        case IDOK:

```

Capítulo IV Desarrollo del Sistema

```
// OK acepta borrar al elemento, sin embargo pregunta cual
// será el movimiento que hará, borrarlo definitivamente o
// borrarlo de sus componentes

// Tenemos que activar nuevamente el bloqueo
dbData.fnSetBufferPadre ( &dbPadre );
nStatus = dbData.fnStatus ( );
strString2.LoadString ( IDC_STB_BORRADO );

if ( IDC_RPTN_1 == dlgAccion.m_uiTipoAccion )
{
    // En maquinaria a diferencia de lo que sucede en análisis
    // no se debe borrar al registro de CH ya que el elemento
    // solo es parte de la datos no equivale a un componente
    // como es el caso de maquinaria y equipo "
    // El usuario ha decidido borrar al elemento, y colocar en
    // DBA_NULL todos los componentes en los que aparezca
    iCampoMaq = MAQUINAEQUIPO_DB_ADDRCLAVELLANEQADI;

    for ( iCont = 0; iCont < 9; iCont ++ )
    {
        for ( nStatus = dbData.dt_searchkey ( iCampoMaq,
            (LPSTR) &dbAddr );
            S_OKAY == nStatus;
            nStatus != dt_keynext ( iCampoMaq,
                dbPadre.dbTask,
                dbPadre.iNumDataBase );
            {
                fnCambiaStatusBar( strStringMaq );
                dt_keyread ( (LPSTR) &dbAddrMaq, dbPadre.dbTask );
                if ( dbAddrMaq == dbAddr )
                {
                    // Coloca en el campo de dbADDR el DBA_NULL
                    nStatus = dt_crwrite (
                        iCampoMaq,
                        (LPSTR) &dbAddrAux,
                        dbPadre.dbTask,
                        dbPadre.iNumDataBase );
                    fnCambiaStatusBar( strStringQ );
                }
            }
        }
        iCampoMaq++;
    }
}

break;

case IDCANCEL:
    return; //Se arrepintio

}

}

// Tenemos que preguntar
```

```

wprintf ( szClave, "sisi",
          (LPCSTR) m_szTipoData,
          (LPCSTR) m_Clave );

if ( IDNO == DDInDeptyErrorUsuario | m_hWnd,
     theApp.m_hInstance,
     theApp.m_szUsuario,
     szClave,
     IDS_STRING158,
     MR_ICONQUESTION | MB_YESNO ) )
{
    return;
}

endif
endif

// Ahora si procede a borrar al elemento de la base de datos en sus
// datos generales
dbPadre.lCampo = lCampo;
dbPadre.lRegActual = nRecord;
dbPadre.pszBuffer = szClave;

dbData.fnSetBool ( DB_BOOL_INDEXADO, TRUE ); //Acceso Indexado
strStringMsg.LoadString ( IDST_SB_BUSCANDO_INSUMOS );
fnCambiaStatusBar( strStringMsg );

if ( S_OKAY == ( nStatus = dbData.fnKeyFind ( lCampo,
                                             szClave,
                                             TRUE ) ) )
{
    // Colocamos información sobre el registro hijo a acceder
    dbHijo.pszBufferMiembro = NULL;
    dbHijo.lIndicaOwner = 0;
    dbData.fnSetBufferHijo ( &dbHijo, -1 );

    //Si hay elementos, se borran los elementos del insumo
    dbData.fnBorraMiembros ();

    // Borramos la descripción del insumo
    dbHijo.lRecordMiembro = DESCINSUMOS;
    dbHijo.lCampoRegHijo = DESCINSUMOS_SZDESC;
    dbHijo.pszBufferMiembro = NULL;
    dbHijo.lIndicaOwner = 0;
    lConjunto = dbHijo.lSet;
    dbHijo.lSet = nConjuntoDescrip;

    dbData.fnSetBufferHijo ( &dbHijo, lConjunto );

    strStringMsg.LoadString ( IDST_SB_DESCONEC_ELE );
    fnCambiaStatusBar( strStringMsg );
    dbData.fnBorraMiembros ();

    strStringMsg.LoadString ( IDST_SB_BORRA_CLAVE );
    strStringMsg += szClave;
    fnCambiaStatusBar( strStringMsg );
    dbData.fnBorraRegistroPadre ();

    if ( dbData.fnStatus() == S_OKAY ) {
        OnLimpiar();
        m_lVecesChecallabilitar = 0;
    }

    endif //No limpiar campos en caso de error
}

endifunction

strStringMsg.LoadString ( AFX_IDS_IDLEMESSAGE );
fnCambiaStatusBar( strStringMsg );

endifunction

```

```

/*
FUNCION: fnPreparaBuscaData
DESCRIPCION:
Esta función se encarga de realizar las búsquedas que sean necesarias
tanto por clave como descripción. Si la búsqueda se hace por descrip-
ción y resulta exitosa en la variable iTipoBusqueda se deja la posición
en donde se encuentra la información buscada, mucho ojo, con eso
*/

```

```

int CFormatral::fnPreparaBuscaData (
    CSALDataBase & dbData,
    DBSUSUGEN & dbInsu,
    DBDESCINSU & dbDescrip,
    dbRecordPadre & dbPadre,
    dbRecordHijo & dbHijo,
    LPCSTR lpszClave,
    LPCSTR lpszSeedClan,
    int iTipoBusqueda,
    int nSubTipoBusqueda,
    BOOL bPresentaMsg
)

```

```

CMMGensAI *pMDIActiva;
int nStatus;
BOOL bBanderas = FALSE;
CString strDescripFrag,
strDescrip;
dAddrAux;
nConjuntoFam;

```

Esta función miembro es la clave para realizar cualquier tipo de búsqueda dentro de las formas de edición. Aquí se controla si la búsqueda es por Clave o búsqueda en descripción.

```

pMDIActiva = (CMMGensAI *) GetParentFrame (1);

// Se tiene que checar el tipo de búsqueda a realizar, ya que puede
// ser una búsqueda por descripción
// entendido
switch ( nSubTipoBusqueda )
{
    case IDC_INICIO:
    case IDC_FINAL:
    case IDC_NO_SABE_POS:
        // Busca la descripción que se le pidió
        pMDIActiva->m_ibanderas |= BAN_POR_DESC;
        break;

    case IDC_CLAVE_SIGUIENTE:
    case -1:
        pMDIActiva->m_ibanderas |= -BAN_POR_DESC;
        break;
}

if ( pMDIActiva )
{
    if ( IDC_BUSCA_CLAVE != nSubTipoBusqueda )
    {
        bBanderas = (BAN_POR_DESC & pMDIActiva->m_ibanderas );
    }
    else
    {
        bBanderas = FALSE;
    }
}

dbPadre.dbTask = DDfMioClanArea ( m_lpoBrainfo );
dbPadre.ubloqueo = ECLANCLANT.CONJUNTO;

```

```

m_Descrip = "";
if ( bBanderas )
{
    dbPadre.lCampo = m_lCampoBuscar;
    dbPadre.lRegActual = m_lRecordChecar;
    dbPadre.pszBuffer = (LPSTR) &dbInsu;
    dbData.fnSetBufferPadre ( &dbPadre );
    MarcaTexto.nIDControl = IDCE_DESCRIP;

    switch ( m_nTemplate )
    {
        case IDD_ANALISIS:
            dbHijo.lSet = ANALTSDESC;
            dbHijo.lRecordMiembro = DESCINSUMOS;
            dbHijo.lCampoRegHijo = DESCINSUMOS_TIPDESCINSUMOS;
            nConjuntoFam = FAMILANALIS;
            break;

        default:
            dbHijo.lSet = INSUBDESC;
            dbHijo.lRecordMiembro = DESCINSUMOS;
            dbHijo.lCampoRegHijo = DESCINSUMOS_TIPDESCINSUMOS;
            nConjuntoFam = FAMILINSU;
    }
}

dbHijo.pszBufferMiembro = (LPSTR) &dbDescrip;
dbHijo.iIndicaOwner = -1;

dbData.fnSetBufferHijo ( &dbHijo, -1 );

// Colocamos el apuntador de control de buffer hijo apuntando a
// Un hijo auxiliar
dbData.fnSetBool ( DB_BOOL_INDEXADO, TRUE );
dbDescrip.iTipo = MAKEINT ( m_szTipoData[0], m_szTipoData[1] );

if ( S_OKAY == fnBuscaDescrip ( dbData,
                                dbPadre,
                                dbHijo,
                                iTipoBusqueda,
                                nSubTipoBusqueda,
                                bPresentaMsg,
                                strDescripFrag,
                                lpszClave,
                                dbInsu.szClave,
                                dbDescrip,
                                strDescrip,
                                dbAddrAux ) )
{
    // efectivamente encontro información de cierta descripción,
    // lo que tenemos que hacer es que el padre se convierta en el
    // registro actual, es decir el insumo

    //Si hay datos de la clave indicada
    m_addrActual = dbAddrAux;
    m_ClaveBusqueda = (LPCTSTR) &dbDescrip.iTipo;

    dt setro ( dbHijo.lSet, dbPadre.dbTask, dbPadre.lNumDataBase);

    // bueno ya tenemos todo lo necesario para leer la información del
    // insumo.
    dbData.fnLeeRecord ( (LPSTR) &dbInsu );
    m_ClaveAnterior = dbInsu.szClave;

    dbPadre.lCampo = FAMILIA_S7SEUDONIMO;
    dbPadre.lRegActual = FAMILIA;
    dbPadre.pszBuffer = lpszSeudCrian;

    // Segundo la información sobre el registro hijo
    dbHijo.pszBufferMiembro = (LPSTR) &dbInsu;
    dbHijo.iIndicaOwner = -1;
}

```

Capítulo IV Desarrollo del Sistema

```
dbHijo.iRecordMiembro = m_iRecordChecar;
dbHijo.iCampoRegHijo = m_iCampoUscar;
dbHijo.iSet = nConjuntoFam;

if ( lpszSeudCBlan != (LPCSTR) NULL )
{
    nStatus = dt_ismember( dbHijo.iSet,
                          dbPadre.dbTask,
                          dbPadre.iNumDataBase );

    if ( nStatus == S_OKAY )
    {
        //Si esta conectado al conjunto, lo fijamos y leemos la información del
        //campo del padre que nos ha sido indicada
        nStatus = dt_setm( dbHijo.iSet,
                          dbPadre.dbTask,
                          dbPadre.iNumDataBase );

        if ( nStatus == S_OKAY )
        {
            nStatus = dt_csoread ( dbHijo.iSet,
                                  dbPadre.iCampo,
                                  lpszSeudCBlan,
                                  dbPadre.dbTask,
                                  dbPadre.iNumDataBase );

        }endif
    }
    else {
        if ( nStatus == S_EOS )
            nStatus = S_OKAY;
        endif
    }endif
}endif

}
else
{
    // Utilizamos la misma variable que se utiliza en la búsqueda de
    // descripciones, de esta forma ahorramos cierto espacio
    MarcaTexto.nIDControl = IDC_INSUMOS;
    wsprintf ( dbInsu.szClave,
              "Insu",
              (LPCSTR) m_szTipoDato,
              (LPCSTR) lpszClave );

    // Inicializamos la información sobre la base de datos SAICIC
    // Primero la información del registro Padre
    dbPadre.dbTask = DDFnMaObtTarea | m_lpObraInfo ;
    dbPadre.ulBloqueo = BLOQUEO_LECT_CONJUNTO;

    if ( m_nTemplate == IDD_ANALISIS )
    {
        dbHijo.iSet = FAMILIARIS;
    }
    else
    {
        dbHijo.iSet = FAMILIARIS;
    }
}
```

```

}
endif

dbPadre.lCampo = FAMILIA_SEUDONIMO;
dbPadre.iRegActual = FAMILIA;
dbPadre.pszBuffer = lpszSeudCBian;
dbData.fnSetBufferPadre ( &dbPadre );

// Colocamos el puntador de control de buffer hijo apuntando a
// dbDataMat.m_dbHijo
dbHijo.iRecordMiembro = m_iRecordChecar;
dbHijo.lCampoRegHijo = m_lCampoBuscar;
dbHijo.iSzBufferMiembro = (LPSTR) &dbInsu;
dbHijo.lIndicaOwner = -1;
dbData.fnSetBufferHijo ( &dbHijo, -1 );

dbData.fnSetBool ( DB_BOOL_INDEXADO, TRUE ); //Acceso indexado

// Debemos investigar si el usuario se encuentra en modo de búsqueda
// el -1 en SubTipoBusqueda nos indica si se encuentra en este modo
if ( -1 == nSubTipoBusqueda )
{
    strDescripFrag = lpszClave;
    strDescripFrag.MakeUpper ();
    nStatus = dbData.fnBuscarMemberConIndice (
        lpszSeudCBian, //El seudonimo es el que leeremos
        dbInsu.szClave, //Ultima clave accesada
        m_lClaveBusqueda, //Ultima clave accesada
        iTipoBusqueda, //Tipo de Busqueda
        lPresentaMsg,
        m_szTipoDato.GetLength () );

    if ( dbData.fnStatus () == S_OKAY )
    {
        // Empezamos a hacer el movimiento para realizar comparaciones
        strDescrip = dbInsu.szClave + TAM_TIPO;
        strDescrip.MakeUpper ();
        // Strip ( strDescripFrag, " " );
        iTipoBusqueda = strDescrip.Find ( strDescripFrag );

        // La descripción esta ok
        if ( 0 == iTipoBusqueda )
        {
            // Se encontro la palabra deseara dentro de la descripción
            // Obten el primer miembro
            MarcaTexto.nInicial = 0;
            MarcaTexto.nFinal = iStrlen ( lpszClave );
        }
    }
    else
        return S_NOTFOUND;
    endif
}
endif

}
else {
    strDescripFrag = dbInsu.szClave;
    strDescripFrag.MakeUpper ();
    nStatus = dbData.fnBuscarMemberConIndice (
        lpszSeudCBian, //El seudonimo es el que leeremos
        dbInsu.szClave, //Ultima clave accesada
        m_lClaveBusqueda, //Ultima clave accesada
        iTipoBusqueda, //Tipo de Busqueda
        lPresentaMsg,
        m_szTipoDato.GetLength () );

    // Chequemos si el usuario esta en búsqueda
    if ( IDC_CLAVE_SIN_USAR == nSubTipoBusqueda )

```

```

    {
        strDescrip = dbInsu.szClave;
        strDescrip.MakeUpper ();

        if ( 0 != strDescrip.Find ( strDescripTag ) )
        {
            return S_NOTFOUND;
        }

        }endif

    }endif

    MarcaTexto.nPosInicial = 0;
    MarcaTexto.nPosFinal = strlen ( lpszClave );

}endif

if ( S_OKAY == nStatus )
{
    //Si hay datos de la clave indicada
    // El registro actual obten su db_...
    m_ClaveAnterior = dbInsu.szClave;
    m_ClaveBusqueda = dbInsu.szClave;

}endif

}endif

return nStatus;

}endifunction

void CFormaGral::fnObtenDescrip ( CSAIDataBase & dbsData,
                                dbRecordPadre & dbPadre,
                                dbRecordHijo & dbHijo,
                                LPSTR lpszDescrip,
                                int iConjunteAnt )
{
    //Obtenemos la descripción
    // Llenamos información sobre la descripción

    if ( m_Descrip.IsEmpty () )
    {
        dbHijo.iRecordMiembro = DESCINSUMOS;
        dbHijo.iCampoRegHijo = DESCINSUMOS_SZDESC;
        dbHijo.pszBufferMiembro = lpszDescrip;
        dbHijo.iIndicaOwner = 00;
        dbHijo.iSet = INSUDESC;

        dbsData.fnSetBufferHijo ( &dbHijo, iConjunteAnt );
        dbsData.fnObtenDescripcion( m_Descrip, m_szClaveData );

    }endif

}endifunction

```



```

int CFormaGral::fnBuscaDescrip (
    CSAIDataBase & dbData,
    dbRecordPadre & dbPadre,
    dbRecordHijo & dbHijo,
    int iTipoBusqueda,
    int nSubTipoBusqueda,
    BOOL fPresentaMsg,
    CString & strDescripcion,
    LPCSTR lpszClave,
    LPSTR lpszAlamarenaPadre,
    DBSEDESCINSU & dbDescrip,
    CString & strDescrip,
    DB_ADDR & dbAddrAux )
{
    int nIndice,
        nLongitud;

    switch ( nSubTipoBusqueda )
    {
        case IDC_FINAL: {
            strDescripFrag = lpszClave;
            strDescripFrag.MakeReverse ();
            strDescripFrag.MakeUpper ();

            dbData.fnBuscarMemberConIndiceEX (
                lpszAlamarenaPadre,
                (LPSTR) &dbDescrip.ITipo,
                m.ClaveBusqueda,
                m.addrActual,
                iTipoBusqueda,
                fPresentaMsg );

            while ( dbData.fnStatus () == S_OKAY )
            {
                // OK tenemos la primera descripción de este tipo
                // Ahora se obtiene el db_addr del elemento que es el índice
                // de la descripción
                dbGet ( &dbAddrAux,
                    dbPadre.dbTask,
                    dbPadre.lNumDataBase );

                dbData.fnObtenDescripcion( strDescrip, m.nTipoDato );
                m.Descrip = strDescrip;
                strDescrip.MakeUpper ();
                strDescrip.MakeReverse ();

                // la descripción esta ok
                nIndice = strDescrip.Find ( strDescripFrag );

                if ( nIndice == 0 )
                {
                    // Se encontro la palabra desuada dentro de la descripción
                    // Obten el primer miembro, OK señoras y señores lo encontramos
                    // Ojo amigos, la longitud de la cadena nos indica en que pos
                    // ción, se lo restas a la longitud de la
                    // descripción y asunto arreglado, a partir de esa posición
                    // es donde tienes que marcar el texto
                    nLongitud = strDescrip.GetLength();

                    MarcaTexto.nPosInicial =
                        nLongitud - strDescripFrag.GetLength ();
                    MarcaTexto.nPosFinal = nLongitud;
                    iTipoBusqueda = SIGUIENTE; // así continuamos con la
                                                // búsqueda
                    dbData.fnFindFirstMember( TRUE );
                    break;
                }
            }
        }
        else
    }
}

```

Este método se encarga de buscar palabras dentro de las descripciones algo que al usuario le es sumamente funcional

Capítulo IV Desarrollo del Sistema

```
        m_Descrip = "";
    endif

    dbData.fnBuscarMemberConIndice (
        lpszAlamarenaPadre,
        (LPSTR) &dbDescrip.iTipo,
        SIGUIENTE,
        bPresentaMsg,
        m_szTipoDato.GetLength() );

}endwhile

break;
}endblock

case IDC_NO_SABE_PCS:

    // No conoce el usuario la posición de la clave, la empezamos
    // a buscar, secuencialmente
    strDescripFrag = lpszClave;
    strDescripFrag.MakeUpper ();

    dbData.fnBuscarMemberConIndiceEX (
        lpszAlamarenaPadre,
        (LPSTR) &dbDescrip.iTipo,
        m_ClaveBusqueda,
        m_addrActual,
        iTipoBusqueda,
        bPresentaMsg,
        m_szTipoDato.GetLength() );

    while ( dbData.fnStatus () == S_OKAY )
    {

        dt_orget ( &RAddrAux,
            dtFadre.dbTask,
            dtFadre.iNumDataBase );
        dbData.fnObtenDescripcion( strDescrip, m_szTipoDato );
        m_Descrip = strDescrip;
        strDescrip.MakeUpper ();
        nindice = strDescrip.Find ( strDescripFrag );

        // La descripción esta ok
        if ( nindice >= 0 )
        {

            // Se encontro la palabra deseada dentro de la descripción
            // Obten el primer miembro
            dbData.fnInsFirstMember( TRUE );
            MarcaTexto.r5.Inicial = nindice;
            MarcaTexto.r5.PosFinal = nindice +
                strDescripFrag.GetLength ();
            iTipoBusqueda = SIGUIENTE;
            break;
        }
        else

            m_Descrip = "";
    }
    endif

    dbData.fnBuscarMemberConIndice (
        lpszAlamarenaPadre,
        (LPSTR) &dbDescrip.iTipo,
        SIGUIENTE,
        bPresentaMsg,
        m_szTipoDato.GetLength() );

}endwhile

break;
```

```

case IDC_INICIO:
    // Lista la descripción a buscar, ahora a llenar la información que
    // es necesaria
    // Inicializamos la información sobre la base de datos SAICIC
    // Primero la información del registro Padre
    strDescripFrag = lpszClave;
    strDescripFrag.MakeUpper ();
    lstrcpy ( dbDescrip.szDesc, lpszClave );
    dbData.fnBuscarMemberConIndiceEX (
        lpszAlamacaPadre,
        (LPSTR) &dbDescrip.iTipo, // Descripción a buscar
        m_ClaveBusqueda, // Última clave accedida
        m_addrActual,
        iTipoBusqueda, // Tipo de Búsqueda

        bPresentaMsg,
        m_szTipoData.GetLength () );

    if ( dbData.fnStatus () == S_OKAY )
    {
        dt_crget ( &dbAddrAux,
            dbPadre.dbTask,
            dbPadre.iNumDataBase );
        m_Descrip = strDescrip = dbDescrip.szDesc;
        strDescrip.MakeUpper ();
        nIndice = strDescrip.Find ( strDescripFrag );

        // La descripción esta ok
        if ( 0 == nIndice )
        {
            // Se encontro la palabra deseada dentro de la descripción
            // Obten el primer miembro
            iTipoBusqueda = SIGUIENTE;
            MarcaTexto.nPosInicial = 0;
            MarcaTexto.nPosFinal = strDescripFrag.GetLength ();
            break;
        }
        else
            return S_NOTFOUND;
    }
    endif

break;

default :
    dbDescrip.iTipo = MAKEINT ( m_szTipoData[0], m_szTipoData[1] );
    lstrcpy ( dbDescrip.szDesc, lpszClave );
    // Lista la descripción a buscar, ahora a llenar la información que
    // es necesaria
    // Inicializamos la información sobre la base de datos SAICIC
    // Primero la información del registro Padre
    dbData.fnBuscarMemberConIndiceEX (
        lpszAlamacaPadre, // Es la clave del Insumo
        (LPSTR) &dbDescrip.iTipo, // Descripción a buscar
        m_ClaveBusqueda, // Última clave accedida
        m_addrActual,
        iTipoBusqueda, // Tipo de Búsqueda
        bPresentaMsg,
        m_szTipoData.GetLength () );

    dt_crget ( &dbAddrAux,
        dbPadre.dbTask,
        dbPadre.iNumDataBase );

endswitch

return dbData.fnStatus ();

```

Capítulo IV Desarrollo del Sistema

```
len:function
void CFormaGral::OnPrimerRegistro()
{
    int nTipoBusqueda = PRIMERO,
        nSubTipoBusqueda = 0;

    if ( ! m_Spread.GetEditMode() )
    {
        m_Spread.SendMessage ( WM_KEYDOWN, VK_RETURN, 1 );
        m_Spread.SendMessage ( WM_KEYUP, VK_RETURN, 1 );
    }

    }endif

    fnCargaDatos( "", nTipoBusqueda, nSubTipoBusqueda );
}endifunction

void CFormaGral::OnUltimoRegistro()
{
    int nTipoBusqueda = ULTIMO,
        nSubTipoBusqueda = 0;

    if ( ! m_Spread.GetEditMode() )
    {
        m_Spread.SendMessage ( WM_KEYDOWN, VK_RETURN, 1 );
        m_Spread.SendMessage ( WM_KEYUP, VK_RETURN, 1 );
    }

    }endif

    fnCargaDatos( "", nTipoBusqueda, nSubTipoBusqueda );
}endifunction

void CFormaGral::OnRegistroAnterior()
{
    int nTipoBusqueda = ANTERIOR,
        nSubTipoBusqueda = 0;

    if ( ! m_Spread.GetEditMode() )
    {
        m_Spread.SendMessage ( WM_KEYDOWN, VK_RETURN, 1 );
        m_Spread.SendMessage ( WM_KEYUP, VK_RETURN, 1 );
    }

    }endif

    fnCargaDatos( "", nTipoBusqueda, nSubTipoBusqueda );
}endifunction

void CFormaGral::OnRegistroPosterior()
{
    int nTipoBusqueda = SIGUIENTE,
        nSubTipoBusqueda = 0;

    if ( ! m_Spread.GetEditMode() )
    {
        m_Spread.SendMessage ( WM_KEYDOWN, VK_RETURN, 1 );
        m_Spread.SendMessage ( WM_KEYUP, VK_RETURN, 1 );
    }

    }endif

    fnCargaDatos( "", nTipoBusqueda, nSubTipoBusqueda );
}endifunction
```

```

void CFormaGral::OnMuestraClaveElegidaEnCat (
    LPCSTR    lpszBuscar,
    INT       nTipoBusqueda,
    UINT      nTemplate,
    UINT      nFormaAnalisis )
{
    CString    strClave;
    POSITION     pos;
    CInfoMDI   *pInfoDocu;
    CFormaGral *pFormaGral;
    int        nSubTipoBusqueda = 0;

    if ( nTemplate == m_nTemplate )
    {
        if ( IDD_ANALISIS != nTemplate )
        {
            InCargaDatos ( lpszBuscar, nTipoBusqueda, nSubTipoBusqueda );
            return;
        }
    }

    // Antes de intentar crear una nueva ventana de cierto tipo, tenemos
    // que checar si ya existe alguna ventana del tipo abierta dentro
    // de las opciones activas de la obra
    for ( pos = m_lpObraInfo->ObjListaVent.GetHeadPosition();
          pos != NULL; )
    {
        pInfoDocu = ( CInfoMDI * ) m_lpObraInfo->ObjListaVent.GetNext( pos );

        // Bueno averigua que elemento tiene como activo
        pFormaGral = ( CFormaGral * )
            ( ( CHDICHildWnd * ) pInfoDocu->m_pMDIWnd )->GetActiveView ();

        if ( pFormaGral->m_nTemplate == nTemplate )
        {
            if ( pFormaGral->m_nTemplate == IDD_ANALISIS )
            {
                if ( ( ( CAnalysisView * ) pFormaGral )->m_nTipoDlgDinamico
                    == nFormaAnalisis )
                {
                    pInfoDocu->m_pMDIWnd->BringWindowToTop ();
                    pFormaGral->OnMuestraClaveElegidaEnCat (
                        lpszBuscar,
                        nTipoBusqueda,
                        nTemplate,
                        nFormaAnalisis );
                    pwndCatClaves->BringWindowToTop ();
                    return;
                }
            }
            else
            {
                pInfoDocu->m_pMDIWnd->BringWindowToTop ();
                pFormaGral->OnMuestraClaveElegidaEnCat (
                    lpszBuscar,
                    nTipoBusqueda,
                    nTemplate,
                    nFormaAnalisis );
                pwndCatClaves->BringWindowToTop ();
                return;
            }
        }
    }
}

```

Capítulo IV Desarrollo del Sistema

```
        }endif
    }endifor

    // lamentablemente no son del mismo tipo, por lo cual tenemos que
    // crear una nueva ventana
    theApp.m_strClaveMostrar = lpzBuscar;

    switch ( nTemplate )
    {
        case IDD_MATERIALES:
        case IDD_OBRA_MANO:
        case IDD_SUBCONTRATO:
        case IDD_MQUINARIA:

            theApp.m_iTipoForma = nTemplate;
            theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_INSUMOS,
                m_lpObraInfo );
            pwndCatClaves->PringWindowToTop ();
            break;

        case IDD_ANALISIS:

            theApp.m_iTipoForma = nFormaAnálisis;
            theApp.m_pPanelCentral->fnAgrRelOpciones( TEMPLATE_ANALISIS,
                m_lpObraInfo );
            pwndCatClaves->PringWindowToTop ();
            break;

        case IDD_FAMILIAS:
            theApp.m_iTipoForma = IDD_FAMILIAS;
            theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_INSUMOS,
                m_lpObraInfo );
            break;

        case IDD_ZONAS:
            theApp.m_iTipoForma = IDD_ZONAS;
            theApp.m_pPanelCentral->fnAgrRelOpciones ( TEMPLATE_INSUMOS,
                m_lpObraInfo );
            break;
    }endswitch
}endifunction

BOOL CFormaGral::OnSigueBusqueda ( LPCSTR    lpzCadenaBuscar,
                                  int        nTipoBusqueda,
                                  int        nSubtipoBusqueda,
                                  BOOL       bRecordar )
{
    CString strClave;
    BOOL    bIntentadoNuevo;
    CEdit * pWndEdit;

    pWndEdit = (CEdit *) GetDlgItem ( MarcaTexto.nIDControl );

    // Que tipo de sub-busqueda se tiene
    if ( -1 == nSubtipoBusqueda || 0 == nSubtipoBusqueda )
    {
        // Se esta buscando por clave, envia lo siguiente
        nSubtipoBusqueda = IDD_CLAVE_SIGUIENTE;
    }endif

    if ( S_OKAY == fnCargaDatos ( lpzCadenaBuscar,
                                nTipoBusqueda,
                                nSubtipoBusqueda,
                                FALSE ) )
    {

```

```

// Checa si se trataba de búsquedas por descripción para marcar
// el texto

pWndEdit->SetSel ( MarcaTexto.nPosInicial, MarcaTexto.nPosFinal );
bIntentadeNuevo = FALSE;

}
else
{
    bIntentadeNuevo = TRUE;

    if ( -1 != nSubtipoBusqueda && 0 != nSubtipoBusqueda )
    {
        m_addrActual = NULL_DPA;
    }
    }endif

pWndEdit->SetSel ( -1, 0 );
MarcaTexto.nPosInicial = MarcaTexto.nPosFinal = 0;

if ( IDC_CLAVE_SIGUIENTE == nSubtipoBusqueda )
{
    // La búsqueda es por clave, por favor interpreta
    nSubtipoBusqueda = -1;
    nTipoBusqueda = BUSE APROX;
}
}endif

}endif

return bIntentadeNuevo;
}endifunction

void CFormaGrat::OnBuscarClave()
{
    ASSERT ( m_ulIDControlChecar != 0 ); //Se tiene que inicializar esta variable
                                        //con la constante del campo a checar

    CPusClave dlgBuscaClave; //Diálogo de Búsqueda de claves
    CWnd * pWndclave = GetDlgItem ( IDC_BUSCAMOS );

    pWndclave->GetWindowText ( dlgBuscaClave.m_Clave );
    dlgBuscaClave.m_nTemplate = m_nTemplate;
    CMDIGenSAI *pMDIActiva;

    pMDIActiva = (CMDIGenSAI *) GetParentFrame ();
    dlgBuscaClave.m_szTipoData = m_szTipoData;

    if ( dlgBuscaClave.DoModal() == IDOK )
    {
        // Chequemos de que tipo de búsqueda se trata, primero tenemos que
        // limpiar la variable que indica que ya se ha hecho una búsqueda
        // anteriormente por descripción, esto por omiquitos
        if ( BUSCAR_DESCRIP != dlgBuscaClave.m_nTipoDeBusqueda )
        {
            // Desde ahora búsquedas por clave
            pMDIActiva->m_fBanderaB = -BANDER FOR_DESC;
        }
    }
    else
    {
        m_addrActual = NULL_DPA;
    }
}endif

if ( ! OKAY == InCargados ( dlgBuscaClave.m_Clave,
                            dlgBuscaClave.m_nTipoBusqueda,

```

```

        dlgBuscaClave.m.nSubTipoBusqueda ) }
    {
        pWndclave = GetDlgItem ( MarcaTexto.nIDControl );
        ( ( CEdit * ) pWndclave )->SetSel (
            MarcaTexto.nPosInicial, MarcaTexto.nPosFinal );
    }
    else
    {
        MarcaTexto.nPosInicial = MarcaTexto.nPosFinal = 0;
    }
    endif

    if ( !pwndBusqueda )
    {
        TRY
        {
            pwndBusqueda = new CWndBusqueda;
        }
        CATCH ( CMemoryException, e )
        {
            DdFnDappError ( m.mWnd,
                hLibFunc,
                theApp.m.mDdgnario,
                IDST_ERR_NO_MEM );

            pWndclave->SetFocus ();

            return;
        }
        }END_CATCH

    }endif

    pwndBusqueda->m_SigueRrr.m.strCadena = dlgBuscaClave.m.Clave;
    pwndBusqueda->m_nTipoBusqueda = dlgBuscaClave.m.nTipoBusqueda;
    pwndBusqueda->m_nSubTipoBusqueda = dlgBuscaClave.m.nSubTipoBusqueda;

    if ( !IsWindow ( pwndBusqueda->GetSafeHwnd () ) )
    {
        // Crea la nueva ventana de búsqueda
        if ( !pwndBusqueda->Create ( IDS_DIALOG_BUSQUEDA, this ) )
            return;

        }endif

        pwndBusqueda->m_bInicio = FALSE;
        pwndBusqueda->m_piquebar.GetDlgItem ( IDC_CAFENA_BUSCAR )->SetWindowText (
            dlgBuscaClave.m.Clave );
        pwndBusqueda->m_bInicio = TRUE;

    }
    else
    {
        // bueno estimado amigo, presentamos a la barra de búsquedas
        pwndBusqueda->ShowWindow ( SW_SHOW );
        pwndBusqueda->UpdateWindow ();
        pwndBusqueda->m_bInicio = FALSE;
        pwndBusqueda->m_piquebar.GetDlgItem ( IDC_CAFENA_BUSCAR )->SetWindowText (
            dlgBuscaClave.m.Clave );
        pwndBusqueda->m_bInicio = TRUE;

    }
    }endif
}

```


Capítulo IV Desarrollo del Sistema

```
        m_iVecesChecaHabilitar++;
    }
    else
        m_iVecesChecaHabilitar = 0;
    endif

    switch ( pCmdUI->m_nID )
    {
        case ID_RECUERDA_CLAVE:
        case ID_BORRAR:
            ::GetWindowText ( ::GetDlgItem ( m_hWnd, IDC_INSUMOS ),
                szClave,
                sizeof INSGEN_SZCVFORLAN );
            StrLTrim ( szClave );
            pCmdUI->Enable ( m_bHabilitar && (BOOL) *szClave );
            break;

        default:
            pCmdUI->Enable ( m_bHabilitar );
    }

}endswitch

break;

}endswitch

}endfunction

BOOL CFormaGral::OnCommand( UINT wParam, LPARAM lParam )
{
    BOOL bRet = CFormView::OnCommand( wParam, lParam );

    if ( !IsWindow ( m_Spread.m_hWnd ) )
    {
        m_Spread.InChecaEstatusKey ();
    }

}endif

if ( !bRet ) {
    UINT nID = wParam;

    if ( ID_CAT_MONEDA == nID )
    {
        CCatMoneda dlgCatMoneda;
        dlgCatMoneda.DoModal();

        // Levanta al catálogo de moneda
        return TRUE;
    }

}endif

// ¿Es un comando ?
if ( ! nID == 0 || ( LOWORD( lParam ) != 0 ) ) {
    // Los ids de menús no pueden ser cero
    // También, los menús tienen lowords iguales a cero
    return FALSE;
}
else {
    if ( nID >= ID_FIRST_MENU_DYN &&
        nID <= ID_LAST_MENU_DYN )
    {
    }
}
}
```

```

if ( !fnGetBool ( FM_MENU_ACTIVO ) ) {
    fnSetBool ( FM_MENU_ACTIVO, FALSE );
    DDInMacCmdDevDir( m_hWnd, m_lpObrainfo, TRUE );
    fnProcesaPopupCommand ( nID - (UIST)ID_FIRST_MENU_DYN );
    return TRUE;
}endif

}endif

}endif

return bRet;
}endiffunction

void CFormaGral::OnUpdatePaste (CCmdUI* pCmdUI)
{
    CClipboard::UpdatePaste ( this, pCmdUI );
}endiffunction

void CFormaGral::fnReallyLimpia ()
{
    m_Spread.fnLimpiaSpread();
    fnLimpiaCampos ( m_hWnd );

    m_CostoUnitario = 0.0;
    m_Descrip = "";
    m_Familia = "";
    m_Unidad = "";
    m_ClaveAnterior = ""; //Limpiamos la clave anterior

    switch ( m_szTipoDato { 0 } )
    {
        case TIPO_BASICO :
            m_Clave = oConfigura.m_szBas;
            // Ahora con las fechas que el usuario hay puesto como default
            GetDlgItem ( IDC_Fecha_COSTO )->SetWindowText ( oConfigura.m_szFechaBas );
            GetDlgItem ( IDC_Fecha_CAL )->SetWindowText ( oConfigura.m_szFechaCalBas );
            break;

        case TIPO_CAPITULO :
            m_Clave = oConfigura.m_szCap;
            GetDlgItem ( IDC_Fecha_COSTO )->SetWindowText ( oConfigura.m_szFechaCap );
            GetDlgItem ( IDC_Fecha_CAL )->SetWindowText ( oConfigura.m_szFechaCalCap );
            break;

        case TIPO_INTEGRADO :
            m_Clave = oConfigura.m_szInt;
            GetDlgItem ( IDC_Fecha_COSTO )->SetWindowText ( oConfigura.m_szFechaInt );
            GetDlgItem ( IDC_Fecha_CAL )->SetWindowText ( oConfigura.m_szFechaCalInt );
            break;

        case TIPO_MAT :
            m_Clave = oConfigura.m_szMat;
            GetDlgItem ( IDC_Fecha_COSTO )->SetWindowText ( oConfigura.m_szFechaMat );
            GetDlgItem ( IDC_Fecha_CAL )->SetWindowText ( oConfigura.m_szFechaCalMat );
            GetDlgItem ( IDC_COSTO )->SetWindowText ( "" );
            break;

        case TIPO_ORRAMAHO :
            m_Clave = oConfigura.m_szObr;
            GetDlgItem ( IDC_Fecha_COSTO )->SetWindowText ( oConfigura.m_szFechaObr );
            GetDlgItem ( IDC_Fecha_CAL )->SetWindowText ( oConfigura.m_szFechaCalObr );
    }
}

```



```

}endif

m_bCambios = FALSE;

}endifunction

void CFormaGral::OnLimpiar()
{
    fnReallyLimpiar ( );
    UpdateData ( FALSE );
    m_bCambios = FALSE;

    CWnd *pWndCosto;

    if ( NULL != ( pWndCosto = GetDlgItem ( IDC_SCOSTO ) ) )
    {
        pWndCosto->SetWindowText ( "" );
    }

}endif

}endifunction

BOOL CFormaGral::fnHayPrecioSel ( )
{
    if ( m_Spread.GetEditMode() )
    {
        m_Spread.SendMessage ( WM_KEYDOWN, VK_RETURN, 1L );
        m_Spread.SendMessage ( WM_KEYUP, VK_RETURN, 1 );
    }

}endif

if ( m_Spread.GetDataCnt ( ) )
{
    if ( 0 == m_Spread.m_lRowBotonSel )
    {
        if ( 1 == m_Spread.m_lNumRenglon )
        {
            m_Spread.m_lRowBotonSel = 1;
        }
        else
        {
            // No existe botón seleccionado, solicitamos una acción al usuario
            CMensajePrecio dlgAccion;

            dlgAccion.m_nTipoMensaje = IDS_NO_HAY_PRECIO;
            dlgAccion.m_strMensaje.LoadString ( IDS_NO_HAY_PRECIO );
            dlgAccion.m_strRadio1.LoadString ( IDS_QUEJESA_CAPTURA );
            dlgAccion.m_strRadio2.LoadString ( IDS_PRECIO_DEFAULT );

            if ( IDOK == dlgAccion.DoModal ( ) )
            {
                if ( IDC_BTN1 == dlgAccion.m_nTipoAccion ) {
                    return FALSE; // Quiere continuar con la captura de información
                }

            }

            m_Spread.m_lRowBotonSel = 1;
        }
    }
    else
    {
        return FALSE;
    }
}

```

```

        }
        }
    }
}

void CFormaGral::OnSetFocusSpread ()
{
    m_Spread.SendMessage ( WM_KEYDOWN, VK_SHIFT, 0 );
    m_Spread.SendMessage ( WM_KEYUP, VK_SHIFT, 0 );
}

void CFormaGral::OnRecuerdaClave ()
{
    // OK tenemos que recordar las claves, espero que sea aquí donde
    // tenemos que veer
    // OK obtén la clave y agréjala a la lista
    CString strClave,
            strTipo;

    strTipo = m_szTipoData;
    GetDlgItem ( IDC_INSUMOS )->GetWindowText ( strClave );
    strTipo += strClave;

    // Dentro del siguiente procedimiento verificamos que la clave no
    // este repitiendose
    for ( int nCont = m_ipObrainfo->m_StrClavesRem.GetSize () - 1;
          nCont >= 0;
          nCont-- )
    {
        strClave = m_ipObrainfo->m_StrClavesRem [ nCont ];

        if ( !strClave.CompareNoCase ( strTipo ) )
        {
            // Son iguales, adica, no agregar
            return;
        }
    }

    m_ipObrainfo->m_StrClavesRem.Add ( strTipo );
}

void CFormaGral::OnListaConRecuerdos ()
{
    CBookmark dlgBookmark;
    dlgBookmark.m_ipMDITref = m_ipObrainfo;
    if ( !IDOK == dlgBookmark.DoModal ( ) )

```


Capítulo IV Desarrollo del Sistema

```
case IDC_FAMILIA:
    // Desea el catálogo de familias
    pwndCatClaves->m_eTipoCatalogo = CLAVES;
    pwndCatClaves->m_eTipoUso = CLAVE_REGRESAR;
    pwndCatClaves->m_strTipoDato = IDENT_FAMILIAS;
    pwndCatClaves->m_nTemplate = IDD_FAMILIAS;
    break;

case IDCSP_PRECIOS:
{
    SS_COORD ssRen, ssCol;

    m_Spread.GetActiveCell ( &ssCol, &ssRen );

    // Who are you?
    switch ( m_nTemplate )
    {
        case IDD_ANALISIS:
            pwndCatClaves->m_eTipoCatalogo = TODAS_DESCRIPCIONES;
            pwndCatClaves->m_eTipoUso = CLAVE_REGRESAR;
            pwndCatClaves->m_strTipoDato = " ";
            pwndCatClaves->m_nTemplate = m_nTemplate;
            break;

        default:
            // chequea si la columna activa es el de la zona
            // Abre el catálogo por favor de Zonas
            pwndCatClaves->m_strTipoDato = IDENT_ZONAS;
            pwndCatClaves->m_eTipoCatalogo = CLAVES;
            pwndCatClaves->m_eTipoUso = CLAVE_REGRESAR;
            pwndCatClaves->m_nTemplate = IDD_ZONAS;
            break;
    }

    }endswitch

    break;

}endblock

default:

    // Desea el catálogo de este tipo de Elemento
    pwndCatClaves->m_eTipoCatalogo = CLAVES;
    pwndCatClaves->m_eTipoUso = CLAVE_ABRIR;
    pwndCatClaves->m_strTipoDato = m_szTipoDato;
    pwndCatClaves->m_nTemplate = m_nTemplate;
    break;

}endswitch

CString strDirectorio;

strDirectorio = TIT_CAT;
strDirectorio += '\\';
strDirectorio += m_lpObraInfo->szDrive;
strDirectorio += m_lpObraInfo->szDir;
strDirectorio += '\\';

if ( !IsWindow ( pwndCatClaves->GetSafeHwnd () ) )
{

    if ( (pwndCatClaves->Create ( strDirectorio, pWndPadre ) )
    {

        return;

    }endif

}
else
{
```



```

pwndCatClaves->SetWindowText ( strDirectorio );
pwndCatClaves->fnCambiaTipo ();

}endif

pwndCatClaves->m_pWndPadre = pWndPadre;
pwndCatClaves->ShowWindow ( SW_NORMAL );
pwndCatClaves->UpdateWindow ();

endifunction

void CFormaGral::OnOK ()

{
    CWnd *pEditInsumos = GetDlgItem ( IDC_INSUMOS );
    CMDIGenSAI *pMDIActiva;
    int nBanderas;
    int nTipoBusqueda = BUSC_APROX,
        nSubTipoBusqueda = 0,
        nNumero;
    CWnd *pWndFocus = GetFocus ();

    pMDIActiva = (CMDIGenSAI *) GetParentFrame ();

    // Busca información que ha solicitado el usuario por favor
    if ( pWndFocus == pEditInsumos )
    {
        //nBanderas = pMDIActiva->m_iBanderas;
        pEditInsumos->GetWindowText ( m_Clave );

        // Tenemos que ver si se trata de un numero
        nNumero = StringToInt ( (LPCTSTR) (LPCSTR) m_Clave );

        if ( !oConfigura.m_ClavesNumericas || nNumero <= 9999 || nNumero > 0 )
        {
            // Se trata de un numero por favor rellenalos con zeros
            wsprintf ( szClave, "%04i", nNumero );
            m_Clave = szClave;
        }

        }endif

        nSubTipoBusqueda = IDC_BUSCA_CLAVE;
        fnCargaDatos( m_Clave, nTipoBusqueda, nSubTipoBusqueda );

        //pMDIActiva->m_iBanderas = nBanderas;
    }
    else
    {
        pEditInsumos = GetDlgItem ( IDCE_DESCRIP );

        if ( pWndFocus == pEditInsumos )
        {
            // El campo que tiene el focus es la descripción por favor
            // busca dicha descripción
            nBanderas = pMDIActiva->m_iBanderas;
            nSubTipoBusqueda = IDC_NO_SABE_POS;
            pEditInsumos->GetWindowText ( m_Clave );

            fnCargaDatos( m_Clave, nTipoBusqueda, nSubTipoBusqueda );

            pMDIActiva->m_iBanderas = nBanderas;
        }
        }endif
    }endif

endifunction

void CFormaGral::OnCancel ()

```

Capítulo IV Desarrollo del Sistema

```
(
    if ( fnGetBool ( FM_ELIGE_INFO ) )
    {
        //Valida la forma el usuario desea salir de este estatus
        fnValidaForma ();
        fnSetBool ( FM_ELIGE_INFO, FALSE );
    }
}
}endif
}endifunction
void CFormaGral::OnUpdateCut ( CCmdUI * pCmdUI )
{
    if ( ::GetFocus () == m_Spread.GetSafeHwnd () )
    {
        pCmdUI->Enable ( FALSE );
    }
    else
    {
        pCmdUI->Enable ( TRUE );
    }
}
}endif
}endifunction
void CFormaGral::OnSpreadCut ()
{
    // Envía cut al spread
    m_Spread.SendMessage ( WM_COMMAND, ID_SPREAD_CUT, 0L );
}
}endifunction
void CFormaGral::OnSpreadCopy ()
{
    // Envía cut al spread
    m_Spread.SendMessage ( WM_COMMAND, ID_SPREAD_COPY, 0L );
}
}endifunction
void CFormaGral::OnSpreadPaste ()
{
    // Envía cut al spread
    m_Spread.SendMessage ( WM_COMMAND, ID_SPREAD_PASTE, 0L );
}
}endifunction
BOOL CFormaGral::OnDrop(
    COleDataObject* pDataObject,
    DROPEFFECT dropEffect,
    CPoint point)
{
    ASSERT_VALID(this);
    BOOL bRegreso = FALSE;

    // Existe una condición muy especial en el caso de arrastrar y
    // soltar, esta ventana en la cual se está haciendo el drop no se
    // encuentra activa, por ende no nos hemos cambiado al directorio origen
    // de la obra, que problemas representa pues el hecho de que estamos
    // tratando de leer la información sobre la obra pero no
    // estamos físicamente posicionados en el directorio de la misma,
    // Craso error, pero tiene solución
    CD(m_mCamDir(m_hWnd, m_lpObraInfo, TRUE );
```

Bloque de funciones que
permite habilitar el
manejo de drag and drop
dentro de las formas de
edición.

```

// Llevamos a cabo el paste de la información
// Tenemos que tipo de información se encuentra dentro del
// clipboard
if ( dropEffect != DROPEFFECT_LINK && DROPEFFECT_NONE != dropEffect ) (
    if ( fnGetBool ( EM_EMPTY_INFO ) )
    {
        OnPaste ();
        bRegreso = TRUE;
    }
    else (
        fnObtenControl ( point ); // Siempre regresa m_hwndHijo
        fnPasteSpecial ( m_hwndHijo, point );
        bRegreso = TRUE;
    )
}endif

( ( CMDIFrameWnd * ) theApp.m_pMainWnd )->MDIActivate ( GetParent ( ) );

return bRegreso;
}endifunction

DROPEFFECT CFormaGral::OnDragEnter(
    COleDataObject* pDataObject,
    DWORD grfKeyState,
    CPoint point)
{
    BringWindowToTop ();

    return OnDragOver( pDataObject, grfKeyState, point );
}

DROPEFFECT CFormaGral::OnDragOver(
    COleDataObject* pDataObject,
    DWORD grfKeyState,
    CPoint point )
{
    static DROPEFFECT dropEffect;
    static HWND hwndHijo= NULL;
    static CRect rect;

    // Checamos si el punto cae dentro de alguna de las ventanas que pueden
    // utilizarse para pasarles información
    dropEffect = DROPEFFECT_NONE;
    hwndHijo = fnObtenControl ( point );

    if ( pDataObject->IsDataAvailable( CF_TEXT ) ) (
        if ( hwndHijo ) (
            ::GetWindowRect ( hwndHijo, (LPRECT) rect );
            ScreenToClient ( rect );

            if ( !rect.PtInRect ( point ) ) (
                hwndHijo = NULL;
            )endif
        )endif
    )endif

    if ( hwndHijo ) (
        // 1. Verificamos que existan datos disponibles en el formato texto
        // y en el formato privado, puede tratarse del apread
    )
}

```

Capítulo IV Desarrollo del Sistema

```
if ( ( grfKeyState & ( MK_CONTROL | MK_SHIFT ) )
    ==
    ( MK_CONTROL | MK_SHIFT ) ) {
    dropEffect = DROPEFFECT_COPY;
// 2. Checamos si forzamos un copy
}
else {
    if ( ( grfKeyState & MK_CONTROL ) == MK_CONTROL )
        dropEffect = DROPEFFECT_COPY;
// 3. Checamos por si forzamos un cut
else {
    if ( ( grfKeyState & MK_ALT ) == MK_ALT ) {
        // Checa sino es analisis
        if ( IDD_ANALISIS != m_nTemplate )
            dropEffect = DROPEFFECT_MOVE;
        endif
    }
    else {
        if ( IDD_ANALISIS != m_nTemplate )
            dropEffect = DROPEFFECT_MOVE;
        endif
    }
}
endif
endif
endif
endif
endif
endif
endif
// 4. Regresamos el valor correspondiente a lo solicitado
return dropEffect;
}
endifunction

LRESULT CFormaGral::OnDragSpread ( WPARAM wParam, LPARAM lParam )
{
    // Tenemos ya la información sobre el spread pasemos a registrarla
    COleDataSource *pDataInsu = NULL; // Creamos nuestro data source para llevar
    // a efecto el copiado al clipboard
    RECT rc;
    HGLOBAL hData = (HGLOBAL)LOWORD ( lParam );
    DROPEFFECT dropResult;
    // Creamos la información necesaria para el COleDataSource
    pDataInsu = new COleDataSource;
    pDataInsu->CacheGlobalData ( CF_TEXT, hData );
    // 5. Colocamos la información dentro del clipboard
    pDataInsu->SetClipboard ();
    dropResult = pDataInsu->DoDragDrop(
        DROPEFFECT_COPY | DROPEFFECT_MOVE,
        0 );
    return MAKELPARAM ( dropResult, 0 );
}
```

```

ndfunction

ID CFormaGral::fnObtenControl ( CPoint point )

static CRect rect;

for ( m_hwndHijo = ::GetWindow ( m_hWnd, GW_CHILD );
      m_hwndHijo;
      m_hwndHijo = ::GetWindow ( m_hwndHijo, GW_HWNDNEXT ) )
{
    // OK tenemos ya un control ahora chequeemos si a ese control esta
    // dentro de la lista de controles a los que se le puede copiar info
    if ( fnEsControlCopiar ( m_hwndHijo ) )
    {
        // Es de los controles a copiarle info chequeemos si tiene el punto
        // indicado por poin
        // indicado por poin
        ::GetWindowRect ( m_hwndHijo, (LPRECT) rect );
        ScreenToClient ( rect );

        if ( rect.PtInRect ( point ) ) {
            // el punto se encuentra dentro del control
            break;
        }
    }
}

return m_hwndHijo;
ndfunction

```

Bloque de funciones que permiten el cortar, copiar, pegar y drag and drop dentro de las formas de edición de datos

```

ID CFormaGral::OnCut ()

if ( fnGetBool ( FM_DRAG_DROP ) == FALSE ) {
    // Nosotros no empezamos la acción de drag por lo cual podemos llevar
    // a cabo acciones para manipular la información.
    if ( fnGetBool ( FM_EDITGE_INFO ) ) {
        OnCopyGen( FALSE );
    }
    else {
        if ( !CClipboard::fnClipboardSpread ( this,
                                             m_Spread,
                                             ID_SPREAD_CUT ) )
            OnCopyGen( FALSE );
    }
    else
    {
        // Se esta copiando el spread, reajusta datos
        m_arrIDS.RemoveAll ();
        m_arrIDS.Add ( IDCOPY_PRECIOS );
        fnAjustaAlgo ( m_arrIDS,
                      DROPEFFECT_MOVE );
        m_arrIDS.RemoveAll ();
    }
}

}

}

```

Capítulo IV Desarrollo del Sistema

```

}endfunction

void CFormaGral::OnCopy ()
{
    if ( !fnGetBool ( FM_DRAG_DROP ) == FALSE ) {
        // Nosotros no empezamos la acción de drag por lo cual podemos llevar
        // a cabo acciones para manipular la información.
        OnCopyGen ();
    }
}endfunction

void CFormaGral::OnCopyGen ( BOOL bCopy /* = TRUE */ )
{
    // El control de spread tiene el focus y desean copiar, enviamos
    // mensaje al spread para que lo procese
    if ( !fnGetBool ( FM_ELIGE_INFO ) ) {
        fnReallyCopy();
        fnSetBool ( FM_ELIGE_INFO, FALSE );

        if ( !bCopy ) {
            // Tenemos que limpiar la forma, pero solamente los controles que
            // se encuentran seleccionados
            fnLimpiaControles ();
        }

        fnValidaForma ();
    }
    else {
        // No existe selección dentro de la forma el usuario desea llevar
        // a cabo un copiado de información desde otro tipo de control,
        // checamos por el spread
        if ( !CClipboard::fnClipboardSpread ( this,
            m_Spread,
            ID_SPREAD_COPY ) ) {

            // No es el spread el que tiene el focus, entonces se trata de un
            // control de edición
            CEdit *pWndControl = ( CEdit * ) GetFocus ();

            if ( !bCopy )
                pWndControl->Cut ();
            else
                pWndControl->Copy ();
        }
    }
}endfunction

ColeDataSource * CFormaGral::fnReallyCopy ()
{
    CClipboard Clipboard;

    return Clipboard.OnCopy ( this,

```

```

        m_arrIDS,
        m_Spread,
        1, 1,
        m_Spread.GetMaxCols (),
        m_Spread.GetDataCnt () );
}endfunction

void CFormaGral::OnPaste ()
{
    if ( fnGetBool ( EM_ELIGE_INFO ) ) {
        // Esta elegida la información, tenemos que aplicar el algoritmo
        // que permita solo pasar información a los elementos que se encuen-
        // tran seleccionados
        CClipboard Clipboard;
        Clipboard.OnPaste ( this, m_arrIDS, m_Spread, 1, 1 );
        fnAjustaAlgo ( m_arrIDS,
            DROPEFFECT_MOVE );
    }
    else {
        // OK, tenemos que encontrar quien en el control que tiene el
        // focus
        m_hwndHijo = ::GetFocus ();

        if ( ::IsChild ( m_hwnd, m_hwndHijo ) )
        {
            CPoint point (0,0);
            fnPasteSpecial ( m_hwndHijo, point );

            // Se trata del spread?
            if ( ::GetDlgCtrlID ( m_hwndHijo ) )
            {
                m_arrIDS.RemoveAll ();
                m_arrIDS.Add ( IDCSP_PRECIOS );
                fnAjustaAlgo ( m_arrIDS,
                    DROPEFFECT_MOVE );
                m_arrIDS.RemoveAll ();
            }
        }
    }
}endfunction

// FUNCION: fnPasteSpecial ( HWND hwndHijo )
// DESCRIPCION : Esta función se encarga de pasar información desde el clip-
// board hacia los controles que se encuentran desde hwndHijo
// hacia adelante.
// PARAMETROS: HWND hwndHijo es el handle del control desde donde comienza
// el copiado de información a los controles de
// la forma desde el clipboard
void CFormaGral::fnPasteSpecial ( HWND hwndHijo,
    CPoint point )
{
    CRect rcSpread;
    BOOL dropResult = FALSE;

    // Ok ahora tenemos que volver a enumerar los controles a partir del
    // control donde se salto el mouse, e irlos agregando a la lista de
    // controles donde copiaremos información
    m_arrIDS.RemoveAll (); // Limpialo antes de utilizarlo

    while ( hwndHijo ) {
        if ( fnEsControlCopiar ( hwndHijo ) ) {

```

```
        m_arrIDS.Add ( ::GetDlgCtrlID ( hwndHijo ) );
        dropResult = TRUE;
    }endif

    hwndHijo = ::GetWindow ( hwndHijo, GW_HWNDNEXT );
}endwhile

if ( dropResult ) {
    // Hay elementos seleccionados, ahora realicemos el paste de información
    CClipboard Clipboard;
    SS_COORD ssCol, ssRen;
    CRect rect;

    // El punto viene en coordenadas relativas a la vista, tenemos que
    // pasariias a las del spread
    MapWindowPoints ( &m_Spread, &point, 1 );
    m_Spread.GetClientRect ( &rect );

    if ( rect.PtInRect ( point ) )
    {
        // Ok ya tenemos el punto para que se obtengan el renglón y columna
        // desde donde se esta soltando el mouse
        m_Spread.GetCellFromPixel ( &ssCol, &ssRen, point.x, point.y );
    }
    else
    {
        // Bueno por favor obten la celda activa
        m_Spread.GetActiveCell ( &ssCol, &ssRen );
    }endif

    Clipboard.OnPaste ( this, m_arrIDS, m_Spread, ssCol, ssRen );
}endif

m_arrIDS.FreeExtra ();
m_arrIDS.RemoveAll ();
}endfunction
```

Como se observa el crear una clase general es complejo, pero indiscutiblemente permite ahorrar mucho tiempo ya que la funcionalidad se centra en una sola clase y basta hacer nuevas clases derivadas de la base para obtener la misma funcionalidad.

Ahora bien las vistas son presentadas dentro de la Ventana Frame del documento. Esta ventana es una ventana hija que se presenta dentro de la ventana principal de nuestra aplicación.

La plantillas de documentos orquestan la creación de documentos, vistas y ventanas de Frame. Una clase de plantilla de documento crea y maneja todos los documentos

Capítulo IV Desarrollo del Sistema

```
ON_BN_CLICKED ( IDC_DISENA, OnDiseña)
ON_BN_CLICKED ( IDC_MOVER_ARRIBA, OnMueveArriba)
ON_BN_CLICKED ( IDC_MOVER_ABAJO, OnMoverAbajo)
ON_BN_CLICKED ( IDC_CHECK1, OnCheckOrdena )
ON_COMMAND ( IDAYUDA, OnAyuda )
ON_COMMAND ( IDCANCEL, OnCancel )
ON_EN_KILLFOCUS ( IDC_REP_INICIAL, OnKillFocusRepIni )
ON_EN_KILLFOCUS ( IDC_REP_FINAL, OnKillFocusRepFinal )
END_MESSAGE_MAP()

////////////////////////////////////
// CReporte message handlers

BOOL CReporte::OnInitDialog()
{
    m_hRepLib = LoadLibrary ("repestud.dll");

    if (m_hRepLib > HINSTANCE_ERROR ) {

        HWND hStatic = ::GetDlgItem ( m_hWnd, IDC_STATIC );
        long lEstilo;

        fnPonBitmapBoton( m_hWnd, IDC_DISENA, MAKEINTRESOURCE ( IDB_DISENA ) );
        DDfnCargaBitmapAyAcCa( m_hWnd, IDAYUDA );

        CDialog::OnInitDialog();

        CVBControl * pTab =
            (CVBControl *) GetDlgItem ( IDC_REPORTES );

        pTab->SetNumProperty ( "BorderStyle", 0 );
        pTab->SetNumProperty ( "ActiveTab", m_nTipoDlgDinamico );

        lEstilo = GetWindowLong ( hStatic, GWL_STYLE );
        lEstilo |= WS_CLIPSIBLINGS;
        SetWindowLong ( hStatic, GWL_STYLE, lEstilo );

        OnDescartar();

        return FALSE; // return TRUE unless you set the focus to a control
    }
    else
    {
        CDialog::OnCancel ();
        return FALSE;
    }
}

void CReporte::OnDescartar()
{
    //int nOpcionTemp = m_nTipoDlgDinamico;
    int nOpcionTemp = m_nTipoDlgDinamico;

    // Desean desacartar la información que han editado, pasamos
    // a grabar la información desde PARAM_INI hacia SAICIC_INI
    //fnObtenInformacion ( PARAM_INI, SAICIC_INI );

    m_nTipoDlgDinamico = -1;
    // Presentamos la información correspondiente a la pantalla
    fnCargaInfoTabs ( nOpcionTemp );

    m_nTipoDlgDinamico = nOpcionTemp;
}

void CReporte::OnCambiaFormato ()
```

```

{
    ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_REP_PRECIO ), TRUE );
    ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_REP_PREC_DESU ), TRUE );
}
endif

void CReporte::fnCargaInfoTabs ( int nOpcion )
{
    BOOL bNoCargaControl;

    if ( ( m_nTipoDlgDinamico >= 0 && m_nTipoDlgDinamico <= 3 )
        &&
        ( nOpcion >= 0 && nOpcion <= 3 ) )
        bNoCargaControl = TRUE;
    else
        if ( ( m_nTipoDlgDinamico >= 6 && m_nTipoDlgDinamico <= 10 )
            &&
            ( nOpcion >= 6 && nOpcion <= 10 ) )
            bNoCargaControl = TRUE;
        else
            bNoCargaControl = (BOOL) ( m_nTipoDlgDinamico == nOpcion );
        endif
    endif

    switch ( m_nTipoDlgDinamico ){
        case 0:
            // Salvar información sobre materiales
            fnSalvarInfoInsumos ( SECC_REP_MATERIAL );
            break;

        case 1:
            // Salvar información sobre obra de mano
            fnSalvarInfoInsumos ( SECC_REP_OBRAMAN );
            break;

        case 2:
            // Salvar información de maquinaria y equipo
            fnSalvarInfoInsumos ( SECC_REP_MAQEQUO );
            break;

        case 3:
            // Salvar información de subcontratos
            fnSalvarInfoInsumos ( SECC_REP_SUBCONTR );
            break;

        case 4:
            // Salvar información de FSR
            fnSalvarInfoFSR ( SECC_REP_FSR );
            break;

        case 5:
            // Salvar información de Presupuestos
            fnSalvarInfoPresupue ( SECC_REP_PRESUPUE );
            break;

        case 6:
            // Salvar información de básicos
            fnSalvarInfoAnalisis ( SECC_REP_BASICOS );
            break;

        case 7:
            // Salvar información de integrados
            fnSalvarInfoAnalisis ( SECC_REP_INTEGRADOS );
            break;
    }
}

```

```
case 8:
    // Salvar información de partidas
    fnSalvaInfoAnálisis ( SECC_REP_PARTIDAS );
    break;

case 9:
    // Salvar información de capítulos
    fnSalvaInfoAnálisis ( SECC_REP_CAPITULOS );
    break;

case 10:
    // Salvar información de extraordinarios
    fnSalvaInfoAnálisis ( SECC_REP_EXTRA );
    break;

case 11:
    // Es el caso de configuración
    m_nNoUtilizarTipo =
        ( (CHutton * ) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    fnSalvaConfiguración ();
    break;
}endswitch

switch ( nOpcion ) {

case 0:
    // Obtener información sobre materiales
    m_nTipoDlgDinámico = 0;
    fnCargaControlInsumos ( bNoCargaControl );
    fnInicializaInsumos ( SECC_REP_MATERIAL, 0 );

    break;

case 1:
    // Obtener información sobre obra de mano
    m_nTipoDlgDinámico = 1;
    fnCargaControlInsumos ( bNoCargaControl );
    fnInicializaInsumos ( SECC_REP_OBRMAN, 0 );

    break;

case 2:
    // Obtener información de maquinaria y equipo
    m_nTipoDlgDinámico = 2;
    fnCargaControlInsumos ( bNoCargaControl );
    fnInicializaInsumos ( SECC_REP_MAQEQU, 0 );

    break;

case 3:
    // Obtener información de subcontratos
    m_nTipoDlgDinámico = 3;
    fnCargaControlInsumos ( bNoCargaControl );
    fnInicializaInsumos ( SECC_REP_SUBCONTR, 0 );

    break;

case 4:
    // Salvar información de FSR
    m_nTipoDlgDinámico = 4;
    LoadAndCreateControls ( IDD_REP_FSR,
        IDC_ANCHOR, IDC_ANCHOR );
    ::InvalidateRect ( ::GetDlgItem ( m_hWnd, IDC_REP_OPCI_DEFINIDO),
        NULL, TRUE );
    fnInicializaFSR ( SECC_REP_FSR, 0 );
    break;

case 5:
    // Salvar información de Presupuestos
    m_nTipoDlgDinámico = 5;

    if ( !bNoCargaControl ) {
```

```

LoadAndCreateControls ( IDD_REPORT_PRE,
                      IDC_ANCHOR,
                      IDC_ANCHOR );

::InvalidataRect ( ::GetDlgItem ( m_hWnd, IDC_REP_OFCL_DEFINIXO),
                  NULL, TRUE );

CComboBox *pNivel = (CComboBox *) GetDlgItem ( IDC_NIVEL );
pNivel->AddString ( TIPODOC_BASICOS );
pNivel->AddString ( TIPODOC_INTEGRADOS );
pNivel->AddString ( TIPODOC_PARTIDAS );
pNivel->AddString ( TIPODOC_CAPITULOS );
pNivel->AddString ( TIPODOC_EXTRAORDIN );

GotoDlgCtrl ( (CWnd *) pNivel );

}endif

fnInicializaPresupuesto ( SECC_REP_PRESUPUE, 5 );
break;

case 6:
// Salvar información de básicos
m_nTipoDlgDinamico = 6;
fnCargaControlAnálisis ( bNoCargaControl );
fnInicializaAnálisis ( SECC_REP_BASICOS, 6 );

break;

case 7:
// Salvar información de integrados
m_nTipoDlgDinamico = 7;
fnCargaControlAnálisis ( bNoCargaControl );
fnInicializaAnálisis ( SECC_REP_INTEGRADOS, 7 );

break;

case 8:
// Salvar información de partidas
m_nTipoDlgDinamico = 8;
fnCargaControlAnálisis ( bNoCargaControl );
fnInicializaAnálisis ( SECC_REP_PARTIDAS, 8 );

break;

case 9:
// Salvar información de capítulos
m_nTipoDlgDinamico = 9;
fnCargaControlAnálisis ( bNoCargaControl );
fnInicializaAnálisis ( SECC_REP_CAPITULOS, 9 );

break;

case 10:
// Salvar información de extraordinarios
m_nTipoDlgDinamico = 10;
fnCargaControlAnálisis ( bNoCargaControl );
fnInicializaAnálisis ( SECC_REP_EXTRA, 10 );

break;

case 11:
m_nTipoDlgDinamico = 11;
// OK por favor coloca la info del reporte
loadAndCreateControls ( IDD_REP_CONFIG,
                      IDC_ANCHOR,
                      IDC_ANCHOR );

// Ahora coloca por favor el elemento
( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->SetCheck (
    m_nUsarUtilizarTipo );
fnCargaConfiguracion ();

}endswitch

```



```

}endfunction

void CReporte::fnObtenInfoDeSobreCosto ( ST_HEADER_REP * lpsHeader )
{
    CComboBox * pboxFormato = (CComboBox *) GetDlgItem ( IDC_REP_FORMATO );
    CButton * pradACosto = (CButton *) GetDlgItem ( IDC_REP_COSTO );
    CButton * pchkDesgloce = (CButton *) GetDlgItem ( IDC_REP_PREC_DESG );

    //Formato de Sobrecosto A Aplicar
    lpsHeader->fmoSobreCosto = LOWORD( pboxFormato->GetCurSel() );

    lpsHeader->IACostoOPrecio = REPTE_A_COSTO;

    //Si el botón A costo directo No está checado
    //esto quiere decir que el reporte será a precio
    if ( !pradACosto->GetCheck() )
    {
        lpsHeader->IACostoOPrecio = REPTE_PRECIO_SIN_DESGLOCE;

        if ( pchkDesgloce->GetCheck() )
        {
            lpsHeader->IACostoOPrecio = REPTE_PRECIO_CON_DESGLOCE;
        }
    }
}

}endfunction

void CReporte::OnAyuda ()
{
    SendMessage ( WM_COMMAND, ID_HELP, 0L );
}

}endfunction

void CReporte::OnDestroy()
{
    CDlgDinamico::OnDestroy();
    FreeLibrary ( m_hRepLib );
}

void CReporte::fnCargaConfiguracion ()
{
    char szTipo [] = { 0, 0, 0 };
    INT nElemento;
    CString strOrden;
    CListBox *pList = (CListBox *) GetDlgItem ( IDC_LIST1 );

    strOrden = theApp.GetProfileString ( SECC_REP_CONFIG,
                                        CAMPO_APARLCE_ELE,
                                        ORDEN_DEFAULT );

    // Elige el orden que el usuario dara a los elementos del análisis
    for ( int nContador = 0;
          nContador <= NUM_ELE_ORDEN - TAM_TIPO;
          nContador += TAM_TIPO
        )
    {
        // Copia parte del buffer para obtener elemento
        Strncpy ( szTipo, (LPCSTR) ( (LPCSTR) strOrden + nContador ), TAM_TIPO );

        // Obten el primer elemento de orden
        switch ( szTipo [0] )
        {

```

```
case TIPO_BASICO:
    nElemento = pList->AddString ( TIPODOC_BASICOS );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
    break;

case TIPO_CAPITULO:
    nElemento = pList->AddString ( TIPODOC_CAPITULOS );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
    break;

case TIPO_HERRAMIENTA:
    switch ( szTipo[1] )
    {
        case TIPO_MANDOS_INTER:
            nElemento = pList->AddString ( STR_TIPO_MANDOS_INTER );
            pList->SetItemData ( nElemento,
                MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
            break;

        case TIPO_HERR_MATER:
            nElemento = pList->AddString ( STR_TIPO_HERR_MATER );
            pList->SetItemData ( nElemento,
                MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
            break;

        case TIPO_HERR_PRELIMINARES:
            nElemento = pList->AddString ( STR_TIPO_HERR_PRELIMINARES );
            pList->SetItemData ( nElemento,
                MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
            break;

        case TIPO_HERR_MAQ:
            nElemento = pList->AddString ( STR_TIPO_HERR_MAQ );
            pList->SetItemData ( nElemento,
                MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
            break;

        case TIPO_HERR_SURCT:
            nElemento = pList->AddString ( STR_TIPO_HERR_SURCT );
            pList->SetItemData ( nElemento,
                MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
            break;

        default:
            nElemento = pList->AddString ( STR_TIPO_HERRAMIENTA );
            pList->SetItemData ( nElemento,
                MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
            break;
    }
    }endswitch

    break;

case TIPO_INTEGRADO:
    nElemento = pList->AddString ( TIPODOC_INTEGRADOS );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
    break;

case TIPO_MAT:
    nElemento = pList->AddString ( TIPODOC_MAT );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
    break;

case TIPO_OBRAMANO:
    nElemento = pList->AddString ( TIPODOC_OBRMAN );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT{szTipo[0], szTipo[1]}, 0 ) );
    break;
```



```

case TIPO_PARTIDA:
    nElemento = pList->AddString ( TIPODOC_PARTIDAS );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT(szTipo[0], szTipo[1]), 0 ) );
    break;

case TIPO_MAQUINARIA:
    nElemento = pList->AddString ( TIPODOC_MAQUINI );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT(szTipo[0], szTipo[1]), 0 ) );
    break;

case TIPO_SUBCONTRATO:
    nElemento = pList->AddString ( TIPODOC_SUBCONT );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT(szTipo[0], szTipo[1]), 0 ) );
    break;

case TIPO_EXTRA:
    nElemento = pList->AddString ( TIPODOC_EXTRAORDIN );
    pList->SetItemData ( nElemento,
        MAKELONG ( MAKEINT(szTipo[0], szTipo[1]), 0 ) );
    break;

}endswitch

}endif

// Bueno obtenemos el elemento que deseamos para el modo de captura
{ (CButton *) GetDlgItem ( IDC_CHECK2 ) }->SetCheck (
    theApp.GetProfileInt ( SECC_REP_CONFIG,
        CAMPO_RESPETA_CAP,
        1 ) );

{ (CButton *) GetDlgItem ( IDC_CHECK3 ) }->SetCheck (
    theApp.GetProfileInt ( SECC_REP_CONFIG,
        CAMPO_ORDEN_COMPO_ALIAS,
        0 ) );

if ( !theApp.GetProfileInt ( SECC_REP_CONFIG,
    CAMPO_ORDENA,
    1 ) )
{
    { (CButton *) GetDlgItem ( IDC_RADIO2 ) }->SetCheck ( 1 );
}
else
{
    { (CButton *) GetDlgItem ( IDC_RADIO3 ) }->SetCheck ( 1 );
}
}endif

// Habilita elementos de control
OnCheckOrdena ();

endfunction

oid CReporte::OnKillFocusRepFinal ()

CWnd *pEditInsumos = GetDlgItem ( IDC_REP_FINAL );
int nNumero;

//nBanderas = pMDIActiva->m_iBanderas;
pEditInsumos->GetWindowText ( szClave, sizeof szClave );

// Tenemos que ver si se trata de un numero
nNumero = StringToInt ( (LPCTSTR) (LPCSTR) szClave );

if ( !oConfigura.m_bClavesNumericas && nNumero <= 9999 && nNumero > 0 )
{
    // Se trata de un numero por favor rellenaló con ceros
    wsprintf ( szClave, "%04i", nNumero );
}

```

Capítulo IV Desarrollo del Sistema

```
        pEditInsumos->SetWindowText ( szClave );
    }
}

void CReporte::OnKillFocusRepIn ()
{
    CWnd *pEditInsumos = GetDlgItem ( IDC_REP_INICIAL );
    int nNumero;

    //nBanderas = pMDIActiva->m_lBanderas;
    pEditInsumos->GetWindowText ( szClave, sizeof szClave );

    // Tenemos que ver si se trata de un numero
    nNumero = StringToInt ( (LPCTSTR) (LPCSTR) szClave );

    if ( oConfigura.m_bClavesNumericas && nNumero <= 9999 && nNumero > 0 )
    {
        // Se trata de un numero por favor relleno con ceros
        wprintf ( szClave, "%04i", nNumero );
        pEditInsumos->SetWindowText ( szClave );
    }
}

int fnFuncionChecaNumEle (
    HWND m_hWnd,
    int nRegistro,
    LPMDIVentStruct m_lpInfoObra,
    int nNumElePermitidos,
    UINT nMensaje
);

void CReporte::OnOk()
{
    CButton * pbtnPantalla = (CButton *) GetDlgItem ( IDC_REP_PANTALLA );
    CButton * pbtnImpresora = (CButton *) GetDlgItem ( IDC_REP_IMPRESORA );
    CComboBox * pboxRepRegist = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );
    CComboBox * pboxOrdenadoPor = (CComboBox *) GetDlgItem ( IDC_REP_ORDENA );
    long lOrdenaPor = 0;
    CActualiza oActualiza;
    ST_HEADER_REP sHeader;
    CString strDir;
    BOOL bAnálisis = FALSE;

    if ( !l == m_nTipoDlgDinamico )
    {
        // Salva la información de configuración
        m_nNoUtilizarTipo =
            ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
        fnSalvaConfiguracion ();

        MessageBeep ( NB_ICONSTOP );
        return;
    }

    sHeader.hVentanaPrincipal = theApp.m_pMainWnd->GetSafeHwnd ();

    if ( theApp.fnGetBool ( REP_ACTIVO ) )
    {
        // El Repase se encuentra activo informalo
        DDfnDspyErrorUsuario ( m_hWnd,
            theApp.m_hInstance,
            theApp.m_szUsuario,
            "",
            IDST_ERR_SALIC_REP,
            MB_OK | MB_ICONINFORMATION );
    }
}
```

```

return;
}endif

HMODULE hMod = NULL;
MODULEENTRY me;

MemSet( &me, 0, sizeof (MODULEENTRY) );
me.dwSize = sizeof (MODULEENTRY);

if( hMod = ModuleFindName ( &me, "PEPESTUD" ) ){
    if ( GetModuleUsage (hMod) >= 2 )
    {
        // Ya se encuentra corriendo el editor, por favor indicalos
        DdInDspyErrorUsuario ( m_hWnd,
            theApp.m_hInstance,
            theApp.m_szUsuario,
            "",
            IDST_ERR_SAICIC_REP,
            MA_OK | MB_ICONINFORMATION );

        return;
    }endif
}endif

// Tomar en cuenta que el tab de preseupuestos no tiene el combobox de
// ordenamiento RCR 18/Abr/94
if ( pboxOrdenadoPor &&
    CB_ERR == pboxOrdenadoPor->GetCurSel() )
{
    MessageBeep ( MB_ICONSTOP );
    pboxOrdenadoPor->SetFocus ();
}endif

// Limpiamos la estructura del header
Zero ( sHeader );

//Se llenan los campos generales de la
//estructura que solicita la generación
//del reporte.

//Directorio de la Obra
strDir = m_lpInfoObra->szDrive;
strDir += m_lpInfoObra->szDir;
lstrcpy(sHeader.szDirObra, strDir );

//apuntador a la vista activa
sHeader.hVentImp = NULL;

// Colocamos el modo de apertura para la obra
sHeader.nModoAperturaObra = m_lpInfoObra->n(Banderas);
DdInMaCamDrvDir( m_hWnd, m_lpInfoObra, TRUE );

//Instancia de SAICIC
sHeader.hInstance = theApp.m_hInstance;
sHeader.hPrevInstance = theApp.m_hPrevInstance;
sHeader.bObtenerCostoHorario = ( BOOL ) GetPrivateProfileInt (
    ID_SECC_FACTORES,
    ID_CAMPO_DESGLOSA_MAQ,
    0,
    FACTORES_INF );

//Reporte Seleccionado de la Lista de Disponibles
if ( CB_ERR == ( sHeader.lNoRegistro = pboxRepRegist->GetCurSel() ) )
{
    pboxRepRegist->SetFocus ();
    MessageBeep ( MB_ICONSTOP );
    return;
}

```

```

}endif

//Tipo de Concepto Del Cual Se emitirá el reporte
fnObtenTipoRep ( sHeader.cConcepto );

//Coloca el apuntador de la tabla de formatos activa
MemCpy ( sHeader.pFormatos, &oConfigura.m_nDecCostosPrecios,
        sizeof ( Formateo ) * NUM_ELE_FORMATOS );

//Dispositivo donde será enviado el reporte
if ( pbtnPantalla->GetCheck() )
{
    sHeader.cDispositivo = PANTALLA;
}
else
{
    if ( pbtnImpresora->GetCheck() )
    {
        sHeader.cDispositivo = IMPRESORA;
    }endif
}endif

//Le indica al núcleo del reporteador que el
//scrip de definición del reporte lo cargue
//de un archivo en disco

sHeader.lReporte = SCRIPT_REGISTRADO;
sHeader.ifmtoSobreCosto = CB_ERR;

//Ordenamiento del Reporte.
lOrdenaPor = 0;

if ( NULL != pboxOrdenadoPor )
{
    //Si el Tab que está activo no es el de
    //Presupuestos Ni el FSR es necesario
    //obtener El tipo de ordenamiento que se
    //le dará al reporte.
    lOrdenaPor = pboxOrdenadoPor -> GetCurSel();
}endif

sHeader.iTipoSort = (int) ( LOWORD ( lOrdenaPor ) | ORDENA_ASCENDENTE );

long lModulo;
long lConcepto;

// Leamos que configuración de moneda tenemos
sHeader.ulConfigMoneda = theApp.GetProfileInt ( SECC_REP_CONFIG,
        CAMPO_MONEDA,
        TRUE );

switch ( m_nTipoDlgDinamico )
{
case 0:
    //Salvamos la información de Materiales
    fnSalvaInfoInsumos ( SECC_REP_MATERIAL );

    lModulo = (long)MODULO_INSUMOS;
    lConcepto = (long)SUB_MOD_MATERIALES;

    // Se tiene que checar que la bandera de materiales este desactivada
    // Esperamos pues
    if ( GetPrivateProfileInt ( ID_SECC_ACTUALIZA,
        ID_CAMPO_ACTUAL_MAT,
        0,

```

```

                ARCH_OBRA_INI ) )

// Los elementos de materiales no se encuentran actualizados
// Pregunta al usuario
CEjecutaActualizacion dlgEjeActualiza ( IDD_EJE_ACTUALIZACION );

dlgEjeActualiza.m_nTipoMensaje = IDS_STRING153;
dlgEjeActualiza.m_strMensaje.LoadString ( IDS_STRING153 );

switch ( dlgEjeActualiza.DoModal () )
{
    case IDCANCEL:
        // Adios el usuario se arrepintio
        return;

    case IDOK:
        // Manda a ejecutar la actualización
        if ( IDC_RBTN_1 == dlgEjeActualiza.m_ulTipoAccion )
        {
            BeginWaitCursor ();
            oActualiza.fnActualizaAutomatico (
                TIPO_ACT_MATERIALES,
                dlgEjeActualiza.m_lFecha );
            EndWaitCursor ();
        }
        }endif
        // Sigue tu camino viejo
    }endswitch

}endif

sHeader.bDescendente =
    ( CButton * ) GetDlgItem ( IDC_CHECK1 ) ->GetCheck ();

break;

case 1:
    // Salvar Información sobre obra de mano
    fnSalvaInfoInsumos ( SECC_REP_OBRMAN );

    //Indica si se trata de un catálogo
    lModulo = (long)MODULO_INSUMOS;
    lConcepto = (long)SUB_MOD_MANO_OBRA;

    if ( GetPrivateProfileInt ( ID_SECC_ACTUALIZA,
        ID_CAMPO_ACTUAL_OBRA_MAN,
        0,
        ARCH_OBRA_INI ) )
    {
        CEjecutaActualizacion dlgEjeActualiza ( IDD_EJE_ACTUALIZACION );

        dlgEjeActualiza.m_nTipoMensaje = IDS_STRING154;
        dlgEjeActualiza.m_strMensaje.LoadString ( IDS_STRING154 );

        switch ( dlgEjeActualiza.DoModal () )
        {
            case IDCANCEL:
                // Adios el usuario se arrepintio
                return;

            case IDOK:
                // Manda a ejecutar la actualización
                if ( IDC_RBTN_1 == dlgEjeActualiza.m_ulTipoAccion )
                {
                    BeginWaitCursor ();
                    oActualiza.fnActualizaAutomatico (
                        TIPO_ACT_OBRA_DE_MANO,
                        dlgEjeActualiza.m_lFecha );
                }
            }
        }
    }
}

```

Capítulo IV Desarrollo del Sistema

```
        EndWaitCursor ();

        }endif
        // Sigue tu camino viejo

    }endawitch

}endif

sHeader.bDescendente =
    ( CButton * ) GetDlgItem ( IDC_CHECK1 )->GetCheck ();

break;

case 2:
    // Salvar información de maquinaria y equipo
    fnSalvaInfoInsumos ( SECC_REP_MAQEQU );

    lModulo = (long)MODULO_INSUMOS;
    lConcepto = (long)SUB_MOD_MAQ_EQPO;

    if ( GetPrivateProfileInt ( ID_SECC_ACTUALIZA,
                                ID_CAMPO_ACTUAL_MAQ,
                                0,
                                ARCH_OBRA_INI ) )
    {

        CEjecutaActualizacion dlgEjeActualiza ( IDD_EJE_ACTUALIZACION );

        dlgEjeActualiza.m_nTipoMensaje = IDS_STRING155;
        dlgEjeActualiza.m_strMensaje.LoadString ( IDS_STRING155 );

        switch ( dlgEjeActualiza.DoModal () )
        {

            case IDCANCEL:
                // Adios el usuario se arrepintio
                return;

            case IDOK:
                // Manda a ejecutar la actualización
                if ( IDC_RBTN_1 == dlgEjeActualiza.m_uTipoAccion )
                {

                    BeginWaitCursor ();
                    oActualiza.fnActualizaAutomatico (
                        TIPO_ACT_MAQUINARIA,
                        dlgEjeActualiza.m_lFecha );
                    EndWaitCursor ();

                }endif
                // Sigue tu camino viejo

            }endawitch

        }endif

        sHeader.bDescendente =
            ( CButton * ) GetDlgItem ( IDC_CHECK1 )->GetCheck ();

        break;

    case 3:
        // Salvar información de subcontratos
        fnSalvaInfoInsumos ( SECC_REP_SUBCONTR );
        lModulo = (long)MODULO_INSUMOS;
        lConcepto = (long)SUB_MOD_SUBCONTRATOS;
        sHeader.bDescendente =
            ( CButton * ) GetDlgItem ( IDC_CHECK1 )->GetCheck ();

        break;

    case 4:
        // Salvar información de ESR
        fnSalvaInfoESR ( SECC_REP_ESR );

        break;
```

```

case 5:
{
CComboBox * pboxNivel = (CComboBox *)GetDlgItem(IDC_NIVEL);

sHeader.uiConfigMoneda = TRUE; // Usar Moneda base

// Salvar información de Presupuestos
fnSalvaInfoPresupue ( SECC_REP_PRESUPUE );

lModulo = (long)MODULO_PRESUPUESTOS;
fnObtenInfoDeSobreCosto( sHeader );

switch ( pboxNivel->GetCurSel() )
{
case 0:
lConcepto = (long)SUB_MOD_BASICOS;
break;
case 1:
lConcepto = (long)SUB_MOD_INTEGRADOS;
break;
case 2:
lConcepto = (long)SUB_MOD_PARTIDAS;
break;
case 3:
lConcepto = (long)SUB_MOD_CAPITULOS;
break;
case 4:
lConcepto = (long)SUB_MOD_EXTRAORDINARIOS;
break;
}

if ( GetPrivateProfileInt ( TD_SECC_ACTUALIZA,
                          TD_CAMPO_ACTUAL_AHA,
                          0,
                          ARCH_OBRA_INI ) )
{
EjecutaActualizacion dlgEjeActualiza ( IDD_EJE_ACTUALIZACION );

dlgEjeActualiza.m_nTipoMensaje = IDS_STRING156;
dlgEjeActualiza.m_strMensaje.LoadString ( IDS_STRING156 );

switch ( dlgEjeActualiza.DoModal () )
{
case IDCANCEL:
// Adios al usuario se arrepiento
return;

case IDOK:
// Manda a ejecutar la actualización
if ( TDC_RBTN_1 == dlgEjeActualiza.m_uiTipoAccion )
{
BeginWaitCursor ();
oActualiza.fnActualizaAutomatico (
TIPO_ACT_ANALISIS,
dlgEjeActualiza.m_lFecha );
EndWaitCursor ();

}endif
// Sigue tu camino viejo

}endswitch

}endif

bAnalisis = FALSE;

break;

}endblock

```

```

case 6:
    // Salvar información de básicos
    fnSalvarInfoAnálisis ( SECC_REP_BASICOS );
    bAnálisis = TRUE;
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_BASICOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 7:
    // Salvar información de integrados
    fnSalvarInfoAnálisis ( SECC_REP_INTEGRADOS );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_INTEGRADOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 8:
    // Salvar información de partidas
    fnSalvarInfoAnálisis ( SECC_REP_PARTIDAS );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_PARTIDAS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 9:
    // Salvar información de capítulos
    fnSalvarInfoAnálisis ( SECC_REP_CAPITULOS );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_CAPITULOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 10:
    // Salvar información de extraordinarios
    fnSalvarInfoAnálisis ( SECC_REP_EXTRA );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_EXTRAORDINARIOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;
}endswitch

if ( 4 != m_nTipoDlgDinamico )
{
    //Si el reporte a imprimir no es
    //de factor de salario real
    //se obtiene el criterio de selección
    //si se requiere
    CString strValIni;
    CString strValFin;

    //Criterio de selección que se utilizará en el reporte
    if ( ( (CButton *) GetDlgItem ( IDC_REP_RANGO ) )->GetCheck() )
    {
        ((CEdit *)GetDlgItem(IDC_REP_INICIAL))->GetWindowText(strValIni);
        ((CEdit *)GetDlgItem(IDC_REP_FINAL))->GetWindowText(strValFin);
    }
}

```



```

case 6:
    // Salvar información de básicos
    fnSalvarInfoAnálisis ( SECC_REP_BASICOS );
    bAnálisis = TRUE;
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_BASICOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 7:
    // Salvar información de integrados
    fnSalvarInfoAnálisis ( SECC_REP_INTEGRADOS );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_INTEGRADOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 8:
    // Salvar información de partidas
    fnSalvarInfoAnálisis ( SECC_REP_PARTIDAS );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_PARTIDAS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 9:
    // Salvar información de capítulos
    fnSalvarInfoAnálisis ( SECC_REP_CAPITULOS );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_CAPITULOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;

case 10:
    // Salvar información de extraordinarios
    fnSalvarInfoAnálisis ( SECC_REP_EXTRA );
    lModulo = (long)MODULO_ANALISIS;
    lConcepto = (long)SUB_MOD_EXTRAORDINARIOS;
    fnObtenInfoDeSobreCosto(&shHeader);
    bAnálisis = TRUE;
    shHeader.bDescendente =
        ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ();
    break;
}

}

if ( 4 != m_nTipoDlgD(namico) )
{
    //Si el reporte a imprimir no es
    //de factor de salario real
    //se obtiene el criterio de selección
    //si se requiere
    CString strValIni;
    CString strValFin;

    //Criterio de Selección que se utilizará en el reporte
    if ( (CButton *) GetDlgItem ( IDC_REP_RANGO )->GetCheck() )
    {
        ((CEdit *) GetDlgItem(IDC_REP_INICIAL))->GetWindowText(strValIni);
        ((CEdit *) GetDlgItem(IDC_REP_FINAL))->GetWindowText(strValFin);
    }
}

```

```

// SE ACUERDAN del primero y ultimo, puesto checa
if ( StriEqu ( SEL_PRIMERO, (LPSTR) (LPCSTR) strValIni ) )
{
    strValIni = "";
}
endif

if ( StriEqu ( SEL_ULTIMO, (LPSTR) (LPCSTR) strValFin ) )
{
    strValFin = "";
}
endif

DDfnArmaCriterioDeSeleccion (sHeader.szCritSelec,
                             strValIni,
                             strValFin,
                             lModulo,
                             lConcepto,
                             TRUE
                             )
}
else
{
    //Para el caso del presupuesto cuando se piden todos
    //los elementos.
    strValIni = "";
    strValFin = "";
    DDfnArmaCriterioDeSeleccion (sHeader.szCritSelec,
                                 strValIni,
                                 strValFin,
                                 lModulo,
                                 lConcepto,
                                 TRUE );
}
endif
endif

if ( bAnalista )
{
    // Pregunta al usuario si desea actualización
    if ( GetPrivateProfileInt ( ID_SECC_ACTUALIZA,
                              ID_CAMPO_ACTUAL_ANA,
                              0,
                              ARCHI_CBRA_INI ) )
    {
        CEjecutaActualizacion dlgEjeActualiza ( IDD_EJE_ACTUALIZACION );
        dlgEjeActualiza.m_nTipoMensaje = IDS_STRING157;
        dlgEjeActualiza.m_strMensaje.LoadString ( IDS_STRING157 );
        switch ( dlgEjeActualiza.DoModal () )
        {
            case IDCANCEL:
                // Adios el usuario se arrepintio
                return;
            case IDOK:
                // Manda a ejecutar la actualización
                if ( IDC_BTN_1 == dlgEjeActualiza.m_ulTipoAccion )
                {
                    BeginWaitCursor ();
                    cActualiza.fnActualizaAutomatico (
                        TIPO_ACT_ANALISIS,
                        dlgEjeActualiza.m_lFecha );
                    EndWaitCursor ();
                }
            }
        }
    }
}

```

```
        )endif
        // Sigue tu camino viejo
    }endswitch
}endif
}endif
DDfnMaCamDrvDir( m_hWnd, m_lpInfoObra, TRUE );
if ( oConfigura.m_bDemo )
{
    if ( !fnFuncionChecaNumEle (
        m_hWnd,
        ANALISGEN,
        m_lpInfoObra,
        30,
        IDS_STRING455 ) )
    {
        return;
    }endif
    if ( !fnFuncionChecaNumEle (
        m_hWnd,
        ANALISGEN,
        m_lpInfoObra,
        80,
        IDS_STRING456 ) )
    {
        return;
    }endif
}endif
CDialog::OnOK();
if ( m_nNoUtilizarTipo )
{
    mHeader.nDesplazaTipo = TAM_TIPO;
}
else
{
    mHeader.nDesplazaTipo = 0;
}endif
CString strOrden;
strOrden = theApp.GetProfileString ( SECC_REP_CONFIG,
    CAMPO_APARECE_ELE,
    ORDEN_DEFAULT );
lstrcpy ( mHeader.szOrdenEnAnalisis, strOrden );
// Leemos si es necesario cambiar el tipo de orden de captura
// de los datos
mHeader.bNoRespetaOrdenAna = theApp.GetProfileInt (
    SECC_REP_CONFIG,
    CAMPO_RESPETA_CAP,
    1 );
mHeader.uiConfigAnalPres = theApp.GetProfileInt (
    SECC_REP_CONFIG,
    CAMPO_ORDEN_COMPO_ALIAS,
    0 );
dt_closeall ( DDfnMaObtTarza ( theApp.m_lpMDICen ) );
```

```

// Bueno es necesario separar el código de debug del código de la
// liberación es el unico punto donde existen diferencias

#ifdef DEBUG

UINT uiCodeErr=0;
HINSTANCE hNucleoLib;

hNucleoLib = LoadLibrary ("nucleo.rp.dll");

if ( hNucleoLib > HINSTANCE_ERROR )
{
//Cargamos la función que llena el combo Box
//Con Todos los Reportes Predefinidos.
SOLGENREPORTE DDfnSolGenReporte;

DDfnSolGenReporte = ( SOLGENREPORTE ) GetProcAddress ( hNucleoLib,
"DDfnSolGenReporte"
);

DDfnSolGenReporte ( &shellader, LLAMADO_DESDE_SATCIC, &uiCodeErr );
theApp.fnSetBool ( REP_ACTIVO, FALSE );
FreeLibrary ( hNucleoLib );
}

#endif

#ifdef COPIA
COPIAPARAMETROS fnCopiaParam = ( COPIAPARAMETROS )
GetProcAddress ( m_hReplib,
"fnCopiaParametros" );
CString strRepEstudio;

fnCopiaParam ( &shellader );

// Manda a ejecutar al reporteador
strRepEstudio.LoadString( IDS_REPESTUDIO );
strRepEstudio += " /EJECUTOR";

UINT uiCodeErr = WinExec ( strRepEstudio,
SW_SHOW );

if ( uiCodeErr < 32 )
{
DDfnDapError( NULL,
theApp.m_hinstance,
theApp.m_szUsuario,
IDS_NO_FUERO_CARGAR_REPESTUDIO );
}

#endif

DDfnMaCamDrvDir( m_hWnd, m_lpInfoObra, TRUE );
}

void CReporte::OnRepTodasClaves ()
{
if ( ( (CButton *) GetDlgItem ( IDC_REP_TODAS ) )->GetCheck()
== 1 ) {
GetDlgItem ( IDC_REP_INICIAL )->EnableWindow ( FALSE );
GetDlgItem ( IDC_REP_FINAL )->EnableWindow ( FALSE );
GetDlgItem ( IDC_REP_CLAVE_INICIAL )->EnableWindow ( FALSE );
GetDlgItem ( IDC_REP_CLAVE_FINAL )->EnableWindow ( FALSE );
}
}

```

```
void CReporte::OnRepRangoClaves ()
{
    if ( ( (CButton *) GetDlgItem ( IDC_REP_RANGO ) )->GetCheck()
        == 1 ) {
        GetDlgItem ( IDC_REP_INICIAL )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_FINAL )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_CLAVE_INICIAL )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_CLAVE_FINAL )->EnableWindow ( TRUE );

        GotoDlgCtrl ( GetDlgItem ( IDC_REP_INICIAL ) );
    }
}

void CReporte::OnRepPantalla ()
{
    if ( ( (CButton *) GetDlgItem ( IDC_REP_PANTALLA ) )->GetCheck()
        == 1 ) {
        GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( FALSE );
        GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( FALSE );
    }
}

void CReporte::OnRepImpresora ()
{
    if ( ( (CButton *) GetDlgItem ( IDC_REP_IMPRESORA ) )->GetCheck()
        == 1 ) {
        GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( FALSE );
        GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( FALSE );
    }
}

void CReporte::OnRepArchivo ()
{
    //En la versión Beta No Se Permitirá La Impresión En Archivo
    #ifndef SAICIC_BETA
    if ( ( (CButton *) GetDlgItem ( IDC_REP_ARCH ) )->GetCheck()
        == 1 ) {
        GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( TRUE );
    }
    #endif
}

void CReporte::OnEditarFormatos ()
{
    //Se desean editar los formatos de Sobre Costo
    CFactorSobreCosto dlgFactores;

    dlgFactores.DoModal();

    CString strRespuesta = "";
    CComboBox * pFormatos = (CComboBox *) GetDlgItem ( IDC_REP_FORMATO);

    //Tenemos que borrar la información del combobox de Factores
    pFormatos->ResetContent();

    CFactorSobreCosto::fnEligeFactorSobreCosto ( (CComboBox *) pFormatos,
```

```

        strReapuesta
    };

endfunction

oid CReporte::OnRegistrar()
{
    CComboBox * pOpciones = (CComboBox *) GetDlgItem ( IDC_REP_OPCT_DEFINIDO );

    DLGREGITROREP DDefnLlamaDlgRegistroRep;

    //Cargamos la función que llena el combo Box
    //Con Todos los Reportes Predefinidos.
    DDefnLlamaDlgRegistroRep = (DLGREGITROREP) GetProcAddress (
        m_hRepLib,
        "DDefnLlamaDlgRegistroRep" );

    if ( DDefnLlamaDlgRegistroRep )
    {
        CString strDir = "";
        char cTipoInsumo;

        strDir = m_lpInfoObra->szDrive;
        strDir += m_lpInfoObra->szDir;
        strDir += '\\';

        fnObtenTipoRep ( cTipoInsumo );

        DDefnLlamaDlgRegistroRep ( ID_AGREGAR,
            -1L,
            (HWND)pOpciones -> m_hWnd,
            cTipoInsumo,
            (LPSTR)(LPCSTR)strDir );

        DDefnMaCamDrvDir( m_hWnd, m_lpInfoObra, TRUE );
    }
}

//Fin CReporte::OnRegistrar

void CReporte::OnModificar()
{
    CComboBox * pOpciones = (CComboBox *) GetDlgItem ( IDC_REP_OPCT_DEFINIDO );

    if ( pOpciones->GetCount () )
    {
        DLGREGITROREP DDefnLlamaDlgRegistroRep;

        //Cargamos la función que llena el combo Box
        //Con Todos los Reportes Predefinidos.
        DDefnLlamaDlgRegistroRep = (DLGREGITROREP) GetProcAddress (m_hRepLib,
            "DDefnLlamaDlgRegistroRep"
        );

        if ( DDefnLlamaDlgRegistroRep )
        {
            CString strDir = "";
            char cTipoInsumo;

            strDir = m_lpInfoObra->szDrive;
            strDir += m_lpInfoObra->szDir;
            strDir += '\\';
            fnObtenTipoRep ( cTipoInsumo );

            DDefnLlamaDlgRegistroRep ( ID_MODIFICAR,
                (long)pOpciones -> GetCursel(),
                (HWND)pOpciones -> m_hWnd,
                cTipoInsumo,
                (LPSTR)(LPCSTR)strDir );

            DDefnMaCamDrvDir( m_hWnd, m_lpInfoObra, TRUE );
        }
    }
}

```

```

}
else
{
    MessageBeep ( MB_ICONSTOP );
}
endif
}endfunction

void CReporte::OnBorrar()
{
    CComboBox * pOpciones = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );
    if ( pOpciones->GetCount () )
    {
        BORRAREPOTE DdfBorraRepRegDesdeSalicic;

        //Cargamos la función que llena el combo Box
        //Con Todos los Reportes Predefinidos.
        DdfBorraRepRegDesdeSalicic =(BORRAREPOTE) GetProcAddress ( m_hReplib,
                                                                    "DdfBorraRepRegDesdeSalicic"
                                                                    );

        if ( DdfBorraRepRegDesdeSalicic )
        {
            CString strDir = "";
            char    cTipoInsumo;

            strDir = m_lpInfoObra->szDrive;
            strDir += m_lpInfoObra->szDir;
            strDir += '\\';
            fnObtenTipoRep ( cTipoInsumo );

            DdfBorraRepRegDesdeSalicic ( this -> m_hWnd,
                                        (HWND)pOpciones -> m_hWnd,
                                        (long)pOpciones -> GetCurSel(),
                                        cTipoInsumo,
                                        (LPSTR)(LPCSTR)strDir );

        }
        endif
    }
    else
    {
        MessageBeep ( MB_ICONSTOP );
    }
}endfunction

void CReporte::OnTabactivateReportes(UINT, int, CWnd*, LPVOID lpParams)
{
    UINT  TabToActivate = AFX_NUM_EVENTPARAM(int, lpParams);
    fnCargaInfoTabs ( TabToActivate );
}endfunction

void CReporte::OnClickCosto ()
{
    CButton * pCosto = (CButton*) GetDlgItem ( IDC_REP_COSTO );
    if ( pCosto->GetCheck () == 1 ) {
        // Tenemos que deshabilitar el checkbox de desglose
        GetDlgItem ( IDC_REP_PREC_DESG )->EnableWindow ( FALSE );
    }
}

```

```

    }endif
}endifunction
void CReporte::OnClickPrecio ()
{
    CButton * pPrecio = (CButton*) GetDlgItem ( IDC_REP_PRECIO );
    if ( pPrecio->GetCheck () == 1 ) {
        // Tenemos que deshabilitar el checkbox de desglose
        GetDlgItem ( IDC_REP_PREC_DESG )->EnableWindow ( TRUE );
    }endif

}endifunction
void CReporte::OnDisena()
{
    CComboBox * pboxRepRegist = (CComboBox * ) GetDlgItem ( IDC_REP_ORPCI_DEFINIDO );
    CInvocaRepEstudio dlgInvRepEstudio(IDD_MENSAJE_REPORTE);

    if ( 11 == m_nTipoDigDinamico )
    {
        MessageBeep ( MB_ICONSTOP );
        return;
    }endif

    int iReporteSel = pboxRepRegist->GetCurSel();
    if ( theApp.fnGetBool ( REP_ACTIVADO ) )
    {
        // RepEste ya se encuentra activo, por favor deshabilita
        DDfnDsppErrorUsuario ( m_hWnd,
                               theApp.m_hInstance,
                               theApp.m_szUsuario,
                               "",
                               IDST_ERR_SAICIC_REP,
                               MB_OK | MB_ICONINFORMATION );

        return;
    }endif

    HMODULE hMod = NULL;
    MODULEENTRY me;

    MemSet( &me, 0, sizeof (MODULEENTRY) );
    me.dwSize = sizeof (MODULEENTRY);

    if( hMod = ModuleFindName ( &me, "REPESTUD" ) ){
        if ( GetModuleUsage (hMod) >= 2 )
        {
            HWND hwndRepStudio;
            CHAR szTempBuffer[64];

            if ( hwndRepStudio = ::GetDesktopWindow () )
            {
                // Bueno ahora por favor el primer hijo
                for ( hwndRepStudio = ::GetTopWindow ( hwndRepStudio );
                    hwndRepStudio;
                    hwndRepStudio = ::GetNextWindow ( hwndRepStudio, GW_HWNDNEXT ) )
                {
                    ::GetWindowText ( hwndRepStudio, szTempBuffer, sizeof szTempBuffer );

                    if ( StrStr ( szTempBuffer, "SAICIC Rep Estudio" ) )

```



```

        (
            // Lo encontramos, bueno activalo
            ::ShowWindow ( hwndRepStudio, SW_SHOW );
            ::BringWindowToTop ( hwndRepStudio );
            break;
        )
    }
}

// Ya se encuentra corriendo el editor, por favor indicalos
return;

}
}
}

if ( CB_ERR != iReporteSel )
{
    CString strGrupo;
    CString strRepVer;
    CString strRepHor;

    strGrupo.LoadString( IDS_WINSAIREP );
    strRepVer.LoadString( IDS_REP_VERTICAL );
    strRepHor.LoadString( IDS_REP_HORIZONTAL );

    char szTempBuffer[256];
    char szSeccion [64];

    char cConcepto;

    fnObtenTipoRep ( cConcepto );

    //Obtiene la ubicación de los scrips que conforman el
    //reporte.

    wprintf ( szSeccion,
        "%c",
        cConcepto,
        (LPCTSTR)strGrupo );

    wprintf ( szTempBuffer,
        "%s",
        (LPCTSTR) strRepVer,
        iReporteSel );
    strRepVer = szTempBuffer;
    wprintf ( szTempBuffer,
        "%s",
        (LPCTSTR) strRepHor,
        iReporteSel );
    strRepHor = szTempBuffer;

    ::GetPrivateProfileString ( szSeccion,
        strRepVer,
        "",
        szTempBuffer,
        sizeof(szTempBuffer),
        REGISTRO_INI );
    dlgInvRepEstudio.m_strRepVer= szTempBuffer;

    ::GetPrivateProfileString ( (LPCSTR) szSeccion,
        (LPCSTR) strRepHor,
        "",
        szTempBuffer,
        sizeof(szTempBuffer),
        REGISTRO_INI );
    dlgInvRepEstudio.m_strRepHor= szTempBuffer;

    if ( dlgInvRepEstudio.m_strRepVer.GetLength()
        <<

```

```

        dlgInvRepEstudio.m_strRepHor.GetLength() )
    {
        dlgInvRepEstudio.m_ulOrientacion = OR_VERT_HOR;
    }
}

if ( dlgInvRepEstudio.m_strRepVer.GetLength()
    "
    !dlgInvRepEstudio.m_strRepHor.GetLength() )
{
    dlgInvRepEstudio.m_ulOrientacion = OR_VERTICAL;
}

if ( !dlgInvRepEstudio.m_strRepVer.GetLength()
    "
    dlgInvRepEstudio.m_strRepHor.GetLength() )
{
    dlgInvRepEstudio.m_ulOrientacion = OR_HORIZONTAL;
}

}

dlgInvRepEstudio.m_nTipoMensaje = IDS_INVOCAR_REPESTUDIO;
dlgInvRepEstudio.m_strMensaje.LoadString (IDS_INVOCAR_REPESTUDIO);
dlgInvRepEstudio.m_strRadio1.LoadString (IDS_RADIO1_REP_ACTUAL);
dlgInvRepEstudio.m_strRadio2.LoadString (IDS_RADIO2_REPEVO);

if ( IDOK == dlgInvRepEstudio.DoModal() )
{
    CDialog::OnOK ();

    // Antes de ejecutar al rep estudio por favor tenemos que verificar
    // la trayectoria a seguir

    UINT ulCodErr = WinExec ( dlgInvRepEstudio.m_strRepVer,
        SW_SHOW );

    if (ulCodErr < 32)
    {
        DdFnDspError( NULL,
            theApp.m_hInstance,
            theApp.m_szUsuario,
            IDS_NO_PUDO_CARGAR_REPESTUDIO );
    }
}

}

}

void CReporte::OnCancel ()
{
    DdFnMaCamDrvDir( m_hWnd, m_lpInfoObra, TRUE );
    CDialog::OnCancel ();
}

}

void CReporte::OnMueveArriba ()
{
    // Bueno obten el elemento que esta seleccionado
    CListBox *pList = (CListBox *) GetDlgItem ( IDC_LIST1 );
    int nElemento;
    CString strAbajo;
    int nValorAbajo;
}

```

Capítulo IV Desarrollo del Sistema

```
if ( LR_ERR != ( nElemento = pList->GetCurSel ( ) ) )
{
    // Bueno ahora por favor chequea sino es elemento cero
    if ! nElemento > 0 {
        // Bueno trae el texto que necesitamos
        pList->GetText ( nElemento, strAbajo );
        nValorAbajo = LOWORD ( pList->GetItemData ( nElemento ) );

        // Ahora por favor intercambia los datos
        pList->DeleteString ( nElemento );

        // Ya borramos los elementos, ahora por favor inserta lo nuevo
        nElemento = pList->InsertString ( nElemento - 1, strAbajo );
        pList->SetItemData ( nElemento,
            MAKELONG ( nValorAbajo, 0 ) );
        pList->SetCurSel ( nElemento );
    }
    else
    {
        MessageBeep ( MB_ICONSTOP );
    }
}
endif

}
else
{
    MessageBeep ( MB_ICONSTOP );
}
endif
}
endif

void CReporte::OnMoverAbajo ( )
{
    // Bueno obten el elemento que esta seleccionado
    CListBox *pList = (CListBox *) GetDlgItem ( IDC_LIST );
    int nElemento, nElementoBorrar;
    CString strAbajo;
    int nValorAbajo;

    if ( LR_ERR != ( nElemento = pList->GetCurSel ( ) ) )
    {
        // Bueno ahora por favor chequea sino es elemento cero
        if ( nElemento < pList->GetCount ( ) - 1 )
        {
            pList->GetText ( nElemento, strAbajo );
            nValorAbajo = LOWORD ( pList->GetItemData ( nElemento ) );
            nElementoBorrar = nElemento;

            // Bueno como es sabido tenemos que avanzar e insertar el dato
            // abajo del elemento siguiente
            nElemento = pList->InsertString ( nElemento + 1, strAbajo );
            pList->SetItemData ( nElemento,
                MAKELONG ( nValorAbajo, 0 ) );
            pList->SetCurSel ( nElemento - 1 );
            pList->DeleteString ( nElementoBorrar );
        }
    }
    else
    {

```

```

        MessageBeep ( MB_ICONSTOP );
    }endif
}
else
{
    MessageBeep ( MB_ICONSTOP );
}endif
}endifunction

void CReporte::OnCheckOrdena ()
{
    if ( ( (CButton * ) GetDlgItem ( IDC_CHECK2 ) )->GetCheck () )
    {
        ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_LIST1 ), TRUE );
        ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_MOVER_ARRIBA ), TRUE );
        ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_MOVER_ABAJO ), TRUE );
    }
    else
    {
        ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_LIST1 ), FALSE );
        ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_MOVER_ARRIBA ), FALSE );
        ::EnableWindow ( ::GetDlgItem ( m_hWnd, IDC_MOVER_ABAJO ), FALSE );
    }
}endif
}endifunction

int fnFuncionChecaNumEle (
    HWND m_hWnd,
    int nRegistro,
    LPMDIVentStruct m_lpInfoObra,
    int nNumElePermitidos,
    UINT nMensaje
)
{
    if ( S_OKAY == dt_reclock ( nRegistro,
        LR_SOLO_LEER,
        DDFnMaObtTarea( m_lpInfoObra ),
        CURR_DB ) )
    {
        DB_ADDR dbAddr;
        long lslot = 0;
        int nFile;

        if ( S_OKAY == dt_reclast ( nRegistro,
            DDFnMaObtTarea( m_lpInfoObra ),
            CURR_DB ) )
        {
            // OK prepara todo lo necesario
            dt_crget ( &dbAddr, DDFnMaObtTarea( m_lpInfoObra ), CURR_DB );
            d_decode_dba ( dbAddr, &nFile, &lslot );
        }
    }endif
    dt_recfree ( nRegistro,
        DDFnMaObtTarea( m_lpInfoObra ),
        CURR_DB );

    if ( lslot > nNumElePermitidos )
    {

```

Capítulo IV Desarrollo del Sistema

```
// Ahhhhh, usuario una versión demo solo contiene 30 anali-
// sis
DDfnDapyError ( m_hWnd,
                theApp.m_hInstance,
                theApp.m_szUsuario,
                nMensaje );

return FALSE;

}endif

return TRUE;

}endif

return FALSE;

}endifunction

/*
FUNCION: fnInicializaFSR ( LPCSTR lpszSeccion, UINT uiOpcionActual)
PARAMETROS: LPCSTR lpszSeccion contiene la sección de donde se lee la
información para configurar el diálogo de reportes de
FSR
            UINT uiOpcionActual indica la opción que se esta invocando
actualmente

DESCRIPCION: Esta función se encarga de de inicializar las opciones de
insumos en los reportes. Es muy importante recalcar que en esta
versión se obtiene la información de archivo
*/
void CReporte::fnInicializaFSR ( LPCSTR lpszSeccion,
                                UINT uiOpcionActual
                                )
{
    CString strRespuesta = "";
    int iTipoOpcion;
    CComboBox * pCombo;

    pCombo = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );

    //Borra el contenido de la lista con el fin de
    //cargar los reportes registrados del FSR
    //que se está especificando.
    fnLargaRepRegistrados ( uiOpcionActual );

    iTipoOpcion = theApp.GetProfileInt ( lpszSeccion,
                                        CAMPO_ENVIAR REP,
                                        REP_PANTALLA );

    switch ( iTipoOpcion )
    {
        case REP_PANTALLA:
            ( (CButton *) GetDlgItem ( IDC_REP_PANTALLA ) )->SetCheck( TRUE );
            ( (CButton *) GetDlgItem ( IDC_REP_IMPRESORA ) )->SetCheck( FALSE );
            ( (CButton *) GetDlgItem ( IDC_REP_ARCH ) )->SetCheck( FALSE );
            GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( FALSE );
            GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( FALSE );
            break;

        case REP_IMPRESORA:
            ( (CButton *) GetDlgItem ( IDC_REP_IMPRESORA ) )->SetCheck( TRUE );
            ( (CButton *) GetDlgItem ( IDC_REP_PANTALLA ) )->SetCheck( FALSE );
            ( (CButton *) GetDlgItem ( IDC_REP_ARCH ) )->SetCheck( FALSE );
            GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( FALSE );
            GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( FALSE );
            break;

        case REP_ARCHIVO:
            ( (CButton *) GetDlgItem ( IDC_REP_ARCH ) )->SetCheck( TRUE );
            ( (CButton *) GetDlgItem ( IDC_REP_PANTALLA ) )->SetCheck( FALSE );
            ( (CButton *) GetDlgItem ( IDC_REP_IMPRESORA ) )->SetCheck( FALSE );
    }
}
```

```

        GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( TRUE );
        break;
    }
}

strRespuesta = theApp.GetProfileString ( lpszSeccion,
                                        CAMPO_PREDEFINIDO,
                                        strRespuesta );

if ( !strRespuesta.IsEmpty () )
{
    // Se tiene que acceder la información como lo deseamos
    iTipoOpcion = pCombo->FindStringExact ( -1, strRespuesta );

    if ( iTipoOpcion != CB_ERR )
    {
        pCombo->SetCurSel ( iTipoOpcion );
    }
}

}

}

}

/*
FUNCION: fnInicializaInsumos( LPCSTR lpszSeccion, UINT uiOpcionActual)
PARAMETROS: LPCSTR lpszSeccion contiene la sección de donde se lee la
información para configurar el diálogo de reportes de
insumos
            UINT uiOpcionActual indica la opción que se está invocando
            actualmente
DESCRIPCION: Esta función se encarga de de inicializar las opciones de
insumos en los reportes. Es muy importante recalcar que en esta
versión se obtiene la información de archivo
*/

void CReporte::fnInicializaInsumos ( LPCSTR lpszSeccion,
                                   UINT uiOpcionActual
                                   )
{
    CComboBox * pOrden = (CComboBox *) GetDlgItem ( IDC_REP_ORDENA );
    int iTipoOpcion = 0;

    //Borra el contenido de la lista con el fin de
    //cargar los reportes registrados de del tipo
    //de insumo que se está especificando.

    fnCargaRepRegistrados ( uiOpcionActual );

    //Elegimos el tipo de orden en que será obtenido el reporte
    iTipoOpcion = theApp.GetProfileInt ( lpszSeccion,
                                       CAMPO_SELECCION,
                                       0 );

    pOrden->SetCurSel ( iTipoOpcion );

    fnInicializaComun ( lpszSeccion );

    ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->SetCheck (
        theApp.GetProfileInt ( lpszSeccion,
                              CAMPO_ORDENAMIENTO,
                              0 ) );
}

}

/*
FUNCION: fnInicializaComun ( LPCSTR lpszSeccion, CString strRespuesta )
PARAMETROS: LPCSTR lpszSeccion indica la sección de donde se obtiene la

```

```

                                Informacion para inicializar el dialogo
*/
void CReporte::fnInicializaComm ( LPCSTR lpszSeccion )
(
    *
    CString  strRespuesta = "";
    INT      iTipoOpcion = 0;
    char     szSI[] = "SI";
    CComboBox * pCombo;

    pCombo = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );

    strRespuesta = theApp.GetProfileString ( lpszSeccion,
                                           CAMPO_PREDEFINIDO,
                                           strRespuesta );

    // Checa si tiene info
    // Obtenemos cual es el reporte definido por default
    if ( !strRespuesta.IsEmpty () )
    {

        // Se tiene que acceder la información como lo deseamos
        iTipoOpcion = pCombo->FindStringExact ( -1, strRespuesta );

        if ( !iTipoOpcion != CB_ERR )
        {

            pCombo->SetCurSel ( iTipoOpcion );

        }
    }

    strRespuesta = theApp.GetProfileString ( lpszSeccion,
                                           CAMPO_RANGO_TODAS,
                                           szSI );

    if ( strRespuesta.CompareNoCase ( szSI ) ) {

        //Se tiene elegido el campo de Rango
        ( CButton *) GetDlgItem ( IDC_REP_RANGO )->SetCheck( TRUE );
        ( CButton *) GetDlgItem ( IDC_REP_TODAS )->SetCheck( FALSE );
        GetDlgItem ( IDC_REP_INICIAL )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_FINAL )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_CLAVE_INICIAL )->EnableWindow ( TRUE );
        GetDlgItem ( IDC_REP_CLAVE_FINAL )->EnableWindow ( TRUE );

    }
    else {

        //Se tiene elegido el campo de todas las claves
        ( CButton *) GetDlgItem ( IDC_REP_TODAS )->SetCheck( TRUE );
        ( CButton *) GetDlgItem ( IDC_REP_RANGO )->SetCheck( FALSE );
        GetDlgItem ( IDC_REP_INICIAL )->EnableWindow ( FALSE );
        GetDlgItem ( IDC_REP_FINAL )->EnableWindow ( FALSE );
        GetDlgItem ( IDC_REP_CLAVE_INICIAL )->EnableWindow ( FALSE );
        GetDlgItem ( IDC_REP_CLAVE_FINAL )->EnableWindow ( FALSE );

    }

    //Tenemos que leer el rango inicial, leemos clave inicial y clave
    //final
    strRespuesta = theApp.GetProfileString ( lpszSeccion,
                                           CAMPO_CLAVE_INT );
    GetDlgItem ( IDC_REP_INICIAL )->SetWindowText ( strRespuesta );

    strRespuesta = theApp.GetProfileString ( lpszSeccion,
                                           CAMPO_CLAVE_FIN );
    GetDlgItem ( IDC_REP_FINAL )->SetWindowText ( strRespuesta );

    iTipoOpcion = theApp.GetProfileInt ( lpszSeccion,
                                       CAMPO_ENVIAR_REP,
                                       REP_PARTIAL );

    switch ( iTipoOpcion ) {

```

```

case REP_PANTALLA:
    { (CButton *) GetDlgItem ( IDC_REP_PANTALLA )->SetCheck( TRUE );
      (CButton *) GetDlgItem ( IDC_REP_IMPRESORA )->SetCheck( FALSE );
      (CButton *) GetDlgItem ( IDC_REP_ARCH )->SetCheck( FALSE );
      GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( FALSE );
      GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( FALSE );
      break;

case REP_IMPRESORA:
    { (CButton *) GetDlgItem ( IDC_REP_IMPRESORA )->SetCheck( TRUE );
      (CButton *) GetDlgItem ( IDC_REP_PANTALLA )->SetCheck( FALSE );
      (CButton *) GetDlgItem ( IDC_REP_ARCH )->SetCheck( FALSE );
      GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( FALSE );
      GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( FALSE );
      break;

case REP_ARCHIVO:
    { (CButton *) GetDlgItem ( IDC_REP_ARCH )->SetCheck( TRUE );
      (CButton *) GetDlgItem ( IDC_REP_PANTALLA )->SetCheck( FALSE );
      (CButton *) GetDlgItem ( IDC_REP_IMPRESORA )->SetCheck( FALSE );
      GetDlgItem ( IDC_REP_ARCHIVO )->EnableWindow ( TRUE );
      GetDlgItem ( IDC_REP_REP_ARCH )->EnableWindow ( TRUE );
      break;

}endswitch

strRespuesta = theApp.GetProfileString ( lpzSeccion,
                                        CAMPO_REP_ARCH,
                                        "" );
GetDlgItem ( IDC_REP_ARCHIVO )->SetWindowText ( strRespuesta );
}endfunction

/*
FUNCION: fnSalvaInfoFSR( LPCSTR lpzSeccion )
PARAMETROS: LPCSTR lpzSeccion indica el nombre de la seccion donde
            seran grabados los datos de dialogo
DESCRIPCION: Esta función se encarga de salvar las opciones de materiales
            que se encuentran en el dialogo actual hacia el archivo de inicializa-
            ción
*/
void CReporte::fnSalvaInfoFSR ( LPCSTR lpzSeccion )
{
    CComboBox * pCombo;
    CString strRespuesta = "";
    INT iTipoOpcion;

    pCombo = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );

    if ( pCombo->GetCount () )
    {
        if ( CB_ERR != ( iTipoOpcion = pCombo->GetCurSel () ) )
        {
            pCombo->GetLBText ( iTipoOpcion, strRespuesta );
        }
    }
}endfunction

// Obtenemos la selección que se encuentra dentro del combobox de
// reportes
theApp.WriteProfileString ( lpzSeccion,
                           CAMPO_PREDEFINIDO,
                           strRespuesta );

// Grabamos la información de los datos comunes a todos los reportes
if ( ( (CButton *) GetDlgItem ( IDC_REP_PANTALLA )->GetCheck() == 1 ) )

```


Capítulo IV Desarrollo del Sistema

```
        iTipoOpcion = REP_PANTALLA;
    }
    else {
        if ( ( CButton * ) GetDlgItem ( IDC_REP_IMPRESORA ) ->GetCheck() == 1 ) {
            iTipoOpcion = REP_IMPRESORA;
        }
        else {
            iTipoOpcion = REP_ARCHIVO;
        }
    }
}
#endif

GetDlgItem ( IDC_REP_ARCHIVO)->GetWindowText ( strRespuesta );
theApp.WriteProfileString ( lpszSeccion, CAMPO_REP_ARCH,
                           strRespuesta );

theApp.WriteProfileInt ( lpszSeccion,
                        CAMPO_ENVIAR_REP,
                        iTipoOpcion );

}endfunction

/*
FUNCION: fnSalvaInfoInsumos( LPCSTR lpszSeccion )
PARAMETROS: LPCSTR lpszSeccion indica el nombre de la seccion donde
            seran gradados los datos del dialogo
DESCRIPCION: Esta función se encarga de salvar las opciones de materiales
            que se encuentran en el dialogo actual hacia el archivo de inicializa-
            ción
*/
void CReporte::fnSalvaInfoInsumos ( LPCSTR lpszSeccion )
{
    CComboBox * pCombo;
    CString      strRespuesta = "";
    int          nCurSel;

    pCombo = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );
    nCurSel = pCombo->GetCurSel();

    if ( pCombo->GetCount () && nCurSel != CB_ERR )
    {
        pCombo->GetLBText ( nCurSel,
                           strRespuesta );

        theApp.WriteProfileString ( lpszSeccion,
                                    CAMPO_PREDEFINIDO,
                                    strRespuesta );

        // Obtenemos la información que se encuentra dentro del combobox de
        // ordenado por
        pCombo = ( CComboBox * ) GetDlgItem ( IDC_REP_ORDENA );
        theApp.WriteProfileInt ( lpszSeccion,
                                CAMPO_SELECCION,
                                pCombo->GetCurSel () );

        // Grabamos la información de los datos comunes a todos los reportes
        fnGrabaCamposComunes ( lpszSeccion );
    }
}

// Checa cual es la variable para orden del reporte a imprimir
theApp.WriteProfileInt ( lpszSeccion,
                        CAMPO_ORDENAMIENTO,
                        ( CButton * ) GetDlgItem ( IDC_CHECK1 ) ->GetCheck () );
```

ndfunction

FUNCION: fnSalvaInfoPresupue (LPCSTR lpszSeccion)
 PARAMETROS: LPCSTR lpszSeccion indica el nombre de la seccion donde serán grabados los datos del dialogo
 DESCRIPCION: Esta función se encarga de salvar las opciones de presupuestos. Esta rutina con mayor tiempo puede ser la misma a la de los análisis

id CReporte::fnSalvaInfoPresupue (LPCSTR lpszSeccion)

```

CComboBox * pOpciones = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );
CComboBox * pFormatos = (CComboBox *) GetDlgItem ( IDC_REP_FORMATO );
HINSTANCE hInstance = AfxGetInstanceHandle();
char szSI[] = "SI";
CString strRespuesta = "";
INT iTipoOpcion = 0;
int nCurSel;

nCurSel = pOpciones->GetCurSel();

// Salvamos el tipo de reporte que se encuentra elegido
if ( pOpciones->GetCount () && nCurSel != CB_ERR )
{
    pOpciones->GetLBText ( nCurSel,
                        strRespuesta );

    // Obtenemos la selección que se encuentra dentro del combobox de
    // reportes
    theApp.WriteProfileString ( lpszSeccion,
                              CAMPO_PREDEFINIDO,
                              strRespuesta );

    // Salvamos la información del formato de sobre costo elegido
    iTipoOpcion = pFormatos->GetCurSel();

    if ( CB_ERR != iTipoOpcion ) {
        pFormatos->GetLBText ( iTipoOpcion, strRespuesta );
        ::WritePrivateProfileString ( ID_SECC_FACTORES,
                                    ID_CAMPO_FORMATO,
                                    strRespuesta,
                                    FACTORES_INI );
    }

}

// Obtenemos la información sobre el combobox de nivel de presupuesto
pOpciones = ( CComboBox *) GetDlgItem ( IDC_NIVEL );
theApp.WriteProfileInt ( lpszSeccion,
                       CAMPO_SELECCION,
                       pOpciones->GetCurSel () );

// Grabamos información si los reportes serán impresos a costo o a precio
fnInfoCosto ( lpszSeccion );

// Grabamos la información de los datos comunes
fnGrabaCamposComunes ( lpszSeccion );

}

ndfunction

```

FUNCION: fnSalvaInfoAnalisis(LPCSTR lpszSeccion)
 PARAMETROS: LPCSTR lpszSeccion indica el nombre de la seccion donde serán grabados los datos del dialogo
 DESCRIPCION: Esta función se encarga de salvar las opciones de analisis que se encuentran en el dialogo actual hacia el archivo de inicialización

Capítulo IV Desarrollo del Sistema

```
*/
void CReporte::fnSalvaInfoAnalisis ( LPCSTR lpszSeccion )
{
    CComboBox * pOpciones = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );
    CComboBox * pFormatos = (CComboBox *) GetDlgItem ( IDC_REP_FORMATO );
    HINSTANCE hInstance = AfxGetInstanceHandle();
    char szSI[] = "SI";
    CString strRespuesta = "";
    INT      iTipoOpcion = 0;
    int      nCurSel;

    nCurSel = pOpciones->GetCurSel();

    // Salvamos el tipo de reporte que se encuentra elegido
    if ( pOpciones->GetCount () && nCurSel != CB_ERR )
    {
        pOpciones->GetLBText ( nCurSel, strRespuesta );
        // Obtenemos la selección que se encuentra dentro del combobox de
        // reportes
        theApp.WriteProfileString ( lpszSeccion,
                                   CAMPO_PREDEFINIDO,
                                   strRespuesta );

        //Salvamos la información del formato de sobre costo elegido
        iTipoOpcion = pFormatos->GetCurSel();

        if ( CB_ERR != iTipoOpcion ) {
            pFormatos->GetLBText ( iTipoOpcion, strRespuesta );
            ::WritePrivateProfileString ( ID_SECC_FACTORES,
                                         ID_CAMPO_FORMATO,
                                         strRespuesta,
                                         FACTORES_INI );
        }

        // Obtenemos la información que se encuentra dentro del combobox de
        // ordenado por
        pOpciones = ( CComboBox *) GetDlgItem ( IDC_REP_ORDENA );
        theApp.WriteProfileInt ( lpszSeccion,
                               CAMPO_SELECCION,
                               pOpciones->GetCurSel ( ) );

        // Grabamos información si los reportes serán impresos a costo o a precio
        fnInfoCosto ( lpszSeccion );

        //Grabamos la información de los datos comunes
        fnGrabaCamposComunes ( lpszSeccion );
    }

    theApp.WriteProfileInt ( lpszSeccion,
                            CAMPO_ORDENAMIENTO,
                            ( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->GetCheck ( ) );
}

/*
FUNCION: fnGrabaCamposComunes ( LPCSTR lpszSeccion, CString & strRespuesta )
PARAMETROS: LPCSTR lpszSeccion indica la seccion que sera salvada dentro
del archivo de inicializacion con las opciones de reportes
comunes

CString &strRespuesta CString Auxiliar.

DESCRIPCION: Esta funcion se encarga de salvar la información de los campos
comunes dentro de los dialogos de reportes
*/
void CReporte::fnGrabaCamposComunes ( LPCSTR lpszSeccion )
{
```

```

INT      iTipoOpcion = 0;
char     szSI[] = "SI";
CString  strRespuesta = "";

// Guardamos la información sobre impresión de claves
if ( ( (CButton *) GetDlgItem ( IDC_REP_TODAS )->GetCheck() == 1 ) {

    theApp.WriteProfileString ( lpszSeccion, CAMPO_RANGO_TODAS,
                               szSI );

}
else {

    theApp.WriteProfileString ( lpszSeccion, CAMPO_RANGO_TODAS,
                               "NO" );

}

}endif

GetDlgItem ( IDC_REP_INICIAL )->GetWindowText ( strRespuesta );
theApp.WriteProfileString ( lpszSeccion,
                           CAMPO_CLAVE_INI,
                           strRespuesta );
GetDlgItem ( IDC_REP_FINAL )->GetWindowText ( strRespuesta );
theApp.WriteProfileString ( lpszSeccion,
                           CAMPO_CLAVE_FIN,
                           strRespuesta );

if ( ( (CButton *) GetDlgItem ( IDC_REP_PANTALLA ) )->GetCheck() == 1 ) {

    iTipoOpcion = REP_PANTALLA;

}
else {

    if ( ( (CButton *) GetDlgItem ( IDC_REP_IMPRESORA ) )->GetCheck() == 1 ) {

        iTipoOpcion = REP_IMPRESORA;

    }
    else {

        iTipoOpcion = REP_ARCHIVO;

    }

}endif

}endif

GetDlgItem ( IDC_REP_ARCHIVO )->GetWindowText ( strRespuesta );
theApp.WriteProfileString ( lpszSeccion, CAMPO_REP_ARCHI,
                           strRespuesta );

theApp.WriteProfileInt ( lpszSeccion,
                        CAMPO_ENVIAR_REP,
                        iTipoOpcion );

}endifunction

/*
FUNCION: fnInicializaAnálisis( LPCSTR lpszSeccion )

PARAMETROS: LPCSTR lpszSeccion contiene la sección de donde se lee la
información para configurar el diálogo de reportes de
análisis

DESCRIPCION: Esta función se encarga de de inicializar las opciones de
Análisis en los reportes. Es muy importante recalcar que en esta
versión se obtiene la información de archivo

*/
void CReporte::fnInicializaAnálisis ( LPCSTR lpszSeccion,
                                     UINT uiOpcionActual
                                     )
{

```

Capítulo IV Desarrollo del Sistema

```
CComboBox * pOrden = (CComboBox *) GetDlgItem ( IDC_REP_ORDENA );
CComboBox * pFormatos = (CComboBox *) GetDlgItem ( IDC_REP_FORMATO );
CString strRespuesta = "";
UINT nEleSel;

//Agregamos los elementos en el combo de tipo de reporte
fnCargaRepRegistrados ( uiOpcionActual );

//Elegimos el tipo de orden en que será obtenido el reporte
if ( pOrden->GetCount () )
{
    nEleSel = theApp.GetProfileInt ( lpszSeccion, CAMPO_SFLECCION, 0 );
    pOrden->SetCurSel ( nEleSel );
}
}endif

CFactorSobreCosto::fnEligeFactorSobreCosto ( pFormatos, strRespuesta );
fnIniciaCosto ( lpszSeccion );
fnInicializaComun ( lpszSeccion );

( (CButton *) GetDlgItem ( IDC_CHECK1 ) )->SetCheck (
    theApp.GetProfileInt ( lpszSeccion,
        CAMPO_CHECKEAMIENTO,
        0 ) );
}endifunction

/*
FUNCION: fnInicializaPresupuesto( LPCSTR lpszSeccion )
PARAMETROS: LPCSTR lpszSeccion contiene la sección de donde se lee la
información para configurar el diálogo de reportes de
análisis
DESCRIPCION: Esta función se encarga de de inicializar las opciones de
presupuestos en los reportes. Es muy importante recalcar que en esta
versión se obtiene la información de archivo
*/
void CReporte::fnInicializaPresupuesto ( LPCSTR lpszSeccion,
    UINT uiOpcionActual )
{
    CComboBox * pNivel = (CComboBox *) GetDlgItem ( IDC_NIVEL );
    CComboBox * pFormatos = (CComboBox *) GetDlgItem ( IDC_REP_FORMATO );
    CString strRespuesta = "";

    int nNivelPresu = fnObtenNivelPresupuesto ( this,
        strRespuesta,
        IDC_NIVEL,
        DEF_MyObtTarea( m_pInfoOtra ) );

    if ( nNivelPresu > 0 )
    {
        pNivel->SetCurSel ( nNivelPresu - 1 );
    }
}endif

//Agregamos los elementos en el combo de tipo de reporte
fnCargaRepRegistrados ( uiOpcionActual );
CFactorSobreCosto::fnEligeFactorSobreCosto ( pFormatos, strRespuesta );
fnIniciaCosto ( lpszSeccion );
fnInicializaComun ( lpszSeccion );
}endifunction

// FUNCION fnCargaRepRegistrados ( CComboBox * pReportes, UINT uiOpcion )
// DESCRIPCION: Esta función se encarga de llenar el combobox con las opciones
// que se encuentran disponibles para elegir como reportes dentro
```

```

una opción de SAICIC Windows
PARAMETROS:
  CComboBox *pReportes es el apuntador al combobox que será llenado
                    con las diferentes opciones.
  UINT      uiOpcionActual es la opción de la cuál se desean llenar
                    sus reportes dentro del combobox.

void CReporte::fnCargaRepRegistrados ( UINT uiOpcionActual )

char      cTipoInsumo;
CString  strDir = "";
CARGACOMBO fnCargaReportesAlCombo;
CComboBox * pOpciones = (CComboBox *) GetDlgItem ( IDC_REP_OPCI_DEFINIDO );

pOpciones->ResetContent();

//Cargamos la función que llena el combo Box
//Con Todos los Reportes Predefinidos.
fnCargaReportesAlCombo = (CARGACOMBO) GetProcAddress (
                                m_hReplib,
                                "DDfnCargaReportesAlCombo" );

if ( fnCargaReportesAlCombo )
{

    // encontramos desde que directorio se carga la información
    fnObtenTipoRep ( cTipoInsumo );
    strDir = cTipoInsumo;

    strDir = m_lpInfoObra->szDirivo;
    strDir += m_lpInfoObra->szDir;
    * strDir += '\\';

    fnCargaReportesAlCombo ( (HWND)pOpciones->m_hWnd,
                             cTipoInsumo,
                             (LPSTR)(LPCSTR)strDir );

}

}

endif
endfunction

/ FUNCION: fnLlenaOpciOrdenaInsa ()
/ DESCRIPCION: Esta función se encarga de llenar el combobox de ordenamiento
/ con las opciones de ordenamiento
void CReporte::fnLlenaOpciOrdenaInsa ()

CComboBox * pOrden = (CComboBox *) GetDlgItem ( IDC_REP_ORDEMA );

pOrden->AddString ( TIPO_ORDEN_CLAVE );
pOrden->AddString ( TIPO_ORDEN_COSTO );
pOrden->AddString ( TIPO_ORDEN_UNIDAD );
pOrden->AddString ( TIPO_ORDEN_DESCRIP );
pOrden->AddString ( TIPO_ORDEN_ALTAS );

}

endif
endfunction

void CReporte::fnInfoCosto ( LPCSTR lpszSeccion )
{

// Tenemos que indicar como se imprimiran los reportes
CButton * pCosto = (CButton *) GetDlgItem ( IDC_REP_COSTO );

if ( pCosto->GetCheck () == 1 ) {

    // El radiobutton de costo unitario está checado
    theApp.WriteProfileString ( lpszSeccion,
                              CAMPO_COSTO_UNIT,
                              "SI" );
}
}

```



```

}endfunction

void CReporte::fnObtenTipoRep ( char &cTipo )
{
    switch ( m_nTipoDlgDinamico ) {
        case 0: // Materiales
            cTipo = TIPO_MAT;
            break;

        case 1:
            cTipo = TIPO_OBRAMANO;
            break;

        case 2:
            cTipo = TIPO_MAQUINARIA;
            break;

        case 3:
            cTipo = TIPO_SUBCONTRATO;
            break;

        case 4:
            cTipo = TIPO_FSR;
            break;

        case 5:
            cTipo = IDENT_PRESUPUE_TIPO;
            break;

        case 6:
            cTipo = TIPO_BASICO;
            break;

        case 7:
            cTipo = TIPO_INTEGRADO;
            break;

        case 8:
            cTipo = TIPO_PARTIDA;
            break;

        case 9:
            cTipo = TIPO_CAPITULO;
            break;

        case 10:
            cTipo = TIPO_EXTRA;
            break;
    }
}endswitch

}endfunction

void CReporte::fnSalvaConfiguracion ()
{
    char    szTipo [] = { 0, 0, 0 };
    INT     nNumEle;
    UINT    nValor;
    CString strOrden;
    CListBox *pList = (CListBox *) GetDlgItem ( IDC_LIST1 );

    // Elige el orden que el usuario dara a los elementos del análisis
    nNumEle = pList->GetCount ();

    for ( int nContador = 0;
          nContador < nNumEle;
          nContador++
        )
    {

```


Capítulo IV Desarrollo del Sistema

```
// Tienes que traer el itemdata del elemento actual
nValor = LOWORD ( pList->GetItemData ( nContador ) );
szTipo[1] = HIBYTE ( nValor );
szTipo[0] = LOBYTE ( nValor );
strOrden += szTipo;

}endif

// Bueno ahora graba por favor la lista de orden
theApp.WriteProfileString ( SECC_REP_CONFIG,
                           CAMPO_APARECE_ELE,
                           strOrden );

theApp.WriteProfileInt ( SECC_REP_CONFIG,
                        CAMPO_RESPETA_CAP,
                        ( (CButton *) GetDlgItem ( IDC_CHECK2 ) )->GetCheck ( ) );
theApp.WriteProfileInt ( SECC_REP_CONFIG,
                        CAMPO_ORDEN_COMPO_ALIAS,
                        ( (CButton *) GetDlgItem ( IDC_CHECK3 ) )->GetCheck ( ) );

// Utilizar o no la moneda base para el control de monedas
theApp.WriteProfileInt (
    SECC_REP_CONFIG,
    CAMPO_MONEDA,
    ( (CButton *) GetDlgItem ( IDC_RADIO3 ) )->GetCheck ( ) );

}endifunction
```

1.2.2 INTEGRACIÓN DE APLICACIONES CON TECNOLOGÍA DE ORIENTACIÓN DE OBJETOS

Una de las ventajas que ofrece la tecnología orientada a objetos, es la interacción de diferentes aplicaciones a través de los denominados 'objetos', de forma que al incluir estas características al sistema de precios unitarios se busca una integración a un modelo de automatización global que permita la participación de todas las áreas involucradas, tanto administrativas como técnicas. El mercado actual debe orientarse al uso de documentos compuestos, que contengan datos procesados por diferentes aplicaciones pero que representen un mismo documento para el usuario final. Con ello el usuario tendrá las utilerías y aplicaciones específicas para cada área o cada proceso que deba realizarse, sin embargo para que esto sea posible, es necesario que todas las aplicaciones cumplan con los requerimientos mínimos de orientación a objetos, de forma que cualquier otra aplicación tenga acceso a la información procesada.

Existe una gran variedad de aplicaciones que pueden interactuar con el sistema de precios unitarios, es decir, hay aún muchas ventajas por incluir en el sistema, las cuales deberán realizarse de acuerdo a las prioridades definidas por los usuarios finales.

4.2.2.1 PLANEACIÓN DE PROYECTOS CON MS-PROJECT

Se requiere de un módulo de Planeación de Proyectos dentro del sistema de precios unitarios, para ello se ha decidido la utilización del programa MS_PROJECT, por sus características de facilidad de uso y su cumplimiento de estándares para el ambiente

Windows, por lo tanto el realizar un módulo especial tomaría bastante tiempo de desarrollo y no podría ofrecer las ventajas que actualmente tiene esta aplicación, por estas razones se hace necesaria la interacción de las dos aplicaciones de forma que un usuario constructor pueda realizar su trabajo de manera práctica y sencilla manejando los conceptos a los cuales está acostumbrado.

La transferencia de información se lleva a cabo de la siguiente forma:

Dentro del presupuesto se tienen normalmente las partidas, las cuales pasan a formar conceptos en planeación y cada componente de las partidas pasa a ser una actividad dejando lista la información para que el usuario proporcione las duraciones correspondientes así como los responsables asignados y la secuencia de ejecución.

En Project es posible la elaboración de plantillas que permiten ajustar las posiciones de las columnas, los resultados, los nombres tal y como el usuario los necesite, por lo tanto se han elaborado un conjunto de plantillas para que el constructor aprenda a utilizar la aplicación si fuese exclusivamente para construcción.

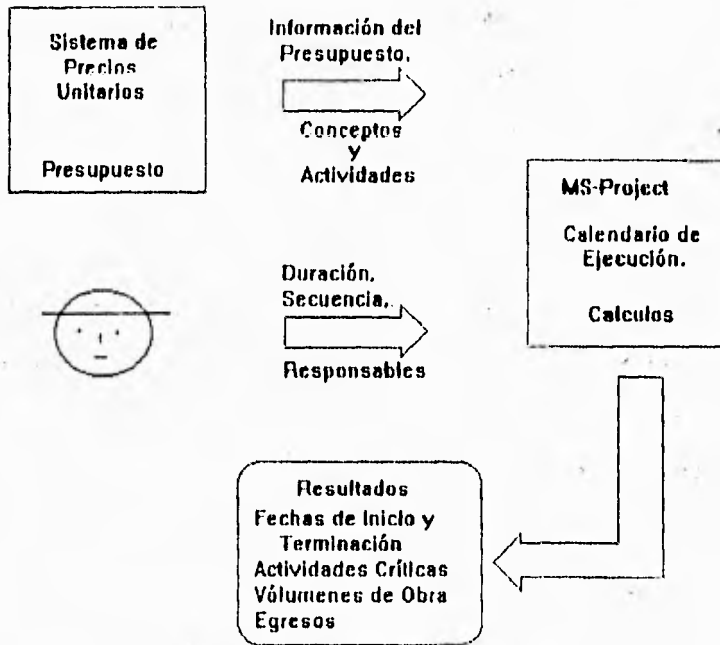


Fig. 4.2.2.1 Flujo de información entre el Sistema de P.U. y MS-Project

En el diagrama se puede apreciar el flujo de información para obtener una aproximación de la planeación del proyecto, el usuario podrá elegir cuales son sus actividades, en un presupuesto típico los integrados son los que se convierten en actividades, y las partidas pasan a ser conceptos, pero dependiendo del nivel de detalle al cual el usuario quiera hacer su planeación, podrá traer la información adicional como las cuadrillas o actividades básicas.

Una vez copiada la información en Project, el usuario deberá asignar de acuerdo a su experiencia las duraciones para cada actividad, las cantidades totales a ejecutar han sido copiadas del presupuesto así como los precios de cada actividad. El usuario final podrá agregar de manera opcional los responsables de cada actividad con el fin de tener una mejor información de las actividades de cada equipo durante la ejecución de la obra.

El orden en que deben ejecutarse las actividades también será proporcionado por el usuario, en caso de ser necesario el uso de traslape de actividades, el usuario deberá indicarlo para que el cálculo considere estos factores.

Una vez capturada esta información, desde Project podrá obtenerse la siguiente información:

- Calendario de actividades.
- Fecha de Inicio y Terminación del Proyecto (Programado)
- Volumen de Obra que debe cubrirse en cada periodo.
- Programa Egresos por Semana.
- Programa Egresos por Mes.
- Fecha de Inicio y Fin para cada actividad.
- Diagrama de Gantt.
- Diagrama de Pert.
- Explosión de Insumos en el Tiempo.
- Programa de Compras.
- Programa Financiero.

Toda esta información se presenta bajo la terminología utilizada en construcción y con la información que comúnmente es necesario entregar a las dependencias gubernamentales. Por las características propias de Project se podrán generar nuevos reportes que cubran cualquier necesidad específica.

Como los catálogos de insumos se encuentran dentro del sistema de precios unitarios, es de suma importancia que Project consulte esta información y así poder generar el reporte de explosión en el tiempo, la cual es de suma importancia para preveer las compras y almacenamiento de materiales.

Una vez iniciada la ejecución de obra, es posible capturar los avances reales para así establecer un reporte comparativo del comportamiento real contra lo programado y a su vez, tomar acciones de control si es necesario o sirve de experiencia para la elaboración de futuros programas. Con la información de avances reales se pueden generar todos los reportes que el usuario desee.

A continuación se presentan algunas de las plantillas requeridas para el manejo de la información en la planeación del proyecto, con el objetivo de mostrar la sencillez y la adaptación para la industria de la construcción que puede lograrse con estas herramientas.

4.3 SISTEMAS DE AYUDAS

4.3.1 BASES PARA EL DISEÑO DEL SISTEMA DE AYUDAS DE EL SISTEMAS DE PRECIOS UNITARIOS PARA WINDOWS

La ayuda en línea de una aplicación provee al usuario de una herramienta que le facilita entender y asimilar rápidamente el producto, por lo cual el diseño de la misma debe ser pensado en el usuario final.

Este documento contempla los siguientes puntos:

- Características de las ayudas
- Pasos a seguir para la creación de un sistema de ayudas
- Planeación del sistema de ayudas.

CARACTERÍSTICAS DE LA AYUDA EN LÍNEA DE UNA APLICACIÓN

1. Las ayudas deben contar con listas alfabéticas de los temas más importantes de la aplicación, sin proporcionar más de dos listas de temas para obtener información.
2. Los temas disponibles deben cubrir los diferentes aspectos de la aplicación, en forma ordenada, para proporcionar una referencia rápida de piezas específicas de información.
3. No deben proporcionar exceso de información sobre un tema específico, pues el usuario podría perderse. La información proporcionada de un tema debe ser concreta y clara para ubicar al usuario en un tiempo corto por lo que se deben utilizar párrafos breves.
4. Consistencia en la navegación de la información; la apariencia de los temas debe ser la misma en todo el sistema de ayuda, debe haber similitud en el uso de títulos, tipo, tamaño, y color de letra, y respetar la posición del texto

5. No enviar mensajes dentro de la ayuda.
6. Utilizar un lenguaje adecuado al tipo de usuario de la aplicación; no utilizar un lenguaje sofisticado.

PASOS A SEGUIR PARA CREAR UN SISTEMA DE AYUDAS EN UNA APLICACIÓN WINDOWS

1. Obtener información de los temas de ayuda
2. Planear el sistema de ayudas
3. Escribir los textos para cada tema de ayuda
4. Formatear el archivo de texto por medio de códigos de control para realizar saltos y definiciones de temas, asignar contextos, títulos, secuencia de temas, llaves para búsqueda y construir etiquetas; todo esto sirve para indicar el movimiento que tendrá el usuario en el sistema de ayuda (el archivo así generado en un archivo con extensión **.RTF** rich text format).
5. Crear el archivo del proyecto (con extensión **.HPJ**), con información que el compilador de ayudas de Windows necesita para construir el archivo de recursos.
En este archivo de proyecto se indican los archivos con los temas de ayuda que serán incluidos, el índice principal de ayuda; el título del sistema de ayudas, y otras opciones.
6. Construir el archivo de recursos de la ayuda (con extensión **.HLP**), a partir del archivo de proyecto, el comando utilizado para tal efecto es:
hc archivo.HPJ
El archivo generado por el compilador es un archivo con el mismo nombre del proyecto y extensión **.HPL**, por lo que en este ejemplo el archivo de recursos de ayuda construido es **archivo.HLP**.
7. Programar la aplicación para acceder la ayuda, esto es, incluir el código fuente necesario para realizar el llamado (por medio de la función **WinHelp**) del archivo de ayuda (archivo.**HLP**).

4.3.2 INTEGRACIÓN DEL SISTEMA DE AYUDAS AL SISTEMA DE PRECIOS UNITARIOS

PLANEACIÓN DEL SISTEMA DE AYUDAS

En la planeación del sistema de ayudas se seguirán los siguientes pasos:

1. DEFINICIÓN DEL TIPO DE USUARIOS

Es importante ubicar el tipo de usuarios de la aplicación para crear una ayuda adecuada. Los usuarios de la aplicación EL SISTEMA DE PRECIOS UNITARIOS Windows, serán usuarios del Sistema de Precios Unitarios Texto que ya manejan la aplicación pero que posiblemente no conozcan el ambiente Windows, por otro lado existirán usuarios nuevos del Sistema de Precios Unitarios, que desconocerán la aplicación y el ambiente Windows, e inclusive desconozcan el uso de una computadora (en muchas ocasiones el Ingeniero Civil, delega actividades del sistema a secretarias, aprendices, etc.).

2. DEFINICIÓN DEL CONTENIDO DE LA AYUDA

El contenido de la ayuda debe ser de acuerdo al tipo de usuarios que se piensa dirigir la aplicación, además debe contener los temas principales de la aplicación. Los manuales actuales del Sistema de Precios Unitarios servirán para cubrir este

punto, pero se tendrá en mente cubrir los temas en la forma más concreta posible.

No es recomendable tener disponible todo el manual en línea.

ESTRUCTURACIÓN DE LA AYUDA

Los temas que estarán disponibles en la ayuda se estructurarán siguiendo una jerarquía, contando en primer orden con un índice de temas disponibles para el usuario y con los atributos necesarios para proporcionarle lo siguiente:

- Movimiento a través de las listas de los temas para obtener la información deseada.
- Saltos de un tema a otro.
- Clasificación de palabras ligadas a temas específicos, para realizar fácilmente la búsqueda del tema deseado.
- Movimiento a través de temas ordenados por medio de una secuencia lógica, esto es, para usuarios que gustan de leer varios temas al mismo tiempo.
- Gráficos para proveer al usuario de una ayuda visual, tener cuidado de que estos gráficos sean imágenes apropiadas del tema en cuestión.

Para estructurar los temas de ayuda se incluye en el archivo que los contiene, el código de control necesario para establecer la forma en que el usuario se moverá a

través de la ayuda. Este código de control consiste en notas de pie de página, texto subrayado y texto con doble raya, que indicarán al compilador de ayudas una acción a realizar. El archivo así generado es un archivo **.RTF**

A continuación se muestran las acciones y el código de control asociado de un archivo .RTF:

a) Definir una etiqueta que especifica los temas que el compilador construirá dentro del sistema de ayudas (nota *).

b) Definir el contexto que identifica de forma única el tema (nota #)

c) Asignar un título a un tema (nota \$).

d) Definir palabras claves para realizar la búsqueda de los temas (nota K).

e) Asignar una secuencia lógica al tema (nota +).

f) Realizar referencias cruzadas, que permiten al usuario saltar a otro tema (el texto identificador del tema debe estar con doble raya)

g) Permitir desplegar una caja temporal de información mientras el usuario mantenga la tecla Enter presionada (el texto identificador del tema debe estar subrayado). **El usuario no necesita tener la tecla enter presionada, esto era para la versión windows 3.0.**

Estructurar la ayuda es un punto dentro de la planeación muy importante, pues será consecuencia de la eficiencia y claridad que se logre de la ayuda en línea.

Por ejemplo para el Sistema de Precios Unitarios Windows el sistema de ayudas quedaría estructurado de la siguiente manera:

Esquema Global

- | | |
|---------------------------------|--|
| 1. Índice General | Conteniendo la lista de los temas principales del Sistema de Precios Unitarios. |
| 2. Uso del Teclado | Explicación breve del uso de teclas especiales del Sistema de Precios Unitarios y de Windows. |
| 3. Tutorial | Ejemplo práctico del manejo del Sistema de Precios Unitarios, abarcando todas opciones principales. |
| 4. Uso de la Ayuda | Explicación de que es la ayuda y de como utilizarla. |
| 5. Acerca del Sistema Unitarios | Información General del Sistema de Precios Unitarios contemplando número de versión, fecha de creación, Compañía creadora y la Compañía con licencia para distribución del producto. |

Cada uno de los subtemas debe ser desglosado por ejemplo:

Índice General

- | | |
|--|--|
| 1. Que es EL SISTEMA DE PRECIOS UNITARIOS? | Explicación general del Sistema de PRECIOS UNITARIOS. |
| 2. Manejo de Obras | Explicación necesaria para el borrado, copiado y reconstrucción de una obra. |
| 3. Precios Unitarios | Contará con una lista de temas agrupándolos en Insumos, Factor de Salario Real, Análisis, y Presupuestos. |
| 4. Control de Obras | Contará con una lista de temas agrupándolos en Escalatorias, Estimaciones Normales, y Estimaciones Escaladas. |
| 5. Ruta Crítica (Por definir) | Explicación de todo lo referente a Ruta Crítica. Desplegado de una lista de temas agrupándolos en Manejo de Proyectos, Manejo de Datos, Cálculo de la Ruta Crítica y Reportes Disponibles. |
| 6. Utilerías | Explicación del manejo de las utilerías con las que cuenta EL SISTEMA DE PRECIOS UNITARIOS, como el editor de texto, la calculadora y la hoja de cálculo. |

7. Consulta de Errores Mostrará la relación de errores que se pueden desplegar durante el uso del Sistema de Precios Unitarios.
Relación que estará clasificada por tipo de error.
8. Glosario de Términos Mostrará una lista de términos utilizados en EL SISTEMA DE PRECIOS UNITARIOS con una breve descripción del mismo. La lista estaría ordenada alfabéticamente.
9. Uso del Toolbar Breve explicación del Toolbar General y Particular y de cada uno de sus elementos.

ada tema desplegado en el índice será de igual manera desglosado con el fin de describir los temas mencionados.

odos los archivos con los temas de ayuda estarían concentrados en un sólo directorio, y se identificarían los nombres de éstos con un prefijo (carácter *h* identificador de archivos de ayuda), y 7 caracteres (como máximo) para identificar el tema, por ejemplo, *hutilsa.HLP* que haría referencia al archivo de ayuda con temas de las utilerías del Sistema de Precios Unitarios, *hprecios.HLP* identificador del archivo de ayuda con temas del módulo de Precios Unitarios, etc.

(El identificador que propuse para los archivos es hs, la h de help y la s de Sistema de Precios Unitarios)

En la estructura de la ayuda se deben considerar las tres opciones de venta que tiene actualmente EL SISTEMA DE PRECIOS UNITARIOS, a continuación se muestran dichas opciones y la forma en que podría implementarse la ayuda:

a) Paquete con Precios Unitarios, Ruta Crítica y Control de Obra. En el sistema de ayudas se habilitarían todos los temas para consulta.

b) Paquete con Precios Unitarios y Control de Obra. En el sistema de ayudas, en el índice General, aparecería deshabilitado el tema de Ruta Crítica, y habilitados los demás temas.

c) Paquete con Ruta Crítica. En el sistema de ayudas, en el índice General se deshabilitarían los temas de Manejo de Obras, Precios Unitarios y Control de Obras.

OBSERVACIONES:

1.-ROBOHELP es una herramienta que permite agilizar la creación de la ayuda de una aplicación, al proveer de elementos para generar fácilmente el archivo RTF, implementando el código de control necesario para generar la ayuda según se haya planeado, además ROBOHELP genera el archivo de proyecto y realiza la compilación de los recursos generando el archivo de ayuda. Por lo que los pasos 4

al 6 del proceso de creación de un sistema de ayudas en una aplicación Windows quedarían cubiertos con esta herramienta

2.-Los temas de la ayuda se pueden colocar en un sólo archivo utilizando páginas para separar cada tema, o se puede colocar cada tema en un archivo. Tener un archivo muy grande de ayuda o muchos archivos pequeños, puede presentar dificultades en la creación del archivo de recursos de la ayuda.

3.-Cada tema que puede ser accedido vía una liga hipertexto, debe tener un identificador único (context string); cada tema puede tener un título; una lista de palabras claves asociadas; una etiqueta identificadora; y puede tener una secuencia lógica asignada

4.-Es importante hacer notar la comunicación que debe existir en la implementación de la ayuda, entre el programador de la ayuda quien realizará el llamado a ésta y el escritor de la ayuda quien escribirá los temas e implementará los atributos que estos tendrán; para lograr la consistencia de la misma.

PERFIL DEL PROGRAMADOR DE LA AYUDA:

- Conocimiento de programación Windows
- Conocimiento del manejo de ayudas en Windows

Capítulo IV Desarrollo del Sistema

- Conocimiento de la estructura del sistema de ayudas EL SISTEMA DE PRECIOS UNITARIOS Windows (la ayuda primero se de debe estructurar, y posteriormente escribir y programar).

PERFIL DEL ESCRITOR DE LA AYUDA:

- Amplio conocimiento de la aplicación EL SISTEMA DE PRECIOS UNITARIOS.
- Conocimiento de los problemas más frecuentes en el uso del Sistema de Precios Unitarios por parte de los usuarios.
- Conocimiento de la estructura del sistema de ayudas EL SISTEMA DE PRECIOS UNITARIOS Windows.
- Manejo de Microsoft Word

4.3.3 MANUAL DE USUARIO

SECCIÓN 1 INTRODUCCIÓN



SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS.

EL SISTEMA DE PRECIOS UNITARIOS ofrece todas las bondades del ambiente gráfico que es ya un estándar en la industria, la interoperabilidad con otras herramientas de primer nivel como la línea Office de Microsoft, la compatibilidad con versiones anteriores y muchas otras ventajas, por eso, estamos seguros de que este sistema cubrirá sus más altas expectativas.

¿QUÉ ES EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS?

El Sistema de Precios Unitarios para Windows fue diseñado para resolver las diversas necesidades que tienen las Compañías Constructoras.

El Sistema de Precios Unitarios para Windows está diseñado en forma modular lo cual facilita su utilización, integrando como primer objetivo el módulo de Precios Unitarios, cuyas opciones son las siguientes:

INSUMOS

Dentro de este módulo se pueden editar los diferentes Materiales, la Obra de Mano, la Maquinaria y Equipo y los Subcontratos, para elaborar el Presupuesto de la Obra. Además cuenta con un Catálogo de Insumos que puede accederse en el momento que así se desee.

ANÁLISIS

Dentro de este módulo se clasifican en cinco niveles los Análisis o Matrices, que son: Básicos, Integrados, Partidas, Capítulos y Extraordinarios. Este módulo permite analizar otras Matrices al mismo tiempo que se realiza la captura de un Análisis.

FACTOR DE SALARIO REAL

Dentro de este módulo, el Sistema de Precios Unitarios para Windows permite realizar el cálculo del Factor de Salario Real agregando conceptos o eliminándolos, o si usted así lo desea tomar en cuenta el factor que éste le proporciona. Este factor será aplicado en el módulo de Obra de Mano.

PRESUPUESTOS

Dentro de este módulo se realizan las operaciones necesarias para llevar a cabo las Actualizaciones de Precios a nivel de Obra y Análisis o Matrices. Esta actualización es necesaria ya que si se modifica el costo de algún Insumo que intervenga en el Presupuesto, es necesario recalcular los precios de todos los Análisis donde intervenga directa o indirectamente el Insumo modificado, para obtener el costo real de la Obra. También dentro de este módulo podrá definir sus formatos de Sobre Costo y realizar la Explosión de Materiales.

RELACIONES

Dentro de este módulo se puede editar información acerca de las Zonas Económicas relacionadas con sus Insumos. Así como también editar las Familias para la agrupación de Insumos y la captura de los diferentes tipos de Moneda a utilizar.

VENTAJAS

El Sistema de Precios Unitarios para Windows le permite:

- Manejar un número ilimitado de Obras.
- Generar un Catálogo Maestro de Precios.
- Tener Materiales con porcentajes de Flete y Descuento.
- Tener la Obra de Mano con el Factor de Salario Real Integrado.
- Tener la Maquinaria y Equipo con cálculo de Costo Hora Máquina integrado.
- Realizar Análisis o Matrices de precios a 5 niveles: Básicos, Integrados, Partidas, Capítulos y Extraordinarios

Además el Sistema de Precios Unitarios para Windows ofrece las siguientes ventajas:

- Trabajar en forma individual o con varios usuarios dentro de la misma Obra, explotando las facilidades de ambientes multiusuario.

- Posibilidad de mantener varias Obras abiertas en sus diferentes opciones, al mismo tiempo (Dependiendo de la memoria con la que cuente su equipo).
- Panel de Control, desde donde se puede editar información relevante, relacionada con su Obra.
- Acceso a cualquier opción del Sistema de Precios Unitarios desde cualquier pantalla (Ventana).
- Con el Sistema de Precios Unitarios-XIE usted puede enviar datos de su Obra a: dBase, Excel, Access.
- Nuevos tipos de reportes con diseños mejorados, así como también la posibilidad de generar sus propios formatos de reportes y adecuarlos a sus necesidades de una manera muy ágil y sencilla.
- Compartir datos generados en el Sistema de Precios Unitarios con otras aplicaciones Windows.
- Un aprendizaje rápido del manejo del Sistema de Precios Unitarios para Windows con la ayuda de los tips del Sistema de Precios Unitarios.
- Comunicación directa con Project de Microsoft, para el cálculo de la Ruta Crítica.

Dentro del módulo de Insumos:

- Manejo de múltiples precios por Insumo
- Captura de precios en moneda nacional o extranjera.
- Nuevo modo de edición de Insumos en forma tabular.
- Acceso a cualquier Insumo o Matriz que se encuentre en otra Obra.
- Precios asociados a una Zona Económica.
- Copiado de información con sólo seleccionar los datos, arrastrarlos y soltarlos sobre el Insumo destino.
- Manejo de descripciones sin límite de longitud.
- Manejo de Familias para agrupación de Insumos.
- Manejo de claves alfanuméricas con 25 caracteres de longitud.

Para los Análisis:

- Manejo de Familias para agrupación de Análisis.
- Captura de Insumos desde el módulo de Análisis.

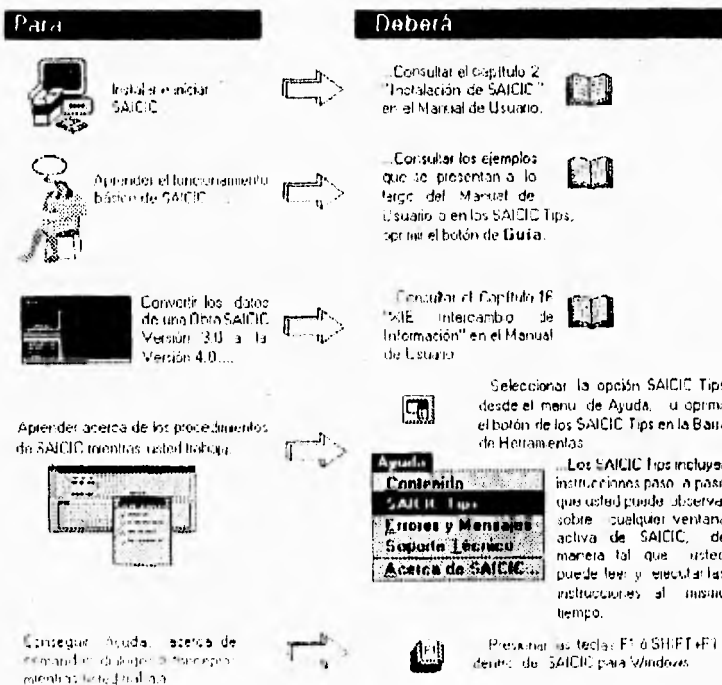
- Acceso a cualquier Insumo o Análisis que se encuentre en otra Obra.








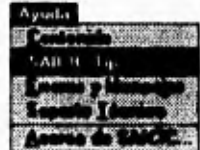


ra los Presupuestos:

- Definir sus propios formatos de Sobre Costo.

ITINERARIO DEL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Los usuarios del Sistema de Precios Unitarios para Windows pueden guiarse con el siguiente itinerario para viajar a través del sistema.



Para	Deben
 Instalar e iniciar SAICIC...	...Consultar el capítulo 2 "Instalación de SAICIC" en el Manual de Usuario. 
 Aprender el funcionamiento básico de SAICIC.....	...Consultar los ejemplos que se presentan a lo largo del Manual de Usuario o en los SAICIC Tips, oprimir el botón de Guía.
 Convertir los datos de una Obra SAICIC Versión 3.0 a la Versión 4.0....	...Consultar el Capítulo 16 "XIE Intercambio de Información" en el Manual de Usuario. 
Aprender acerca de los procedimientos de SAICIC mientras usted trabaja... 	...Seleccionar la opción SAICIC Tips desde el menú de Ayuda, u oprimir el botón de los SAICIC Tips en la Barra de Herramientas.   ... Los SAICIC Tips incluyen instrucciones paso a paso que usted puede observar sobre cualquier ventana activa de SAICIC, de manera tal que usted puede leer y ejecutar las instrucciones al mismo tiempo.
Conseguir Ayuda, acerca de comandos, diálogos o conceptos mientras usted trabaja... 	... Presionar las teclas F1 ó SHIFT+F1 dentro de SAICIC para Windows. 

CÓMO CONSEGUIR INFORMACIÓN ACERCA DEL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Si usted quiere obtener información acerca del Sistema de Precios Unitarios para Windows, mientras usted trabaja utilice alguno de los siguientes recursos:

- Consulte la ayuda en línea disponible desde el menú de **Ayuda** de la aplicación.
- Consulte el Manual de Usuario.
- Revise los requerimientos de Sistema de Precios Unitarios para Windows, dentro de la Ayuda de Soporte Técnico.

UTILIZANDO LA AYUDA EN LÍNEA Y LOS TIPS DEL SISTEMA DE PRECIOS UNITARIOS

La Ayuda en Línea del Sistema de Precios Unitarios está compuesta de varios archivos, los cuales cuentan con información completa acerca de cada uno de los módulos del Sistema de Precios Unitarios. A continuación, se describe el contenido de cada uno de ellos.

AYUDA DEL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

La Ayuda Principal, contiene información acerca de conceptos, comandos y diálogos del Sistema de Precios Unitarios.


Si desea:

Puede invocar la ayuda:

Obtener información del funcionamiento y componentes de los diálogos.

Oprimiendo el botón de **Ayuda** que se encuentra dentro de cada uno de los diálogos del Sistema de Precios Unitarios.

Obtener información acerca de alguna opción del menú o comando dentro del Sistema de Precios Unitarios

Oprimiendo el botón  correspondiente a la Ayuda Sensitiva que se encuentra dentro de la Barra de Herramientas General y posteriormente, seleccionando con la ayuda del mouse, la opción de la cual desea obtener información.

Revisar el contenido de la Ayuda

Seleccionando desde el menú de **Ayuda** la opción de **Contenido**

MANEJO DE ERRORES

Este archivo de Ayuda contiene los Mensajes y Errores que pueden presentarse dentro de una sesión del Sistema de Precios Unitarios, en él se describen las causas que originaron el error y sus posibles soluciones.

Para invocar esta Ayuda presione el botón de **Ayuda** que se encuentra en cada uno de los Mensajes o Errores del Sistema de Precios Unitarios, u oprima la tecla F1.

SOPORTE TÉCNICO

En caso de que ocurra algún problema con el Sistema de Precios Unitarios, usted puede consultar esta Ayuda en Línea, la cual contiene los principales requerimientos que el Sistema de Precios Unitarios necesita para un funcionamiento adecuado.

Para invocarla seleccione del menú de **Ayuda**, la opción de **Soporte Técnico**.

INTERCAMBIO DE INFORMACIÓN

Dentro de la Ayuda del Programa de Intercambio de Información, usted podrá consultar como realizar las conversiones de los datos del Sistema de Precios Unitarios hacia Excel, Acces, dBase, así como también obtener información de cómo realizar un copiado entre Obras del Sistema de Precios Unitarios.

Esta Ayuda puede ser invocada desde el menú de **Ayuda** del Programa de Intercambio de Información.

REPORTES

La Ayuda de Reportes contiene información acerca de cómo diseñar y cómo modificar un reporte dentro del Rep Estudio.

Usted podrá invocar esta Ayuda desde el menú de **Ayuda** que se encuentra dentro del Rep Estudio.


INSTALACIÓN

En ella podrá consultar cualquier duda que surja referente a la instalación del Sistema de Precios Unitarios para Windows.

Para invocarla oprima el botón de **Ayuda** que se encuentra dentro de cada uno de los diálogos que componen la Instalación.

PRECIOS UNITARIOS TIPS

Los Tips del Sistema de Precios Unitarios incluyen instrucciones paso a paso acerca de cada una de las tareas que se puede realizar dentro del Sistema de Precios Unitarios. Los Tips le permiten leer y ejecutar un conjunto de instrucciones de manera simultánea.

Para usar los Tips del Sistema de Precios Unitarios, oprima el botón  correspondiente a los Tips dentro de la Barra de Herramientas General, o desde el menú de **Ayuda** seleccione la opción **Precios Unitarios Tips**.

MANUAL DE USUARIO DEL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Este manual tiene como objetivo el explicar al usuario como utilizar el Sistema de Precios Unitarios para Windows.

CONTENIDO DEL MANUAL

Este manual, se encuentra dividido de la siguiente manera:

PARTE 1 COMENZANDO CON EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS - La información contenida en esta parte le ayudará a familiarizarse con el Sistema de Precios Unitarios para Windows. En ellos encontrará información acerca de la instalación del Sistema, de la creación o apertura de una Obra, de cómo editar información referente a la Obra dentro del Panel de Control y de cómo cerrar una Obra que se encuentra activa.

PARTE 2 EDICIÓN DE DATOS Y MÓDULOS DEL SISTEMA DE PRECIOS UNITARIOS - Esta parte contiene información acerca de los principales comandos de edición que se encuentran dentro del Sistema de Precios Unitarios para Windows.

PARTE 3 PRECIOS UNITARIOS - En esta parte usted puede encontrar información acerca de las opciones que nos presenta el módulo de Precios Unitarios, como son: Insumos, Análisis, Cálculo del Factor de Salario Real y Presupuestos.

CONVENCIONES DENTRO DEL MANUAL

Para ayudarle a localizar e interpretar la información contenida dentro del Manual de una manera más sencilla, se utilizarán las siguientes convenciones:

Estilo de letra:	Se utilizan para:	Ejemplo:
MAYUSCULAS	Las teclas rápidas que se usan dentro del Sistema de Precios Unitarios.	CTRL+ALT+D
	Los nombres de directorios y archivos.	C:\OBRAS\TUNEL\SISTEMA DE PRECIOS UNITARIOS.DBD
Negritas	Escribir los nombres de los menús del Sistema de Precios Unitarios.	Edición
	Los títulos que aparecen dentro de los botones del Sistema de Precios Unitarios.	Aceptar

También a lo largo del manual usted podrá encontrar los siguientes símbolos:

Símbolo: **Utilizado para:**

M

Describir los posibles problemas que pueden presentarse a lo largo de una sesión del Sistema de Precios Unitarios, si no se tiene un manejo adecuado de sus opciones.

@

Indicar una nota dentro del texto del manual.



Marcar los tips que pueden ser de gran utilidad dentro del Sistema de Precios Unitarios.

+

Indicar que las teclas deben oprimirse simultáneamente, este símbolo no debe teclearse. Por ejemplo, si se tiene la siguiente expresión CTRL + P, le indica que debe presionar la tecla CTRL y a continuación la tecla P, sin dejar de presionar la primera, de manera que las dos queden presionadas a un mismo tiempo.

SECCIÓN 2 INSTALACIÓN DEL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS



Esta sección describe cómo usar el Programa de Instalación de Sistema de Precios Unitarios para Windows.

REQUERIMIENTOS DEL SISTEMA

El Sistema de Precios Unitarios para Windows requiere de la siguiente configuración mínima:

- Una Computadora Personal IBM o 100% compatible ejecutando Microsoft Windows 3.1 o mayor.
- Un monitor VGA.
- Un microprocesador 80386 o mayor.
- 4 Megabytes de memoria RAM disponibles (recomendables 8 Megabytes).
- Un disco duro con 10 Megabytes de espacio libre para instalar las opciones que usted necesita. El Programa de Instalación verifica que su disco duro cuente con suficiente espacio para la instalación, antes de copiar los archivos.

INSTALANDO EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Esta sección es una guía básica del Programa de Instalación del Sistema de Precios Unitarios para Windows.

Usted puede instalar el Sistema de Precios Unitarios para Windows:

- a. Desde discos flexibles hacia el disco duro de su computadora.
o si usted está trabajando en un ambiente de Red usted puede instalarlo:
- b. Desde discos flexibles hacia el Servidor de Red.
- c. Desde el Servidor de Red hacia una Estación de Trabajo.

VÍA RÁPIDA DE LA INSTALACIÓN

Esta sección como un resumen de los procedimientos de la Instalación del Sistema de Precios Unitarios para Windows o simplemente para obtener información acerca del proceso de instalación.

DÓNDE INICIAR?

El Programa de Instalación del Sistema de Precios Unitarios para Windows debe ser ejecutado desde Windows 3.1, eligiendo dentro del Administrador de Programas el menú de **Archivo** y la opción de **Ejecutar**. Le recomendamos que antes de iniciar el programa de Instalación de Sistema de Precios Unitarios para Windows cierre todas las aplicaciones Windows y DOS que se encuentren abiertas.

PROCEDIMIENTOS DE INSTALACIÓN

Para conseguir ayuda en cualquiera de los diálogos de la instalación oprima el botón de **Ayuda** que aparece en cada uno de ellos.

Usted podrá realizar la instalación del Sistema de Precios Unitarios para Windows siguiendo cualquiera de los siguientes caminos:

INSTALACIÓN EXPRESS

En la Instalación Express se instalarán los programas ejecutables del Sistema de Precios Unitarios para Windows y las librerías que éste necesita.

Este tipo de instalación se recomienda para instalar el Sistema de Precios Unitarios para Windows dentro de un Servidor de Red o para una instalación monousuario dentro de su disco duro.

INSTALACIÓN PERSONALIZADA

En este tipo de instalación se abre el diálogo correspondiente a la Instalación Personalizada. En este diálogo usted puede seleccionar los archivos de Sistema de Precios Unitarios para Windows a instalar, o añadir. Para obtener mas información ver "Opciones de la Instalación Personalizada"

Este tipo de instalación se recomienda cuando Sistema de Precios Unitarios para Windows ya se encuentra instalado en un Servidor de Red o en el disco duro de su

computadora y se desean añadir otros componentes, o se requiere realizar la instalación de Sistema de Precios Unitarios para Windows en un equipo con espacio reducido en disco, por ejemplo una Laptop.

INSTALACIÓN EN UNA ESTACIÓN DE TRABAJO

Este tipo de instalación creará el grupo de Sistema de Precios Unitarios para Windows y los íconos correspondientes dentro de una estación de trabajo siempre y cuando éste ya se encuentre instalado en su computadora o Servidor de Red.

Este tipo de instalación solamente es activado si usted previamente ya ha instalado Sistema de Precios Unitarios por medio de alguna de las opciones Express o Personalizada. Usted debe ejecutar el Programa de Instalación desde el directorio donde instaló originalmente Sistema de Precios Unitarios para que esta opción se active.

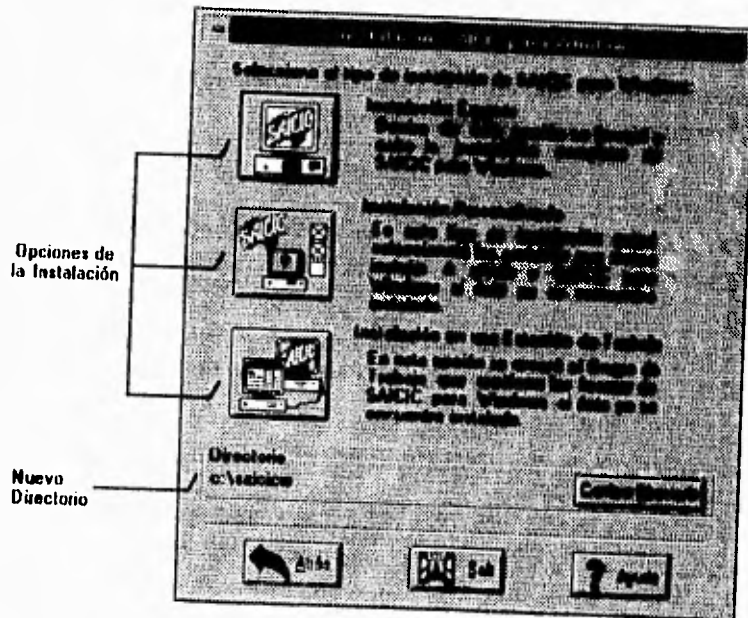


Figura 2.1. Diálogo que se encuentra dentro del Programa de Instalación, que permite elegir el Tipo de Instalación del Sistema de Precios Unitarios para Windows

COMO INSTALAR EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS EN SU DISCO DURO O EN EL SERVIDOR DE RED DESDE SUS DISCOS DE INSTALACIÓN

1. Conecte la interface del Sistema de Precios Unitarios en el puerto paralelo de la computadora, antes de conectarla, es necesario tener el equipo y la impresora apagada para evitar que su interface reciba descargas.
2. Inicialice Windows y verifique que no existen aplicaciones abiertas.
3. Inserte el Disco 1 de la Instalación del Sistema de Precios Unitarios para Windows en la unidad A ó B.
4. Desde el menú de **Archivo** del Administrador de Programas seleccione la opción de **Ejecutar**, y teclee:

A:\INSTALAR.EXE Ó B:\INSTALAR.EXE

b: según la unidad donde se insertó el disco.



Si usted cuenta con la versión de Windows en inglés seleccione desde el menú **File** la opción **Run**.

5. Oprima el botón de **Continuar**. Con lo que se iniciará el Programa de Instalación presentándole el diálogo de Bienvenida, oprima el botón de **Continuar** para proseguir con la instalación. Al oprimir este botón se presentará el diálogo del Nombre del Usuario y de la Compañía.
6. Teclee su Nombre y el de la Compañía y oprima el botón de **Continuar**.
7. Usted tendrá oportunidad de confirmar si la información es correcta, si desea modificarla oprima el botón de **Cambiar**, si está de acuerdo oprima el botón de **Continuar**.
8. Dentro del diálogo de Directorio de la Instalación se le presentará el directorio en donde será instalado el Sistema de Precios Unitarios para Windows, si está de

acuerdo con dicho directorio oprima el botón de **Continuar** de lo contrario oprima el botón de **Cambiar Directorio**.

9. Si decidió cambiar el directorio:
 - Introduzca la ruta donde desea instalar el **Sistema de Precios Unitarios para Windows** dentro del campo de edición **Subdirectorio de Instalación**. O seleccione el directorio dentro de la lista de **Directorios**. Si el directorio no existe éste se creará automáticamente.
 - Oprima el botón de **Aceptar**, con ello regresará al diálogo de Directorio de la Instalación.
 - Si está de acuerdo con el directorio proporcionado oprima el botón de **Continuar**.
10. Seleccione el tipo de instalación que usted desea: **Instalación Express** o **Instalación Personalizada**. La **Instalación** en una **Estación de Trabajo** se encontrará deshabilitada, ya que es la primera vez que usted instala el **Sistema de Precios Unitarios para Windows**.
11. Si usted seleccionó la **Instalación Personalizada** asegúrese que los **checkboxes** asociados a los componentes del **Sistema de Precios Unitarios para Windows** que desea instalar se encuentren seleccionados. Un componente se encuentra seleccionado si el **checkbox** se muestra de la siguiente manera y no se encuentra seleccionado si el **checkbox** se encuentra vacío .
12. Si usted se posiciona en el componente de **Ayudas** o en el componente de **Reportes** se presentará el botón de **Más Opciones**, al oprimir este botón aparecerá el diálogo correspondiente al **Diálogo de las Opciones de Reportes** o al **Diálogo de las Opciones de Ayudas** dependiendo del componente seleccionado.
13. Si oprimió el botón de **Más Opciones**, usted podrá seleccionar los componentes de las **Ayudas** o de los **Reportes** que desea instalar.
14. Si está de acuerdo con los componentes de **Ayudas** o **Reportes** seleccionados oprima el botón de **Continuar**, con lo cual regresará al diálogo de la **Instalación Personalizada**.

- i. Una vez que se encuentra dentro del diálogo de la Instalación Personalizada, si está de acuerdo con los componentes seleccionados, oprima el botón de **Instalar**. Para obtener más información acerca de la Instalación Personalizada ver "Opciones de la Instalación Personalizada".

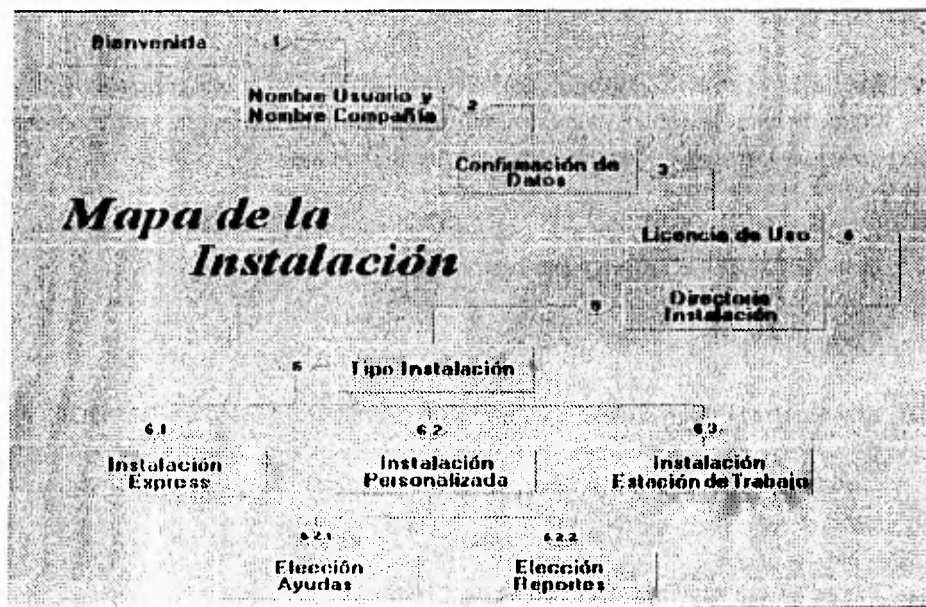


Figura 2.2. Mapa de la Instalación del Sistema de Precios Unitarios para Windows

CÓMO INSTALAR EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS DESDE EL SERVIDOR DE RED O AGREGAR ALGUNO DE SUS COMPONENTES.

En la continuación se describe el procedimiento de cómo realizar la instalación del Sistema de Precios Unitarios para Windows:

- Desde el Servidor de Red hacia una Estación de Trabajo.
- Desde el Servidor de Red hacia el disco duro de una Estación de Trabajo

- Desde una Estación de Trabajo al Servidor de Red, siempre y cuando el Sistema de Precios Unitarios para Windows ya se encuentre instalado dentro del Servidor y usted es el Administrador de la misma.

PROCEDIMIENTO:

1. Instale la interface de Sistema de Precios Unitarios, en el puerto paralelo de la computadora, antes de instalarla, es necesario tener el equipo y la impresora apagada para evitar que su interface reciba descargas.
2. Desde el directorio donde se encuentra el Sistema de Precios Unitarios para Windows ejecute el programa INSTALAR.EXE.
3. Ejemplo: c:\UNITARIOS\INSTALAR.EXE
4. Al ejecutar el Programa de Instalación se presentará el diálogo de Bienvenida de Sistema de Precios Unitarios para Windows, oprima el botón de **Continuar** para proseguir con la instalación, al oprimirlo, se presentará el diálogo del Nombre de Usuario y de la Compañía.
5. Teclee su Nombre y oprima el botón de **Continuar**.
6. Usted tendrá oportunidad de confirmar si la información es correcta, si desea modificarla oprima el botón de **Cambiar**, si está de acuerdo oprima el botón de **Continuar**.
7. Dentro del diálogo de Directorio de la Instalación se le presentará el directorio en donde se encuentra instalado el Sistema de Precios Unitarios para Windows. Si desea compartir el ejecutable del Sistema de Precios Unitarios para Windows que se encuentra instalado en el Servidor de Red o agregarle más componentes, oprima el botón de **Continuar**, si desea instalar el Sistema de Precios Unitarios para Windows dentro del disco duro de la Estación de Trabajo oprima el botón de **Cambiar Directorio**, así usted podrá seleccionar el directorio de su disco duro en el cual se instalará el Sistema de Precios Unitarios para Windows.

8. Si decidió cambiar el directorio:

- Introduzca la ruta donde desea instalar el Sistema de Precios Unitarios para Windows dentro del campo de edición **Subdirectorio de Instalación**. O seleccione el directorio dentro de la lista de **Directorios**.
- Oprima el botón de **Aceptar**, con ello regresará al diálogo de Directorio de la Instalación.
- Una vez que usted se encuentra en el diálogo de Directorio de la Instalación, si está de acuerdo con el directorio proporcionado, oprima el botón de **Continuar**.

9. Dentro del diálogo donde se elige el Tipo de Instalación, usted deberá elegir:

La Instalación Express si:

- Desea realizar la instalación completa del Sistema de Precios Unitarios para Windows desde el Servidor de Red hacia el disco duro de una Estación de Trabajo

La Instalación Personalizada si:

- Desea seleccionar los componentes del Sistema de Precios Unitarios para Windows a instalar desde el Servidor de Red, hacia el disco duro de una Estación de Trabajo.
- Desea añadir algún o algunos de los componentes del Sistema de Precios Unitarios para Windows dentro del Servidor de Red

La Instalación en una Estación de Trabajo si:

- Desea compartir el Sistema de Precios Unitarios para Windows que se encuentra instalado en el Servidor de Red

10. Si usted seleccionó la Instalación Express o Personalizada el Programa de instalación le preguntará desde qué unidad leerá los Discos de la Instalación. Por

ejemplo si la lectura de los discos se realizará desde la unidad A, teclee lo siguiente:

A:\

11. Si el tipo de instalación que usted eligió fue Instalación Personalizada asegúrese que los checkboxes asociados a los componentes del Sistema de Precios Unitarios para Windows que desea instalar se encuentren seleccionados. Un componente se encuentra seleccionado si el checkbox se muestra de la siguiente manera y no se encuentra seleccionado si el checkbox se encuentra vacío . Si usted se posiciona en el componente de Ayudas o en el de Reportes se presentará el botón de **Más Opciones**, al oprimir este botón aparecerá el diálogo correspondiente al Diálogo de las Opciones de Reportes o al Diálogo de las Opciones de Ayudas dependiendo del componente seleccionado.

Si oprimió el botón de **Más Opciones**, usted podrá seleccionar los componentes de las Ayudas o de los Reportes que desea instalar.

Si está de acuerdo con los componentes seleccionados oprima el botón de **Continuar**, con lo cual regresará al diálogo de la Instalación Personalizada, una vez en dicho diálogo, si está de acuerdo con los componentes seleccionados, oprima el botón de **Instalar**.

12. Una vez que empieza el proceso de copiado de archivos, si usted eligió la Instalación Express o Personalizada, el Programa de Instalación le pedirá que introduzca el Disco 1 en la unidad que usted proporcionó con anterioridad.

OPCIONES DE LA INSTALACIÓN PERSONALIZADA

Cuando usted selecciona la Instalación Personalizada dentro del diálogo de Tipo de Instalación aparece el siguiente diálogo:

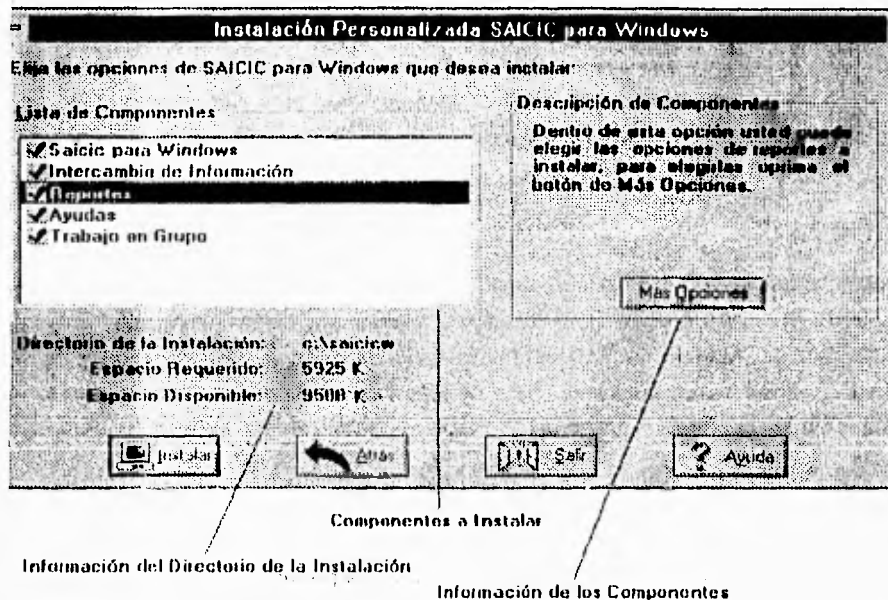


Figura 2.3 Diálogo que se presenta dentro del Programa de Instalación que le permite seleccionar los componentes de Sistema de Precios Unitarios para Windows a instalar.

En la continuación describiremos cada uno de sus componentes:

SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Si este componente se encuentra seleccionado se instalarán los programas ejecutables y las librerías necesarias para el funcionamiento del Sistema de Precios Unitarios para Windows, en su módulo de Precios Unitarios.

INTERCAMBIO DE INFORMACIÓN

Al elegir este componente se instalará el Programa de Intercambio de Información (XIE). Este programa puede convertir datos de Excel, Acces, dBase y archivos de tipo TEXTO o ASCII al Sistema de Precios Unitarios para Windows

REPORTES

Dentro de este componente usted puede elegir las opciones de reportes a instalar, las cuales son el Rep Estudio y las Plantillas de Reportes. Usted puede utilizar el botón de **Más Opciones** para abrir el diálogo que nos presenta las opciones de **Reportes**. Para obtener mas información ver. "Opciones de Reportes".

AYUDAS

Con este componente usted puede elegir si desea la instalación de las ayudas del Sistema de Precios Unitarios para Windows, éstas pueden ser la **Ayuda Principal**, **Soporte Técnico**, **Manejo de Errores**, **Intercambio de Información (XIE)**, **Reportes**, e **Instalación**. Usted puede utilizar el botón de **Más Opciones** para abrir el diálogo que nos presenta las opciones de **Ayudas**. Para obtener mas información ver "Opciones de Ayudas".

TRABAJO EN GRUPO

Este componente le permitirá instalar el programa que permite al Sistema de Precios Unitarios para Windows trabajar en diferentes estaciones de trabajo con una misma obra.

DESCRIPCIÓN DE COMPONENTES

Dentro de este campo se presenta una breve descripción del componente seleccionado dentro de la **Lista de Componentes**.

Así como también podrá observar el botón de **Más Opciones** cuando usted selecciona los componentes de **Reportes y Ayudas**.

MÁS OPCIONES

El botón de **Más Opciones** muestra los componentes de las opciones de **Reportes y Ayudas** que usted puede seleccionar dentro de este Tipo de Instalación.

DIRECTORIO DE LA INSTALACIÓN

Campo que nos muestra el directorio en el cual será instalado el Sistema de Precios Unitarios para Windows.

ESPACIO REQUERIDO

Presenta el espacio requerido en su disco duro para instalar los componentes del Sistema de Precios Unitarios para Windows que se encuentran seleccionados. Este se irá modificando conforme usted agregue o elimine componentes a instalar

ESPACIO DISPONIBLE

Presenta el espacio con el que cuenta la unidad de disco seleccionada para la instalación.

OPCIONES DE REPORTE

Al oprimir el botón de **Más Opciones** dentro del componente de Reportes, usted podrá seleccionar los siguientes componentes:

REP-ESTUDIO

Instalará el Rep Estudio del Sistema de Precios Unitarios para Windows. Este programa le ayudará a diseñar sus propios reportes para que cumplan con sus necesidades ya que usted puede utilizar diferentes tipos de letras en sus reportes, agregarles imágenes y sacar provecho de los colores si usted cuenta con impresora a color.

PLANTILLAS DE REPORTE

Instalará las plantillas de reportes base del Sistema de Precios Unitarios para Windows. Con estas plantillas usted puede enviar datos a impresora, además usted puede modificar los formatos de acuerdo a sus necesidades con el Rep Estudio.

OPCIONES DE AYUDAS

Al oprimir el botón de **Más Opciones** dentro del componente de Ayudas, usted podrá seleccionar los siguientes archivos de ayuda:

AYUDA PRINCIPAL

Al elegir este componente se instalará la Ayuda Principal donde podrá consultar el funcionamiento, los módulos y las principales características del Sistema de Precios Unitarios para Windows.

SOPORTE TÉCNICO

En este componente se instalará la ayuda en donde usted podrá consultar los pasos a seguir para ponerse en contacto con el Sistema de Precios Unitarios-Team en caso de no poder resolver algún problema con el Sistema de Precios Unitarios para Windows.

MANEJO DE ERRORES

En esta ayuda se encuentran los errores que pudieran ocurrir al utilizar el Sistema de Precios Unitarios para Windows, dentro de esta ayuda se encuentra el tipo de error, las posibles causas y como solucionarlo.

INTERCAMBIO DE INFORMACIÓN

En este componente se instalará la ayuda del programa de intercambio de datos, en el cual usted puede consultar el funcionamiento del programa que le permite convertir sus datos hacia Excel, Acces, dBase o un archivo tipo TEXTO o ASCII y realizar el copiado entre obras del Sistema de Precios Unitarios para Windows.

REPORTES

Dentro de la ayuda de reportes usted podrá consultar como diseñar un reporte del Sistema de Precios Unitarios para Windows y cómo modificar reportes ya existentes.

INSTALACIÓN

En ella podrá consultar cualquier duda que surja referente a la instalación del Sistema de Precios Unitarios para Windows.

Para:

Obtener más información acerca de cada uno de los diálogos de la Instalación de Sistema de Precios Unitarios

Ver:


La Ayuda en Línea correspondiente a la Instalación.

SECCIÓN 3 SISTEMA DE PRECIOS UNITARIOS



CÓMO CREAR UNA OBRA

Para Crear una Obra dentro del Sistema de Precios Unitarios para Windows es necesario que siga los siguientes pasos:

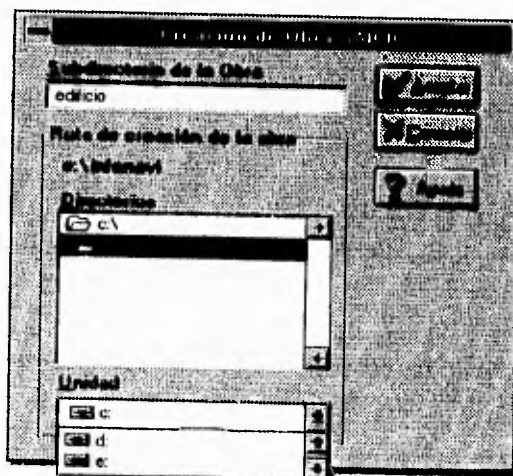
1. Oprima el botón  dentro de la Barra de Herramientas General el cual corresponde a la creación de una Obra. O seleccione del menú de **Archivo** la opción de **Crear Obra**.
2. En el combobox de **Unidad**, seleccione la unidad de disco en la cual desea crear y almacenar la información de toda su Obra.
3. Dentro de la lista de **Directorios**, seleccione el directorio en el cual se creará el subdirectorio de la Obra.
4. Dentro del campo de edición de **Subdirectorio de la Obra**, proporcione el subdirectorio en el cual desea crear la Obra, el nombre de éste puede ser de hasta 8 caracteres.
5. Si está de acuerdo con la ruta especificada oprima el botón de **Aceptar**, de lo contrario oprima el botón de **Cancelar**.
6. Si usted está trabajando en un ambiente de Red, Sistema de Precios Unitarios para Windows le preguntará si desea compartir su Obra con más usuarios. Si desea compartir su Obra oprima el botón **Si**, de lo contrario oprima el botón **No**. El compartir una Obra es de gran utilidad en un ambiente de Red, debido a que otros usuarios pueden actualizar y/o editar Insumos y Análisis, así como generar reportes, dentro de la misma Obra desde diferentes computadoras.



Si usted ya conoce la ruta donde desea crear su Obra, en el campo de edición de **Subdirectorío de la Obra**, edite dicha ruta, y oprima el botón de **Aceptar**.

Le recomendamos que utilice los directorios y subdirectorios para catalogar sus Obras por dependencia o por tipo de Obra.

EJEMPLO: En este ejemplo se desea crear una Obra Sistema de Precios Unitarios para Windows dentro del directorio INFONAVI que se encuentra en la unidad de disco c:\, creando sobre este directorio, el subdirectorío EDIFICIO, para ello, es necesario editar lo siguiente dentro del diálogo de Creación de Obra Sistema de Precios Unitarios y posteriormente oprimir el botón de **Aceptar**.



Para:

Ver:

Obtener más información acerca del diálogo de Creación de Obra Sistema de Precios Unitarios


La opción de "**Crear una Obra**" dentro del tema "**Menú de Archivo**", en la Ayuda en Línea de Sistema de Precios Unitarios.

Observar el procedimiento de la creación de una Obra por medio de un ejemplo

Consultar el ejemplo que se encuentra dentro del tema de "**Crear una Obra**", en los Tips del Sistema de Precios Unitarios.

CÓMO ABRIR UNA OBRA

Si su obra ya se encuentra creada, usted debe utilizar esta opción para poder editar algunos de sus datos, para ello, efectúe los siguientes pasos.

1. Oprima el botón  de la barra de herramientas del Sistema de Precios Unitarios correspondiente a la apertura de una Obra.
O seleccione del menú de **Archivo** la opción de **Abrir una Obra**.
2. Dentro del combobox de **Unidad**, seleccione la unidad de disco en la cual desea buscar su Obra.
3. Seleccione el subdirectorio de la Obra dentro de la lista de **Directorios**.
4. Seleccione dentro de la lista de **Nombre Archivo** el archivo .DBD.



Siempre, el Sistema de Precios Unitarios para Windows crea una Obra o la identifica por medio del archivo .DBD.

5. Si ya se encuentra seleccionado el archivo correspondiente a la Obra que desea abrir oprima el botón de **Aceptar**, si desea suspender la apertura de Obra oprima el botón de **Cancelar**.

6. Si usted está trabajando en un ambiente de Red, el Sistema de Precios Unitarios para Windows le preguntará si desea compartir su Obra con más usuarios. Si desea compartir su Obra oprima el botón **Sí** de lo contrario oprima el botón **No**.
 - El compartir una Obra es de gran utilidad en un ambiente de Red, debido a que otros usuarios pueden actualizar y/o editar Insumos y Análisis, así como generar reportes, dentro de la misma Obra desde diferentes computadoras.

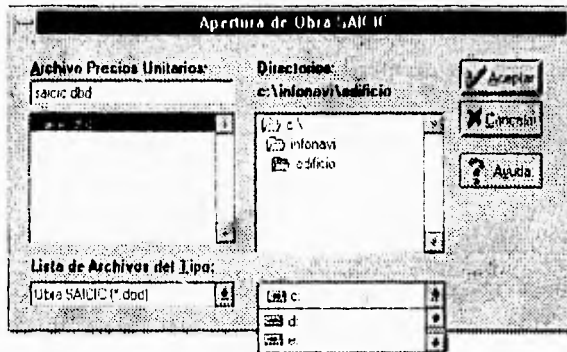


Si usted ya conoce la ruta donde se encuentra su Obra, en el campo de edición de Nombre Archivo, edite dicha ruta, y oprima el botón de Aceptar.



Sistema de Precios Unitarios para Windows le presentará dentro del menú de Archivo las últimas cuatro obras que usted ha consultado. Usted puede abrir la Obra que desea, con sólo seleccionarla de esta lista.

EJEMPLO: En este ejemplo se desea abrir la obra EDIFICIO que fue creada con Sistema de Precios Unitarios para Windows, esta obra se encuentra dentro del directorio INFONAV\EDIFICIO en la unidad de disco c:\, para ello, es necesario editar lo siguiente dentro del diálogo de Apertura de Obra Sistema de Precios Unitarios y posteriormente oprimir el botón de **Aceptar**.



Para:

Obtener más información acerca del diálogo de apertura de Obra Sistema de Precios Unitarios

Observar el procedimiento de la apertura de una Obra por medio de un ejemplo

Ver:

En la opción "**Abrir una Obra**" dentro del tema "**Menú de Archivo**" en la Ayuda en Línea del Sistema de Precios Unitarios

Consultar el ejemplo que se encuentra dentro del tema "**Abrir una Obra**", en los Sistema de Precios Unitarios Tips

CÓMO CONFIGURAR EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Dentro del módulo de configuración del Sistema de Precios Unitarios para Windows usted puede proporcionar lo siguiente:

- Las fechas de Costo y Cálculo y los caracteres de la Clave, que por default desee usted que se presenten dentro de la captura de cualquier Insumo o cualquier Análisis, ésto lo ayudará a agilizar el proceso de captura de los mismos
- Definir los colores que se utilizarán para identificar a cada uno de los elementos de una tabla, los elementos pueden ser componentes que ya se encuentran dentro de un catálogo o componentes no existentes, para el caso de los

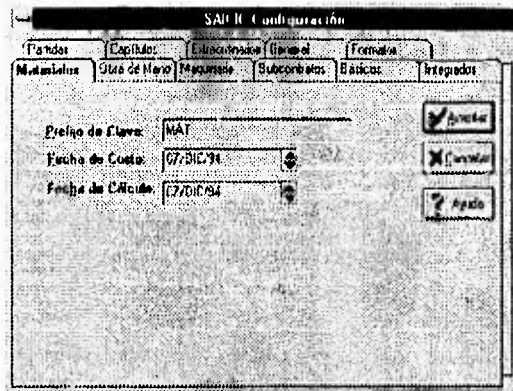
elementos del Análisis, los elementos también pueden encontrarse no analizados.

- Configurar las tablas que se utilizan para la captura de información, es decir cambiar el color de sus bordes, cambiar la manera de desplazarse entre las celdas, decidir si cuando existe texto en una celda y se desea agregar un texto nuevo, el texto existente se sustituirá por el nuevo o se agregará el texto nuevo al final del que se encuentra en la celda, etc.
- Definir si después de grabar un Insumo o Análisis se limpiarán todos los campos de la ventana de edición.
- Definir la cantidad de decimales con los que se presentará cualquier campo numérico dentro del Sistema de Precios Unitarios, así como también cuantos decimales de dicha cantidad se tomarán en cuenta en caso de que se realice algún cálculo.

Este diálogo de Configuración está formado por varios separadores en los cuales usted puede realizar las configuraciones antes mencionadas, a continuación explicaremos cada uno de ellos.

SEPARADOR DE MATERIALES, OBRA DE MANO, MAQUINARIA Y EQUIPO, SUBCONTRATOS, BÁSICOS, INTEGRADOS, PARTIDAS, CAPÍTULOS Y EXTRAORDINARIOS

Dentro de estos separadores usted puede inicializar los campos de **Fecha de Costo**, **Fecha de Cálculo** y **Clave**, para que cuando usted realice una captura de un Insumo o Análisis, estos campos siempre se encuentren inicializados con los caracteres que usted desea.



En este ejemplo le estamos solicitando al Sistema de Precios Unitarios para Windows que cada vez que se capture una Clave en Materiales, automáticamente aparezca el prefijo "MAT".

Para editar estos datos de configuración, realice lo siguiente:

1. Dentro del campo **Prefijo de la Clave**, proporcione los caracteres que desea que aparezcan en el campo de **Clave**, cada vez que realice la captura de un Insumo o Análisis, el prefijo de la Clave puede tener una longitud máxima de 20 caracteres.
2. En el campo de **Fecha de Costo**, capture la fecha que desea que aparezca cada vez que realice la captura de un Insumo o Análisis, recuerde que la Fecha de Costo es la fecha en la cual se le asigna el Costo al Insumo o Análisis.
3. Dentro del campo de **Fecha de Cálculo**, capture la fecha que desea que aparezca cada vez que realice la captura de un Insumo o Análisis, recuerde que la Fecha de Cálculo es la fecha en la cual se actualiza el Costo del Insumo o Análisis.

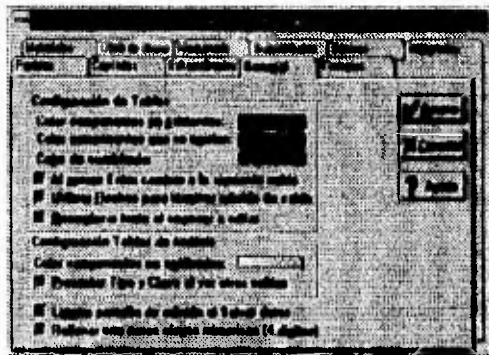
Para introducir estas fechas, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas:

Oprima la Tecla Si desea

- | | |
|-----|--|
| F11 | Colocar la fecha actual |
| F12 | Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios |
| F9 | Limpiar el campo de fecha |

SEPARADOR GENERAL

En este separador puede configurar la información de sus Análisis y de las tablas que se incluyen en el Sistema de Precios Unitarios, por ejemplo, la de Datos Particulares del Insumo o la de los elementos del Análisis, también puede definir que se limpien los campos de la ventana de edición después de grabar un registro o si usted decide capturar claves numéricas, éstas se rellenen con ceros.



Para configurar las tablas que se encuentran dentro del Sistema de Precios Unitarios, realice lo siguiente:

1. Seleccione el color con el cual se identificarán los elementos de la tabla, para ello:
 - a. Posicione el apuntador del mouse sobre el botón que se encuentra relacionado con la opción de **Color componentes ya existentes**, estos elementos son los que se ya se encuentran capturados en algún catálogo, el color por default para estos elementos es el Negro.

Color del elemento:	Para:
Azulado	Componentes ya existentes.
Azul	Componentes no analizados
Rojo	Componentes que no existen

- b. Oprima el botón izquierdo del mouse, con ello se abrirá un diálogo que contiene una paleta de colores, los cuales puede usted seleccionar.
- c. Posicione el apuntador del mouse sobre el color deseado y oprima el botón izquierdo de éste.

Si desea definir su propio color, oprima el botón de **Definir colores personalizados**.

- d. Una vez que ya se encuentra seleccionado el color, oprima el botón de **Aceptar**.


2. Seleccione el color con el cual se identificarán los componentes de la tabla, que no se encuentran capturados dentro de ningún catálogo, para ello, realice el procedimiento descrito en el punto anterior, pero seleccionando el botón relacionado con la opción de **Color componentes que no existen**, el color por default para estos elementos es el Rojo.
3. Seleccione el color con el cual aparecerán los bordes de las tablas, para ello, realice el procedimiento descrito en el punto uno, pero seleccionando el botón relacionado con la opción de **Color de cuadrícula**, el color por default para la cuadrícula es el Gris Oscuro.
4. Seleccione el checkbox **Al oprimir ENTER cambiar a la siguiente celda**, si usted desea que al terminar de editar la información dentro de una celda, con sólo oprimir la tecla ENTER pueda cambiarse a la celda siguiente, por default, esta opción no se encontrará seleccionada.

5. Seleccione el checkbox **Utilizar Flechas para terminar edición de celda**, usted desea que con sólo oprimir las flechas del teclado pueda terminar edición de información y cambiarse de celda, por default, esta opción no se encontrará seleccionada.
6. Seleccione el checkbox **Reemplazar texto al empezar a editar** si quiere sustituir el texto que se encuentra dentro de la celda por el nuevo texto a capturar. Por default, esta opción no se encontrará seleccionada, es decir, no se reemplazará el texto dentro de la celda.

Si este checkbox no se encuentra seleccionado, cuando usted capture texto dentro de una celda que ya contiene información, este texto se insertará al final del ya existente. Para configurar la información de sus Análisis, realice lo siguiente:

1. Seleccione el color con el cual se identificarán los componentes que no se encuentran analizados, el color por default para estos componentes es el Azul.
2. Seleccione el checkbox **Presentar Tipo y Clave al ver otras celdas**, si desea que los campos correspondientes al **Tipo** y la **Clave** del elemento en el Análisis siempre permanezcan visibles aunque usted se desplace a través de las columnas. Por default esta opción no se encontrará seleccionada.

Dentro de este diálogo, también se puede configurar lo siguiente:


1. Seleccione el checkbox **Limpiar pantalla de edición al Salvar datos**, si desea que al oprimir el botón de **Grabar**  dentro de cualquier ventana de edición se limpien todos los campos contenidos en ella.
2. Seleccione el checkbox **Rellenar con ceros claves numéricas (4 dígitos)**, si usted decide capturar claves numéricas y desea que se rellenen con ceros, por ejemplo, si usted desea capturar la clave 0001, y esta opción se encuentra seleccionada, usted puede teclear solamente en número 1 y el Sistema de Precios Unitarios para Windows agregará a la izquierda del número, los ceros correspondientes.

5. Seleccione el checkbox **Utilizar Flechas para terminar edición de celda**, usted desea que con sólo oprimir las flechas del teclado pueda terminar edición de información y cambiarse de celda, por default, esta opción no se encontrará seleccionada.
6. Seleccione el checkbox **Reemplazar texto al empezar a editar** si quiere sustituir el texto que se encuentra dentro de la celda por el nuevo texto a capturar. Por default, esta opción no se encontrará seleccionada, es decir, no se reemplazará el texto dentro de la celda.

Si este checkbox no se encuentra seleccionado, cuando usted capture texto dentro de una celda que ya contiene información, este texto se insertará al final del ya existente. Para configurar la información de sus Análisis, realice lo siguiente:

1. Seleccione el color con el cual se identificarán los componentes que no se encuentran analizados, el color por default para estos componentes es el Azul.
2. Seleccione el checkbox **Presentar Tipo y Clave al ver otras celdas**, si desea que los campos correspondientes al **Tipo** y la **Clave** del elemento en el Análisis siempre permanezcan visibles aunque usted se desplace a través de las columnas. Por default esta opción no se encontrará seleccionada.

Dentro de este diálogo, también se puede configurar lo siguiente:

1. Seleccione el checkbox **Limpiar pantalla de edición al Salvar datos**, si desea que al oprimir el botón de **Grabar**  dentro de cualquier ventana de edición se limpien todos los campos contenidos en ella.
2. Seleccione el checkbox **Rellenar con ceros claves numéricas (4 dígitos)**, si usted decide capturar claves numéricas y desea que se rellenen con ceros, por ejemplo, si usted desea capturar la clave 0001, y esta opción se encuentra seleccionada, usted puede teclear solamente el número 1 y el Sistema de Precios Unitarios para Windows agregará a la izquierda del número, los ceros correspondientes.

SEPARADOR FORMATOS

Dentro de este separador, usted puede definir para los siguientes conceptos: Costos y Precios, Cantidades, Importes, Días en el FSR, Factores del FSR, Factores del Costo Horario de la Maquinaria y Equipo, Porcentajes y otros Factores, la cantidad de decimales con los cuales se presentarán y cuántos decimales se tomarán en cuenta cuando se realice con alguno de ellos una operación.

Formato	Longitud	Decimales	Dec. de Cálculo
Costos y Precios	16	2	4
Cantidades	10	4	2
Importes	16	2	4
Días en FSR	8	4	4
Factores del FSR	8	4	4
Factores CHM	9	4	4
Porcentajes	7	4	4

A continuación se describirá el procedimiento que usted debe realizar para capturar esta información

1. Dentro de la celda de **Decimales**, capture la cantidad de decimales con las que desea que aparezca el concepto, por default el Sistema de Precios Unitarios le presentará estos conceptos con 2 y 4 decimales, según el concepto, para cantidades monetarias el número de decimales será igual a 2. El nombre del concepto a formatear y su longitud se presentarán dentro de las celdas de **Formato** y **Longitud** respectivamente.
2. En la celda de **Dec. de Cálculo**, capture el número de decimales que se tomarán en cuenta para realizar algún cálculo con este concepto, por default el Sistema de Precios Unitarios tomará para el cálculo 2 y 4 decimales, según el concepto.
3. Repita los pasos 1 y 2 para cada uno de los conceptos a formatear.
Usted puede observar en la celda de **Ejemplo**, como se presentaría dentro de el Sistema de Precios Unitarios el concepto formateado.

Si usted designa un número diferente de decimales para el formato y para el cálculo, puede tener las siguientes implicaciones:

- Si el número de Decimales de Cálculo, es mayor, le sumará precisión a su cálculos.
- Si el número es menor, las operaciones se realizarán con una precisión menor.



Es muy importante que el número de decimales de los formatos de pantalla e impresora para los mismos elementos coincidan, de lo contrario sucederá que en la pantalla se muestra un resultado y en el papel otro.



En el caso de que el número de Decimales del Formato, no coincida con el número de Decimales de Cálculo, ejemplo:

Formato	Decimales	Dec Cálculo
Costos y Precios	2	7

puede causar que los Importes calculados dentro de los reportes no coincidan con los cálculos que usted realizara manualmente, por ejemplo:

El Importe del componente de un Análisis se calcula como:

$$\text{IMPORTE} = (\text{CANTIDAD} / \text{RENDIMIENTO}) * \text{COSTO UNITARIO}$$

si el Costo Unitario es igual a 3.1658298 y tiene el siguiente formato:

Formato	Decimales	Dec Cálculo
Costos y Precios	2	7

dentro de los reportes se observaría lo siguiente.

Cantidad	Moneda	Costo Unitario	Rendimiento	Importe
3	N\$	3.16	1	9.49

que como se puede observar, el Importe no es el resultado de multiplicar 3 por 3.16 el cual nos daría un Importe de 9.48.

DITANDO INFORMACIÓN DE LA OBRA DENTRO DEL PANEL DE CONTROL

El Panel de Control le permite editar Información General de su Obra, éste aparecerá después de que usted creó una Obra. Si usted se encuentra dentro de cualquier opción del Sistema de Precios Unitarios, puede invocarlo, si oprimo simultáneamente las teclas CTRL+F5.

El Panel de Control está formado por varios separadores: **Proyecto**, **Concurso**, **Reportes**, **Notas** y **Usuarios**, estos separadores permiten editar los datos generales de la Obra y cada uno contiene un conjunto de botones que nos permiten manipular la información.

En la continuación se describirá el procedimiento que se debe seguir para capturar información en cada uno de ellos.

SEPARADOR PROYECTO

En él usted podrá editar los siguientes datos: el nombre de la Obra, la fecha de inicio y terminación de la Obra, el Propietario, la Constructora y la Dirección, la información



En el caso de que el número de Decimales del Formato, no coincida con el número de Decimales de Cálculo, ejemplo:

Formato	Decimales	Dec Cálculo
Costos y Precios	2	7

puede causar que los Importes calculados dentro de los reportes no coincidan con los cálculos que usted realizara manualmente, por ejemplo:

El Importe del componente de un Análisis se calcula como:

$$\text{IMPORTE} = (\text{CANTIDAD} / \text{RENDIMIENTO}) * \text{COSTO UNITARIO}$$

si el Costo Unitario es igual a 3.1658298 y tiene el siguiente formato:

Formato	Decimales	Dec Cálculo
Costos y Precios	2	7

dentro de los reportes se observaría lo siguiente:

Cantidad	Moneda	Costo Unitario	Rendimiento	Importe
3	N\$	3.16	1	9.49

que como se puede observar, el Importe no es el resultado de multiplicar 3 por 3.16 el cual nos daría un Importe de 9.48.

DITANDO INFORMACIÓN DE LA OBRA DENTRO DEL PANEL DE CONTROL

El Panel de Control le permite editar Información General de su Obra, éste aparecerá, después de que usted creó una Obra. Si usted se encuentra dentro de cualquier opción del Sistema de Precios Unitarios, puede invocarlo, si oprime simultáneamente las teclas CTRL+F5.

El Panel de Control está formado por varios separadores: **Proyecto**, **Concurso**, **Reportes**, **Notas** y **Usuarios**, estos separadores permiten editar los datos generales de la Obra y cada uno contiene un conjunto de botones que nos permiten manipular la información.

En la continuación se describirá el procedimiento que se debe seguir para capturar información en cada uno de ellos.

SEPARADOR PROYECTO

En él usted podrá editar los siguientes datos: el nombre de la Obra, la fecha de inicio y terminación de la Obra, el Propietario, la Constructora y la Dirección, la información

aquí capturada, además de servir como referencia, es de gran utilidad para el diseño de Reportes, colocando estos datos en los encabezados. Para editar los datos mencionados anteriormente, es necesario realizar lo siguiente:

1. Dentro del campo **Fecha de Inicio**, introduzca la fecha de inicio de la Obra.
2. En el campo **Fecha de Terminación**, introduzca la fecha de terminación de la Obra.

Para introducir las Fechas de Inicio y Terminación, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas:

Oprima la Tecla: Si desea:

F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios
F9	Limpiar el campo de fecha

3. Dentro del campo de **Propietario**, proporcione el nombre de la persona o empresa propietaria de la Obra, dentro de este campo de edición usted puede capturar hasta 120 caracteres.
4. En el campo de **Constructora**, capture el nombre de la Empresa Constructora que presenta el concurso.
5. Dentro del campo de **Dirección**, capture el sitio donde se encuentra la Obra (Calle, Número, Colonia, etc.).
6. Si desea guardar los datos que usted proporcionó oprima el botón de **Grabar**, de lo contrario oprima el botón de **Anular**.

SEPARADOR CONCURSO

Dentro de este separador usted puede editar información relevante del Concurso, como son el Número de Concurso, la Fecha del Concurso, el Representante Legal y datos de la Constructora, la información aquí capturada, además de servir como

referencia, es de gran utilidad para el diseño de Reportes, colocando estos datos en los encabezados. Para la edición de los datos mencionados, es necesario llevar a cabo los siguientes pasos

1. Dentro del campo de **Número de Concurso**, proporcione el número de éste. El número no puede ser mayor a 40 caracteres.
2. En el campo de **Fecha del Concurso**, introduzca la fecha de realización del concurso.
Para introducir la Fecha del Concurso, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas.

Oprima la Tecla:

Si desea:

F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios
F9	Limpiar el campo de fecha

3. Dentro del campo de **Representante Legal**, proporcione el nombre de dicho representante.
4. En el campo de **Registro del CNIC**, capture el registro de la Cámara Nacional de la Industria de la Construcción, este registro no puede ser mayor a 20 caracteres.
5. Dentro del campo **Número de I.V.A.**, indique el número de registro del Impuesto al Valor Agregado (IVA), este registro no puede ser mayor a 20 caracteres
6. Dentro del campo de **R.F.C.**, proporcione el Registro Federal de Causantes, éste no puede ser mayor a 20 caracteres.
7. Si desea guardar los datos que usted proporcionó oprima el botón de **Grabar**, de lo contrario oprima el botón de **Anular**.

referencia, es de gran utilidad para el diseño de Reportes, colocando estos datos en los encabezados. Para la edición de los datos mencionados, es necesario llevar a cabo los siguientes pasos:

1. Dentro del campo de **Número de Concurso**, proporcione el número de éste. El número no puede ser mayor a 40 caracteres.
2. En el campo de **Fecha del Concurso**, introduzca la fecha de realización del concurso.
Para introducir la Fecha del Concurso, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas:

Oprima la Tecla:	Si desea:
F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios
F9	Limpiar el campo de fecha

3. Dentro del campo de **Representante Legal**, proporcione el nombre de dicho representante.
4. En el campo de **Registro del CNIC**, capture el registro de la Cámara Nacional de la Industria de la Construcción, este registro no puede ser mayor a 20 caracteres.
5. Dentro del campo **Número de I.V.A.**, indique el número de registro del Impuesto al Valor Agregado (IVA), este registro no puede ser mayor a 20 caracteres
6. Dentro del campo de **R.F.C.**, proporcione el Registro Federal de Causantes, éste no puede ser mayor a 20 caracteres
7. Si desea guardar los datos que usted proporcionó oprima el botón de **Grabar**, de lo contrario oprima el botón de **Anular**.

SEPARADOR IMPORTES

La información que usted puede editar dentro de este separador es la relacionada con el Importe Total de la Obra, en él puede indicar si el Importe Total de la Obra o Presupuesto se va a manejar a Costo Directo o a Precio Unitario y si el importe será Automático, es decir, se manejará el importe que resulte del cálculo del presupuesto, o Asignado, es decir, capturando esta opción. Los datos en este separador son únicamente de carácter informativo, con el objeto de conocer de una manera rápida el Importe Total de la Obra y el Anticipo otorgado.

Para realizar la edición de los datos anteriores, es necesario ejecutar los siguientes pasos:

1. Seleccione el radiobutton asociado al cálculo del Importe, éste puede ser solicitado a **Costo Directo** o a **Precio Unitario**.
2. Seleccione el radiobutton asociado al Tipo de Cálculo del Importe, éste puede ser **Automático, Asignado**. Si es Automático, el Sistema de Precios Unitarios calculará el Importe Total de la Obra, sumando todos los Importes Totales de los Análisis existentes en el mayor nivel (Partidas o Capítulos).
3. Si seleccionó el **Importe Automático**, y desea considerar los extraordinarios marque el checkbox **Incluye Extraordinarios**.
4. Si seleccionó el **Importe Asignado**, dentro del campo de **Imp. Total** proporcione el importe de la Obra.
5. Seleccione el radiobutton asociado con el tipo de **Anticipo de la Obra**, éste puede ser por **Porcentaje** o **Importe**.
6. Si seleccionó el **Porcentaje**, dentro del campo de edición que se encuentra junto a dicha opción, capture el porcentaje del Importe Total de la Obra que será el anticipo.
7. Si seleccionó **Importe**, dentro del campo de edición **Anticipo**, proporcione el anticipo del Importe.

8. Si desea guardar los datos que usted proporcionó oprima el botón de **Grabar**, de lo contrario oprima el botón de **Anular**.

SEPARADOR NOTAS

El separador de Notas le permite almacenar cualquier comentario referente a la Obra o el Presupuesto que se está presentando, las Notas de la Obra sólo son de carácter informativo, para editarlas es necesario realizar lo siguiente:

1. Dentro del campo de edición de **Notas de la Obra**, capture cualquier tipo de información asociada con su Obra.
2. Si desea guardar los datos que usted proporcionó oprima el botón de **Grabar**, de lo contrario oprima el botón de **Anular**.

SEPARADOR USUARIOS

Este separador le permite conocer con cuántos y con cuáles usuarios se encuentra compartiendo su Obra, esto es si usted está trabajando con el Sistema de Precios Unitarios para Windows en un ambiente de Red, en él no podrá capturar información, sólo podrá consultarla.

1. Dentro del separador de **Usuarios**, podrá observar con cuántos usuarios se encuentra compartiendo su Obra, la fecha y la hora en la que hicieron acceso.



Al oprimir el botón de **Grabar** en este separador, se grabará la información que usted capturó en los separadores de **Proyecto**, **Concurso**, **Importes** y **Notas**, si oprime el botón de **Anular**, la información editada en los separadores mencionados anteriormente no se almacenará.

Para:

Ver:

Obtener información acerca del Panel de Control


El tema "**Panel de Control**" dentro de la Ayuda en Línea del Sistema de Precios Unitarios

Observar el procedimiento para editar los Datos Generales de la Obra

Consultar el tema "**Editar datos dentro del Panel de Control**" dentro de los Tips del Sistema de Precios Unitarios

CÓMO BORRAR UNA OBRA

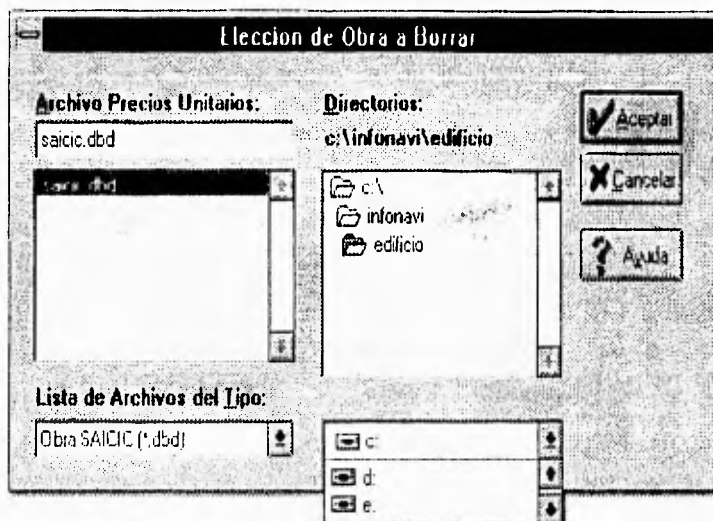
Si desea eliminar por completo toda la información de una Obra que fue generada con el Sistema de Precios Unitarios para Windows es necesario que realice lo siguiente:

1. Oprima el botón  de la barra de Herramientas General correspondiente al borrado de una Obra.
O seleccione del menú de **Utilerías** la opción de **Borrar una Obra**.
2. Dentro del combobox de **Unidad**, seleccione la unidad de disco en la cual se encuentra la Obra que desea borrar.
3. Dentro de la lista de **Directorios**, seleccione el subdirectorío de la Obra a borrar.
4. Seleccione dentro de la lista de **Nombre Archivo** el archivo .DBD, recuerde que este archivo nos permite identificar a una Obra del Sistema de Precios Unitarios para Windows.
5. Si ya se encuentra seleccionado el archivo correspondiente a la Obra que desea borrar oprima el botón de **Aceptar**, si desea suspender el borrado de la Obra oprima el botón de **Cancelar**.

M

Al utilizar esta opción debe tener cuidado, ya que una vez que la Obra ha sido borrada, no habrá manera de recuperarla.

EJEMPLO: En este ejemplo se desea borrar la obra EDIFICIO que fue creada con el Sistema de Precios Unitarios para Windows, esta obra se encuentra dentro del directorio INFONAVI\EDIFICIO en la unidad de disco C:\, para ello, es necesario editar lo siguiente dentro del diálogo de Elección de Obra a Borrar y posteriormente oprimir el botón de **Aceptar**.



CÓMO CERRAR UNA OBRA

Si usted terminó de realizar su Presupuesto o simplemente desea cerrar la Obra con la que se encuentra trabajando, es necesario realizar lo siguiente:

Seleccione del menú de **Archivo** la opción de **Cerrar Obra**.

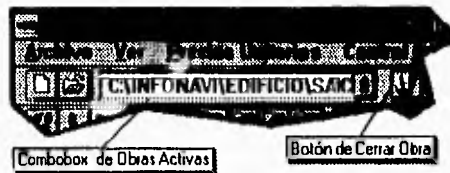
O dentro de la Barra de Herramientas Sistema de Precios Unitarios:

1. Seleccione dentro del combobox de **Obras Activas** la Obra que desea cerrar.
2. Una vez seleccionada la Obra que desea cerrar oprima el botón de **Cerrar Obra**.

Esta opción sólo se presentará en el menú de **Archivo**, cuando no se encuentre ninguna ventana abierta (por ejemplo, de edición de Materiales o de edición de Básicos) dentro del Sistema de Precios Unitarios para Windows.

M Es necesario que antes de cerrar la Obra, verifique que todas las modificaciones que efectuó sobre la Obra se encuentren salvadas.

Figura 1.1 En este ejemplo se desea cerrar la Obra que se encuentra dentro clasificada como: C:\INFONAVI\EDIFICIO, para ello, es necesario que dentro del combobox de Obras Activas seleccione esta Obra y oprima el botón de **Cerrar Obra**.



Para:

Obtener información acerca de cómo cerrar las Obras Activas dentro del Sistema de Precios Unitarios

Ver:

La opción de "**Cerrar una Obra**" dentro del tema "**Menú de Archivo**" en la Ayuda en Línea del Sistema de Precios Unitarios

Procedimiento **Para obtener información**

1. Ejecute el Programa de Instalación de SAICIC para Windows



Consulte el capítulo 2 "Instalación de SAICIC" en el Manual de Usuario.

2. Utilice el Programa de Intercambio de Información incluido dentro del grupo de SAICIC para Windows, el cual le permitirá convertir sus Obras generadas en SAICIC para DOS a Obras reconocidas en SAICIC para Windows



Consulte el Capítulo 16 "XIE Intercambio de Información" contenido en el Manual de Usuario.

3. Abra la nueva Obra generada en Windows desde el diálogo de "Apertura de Obra SAICIC"



Consulte el Capítulo 3 "Iniciando con SAICIC" contenido en el Manual de Usuario

SALIR DEL SISTEMA DE PRECIOS UNITARIOS

Esta opción del Sistema de Precios Unitarios para Windows le permite terminar una sesión de trabajo, y regresar al Administrador de Programas, para realizar esta acción es necesario que seleccione la opción de **Salir** del menú de **Archivo**.

Usted también podrá salir del Sistema de Precios Unitarios, ejecutando cualquiera de las siguientes acciones:

- Oprimiendo simultáneamente las teclas: ALT+F4.
- O seleccionando la opción **Close** del menú del sistema.



Manual de Usuario

Antes de salir del Sistema de Precios Unitarios para Windows, es necesario que salve toda la información que se modificó dentro de la Obra, ésto lo puede llevar a cabo, presionando el botón de **Grabar** de cada ventana que contenga información factible de almacenar en su Obra.

Para:

Ver:

Obtener información acerca de cómo salir del Sistema de Precios Unitarios

La opción de **"Salir"** dentro del tema **"Menú de Archivo"** en la Ayuda en Línea del Sistema de Precios Unitarios

SECCIÓN 4 APRENDIENDO LAS PRINCIPALES OPCIONES DE EDICIÓN



Dentro de los módulos del sistema de precios unitarios para Windows que permiten la edición, existen procedimientos que se efectúan de manera semejante, por lo que en este capítulo describiremos sólo los principales. Una vez que usted se familiarice con ello, usted podrá aplicarlos dentro de cualquier módulo de edición en el sistema de precios unitarios.

COMO DESPLAZARSE ENTRE REGISTROS


A lo largo de este tema describiremos cómo puede realizar un desplazamiento entre registros en el sistema de precios unitarios, ya sea con un desplazamiento uno a uno, o desplazándose hasta el principio o al final de ellos.



Si no existen registros capturados dentro de la Obra, los botones de Registro Posterior, Registro Anterior, Ultimo Registro y Primer Registro permanecerán deshabilitados.

REGISTRO POSTERIOR

Si usted desea realizar un desplazamiento al siguiente registro y que EL SISTEMA DE PRECIOS UNITARIOS le presente los datos asociados en pantalla, realice lo siguiente:


1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.
O presione la tecla F4.
O seleccione del menú de Edición la opción de **Registro Posterior**.



Si usted se encuentra en el último registro y avanza un registro, aparecerán los datos del primer registro capturado. A esto le llamamos presentación circular.

REGISTRO ANTERIOR

Si por el contrario, usted desea realizar un desplazamiento al registro anterior y que EL SISTEMA DE PRECIOS UNITARIOS le presente los datos asociados, realice lo siguiente:


1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.
O presione la tecla F3.
O seleccione del menú de **Edición** la opción de **Registro Anterior**.
-



Si usted se encuentra en el primer registro y retrocede un registro, aparecerán los datos del último registro capturado. A esto le llamamos presentación circular.

ULTIMO REGISTRO


Si usted quiere avanzar al último registro y no desea realizar el desplazamiento, registro por registro, realice lo siguiente:

1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.

- O presione simultáneamente las teclas CTRL+F4.
- O seleccione del menú de **Edición** la opción de **Ultimo Registro**.


PRIMER REGISTRO

Si desea presentar el primer registro:

1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.
 - O presione simultáneamente las teclas CTRL+F3.
 - O seleccione del menú de **Edición** la opción de **Primer Registro**.

CÓMO GRABAR UN REGISTRO

Si usted ya editó todos los datos relacionados con un registro y desea grabarlos:

1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.
 - O si así lo desea oprima la tecla F2.
 - O seleccione del menú de **Edición** la opción de **Grabar Registro**.


Una vez que el registro se grabó, automáticamente los campos dentro de la ventana se limpian, claro a menos de que usted haya declarado un prefijo para la Clave del Insumo o Análisis, o que haya definido una Fecha de Cálculo y de Costo, dentro del diálogo de Configuración del sistema de precios unitarios.



El botón de **Grabar**, se encontrará deshabilitado, hasta que usted proporcione la Clave del Insumo, Análisis, Herramienta o Estimación.


CÓMO LIMPIAR LOS CAMPOS DE EDICIÓN

Si usted desea limpiar todos los campos relacionados con un registro para realizar la captura de uno nuevo:

1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.
 - O presione simultáneamente las teclas CTRL+L.
 - O seleccione del menú de **Edición** la opción de **Limpiar Campos**.

CÓMO BORRAR UN REGISTRO

Si desea borrar un registro:

1. Usted deberá desplazarse a través de los registros hasta que los datos del registro que desea borrar estén presentes.
2. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.
 - O presione la tecla F5.
 - O seleccione del menú de **Edición** la opción **Borrar Registro**.

Cuando el registro que desea borrar es un Insumo o Análisis y éste se encuentra incluido a su vez en otro Análisis o Estimación, EL SISTEMA DE PRECIOS UNITARIOS para Windows le presentará un mensaje en donde le preguntará si desea eliminar el elemento aunque se encuentre contenido en un Análisis o Estimación, si selecciona la opción de:

Borrar únicamente el elemento - Este elemento se borrará dentro del catálogo de Insumos o Análisis, mas no así del Análisis que lo contiene. Si usted selecciona esta opción, el Insumo o Análisis puede todavía ser recuperado, con sólo grabar nuevamente el Análisis o Estimación que lo contiene.

Borrar el elemento y sus ligas con otros elementos - Si usted selecciona esta opción el Insumo o Análisis se borrará de su catálogo y de todos los Análisis o Estimaciones a los que pertenezca.



El botón de **Borrar**, se encontrará deshabilitado, si el registro que se encuentra presente no contiene la Clave del Insumo o Análisis.



Antes de borrar un registro, es necesario que se asegure que el registro que se encuentra visualizando en la ventana, es el mismo que desea borrar, ya que una vez eliminado éste de la base de datos, no habrá manera de recuperarlo.

Para:

Obtener más información acerca de los comandos principales de edición

Ver:

El tema "**Menú de Edición**" dentro de la Ayuda en Línea del sistema de precios unitarios para Windows


CÓMO REALIZAR BÚSQUEDAS DE INFORMACIÓN DENTRO DEL SISTEMA DE PRECIOS UNITARIOS

Dentro del sistema de precios unitarios para Windows usted puede realizar la búsqueda de un Insumo o Análisis a través de su clave o su descripción.

Las búsquedas sólo se realizarán en base al módulo del sistema de precios unitarios en el que se encuentre, por ejemplo, si se encuentra editando un Material, usted sólo puede realizar la búsqueda sobre los Materiales de la Obra.

Búsquedas por Clave

Para realizar una búsqueda por clave:

1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS.

O seleccione del menú de **Edición** la opción de **Búsqueda de Datos**, con ello aparecerá un submenú, de éste seleccione la opción **Buscar Registro**.

Una vez dentro del diálogo de Búsquedas:

2. Seleccione el separador **Por Clave**.
3. Dentro del campo de edición de **Clave a Buscar**, teclee la clave del Insumo o Análisis que desea buscar.
4. Si desea realizar una búsqueda exacta de la clave del Insumo o Análisis, marque el checkbox de **Realizar Búsqueda Exacta**, EL SISTEMA DE PRECIOS UNITARIOS le traerá el Insumo o Análisis con la clave proporcionada, si éste no existe, le presentará un mensaje de notificación en el cual le informa que ese elemento no se encuentra en la base de datos.


Si por el contrario, este checkbox no se encuentra seleccionado y el Insumo o Análisis no fué encontrado, EL SISTEMA DE PRECIOS UNITARIOS le presentará el elemento que contenga la Clave más aproximada a la que usted proporcionó para la búsqueda.

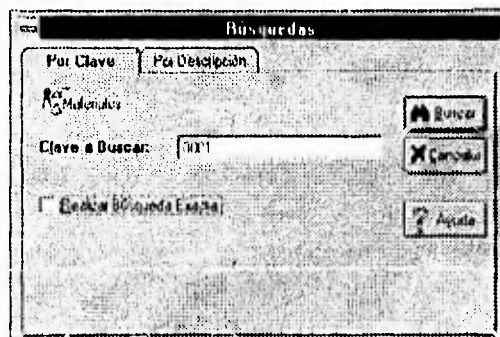
5. Una vez que ha proporcionado la Clave, oprima el botón de **Buscar**, para iniciar la búsqueda del registro, una vez encontrado, EL SISTEMA DE PRECIOS UNITARIOS lo posicionará en el registro con la clave elegida, si desea suspender la búsqueda por clave oprima el botón de **Cancelar**.
6. EL SISTEMA DE PRECIOS UNITARIOS para Windows le permite observar todos los Insumos o Análisis, cuya clave, contiene los caracteres que usted proporcionó para realizar la búsqueda, para ello, oprima dentro de la Barra de Búsquedas del SISTEMA DE PRECIOS UNITARIOS, el botón de **Buscar Siguiente**.

7. Si desea realizar otra búsqueda, dentro del campo de edición **Cadena a Buscar** de la Barra de Búsquedas, proporcione la nueva clave, si por el contrario desea suspender la búsqueda oprima el botón de **Cancelar**.

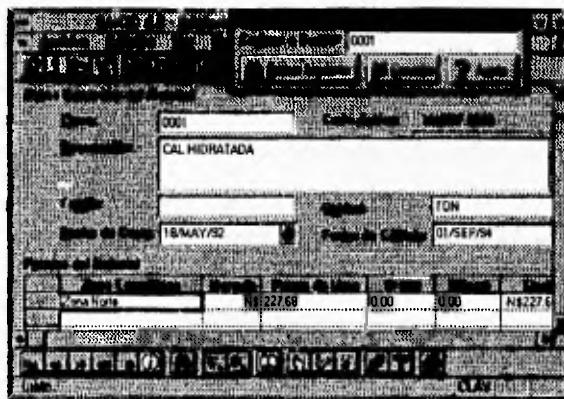
Si ahora desea realizar la búsqueda de un registro, a través de la Descripción:

- a. Posicione el apuntador del mouse dentro de la Barra de Búsquedas y oprima el botón derecho de éste, con ello aparecerá un menú, de él seleccione la opción de **Búsqueda por Descripción**, al seleccionarla aparecerá un submenú el cual le indicará dónde desea buscar la cadena dentro de la descripción, esto puede ser al inicio (**Buscar al Inicio**), al final (**Buscar al Final**), o no importa su posición (**No importa su posición**).
- b. Dentro del campo de edición **Cadena a Buscar** proporcione la cadena de la descripción a buscar.

EJEMPLO: En el siguiente ejemplo se desea buscar el Material con la Clave 001 que se encuentra dentro de la Obra Edificio, para ello, después de seleccionar el botón de búsqueda , es necesario capturar la siguiente información dentro del diálogo de Búsquedas.




Al oprimir el botón de buscar aparecerá dentro de la ventana de Materiales la siguiente información, observe que mediante la Barra de Herramientas de Búsqueda usted puede seguir buscando los elementos que contengan dentro de su Clave la Cadena a Buscar:



BÚSQUEDAS POR DESCRIPCIÓN

Para realizar una búsqueda por Descripción:

1. Oprima el botón  dentro de la Barra de Captura del SISTEMA DE PRECIOS UNITARIOS. O seleccione del menú de Edición la opción de **Búsqueda de Datos**, con ello aparecerá un submenú, de éste seleccione la opción **Buscar Registro**.

Una vez dentro del diálogo de Búsqueda:

2. Seleccione el separador **Por Descripción**.

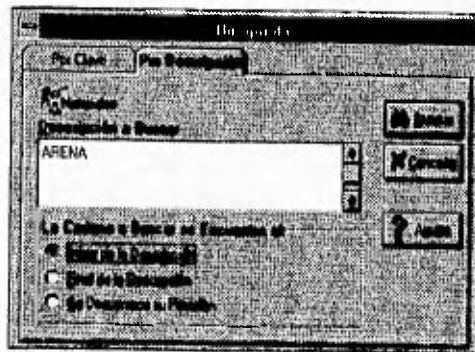
3. Dentro del campo de edición de **Descripción a Buscar**, teclee la descripción del Insumo o Análisis que desea buscar, si no recuerda toda la descripción usted puede proporcionar sólo una parte de ella.
4. Indique en qué posición de la descripción se encuentra la cadena que desea buscar, si se encuentra al principio de la descripción seleccione la opción **Inicio de la Descripción**, si la cadena se encuentra al final, seleccione la opción **Final de la Descripción** si usted no sabe la localización exacta de la cadena, seleccione la opción **Se Desconoce su Posición**.
5. Una vez que proporcionó la cadena a buscar oprima el botón de **Buscar**, EL SISTEMA DE PRECIOS UNITARIOS lo posicionará en el registro que contiene la cadena proporcionada, si desea suspender la búsqueda por Descripción oprima el botón de **Cancelar**.
6. Si desea observar todos los registros que contengan la descripción con la que efectuó la búsqueda, oprima dentro de la Barra de Búsquedas del SISTEMA DE PRECIOS UNITARIOS, el botón de **Buscar Siguiente**, si por el contrario desea suspender la búsqueda oprima el botón de **Cancelar**.
7. Si desea realizar otra búsqueda, dentro del campo de edición **Cadena a Buscar** de la Barra de Búsquedas, proporcione la nueva descripción, si por el contrario desea suspender la búsqueda oprima el botón de **Cancelar**.

Si estando en búsquedas por Descripción y desea cambiar a búsquedas por Clave:

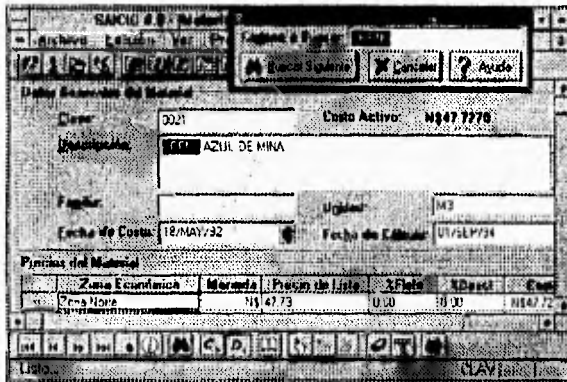
- a. Posicione el apuntador del mouse dentro de la Barra de Búsquedas y oprima el botón derecho de éste, con ello aparecerá un menú, de él seleccione la opción de **Buscar por Clave**.
- b. Dentro del campo de edición de Cadena a Buscar proporcione la clave a buscar.

EJEMPLO:

Dentro de la Obra Edificio, se desea buscar todos los Materiales que contiene la palabra ARENA al inicio de su Descripción, para ello, después de seleccionar el botón de búsqueda **A**, es necesario capturar la siguiente información dentro del diálogo de Búsquedas.



Al oprimir el botón de buscar aparecerá dentro de la ventana de **Materiales** la siguiente información, observe que mediante la Barra de Herramientas de Búsqueda usted puede seguir buscando los elementos que contengan dentro de su Descripción la Cadena a Buscar:



Para:

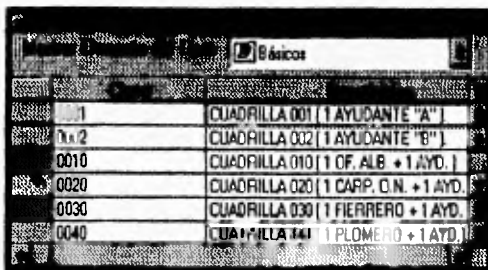
Obtener información acerca de las búsquedas en el sistema de precios unitarios

Ver:

El tema "**Búsquedas por Clave o Por Descripción**" dentro de la Ayuda en Línea del sistema de precios unitarios


CONSULTA INMEDIATA DE CATÁLOGOS

Con esta opción, EL SISTEMA DE PRECIOS UNITARIOS para Windows le ofrece la posibilidad de consultar en forma inmediata los catálogos de Insumos: Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos y los catálogos de Análisis: Básicos, Integrados, Partidas, Capítulos y Extraordinarios, en el mismo momento en el que usted está realizando la captura de cualquier Insumo o Análisis, por ejemplo dentro del Costo Hora/Máquina y de los Análisis mismos, lo que elimina la molestia de estar consultando constantemente Catálogos Impresos



Clave	Descripción
001	CUADRILLA 001 [1 AYUDANTE "A"]
002	CUADRILLA 002 [1 AYUDANTE "B"]
0010	CUADRILLA 010 [1 OF. ALB. + 1 AYD.]
0020	CUADRILLA 020 [1 CARP. O.N. + 1 AYD.]
0030	CUADRILLA 030 [1 FERRERO + 1 AYD.]
0040	CUADRILLA 040 [1 PLOMERO + 1 AYD.]

CÓMO SELECCIONAR UN ELEMENTO DEL CATÁLOGO

1. Seleccione esta opción del sistema de precios unitarios desde el menú de **Ver** u oprima el botón  que se encuentra dentro de la Barra de Opciones, con ello aparecerá una ventana conteniendo este Catálogo que le muestra la Clave y la Descripción de los elementos que conforman al catálogo de Insumos (Materiales, Obra de Mano, Maquinaria y Equipo o Subcontratos) o Análisis (Básicos, Integrados, Partidas, Capítulos o Extraordinarios), según usted lo seleccione.
2. Si usted desea observar los datos relacionados con un elemento del Catálogo, en el combobox **Mostrar el Elemento del Tipo** seleccione el tipo de Insumo o Análisis del cual desea observar todos sus elementos, estos pueden ser: Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos, Básicos, Integrados, Partidas, Capítulos y Extraordinarios.
3. Posicione el apuntador del mouse sobre el elemento del cual quiere obtener información y:

Oprima:

Si:

Una vez el botón izquierdo del mouse

Desea observar los datos del Insumo o Análisis.

Dos veces el botón izquierdo del mouse. Además de observar los datos del Insumo o Análisis, desea cerrar el Catálogo.



Cuando usted se encuentra capturando un componente en un Análisis o un Insumo dentro del Costo Hora/Máquina y realiza cualquiera de las dos acciones anteriores, el elemento seleccionado y sus datos, se agregarán como un componente del Análisis o del Costo Hora/Máquina.

CÓMO CERRAR EL CATÁLOGO

Usted puede salir del Catálogo, con sólo posicionar el apuntador del mouse sobre el menú del sistema y seleccionar la opción de **Close**.



Para:

Obtener información acerca del Catálogo del SISTEMA DE PRECIOS UNITARIOS

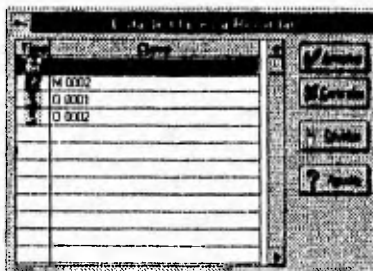
Ver:

El tema "**Menú Ver**" dentro de la Ayuda en Línea del sistema de precios unitarios para Windows

CÓMO MANEJAR LAS CLAVES A RECORDAR DENTRO DEL SISTEMA DE PRECIOS UNITARIOS.


EL SISTEMA DE PRECIOS UNITARIOS para Windows presenta dentro de esta versión el Manejo de Claves a Recordar. esta característica le permitirá llevar un registro de los Insumos o Análisis que usted considere más importantes dentro de su Obra. Este registro se mantendrá hasta que usted cierre la Obra.

Por medio de esta lista usted podrá tener un acceso más rápido hacia los Insumos o Análisis que se encuentran en ella.



CÓMO AGREGAR UNA CLAVE A LA LISTA DE CLAVES A RECORDAR.


Para agregar un Insumo o Análisis dentro de la Lista de Claves a recordar:

1. Usted deberá desplazarse a través de los registros hasta que los datos del registro que desea guardar en la lista, estén presentes en la ventana.
Para ello puede realizar la búsqueda de dicho registro o bien utilizar los botones de desplazamiento entre registros que se encuentran dentro de la Barra de Captura.
2. Oprima el botón  dentro de la Barra de Captura, con ello la clave será almacenada como una clave a recordar, usted puede agregar los Insumos o Análisis que desee a esta lista.

Usted puede almacenar la cantidad de elementos que desee a la Lista de Claves a Recordar.


CÓMO IR A UN INSUMO O ANÁLISIS DESDE LA LISTA DE CLAVES A RECORDAR.

Para ir a un Insumo o Análisis desde la Lista de Claves a Recordar:

1. Oprima el botón  que se encuentra dentro de la Barra de Captura, con lo cual se abrirá una ventana que contiene la lista de Claves a Recordar.

1. En la ventana abierta, posicione en el renglón que contiene la Clave del Insumo o Análisis que desea observar.
2. Oprima dos veces el botón izquierdo del mouse u oprima el botón de **Aceptar**.
3. Automáticamente aparecerá la ventana correspondiente al Insumo o Análisis solicitado.

CÓMO ELIMINAR UN ELEMENTO DE LA LISTA DE CLAVES A RECORDAR

1. Oprima el botón  que se encuentra dentro de la Barra de Captura, con lo cual se abrirá una ventana que contiene la lista de Claves a Recordar.
2. En la ventana abierta, posicione en el renglón que contiene la Clave del Insumo o Análisis que desea eliminar de la lista.
3. Oprima el botón de **Olvidar**.

CÓMO LIMPIAR LA LISTA DE CLAVES A RECORDAR

Para eliminar todos los elementos de la Lista de Claves a Recordar:

- 1 Oprima el botón  dentro de la Barra de Captura

Para:

Obtener información acerca de la Lista de Claves a Recordar

Ver:

El tema "**Lista de Claves a Recordar**" dentro de la Ayuda en Línea del sistema de precios unitarios

SECCIÓN 5 FACTOR DE SALARIO REAL



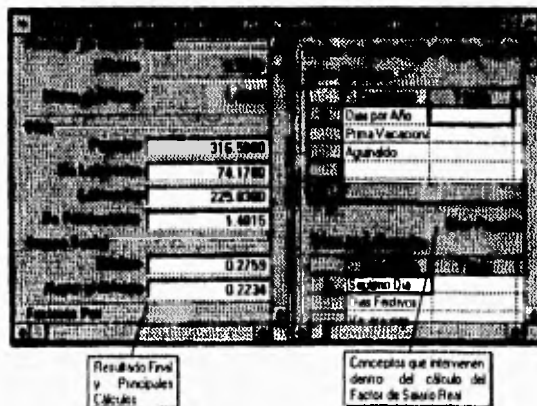
Dentro de este módulo del sistema de precios unitarios para Windows, se calculará el Factor de Salario Real, para salarios mínimos, o para mayores al mínimo, el cálculo podrá ser para Obra Pública o Privada.

EL SISTEMA DE PRECIOS UNITARIOS para Windows le ofrece la ventaja de modificar los factores relacionados con la Ley de Obra Pública o con el Seguro Social y agregar nuevos conceptos para considerarlos dentro de su cálculo final.

EL SISTEMA DE PRECIOS UNITARIOS para Windows le presentará la ventana del Factor de Salario Real dividida en dos partes:

1a. Parte - Contiene el resultado final del Factor de Salario Real, así como sus principales cálculos.

2a. Parte - Puede capturar los diferentes factores según la Ley Federal del Trabajo, por Seguro Social y por Otros Conceptos.



EDICIÓN DEL FACTOR DE SALARIO REAL

Para modificar los factores:

1. Desde la Barra de Herramientas oprima el botón  correspondiente al Factor de Salario Real.

O del menú de **Precios Unitarios**, seleccione la opción de **Factor de Salario Real** y aparecerá la ventana correspondiente a este módulo.


Una vez que usted se encuentra dentro de la ventana del Factor de Salario Real:

DÍAS PAGADOS

2. Oprima simultáneamente las teclas ALT+P, para posicionarse dentro de las celdas de captura de los **Días Pagados**.
3. Capture el **Título del Concepto** para los **Días por Año**, este campo es opcional y como ya lo explicamos anteriormente tiene por objeto el cambiar el nombre del concepto dentro de sus reportes.
4. Dentro de la celda de **Dato** proporcione la cantidad de días para los **Días por Año**.
5. Capture el **Título** y el **Dato** para la Prima Vacacional y el Aguinaldo repitiendo los pasos 3 y 4.

EDICIÓN DEL FACTOR DE SALARIO REAL

Para modificar los factores:

1. Desde la Barra de Herramientas oprima el botón  correspondiente al Factor de Salario Real.

O del menú de **Precios Unitarios**, seleccione la opción de **Factor de Salario Real** y aparecerá la ventana correspondiente a este módulo.

Una vez que usted se encuentra dentro de la ventana del Factor de Salario Real:

DÍAS PAGADOS

2. Oprima simultáneamente las teclas ALT+P, para posicionarse dentro de las celdas de captura de los **Días Pagados**.
3. Capture el **Título del Concepto** para los **Días por Año**, este campo es opcional y como ya lo explicamos anteriormente tiene por objeto el cambiar el nombre del concepto dentro de sus reportes.
4. Dentro de la celda de **Dato** proporcione la cantidad de días para los **Días por Año**.
5. Capture el **Título** y el **Dato** para la Prima Vacacional y el Aguinaldo repitiendo los pasos 3 y 4.

6. Si desea agregar un nuevo concepto a los **Días Pagados** posicione en el renglón correspondiente al concepto **Otros 1** que como puede usted observar se encuentra vacío.
7. Repita los pasos 3 y 4 para cada uno de los conceptos nuevos (**Otros 1**, **Otros 2** y **Otros 3**).



Usted puede agregar hasta tres conceptos nuevos, para ello, cuente con los conceptos **Otros 1**, **Otros 2** y **Otros 3**.

Dentro del campo de **Días Pagados** que se encuentra en la parte superior izquierda de la ventana del **Factor de Salario Real**, se presentará el **Total de Días**, el cual se calcula como:

DÍAS POR AÑO + PRIMA VACACIONAL + AGUINALDO + OTROS 1 + OTROS 2 + OTROS 3

Si usted realiza algún cambio en el número de días, éste se reflejará inmediatamente dentro del campo de **Días Pagados**.

DÍAS NO LABORADOS

8. Oprima simultáneamente las teclas **ALT+L**, para posicionarse dentro de las celdas de captura de los **Días No Laborados**.
9. Capture el **Título** y el **Dato** de todos los conceptos relacionados con los **Días No Laborados** (**Séptimo Día**, **Días Festivos**, **Vacaciones**, **Fiestas de Costumbre**, **Enfermedad** y **Mal Tiempo**), repitiendo los pasos 3 y 4.

10. Si desea agregar un nuevo concepto a los **Días No Laborados** posicione en el renglón correspondiente al concepto **Otros 1** que como puede observar se encuentra vacío.
11. Repita los pasos 3 y 4 para cada uno de los conceptos nuevos (**Otros 1**, **Otros 2**, **Otros 3**).



Usted puede agregar hasta tres conceptos nuevos, para ello, cuente con los conceptos **Otros 1**, **Otros 2** y **Otros 3**.

Dentro del campo de **Días No Laborados** que se encuentra en la parte superior izquierda de la ventana del **Factor de Salario Real**, se presentará el **Total de Días No Laborados**, el cual se calcula como:

SÉPTIMO DÍA + DÍAS FESTIVOS + VACACIONES + FIESTAS DE COSTUMBRE +
ENFERMEDAD + MAL TIEMPO + OTROS 1 + OTROS 2 + OTROS 3

Si usted realiza algún cambio en el número de días no laborados, éste se reflejará inmediatamente dentro del campo de **Días No Laborados**.

SEGURO SOCIAL

12. Oprima simultáneamente las teclas **ALT+S**, para posicionarse dentro de las celdas de captura del **Factor de Salario Real por Seguro Social**.
13. Capture el **Título del Concepto**, para los Riesgos del Trabajo, la captura de este campo es opcional.

14. Dentro de la celda de **Mínimo** proporcione los porcentajes para Salarios Mínimos relacionados con cada uno de los conceptos por Seguro Social (Riesgos del Trabajo, Enfermedad y Maternidad e Invalidez y Vejez).
15. Dentro de la celda de **Mayor al Mínimo** proporcione el porcentaje para Salarios Mayores al Mínimo para cada uno de los conceptos por Seguro Social (Riesgos del Trabajo, Enfermedad y Maternidad e Invalidez y Vejez).
16. Si desea agregar un nuevo Concepto al Factor del Seguro Social posiciónese en el renglón correspondiente al concepto Otros 1 que como usted puede observar se encuentra vacío.
17. Repita los pasos del 13 al 15 para cada uno de los conceptos nuevos (Otros 1, Otros 2, Otros 3, Otros 4 y Otros 5).



Usted puede agregar hasta cinco conceptos nuevos, tanto para Salarios Mínimos como para Mayores al Mínimo, para ello, cuenta con los conceptos Otros 1, Otros 2, Otros 3, Otros 4 y Otros 5.

Dentro de los campos **Seguro Social Mínimo** y **Mayor al Mínimo** que se encuentran en la parte izquierda de la ventana del Factor de Salario Real , se presentará el Total del Seguro Social para Salarios Mínimos y para Mayores al Mínimo, los cuales se calculan como:

$$\begin{aligned} \text{SUMA DE LOS CONCEPTOS MÍNIMOS} &= \text{RIESGOS DEL TRABAJO} + \\ &\text{ENFERMEDAD Y MATERNIDAD} + \\ &\text{INVALIDEZ} + \text{OTROS1} + \text{OTROS2} + \\ &\text{OTROS3} + \text{OTROS4} + \text{OTROS5} \end{aligned}$$

$$\text{FACTOR PARA SALARIOS MÍNIMOS} = \frac{(\text{SUMA DE LOS CONCEPTOS MÍNIMOS}/100) * \text{DÍAS PAGADOS}}{\text{DÍAS LABORADOS}}$$

$$\text{SUMA DE CONCEP. MAY. AL MIN} = \text{RIESGOS DEL TRABAJO} + \text{ENFERMEDAD} + \text{MATERNIDAD} + \text{INVALIDEZ} + \text{OTROS1} + \text{OTROS2} + \text{OTROS3} + \text{OTROS4} + \text{OTROS5}$$

$$\text{FACTOR PARA SALARIOS MAYORES AL MÍNIMO} = \frac{(\text{SUMA DE CONCEPTOS MAY. AL MIN.}/100) * \text{DÍAS PAGADOS}}{\text{DÍAS LABORADOS}}$$

Si usted realiza algún cambio en el valor de los conceptos relacionados con el Seguro Social, éste se reflejará inmediatamente dentro de los campos de **Seguro Social Mínimo y Mayor al Mínimo**.

OTROS FACTORES

18. Oprima simultáneamente las teclas ALT+O, para posiciones dentro de las celdas de captura del **Factor de Salario Real por Otros Conceptos**.
19. Capture el **Título del Concepto**, para Guarderías, Infonavit, ISRP y SAR, la captura de este campo es opcional.
20. Dentro de la celda de **Cantidad** proporcione el porcentaje relacionado con los conceptos anteriores.

21. Dentro de la celda **Calculado Por**, proporcione para cada uno de los conceptos antes mencionados, si el cálculo del factor se realizará con los Días por Año con los Días Pagados.
22. Si desea agregar un nuevo concepto al cálculo del Factor de Salario Real existen los conceptos Otros 1, Otros 2 y Otros 3 para los factores que se calcularán con los Días por Año y los conceptos Otros 1, Otros 2 y Otros 3 para los factores que se calcularán con los Días Pagados, que como usted puede observar se encuentran vacíos.
23. Repita los pasos del 19 al 21 para cada uno de los nuevos conceptos (Otros 1 Otros 2, y Otros 3 para Días por Año y Otros 1, Otros 2 y Otros 3 para Días Pagados).



Usted puede agregar hasta tres conceptos nuevos para Días por Año y tres conceptos nuevos para Días Pagados.

Dentro de los campos **Guarderías, INFONAVIT, ISRP, SAR y Otros** que se encuentran en la parte izquierda de la ventana del Factor de Salario Real, se presentará el resultado final para cada uno de estos factores, los cuales se calculan como:

FACTOR DE GUARDERÍAS = $((\% \text{GUARDERÍAS} / 100) * \text{DÍAS POR AÑO}) / \text{DÍAS LABORADOS}$

FACTOR DE INFONAVIT = $((\% \text{INFONAVIT} / 100) * \text{DÍAS POR AÑO}) / \text{DÍAS LABORADOS}$

FACTOR ISRP = $((\% \text{ISRP} / 100) * \text{DÍAS PAGADOS}) / \text{DÍAS LABORADOS}$

$$\text{FACTOR SAR} = ((\% \text{SAR}/100) * \text{DÍAS PAGADOS}) / \text{DÍAS LABORADOS}$$

$$\text{OTROS 1,2 ó 3 POR DÍAS POR AÑO} = \frac{(\% \text{DE OTROS 1, 2 ó 3}/100) * \text{DÍAS POR AÑO}}{\text{DÍAS LABORADOS}}$$


$$\text{OTROS 1,2 ó 3} = \frac{(\% \text{DE OTROS 1, 2 ó 3}/100) * \text{DÍAS PAGADOS}}{\text{DÍAS LABORADOS}}$$

$$\text{OTROS} = \text{OTROS 1} + \text{OTROS 2} + \text{OTROS 3 POR DÍAS POR AÑO} + \text{OTROS 1} + \text{OTROS 2} + \text{OTROS 3 PARA DÍAS PAGADOS}$$

Los cambios que realice dentro de las celdas se reflejarán dentro de los campos que se encuentran en la parte izquierda de la ventana del Factor de Salario Real.

Si usted realiza algún cambio en el valor de los conceptos anteriores, éste se reflejará inmediatamente dentro de los campos de **Guarderías, INFONAVIT, ISRP, SAR y Otros.**

GRABAR LA INFORMACIÓN

- Una vez que los datos proporcionados sean correctos, oprima el botón  de la barra de herramientas del SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar.

Para:

Obtener información acerca de la edición del Factor de Salario Real

Ver:

El tema de "**Factor de Salario Real**" dentro de la Ayuda en Línea del sistema de precios unitarios

SECCIÓN 6 SUMOS



Este módulo le permite crear su catálogo de Insumos para el Presupuesto de Obra en Gestión, DEL SISTEMA DE PRECIOS UNITARIOS para Windows maneja cuatro tipos diferentes de Insumos: Materiales, Obra de mano, Maquinaria y Equipo y Subcontratos.

Dentro de este capítulo veremos cómo editar los diferentes tipos de Insumos, así como algunas de sus principales características.

EDICIÓN EN MODO REGISTRO

En el SISTEMA DE PRECIOS UNITARIOS para Windows la edición de Insumos y análisis se realizará dentro de la Edición en Modo Registro, a continuación se explicarán las principales características de este modo.

Zona de Construcción	Moneda	Precio de Lista	Índice	Módulo	Costo	
Zona de Construcción	Moneda	N\$ 40.91	Índice	0.30	Módulo	N\$ 140.3

CARACTERÍSTICAS

La edición en **Modo Registro**, le ofrece las siguientes características:

- Le permite visualizar un registro a la vez y la información asociada a éste.
- La captura de los Insumos es más ágil y más sencilla.
- Ofrece una funcionalidad semejante a la captura de un registro dentro de una base de datos.
- Los datos comunes en todos los Insumos podrán capturarse mediante campos mientras que los datos particulares de éstos se capturarán mediante un conjunto de celdas.

CÓMO DESPLAZARSE EN MODO DE EDICIÓN REGISTRO

En la Edición en **Modo Registro**, el **SISTEMA DE PRECIOS UNITARIOS** para **Windows**, posicionará el cursor en el primer campo de edición, en espera de que usted introduzca la información.

Usted puede desplazarse a través de los campos en este **Modo** de Edición:

- a. Si se encuentra capturando los Datos Generales del Insumo:

Utilizando las teclas:	Para:
TAB	Desplazarse hacia el campo de edición siguiente.
SHIFT+TAB	Regresar al campo de edición anterior.
ALT+LETRA SUBRAYADA	Ir a un campo en especial. Ejemplo: <u>C</u> lave, teclear ALT+C.

- b. Si se encuentra dentro de las celdas de captura introduciendo la información particular del Insumo:

Utilizando las teclas:	Para:
←	Retroceder una celda
→	Avanzar una celda
↑	Ir al renglón anterior
↓	Ir al renglón siguiente

CÓMO EDITAR LOS DATOS GENERALES DE INSUMOS.

EL SISTEMA DE PRECIOS UNITARIOS para Windows, cuenta con una parte de información general común a todos los Insumos como son su Clave, Descripción, Unidad, Fechas de Costo y Cálculo, Familia (Manejo de agrupaciones), y Costo Unitario. Debido a lo anterior se realizará la explicación de la captura de esta Información General sin enfocarse a un Insumo en particular.

Para editar los datos generales del Insumo realice los siguientes pasos:

1. En el campo de edición de **Clave**, proporcione la clave del Insumo, esta clave puede ser de hasta 25 caracteres y puede contener caracteres alfanuméricos.

Al proporcionar la Clave del Insumo, EL SISTEMA DE PRECIOS UNITARIOS para Windows automáticamente le adicionará a ésta, la letra correspondiente al Tipo de Insumo:

<u>Letra</u>	<u>Tipo Análisis</u>
M	Materiales
O	Obra de mano
Q	Maquinaria y Equipo
T	Subcontratos

El Tipo del Insumo le servirá a el SISTEMA DE PRECIOS UNITARIOS para identificar si el Insumo es un Material, Obra de mano, Maquinaria y Equipo o Subcontrato, esto debido a que, en esta versión, es posible capturar Claves numéricas.


2. Dentro del campo de edición de **Descripción**, proporcione la descripción del Insumo, ésta puede ser tan grande como usted lo desee y puede contener dígitos, letras, espacios y cualquier signo del teclado estándar.
3. En el campo de **Familia**, proporcione la Familia a la que pertenece el Insumo, la captura de este dato es opcional y puede ser de hasta 11 caracteres alfanuméricos. Si la Familia que usted capturó no se encuentra dada de alta, con sólo teclearla dentro de este campo se agregará al Catálogo de Familias.
4. Dentro del campo de **Unidad**, proporcione la unidad de medida del Insumo, el nombre de la unidad de medida puede ser de hasta 11 caracteres alfanuméricos.

En el campo de **Fecha de Costo**, capture la fecha en la cual se le asigna el Costo al Insumo.

Dentro del campo de **Fecha de Cálculo**, capture la fecha en la cual se actualiza el Costo del Insumo.

Para introducir las Fechas de Costo y Cálculo, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas:

Oprima la Tecla	Si desea
F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario DEL SISTEMA DE PRECIOS UNITARIOS
F9	Limpiar el campo de fecha

Si desea grabar los datos generales capturados, oprima el botón  dentro de la Barra de Herramientas del SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar, o si desea continuar con la captura de los Datos Particulares del Insumo, presione la tecla TAB.

CÓMO EDITAR LOS DATOS PARTICULARES DE MATERIALES

Dentro del módulo de Materiales, DEL SISTEMA DE PRECIOS UNITARIOS para Windows genera y mantiene actualizados los datos de Materiales para la Obra.

Dentro de los datos particulares de Materiales, usted puede asignar diferentes precios a un Material, estos precios pueden estar asociados o no a una Zona Económica, también cuenta con campos para la captura de información del Precio de

Lista el cual puede ser capturado con diferentes Tipos de Moneda, Porcentaje de Flete aplicado y Porcentaje de Descuento aplicado.

Para editar estos datos:

1. Dentro de la celda correspondiente a la **Zona Económica**, proporcione la Zona a la cual pertenece el Precio de Lista capturado, la captura de este campo es opcional.
2. Dentro de la celda de **Moneda** usted debe proporcionar la Moneda (símbolo con la cual está calculado el precio del Material, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Para seleccionar el símbolo de la Moneda en los Datos Particulares del Insumo, éste deberá estar capturada dentro del Catálogo de Monedas.

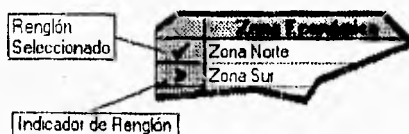
Si esta moneda no se encuentra capturada o desea agregar una nueva, usted puede invocar al Catálogo de Monedas desde este módulo de la siguiente manera:



- a. Oprima el botón derecho del mouse, con ello se le presentará el menú flotante de monedas.
- b. Seleccione la opción de **Catálogo**.

- c. Una vez que la Moneda ha sido capturada y seleccionada la Moneda Base dentro del Catálogo de Monedas, oprima el botón de **Aceptar** que se encuentra dentro de este diálogo.
 - d. Ya que se encuentre nuevamente dentro de la ventana de captura de Insumos, en la celda de **Moneda**, oprima nuevamente el botón derecho del mouse y seleccione del menú la Moneda deseada.
3. Proporcione el Precio de Lista del Material dentro de la celda de **Precio de Lista**, el precio no podrá ser mayor a 16 caracteres y los decimales con los que se presenta puede definirlos dentro del diálogo de Configuración del SISTEMA DE PRECIOS UNITARIOS
 4. Dentro de la celda de **%Flete** capture el Porcentaje de Flete del Material, la captura de este dato es opcional y no podrá ser mayor a 7 caracteres. El Porcentaje de Flete proporcionado se aplica sobre el Precio Unitario del Material.
 5. Dentro de la celda de **%Desct** proporcione el Porcentaje de Descuento del Material, la captura de este dato es opcional y no podrá ser mayor a 7 caracteres. El Porcentaje de Descuento proporcionado se aplica sobre el Precio Unitario del Material.
 6. Si desea capturar un nuevo Precio de Lista, repita los pasos del 1 al 5.
 7. Una vez capturados los Precios, dentro de la columna del indicador de renglón elija el Precio de Lista del Material con el que se realizará el cálculo del Costo Unitario, es decir, el Precio de Lista Activo.

Para seleccionar el renglón que contiene el Precio de Lista que se tomará en cuenta para realizar el cálculo del Costo Unitario:



- a. Posiciónese dentro de la celda del Indicador de Renglón que contiene los datos con los que se realizará el cálculo del Costo Unitario.
- b. Oprima dos veces la BARRA ESPACIADORA del teclado.

O posicione el apuntador del mouse sobre la celda del Indicador del Renglón con el cual desea realizar el cálculo del Costo Unitario y oprima el botón izquierdo de éste.

El Costo Unitario del Material es calculado de la siguiente manera:


$$\text{COSTO UNITARIO} = \text{PRECIO DE LISTA} + \text{FLETE} - \text{DESCUENTO}$$

o bien:

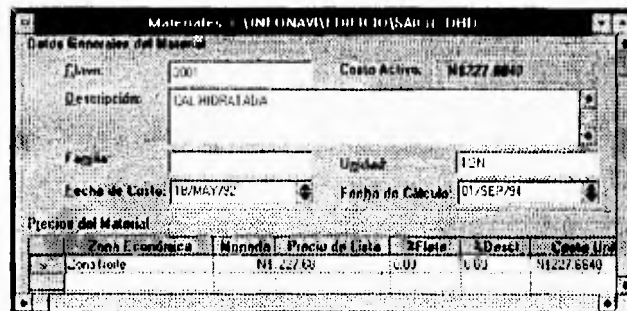
$$\text{COSTO UNITARIO} = \text{PRECIO DE LISTA} + (\% \text{ DE FLETE} * \text{PRECIO} / 100) - (\% \text{ DE DESCUENTO} * \text{PRECIO} / 100)$$

Si usted no indica con qué Precio de Lista se realizará el cálculo del Costo Unitario del Material e intenta grabar los datos, el SISTEMA DE PRECIOS UNITARIOS le preguntará si desea elegir el primer Precio que se encuentra dentro de los Datos

rticulares del Material o desea regresar a la ventana de captura para elegir un precio diferente.

Una vez que proporcionó los datos generales y particulares del Material, oprima el botón  dentro de la Barra de Herramientas DEL SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar.

EMPLO: Para elaborar el Presupuesto de la Obra Edificio es necesario capturar los Materiales que se utilizarán dentro de ella, en el siguiente ejemplo se muestra cómo se realiza la captura de la Cal Hidratada con Clave 0001.



Zona Económica	Moneda	Precio de Lista	Bienes	Desti	Costo Unit
Control	M	27/92	U03	U03	M6227.8640

Para:

Obtener información acerca de la edición de Materiales

Obtener información acerca de las Opciones de Edición

Ver:

El tema de "**Materiales**" dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS

El tema de "**Aprendiendo las Principales Opciones de Edición**" dentro de este Manual.

Obtener información acerca de cómo editar una **Moneda** dentro del Catálogo.

El tema de "**Catálogo de Monedas**", dentro de este Manual o dentro de la Ayuda en Línea de SISTEMA DE PRECIOS UNITARIOS.

Obtener información acerca de cómo editar los datos correspondientes a una **Zona**

El tema de "**Cómo Editar una Zona**", dentro de este Manual, o dentro de la Ayuda en Línea de SISTEMA DE PRECIOS UNITARIOS.

Obtener información acerca de cómo editar los datos correspondientes a una **Familia**

El tema de "**Cómo Editar una Familia**", dentro de este Manual, o dentro de la Ayuda en Línea de SISTEMA DE PRECIOS UNITARIOS.

CÓMO EDITAR LOS DATOS PARTICULARES DE UNA OBRA DE MANO

Dentro del módulo de Obra de mano, DEL SISTEMA DE PRECIOS UNITARIOS para Windows genera y mantiene actualizado el catálogo de la Obra de mano.

Dentro de los datos particulares de la Obra de mano, usted puede asignar diferentes salarios a una Obra de mano, estos salarios pueden estar asociados o no a una Zona Económica, también se cuenta con campos para la captura del Salario Base el cual puede ser capturado con diferentes Tipos de Moneda así como el resultado del Cálculo del Factor de Salario Real (Mínimo, Mayor al Mínimo y definición propia).

Para editar estos datos:

1. Dentro de la celda correspondiente a la **Zona Económica**, proporcione la Zona a la cual pertenecerá el Salario Base, la captura de este campo es opcional.

2. En la celda de **Moneda**, capture el símbolo de la Moneda bajo la cual está el Salario Base de la Obra de mano, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Para seleccionar el símbolo de la Moneda en los Datos Particulares del Insumo, ésta debe estar capturada dentro del Catálogo de Monedas.

3. Proporcione el Salario Base de la Obra de mano dentro de la celda de **Salario Base**, éste no podrá ser mayor a 16 caracteres y los decimales con los que se presenta, puede definirlos dentro del diálogo de Configuración del SISTEMA DE PRECIOS UNITARIOS.
4. Dentro de la celda del Tipo de Factor, seleccione el Tipo de Factor a aplicar.

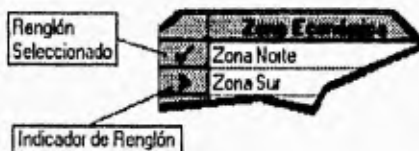
Usted puede seleccionar el Tipo de Factor si se encuentra dentro de esta celda, con sólo oprimir la tecla que corresponde a la letra que identifica al Tipo de Factor.

<u>Letra</u>	<u>Tipo de factor</u>
M	Factor Mínimo
A	Factor Mayor al Mínimo
P	Factor Propio

5. En la celda de **Factor a Aplicar**, proporcione el valor de dicho factor si eligió capturar un Factor Propio, si eligió el Factor Mínimo o Mayor al Mínimo éste aparecerá automáticamente dentro de la celda, de acuerdo a lo calculado en el módulo del Factor de Salario Real.
6. Si desea capturar un nuevo Salario Base, repita los pasos del 1 al 5.

7. Una vez capturados los Salarios Base relacionados con la Obra de mano, dentro de la columna del indicador de renglón, elija el Salario Base de la Obra de mano con el que se realizará el cálculo del Costo Unitario, es decir, el Salario Base Activo.

Para seleccionar el renglón que contiene el Salario Base que se tomará en cuenta para realizar el cálculo del Costo Unitario:




- a. Posiciónese dentro de la celda del Indicador de Renglón que contiene los datos con los que se realizará el cálculo del Costo Unitario.
- b. Oprima dos veces la BARRA ESPACIADORA del teclado.
O posicione el apuntador del mouse sobre la celda del Indicador del Renglón con el cuál desea realizar el cálculo del Costo Unitario y oprima el botón izquierdo de éste.

El Costo Unitario de la Obra de mano es calculado de la siguiente manera:

$$\text{COSTO UNITARIO} = \text{SALARIO BASE} * \text{FACTOR DE SALARIO REAL} \\ \text{SELECCIONADO}$$

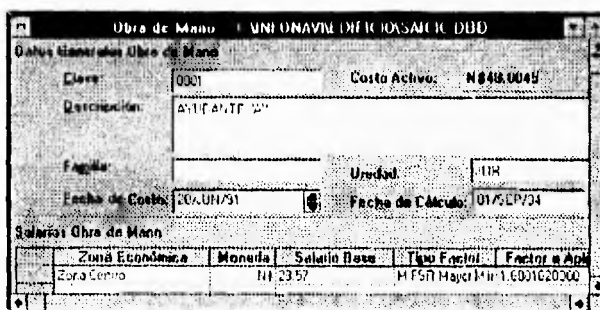
Si usted no indica con qué Salario Base se realizará el cálculo del Costo Unitario de la Obra de mano e intenta grabar los datos, el SISTEMA DE PRECIOS UNITARIOS le preguntará si desea elegir el primer Salario Base que se encuentra dentro de los

datos particulares de la Obra de mano o desea regresar a la ventana de captura para elegir un salario diferente.

8. Una vez que proporcionó los datos generales y particulares de la Obra de mano, oprima el botón  de la Barra de Herramientas DEL SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar.

EJEMPLO:

Para elaborar el Presupuesto de la Obra Edificio, es necesario capturar la Obra de mano que se necesitará dentro de ella, en el siguiente ejemplo se muestra cómo se realiza la captura de un Ayudante Tipo "A" con Clave 0001, con un Salario Base de N\$28.57.



Zona Económica	Moneda	Salario Base	Tipo Frecu.	Factor a Apl.
Zona Centro	N	28.57	MFSN Mayor	1.6001620000

Para:

Obtener más información acerca de la edición de la Obra de mano

Obtener información acerca de las Opciones de Edición

Ver:

El tema de "**Obra de mano**" dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS

El tema de "**Aprendiendo las Principales Opciones de Edición**" dentro de este Manual".

Obtener información acerca de cómo editar una Moneda dentro del Catálogo.

El tema de "**Catálogo de Monedas**", dentro de este Manual, o dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS.

Obtener información acerca de cómo editar los datos correspondientes a una Zona

El tema de "**Cómo Editar una Zona**", dentro de este Manual, o dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS.

Obtener información acerca de cómo editar los datos correspondientes a una Familia

El tema de "**Cómo Editar una Familia**", dentro de este Manual, o dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS.

CÓMO EDITAR LOS DATOS PARTICULARES DE LA MAQUINARIA Y EQUIPO Y EL COSTO HORARIO

Dentro del módulo de Maquinaria y Equipo, DEL SISTEMA DE PRECIOS UNITARIOS para Windows genera y mantiene actualizado el catálogo de Maquinaria para la Obra.

Para editar los datos particulares de la Maquinaria:

1. Dentro de la celda correspondiente a la **Zona Económica**, proporcione la Zona a la cual pertenecerá el Costo de la Maquinaria, la captura de este campo es opcional.
2. Dentro de la celda de **Moneda**, capture el símbolo de la Moneda bajo la cual está el Costo Unitario de la Maquinaria, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.


Para seleccionar el símbolo de la Moneda en los Datos Particulares del Insumo, ésta debe estar capturada dentro del Catálogo de Monedas.

- 3 Si usted no desea realizar el análisis del Costo Horario de la Maquinaria, dentro de la celda de **Costo Rápido**, proporcione el Costo Unitario de la Maquinaria, esto puede ser de utilidad si usted renta el equipo.

Si usted decide realizar el análisis del Costo Horario de la Maquinaria, posicione el apuntador del mouse sobre la celda **Costo Analizado** y oprima dos veces el botón izquierdo de éste, con ello aparecerá el diálogo de Cálculo Costo Horario, en el cual usted debe capturar información que servirá para realizar el cálculo del Costo Horario de la Maquinaria y Equipo.

- 4 Una vez capturados los Costos Horarios relacionados con la Maquinaria y Equipo, dentro de la columna del indicador de renglón, elija el Costo Horario de la Maquinaria con el que se realizará el cálculo del Costo Unitario, es decir, el Costo Horario de la Maquinaria Activo.

Si usted no indica con qué Costo Horario se realizará el cálculo del Costo Unitario de la Maquinaria y Equipo e intenta grabar los datos, DEL SISTEMA DE PRECIOS UNITARIOS le preguntará si desea elegir el primer Costo Horario Base que se encuentra dentro de los datos particulares de la Maquinaria o desea regresar a la ventana de captura para elegir un Costo diferente

- 5 Una vez que proporcionó los datos generales y particulares de la Maquinaria y Equipo, oprima el botón  de la Barra de Herramientas DEL SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar.

CÁLCULO DEL COSTO HORARIO DE LA MAQUINARIA Y EQUIPO

El diálogo del Cálculo Costo Horario está formado por varios separadores: **Gastos Equipo, Factores, Motor y Gastos Operación**, estos separadores le permiten editar los datos necesarios para este cálculo.

A continuación se describirá el procedimiento que se debe seguir para capturar información en cada uno de estos separadores.

SEPARADOR GASTOS EQUIPO

En este separador usted puede capturar los costos de la Máquina Base y del Equipo Adicional, el precio de las llantas y el valor de rescate de la Maquinaria y Equipo, para ello, es necesario realizar lo siguiente:

Dentro de la tabla **Costo de la Maquinaria y Equipo Adicional**:

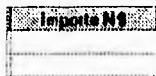
1. En la celda de **Moneda**, capture el símbolo de la Moneda bajo la cual está el Costo de la Máquina Base, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Para seleccionar el símbolo de la Moneda, ésta debe estar capturada dentro del Catálogo de Monedas.

2. Dentro de la celda de **Costo**, proporcione el Costo de Adquisición de la Máquina Base.

En la celda de **Importe**, se realizará la conversión de la moneda del Costo a la moneda que usted seleccionó como base en el Catálogo de Monedas. La moneda

que seleccionó como base aparecerá en el encabezado de la celda después de la palabra **Importe** como se muestra a continuación:



3. Edite la información anterior para el Equipo, siguiendo los pasos anteriores.

Dentro de la tabla **Precio de las Llantas Máquina Base y Equipo Adicional**:

4. En la celda de **Clave**, proporcione la clave de las Llantas de la Máquina Base. Al proporcionar la clave de las llantas, esta clave y sus datos relacionados pueden aparecer de diferente color, siguiendo las siguientes reglas:

Color del Elemento (por Default):

Si:

Negro

Ya se encuentra capturado dentro de los Insumos.

Rojo

Los datos del Insumo no han sido capturados.



El color puede variar si usted cambia las especificaciones en el módulo de Configuración del SISTEMA DE PRECIO UNITARIOS.



El tipo de la Clave no se muestra debido a que las Llantas de la Máquina Base y del Equipo Adicional siempre se consideran como Materiales.

Si el elemento existe:

Automáticamente se llenarán los campos de **Descripción**, **Moneda** y **Costo Unitario**.



Como las llantas serán editadas como un Insumo, el Costo Unitario de éstas, será calculado en base al Precio Activo de éste.



El Costo Unitario de las llantas se presentará en base a la Moneda con la que éste fue calculado.

Si el elemento no existe:

- a. Dentro de la celda de **Descripción**, capture la descripción del tipo de llanta utilizado.
 - b. En la celda de **Moneda**, proporcione la Moneda bajo la cual se encuentra el Costo Unitario, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.
 - c. En la celda de **Costo**, proporcione el costo de las llantas.
5. Dentro de la celda de **No.**, proporcione el número de llantas, éste no puede ser mayor a 10 caracteres.

Una vez que usted capturó el número de llantas, dentro de la celda de **Importe**, aparecerá el importe de las llantas, que es calculado como:

$$\text{COSTO} * \text{NÚMERO}$$

En el importe de las llantas, se realizará la conversión de la moneda del Costo a la moneda que usted seleccionó como base en el Catálogo de Monedas. La moneda que seleccionó como base aparecerá en el encabezado de la celda después de la palabra Importe.



Las Llantas de la Máquina Base y del Equipo Adicional serán capturada dentro del SISTEMA DE PRECIOS UNITARIOS en el módulo d Materiales

6. Repita los pasos 4 y 5, pero ahora capturando la información correspondiente a las llantas del Equipo Adicional.

Una vez que usted realizó la captura de estos datos, dentro del campo de **Valor Total de las Llantas (VII)**, se presentará el valor de adquisición de las llantas, calculado como:

$$\begin{array}{l} \text{IMPORTE DE LAS LLANTAS} \\ \text{DE MAQUINA BASE} \end{array} + \begin{array}{l} \text{IMPORTE DE LAS LLANTAS DE} \\ \text{EQUIPO ADICIONAL} \end{array}$$

A su vez en el campo de **Valor de Adquisición (Va)**, se calculará el valor inicial de la Maquinaria y Equipo de la siguiente manera:

$$\text{COSTO MÁQUINA BASE} + \text{COSTO DE EQUIPO ADICIONAL} - \text{VALOR TOTAL DE LAS LLANTAS}$$

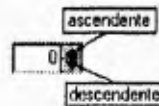
En el campo de **Valor de Rescate%** proporcione el porcentaje a aplicar al valor de adquisición de la Maquinaria y Equipo. Una vez que usted proporciona este porcentaje en el campo **Vr** se presentará el resultado de aplicar este porcentaje, el cual se calcula como:

$$\text{VALOR DE ADQUISICIÓN} * \text{PORCENTAJE} / 100$$

SEPARADOR FACTORES

En este separador usted puede capturar la Vida Económica en horas, la Vida de las Llantas, las Horas Anuales, la Prima de Seguro Anual, los Factores de Almacenaje, Mantenimiento y Operación, la Potencia Nominal y la Tasa de Interés Anual, para ello, es necesario realizar lo siguiente:

1. Dentro del campo de **Vida Económica en Horas (Ve)** proporcione la vida económica en horas de trabajo de la Maquinaria y Equipo.



Usted puede incrementar o decrementar las horas con la ayuda de la flecha ascendente o descendente respectivamente, para ello:

- a. Posicione el apuntador del mouse sobre la flecha ascendente o descendente.
- b. Oprima el botón izquierdo del mouse.

O si lo desea puede incrementar o decrementar las horas con sólo teclear las flechas verticales del teclado (- $\bar{\bar{v}}$).

2. Dentro del campo de **Vida de las Llantas en Horas (HVLL)**, capture la vida económica de las llantas de la Maquinaria y Equipo, expresada en horas de trabajo, para ello, puede seguir el procedimiento descrito en el punto anterior.
3. En el campo de **Horas Anuales (Ha)**, capture el número de horas efectivas que el equipo trabaja durante el año, para ello puede seguir el procedimiento descrito en el punto 1.

Dentro del campo de **Prima de Seguro Anual (s)**, capture la prima anual de la Maquinaria y Equipo, expresada en porcentaje.

En el campo de **Factor de Almacenaje (Fa)**, capture el factor de almacenaje para la Maquinaria y Equipo.



Recuerde que si está trabajando para Obra Pública este porcentaje *n* se debe aplicar.

Dentro del campo de **Factor de Mantenimiento (Q)**, capture el coeficiente que incluye tanto el mantenimiento mayor como el menor de la maquinaria.

En el campo de **Factor de Operación**, capture el factor de operación de la Maquinaria y Equipo.

Dentro del campo de **Potencia Nominal**, capture la potencia nominal del motor de la maquinaria.

1. En el campo de **Tasa de Interés Anual (i)**, capture la tasa de interés anual, expresada como un porcentaje.

SEPARADOR MOTOR

En este separador usted puede editar la información relacionada con el Tipo de Motor de la Maquinaria, para ello, es necesario realizar lo siguiente:

1. Seleccione el radiobutton asociado al Tipo de Motor, éste puede ser **Diesel**, **Gasolina**, o **Eléctrico**

Usted puede observar que dentro de la tabla **Información sobre el Tipo de Motor** va a cambiar la información que puede editar, dependiendo del Tipo de Motor que usted seleccione.

Para el tipo de Motor Diesel:

- a. En la celda de **Clave**, proporcione la clave del combustible. Al proporcionar esta clave, ésta y sus datos relacionados pueden aparecer de diferente color siguiendo las siguientes reglas:

**Color del Elemento
(por Default):**

Si:

Negro

Ya se encuentra capturado dentro de los Insumos.

Rojo

Los datos del Insumo no han sido capturados.

@

El color puede variar si usted cambia las especificaciones en el módulo de Configuración del SISTEMA DE PRECIOS UNITARIOS.

@

El tipo de la Clave no se muestra debido a que el Combustible Lubricante, Lubricante Hidráulico, Gasolina de Arranque, y Otra Fuentes de Energía son considerados siempre como Materiales.

Si el elemento existe:

Automáticamente se llenarán los campos de **Descripción, Moneda y Costo Unitario**.

@

Como el combustible será editado como un Insumo, el Costo Unitario de éste, será calculado en base al Precio Activo de éste.

@

El Costo Unitario del combustible se presentará en base a la Moneda con la que éste fue calculado.

Si el elemento no existe:

- i. Dentro de la celda de **Descripción**, capture la descripción del combustible utilizado.
- ii. En la celda de **Moneda**, proporcione la Moneda bajo la cual se encuentra el Costo Unitario, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Si esta moneda no se encuentra capturada es necesario agregarla al Catálogo de Monedas.

- iii. En la celda de **Costo**, proporcione el Costo del combustible utilizado.

En la celda de **Importe**, se realizará la conversión de la moneda del Costo a la moneda que usted seleccionó como base en el Catálogo de Monedas. La moneda que seleccionó como base aparecerá en el encabezado de la celda después de la palabra Importe.

- b. En la celda de **Factor**, capture la cantidad de combustible por hora efectiva de trabajo.
- c. Repita los pasos a y b para el **Lubricante, Lubricante Hidráulico, Gasolina de Arranque, y Otras Fuentes Energía.**



Si el Combustible, Lubricante, Lubricante Hidráulico, Gasolina d Arranque, y Otras Fuentes de Energía no existen, al ser capturados e el Costo Horario se almacenarán automáticamente en el Catálogo d Materiales.

Para el tipo de Motor de Gasolina

- a. En la celda de **Clave**, proporcione la clave del combustible. Al proporcionar esta clave, ésta y sus datos relacionados pueden aparecer de diferente color, siguiendo las siguientes reglas:

**Color del Elemento
(Por default):**

Si:

Negro

Ya se encuentra capturado dentro de los Insumos.

Rojo

Los datos del Insumo no han sido capturados.

@

El color puede variar si usted cambia las especificaciones en el módulo de Configuración del SISTEMA DE PRECIO UNITARIOS.

@

El tipo de la Clave no se muestra debido a que el Combustible, Lubricante, Lubricante Hidráulico, y Otras Fuentes de Energía son considerados siempre como Materiales.

Si el elemento existe:

Automáticamente se llenarán los campos de **Descripción, Moneda y Costo Unitario.**

@

Como el combustible será editado como un Insumo, el Costo Unitario de éste, será calculado en base al Precio Activo de éste.

@

El Costo Unitario del combustible se presentará en base a la Moneda con la que éste fue calculado.

Si el elemento no existe:

- i. Dentro de la celda de **Descripción**, capture la descripción del combustible utilizado.

- i. En la celda de **Moneda**, proporcione la Moneda bajo la cual se encuentra el Costo Unitario, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Si esta moneda no se encuentra capturada es necesario agregarla al Catálogo de Monedas.

- ii. En la celda de **Costo**, proporcione el costo del combustible utilizado.

En la celda de **Importe**, se realizará la conversión de la moneda del Costo a la moneda que usted seleccionó como base en el Catálogo de Monedas. La moneda que seleccionó como base aparecerá en el encabezado de la celda después de la palabra Importe.

- b. En la celda de **Factor**, capture la cantidad de combustible por hora efectiva de trabajo.
- c. Repita los pasos a y b para el **Lubricante, Lubricante Hidráulico, y Otras Fuentes Energía**.



Si el Combustible, el Lubricante, el Lubricante Hidráulico y Otra Fuentes de Energía no existen, al ser capturados en el Costo Horari se almacenarán automáticamente en el Catálogo de Materiales.

Para el tipo de Motor Eléctrico:

- a. En la celda de **Clave**, proporcione la clave del kilowatt/hora. Al proporcionar esta clave, ésta y sus datos relacionados pueden aparecer de diferente color, siguiendo las siguientes reglas

Color del Elemento (Por default):

Si:

Negro

Ya se encuentra capturado dentro de los Insumos.

Rojo

Los datos del Insumo no han sido capturados.

@

El color puede variar si usted cambia las especificaciones en el módulo de Configuración del SISTEMA DE PRECIOS UNITARIOS.

@

El tipo de la Clave no se muestra debido a que el Kilowatt/Hora es considerado siempre como un Material.

Si el elemento existe:

Automáticamente se llenarán los campos de **Descripción**, **Moneda** y **Costo Unitario**.

@

Como el kilowatt/hora será editado como un Insumo, el Costo Unitario de éste, será calculado en base al Precio Activo de éste.

@

El Costo Unitario del kilowatt/hora se presentará en base a la Moneda con la que éste fue calculado.

Si el elemento no existe:

- i. Dentro de la celda de **Descripción**, capture la descripción del kilowatt/hora utilizado.
- ii. En la celda de **Moneda**, proporcione la Moneda bajo la cual se encuentra el Costo Unitario, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Si esta moneda no se encuentra capturada es necesario agregarla al Catálogo de Monedas.

iii. En la celda de **Costo**, proporcione el costo del kilowatt/hora utilizado.

En la celda de **Importe**, se realizará la conversión de la moneda del Costo a la moneda que usted seleccionó como base en el Catálogo de Monedas. La moneda que seleccionó como base aparecerá en el encabezado de la celda después de la palabra Importe.

b. En la celda de **Factor**, capture la cantidad necesaria por hora efectiva de trabajo.



Si Kilowatt/Hora no existe, al ser capturado en el Costo Horario s almacenará automáticamente en el Catálogo de Materiales.

2. Dentro del campo **Capacidad del Carter**, proporcione la capacidad del carter en litros.

3. En el campo **Tiempo de Cambio**, capture el tiempo en horas transcurrido entre dos cambios de aceite.

SEPARADOR GASTOS OPERACIÓN

En este separador usted puede editar la información necesaria para calcular el importe de los cargos por operación de la Maquinaria, para ello, es necesario realizar lo siguiente:

1. En el campo **Factor de Operación Obra de mano**, capture el factor de operación de la Obra de mano por hora efectiva de trabajo.
2. Dentro del campo **Horas Diarias**, proporcione las horas diarias de trabajo de la Obra de mano.

Dentro de la tabla **Obra de mano Utilizada**:

3. En la celda de **Clave**, proporcione la clave de la Obra de mano utilizada. Al proporcionar esta clave, ésta y sus datos relacionados pueden aparecer de diferente color, siguiendo las siguientes reglas:

Color del Elemento (Por default):	Si:
Negro	Ya se encuentra capturada dentro de los Insumos.
Rojo	Los datos del Insumo no han sido capturados.



El color puede variar si usted cambia las especificaciones en el módulo de Configuración del SISTEMA DE PRECIO UNITARIOS.



El tipo de la Clave no se muestra debido a que los Operadores de la Maquinaria serán considerados siempre como una Obra de mano.

Si la Obra de mano existe:

Automáticamente se llenarán los campos de **Categoría, Moneda y Salario.**



El Costo Unitario de la Obra de mano, será calculado en base al Salario Base Activo.



El Costo Unitario de la Obra de mano se presentará en base a la Moneda con la que éste fue calculado.

Si la Obra de mano no existe:

- a. Dentro de la celda de **Categoría**, capture la descripción o categoría de la Obra de mano.
- b. En la celda de **Moneda**, proporcione la Moneda bajo la cual se encuentra el Salario de la Obra de mano, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Si esta moneda no se encuentra capturada es necesario agregarla al Catálogo de Monedas

- c. En la celda de **Salario**, proporcione el salario de la Obra de mano.



Si los Operadores de la Maquinaria no existen, al ser capturados en el Costo Horario se almacenarán automáticamente en el Catálogo de Obra de mano.

4. En la celda de **Cant.**, capture la cantidad de ese tipo de operador.

Una vez que usted capturó la cantidad de operadores, dentro de la celda de **Importe** aparecerá el importe de los salarios de la Obra de mano, que es calculado como:

$$\text{SALARIO} * \text{CANTIDAD}$$

En el importe de la Obra de mano, se realizará la conversión de la moneda del costo a la moneda que usted seleccionó como base en el Catálogo de **Monedas**. La moneda que seleccionó como base aparecerá en el encabezado de la celda después de la palabra **Importe**.

En el campo de **Suma de Salarios**, se reflejará la suma de salarios del personal que opera la Maquinaria y Equipo de la siguiente manera:


$$\text{IMPORTE 1} + \text{IMPORTE 2} + \text{IMPORTE 3}$$

En el campo **Operación**, se reflejará la operación aritmética a realizar para el cálculo de los **Cargos por Operación**, si usted modifica algunos de los datos de la Obra de mano para la Maquinaria, estos cambios también se reflejarán en este campo.

Dentro del campo **Cargos por Operación**, se reflejará el cargo por operación de la Maquinaria y Equipo por hora efectiva de trabajo, el cual se calcula como:

$$\text{SUMA DE SALARIOS} / (\text{FACTOR DE OPERACIÓN} * \text{HORAS DIARIAS})$$

Usted puede salir del diálogo del Cálculo Costo Horario, con sólo oprimir el botón de **Salir** que se encuentra dentro de cualquier separador.

Si desea grabar el cálculo del Costo Horario de la Maquinaria que analizó, oprima el botón  de la Barra de Herramientas DEL SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar.

El cálculo final del Costo Unitario de la Maquinaria y Equipo se realiza de la siguiente manera:

$$\text{VALOR DE RESCATE} = \frac{\text{VALOR DE ADQUISICIÓN} * (\% \text{ VALOR DE RESCATE})}{100}$$

$$\text{POTENCIA DE OPERACIÓN} = \text{POTENCIA} * \text{FACTOR DE OPERACIÓN}$$

$$D = \frac{(\text{VALOR DE ADQUISICIÓN} - \text{VALOR DE RESCATE})}{\text{VIDA ECONÓMICA EN HORAS}}$$

$$I = \frac{(\text{VALOR DE ADQUISICIÓN} + \text{VALOR DE RESCATE}) * \text{TASA DE INTERÉS ANUAL}}{(2 * \text{HORAS ANUALES}) * 100}$$

$$S = \frac{(\text{VALOR DE ADQUISICIÓN} + \text{VALOR DE RESCATE}) * \text{PRIMA DE SEGURO ANUAL}}{(2 * \text{HORAS ANUALES}) * 100}$$

$$\text{ALMACENAJE} = D * \text{FACTOR DE ALMACENAJE}$$

$$M = \text{FACTOR DE MANTENIMIENTO} * D$$

Para Motor Diesel:

$$\text{GASOLINA DE ARRANQUE} = \text{FACTOR DE GASOLINA} * \text{POTENCIA DE OPERACIÓN DE ARRANQUE}$$

$$\text{COMBUSTIBLE} = \text{FACTOR} * \text{POTENCIA} * \text{COSTO}$$

Manual de Usuario

DE COMBUSTIBLE DE OPERACIÓN COMBUSTIBLE

LUBRICANTE = ((CAPACIDAD CARTER / TIEMPO CAMBIO) + FACTOR
LUBRICANTE)*POTENCIA OPERACION*COSTO LUBRIC.

LUBRICANTE = FACTOR LUBRICANTE * COSTO LUBRICANTE
HIDRÁULICO HIDRÁULICO HIDRÁULICO

CONSUMO LLANTAS = VALOR TOTAL / VIDA DE LAS LLANTAS
DE LAS LLANTAS EN HORAS

CARGOS POR CONSUMO = COMBUSTIBLE+LUBRICANTE+LUBRICANTE
HIDRÁULICO + CONSUMO DE LLANTAS + GASOLINA
ARRANQUE + OTRAS FUENTES ENERGÍA

Para Motor Gasolina:

COMBUSTIBLE = FACTOR * POTENCIA * COSTO
COMBUSTIBLE DE OPERACIÓN COMBUSTIBLE

LUBRICANTE = ((CAPACIDAD CARTER/TIEMPO DE CAMBIO)+FACTOR LUBRICANTE)*POTENCIA OPERACIÓN* COSTO LUBRICANTE

LUBRICANTE = FACTOR LUBRICANTE * COSTO LUBRICANTE
HIDRÁULICO HIDRÁULICO

CONSUMO LLANTAS = VALOR TOTAL DE / VIDA DE LAS LLANTAS
LAS LLANTAS EN HORAS

CARGOS POR CONSUMO = COMBUSTIBLE + LUBRICANTE + LUBRICANTE
HIDRÁULICO + CONSUMO DE LLANTAS + OTRAS
FUENTES ENERGÍA

Para Motor Eléctrico

CARGOS POR CONSUMO = FACTOR MOTOR ELÉCTRICO * POTENCIA
OPERACIÓN * COSTO DEL KILOWATT

COSTO HORA MÁQUINA SIN OPERACIÓN = D + I + S + M + CARGOS POR
CONSUMO

COSTO HORA MÁQUINA = COSTO HORA MÁQUINA SIN OPERACIÓN +
CON OPERACIÓN CARGOS POR OPERACIÓN

Para:

Obtener información acerca de la edición de Maquinaria y Equipo

Obtener información acerca de las Opciones de Edición

Ver:

El tema de "**Maquinaria y Equipo**" dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS

El tema de "**Aprendiendo las Principales Opciones de Edición**" dentro de este Manual.

Obtener información acerca de cómo editar una Moneda dentro del Catálogo.

El tema de "**Catálogo de Monedas**", dentro de este Manual, o dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS.

Obtener información acerca de cómo editar los datos correspondientes a una Zona

El tema de "**Cómo Editar una Zona**", dentro de este Manual, o dentro de la Ayuda en Línea del SISTEMA DE PRECIOS UNITARIOS.

Obtener información acerca de cómo editar los datos correspondientes a una Familia

El tema de "**Cómo Editar una Familia**", dentro de este Manual, o dentro de la Ayuda en Línea del sistema de precios unitarios.

CÓMO EDITAR LOS DATOS PARTICULARES DE LOS SUBCONTRATOS

Dentro del módulo de Subcontratos, DEL SISTEMA DE PRECIOS UNITARIOS para Windows genera y mantiene actualizado el catálogo de Subcontratos para la Obra.

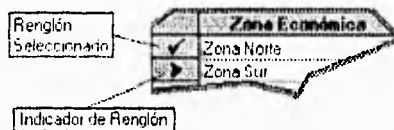
Para editar los datos particulares del Subcontrato:

1. Dentro de la celda correspondiente a la **Zona Económica**, proporcione la Zona a la cual pertenecerá el Costo Unitario capturado, la captura de este campo es opcional.
2. En la celda de **Proveedor**, proporcione el nombre o razón social del Subcontratista, éste no puede ser mayor a 80 caracteres.
3. Dentro de la celda de **Moneda**, capture el símbolo de la Moneda bajo la cual está el Costo Unitario del Subcontrato, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la Moneda deseada.

Para seleccionar el símbolo de la Moneda en los Datos Particulares del Insumo, ésta debe estar capturada dentro del Catálogo de Monedas.

4. En la celda de **Costo Unitario** capture el Costo Unitario del Subcontrato, este costo no podrá ser mayor a 16 caracteres y los decimales con los que se presenta puede definirlos dentro del diálogo de Configuración del sistema de precios unitarios.
5. Si desea capturar un nuevo costo, repita los pasos del 1 al 4.
6. Una vez capturados los costos relacionados con el Subcontrato, dentro de la columna del indicador de renglón, elija el costo del Subcontrato, es decir el costo activo.


Para seleccionar el renglón que contiene los datos que se tomarán en cuenta para realizar el cálculo del Costo Unitario:



- a. Posiciónese dentro de la celda del Indicador de Renglón que contiene los datos con los que se realizará el cálculo del Costo Unitario.
- b. Oprima dos veces la BARRA ESPACIADORA del teclado.

O posicione el apuntador del mouse sobre la celda del Indicador de Renglón con el cual desea realizar el cálculo del Costo Unitario y oprima el botón izquierdo de éste.

Si usted no indica el Costo Unitario del Subcontrato e intenta grabar los datos, DEL SISTEMA DE PRECIOS UNITARIOS le preguntará si desea elegir el primer Costo Unitario que se encuentra dentro de los datos particulares del Subcontrato o desea regresar a la ventana de captura para elegir un costo diferente.

7. Una vez que los datos proporcionados son correctos, oprima el botón  de la Barra de Herramientas DEL SISTEMA DE PRECIOS UNITARIOS correspondiente a grabar.



Para elaborar el Presupuesto de la Obra Edificio es necesario capturar los Subcontratos que se realizarán dentro de ella, en el siguiente ejemplo se muestra la captura del Subcontrato relacionado con la Instalación de Alfombras con Clave 0001.

Zona	Zona Especifica	Proveedor	Cantidad	Costo Unitario
Zona Norte		Alfombras Mexico		NS 10.00

Para:

Obtener información acerca de la edición de Subcontratos

Ver:

El tema de **"Subcontratos"** dentro de la Ayuda en Línea del sistema de precios unitarios

Obtener información acerca de las Opciones de Edición

El tema de "**Aprendiendo las Principales Opciones de Edición**" dentro de este Manual.

Obtener información acerca de cómo editar una Moneda dentro del Catálogo.

El tema de "**Catálogo de Monedas**", dentro de este Manual, o dentro de la Ayuda en Línea del sistema de precios unitarios.

Obtener información acerca de cómo editar los datos correspondientes a una Zona

El tema de "**Cómo Editar una Zona**", dentro de este Manual, o dentro de la Ayuda en Línea del sistema de precios unitarios

Obtener información acerca de cómo editar los datos correspondientes a una Familia

El tema de "**Cómo Editar una Familia**", dentro de este Manual, o dentro de la Ayuda en Línea del sistema de precios unitarios.

INFORMACIÓN ADICIONAL DEL INSUMO

Dentro de este diálogo usted puede capturar información adicional del Insumo, para ello, cuenta con varios separadores que le permiten capturar las Alias y las Leyendas asociadas a éste.






El botón de **Información del Insumo**, se encontrará deshabilitado, hasta que usted proporcione la Clave del Insumo, o edite un Insumo ya existente.

ALIAS

Dentro de este separador usted puede capturar los diferentes Alias asociados con el Insumo. Un Alias es un nombre diferente que se le dará a la Clave del Insumo con Análisis, con el fin de que pueda ser utilizado para diferentes Presupuestos. Ya que la Dependencia Pública, solicita que la Obra que usted va a presupuestar, lleve Claves propias de esta dependencia.

Usted puede capturar tantos Alias como desee para el Insumo, pero dentro de sus reportes sólo se reflejará el Alias que se encuentre activo.

A continuación describiremos el procedimiento utilizado para editar los Alias:

1. Oprima el botón  dentro de la Barra de Captura.

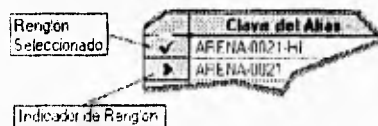
O desde el menú de **Edición** seleccione la opción de **Información**.

U oprima simultáneamente las teclas CTRL+I.

2. Una vez dentro del diálogo de Información de Insumos, seleccione el separador **Alias**.
3. En la celda de **Clave del Alias**, proporcione el Alias del Insumo, éste puede ser de hasta 25 caracteres alfanuméricos.

4. Si desea capturar un nuevo Alias relacionado al Insumo, posicione en el primer renglón que se encuentre vacío y repita el paso anterior.
5. Una vez capturados los Alias relacionados con el Insumo, dentro de la columna del indicador de renglón, elija el Alias con el cual se representará el Insumo dentro de sus reportes, en caso de que usted los solicite con Alias.

Para seleccionar el Alias:



- a. Posiciónese dentro de la celda del Indicador de Renglón que contiene el Alias deseado.
- b. Oprima dos veces la BARRA ESPACIADORA del teclado.

O posicione el apuntador del mouse sobre la celda del Indicador de Renglón que contiene el Alias deseado y oprima el botón izquierdo de éste.

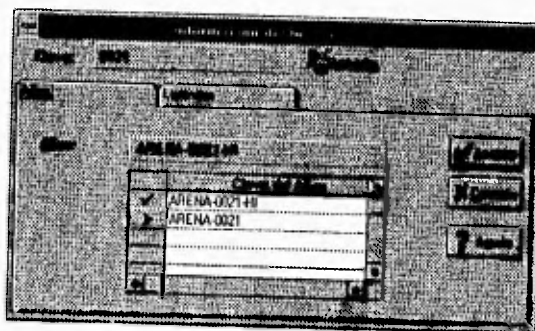


En el campo de **Alias** se mostrará el Alias del Insumo o Análisis que se presentará dentro de los reportes. Cada vez que usted seleccione un Alias diferente, el cambio también se reflejará dentro de este campo.

6. Oprima el botón de **Aceptar** si está de acuerdo con la información proporcionada, de lo contrario oprima el botón de **Cancelar**.

EJEMPLO


El Material con clave 0021, tiene asignados varios Alias, pero el Alias que se encuentra activo y que se presentará dentro de los reportes si así se indica, para este ejemplo es el Alias ARENA-0021-HI.



LEYENDAS

Dentro de este separador podrá capturar la Leyenda correspondiente al Insumo que se encuentre editando. Las Leyendas son textos que usted puede asociar a los diferentes Insumos o Análisis, como una segunda Descripción y poderla utilizar para presupuestar una Obra relacionada con otra dependencia.

Para editar la Leyenda es necesario realizar lo siguiente:

1. Oprima el botón  dentro de la Barra de Captura.

O desde el menú de **Edición** seleccione la opción de **Información**.

U oprima simultáneamente las teclas CTRL+I.

2. Una vez dentro del diálogo de Información del Insumo, seleccione el separador **Leyendas**.

En el campo de **Leyenda**, introduzca el texto correspondiente a la Leyenda del Insumo, la cual puede ser tan grande como usted lo desee.

Oprima el botón de **Aceptar** si está de acuerdo con la información proporcionada, de lo contrario oprima el botón de **Cancelar**.



Sólo se puede manejar una sólo Leyenda para cada Insumo o Análisis, ésta será la Leyenda utilizada en caso de solicitar los reportes con Leyendas.

SECCIÓN 7 ANÁLISIS



DIFERENTES NIVELES DE ANÁLISIS EN EL SISTEMA DE PRECIOS UNITARIOS

Este módulo del Sistema de Precios Unitarios, le permite manejar cinco tipos de Análisis o Matrices: **Básicos, Integrados, Partidas, Capítulos y Extraordinarios.** El Sistema de Precios Unitarios para Windows le presentará el cálculo del importe de cada uno de los elementos que componen el Análisis, así como el costo total de mismo. Por medio de estos niveles, usted puede elaborar sus Precios Unitarios y su Presupuesto, así como aquellos conceptos fuera del Presupuesto. Además, dentro de los Análisis usted puede agregar porcentajes para sus elementos a través de la Herramientas.

Dentro de este tema se hablará en forma general de cómo manejar los cinco tipos de Análisis, debido a que la metodología de su funcionamiento es la misma, lo único que cambia es el cómo están considerados cada uno de esos cinco tipos.

El Sistema de Precios Unitarios para Windows, le permite hacer la agrupación que desee y formar cada uno de los niveles de Análisis a su elección, no existe restricción en cuanto al tipo de elementos o componentes que pueden estar dentro de un Análisis así como el número de éstos. Un Análisis puede contener otro de su mismo nivel, por ejemplo, un Básico puede contener un Insumo u otro Básico. Sin embargo se le recomienda no hacer agrupamientos cíclicos, es decir, que el Básico 1 contenga el Básico 2 y el Básico 2 a su vez al Básico 1.

Para realizar el agrupamiento de los Análisis se le recomienda la siguiente manera:

- a) Básicos: Formados de Básicos e Insumos

Integrados o Precios Unitarios: Formados de Integrados, Básicos e Insumos

b) Partidas: Formadas por Integrados y Básicos
Capítulos: Formados de Partidas e Integrados

En estos cuatro niveles es donde se encuentra el Presupuesto de la Obra o Catálogo Base. Aquí se manejan los volúmenes de Obra.

c) Extraordinarios: Formados de Capítulos y Partidas.






En el nivel anterior se pueden analizar todos los Básicos, Integrados, Partidas y/o Capítulos que están fuera del Presupuesto.

La ventana de Análisis se encuentra dividida en dos partes, en las cuales usted puede visualizar diferentes niveles de Análisis, teniendo la ventaja de poder analizar los componentes de éste en una sola ventana, el procedimiento para realizar esta división se describirá mas adelante.

CÓMO EDITAR LOS DATOS GENERALES DEL ANÁLISIS

La edición de Análisis, al igual que la de Insumos contiene una sección de datos generales para todos los niveles existentes, por lo que a continuación se explicará la captura de estos datos sin enfocarse a un nivel en particular.

1. Desde la Barra de Herramientas del Sistema de Precios Unitarios, oprima el botón correspondiente al nivel de Análisis que desea realizar.

Utilizando el botón:	Para el nivel de:	Ejemplo:
	Básicos	Morteros, concretos, cuadrillas, andamios, etc.
	Integrados	Muros, zapatas, castillos, instalaciones, etc.
	Partidas	Albañilería, cimentación, instalación hidráulica, instalación sanitaria, etc.
	Capítulos	Frentes de obra, edificio uno, edificio dos, conjunto habitacional A, etc.
	Extraordinarios	Son todos aquellos conceptos fuera de Obra.

Una vez que usted se encuentra dentro de la ventana de edición del Análisis:

2. En el campo de edición de **Clave**, proporcione la clave del Análisis, esta clave puede ser de hasta 25 caracteres.

Al proporcionar la Clave del Análisis, el Sistema de Precios Unitarios para Windows automáticamente le adicionará al principio de ésta, la letra correspondiente al Tipo de Análisis:

Letra	Tipo Análisis
B	Básicos
I	Integrados
P	Partidas
C	Capítulos
X	Extraordinarios

El Tipo del Análisis le servirá al Sistema para identificar a qué nivel pertenece el análisis, esto debido a que, en esta versión, es posible capturar Claves alfanuméricas.

3. Dentro del campo de edición **Descripción**, proporcione la descripción del Análisis, usted podrá introducir un texto de hasta 32,000 caracteres, éstos pueden ser dígitos, letras, espacios y cualquier signo del teclado estándar.
4. En el campo de **Familia**, proporcione la Familia a la que pertenece el Análisis, la captura de este dato es opcional y puede ser de hasta 11 caracteres. Si la Familia que usted capturó no se encuentra dada de alta, con sólo teclearla dentro de este campo se agregará al Catálogo de Familias.
5. Dentro del campo de **Unidad**, proporcione la unidad de medida del Análisis, el nombre de la unidad puede ser de hasta 11 caracteres.
6. En el campo de **Fecha de Costo**, capture la fecha en la cual se le asigna el Costo al Análisis.
7. Dentro del campo de **Fecha de Cálculo**, capture la fecha en la cual se recalcula el Costo del Análisis.

Para introducir las Fechas de Costo y Cálculo, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas:

Oprima la Tecla:	Si desea:
F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios
F9	Limpiar el campo de fecha

8. Dentro del campo de **Moneda**, usted debe proporcionar la **Moneda** (símbolo con la cual será calculado el Costo Total del Análisis, ésto lo puede realizar con sólo presionar el botón derecho del mouse dentro de esta celda seleccionar del menú la **Moneda** deseada.

Si el símbolo de la **Moneda** no se encuentra capturado o desea agregar una nueva **Moneda**, usted puede invocar al Catálogo de **Monedas** desde este módulo de la siguiente manera:



- Oprima el botón derecho del mouse, con ello se le presentará el menú flotante de **Monedas**.
- Seleccione la opción de **Catálogo**.
- Una vez que la **Moneda** ha sido capturada y seleccionada la **Moneda Base** dentro del Catálogo de **Monedas**, oprima el botón de **Aceptar** que se encuentra dentro este diálogo.

d Ya que se encuentre nuevamente dentro de la ventana de captura de Análisis, en el campo de **Moneda**, oprima nuevamente el botón derecho del mouse y seleccione del menú la Moneda deseada

CÓMO AGREGAR UN COMPONENTE A UN ANÁLISIS

Para agregar un componente a un Análisis, se sigue el mismo procedimiento no importando el nivel de Análisis que se encuentre editando. Este procedimiento es el siguiente:

1. Dentro de la celda de **Tipo** seleccione el tipo de Insumo o Análisis que desea agregar, estos tipos pueden ser:

Tipo	Descripción
M	Material
O	Obra de Mano
Q	Maquinaria y Equipo
T	Subcontrato
B	Básico
P	Partida
C	Capítulo
X	Extraordinario
H	Herramienta
HI	Mandos Intermedios
HM	Porcentaje sobre Materiales
HQ	Porcentaje sobre Maquinaria y Equipo
HT	Porcentaje sobre Subcontratos
HP	Porcentaje sobre Preliminares

Quando usted captura la clave del componente sin proporcionar el tipo, el Sistema de Precios Unitarios le asociará el tipo correspondiente a un nivel anterior del Análisis que se encuentra editando, por ejemplo:

Si usted se encuentra editando un:

Le asociará el tipo:

Básico	M correspondiente a Materiales
Integrado	B correspondiente a Básicos
Partida	I correspondiente a Integrados
Capítulo	P correspondiente a Partidas

2. En la celda de **Clave**, proporcione la clave del elemento. Al proporcionar la clave del elemento, éste puede aparecer de diferente color, siguiendo las siguientes reglas:

Color del Elemento:	Si es un Insumo:	Si es un Análisis:
Negro	Que ya se encuentra capturado dentro de los Insumos	Que ya se encuentra analizado, es decir ya tiene componentes.
Azul	<No aplica>	Que está capturado pero no contiene componentes
Rojo	Que no se encuentra capturado.	Que no se encuentra capturado

Si el elemento existe:

Automáticamente se llenarán los campos de **Descripción, Unidad, Moneda y Costo Unitario.**



El Costo Unitario del elemento, si se trata de un Insumo, será calculado en base al Precio Activo de éste, si se trata de un Análisis, este costo se calculará como la suma de todos los importes de los elementos que contiene.



El Costo Unitario del elemento se presentará en base a la Moneda con la que éste fue calculado.

Si el elemento existe y usted desea realizar una búsqueda a través de su descripción presione las siguientes teclas, cuando se encuentre en el renglón del elemento que desea capturar.

Teclas:

Para:

CTRL + D

Inicializar la búsqueda por descripción proporcionándole el primer elemento.

CTRL + B

Realizar un desplazamiento hacia el siguiente elemento, siguiendo el orden alfabético de la descripción.

CTRL + SHIFT + B

Realizar un desplazamiento hacia el elemento anterior, siguiendo el orden alfabético de la descripción.



Los elementos se encontrarán ordenados alfabéticamente, en base a su descripción.

Si el elemento no existe:

- a. Dentro de la celda de **Descripción**, capture la descripción del elemento.
- b. Dentro de la celda de **Unidad**, proporcione la unidad de medida del elemento.
- c. En la celda de **Moneda**, proporcione la Moneda bajo la cual se encuentra el Costo Unitario, con sólo presionar el botón derecho del mouse dentro de esta celda y seleccionar del menú la **Moneda** deseada.

Si esta Moneda no se encuentra capturada es necesario agregarla al Catálogo de Monedas,

d. Dentro de la celda de **Costo Unitario**, proporcione el Costo Unitario del elemento.



Si el elemento es una Obra de Mano, usted puede seleccionar el Tipo de Factor de Salario Real a aplicar, con sólo posicionar el apuntador del mouse dentro de esta celda y oprimir dos veces el botón izquierdo de éste, con ello aparecerá el diálogo para la Edición Rápida del Factor de Salario Real.

3. Dentro de la celda de **Cantidad**, proporcione la cantidad del componente a utilizar dentro del Análisis, ésta no puede ser mayor a 10 caracteres.

4. En la celda de **Rendimiento**, proporcione el Rendimiento del elemento, en cuanto usted proporciona el rendimiento la celda de **Importe**, presentará el importe para el elemento que es calculado de la siguiente manera:

$$\text{IMPORTE} = (\text{CANTIDAD} / \text{RENDIMIENTO}) * \text{COSTO UNITARIO}$$


El concepto de "Rendimiento", puede entenderse como desperdicio en el caso de Materiales y como rendimiento en caso de una Obra de Mano, por ejemplo:

Si un Peón tiene un Rendimiento del 95%, usted deberá colocar en este campo 0.95.

Si en un Material, "Tabique" por ejemplo, tengo un desperdicio del 10%, deberá colocar en el campo de **Rendimiento** 0.90.

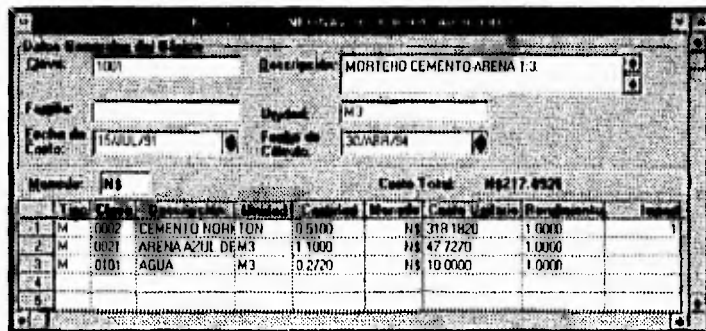
Si no existe desperdicio o el rendimiento es del 100%, deberá colocar el valor de 1 (Uno).

5. Si desea capturar un nuevo elemento, repita los pasos del 1 al 5.

6. Una vez que los datos proporcionados son correctos, oprima el botón  de la barra de herramientas Sistema de Precios Unitarios correspondiente a grabar.

EJEMPLO:

Para el cálculo del Presupuesto de la Obra Edificio, es necesario editar el Básico 1001, que es un Mortero Cemento - Arena 1:3. Este básico contiene tres elementos que son: el Material 002 (Cemento normal Cruz Azul en sacos), el Material 0021 (Arena azul de mina) y el Material 0101 (Agua). Para agregar este Básico es necesario editar lo siguiente y posteriormente oprimir el botón de **Grabar**.



The screenshot shows a software window titled "Datos Generales del Básico" with the following fields:

- Clave: 1001
- Descripción: MORTERO CEMENTO-ARENA 1:3
- Fecha de Costo: 15/JUL/91
- Fecha de Cálculo: 30/ABR/94
- Moneda: M\$
- Costo Total: M\$217.8528

Below the form is a table with the following data:

Id	Material	Descripción	Unidad	Cantidad	Material	Costo Unitario	Rendimiento	Importe
1	M	002 CEMENTO NORHTON		0.5100	M\$	318.1820	1.0000	
2	M	0021 ARENA AZUL DE MINA		1.1000	M\$	47.7270	1.0000	
3	M	0101 AGUA	M3	0.2720	M\$	10.0000	1.0000	
4								
5								

Para:

Ver:

Obtener información acerca de la edición de Análisis

El tema de "**Análisis**" dentro de la Ayuda en Línea del Sistema de Precios Unitarios

Obtener información acerca de los procedimientos a realizar para la edición de Análisis

El tema "**Análisis**" dentro de los Sistema de Precios Unitarios Tips.

Obtener información acerca de las Herramientas dentro de Análisis

El tema "**Como Agregar una Herramienta como un elemento del Análisis**" dentro de este Manual.

Obtener información acerca de la captura rápida del Factor de Salario Real

El tema "**Captura Rápida del Factor de Salario Real**" dentro de la Ayuda en Línea del Sistema de Precios Unitarios.

HERRAMIENTAS

Las Herramientas son porcentajes que usted puede aplicar sobre el total de un mismo tipo de elementos al elaborar sus Análisis.

En el Sistema de Precios Unitarios para Windows se consideran los siguientes porcentajes de Herramientas:

Para:	Ver:
Obtener información acerca de la edición de Análisis	El tema de " Análisis " dentro de la Ayuda en Línea del Sistema de Precios Unitarios
Obtener información acerca de los procedimientos a realizar para la edición de Análisis	El tema " Análisis " dentro de los Sistema de Precios Unitarios Tips.
Obtener información acerca de las Herramientas dentro de Análisis	El tema " Como Agregar una Herramienta como un elemento del Análisis " dentro de este Manual.
Obtener información acerca de la captura rápida del Factor de Salario Real	El tema " Captura Rápida del Factor de Salario Real " dentro de la Ayuda en Línea del Sistema de Precios Unitarios.

HERRAMIENTAS


Las Herramientas son porcentajes que usted puede aplicar sobre el total de un mismo tipo de elementos al elaborar sus Análisis.

En el Sistema de Precios Unitarios para Windows se consideran los siguientes porcentajes de Herramientas:

Tipo:	Descripción:	Se calcula sobre:
H	Herramientas	El total de la Obra de Mano y Cuadrillas
HI	Mandos Intermedios	El total de la Obra de Mano y Cuadrillas
HM	Porcentaje sobre Materiales	El total de Materiales
HQ	Porcentaje sobre Maquinaria y Equipo	El total de la Maquinaria y Equipo
HT	Porcentaje sobre Subcontratos	El total de Subcontratos
HP	Porcentaje sobre Preliminares	El total de Preliminares: Básicos, Integrados, Partidas, Capítulos y Obra Extraordinaria como elementos de un Análisis

EDICIÓN DE HERRAMIENTAS

Para editar las Herramientas que se incluirán dentro de los Análisis:

1. Desde la Barra de Herramientas del Sistema de Precios Unitarios, oprima el botón  correspondiente a la opción de Herramientas.

Una vez que usted se encuentra dentro de la ventana de Herramientas:

2. Dentro de combobox de **Tipo** seleccione el tipo de Herramienta que desea agregar, éste puede ser:

Tipo

Herramienta

Mandos Intermedios

Porcentaje sobre Materiales

Porcentaje sobre Maquinaria

Porcentaje sobre Subcontratos

Porcentaje sobre Preliminares

3. En el campo de edición de **Clave**, proporcione la clave de la Herramienta, esta clave puede ser de hasta 25 caracteres alfanuméricos.

4. Dentro del campo de edición de **Descripción**, proporcione la descripción de la Herramienta, esta descripción puede ser tan grande como lo desee, y puede contener dígitos, letras, espacios y cualquier signo del teclado estándar.

5. Dentro del campo de edición de **Unidad**, proporcione la unidad de cálculo para la herramienta, el Sistema de Precios Unitarios le presentará las siguientes unidades por default, sin embargo usted puede modificarlas de acuerdo a sus necesidades

Unidad	Se asigna a:
%H	Herramienta
%HI	Mandos Intermedios
%HM	Materiales
%HQ	Maquinaria y Equipo
%HT	Subcontratos
%HP	Preliminares

Tipo

Herramienta

Mandos Intermedios

Porcentaje sobre Materiales

Porcentaje sobre Maquinaria

Porcentaje sobre Subcontratos


Porcentaje sobre Preliminares

3. En el campo de edición de **Clave**, proporcione la clave de la Herramienta. esta clave puede ser de hasta 25 caracteres alfanuméricos.

4. Dentro del campo de edición de **Descripción**, proporcione la descripción de la Herramienta, esta descripción puede ser tan grande como lo desee, y puede contener dígitos, letras, espacios y cualquier signo del teclado estándar.

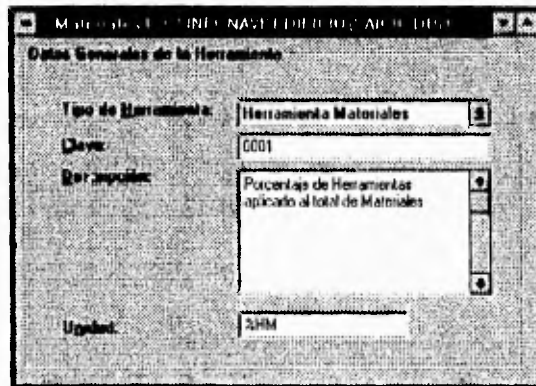
5. Dentro del campo de edición de **Unidad**, proporcione la unidad de cálculo para la herramienta, el Sistema de Precios Unitarios le presentará las siguientes unidades por default, sin embargo usted puede modificarlas de acuerdo a sus necesidades:

Unidad	Se asigna a:
%H	Herramienta
%HI	Mandos Intermedios
%HM	Materiales
%HQ	Maquinaria y Equipo
%HT	Subcontratos
%HP	Preliminares

- Una vez que los datos proporcionados son correctos, oprima el botón  de la barra de herramientas del Sistema de Precios Unitarios correspondiente a grabar.

EJEMPLO

En el siguiente ejemplo se desea agregar una Herramienta, la cual debe aplicarse sobre el total de Materiales del Análisis, esta herramienta contendrá la clave 0001, para ello es necesario capturar los siguientes datos dentro de la ventana de Herramientas y posteriormente oprimir el botón de **Grabar**.



Datos Generales de la Herramienta	
Tipo de Herramienta:	Herramienta Materiales
Clave:	0001
Porcentaje:	Porcentajes de Herramientas aplicado al total de Materiales
Unidad:	SHM

Para:

Ver:

Obtener información acerca de la edición de Herramientas

El tema de "**Herramientas**" dentro de la Ayuda en Línea del Sistema de Precios Unitarios

Obtener información acerca de los procedimientos a realizar para la edición de Herramientas, así como el procedimiento para agregarlas como componente de un Análisis

Dentro del tema Análisis "**Agregar una Herramienta como un elemento del Análisis**" dentro de los Sistema de Precios Unitarios Tips.

CÓMO AGREGAR UNA HERRAMIENTA COMO UN ELEMENTO DEL ANÁLISIS

Para agregar una herramienta como un elemento del Análisis, realice lo siguiente:

Dentro de la sección de captura de elementos:

1. Dentro de la celda de **Tipo** seleccione el tipo de Herramienta que desea agregar.

Tipo	Descripción
H	Herramienta
HI	Mandos Intermedios
HM	Porcentaje sobre Materiales
HQ	Porcentaje sobre Maquinaria y Equipo
HT	Porcentaje sobre Subcontratos
HP	Porcentaje sobre Preliminares

Una vez que usted selecciona el tipo de Herramienta, dentro de la celda de **Costo Unitario**, aparecerá el total de los importes, dependiendo del tipo de Herramienta seleccionado

2. En la celda de **Clave**, proporcione la clave de la Herramienta que desea agregar. Si la Herramienta existe, automáticamente se llenarán los campos de **Descripción, Unidad, Costo Unitario**, si ésta no existe, su clave se pintará de rojo y usted tendrá que proporcionar estos datos, en la ventana de edición de Herramientas, que describimos en el tema anterior.
3. Dentro de la celda de **Cantidad**, proporcione el porcentaje a aplicar de la Herramienta seleccionada, ésta puede ser de hasta 10 caracteres.
4. En la celda de **Rendimiento**, proporcione el Rendimiento de la Herramienta. El cuanto usted proporciona el rendimiento en la celda de **Importe**, presentará el importe para la herramienta, que es calculado de la siguiente manera:

$$\text{IMPORTE} = (\text{TOTAL DE LA HERRAMIENTA} / \text{RENDIMIENTO}) * \text{COSTO UNITARIO}$$

El Rendimiento para la Herramienta, no tiene mucho sentido, usualmente usted deberá colocar un valor de uno (1), pero si usted así lo desea puede colocar cualquier cantidad.
5. Si desea capturar una nueva herramienta, repita los pasos del 1 al 4.

EJEMPLO: En el siguiente ejemplo se quiere agregar al Análisis con clave 0001, un cargo por Herramienta y otro por Mandos Intermedios, que deben ser aplicadas sobre el total de la Obra de Mano del Análisis, para ello es necesario editar lo siguiente:

Tipo	Clave	Descripción	Unidad	Cantidad	Moneda	Costo Unitario	Remanente
	0001	OBRERA LE 01 (14) OBRERA 'A'	JOB	1.00000		49.01507974	1.00000
	0001	SABO PAPA AYUDANTES 'A'	JOB	1.00000		99.07530836	23.00000
	0001	Herramienta	JOB	0.10000		82.77662015	1.00000
	0001	Porcentaje sobre Mandos Intermedios	JOB	0.04000		52.74967015	1.00000

CÓMO ELABORAR OTROS ANÁLISIS MIENTRAS SE MANTIENE UNO ABIERTO

En el Sistema de Precios Unitarios para Windows usted puede elaborar cualquier nivel de Análisis aún cuando otro se encuentre abierto.

Lo anterior lo puede realizar de varias formas, según como usted acostumbre realizar sus Presupuestos. A continuación explicaremos estas formas y cómo puede usted aplicarlas.

CÓMO INDENTAR LOS ELEMENTOS DENTRO DE UN ANÁLISIS

Una de las principales características que el Sistema de Precios Unitarios para Windows le ofrece es la de realizar la captura de todos los elementos que se requieren para el Análisis, desde la misma ventana de edición, esto lo puede realizar, con sólo indentar adecuadamente los elementos de dicho Análisis.

Si para capturar un Análisis, le hace falta editar otro Análisis el cual forma parte de sus componentes, usted puede realizarlo sin necesidad de cambiar de ventana, esto con sólo capturar los elementos que hacen falta para definir el Análisis nuevo y agruparlos

para que el Sistema de Precios Unitarios sepa que forman parte de este nuevo Análisis, esta agrupación se hace por medio de la acción de indentar, por ejemplo, si usted quiere elaborar una Cuadrilla que le hace falta para completar el Análisis del concreto:


	Unid.	Clave	Descripción	Unidad	Cantidad
1	M	0002	CEMENTO NORMAL CRUZ AZUL TON		0.3210
2	M	0021	ARENA AZUL DE MINA	M3	0.4090
3	M	0027	GRAVA DE MINA DE 3/4"	M3	0.7660
4	M	0101	AGUA	M3	0.1700
5	B	0001	CUADRILLA 001 (1 AYUDANTE)	JOR	3.0000
6	O	0001	AYUDANTE "A"	JOR	3.0000
7	O	4001	CABO PARA AYUDANTES "A"	JOR	3.0000
8	B	0010	CUADRILLA 010 (1 OF. ALB. + 1 JOR)		1.0000
9	O	0012	REVOLVEDORA 1 SACO MIPSÁ HR		0.0000

Estos 2 elementos, deben estar indentados con respecto al elemento Cuadrilla 001, con lo que SAICIC entiende que la Cuadrilla 001 es otro Análisis que estará formado por los conceptos "Ayudante A" y un "Cabo para Ayudantes A".

A continuación describiremos el procedimiento para realizar la indentación de los elementos del Análisis:

1. Dentro de los componentes edite como cualquier elemento al Análisis que se encuentra anidado. A este elemento capturado le llamaremos Elemento Padre. En nuestro ejemplo anterior este es la Cuadrilla 001.
2. Agregue como elementos del Análisis a los componentes del Elemento Padre. A todos los componentes del Elemento Padre les llamaremos Elementos Hijos. En nuestro ejemplo anterior son el Ayudante "A" y el Cabo para Ayudantes "A".
3. Una vez que los datos del Elemento Padre y del o de los Elementos Hijo(s) se encuentran capturados, posicione en el renglón del Elemento Hijo.
4. Oprima simultáneamente las teclas CTRL + @, usted puede observar que la **Clave** del Elemento se mostrará con una sangría, la cual le indicará que este elemento es hijo del Elemento Padre.

Para realizar la indentación de los elementos de los Análisis existen las siguientes reglas:

- Sólo se puede realizar la indentación de un elemento, si el elemento que lo precede es de mayor jerarquía, es decir, un Insumo no puede ser el Elemento Padre de un Análisis.
 - El primer elemento del Análisis no puede ser indentado debido a que no existen elementos que lo precedan.
 - Al borrar un elemento que contiene varios hijos, éstos también se borrarán.
5. Si existen más Elementos Hijos, repita el paso anterior para cada uno de ellos.
6. Una vez que los datos proporcionados son correctos, oprima el botón  de la barra de herramientas del Sistema de Precios Unitarios correspondiente a grabar.

Con lo que se grabará el Análisis "Concreto" y automáticamente se generará un nuevo Análisis que es la "Cuadrilla". La próxima vez que se muestre en pantalla el Análisis "Concreto" dentro de los elementos ya sólo aparecerá el elemento Padre "Cuadrilla" y no sus Hijos debido a que ya se encuentra analizado y almacenado en la Base de Datos de su Obra

Si estando en el elemento "Cuadrilla" presiona las teclas CTRL + F3, automáticamente se abrirá una nueva ventana que contiene el Análisis del elemento "Cuadrilla"

EJEMPLO:

Si el Básico con Clave B2003 (Concreto F'C=150 3/4" H. O. Revolvedora) contiene al Básico B0001, y a su vez el Básico B0001, es una cuadrilla que contiene a la Obra de Mano O0001, a la Obra de Mano O4001 y un cargo por Herramienta y otro por Mandos Intermedios, y usted desea capturar la cuadrilla desde el concreto, es necesario editar los datos de la siguiente manera:

Nivel	Código	Clave	Descripción	Unidad	Cantidad	Valor
M	0021		CEMENTO NORMAL CHU	TON	0.3210	14.318182
M	0022		ARENA AZUL DE MINA	M3	0.4090	14.477270
M	0101		GRAVA DE MINA DE 3/4"	M3	0.7060	14.477270
M	0101		AGUA	M3	0.1700	14.100000
B	0001		CUADRILLA 001 (1)	JOR	3.0000	14.675306
B	0001		AYUDANTE "A"	JOR	1.0000	14.48.0045
B	0001		CABO PARA AYUDANTES "A"	JOR	1.0000	14.95.0759
B	0003		Porcentaje sobre Maquinaria y	%M U	0.1000	14.52.7583
B	0001		Porcentaje sobre Mandos	%M U	0.0400	14.52.7501

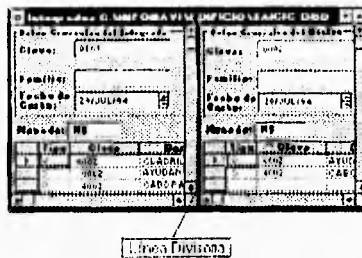
CÓMO VISUALIZAR DOS NIVELES DE ANÁLISIS DIFERENTES EN UNA MISMA VENTANA

Como mencionamos al principio de este capítulo usted puede analizar dos niveles de Análisis dentro de una misma ventana, esto le ayudará si usted está elaborando un Presupuesto complejo, o un Presupuesto desde el principio (es decir sin nada capturado), en el primer caso, usted puede revisar la elaboración de otros Análisis para confirmar su contenido y volúmenes correctos. En el segundo caso usted puede dar de alta los Análisis que se requieren, o modificar Análisis que se dejaron incompletos.

Para visualizar dos niveles de Análisis en una misma ventana, es necesario realizar lo siguiente:

1. Seleccione el nivel de Análisis que desea, presionando el botón dentro de la Barra de Herramientas, correspondiente al nivel de Análisis que desea visualizar.

2. Posicione el apuntador del mouse en la línea divisoria de la ventana, este cambiará a la siguiente forma.



3. Mueva la línea divisoria hacia la izquierda, manteniendo el botón izquierdo del mouse presionado, de manera que usted pueda observar los datos de la parte izquierda y los de la parte derecha.
4. Posiciónese dentro de cualquier control de la parte derecha de la ventana.
5. Oprima simultáneamente la tecla CTRL y el botón correspondiente al nivel de Análisis que desea visualizar.

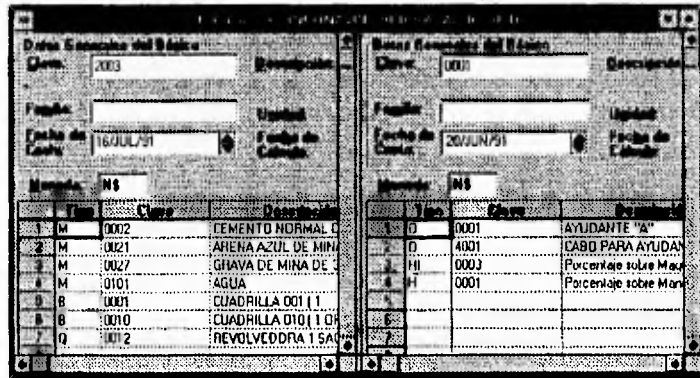


El desplazamiento entre registros y la edición del Análisis, será independiente en cada una de las partes de la ventana.

EJEMPLO

En el ejemplo siguiente se muestra una ventana de Análisis, que es el resultado de ejecutar el procedimiento anterior, en ella podemos observar del lado izquierdo, al Básico B2003 (Concreto F'C=150 3/4" H. O. Revolvedora) que contiene al Básico 0001 que es una Cuadrilla, y en la parte derecha el Análisis correspondiente a esta Cuadrilla.

7



CÓMO ABRIR UNA NUEVA VENTANA PARA EDITAR LOS DATOS CORRESPONDIENTES A UN COMPONENTE DEL ANÁLISIS

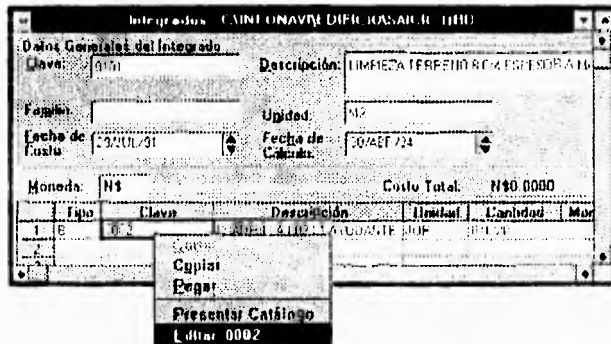
Si usted se encuentra capturando los componentes de un Análisis y desea en ese momento editar los datos correspondientes a este componente, usted puede abrir una nueva ventana la cual corresponderá a la edición del componente en cuestión (Insumo o Análisis), para ello, es necesario realizar lo siguiente:

Una vez que usted proporcione el **Tipo** y la **Clave** del componente:

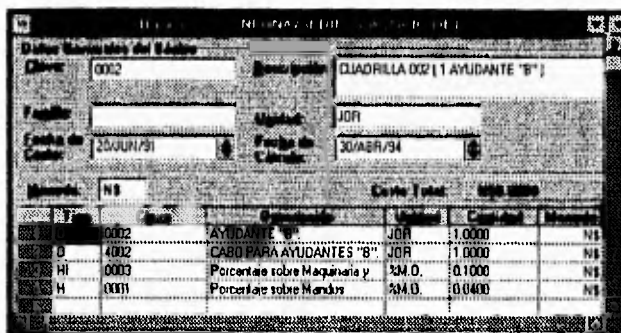
1. Posiciónese dentro de la celda de **Clave**, dentro de los datos del componente.
2. Oprima las teclas CTRL+F3, con ello se abrirá una nueva ventana correspondiente a la edición del componente, éste puede ser un Insumo o un Análisis.

Usted también puede abrir la ventana de edición correspondiente al componente que se encuentra capturando, con sólo oprimir el botón derecho del mouse y del menú flotante que se presenta seleccionar la opción de **Editar**.

EjemPlo: En el ejemplo siguiente nos encontramos editando el Integrado con la clave 0101, que contiene como componente al Básico B0002, para consultar algunos de los datos asociados a este componente, oprimimos el botón derecho del mouse, con ello se presenta un menú flotante como se muestra a continuación.

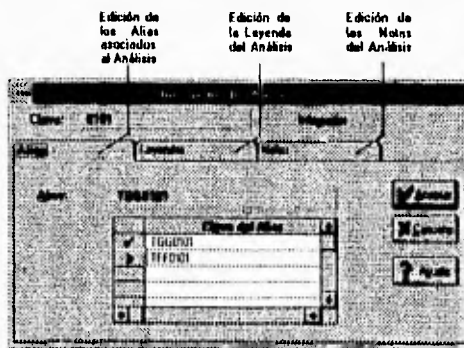


De este menú seleccionamos la opción de **Editar: B0002**, con ello se presentará la ventana de edición del Básico 0002, como se muestra a continuación.



EDICIÓN DE ALIAS, LEYENDAS Y NOTAS EN ANÁLISIS

Dentro de este diálogo usted puede capturar información adicional del Análisis para ello cuenta con varios separadores que le permiten capturar los Alias y las Leyendas asociadas a éste.




El botón de **Información del Análisis**, se encontrará deshabilitado, hasta que usted proporcione la Clave del Análisis, o edite un Análisis ya existente.

ALIAS

Dentro de este separador usted puede capturar los diferentes Alias asociados con el Análisis. Un Alias es un nombre diferente que se le dará a la Clave del Insumo o Análisis, con el fin de que pueda ser utilizado para diferentes presupuestos, ya que la Dependencia Pública, le puede solicitar que la Obra que usted va a presupuestar, lleve las Claves propias de esa dependencia.

Usted puede capturar tantos Alias como desee para el Análisis, pero dentro de sus reportes sólo se reflejará el Alias que se encuentre activo.

A continuación describiremos el procedimiento utilizado para editar los Alias:

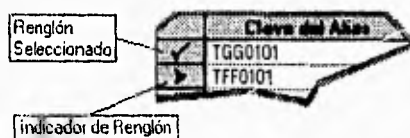
1. Oprima el botón  dentro de la Barra de Captura.

O desde el menú de **Edición** seleccione la opción de **Información**.

U oprima simultáneamente las teclas CTRL+I.
2. Una vez dentro del diálogo de Información de Análisis, seleccione el separador **Alias**.
3. En la celda de **Clave del Alias**, proporcione el Alias del Análisis, éste puede ser de hasta 25 caracteres alfanuméricos.
4. Si desea capturar un nuevo Alias relacionado al Análisis, posicione en el primer renglón que se encuentre vacío y repita el paso anterior

5. Una vez capturados los Alias relacionados con el Análisis, dentro de la columna del indicador de renglón, elija el Alias con el cual se representará el Análisis dentro de sus reportes, en caso de que usted los solicite con Alias.

Para seleccionar el Alias:



- a. Posiciónese dentro de la celda del Indicador de Renglón que contiene el Alias deseado.
- b. Oprima dos veces la BARRA ESPACIADORA del teclado.

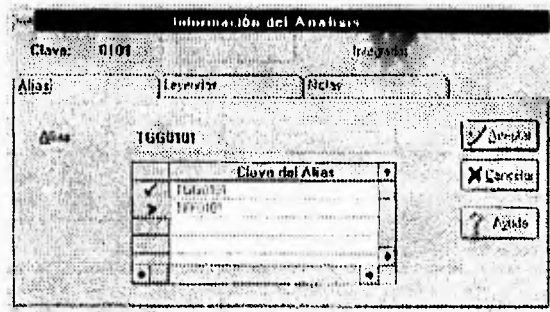
O posicione el apuntador del mouse sobre la celda del Indicador de Renglón que contiene el Alias deseado y oprima el botón izquierdo de éste.



En el campo de **Alias** se mostrará el Alias del Insumo o Análisis que se presentará dentro de los reportes. Cada vez que usted seleccione un Alias diferente, el cambio también se reflejará dentro de este campo.

6. Oprima el botón de **Aceptar** si está de acuerdo con la información proporcionada, de lo contrario oprima el botón de **Cancelar**.

EJEMPLO: El Integrado con clave 0101, tiene asignados varios Alias, pero el Alias que se encuentra activo y que se presentará dentro de los reportes si así se indica, para este ejemplo es el Alias TGG0101.



LEYENDAS

Dentro de este separador podrá capturar la Leyenda correspondiente al Análisis que se encuentre editando. Las Leyendas son textos que usted puede asociar a los diferentes Insumos o Análisis, como una segunda Descripción y poderla utilizar para presupuestar una Obra relacionada con otra dependencia.

Para editar la Leyenda es necesario realizar lo siguiente:

1. Oprima el botón  dentro de la Barra de Captura.

O desde el menú de **Edición** seleccione la opción de **Información**.

U oprima simultáneamente las teclas **CTRL+I**.

2. Una vez dentro del diálogo de Información del Análisis, seleccione el separador **Leyendas**.

3. En el campo de **Leyenda**, introduzca el texto correspondiente a la Leyenda del Análisis, la cual puede ser tan grande como usted lo desee.
4. Oprima el botón de **Aceptar** si está de acuerdo con la información proporcionada, de lo contrario oprima el botón de **Cancelar**.



Sólo se puede manejar una sola Leyenda para cada Insumo o Análisis, ésta será la Leyenda utilizada en caso de solicitar los reportes con Leyendas.

NOTAS

En este separador usted puede capturar anotaciones técnicas o de criterio al Análisis que se encuentra editando. Estas Notas se pueden imprimir en los reportes de Análisis y Presupuestos.

Para editar las Notas es necesario realizar lo siguiente:

1. Oprima el botón  dentro de la Barra de Captura.

O desde el menú de **Edición** seleccione la opción de **Información**.

U oprima simultáneamente las teclas CTRL+I.

2. Una vez dentro del diálogo de Información del Análisis, seleccione el separador **Notas**.
3. En el campo de **Notas**, introduzca las anotaciones correspondientes al Análisis que se encuentra editando, las Notas pueden ser tan grandes como usted lo desee.

4 Oprima el botón de **Aceptar** si está de acuerdo con la información proporcionada, de lo contrario oprima el botón de **Cancelar**.



Sólo se puede manejar una Nota para cada Análisis, ésta será la Nota utilizada en caso de solicitar los reportes con Notas.

SECCIÓN 8 PRESUPUESTOS




CÓMO ACTUALIZAR LOS PRECIOS EN EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

Esta opción del Sistema de Precios Unitarios, le permite actualizar los precios de los Análisis y de sus elementos, individualmente o por Obra.

Si usted modificó el costo de algún Insumo que intervenga en el Presupuesto, es necesario recalcular los precios para obtener el costo real de la Obra. el Sistema le permite realizar el recálculo en toda la Obra, sin importar en qué precio o precios intervenga el Insumo modificado.

Para realizar una Actualización de Precios, es necesario realizar lo siguiente:

1. Desde la Barra de Herramientas oprima el botón  correspondiente a la opción de Actualización de Precios en el Sistema.
2. En el campo de **Fecha a Utilizar**, capture la fecha de Actualización de Precios. Esta fecha será grabada en cada uno de los elementos actualizados como la **Fecha de Cálculo**.

Para introducir la Fecha de Actualización, existe un conjunto de teclas, las cuales pueden ser utilizadas a lo largo de todo el sistema, a continuación se describe cada una de ellas:

Opresión de la Tecla	Si desea
F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios
F9	Limpiar el campo de fecha

3. Dentro del combobox **Nivel de la Actualización**, seleccione el nivel de Actualización, los niveles de Actualización que nos presenta son los siguientes:

Nivel	Esta opción permite:
Obra Completa	Actualizar el precio de todos los Análisis o Matrices que se encuentran en la Base de Datos. Esta opción es recomendable cuando se desea actualizar los precios de todos los Análisis que forman al Presupuesto y también los que no forman parte de él pero sí de la Base de Datos.
Básicos	Actualizar el precio de un rango de Básicos
Integrados	Actualizar el precio de un rango de Integrados.
Partidas	Actualizar el precio de un rango de Partidas.
Capítulos	Actualizar el precio de un rango de Capítulos
Extraordinarios	Actualizar el precio de un rango de Extraordinarios.

Si usted desea seguridad (que todo se encuentre actualizado) y no le importa el tiempo, entonces es conveniente que seleccione la opción de "OBRA COMPLETA" Por el contrario, usted sabe que su Obra está a nivel de Capítulos o Partidas y que ahí está contenido todo su presupuesto y desea velocidad, entonces es recomendable que seleccione sólo el nivel adecuado para la actualización, por ejemplo

"A NIVEL CAPÍTULOS"

Si el Nivel de Actualización seleccionado fue el de Obra Completa, dentro del campo **Nivel del Presupuesto**, se presentará el nivel al cual está formada la Obra (Básicos, Integrados, Partidas o Capítulos).

4. Seleccione el rango de Actualización para Básicos, Integrados, Partidas, Capítulos y Extraordinarios, para ello:
 - a. Dentro del campo de edición **Desde**, proporcione la Clave del Análisis a partir del cual se realizará la Actualización.
 - b. Dentro del campo de edición **Hasta**, proporcione la Clave del Análisis en el cual se detendrá la Actualización.



Si usted desea actualizar todos los elementos del nivel de Análisis seleccionado, no modifique los valores que el Sistema de Precios Unitarios presenta por default dentro del campo **Desde** y **Hasta** ([PRIMERO] [ULTIMO] respectivamente).

5. Para iniciar el proceso de Actualización, oprima el botón de **Aceptar**.

En el campo de **Elemento Actualizado**, se desplegarán las Claves de los elementos (Insumos y Análisis) que componen a la Obra o a los Análisis indicados en el rango, conforme transcurre el proceso de Actualización de Precios.

Una vez terminado el proceso, en el campo **Tiempo de Ejecución**, se muestra el tiempo transcurrido durante la Actualización de Precios. Mientras que en el campo **Número de Registros Procesados**, se mostrará el número de registros que fueron Actualizados.

6. Para cerrar el diálogo de Actualización de Precios, oprima el botón de **Salir**

CÓMO DEFINIR LOS FORMATOS DE SOBRE COSTO


En el Sistema usted puede elaborar sus propios cálculos para definir cualquier Factor de Sobre Costo que se requiera aplicar a los conceptos del Presupuesto y a los Análisis al momento de emitir sus reportes.

El Sistema incluye una serie de formatos ya definidos, siendo éstos los más comunes, además usted puede definir nuevos formatos, o redefinir los ya existentes, utilizando las variables y fórmulas según se requieran.

Conforme se define el formato, usted puede ir observando como se vería dicho formato en sus reportes, esto dentro de la sección **Ejemplo del Valor Elegido**.

Para definir los Formatos de Sobre Costo en el Sistema de Precios Unitarios:

1. Si se encuentra dentro de cualquier opción del Sistema:

Desde la Barra de Herramientas, oprima el botón  correspondiente a la opción de Factores de Sobre Costo.

Si se encuentra dentro de la ventana de Selección de Opciones de Reportes:

- a. Elija el Tipo de Formato dentro del combobox de **Tipo de Formato**.
- b. Oprima el botón de **Editar**.

Una vez que usted se encuentra dentro de la ventana de Factores de Sobre Costo:

2. Dentro del combobox de **Nombre**, teclee el nombre con el cual identificará al formato del Factor de Sobre Costo que desea definir o seleccione el nombre del formato que desea redefinir.

El Sistema le proporciona 8 formatos ya predefinidos que usted puede utilizar o modificar según sus necesidades.

Para:

Obtener información acerca de los 8 formatos que el Sistema de Precios Unitarios le presenta por default.

Ver:

El tema de "**Formatos de Sobrecosto**" dentro de la Ayuda en Línea del Sistema de Precios Unitarios



El nombre de la variable puede contener cualquier caracter alfanumérico, sin embargo, no puede contener espacios en blanco.

3. Dentro de las celdas de **Información de los Conceptos**, aparecerán algunas variables que el Sistema de Precios Unitarios para Windows ya tiene definidas y que le pueden ser de gran utilidad para definir el formato del Factor de Sobre Costo. Estas son:

Variable:	Que es el:
Costo	Costo Unitario del concepto en cuestión, sobre él se aplicará el sobre costo.
TotMat	Total de Materiales del concepto en cuestión.
TotManOb	Total de Mano de Obra del concepto en cuestión.
TotMaqEq	Total de Maquinaria y Equipo del concepto en cuestión.
TotSubCont	Total de Subcontratos



Los valores del Costo Unitario, TotMat, TotManOb, TotMaqEq y TotSubCont que aparecen dentro de la columna **Fórmula**, son valores arbitrarios, es decir, no son los calculados dentro del Presupuesto, usted podrá modificar estos valores lo que le ayudará a realizar un análisis de sensibilidad de cómo se reflejará dentro de sus reportes el tipo de formato que usted definió. Sin embargo, cuando usted imprima su reporte, estas variables si contendrán los valores reales que se obtuvieron dentro del cálculo del Presupuesto.

-
- Si desea agregar una nueva variable al Formato posicione en el primer renglón que se encuentre vacío.
 - En la celda de **Variable**, teclee el nombre de la variable que desea agregar, por ejemplo, "INDIRECTOS", por medio de este nombre identificaremos a la operación que definamos dentro de la celda de **Fórmula**.
 - Dentro de la celda de **Descripción** proporcione la descripción o alguna nota referente a dicha variable, como por ejemplo, 25% DE INDIRECTOS



Esta descripción es la que aparecerá para identificar a cada una de las variables dentro de sus reportes

7. En la celda de **Fórmula**, proporcione la fórmula de la nueva variable, dentro de la fórmula usted puede utilizar las variables que se encuentran definidas dentro del Sistema o las que usted mismo defina.

Por ejemplo si deseamos definir la variable de "INDIRECTOS", la cual se define como:

$$\text{INDIRECTOS} = \text{COSTO UNITARIO} * 0.25$$

deberemos de introducir dentro de la celda de **Fórmula**, correspondiente a "INDIRECTOS" lo siguiente:

$$\text{COSTO} * 0.25$$



Dentro de la fórmula, usted puede teclear las variables con mayúsculas o minúsculas ya que éstas representarán lo mismo.

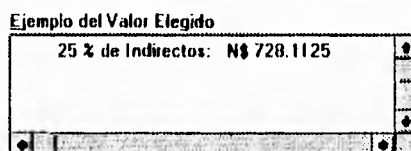
8. Si desea realizar la suma de las variables antes capturadas:
- a. Posiciónese en el primer renglón vacío después de las variables que desea sumar.
 - b. Oprima el botón de **Separador**, al oprimir este botón dentro de su reporte aparecerá una línea que servirá para reflejar un subtotal.

Hasta este paso usted sólo le ha indicado al Sistema de Precios Unitarios que desea realizar un subtotal.

- c. Posiciónese en el siguiente renglón y presione el botón de **Corte**.

Al presionar este botón usted le indicará al Sistema de Precios Unitarios que realice la suma de todas las variables que ha capturado anteriormente. Si desea agregarle una descripción a la suma realizada posiciónese dentro de la celda de **Descripción**, correspondiente al renglón donde se realizó el corte y capture su descripción.

Dentro del campo estático de **Ejemplo del Valor Elegido**, usted puede observar como se reflejará en su reporte el formato que acaba de definir, por ejemplo si usted define la variable "INDIRECTOS" del ejemplo anterior, usted vería dentro de este campo lo siguiente:



9. Una vez que usted ha terminado de definir el o los Formatos de los Factores de Sobre Costo, oprima el botón de **Salir**, para que se grabe la información, o si no desea guardar las modificaciones oprima el botón de **Cancelar**.

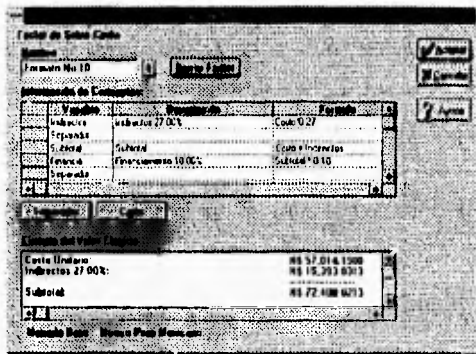
EJEMPLO:

En la Obra Edificio se desea aplicar el siguiente Factor de Sobre Costo al Presupuesto, este formato considera las disposiciones publicadas en el Diario Oficial de la Federación del 13 de Junio de 1994, relativas a desglosar en los Análisis de Precios Unitarios los indirectos correspondientes al S.A.R., INFONAVIT y SECOGEF:

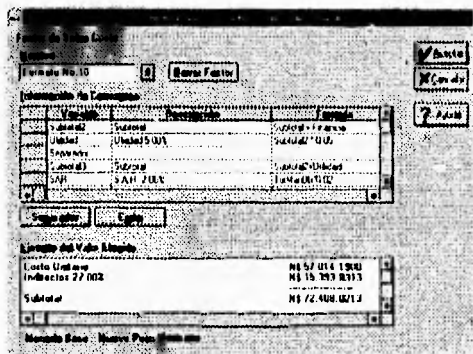
COSTO
INDIRECTOS 27%
COSTO+INDIRECTOS
FINANCIAMIENTO 10%
COSTO+INDIRECTOS+FINANCIAMIENTO
UTILIDAD 5%
COSTO+INDIRECTOS+FINANCIAMIENTO+UTILIDAD
S.A.R. 2% SOBRE EL TOTAL DE LA OBRA DE MANO
COSTO+INDIRECTOS+FINANCIAMIENTO+UTILIDAD+S.A.R.
INFONAVIT 5% SOBRE EL TOTAL DE LA OBRA DE MANO
COSTO+INDIRECTOS+FINANCIAMIENTO+UTILIDAD+S.A.R. +INFONAVIT
SECOGEF 0.50%
PRECIO=COSTO+INDIRECTOS+FINANCIAMIENTO+UTILIDAD+S.A.R. +
INFONAVIT+SECOGEF

Para ello es necesario que al diálogo de Definición de Factores de Sobre Costo se agreguen las siguientes variables a las ya existentes:

Primeramente definiremos la variable llamada INDIRECTOS, la cual se calcula como: el $COSTO * 0.27$ y cuya descripción es "Indirectos 27.00%" (esta y todas las descripciones de las variables son las que aparecerán dentro de los reportes), enseguida oprimiremos el botón **Separador**, el cual nos permitirá indicar que se realizará un subtotal, este subtotal se encuentra definido en el renglón inmediato inferior como la variable SUBTOTAL que se calcula como el $COSTO + INDIRECTOS$, posteriormente definiremos la variable FINANCIAMIENTO la cual se representa como el $SUBTOTAL * 0.10$ con la siguiente descripción "Financiamiento 10.00%" y enseguida un separador para indicar un segundo subtotal, éste se encontrará representado por la variable SUBTOTAL2 que se encuentra declarada en el renglón inmediato inferior y que es definida como $SUBTOTAL + FINANCIAMIENTO$. El diálogo de Definición de factores de Sobre Costo se observará de la siguiente manera:

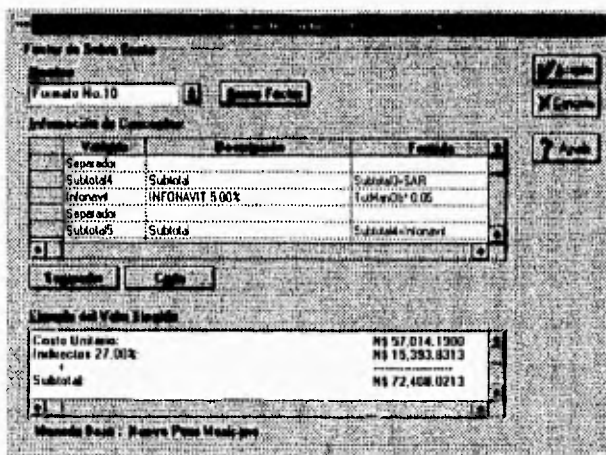


Continuando con el ejemplo declaremos la variable UTILIDAD la cual se define como el $SUBTOTAL2 * 0.05$ con la siguiente descripción "Utilidad 5.00%", posteriormente oprimamos el botón **Separador**, para definir un subtotal, el cual se encuentra representado por la variable SUBTOTAL3 que se encuentra definida en el renglón inmediato inferior como $SUBTOTAL2 + UTILIDAD$, enseguida declaremos la variable SAR que se calcula como $TOTMANOB * 0.02$ ($TOTMANOB = \text{Total de la Obra de Mano}$) y cuya descripción es "S.A.R. 2.00%" como se muestra a continuación:

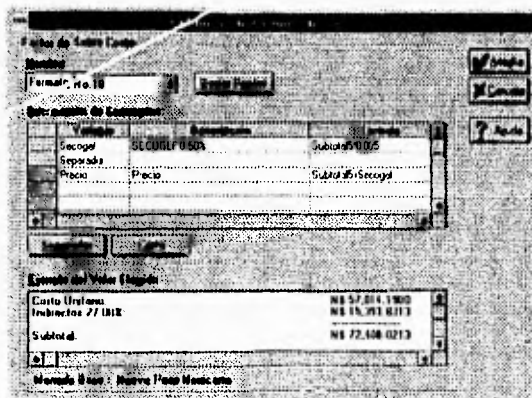


A continuación oprimamos el botón de **Separador** y declaremos la variable SUBTOTAL4, definida como $SUBTOTAL3 + SAR$, posteriormente declaremos la variable INFONAVIT que se calcula como $TOTMANOB * 0.05$ ($TOTMANOB = \text{Total}$

de la Obra de Mano) y cuya descripción es "INFONAVIT 5.00%", realicemos otro subtotal, primero oprimiendo el botón **Separador** y después declarando la variable SUBTOTAL5 en el renglón inmediato inferior, definida como SUBTOTAL4+INFONAVIT, como se muestra a continuación:



Por último definamos la variable SECOGEF como $SUBTOTAL5 * 0.005$ con la descripción "SECOGEF 0.50%", oprimamos el botón **Separador** y en el renglón inmediato inferior, declaremos la última variable que es el PRECIO como el $SUBTOTAL5 + SECOGEF$, como se muestra a continuación:



El campo Ejemplo del Valor Elegido, le mostrará como se aplicará este formato dentro de su reporte, como se muestra a continuación.

Ejemplo del Valor Elegido

Costo Unitario:	N\$ 2.912.4500
Indirectos 27.00%:	N\$ 786.3615
Subtotal:	N\$ 3.698.8115
Financiamiento 10.00%:	N\$ 369.8812
Subtotal:	N\$ 4.068.6927
Utilidad 5.00%:	N\$ 203.4346
Subtotal:	N\$ 4.272.1273
S.A.R. 2.00%:	N\$ 14.2540
Subtotal:	N\$ 4.286.3813
INFONAVIT 5.00%:	N\$ 35.6350
Subtotal:	N\$ 4.322.0163
SECOGEF 0.50%:	N\$ 21.6101
Precio	N\$ 4.343.6264

Para:

Obtener información acerca de la edición de los Formatos de Sobre Costo

Ver:

El tema de "**Factores de Sobre Costo**" dentro de la Ayuda en Línea del Sistema de Precios Unitarios


EXPLOSIÓN DE INSUMOS

Dentro de esta opción del Sistema de Precios Unitarios para Windows, usted puede realizar la Explosión de Materiales para la Obra Presupuestada, así como para Capítulos, Partidas, Integrados y Extraordinarios en forma independiente.

Los datos que se presentarán para cada Insumo resultante de la Explosión (Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos y Herramientas), son la Clave, la Descripción, la Unidad, el Costo Unitario, la Cantidad, el Importe y el Porcentaje de Incidencia con respecto al Importe Total de la Obra, Capítulo, Partida o Integrado explotado. También le presentará el Importe Total de la Obra, Capítulo, Partida o Integrado y el total en importe de cada grupo de Insumos, incluyendo la Herramienta, así como el porcentaje de incidencia total.



El campo **Nivel del Presupuesto**, le indicará a qué nivel está formada s Obra.

1. Desde la Barra de Herramientas oprima el botón  correspondiente a la opción de Explosión de Insumos en el Sistema de Precios Unitarios para Windows.

A continuación se muestra el procedimiento para realizar la Explosión de Insumos.

Dentro del Rango de Explosión:

2. En el combobox Nivel Explosión, seleccione para qué elemento se realizará la Explosión de Insumos, la Explosión puede ser:


EXPLOSIÓN DE INSUMOS

Dentro de esta opción del Sistema de Precios Unitarios para Windows, usted puede realizar la Explosión de Materiales para la Obra Presupuestada, así como para Capítulos, Partidas, Integrados y Extraordinarios en forma independiente.

Los datos que se presentarán para cada Insumo resultante de la Explosión (Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos y Herramientas), son la Clave, la Descripción, la Unidad, el Costo Unitario, la Cantidad, el Importe y el Porcentaje de Incidencia con respecto al Importe Total de la Obra, Capítulo, Partida o Integrado explotado. También le presentará el Importe Total de la Obra, Capítulo, Partida o Integrado y el total en importe de cada grupo de Insumos, incluyendo la Herramienta, así como el porcentaje de incidencia total.



El campo **Nivel del Presupuesto**, le indicará a qué nivel está formada s Obra.

1. Desde la Barra de Herramientas oprima el botón  correspondiente a la opción de Explosión de Insumos en el Sistema de Precios Unitarios para Windows.

A continuación se muestra el procedimiento para realizar la Explosión de Insumos.

Dentro del Rango de Explosión:

2. En el combobox Nivel Explosión, seleccione para qué elemento se realizará la Explosión de Insumos, la Explosión puede ser:

Por:

Obra

Integrado

Partida

Capítulo

Extraordinario

3. Dentro del campo de edición **Desde**, proporcione la Clave del Análisis a partir del cual se realizará la Explosión.
4. Dentro del campo de edición **Hasta**, proporcione la Clave del Análisis hasta donde se desea efectuar la Explosión.



Si la Explosión de Insumos la realizará para toda la Obra, los campo **Desde** y **Hasta** permanecerán deshabilitados.

5. En el combobox **Ordenado Por**, seleccione a través de qué campo se mostrarán ordenados los Insumos resultantes de la Explosión, estos campos son la Clave, la Descripción, la Unidad, el Costo, la Cantidad o el Importe.
6. Oprima el botón de **Explotar**.
7. Una vez que la Explosión de Insumos ha terminado, en la tabla que se muestra en la parte inferior de la ventana, aparecerán los datos asociados a los Insumos resultantes de esta Explosión (Clave, Descripción, Unidad, Costo Unitario, Cantidad, Importe y el Porcentaje de

Incidencia con respecto al Importe Total de la Obra, Capítulo, Partida o Integrado explotado).

También se presentará el Importe Total de la Obra, Capítulo, Partida o Integrado, la suma total de los importes de cada grupo de Insumos incluyendo la Herramienta, así como el porcentaje de incidencia total.

SECCIÓN 9 ESTIMACIONES



Esta opción del Sistema de Precios Unitarios para Windows le permite elaborar las Estimaciones periódicas de todas las Obras que se han presupuestado para poder realizar el cobro oportuno de las mismas.

No es necesario que la Obra esté presupuestada a nivel de Capítulos, ya que el Sistema de Precios Unitarios le permite estimar Partidas, Integrados y hasta los mismos Básicos.


En el Sistema se pueden generar tantas Estimaciones con tantos Conceptos o componentes como se deseen y como el Sistema lleva el registro de todas ellas, usted puede conocer al instante los acumulados anteriores.

La elaboración de Estimaciones es muy sencilla, por lo que no es necesario que las realice un Ingeniero de Costos, el Sistema de Precios Unitarios sólo requiere de la clave del Concepto y la Cantidad elaborada y automáticamente se realizará el cálculo de acumulados anteriores y a la fecha.

Las Estimaciones están basadas en la siguiente idea:

"DE QUÉ ANÁLISIS, QUE ELEMENTO (CONCEPTO) Y QUÉ CANTIDAD FUE REALIZADA."

El procedimiento a seguir para el cálculo de Estimaciones es el siguiente:

1. Desde la Barra de Herramientas, oprima el botón  correspondiente a la opción de Estimaciones.

2. Dentro del campo **Clave**, proporcione la Clave o Número de Estimación. Esta clave no puede ser mayor de 25 caracteres y puede contener cualquier signo del teclado estándar.
3. En el campo **Descripción**, proporcione la descripción de la Estimación. Esta descripción puede ser tan grande como lo desee y puede contener cualquier signo del teclado estándar.
4. Dentro del campo **Fecha de inicio**, proporcione la Fecha de Inicio del período de la Estimación.
5. Dentro del campo **Fecha de Terminación**, proporcione la Fecha de Terminación del período de la Estimación.

Para introducir las Fechas de Inicio y Terminación, existe un conjunto de teclas, que pueden ser utilizadas a lo largo de todo el sistema, a continuación se describen cada una de ellas:

Oprima la Tecla	Si desea
F11	Colocar la fecha actual
F12	Elegir la fecha dentro del Calendario del Sistema de Precios Unitarios
F9	Limpiar el campo de fecha

6. En la celda **Tipo Análisis**, proporcione el tipo al que pertenece el Análisis del cual se van a estimar sus conceptos.

Para seleccionar el Tipo de Análisis oprima la tecla correspondiente a la letra que se asocia al tipo de Análisis:

Tecla	Análisis
B	Básico
	Integrado
P	Partida
C	Capítulo
X	Extraordinario

7. En la celda **Clave Análisis**, proporcione la Clave del Análisis del cual se van a estimar sus Conceptos.



Si esta clave no se encuentra dada de alta dentro del Módulo de Análisis, el Sistema de Precios Unitarios le mandará un mensaje indicándole el error y lo regresará a esta celda para proporcionar nuevamente la Clave.

8. En la celda **Tipo Elemento**, proporcione el tipo al que pertenece el Concepto a estimar.

Para seleccionar el Tipo del Elemento a estimar, oprima la tecla correspondiente a la letra que se asocia al tipo de Insumo o Análisis:

Tecla	Tipo de Insumo o Análisis
M	Material
O	Obra de Mano
Q	Maquinaria y Equipo
T	Subcontrato
B	Básico
I	Integrado
P	Partida
C	Capítulo
X	Extraordinario

9. En la celda **Clave Elemento**, proporcione la Clave del Concepto a estimar.



Si este Concepto no pertenece al Análisis que usted especificó, el Sistema de Precios Unitarios le mandará un mensaje indicándole el error y lo regresará a esta celda para proporcionar nuevamente la Clave.

Al proporcionar la Clave del Concepto, se desplegará automáticamente la **Cantidad** que se presupuestó para este Concepto, el **Acumulado Anterior** donde se muestran los volúmenes de Obra del Concepto reportados en Estimaciones anteriores a ésta y el **Precio Unitario** del Concepto, el importe que aquí se muestra puede ser el Precio Unitario o el Costo Directo.

Si desea realizar sus Estimaciones a Precio Unitario, en el Panel de Control, especifique el Porcentaje de Indirectos a aplicar y seleccione que **NO** se desglose. Si por el contrario, sus Estimaciones van a realizarse a Costo Directo, seleccione a Costo Directo.

10. Dentro de la celda de **Cantidad Estimada**, proporcione el volumen ejecutado del Concepto en esta Estimación. Esta cantidad puede ser positiva o negativa.

Las cantidades positivas se refieren a cantidades de obra ejecutada.

El aplicar cantidades negativas, permite corregir cantidades que se estimaron de más en Estimaciones anteriores.

Una vez capturada esta cantidad se calculará el **Acumulado Total**, que se calcula como:

ACUMULADO ANTERIOR + CANTIDAD ESTIMADA

El **Importe** para el Concepto, calculado como:

CANTIDAD ESTIMADA * PRECIO UNITARIO


ó

CANTIDAD ESTIMADA * COSTO UNITARIO

11. Si desea capturar otro concepto desde esta Estimación repita los pasos del 8 al 10.

Usted puede observar que dentro del campo **Total de Toda la Estimación**, se mostrará el Total de la Estimación, que se calcula como:

SUMA DE LOS IMPORTES DE CADA UNO DE LOS CONCEPTOS ESTIMADOS

12. Una vez que los datos de la estimación son los correctos, oprima el botón  de la barra de herramientas del Sistema de Precios Unitarios correspondiente a grabar.

4.4. DOCUMENTACIÓN TÉCNICA

CÓMO EVITAR DESASTRES



La mayoría de la gente no piensa en la posibilidad de que ocurra un desastre, sin embargo, los desastres ocurren y son algo real. Por ello debe estar usted preparado para tratar de evitar al máximo su ocurrencia o para que en caso de que ocurran se puedan minimizar las pérdidas. Este concepto también aplica con su computadora, los datos que ésta contiene son de vital importancia para su empresa o para su uso personal, por ello debemos tomar medidas previsorias para mantenerlos en un estado óptimo.

Es necesario que cada Obra que usted elabore o actualice, dentro del Sistema de Precios Unitarios cuente con un respaldo, ésto para evitar que los datos contenidos en la Obra no se pierdan en caso de ocurrir algún problema en su computadora.

En esta sección se explica cómo proteger la información de la Obra para evitar un desastre y si ésto llegara a pasar, cómo recuperarla.

COPIADO DE OBRAS

Para evitar la pérdida de información de su Obra, usted puede tener un respaldo de ésta copiándola hacia otro directorio diferente.

Manual Técnico

Para realizar una copia de la Obra con la que se encuentra trabajando, usted puede utilizar el Programa de Intercambio de Información XIE que contiene el SISTEMA DE PRECIOS UNITARIOS.



Es necesario que usted realice el copiado de **TODA LA OBRA**, para ello seleccione el checkbox **Toda la Obra** que se encuentra dentro del diálogo de Selección de Opciones a Copiar.



El directorio que usted seleccione para crear la copia de la Obra debe ser diferente al directorio donde se encuentra la Obra que desea respaldar.

PROGRAMAS DE COMPRESIÓN DE DATOS

Para realizar sus respaldos usted también puede utilizar programas de compresión de datos, en esta sección comentaremos sobre algunos de los métodos más utilizados para respaldo de obras.

PKZIP y PKUNZIP son programas elaborados por la compañía PKWARE INC., el primero crea un archivo con extensión *.ZIP el cual contiene los archivos que usted desea comprimir, el segundo recupera los archivos contenidos en el *.ZIP, es decir restablece los archivos a su estado normal. Este método tiene una ventaja, los archivos que usted comprime ocupan menos espacio que los datos originales, sin embargo es necesario también respaldarlos con un programa de backup, la ventaja consiste en que, con los archivos ya comprimidos y almacenados en un archivo *.ZIP, los respaldos ocuparán menos discos flexibles.

Utilizar los programas BACKUP, MWBACKUP, MSBACKUP y RESTORE que se incluyen a partir de la versión 6.00 del Sistema Operativo MS-DOS, los tres primeros permiten

realizar los respaldos de su información hacia discos flexibles y el último recuperar los respaldos desde los discos flexibles. Los manuales del usuario que se distribuyen con el Sistema Operativo son una buena fuente de documentación sobre los procedimientos a seguir para realizar sus respaldos.



Siempre, los respaldos y las restauraciones deben realizarse con la misma versión de **MS-DOS**.

RECUPERACIÓN DE DATOS

Si por error usted borró los datos relacionados con su Obra, usted puede recuperarlos desde los respaldos de la siguiente manera:

Si utilizó el PKZIP:

Ejecute el programa PKUNZIP, direccionando los archivos que serán restablecidos hacia el directorio donde se encontraba su obra, por ejemplo, si el archivo que almacena el respaldo de los datos se llama RESPALDOS.ZIP, es necesario ejecutar el siguiente comando desde el Sistema Operativo:

```
PKUNZIP RESPALDOS.ZIP C:\INFONAVI\EDIFICIO
```

Si utilizó el programa BACKUP del Sistema Operativo:

Ejecute el programa RESTORE, direccionando los archivos que serán restablecidos hacia el directorio donde se encontraba su obra, por ejemplo, si la unidad de disco donde se encuentran mis respaldos es la unidad A:, es necesario ejecutar el siguiente comando desde el Sistema Operativo:

```
RESTORE A: C:\INFONAVI\EDIFICIO
```



Es muy importante que existan los directorios en la misma forma en que se realizó el respaldo, para que el RESTORE se ejecute adecuadamente.

CÓMO PREVENIR QUE SU EQUIPO SE INFECTE CON UN VIRUS

Para prevenir que su equipo se infecte con algunos de los Virus Informáticos que existen hoy en día, es necesario que antes de que introduzca cualquier disco externo, verifique que el disco no se encuentre infectado, para ello existen varios programas de verificación como el SCAN de MCAFEE ASSOCIATES, el DR SOLOMON'S ANTI-VIRUS TOOLKIT de S&S INTERNACIONAL o el MICROSOFT ANTI-VIRUS, incluido con la versión 6.0 del Sistema Operativo MSDOS. Generalmente cuando un virus ha atacado a su computadora ésta se comporta herráticamente, los programas se comportan de una forma no adecuada, pierden información y no realizan adecuadamente sus funciones. Para auxiliar en la búsqueda de un virus que se encuentre en su computadora mencionamos algunos de los principales Virus Informáticos, su comportamiento y cómo pueden afectar a los datos que se encuentran en su computadora.

MONKEY

Este virus es bastante infeccioso, afecta a los archivos con extensión COM, estos archivos se incrementan por 1063 bytes. El virus tiene una parte residente en memoria y un sistema de infección. Este virus no está encriptado y suprime archivos.

NATAS

Este virus afecta a los sectores de Boot, algunas Particiones y a los archivos con extensión EXE y COM, el virus tiene una parte residente en memoria y un sistema de

infección, no se encuentra encriptado y sobrescribe los archivos infectados. Existen 2 variantes: Natas .4744 y Natas .4746.

PING PONG

Este es uno de los virus más comunes, afecta a los sectores de Boot y algunas Particiones, el virus tiene una parte residente en memoria y un sistema de infección, no se encuentra encriptado y muestra un efecto de juego en la pantalla. Este virus puede ser conocido como Bouncing Ball o como Italian, tiene 10 posibles variantes: Ping Pong.b, Ping Pong.c, Ping Pong.d, Ping Pong.e, Ping Pong.f, Ping Pong.g, Ping Pong.h, Ping Pong.i, Ping Pong.Standard y Ping Pong.Typo.

STONED

Este virus es uno de los más comunes, afecta a los sectores de Boot y algunas Particiones, el virus tiene un sistema de infección residente en memoria, no se encuentra encriptado y cuando se activa despliega un mensaje. Este virus es conocido también como New Zeland o Marihuana. Existen 66 variantes.

VACUNAS

Si su equipo se encuentra infectado con alguno de los virus mencionados anteriormente, puede utilizar algunos de los programas que se encargan de eliminarlos como son el CLEAN de MCAFEE ASSOCIATES, el DR SOLOMON'S ANTI-VIRUS TOOLKIT o el MICROSOFT ANTI-VIRUS, incluido con la versión 6.0 del Sistema Operativo MSDOS.



Es muy importante que no acepte discos flexibles de los cuales desconozca su procedencia sin antes revisarlos utilizando los programas antes mencionados.

CÓMO MANEJAR WINDOWS ADECUADAMENTE

Otro factor importante para un buen resguardo de sus datos es el tener un adecuado funcionamiento de Windows, en los siguientes temas se dan algunas de las recomendaciones para el óptimo funcionamiento de Windows.

INHABILITAR EL DOBLE BUFFERING

El double buffering algunas veces es permitido por la instalación de Windows, pero esto alenta el acceso hacia su disco duro. La instalación de Windows modifica el archivo CONFIG.SYS o AUTOEXEC.BAT para habilitar esta característica, para deshabilitarla es necesario cargar con un editor de textos los archivos CONFIG.SYS y AUTOEXEC.BAT y modificar la línea donde aparece lo siguiente:

```
DEVICE      =      SMARTDRV.EXE /DOUBLE-BUFFER
              a:
REM DEVICE   =      SMARTDRV.EXE /DOUBLE-BUFFER
```

MÁXIMOS BENEFICIOS DE LOS RENDIMIENTOS DE MINI BACKUPS

En Windows 3.1, sus archivos *.INI y *.GRP son los más vulnerables. Si usted carga una nueva aplicación y su sistema ya no funciona adecuadamente, es probable que sea a causa de un problema con estos archivos. Automáticamente usted puede hacer copias backup de estos archivos críticos, para realizarlo utilice NOTEPAD o SYSEDIT para agregar las siguientes líneas a su archivo AUTOEXEC.BAT:

```
CD \WINDOWS
COPY *.INI *.IBK
```

```
COPY *.GRP *.GBK  
CD \
```

Este procedimiento se efectuará cada vez que usted encienda su computadora, ahorrándole trabajo y asegurándole un respaldo efectivo de sus archivos *.INI y *.GRP.

Si obtiene algo equivocado con Windows durante el día usted puede recuperar sin duda su configuración anterior, ¿Cómo?, en la línea de comandos teclee lo siguiente:

```
CD \WINDOWS  
COPY *.IBK *.INI  
COPY *.GBK *.GRP  
CD \
```

Entonces restablezca Windows, y usted podrá recuperar la configuración con la que inició el día.

PRACTICAR MEDICINA PREVENTIVA

Del mismo modo si usted sólo tiene problemas ocasionales con los Errores de Protección General (General Protection Faults), ésta es una buena ocasión para poner Dr.Watson (DRWATSON.EXE) en su grupo Start Up. Esta utilidad que viene junto con Windows, recolecta información del sistema cuando ocurre un General Protection Fault (Falla General de Protección) dicha información la almacena en un archivo llamado DRWATSON.LOG, el cual es creado en su directorio de Windows. Cada nuevo incidente que ocurra es añadido a los incidentes anteriores. La información en este archivo puede ser muy útil para los desarrolladores de Software cuando su programa ha generado una falla general en Windows. También es necesario tener algunas precauciones, el archivo DRWATSON.LOG puede crecer demasiado, DRWATSON le

avisará acerca de este problema, y usted tendrá que borrar el archivo manualmente. DRWATSON iniciará un nuevo rastreo una vez que no encuentre el archivo DRWATSON.LOG.

SALIDA CON PRECAUCIÓN

Es indispensable que usted siempre salga adecuadamente de Windows, si usted pasa por alto esta actividad puede causar problemas a los datos que tiene usted almacenados en el disco duro de su computadora. Al no salir adecuadamente de Windows, usted no está guardando información importante para el óptimo funcionamiento de Windows, esto repercutirá en las subsecuentes ejecuciones de Windows, lo que posiblemente se traduzca en una mala ejecución de aplicaciones o mal funcionamiento de su Windows.

HACER COPIAS DE RESPALDO DE SUS DISCOS ORIGINALES DE PROGRAMAS DE SOFTWARE

Usted está autorizado para crear una copia de respaldo de sus discos de instalación de sus aplicaciones. Es una buena práctica hacerlo, ya que si por alguna causa se dañan los programas de sus aplicaciones, tendrá los discos de donde pueden ser obtenidos nuevamente estos programas. Al hacer copias de respaldo de sus discos originales previene daños a los mismos y conserva así su valiosa inversión.

RECUPERAR VENTANAS PERDIDAS

Si su aplicación en Windows no se encuentra visible, usted puede invocar la lista de aplicaciones posicionando el apuntador del mouse en el área del desktop (escritorio de trabajo) y oprimiendo dos veces el botón izquierdo de éste, ó presionando las teclas: CTRL + ESC. En la lista seleccione la aplicación posicionando el apuntador del mouse y oprimiendo el botón izquierdo de éste, oprima el botón de Cascada o

Mosaico (Para la versión de Windows en inglés Cascade o Tile) según la forma en que desee que aparezcan las ventanas.

¿QUÉ HACER CUANDO WINDOWS NO FUNCIONA DESPUÉS DE UNA INSTALACIÓN?

Esto sucede porque la configuración de su Windows ha sido dañada al ser afectados los archivos WIN.INI y SYSTEM.INI. Siempre que efectúe una instalación de un programa nuevo, haga una copia de respaldo de los archivos antes mencionados. Si su Windows no funciona o no se comporta de manera similar después de la instalación copie nuevamente los archivos desde su respaldo, ésto hará que la nueva aplicación posiblemente no funcione, sin embargo usted ha detectado que la aplicación causa problemas a su Windows, lo que le dará a usted herramientas para reportarlo al proveedor del programa causante de los problemas.

HERRAMIENTAS PARA EL SISTEMA DE PRECIOS UNITARIOS

CÓMO IMPRIMIR UN REPORTE DE LOS EXISTENTES EN EL SISTEMA DE PRECIOS UNITARIOS PARA WINDOWS

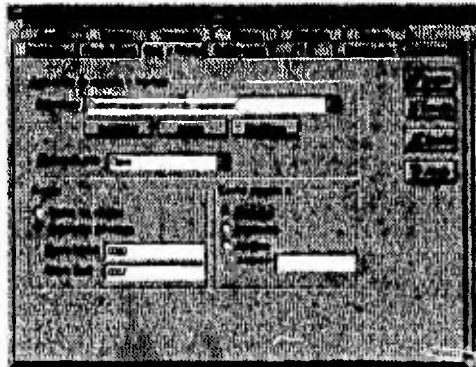


Este capítulo le mostrará cómo imprimir los reportes que ya se encuentran diseñados dentro del Sistema de Precios Unitarios para Windows, ésto a través del Panel de Reportes.


Si usted desea modificar los diseños de los reportes que incluye el Sistema de Precios Unitarios o crear uno nuevo, usted puede también invocar al Rep Estudio desde este Panel.

CÓMO SELECCIONAR UN REPORTE DENTRO DE SISTEMA DE PRECIOS UNITARIOS

En el Sistema de Precios Unitarios para Windows usted puede realizar la impresión de un reporte, ya sea de Insumos, Análisis, Presupuestos, de Formato Fijo (Factor de Salario Real), etc., desde cualquier módulo de Sistema de Precios Unitarios, para ello, es necesario invocar el diálogo de Selección de Opciones de Reportes, el cual le permite elegir el reporte a imprimir o invocar al Rep Estudio para diseñar un reporte nuevo.



A continuación se explicará el procedimiento a seguir para la selección del Tipo de Reporte a Imprimir, así como sus características de impresión.

1. Para invocar al diálogo de Selección de Opciones de Reportes, oprima el botón  dentro de la barra de herramientas del Sistema de Precios Unitarios.

O seleccione desde el menú de **Archivo** la opción de **Imprimir**.

2. Seleccione el separador correspondiente al módulo del Sistema de Precios Unitarios para Windows del cual desea imprimir el reporte (**Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos, Básicos, Integrados, Partidas, Capítulos, Extraordinarios, Presupuestos, Factor de Salario Real, etc.**).

3. Dentro del combobox **Reportes** seleccione el reporte que desea imprimir.

El Sistema de Precios Unitarios para Windows le proporciona ya una serie de reportes predefinidos, buscando ser los más comunes en el medio y entre Dependencias, los reportes que el Sistema de Precios Unitarios para Windows le ofrece son los siguientes:

Nombre del Reporte	Nombre del Archivo que contiene el reporte, (Posición Horizontal)	Nombre del Archivo que contiene el reporte, (Posición Vertical)
MATERIALES		
Catálogo de Materiales Concentrado	CTMTCNHC	CTMTCNVC
Catálogo de Materiales Detallado	CTMTDTHC	CTMTDTVVC
Catálogo de Materiales Análisis de Mercado	CTMTANHC	CTMTANVC
OBRA DE MANO		
Catálogo de Obra de Mano Concentrado	CTOBCNHC	CTOBCNVC

Catálogo de Obra de Mano Detallado	CTOBDTHC	CTOBDTVC
Catálogo de Obra de Mano Análisis de Mercado	CTOBANHC	CTOBANVC

MAQUINARIA Y EQUIPO

Catálogo de Maquinaria y Equipo Concentrado	CTEQCNHC	CTEQCNVC
Costo Horario Directo de Maquinaria Activo	CHEQACHC	CHEQACVC
Costo Horario Directo de Maquinaria Inactivo	CHEQINHC	CHEQINVC

SUBCONTRATOS

Catálogo de Subcontratos Concentrado	CTSBCNHC	CTSBCNVC
Catálogo de Subcontratos Detallado	CTSBOTHC	CTSBOTVC
Catálogo de Subcontratos Análisis de Mercado	CTSBANHC	CTSBANVC

**FACTOR DE SALARIO
REAL**

Factor de Salario Real	FSRHC	FSRVC
Nombre del Reporte	Nombre del Archivo que contiene el reporte, (Posición Horizontal)	Nombre del Archivo que contiene el reporte, (Posición Vertical)

**ANALISIS (PARA BASICOS,
INTEGRADOS, PARTIDAS,
CAPÍTULOS Y
EXTRAORDINARIOS)**

Catálogo de Precios Unitarios	CTPUHC	CTPUVC
Análisis de Precios Unitarios Concentrado de Precios Unitarios	ANPUHC	ANPUVC
Concentrado de Precios Unitarios sin costo	CNPUSHC	CNPUSVC

Análisis de Precios Unitarios con Alias y Leyendas	ANPUALHC	ANPUALVC
Concentrado de Precios Unitarios con Alias y Leyendas	CNPUALHC	CNPUALVC
Concentrado de Precios Unitarios sin costo con Alias y Leyendas	CNPUSAHC	CNPUSAVC

PRESUPUESTO (NIVEL DE BASICOS, INTEGRADOS, PARTIDAS, CAPITULOS Y EXTRAORDINARIOS)

Resumen del Presupuesto	RESUALHC	RESUALVC
Presupuesto	PRSUHC	PRSUVC
Presupuesto con Alias y Leyendas	PRSUALHC	PRSUALVC

Los archivos que identifican a los reportes dentro del Sistema de Precios Unitarios para Windows, contienen la extensión "SRE" (Script Report Estudio).

Usted puede modificar estos reportes de acuerdo a sus necesidades por medio del Rep Estudio.



Si por alguna causa dentro de la lista de este combobox n existen reportes, es necesario registrarlos, para ello, consulte e procedimiento a seguir dentro del siguiente tema.



Dentro de esta lista sólo pueden estar reportes que ya han sido previamente diseñados con el Rep Estudio.

CÓMO REGISTRAR UN REPORTE

Para registrar un reporte dentro del diálogo de Selección de Opciones de Reportes es necesario realizar los siguientes pasos:

1. Oprima el botón de **Registrar**, al oprimirlo aparecerá la ventana de Registro de Reportes.

2. Dentro de la ventana de Registro de Reportes en el campo de edición **Descripción** proporcione el texto con el cual se identificará el reporte, cuente usted con 65 caracteres. Es recomendable que coloque una descripción lo más apegada al reporte que se genera con ese script.

Un script es un archivo con extensión "SRE" (Script Report Estudio) que contiene las instrucciones con las que se puede generar un reporte.

Mientras que una plantilla (archivo con extensión "PRE") es de dónde parto para diseñar un reporte.

3. Dentro del campo de edición **Orientación Vertical**, teclee la trayectoria donde se encuentra el script que contiene el reporte con orientación vertical (los scripts de los reportes que fueron generados con el Rep Estudio, se encuentran almacenados en los archivos con extensión .SRE), si no recuerda en qué directorio se encuentra oprima el botón de **Buscar**.
4. Si usted oprimió el botón de **Buscar**, se abrirá una ventana dentro de la cual usted podrá realizar la búsqueda del archivo, una vez que usted se encuentra en esta ventana, dentro de la lista de **Directorios** seleccione el directorio en el cual se encuentran sus scripts de reportes.
5. Una vez seleccionado el directorio, dentro de la lista **Archivo Diseño de Reporte**, seleccione el nombre del archivo que contiene la plantilla deseada, una vez que usted seleccionó el archivo oprima el botón de **Aceptar**. Los nombres de los archivos siguen una convención que le recomendamos seguir, ésta es:

INICIALES NOMBRE DEL REPORTE+ ORIENTACIÓN + TAMAÑO DE HOJA

por ejemplo:

CTMTCNHC CATÁLOGO DE MATERIALES CONCENTRADO, HORIZONTAL, TAMAÑO CARTA

6. Si usted cuenta con la orientación horizontal de la misma plantilla, repita los pasos del 3 al 5 pero para el campo **Orientación Horizontal**.

CÓMO ORDENAR LA INFORMACIÓN DENTRO DEL REPORTE

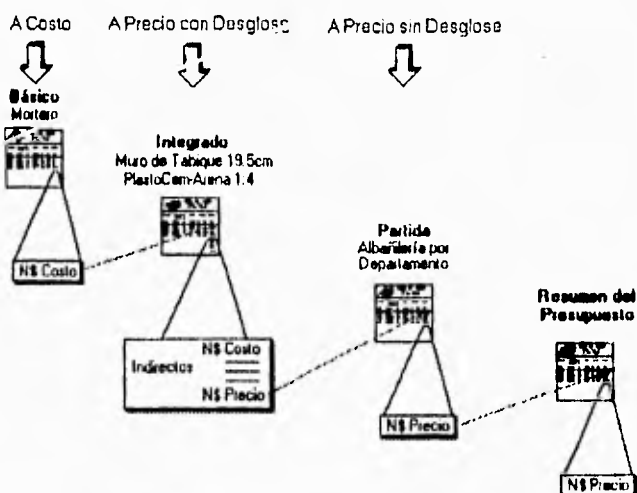
1. Dentro del combobox **Ordenado por**, seleccione bajo qué campo aparecerán ordenados los elementos de sus reportes, éstos pueden ser la **Clave**, el **Costo**, **Unidad** o la **Descripción** del Insumo o Análisis.

CÓMO ELEGIR EL TIPO DE FORMATO DE SOBRE COSTO EN UN REPORTE DE ANÁLISIS

Para generar un reporte de Análisis o un Presupuesto con Factores de Sobre Costo con el objeto de imprimirlos a Precio o a Costo, existen unos campos nuevos que son el **Tipo de Formato**, **Costo Unitario** y **Precio Unitario**, para especificar el Formato y el Tipo de Cálculo, es necesario realizar lo siguiente:

1. Dentro del combobox **Tipo de Formato**, seleccione el tipo de Formato de Sobre Costo de los ya existentes en el Sistema de Precios Unitarios o de los nuevos formatos que usted definió.
Existen 8 Formatos de Sobre Costo en el Sistema de Precios Unitarios, los cuales incluyen los Factores más comunes, por ejemplo, **INDIRECTOS**, **UTILIDAD**, **IVA**, **IVA SOBRE MATERIALES**, etc. y en diferentes agrupaciones y formas de cálculo.
2. Si por alguna causa necesita modificar o crear un nuevo formato, oprima el botón de **Editar**, ésto lo llevará al módulo de Factores de Sobre Costo del Sistema de Precios Unitarios para Windows.
3. Una vez que seleccionó el Tipo de Formato, seleccione si los reportes serán a **Costo Unitario** ó a **Precio Unitario** seleccionando el radiobutton relacionado con la opción deseada. Si eligió el radiobutton de **Precio Unitario**, usted puede elegir si el Precio Unitario se imprimirá desglosando los indirectos (checkbox **Con Desglose** seleccionado) o afectando el costo de los componentes del Análisis para obtener como total un Precio, la figura siguiente ilustra la

diferencia entre seleccionar los reportes a Costo, a Precio con Desglose y a Precio sin Desglose.



De la figura anterior podemos observar que generalmente los Básicos se imprimen a Costo, los Integrados se imprimen a Precio con Desglose y por último las Partidas, Capítulos y el Presupuesto a Precio sin Desglose.

COSTO UNITARIO

Si el reporte lo imprime a Costo Unitario, éste se imprimirá sin aplicarle ningún Factor de Sobre Costo, por lo que se mostrará a Costo Unitario el valor monetario de cada componente y del Análisis.

PRECIO UNITARIO CON DESGLOSE

Si el reporte lo elige a Precio Unitario con desglose en el reporte se imprimirá cada componente del Análisis mostrando su Costo Unitario, a su vez por cada grupo (Preliminares, Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos y Herramienta) se imprimirá un subtotal, cuya suma dará el Costo Unitario del Análisis. Al Costo Unitario obtenido, se le aplican los Indirectos, de acuerdo al formato de Sobre Costo elegido, calculando el Precio Unitario Total.

PRECIO UNITARIO SIN DESGLOSE

En el reporte a Precio Unitario sin desglose, se imprimirá cada componente del Análisis mostrando su Precio Unitario, el cual es obtenido después de aplicar el formato de Sobre Costo elegido al Costo Unitario del componente, a su vez por cada grupo (Preliminares, Materiales, Obra de Mano, Maquinaria y Equipo, Subcontratos y Herramienta) se imprime un subtotal, cuya suma dará el Precio Total del Análisis.

<u>Para:</u>	<u>Ver:</u>
Obtener información acerca de la edición de los Formatos de Sobre Costo	El tema de "Cómo definir los Formatos de Sobre Costos" dentro de este Manual o consulte el tema "Factores de Sobre Costo" dentro de la Ayuda en Línea de Sistema de Precios Unitarios.

SELECCIÓN DE RANGOS DE INFORMACIÓN

Dentro del diálogo de Selección de Opciones de Reportes en la sección de Rangos:

1. Elija si la presentación de su reporte incluirá todas las claves del Insumo o Análisis, para ello seleccione el radiobutton **Todas las Claves**.
2. Si decidió elegir un rango, seleccione el radiobutton **Selección de Rango** y dentro del campo **Clave Inicial**, proporcione la clave para indicar el rango inicial, en el campo **Clave Final**, proporcione la clave para indicar el rango final.

CÓMO ELEGIR A DÓNDE SERÁ ENVIADO EL REPORTE

Dentro del diálogo de Selección de Opciones de Reportes usted también puede seleccionar el dispositivo de salida del reporte, para ello, realice lo siguiente:

1. Seleccione a dónde quiere enviar su reporte, seleccionando el radiobutton relacionado con la opción deseada, éstas pueden ser a **Pantalla o Impresora**.
2. Una vez que todos los datos de su reporte son correctos oprima el botón de **Aceptar**.

Si seleccionó la opción de **Pantalla**, al oprimir este botón comenzará a desplegarse la información solicitada. Usted puede minimizar la ventana en la cual se encuentra generando el reporte y continuar con otras actividades dentro del Sistema de Precios Unitarios, el cual le enviará un mensaje de notificación cuando la generación del reporte termine.

Si seleccionó la opción de **Impresora**, al oprimir este botón aparecerá el siguiente diálogo en el cual se mostrará el tipo de impresora que se encuentra seleccionada, el número de páginas impresas y el número de registros impresos.



Si la impresora no se encuentra conectada a su computadora, o no contiene papel, se le presentará un mensaje notificándole lo anterior, si desea suspender la impresión del reporte, oprima el botón de **Cancelar**, de lo contrario, cheque la conexión de la impresora con su computadora, o suministre una nueva carga de papel y oprima el botón de **Reintentar**.



Si usted cuenta con impresora a color, el Sistema de Precio Unitarios para Windows imprimirá los reportes a color.

Dentro del **Manual Anexo** se encontrarán los ejemplos de algunos de los principales scripts que se incluyen dentro del Sistema de Precios Unitarios para Windows para la generación de sus reportes.

Para:

Ver:

Obtener información acerca del funcionamiento del Rep Estudio y de como diseñar sus propios reportes.

La ayuda en Línea y los Tips de Rep Estudio.

CONCLUSIONES

- ✓ Se logro realizar un programa orientado al usuario final a un costo accesible. Ya que se cuenta, con los elementos adecuados para que el usuario sea productivo desde el momento mismo de empezar a utilizar el programa. Esto se debe al énfasis que se le dio a la elaboración de ayudas, en las cuales se encuentra el sistema de tips y asistentes que el usuario tiene a su alcance, permitiendo incrementar de manera significativa su aprendizaje, teniendo como consecuencia directa la reducción de costos de capacitación.

- ✓ Gracias a la importancia que se le dio a los requerimientos de hardware, el sistema puede ser utilizado en computadoras de bajos recursos, como lo es una computadora 386 DX con 4 Megabytes de memoria.

- ✓ Consideramos que el programa realizado es competitivo dentro del mercado al que se dirige, ya que dentro del análisis se evaluaron otras alternativas de software existente y además las necesidades que éstas no cubrían. Por ejemplo, se cuenta con un generador de reportes que permite diseñar formatos personalizados.

Conclusiones

- ✓ En el desarrollo del producto se ha utilizado tecnología de punta (MS Visual C++, MFC, POO, MS-Windows), cuyos beneficios se verán reflejados en la satisfacción del usuario final, tales como:

Interface Amigable.

Tiempo de Respuesta Corto.

Opciones de Fácil Acceso.

- ✓ Consideramos que el programa como punto inicial es adecuado, pero es posible mejorarlo en diversos aspectos, principalmente en lo que se refiere a la visualización y generación de resultados, así como su interacción con otras aplicaciones.
- ✓ El programa cubre necesidades reales de la industria mexicana de la construcción, aportando una solución eficiente y generando tecnología nacional que pueda ser llevada a mercados externos con características similares.
- ✓ La empresa de hoy requiere de la integración de sus elementos humanos de forma eficiente, hoy en día el trabajo en equipo toma relevancia ya que la participación de diferentes conocimientos aporta mejores ideas logrando con esto una agilización en los procesos de desarrollo de productos de software.

- ✓ El uso de OLE en la aplicación permite una mejor interacción con otras aplicaciones especializadas en áreas administrativas, proporcionando así un modelo de automatización general de las constructoras, integrando a cada una de las áreas que participan en su operación.

BIBLIOGRAFÍA

1. Análisis y diseño orientado a objetos.
James Martin Prentice Hall

2. Concept of object - Oriented software
David M. Smith Mc Graw-Hill

3. Designing object oriented software
Rebeca Wirfs-Brock, Brian Wilkerson Prentice Hall

4. Object-Oriented Modelling and Design
James Rumbaugh, Michael Blaha Prentice Hall

5. Organización y métodos funcionales de la moderna empresa constructora
Manuel Sánchez Editores técnicos asociados.

6. Método de la Ruta Crítica y sus Aplicaciones a la Construcción
James M. Antill, Ronald W. WoodHead. LIMUSA.

7. Inside OLE 2.
Kraig Brockschmidt. Microsoft Press.

Bibliografía

8. Costo y Tiempo en edificación.

Suárez Salazar. 1988, 3a Edición LIMUSA

9. Costos de Construcción Pesada y Edificación. Tomo 3

Ing. Leopoldo Varela. Compuobras S.A. de C.V.

10. Manual de Costos y Precios en la Construcción

Ing. Carlos Suarez Salazar. 1989, 1a Edición LIMUSA.

11. Administración de Empresas Constructoras.

Ing. Carlos Suarez Salazar. 1989.

Cámara Nacional de la Industria de la Construcción.

12. Conceptos Financieros y de Costos en la Industria de la Construcción.

Daniel W. Halpin. Georgia Institute of Technology .

Ed. John Willey & Sons, Inc.

Versión Española

Rafael García Díaz. Ingeniero en Minas de la Universidad de Guanajuato, México.

Ed. LIMUSA México 1991.

13. Ingeniería de Costos y Administración de Proyectos.

Hira N. Anuja. Michael A. Walsn.

Ediciones Alfa Omega, S.A. de C.V.

México, D.F.

14. Administración Integral de Proyectos

Gustavo Arriagada S.

Ediciones del Colegio de Ingenieros de Chile.

15. Distributed Database Systems

DavidBell, Jane Grimson. Addison Wesley.

16. HandBook of Relational Database Design

Candace c. Fleming. Barbara von Halle. Addison Wesley.

17. Advanced Object-Oriented Analysis and Design

James Odell. Object Expo. The National Conference & Exposition.

18. Heuristics fo O-O Analys & Design.

Arthur Riel.

Bibliografía

19. Ingeniería del Software, un enfoque Práctico.

Royer S. Pressman. Segunda Edición.

McGrawHill

20. Developer Network, Development Library.

Microsoft Corporation.