



10
2ej.

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CONTADURIA
Y ADMINISTRACION

APLICACION DE LA INGENIERIA DE SOFTWARE EN
EL DESARROLLO DE UN SISTEMA DE CONTROL
DE OBRA

Seminario de Investigación Informática

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN INFORMATICA
P R E S E N T A

ROBERTO DE JESUS GONZALEZ DAVILA

Asesor : Dra. Judith Zubieta García

TESIS CON
FALLA DE ORIGEN

MEXICO, D.F., 1994.



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 está 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 MI ESPOSA, MARISOL,
POR SU AMOR, AMISTAD
Y APOYO CONSTANTE QUE
IMPULSAN MI VIDA.**

**A MIS PADRES
PORQUE GRACIAS A SU
APOYO, AMOR Y
DIRECCION LLEGO A ESTE
MOMENTO DE MI VIDA.**

**A MI ABUELA ELENA
CON ENORME CARIÑO
Y RESPETO.**

**A JUDITH
POR SU AMISTAD Y
SABIA DIRECCION PARA
LA CONCLUSION DE
ESTE TRABAJO.**

**A MIS MAESTROS Y
COMPAÑEROS
CON MUCHO AFECTO
POR LOS CONSEJOS Y
MOMENTOS QUE
COMPARTIMOS.**

INDICE

Introducción

1. Conceptos Básicos de la Ingeniería de Software	5
1.1. Construcción de Prototipos	7
1.2. Técnicas de Cuarta Generación (T4G)	8
1.3. Ciclo de Vida Clásico	8
1.3.1. Análisis de Requerimientos	10
1.3.2. Diseño	13
1.3.3. Codificación	18
1.3.4. Aseguramiento de la Calidad del Software	19
1.3.5. Prueba del Software	21
1.3.6. Mantenimiento	23
2. El Sistema de Control de Obras y las Estimaciones	29
2.1. Marco de Referencia	30
2.2. Descripción del Sistema Actual	30
2.3. Definición del Problema	36
3. Diseño de un Sistema Alternativo	39
3.1. Descripción del Sistema Propuesto	40
3.1.1. Especificación de requerimientos	41
3.2. Diseño	43
3.2.1. Carta Estructurada	43
3.2.2. Diagramas de Flujo de Datos (D.F.D.)	46
3.2.3. Diccionario de Datos	55
3.2.4. Descripción de archivos	57

4. Desarrollo del Sistema (SISCOB)	77
4.1. Instrumentación	78
4.1.1. Creación de Programas Fuente	78
4.1.2. Plan de Aseguramiento de la Calidad del Softwar	79
4.1.3. Interface con el Usuario	79
4.1.4. Seguridad del sistema	80
4.2. Prueba	80
4.2.1. Tipos de Pruebas	81
4.2.2. Aplicación de Pruebas y Resultados	82
4.3. Mantenimiento	83
4.3.1. Plan de Mantenimiento	85
4.3.2. Formas de Mantenimiento	85
5. Conclusiones	93
APENDICES	97
1 Glosario de términos	
2 Muestra de Código Fuente	
3 Muestra de los reportes que genera el sistema	
4 Formatos de aplicación de pruebas	

Bibliografía

INTRODUCCION

INTRODUCCION

La construcción de obras en México, desde las pequeñas hasta las de gran escala, es algo que se desarrolla frecuentemente, y éstas requieren que se lleve a cabo un control bastante rígido en cuanto al avance de la obra y los pagos que se realizan en la misma para evitar que se presenten problemas en su desarrollo. La dinámica moderna exige un ritmo más acelerado en la realización de actividades y funciones, es por esto que son cada vez más las áreas que se valen de los recursos que una computadora ofrece (rapidez, eficiencia, exactitud, economía, etc.) para realizar partes sustantivas del trabajo y controlar grandes volúmenes de información.

Dentro del área de Ingeniería Civil, existe software para llevar a cabo el control de obras (P.e. OPUS), pero la mayoría de estos paquetes "obligan" al usuario a ajustarse a las capacidades de un sistema determinado y trabajar de esta manera.

La informática es un campo que recientemente ha tenido un auge muy importante ya que son cada vez más las áreas en las que se utiliza y más gente la que requiere de herramientas computarizadas para facilitar sus actividades y realizarlas de una forma eficiente y rápida.

El presente trabajo consta de 5 capítulos que contienen una visión general de las bases para la creación de software, los antecedentes del control administrativo de obras, el diseño del sistema para controlar esta información, el desarrollo de un Sistema de Control de Obras (SISCOB) siguiendo todas las etapas utilizadas para su realización, y finalmente las conclusiones y comentarios que se desprenden de las pruebas aplicadas y de su implementación.

El primer capítulo, Conceptos Básicos de la Ingeniería de Software, presenta una recopilación de los aspectos más importantes a considerar para cualquier diseño de sistemas, tocando los puntos de Construcción de prototipos, Técnicas de cuarta Generación (T4G) y el Ciclo de Vida Clásico, los cuales nos proporcionan las bases y lineamientos a seguir para la construcción del Sistema de Control de Obras.

Posteriormente se presenta en el capítulo 2 toda la información de la forma en que se lleva a cabo el control administrativo de las obras y las estimaciones, así como la definición del problema que presenta el manejo de esta información y, por consiguiente, la búsqueda de una alternativa para su solución.

El Diseño de un Sistema Alternativo describe en el capítulo 3 el sistema propuesto, el diseño del mismo, incluyendo la Carta Estructurada, los Diagramas de flujo de Datos, el Diccionario de Datos y la descripción de archivos, información que nos servirá como guía para su creación y futuras referencias.

Una vez planteado el diseño del sistema, el Capítulo 4 Instrumentación del Sistema SISCOB, presenta el desarrollo del mismo, mostrando la creación de los programas fuente, así como la descripción

de los archivos que utiliza el sistema.

En este capítulo también se describe el plan de aseguramiento de la calidad del software, con los tipos de pruebas a realizar para comprobar su correcto funcionamiento, así como sus resultados y las formas de mantenimiento creadas para este efecto.

Finalmente, el capítulo de conclusiones recopila los resultados obtenidos en la implantación del sistema y una breve descripción de las bondades que ofrece en su operación, como también de sus límites.

CONCEPTOS BASICOS DE LA INGENIERIA DE SOFTWARE

CAPITULO 1

CONCEPTOS BASICOS DE LA INGENIERIA DE SOFTWARE

De acuerdo a Fritz Bauer (1969), se entiende por Ingeniería de Software "el establecimiento y uso de principios de ingeniería robustos, orientados a obtener software que sea viable y funcione eficientemente sobre máquinas reales." (Pressman R. 1988 ; 22)

Para Richard Fairley, la Ingeniería de Software "es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido." (Fairley R. 1989 ; 2)

La Ingeniería de Software se conforma de un conjunto de tres elementos claves - métodos, herramientas y procedimientos - que facilitan al analista controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma eficiente.

Los métodos de la Ingeniería de Software suministran el "cómo construir técnicamente el software". Abarcan un amplio espectro de tareas que incluyen : planificación y estimación de proyectos; análisis de los requerimientos del sistema y del software; diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos; codificación; prueba y mantenimiento. La Ingeniería de Software introduce frecuentemente una notación especial basada en diferentes metodologías para el análisis de requerimientos y un conjunto de criterios para asegurar la calidad del software.

Las herramientas de la Ingeniería de Software suministran un soporte automático o semiautomático para los métodos. Hoy, existen herramientas para soportar cada uno de los métodos mencionados anteriormente. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software, llamado ingeniería del software asistido por computadora (CASE: acrónimo en inglés de Computer-Aided Software Engineering).

Los procedimientos de la Ingeniería de Software son el elemento que une los métodos y herramientas y facilita un desarrollo racional y oportuno del software de computadora. Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las guías que facilitan a los analistas establecer su desarrollo.

El conjunto de pasos aplicados en estos tres elementos podría denominarse "enfoque" de la Ingeniería de Software. Un enfoque para la ingeniería del software se elige basándose en la naturaleza del proyecto y de la aplicación, los métodos y herramientas a usar y los controles y entregas requeridos.

El proceso de desarrollo del software contiene tres fases genéricas, independientemente del

enfoque de ingeniería elegido. Las tres fases, definición, desarrollo, y mantenimiento, se encuentran en todos los desarrollos de software, sin importar el área de aplicación, tamaño del proyecto o complejidad.

La fase de definición se enfoca sobre el qué. Esto es, durante la definición, el analista intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué interfaces han de establecerse, qué ligaduras de diseño existen y qué criterios de validación se necesitan para definir un sistema correcto.

La fase del desarrollo se enfoca sobre el cómo. Esto es, durante el desarrollo, el analista estudia cómo han de diseñarse las estructuras de datos y arquitectura del software, cómo han de implementarse los detalles procedimentales, cómo ha de trasladarse el diseño a un lenguaje de programación y cómo ha de realizarse la prueba.

La fase de mantenimiento se enfoca sobre el cambio que va asociado con una corrección de errores, adaptaciones requeridas por la evolución del entorno del software y modificaciones debidas a los cambios de los requerimientos del usuario para reforzar o aumentar el sistema.

A continuación describimos tres de estos enfoques, dando una visión somera de lo que comprende cada uno.

1.1. CONSTRUCCION DE PROTOTIPOS ¹

Normalmente un usuario definirá, con la asesoría u orientación de un experto en Informática, un conjunto de objetivos generales para el software, en este caso para Control de Obra y Estimaciones. Generalmente, no podrá identificar el detalle de los requerimientos detallados de entrada, de procesamiento o de salida. En otros casos, el analista puede no estar seguro de la eficiencia de un algoritmo, la adaptabilidad de un sistema operativo o la forma en que debe realizarse la interacción hombre-máquina. En ésta y otras situaciones similares, puede ser mejor método de Ingeniería de Software construir un prototipo.

La construcción del prototipo es un proceso que facilita al analista la creación de un modelo del software a construir. El modelo tomará una de las tres formas siguientes: un prototipo en papel que describa la interacción hombre-máquina de forma que facilite al usuario la comprensión de cómo se producirá tal o cual interacción; un prototipo que funcione, y que, dentro de su funcionamiento incluya subconjuntos del software deseado; o un programa existente que ejecute parte o toda la función deseada pero que tenga otras características que deben ser mejoradas en el nuevo trabajo en desarrollo.

La construcción de prototipos comienza con la recolección de los requerimientos. El analista y el usuario se reúnen y definen los objetivos globales para el software a desarrollar, identifican todos los requerimientos conocidos y perfilan las áreas donde será necesaria una mayor definición. Luego se

¹ Brooks, F., 1975, Pressman R., 1988

produce un "diseño rápido". El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario (por ejemplo, métodos de entrada y formatos de salida). El diseño rápido conduce a la construcción de un prototipo.

El prototipo es evaluado por el usuario y se utiliza para refinar los requerimientos del software a desarrollar. Se produce un proceso interactivo e iterativo en el que el prototipo es "afinado" para que satisfaga las necesidades del usuario, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer.

1.2. TECNICAS DE CUARTA GENERACION (T4G)

El término Técnicas de Cuarta Generación (T4G) abarca un amplio espectro de herramientas de software que tienen una cosa en común : facilitar el desarrollo del software. Luego, la herramienta genera automáticamente el código fuente, basándose en la especificación del analista. El enfoque T4G para la Ingeniería de Software se orienta hacia la habilidad de especificar software a un nivel que sea más próximo al lenguaje natural.

Actualmente, un entorno para el desarrollo del software que soporta el enfoque T4G incluye algunas o todas las siguientes herramientas: lenguajes no procedimentales para consulta a bases de datos, generación de informes, manipulación de datos, interacción y definición de pantallas y generación de código; capacidades gráficas de alto nivel; capacidad de hoja de cálculo. Actualmente no hay disponible un entorno T4G que pueda aplicarse con igual facilidad a todas las categorías de aplicaciones del software existentes.

1.3. EL CICLO DE VIDA CLASICO

Este modelo, también llamado "modelo en cascada", (Pressman R. 1988 ; 23), exige un enfoque sistemático y secuencial del desarrollo del software. Su aplicación se desarrolla a través de las actividades de análisis, de diseño, de codificación, de prueba y de mantenimiento. (Ver Fig. 1.1.)

Abarca las siguientes actividades :

INGENIERIA Y ANALISIS DEL SISTEMA. El trabajo de construcción de software comienza estableciendo los requerimientos de todos los elementos del sistema. Esta visión del sistema es esencial cuando el software debe interrelacionarse con otros elementos tales como el hardware, personas y bases de datos.

ANALISIS DE LOS REQUERIMIENTOS DEL SOFTWARE. El proceso de recolección de los requerimientos se centra e intensifica especialmente en el software. Para comprender la naturaleza de los programas que hay que construir, el analista debe comprender el dominio de la información del software a construir, así como la función, rendimiento e interfaces requeridas. Los requerimientos, tanto del sistema como del software, se documentan y revisan con el usuario.

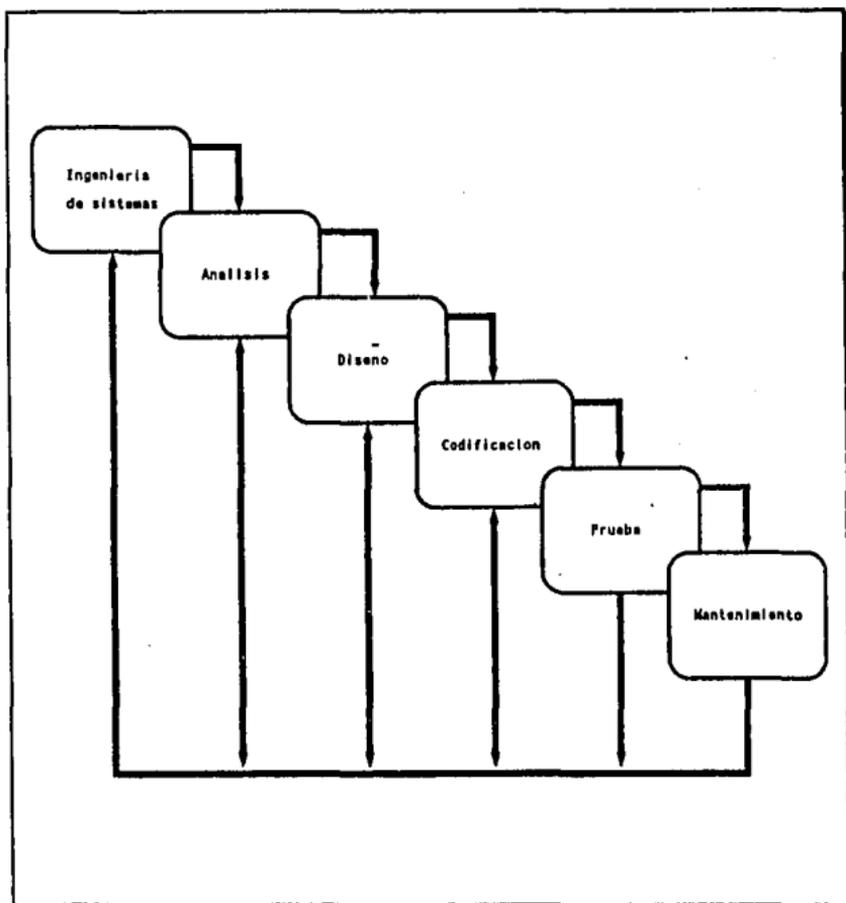


FIGURA 1.1. EL CICLO DE VIDA CLASICO

Fuente: Pressman R., Ingeniería del Software (1988; p. 23)

DISEÑO. El diseño del software es realmente un proceso multietápico que se enfoca en tres atributos del programa : estructura de datos, arquitectura del software y detalle procedimental. El proceso de diseño traduce los requerimientos en una representación del software que puede ser establecida de forma que se obtenga la calidad requerida antes de que comience la codificación. Así como se hizo con los requerimientos, el diseño también se documenta y forma parte de la configuración del software.

CODIFICACION. El diseño debe traducirse en una forma legible para la máquina. La codificación es la etapa donde se realiza esta tarea. Si el diseño se ejecuta de una manera detallada, ésta tarea puede realizarse mecánicamente.

PRUEBA. Una vez que se ha generado el código, comienza la prueba del programa. La prueba se debe enfocar hacia la lógica interna del software, asegurando que todos los procedimientos se hayan probado, y hacia las funciones externas, esto es, realizando pruebas para asegurar que la entrada definida producirá los resultados que realmente se requieren.

MANTENIMIENTO. Indudablemente el software sufrirá cambios una vez que haya sido entregado al usuario. Los cambios ocurrirán debido a que se encontraron posibilidades de mejora, a que el software debe adaptarse por cambios en el entorno externo (por ejemplo, un cambio solicitado debido a que se tiene un nuevo sistema operativo o dispositivo periférico), o a que el usuario requiere modificaciones en las funciones que desarrolla el software o necesita un mejor rendimiento en su operación.

Ahora que hemos repasado los aspectos más importantes de estos tres enfoques, para la creación del Sistema de Control de Obra y Estimaciones se eligió el enfoque del Ciclo de Vida Clásico, por considerarlo como el mejor de acuerdo a las características del proyecto. Este enfoque, además, nos permite llevar un control detallado de todas las etapas a seguir para la construcción del software y dejar una documentación muy completa para poder cumplir satisfactoriamente con el mantenimiento que se requiera. Por ello, describimos a continuación este enfoque con mayor detalle.

1.3.1. ANALISIS DE REQUERIMIENTOS

Para un buen desarrollo de software es esencial especificar detalladamente los requerimientos. Independientemente de lo bien diseñado o codificado que esté, un programa pobremente especificado decepcionará al usuario y hará fracasar el desarrollo.

La tarea del análisis de requerimientos es un proceso de descubrimiento y refinamiento. El ámbito del programa establecido inicialmente durante la etapa de ingeniería del sistema, debe ser refinado en detalle. Se analizan y asignan soluciones alternativas a los distintos elementos de los programas.

En esta especificación de requerimientos, tanto quien desarrolla el software (analista) como el usuario del mismo, juegan un papel muy importante de comunicación e interrelación. El usuario presenta el concepto de lo que quiere en detalles concretos, y el analista actúa como interrogador y

consultor a fin de poder proporcionar opciones para las soluciones.

Esta fase parece una tarea muy sencilla, pero debido a que el volumen de información manejada es muy grande, siempre existen problemas por mala interpretación o falta de información.

El análisis de requerimientos debe facilitar al analista la especificación de las funciones y comportamiento de los programas y la interface con otros módulos del sistema. Así mismo, debe dar al diseñador la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño procedimental.

Según Pressman (1988) el análisis de requerimientos puede dividirse en cuatro áreas : 1) Reconocimiento del problema; 2) Evaluación y Síntesis; 3) Especificación, y 4) Revisión.

Primeramente, es necesario conocer las especificaciones del sistema actual, ya sea manual o automatizado, y comprender el contexto en el cual opera. A continuación debe establecerse la comunicación necesaria para el análisis; esta comunicación debe establecerse con el personal usuario del software y con todas aquellas personas que puedan aportar información relacionada con el mismo. El objetivo del analista es reconocer los elementos básicos del programa, tal como lo percibe el usuario.

La evaluación del problema o problemas que se detectan en las pláticas entre el analista y el usuario, y la síntesis de la solución propuesta para construir el software, es el siguiente paso en el trabajo del análisis.

En la definición del problema o problemas a atacar, en el diseño del software, el analista debe poner mucha atención, porque en ocasiones el usuario presenta los síntomas como el problema en sí y es aquí donde se debe, a partir de estos síntomas, definir el problema real.

Para definir las características y atributos del software, se escribe una especificación de requerimientos formal, que contendrá una descripción de cada uno de ellos.

A continuación, el analista debe evaluar el flujo y estructura de la información, refinar en detalle todas las funciones del programa y establecer las características de la interface del sistema. Cada una de las tareas sirven para describir el problema de forma que pueda sintetizarse un enfoque o solución global.

El usuario no siempre puede precisar lo que quiere. El analista puede no estar seguro de que un enfoque en concreto sea el apropiado para realizar la función y comportamiento deseado. Por éstas y muchas otras razones, puede recurrirse a un enfoque alternativo, como lo es el llamado "construcción de prototipos", para el análisis de requerimientos.

Las tareas asociadas con el análisis y especificación de requerimientos existen para dar una representación del programa que pueda ser revisada y aprobada por el usuario. La especificación de requerimientos debe ser desarrollada conjuntamente entre el usuario y el analista para que los

resultados sean los mejores.

Una vez que se hayan descrito las funciones básicas, comportamiento, interface e información (por ejemplo, altas, bajas, modificaciones, manejo de información, datos de entrada, reportes a emitir, etc.), se especifican los criterios de validación. Estos criterios sirven como base para hacer la prueba durante el desarrollo de los programas.

Los documentos del análisis de requerimientos sirven como base para una revisión detallada que conducirá el usuario y el analista. La revisión de los requerimientos casi siempre produce modificaciones en alguna de las funciones que desarrollará el software, en el comportamiento requerido en alguno de los módulos y en la representación de la información o criterios de validación a utilizar.

Cada método de análisis tiene una única notación y punto de vista. Sin embargo, todos los métodos de análisis están relacionados por un conjunto de principios fundamentales, como los manejados por Pressman (1988), y estos son:

- 1.- El dominio de la información, así como todas las funciones a ejecutar en el mismo, deben ser representadas y comprendidas.
- 2.- El problema debe subdividirse de forma que se descubran los detalles de una manera progresiva (partición del problema global en subpartes para una mejor comprensión).
- 3.- Deben desarrollarse las representaciones lógicas y físicas del sistema.

Aplicando estos principios, el analista está en condiciones de enfocar el problema sistemáticamente. Se examina el dominio de la información de forma que comprenda el manejo que deberá darle en el sistema a desarrollar. La partición se aplica para reducir la complejidad del sistema en el cual se trabaja y poder desarrollarlo por partes.

- EL DOMINIO DE LA INFORMACION

El dominio de la información contiene tres concepciones diferentes de los datos que se procesan mediante programas de computadoras: 1) el flujo de información; 2) el contenido de la información, y 3) la estructura de la información. Para que el analista comprenda completamente el dominio de la información, deben considerarse cada una de estas tres partes, y de esta forma no escapará a su estudio de la información alguna parte importante.

El flujo de la información representa la manera en la que los datos se transforman conforme pasan a través de un sistema. La entrada se convierte en datos intermedios y más adelante se transforma en la salida. A lo largo de este camino de transformación (o caminos), pueden introducirse datos adicionales de almacenes de datos existentes (p. ej., un archivo de disco o un buffer de memoria). Las transformaciones que se aplican a los datos son funciones o subfunciones que un programa debe ejecutar. Los datos que se mueven entre dos transformaciones (funciones) definen la interface de cada función.

El contenido de la información representa los elementos individuales que componen otros elementos mayores de información. Por ejemplo, un registro de nómina es un elemento que está formado por un número de empleado, un sueldo, salarios anuales, impuestos anuales, etc. El contenido de un registro de nómina queda definido por los elementos contenidos en él. Para definir el procesamiento al que hay que someter un registro de nómina debemos primero comprender la información contenida en él, de aquí que para cualquier tipo de información a manejar, lo primero que debe hacerse es saber exactamente qué información se está manejando y de qué tipo es ésta.

La estructura de la información debe representar la organización lógica de los distintos elementos que componen los datos. ¿Deben los datos organizarse como una tabla n-dimensional o con una estructura en árbol jerárquico? Dentro del contexto de la estructura, ¿qué elementos de los datos están relacionados con otros elementos? ¿Está toda la información contenida dentro de una estructura simple de información o se utilizan distintas estructuras? ¿Cómo están relacionados los datos de una estructura de información con los de otra estructura?. Estas preguntas y otras son respondidas al establecer la estructura de la información y en los párrafos subsecuentes se tratará de darles respuesta.

- PARTICION

Normalmente los problemas son demasiado grandes y complejos para ser comprendidos como un todo. Por esta razón, tendemos a particionar (dividir) tales problemas en partes que puedan ser fácilmente comprendidas, y a establecer relaciones (interfaces) entre las partes, de forma que se realice la función global. Durante el análisis de requerimientos, las funciones a ejecutar y el dominio de la información del software pueden ser particionados.

En esencia, la partición descompone un problema en sus partes constituyentes. Conceptualmente, establecemos una representación jerárquica de la función o información y luego partimos el elemento superior mediante: 1) un incremento en el nivel de detalle, moviéndonos verticalmente en la jerarquía, o 2) descomponiendo funcionalmente el problema, moviéndonos horizontalmente en la jerarquía.

1.3.2. DISEÑO.

Esta fase comprende tres pasos: diseño, generación del código (manual o automáticamente) y prueba. El flujo de información se muestra en la Figura 1.2. Los requerimientos del programa, manifestados mediante el dominio de la información, los requerimientos funcionales y de comportamiento, alimentan el paso del diseño. Usando una de las distintas metodologías de diseño, se realiza el diseño de datos, el diseño arquitectónico y el diseño procedimental. El diseño de datos se enfoca hacia la definición de la estructura de datos. El diseño arquitectónico define las relaciones entre los principales elementos estructurales del programa (nos proporciona una visión de lo que será el sistema). El diseño procedimental transforma los elementos estructurales en una descripción de las funciones del software; es decir, aquí se diseñan los algoritmos que darán funcionamiento a los programas que se requieran.

La calidad de un programa para computadora se asienta en esta fase, y a través de él es la

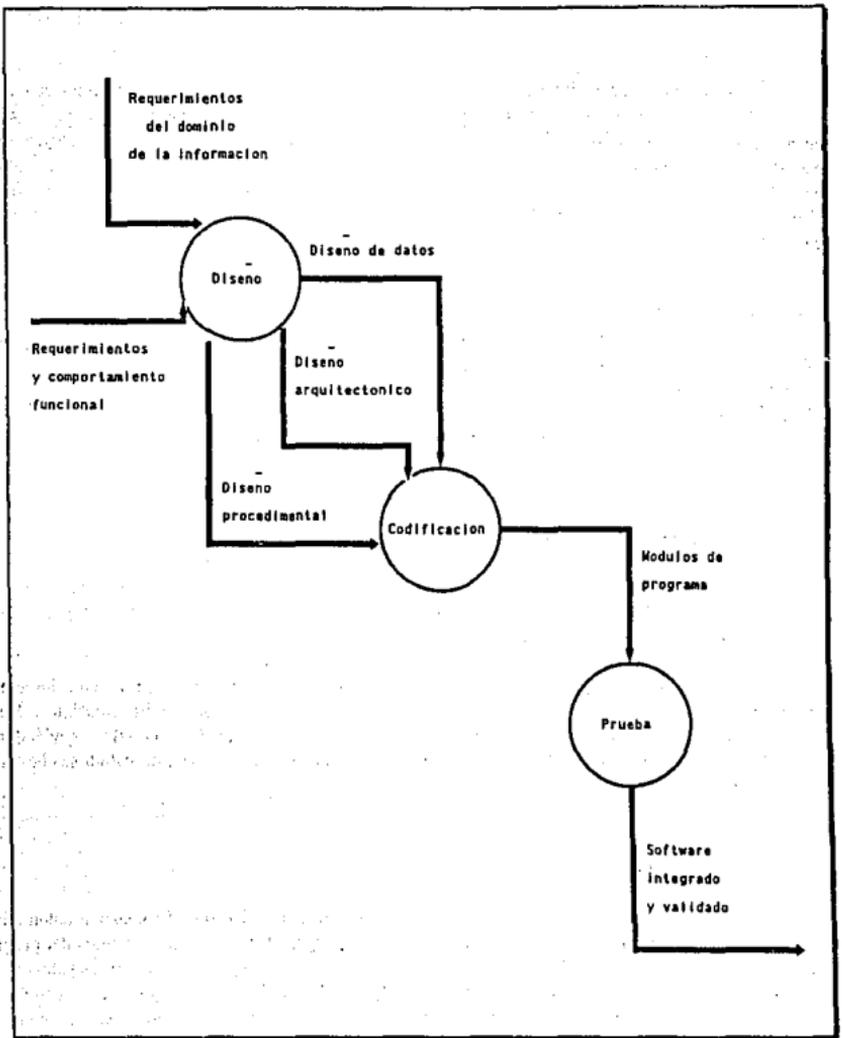


FIGURA 1.2. LA FASE DE DESARROLLO

Fuente: Pressman R., Ingeniería del Software (1988: p.239)

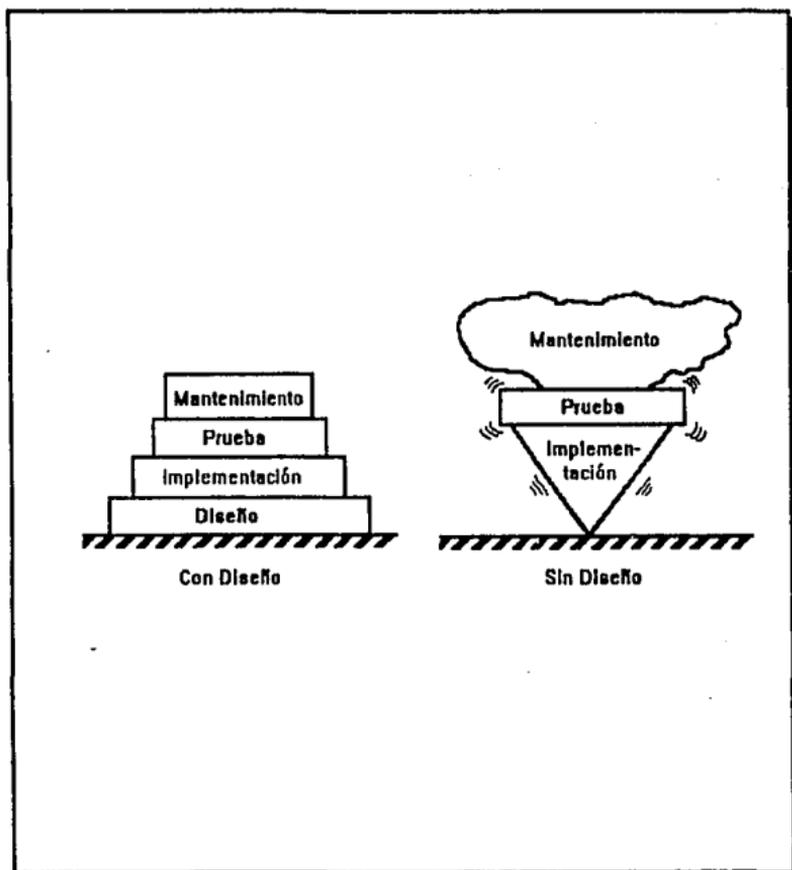


Figura 1.3. IMPORTANCIA DEL DISEÑO

Fuente: Pressman R., Ingeniería del Software (1988; p.239)

única forma mediante la que podemos traducir con precisión los requerimientos del usuario en un producto o sistema acabado.

El diseño de programas sirve como base para todos los pasos del desarrollo y fase de mantenimiento que siguen. Sin el diseño, se corre el riesgo de construir un sistema inestable, que fallará cuando se realicen pequeños cambios.

De aquí que esta fase sea de gran importancia pues si se realiza un diseño cuidadoso y completo las otras fases tendrán una base muy sólida. (Ver Figura 1.3.)

Dentro del diseño de software existen diferentes técnicas para hacerlo, como son la de Niklaus Wirth (1971) llamada refinamiento sucesivo o como las de Warnier-Orr (1974) y Jackson (1975), las cuales están pensadas para representar la estructura de un programa en su notación especial.

Estas técnicas se utilizan dentro de lo que se llama "Arquitectura del software" (Pressman R. 1988) y alude a dos características importantes de un programa de computadora: 1) la estructura del programa, y 2) la estructura de los datos. La arquitectura de software permite que un sistema grande sea definido en unidades más pequeñas y manejables con una clara definición de las relaciones entre las diferentes partes del sistema definidas durante el análisis de requerimientos. La evolución del software y la estructura de datos comienza con una definición del problema. La solución ocurre cuando cada parte del problema se resuelve mediante uno o más elementos de software (programas o módulos). Con base en lo anterior, podemos ver que la Arquitectura de Software es una parte de la Ingeniería de Software que ayuda al analista a efectuar la división del problema principal y poderlos manejar más fácilmente. Esto queda más claro si observamos la Figura 1.4.

- ESTRUCTURA DEL PROGRAMA Y DE DATOS

La estructura del programa representa la organización (frecuentemente jerárquica) de los componentes del programa. No representa aspectos procedimentales del software, tales como secuencia de procesos, ocurrencia/orden de decisiones o repetición de operaciones.

Para representar una estructura de programa por lo general se utiliza un diagrama de árbol, frecuentemente llamado Diagrama de Estructura, o Carta Estructurada.

Debido a que la estructura de la información afectará invariablemente el diseño procedimental final, la estructura de datos es tan importante como la estructura de programas en la representación de la arquitectura del software.

La estructura de datos dicta la organización, métodos de acceso y alternativas de procesamiento para la información.

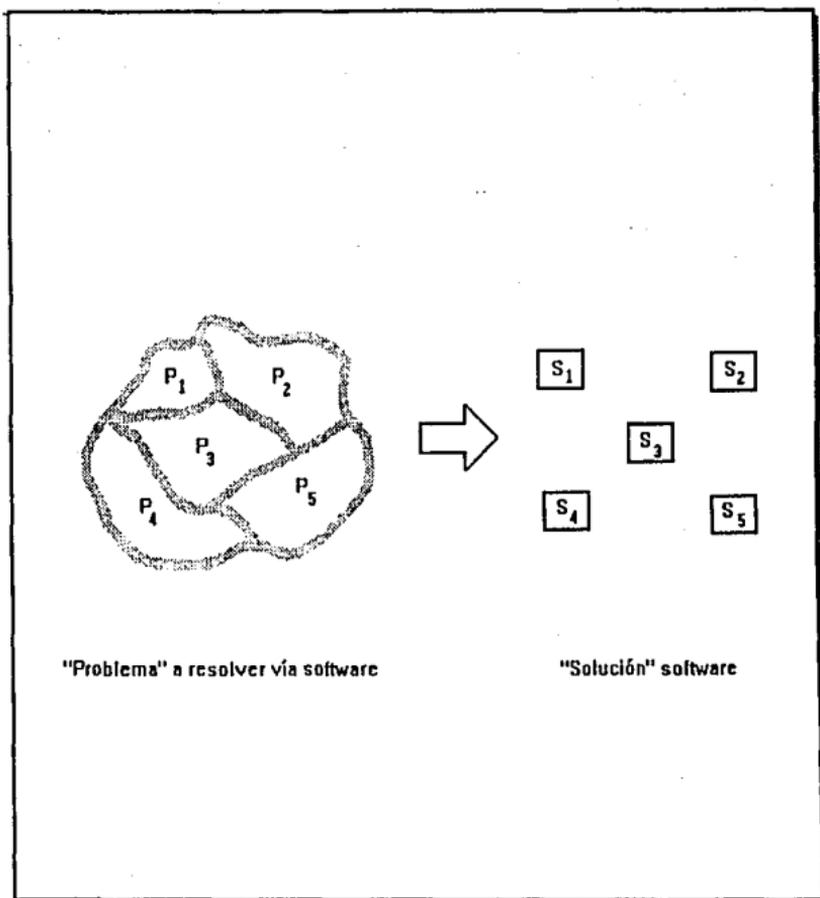


Figura 1.4. EVOLUCION DEL SOFTWARE

Fuente: Pressman R., Ingeniería del Software (1988; p.243)

- MODULARIDAD

La arquitectura de software implica modularidad; esto es, el software se divide en elementos con nombres y direcciones separados, llamados módulos, que se integran para satisfacer los requerimientos de un problema.

Se ha establecido que "la modularidad es el atributo más sencillo del software que permite a un programa ser manejable intelectualmente" (Myers, 1978 Pressman R. 1988 ; 247).

Dentro de la modularidad, se utiliza el concepto de independencia funcional, del cual dice Pressman (1988 ; 256), que "se adquiere desarrollando módulos con una clara función y una 'aversión' a una excesiva interacción con otros módulos". Este concepto quedará más claro al explicar dos criterios cualitativos que se utilizan para medir la independencia funcional : la Cohesión y el Acoplamiento.

La cohesión se halla presente en un módulo, cuando "ejecuta una tarea sencilla en un procedimiento y requiere poca interacción con procedimientos que se ejecutan en otras partes de un programa. Dicho sencillamente, un módulo coherente sólo debe hacer (idealmente) una cosa." (Pressman R. 1988;256)

Con respecto a los tipos de cohesión, Pressman (1988 ; 258) dice : "cuando los elementos de un módulo están y deben ejecutarse en un orden específico, existe cohesión procedimental. Cuando todos los elementos de procesamiento, se concentran sobre una área de una estructura de datos, se presenta una cohesión de comunicación."

El acoplamiento es una medida de la interconexión entre módulos en una estructura de un programa. Este depende de la complejidad de la interface entre módulos, del punto en el que se hace una entrada o referencia a un módulo y de los datos que pasan a través de la interface.

1.3.3. CODIFICACION

La etapa de la codificación traduce una representación del software dada por un diseño detallado a una realización en un lenguaje de programación llamado código fuente. El proceso de traducción continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto dependiente de la máquina. Más tarde, la salida del compilador es traducida a código máquina.

El paso inicial de traducción - del diseño detallado al lenguaje de programación - es un punto fundamental dentro del contexto de la Ingeniería de Software. En el proceso de traducción pueden presentarse errores de muchas formas distintas. La interpretación equivocada de las especificaciones del diseño detallado puede conducir a un código fuente erróneo. La complejidad o las restricciones de un lenguaje de programación pueden conducir a un código fuente muy "problemático" que resulte difícil de probar y de mantener.

Las características del lenguaje tienen un impacto directo sobre la calidad y la eficiencia de la traducción, es por esto que su elección debe ser muy cuidadosa.

El lenguaje de programación se debe escoger de acuerdo a las características del proyecto de software a realizar, como son los tipos de procesos o las estructuras de datos a utilizar. En ocasiones, esta elección no depende totalmente de los motivos mencionados anteriormente, sino que depende de la disponibilidad de herramientas de programación con las que se cuente.

También en la elección del lenguaje de programación deben tomarse en cuenta ciertos criterios para evaluar sus características. De acuerdo con Pressman (1988 ; 453) éstas son : 1) área de aplicación general; 2) complejidad algorítmica y de rendimiento; 3) entorno en que se ejecutará el software; 4) consideraciones de rendimiento, tales como velocidad de procesamiento, funciones soportadas por el lenguaje, etc.; 5) complejidad de las estructuras de datos; 6) conocimiento del lenguaje por la o las personas que desarrollarán el software, y 7) disponibilidad de un buen compilador.

1.3.4. ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

En el desarrollo de cualquier software, el que cuente con calidad es una meta muy importante. Pressman (1988 ; 480) define la calidad como : "concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo completamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente".

Si observamos la definición anterior, podemos notar tres puntos importantes. 1) al desarrollar cualquier software, el analista debe tratar de apearse a los requerimientos iniciales, tanto funcionales como de rendimientos; 2) los estándares especificados definen una guía a seguir para crear el software y por tanto debemos tratar de respetarlos y 3) existe un conjunto de requerimientos implícitos que a menudo no se mencionan (p.ej., el deseo de un buen mantenimiento). Si el software se ajusta a sus requerimientos explícitos pero falla en alcanzar los requerimientos implícitos, la calidad del software queda en entredicho.

Existen factores que determinan la calidad del software. McCall et al (1977) los clasifican en tres aspectos importantes de un producto de software: sus características operacionales (p.ej., tiempos de respuesta, exactitud de resultados arrojados, etc.), su capacidad de soportar los cambios y su adaptabilidad a nuevos entornos. (Ver Figura 1.5.)

Refiriéndonos a los factores indicados en la Figura 1.5., McCall proporciona las siguientes descripciones :

- **Corrección.** Es el grado en que un programa satisface sus especificaciones y consigue los objetivos de la misión encomendada por el usuario. Es decir que realiza la función para la que fue diseñado.
- **Fiabilidad.** Es el grado con el que se puede esperar que un programa lleve a cabo sus funciones con

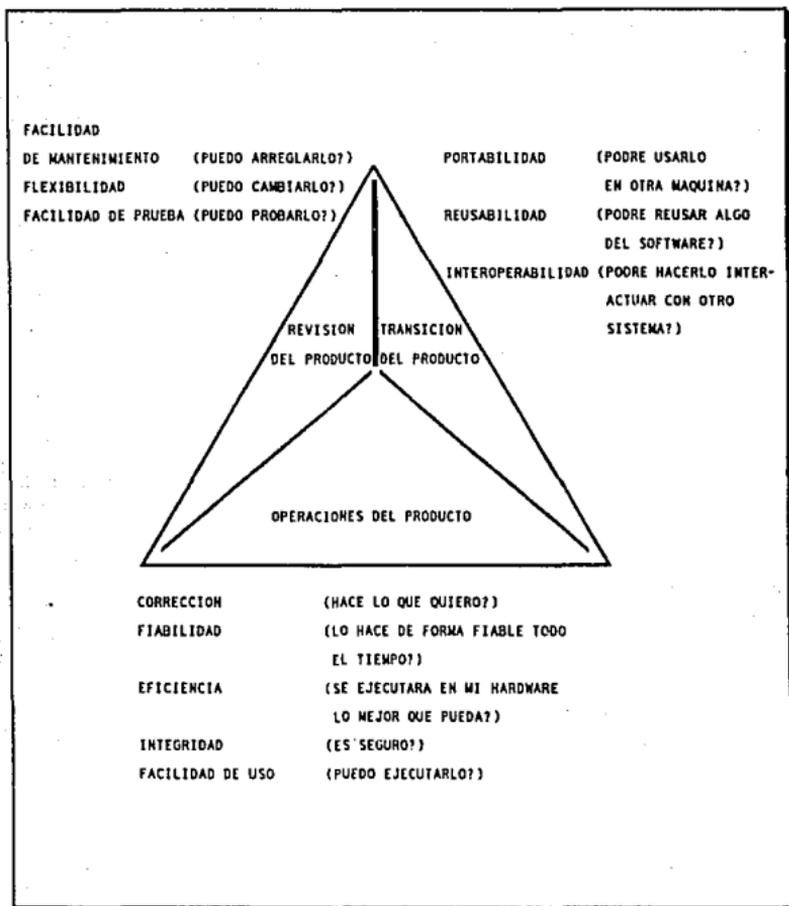


FIGURA 1.5. FACTORES DE CALIDAD DEL SOFTWARE DE McCALL

Fuente: Pressman R., Ingenieria del Software (1988; p.481)

la precisión requerida, y esto se evalúa a través de pruebas en las que ya se conoce el resultado correcto y se compara con el arrojado por el sistema.

- **Eficiencia.** Es la cantidad de recursos de computadora (p. ej., memoria, espacio en disco, etc.) y de código requeridos por un programa para llevar a cabo sus funciones.
- **Integridad.** Es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.
- **Facilidad de uso.** Es el esfuerzo requerido para aprender a manejar un programa, trabajar con él y preparar los datos que lo alimentan e interpretar su salida, y esto sólo lo podremos evaluar a través del propio usuario.
- **Facilidad de mantenimiento.** Es el atributo que debe tener un programa para poder localizar en él las diferentes funciones que lo componen y corregir algún error, modificarlo para su mejor funcionamiento o para cambios en los procesos que realiza.
- **Flexibilidad.** Es la posibilidad de poder modificar cualquier programa operativo.
- **Facilidad de prueba.** Es el esfuerzo requerido para probar un programa de forma que se asegure que realiza su función requerida.
- **Portabilidad.** Es la capacidad de un programa para ser transferido desde un hardware y/o un entorno de sistemas de software a otro.
- **Reusabilidad.** Es el grado en que un programa (o partes de un programa) se puede reusar en otras aplicaciones.
- **Facilidad de interoperación.** Es la posibilidad para acoplar un sistema a otro cualquiera.

La calidad del software debe estar incluida en el diseño, ya que si se siguen conceptos como los de modularidad, independencia funcional y se toman en cuenta los factores mencionados anteriormente por McCall estaremos encaminándonos a un software de calidad; de acuerdo a esto, podemos decir que la calidad no es algo posterior al diseño e instrumentación de un software, sino que debe ser tomado en cuenta en cada una de las fases de su elaboración.

1.3.5. PRUEBA DEL SOFTWARE

La prueba de software es un elemento crítico para la garantía de calidad del software y representa un último repaso de las especificaciones, del diseño y de la codificación.

Glen Myers (1979) establece una serie de reglas que sirven como objetivos de prueba:

- 1) La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- 2) Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- 3) Una prueba tiene éxito si descubre un error no detectado hasta entonces.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Cualquier producto de ingeniería (y sobre todo el software) puede ser probado en dos formas: 1) conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa, y 2) conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que "todas las piezas encajen": es decir, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. La primera aproximación de prueba se denomina prueba de la caja negra y la segunda prueba de la caja blanca.

Cuando se considera el desarrollo de software, la prueba de la caja negra se refiere a las pruebas que se llevan a cabo sobre la interface del software. Esto es, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa (P. ej., archivos de datos) se mantiene.

- PRUEBA DE LA CAJA BLANCA

La prueba de la caja blanca del software se basa en un minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o ciclos. Se puede examinar el "estado del programa" en varios puntos para determinar si el estado real coincide con el esperado o afirmado. Esta prueba es un método de diseño de casos de evaluación y análisis que usa la estructura de control del diseño procedimental para derivar los casos de prueba. Mediante los métodos de prueba de la caja blanca, el ingeniero de software (y en algunas ocasiones con ayuda del usuario) puede derivar casos de prueba que: 1) garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo; 2) se ejerciten todas las decisiones lógicas en sus caras verdadera y falsa; 3) se ejecuten todos los ciclos en sus límites y con sus límites operacionales, y 4) se ejerciten las estructuras de datos internas para asegurar su validez.

A primera vista parecería que una prueba de la caja blanca muy profunda nos llevaría a tener programas cien por ciento correctos. Todo lo que tenemos que hacer es definir todos los caminos lógicos, desarrollar casos de prueba que los ejerciten y evaluar los resultados. Desgraciadamente, una prueba exhaustiva presenta ciertos problemas lógicos. Incluso para pequeños programas, el número de caminos lógicos posibles puede ser enorme.

La prueba de la caja blanca, sin embargo, no se debe desechar como impracticable. Se puede

elegir y ejercitar una serie de importantes caminos lógicos. Se pueden comprobar las estructuras de datos más importantes para ver su validez. Los atributos de la prueba de la caja blanca así como de la caja negra se pueden combinar para llegar a una aproximación que valide la interface del software y asegure selectivamente que el funcionamiento interno del software es correcto.

- PRUEBA DE LA CAJA NEGRA

La prueba de la caja negra es un método que se centra en los requerimientos funcionales del software, es decir, la prueba de la caja negra permite al analista derivar conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa.

La prueba de la caja negra intenta encontrar errores en las siguientes categorías: 1) funciones incorrectas o ausentes; 2) errores de interface; 3) errores en estructuras de datos o en accesos a bases de datos externas; 4) errores de rendimiento, y 5) errores de inicialización y terminación.

A diferencia de la prueba de la caja blanca, que se lleva a cabo previamente, la prueba de la caja negra tiende a ser aplicada durante fases posteriores de prueba, ya que la de la caja negra intencionadamente ignora la estructura de control y concentra su atención en el dominio de la información.

1.3.6. MANTENIMIENTO

A menudo, el mantenimiento ha sido caracterizado como un "iceberg": esperamos que lo que está inmediatamente visible sea todo lo que hay. Realmente sabemos que bajo la superficie se esconde una gran cantidad de problemas potenciales.

En el área de desarrollo de sistemas siempre se requiere de mantenimiento, y esto es porque los programas de computadora siempre están cambiando. Hay que solucionar errores, añadir mejoras y llevar a cabo optimizaciones. Además de los problemas que requieren la realización de cambios, el hecho mismo del cambio lleva a problemas adicionales.

El mantenimiento del software es mucho más que la corrección de errores. Podemos definir el mantenimiento, describiendo cuatro actividades que se llevan a cabo tras lanzar un programa.

La primera actividad se da debido a que no es razonable asumir que la prueba del software haya descubierto todos los errores latentes de un gran sistema de software. Durante el uso de cualquier gran programa, se encontrarán errores, siendo informado el equipo de desarrollo. El proceso que incluye el diagnóstico y la corrección de uno o más errores se denomina mantenimiento correctivo.

La segunda actividad que contribuye a la definición de mantenimiento se da debido al rápido cambio inherente a todo aspecto de la informática. Se presentan cambios en el hardware, aparecen nuevos sistemas operativos o nuevas versiones de los antiguos y frecuentemente se mejoran o modifican los equipos periféricos. Por otro lado, la vida útil del software de aplicación puede difícilmente sobrepasar los diez años, haciéndose obsoleto para el entorno del sistema para el que fue

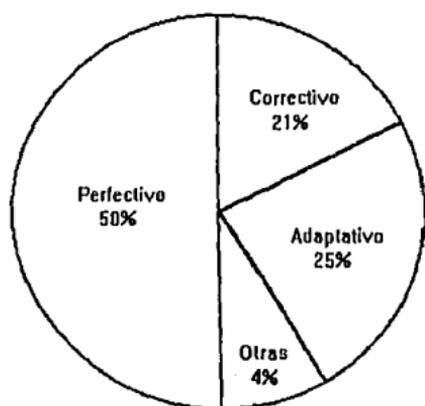


Figura 1.6. DISTRIBUCION DE LAS ACTIVIDADES DE MANTENIMIENTO

Fuente: Pressman R., Ingeniería del Software (1980; p.506)

originalmente desarrollado. Por tanto, el mantenimiento adaptativo - una actividad que modifica el software para que interactúe adecuadamente con su entorno cambiante - es tanto necesaria como corriente.

La tercera actividad que se puede aplicar a la definición del mantenimiento se da cuando un software tiene éxito. A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades, sobre modificaciones de funciones ya existentes y sobre mejoras en general. Para satisfacer estas peticiones, se lleva a cabo el **mantenimiento perfectivo**. Esta actividad contabiliza la mayor cantidad de esfuerzo gastado en el mantenimiento de software.

La cuarta actividad de mantenimiento se da para proporcionar una base mejor para futuras mejoras, se denomina **mantenimiento preventivo** y esta actividad es relativamente rara en el mundo del software.

Para adaptar o perfeccionar debemos determinar nuevos requerimientos, rediseñar, generar código y probar el software existente. Tradicionalmente, tales tareas han sido denominadas en conjunto como mantenimiento.

De hecho, aproximadamente la mitad de todo el mantenimiento de software es perfectivo, tal y como se ilustra en la Figura 1.6. Los porcentajes que se muestran, se basan en el estudio de 487 organizaciones de desarrollo de software llevado a cabo por Lientz y Swason. (Lientz B. 1980) Estos datos dan una mejor información sobre las proporciones ocultas del "iceberg" del mantenimiento.

- PROBLEMAS DEL MANTENIMIENTO

La mayoría de los problemas asociados con el mantenimiento de software se debe a las deficiencias de la forma en que el software ha sido definido y desarrollado. La falta de control y disciplina en las dos primeras fases del proceso de ingeniería del software casi siempre se traduce en problemas para la última fase.

Entre los muchos problemas clásicos asociados con el mantenimiento de software se encuentran los siguientes :

- A menudo es excepcionalmente difícil comprender un programa "ajeno". A medida que existen menos elementos de configuración del software, mayor es la dificultad. Si sólo existe el código indocumentado, es de esperar que aparezcan serios problemas.
- Ese personaje "ajeno" a menudo no se encuentra alrededor para poder explicarse. La movilidad entre los profesionales del software es actualmente muy grande. No podemos esperar una explicación personal del software por el que lo ha desarrollado una vez que se requiere el mantenimiento.
- No existe una documentación apropiada o está mal preparada. Un primer paso es el reconocimiento de que el software debe ser documentado, pero para que la documentación sea de algún valor debe ser comprensible y consistente con el código fuente.

- La mayoría del software no ha sido diseñado previendo el cambio. A menos que el método de diseño lo prevea mediante conceptos tales como independencia funcional o clases de objetos, las modificaciones del software serán difíciles y propensas a errores.
- El mantenimiento no se ve como un trabajo atractivo. Muchas de las causas de esto vienen dadas por el alto nivel de frustración asociado con el trabajo de mantenimiento.

Todas las peticiones de mantenimiento de software deben ser presentadas de una forma estandarizada. Normalmente el equipo de desarrollo de software genera un formulario de petición de mantenimiento, a veces denominado informe de problemas del software, que ha de ser llenado por el usuario que desea la actividad de mantenimiento. Si se encuentra un error, se debe incluir una descripción completa de las circunstancias que llevaron al error (incluyendo datos de entrada, listados y otro material de soporte). Para peticiones de mantenimiento adaptativo o perfectivo, se ha de remitir una breve especificación de cambios (una especificación de requerimientos abreviada).

Este informe es un documento generado exteriormente que se usa como base para la planificación de las tareas de mantenimiento. Internamente, la organización de software desarrolla un informe de cambios del software, indicando: 1) la magnitud del esfuerzo requerido para satisfacer la petición de mantenimiento; 2) la naturaleza de las modificaciones requeridas; 3) la prioridad de la petición, y 4) otros datos sobre las modificaciones.

Históricamente, el registro de información de todas las fases del proceso de ingeniería de software ha sido inadecuado. El registro de información de mantenimiento ha sido prácticamente inexistente. Por esta razón, frecuentemente no somos capaces de asegurar la efectividad de las técnicas de mantenimiento.

El primer problema que aparece con el registro de información es el de comprender qué datos merece la pena registrar. Swason (1976) presenta una lista :

- 1.- Identificación del programa
- 2.- Número de sentencias fuente
- 3.- Número de instrucciones en código máquina
- 4.- Lenguaje de programación usado
- 5.- Fecha de instalación del programa
- 6.- Número de ejecuciones del programa desde la instalación
- 7.- Número de fallos de procesamiento asociados con el punto 6
- 8.- Nivel e identificación de cambios sobre el programa
- 9.- Número de sentencias fuente añadidas en los cambios del programa
- 10.- Número de sentencias eliminadas en los cambios de programa
- 11.- Número de personas-hora por cambio
- 12.- Fecha de cambio del programa
- 13.- Identificación del ingeniero de software
- 14.- identificación del formulario de petición de mantenimiento
- 15.- Tipo de mantenimiento
- 16.- Fechas de comienzo y final del mantenimiento

- 17.- Número de personas-hora acumulado del mantenimiento
- 18.- Beneficios netos asociados con el mantenimiento realizado

- EFECTOS SECUNDARIOS DEL MANTENIMIENTO

La documentación del diseño y una cuidadosa prueba de regresión ayudan a eliminar los errores, pero seguirán apareciendo efectos secundarios del mantenimiento.

Cuando se usa en el contexto del mantenimiento de software, el término efectos secundarios implica un error u otro comportamiento indeseable aparecido como resultado de una modificación. Freedman y Weinberg (Freedman D. 1981) definen tres grandes categorías de efectos secundarios :

1) EFECTOS SECUNDARIOS SOBRE EL CODIGO

Un sencillo cambio sobre una sola sentencia puede a veces tener resultados desastrosos. El cambio inadvertido (e indetectado) de una "," por un "." tiene consecuencias casi trágicas cuando falla el software de control de vuelo de una sofisticada nave espacial. Aunque no todos los efectos secundarios tienen consecuencias tan dramáticas, el cambio invita al error y el error siempre lleva a problemas.

2) EFECTOS SECUNDARIOS SOBRE LOS DATOS

Durante el mantenimiento, a menudo se hacen cambios sobre determinados elementos de una estructura de datos o sobre la propia estructura de datos. Cuando cambian los datos, el diseño del software puede no cuadrar con los datos y aparecer errores. Los efectos secundarios sobre los datos también aparecen como resultado de las modificaciones sobre la estructura de información del software.

3) EFECTOS SECUNDARIOS SOBRE LA DOCUMENTACION

El mantenimiento se debe centrar en la configuración completa del software y no sólo en las modificaciones del código fuente. Los efectos secundarios sobre la documentación se dan cuando no se reflejan los cambios del código fuente en la documentación de diseño y en los manuales orientados al usuario.

Siempre que se haga un cambio sobre el flujo de datos, sobre la arquitectura del diseño, sobre los procedimientos (módulos) o sobre cualquier otra característica asociada, se debe actualizar la documentación técnica de soporte. La documentación de diseño que no refleja fielmente el estado actual del software es peor incluso que la ausencia total de documentación. Cuando un examen inocente de la documentación técnica lleva a una determinación incorrecta de las características del software, se darán efectos secundarios en los subsecuentes esfuerzos de mantenimiento.

EL SISTEMA DE CONTROL DE OBRAS Y LAS ESTIMACIONES

CAPITULO 2

EL SISTEMA DE CONTROL DE OBRA Y LAS ESTIMACIONES

2.1. MARCO DE REFERENCIA

Uno de los conocimientos básicos para la correcta planeación de una obra es el alcance de la obra, lo cual se logra a través del estudio de los planos.

Los planos de cualquier obra deben contener toda la información necesaria para poder llevar a cabo el proyecto y estos, conjuntamente con las especificaciones, forman el lenguaje por medio del cual el proyectista se comunica con el constructor. En el plano quedarán expresadas las tareas por realizar y posteriormente se desglosarán, formándose un catálogo de conceptos. Este es un instrumento muy importante ya que inicialmente sirve para presupuestar las obras por construir y posteriormente se convierte en la base para llevar un control administrativo (control de obra).

El Catálogo de Conceptos nos proporciona información sobre las cantidades de obra por efectuar (esto se comprenderá mejor al revisar el proceso del sistema actual), dato importante para desarrollar tanto los programas de tiempo (gráfica de Gantt), como los recursos.

Esencialmente existen dos tipos de catálogos de conceptos: el de tabulador y el de concurso o contratación. En el primero, los conceptos de obra están definidos en forma simple y no hacen referencia a obra alguna en particular. En el segundo tipo, los conceptos están más ampliamente definidos, indicándose la función de la parte específica de la obra y sirven de base para los concursos o contratos.

Para realizar una obra es necesario programar, es decir, ordenar cronológicamente un conjunto de actividades relacionadas entre sí, indicando la fecha de iniciación y de terminación de todas y cada una de ellas.

En obras de ingeniería, la forma usual de presentar un programa es a través de una gráfica y este documento se emplea para determinar la duración y secuencia de ejecución de un proyecto específico. En este caso, se utiliza el diagrama de Gantt o de Barras para este fin.

Para una mejor comprensión de los términos utilizados en esta área, en el Apéndice 1 se presenta un glosario que contiene los más importantes.

2.2. DESCRIPCION DEL SISTEMA ACTUAL

Dentro de la compañía para la cual se realizó el sistema que aquí se presenta, la Gerencia de

Proyectos es la encargada de llevar a cabo todas las actividades relacionadas con el Control de Obra y las Estimaciones.

Cuando se va a realizar una obra nueva, se realizan una serie de pasos que se ilustran en el Diagrama 2.1. para el cual se utilizó la notación presentada en la Figura 2.1.; a continuación describimos estos pasos o actividades más ampliamente.

En primer término la Gerencia de Proyectos realiza estudios de factibilidad para asegurarse de que la construcción planeada sea viable. A continuación desarrollan todos los planos arquitectónicos de la obra, los cuales contienen las especificaciones de la misma.

Una vez hecho esto, realiza una licitación para que alguna empresa se encargue de ejecutar el anteproyecto y proyecto ejecutivo, el cual una vez terminado es revisado con el presupuesto base y el tiempo de ejecución.

Posteriormente crea un documento llamado Catálogo de Conceptos, formado por partidas, referidas a partes en las cuales puede ser dividida una obra. Estas partes o subdivisiones pueden ser:

- 1) preliminares.- cimentaciones, demoliciones, preparación de superficies para recibir material, etc.
- 2) acabados.- aplicación de impermeabilizantes, banquetas, muros, etc.
- 3) herrería.- colocación de barandales en escaleras, colocación de pérgola tubular, etc.

Además de éstas, puede haber otras partidas que a su vez estén formadas por conceptos descriptivos de las operaciones a realizar dentro de la partida en cuestión. Este catálogo se captura en un procesador de textos y así se maneja durante toda la obra.

El Catálogo de Conceptos incluye la cantidad requerida para cada uno de los materiales y actividades que intervienen en las partidas, así como su unidad de medida correspondiente (litros, m², pieza, etc.).

El Catálogo de Conceptos, los planos arquitectónicos y los datos del tiempo requerido para la ejecución de la obra, son enviados posteriormente a cada uno de los oferentes (constructoras interesadas en la licitación) y se fija una fecha para entregar la oferta correspondiente en las oficinas de la empresa.

Al recibir la respuesta de cada oferente, los datos se vacían en un cuadro comparativo llamado "Cuadro Frfo", el cual contiene la siguiente información:

- Importe de la obra.-
- Costo total de la obra sin incluir costos indirectos.
- Importe total de la oferta.-
- Costo total de la obra incluyendo costos indirectos e I.V.A..
- Desviación.-
- % de diferencia entre lo presupuestado originalmente y la cantidad presupuestada por la

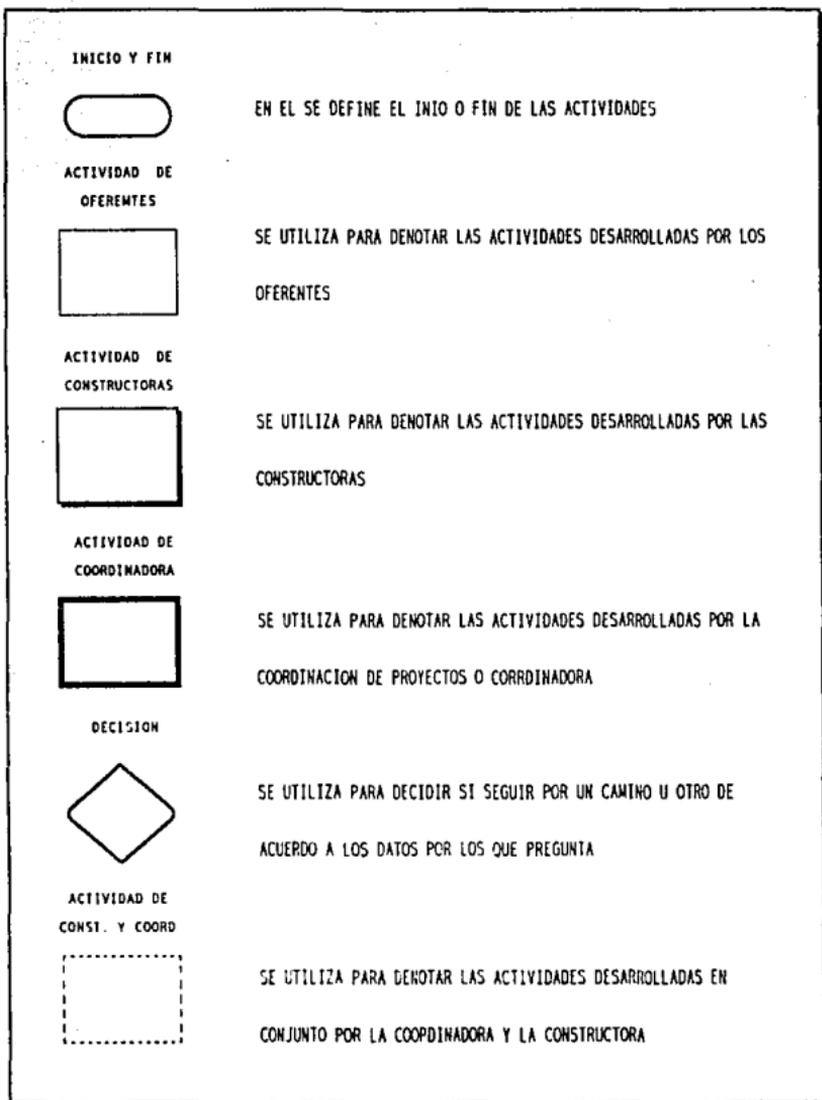


FIGURA 2.1. NOTACION UTILIZADA PARA EL DIAGRAMA DEL SISTEMA ACTUAL

DIAGRAMA DEL SISTEMA ACTUAL

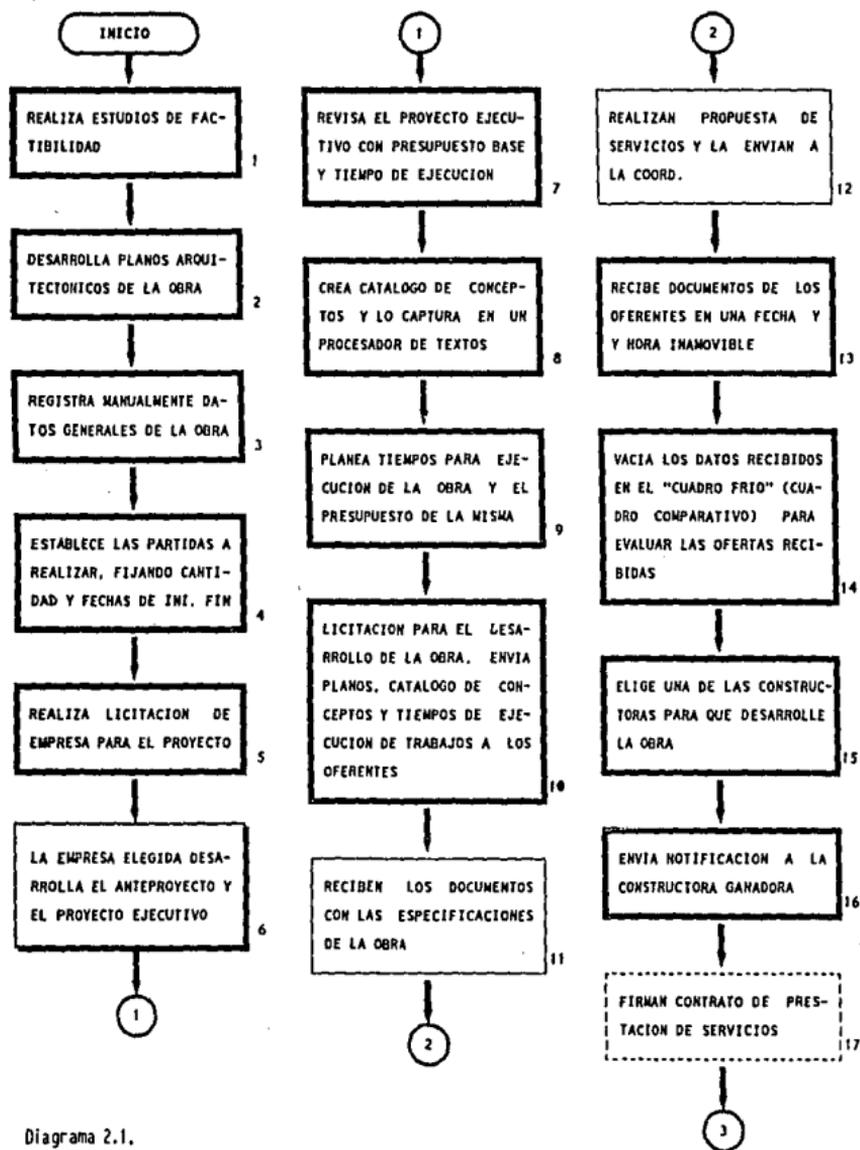


Diagrama 2.1.

DIAGRAMA DEL SISTEMA ACTUAL

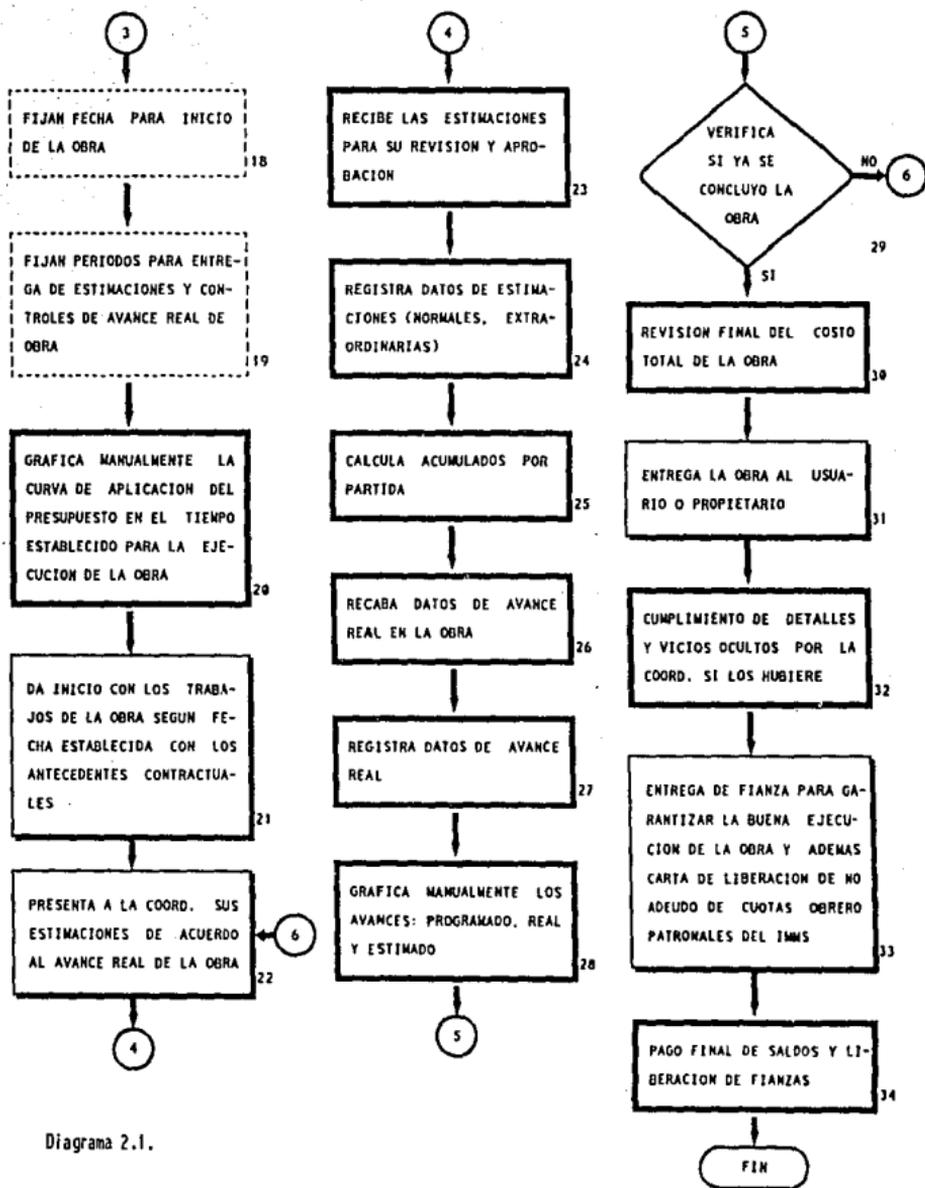


Diagrama 2.1.

constructora.

- Porcentaje de anticipo requerido.-
% de anticipo requerido para dar inicio con la obra.
- Tiempo de ejecución de obra.-
Días para la ejecución total de la obra.
- Vigencia de la oferta.-
Tiempo que respetarán los precios presentados en la oferta
- Etc.

En base a estos datos, se realiza una comparación, y se elige una de las constructoras participantes; posteriormente se fija una fecha para el inicio de la obra, los períodos de entrega de las estimaciones y del avance real de la obra.

El proceso de las estimaciones y el control de avance son continuos a lo largo de toda la obra y los datos pueden manejarse en intervalos distintos de tiempo (una semana, una quincena, un mes). Estos intervalos se fijan al comienzo de la obra y generalmente ya no se modifican, salvo casos especiales.

Con base en el tiempo fijado para la ejecución de la obra, se grafica manualmente la curva que muestra la aplicación del presupuesto a lo largo de la obra. Esto se realiza para poder apreciar qué parte del presupuesto es utilizado en un determinado período de la obra.

Durante la construcción o realización de la obra, se van recibiendo datos que informan del avance o modificaciones de la misma, tales como estimaciones y porcentajes del avance real. Las estimaciones se van registrando en hojas tabulares para después graficarse en la misma hoja que contiene la curva del presupuesto.

Las estimaciones se realizan para el control de pagos de la obra. Se clasifican en tres tipos: normales, adicionales y extraordinarias. Las normales son las que se hacen con base en las partidas contratadas desde la licitación y acordadas con la constructora. Las adicionales se presentan cuando se modifica la cantidad original de una partida normal, ya sea aumentando o disminuyendo la cantidad prevista originalmente.

Las estimaciones extraordinarias se hacen cuando se agrega una partida no contemplada en las especificaciones originales de la obra, y esto puede ser hecho cuando es necesario realizar otros trabajos que se habfan omitido por cualquier circunstancia.

Los datos de las estimaciones y el porcentaje de avance presentados por la constructora, son registrados de acuerdo a la fecha y se agregan a la gráfica como un punto más que se une con el último que se haya graficado, de estimaciones o avance, como le corresponda, o con el punto de partida si es el primero.

Los datos del avance real se utilizan durante la realización de la obra para controlar el grado de avance y si éste se encuentra dentro de los parámetros fijados en el plan de la misma para tratar de prevenir retrasos.

Las estimaciones sirven para controlar los pagos efectuados a la constructora y su graficación se utiliza para constatar que la línea de estimaciones nunca sobrepase la línea de avance real. Si esto sucediera indicaría que la constructora está cobrando más de lo que realmente ha realizado en la obra o viceversa.

Hasta este punto podemos ver que llegamos a la actividad 29 del Diagrama 2.1.: las actividades siguientes (de la 30 a la 34) corresponden al finiquito de la obra, en las cuales se realiza un cálculo con los datos de las estimaciones que se pagaron en cada período y se obtiene el total pagado a la constructora hasta ese momento. De esta forma se pueden pagar saldos pendientes, en caso de que existan, o si ya se cubrió el total de la obra, dar por terminados los pagos a la constructora y realizar las demás actividades ilustradas, concluyendo así la ejecución de la obra.

2.3. DEFINICION DEL PROBLEMA

El manejo de todos los datos relacionados con el control de obra y las estimaciones, tanto en recursos económicos como en tiempos, se manejan de forma manual. En forma periódica deben presentarse informes sobre las cantidades pagadas hasta determinada fecha, de acuerdo con las estimaciones y el total acumulado hasta ese momento. También debe calcularse el grado de avance con base en los datos recogidos en la obra y graficarlo junto con los datos de las estimaciones. La presentación de estos informes, su preparación y los cálculos que implican, llevan mucho tiempo al personal que se encarga de ellos y es una labor repetitiva y tediosa. Todos estos factores en ocasiones propician el retraso en la entrega de los informes a la dirección de la empresa y a los supervisores de obra, los cuales utilizan estos informes para evaluar el grado de avance de la obra y en caso necesario realizar ajustes sobre las actividades que se desarrollan.

Este retraso en la presentación de la información acarrea otro tipo de problemas como son la pérdida de contratos, la terminación de permisos de construcción antes de concluir las obras, decisiones erróneas por falta de información en el momento oportuno, retrasos en la construcción de la obra (lo cual implica que los costos se incrementen y que existan desviaciones muy fuertes sobre el presupuesto original asignado a la obra), posibles errores en los cálculos realizados manualmente, etc. Esta problemática generó el planteamiento de una alternativa para el manejo de esta información. Esta alternativa es un sistema por computadora capaz de manejar eficientemente la totalidad de los datos que se desprenden del control de obra y las estimaciones, de tal manera que se puedan obtener reportes en un tiempo corto y, con base en ellos, tomar decisiones en el transcurso de la obra en cuestión, consultar en el momento que se requiera cualquier información referente a la obra o a otras obras realizadas con anterioridad para realizar comparaciones o revisar acciones tomadas en casos específicos, detectar problemas potenciales e incrementar el control por parte de la Gerencia de Proyectos.

El cambio de las actividades manuales que se desarrollan para el control de obra por un manejo a través de un sistema computarizado traería como resultado muchos beneficios para el área de proyectos. Entre ellos, destaca: el rápido acceso a información de la o las obras que se desarrollen simultáneamente en un momento determinado, posibilidad de almacenamiento (en espacios muy pequeños) y consulta de datos históricos (como precios unitarios, importes de partidas, tiempos de

ejecución reales, antecedentes de constructoras participantes en licitaciones anteriores), obtención de reportes o consulta de información para toma de decisiones en caso de que se presenten problemas en la obra, o bien contar con la opción de buscar alternativas para solucionarlos a tiempo.

Otro cambio que conlleva el sistema propuesto radica en los procedimientos que sigue la gerencia de proyectos, pues el personal de costos que realiza las labores de control de obra y de las estimaciones, podrá realizar más rápido estas tareas, dejándoles libre más tiempo para realizar otras actividades de las que también son responsables.

El sistema que aquí se propone será manejado por el personal de costos en el área de proyectos y se estima que sólo requerirá capacitación en la operación del sistema pues la mayoría cuenta con experiencia en el manejo de PC's, lo cual facilita su puesta en marcha. Será este mismo personal quien alimente la información que se requiera en el sistema (como son datos de catálogos, datos generales de las obras, datos de las licitaciones, de las partidas, etc) y podrán consultar todos los datos que sean necesarios y en su caso, actualizar o modificar datos que cambien dentro de una obra e imprimir los reportes necesarios para presentar informes a la dirección de la empresa y a los encargados de supervisar el correcto desarrollo de las obras ayudándose con los informes de estimaciones y avance real.

DISEÑO DE UN SISTEMA ALTERNATIVO

CAPITULO 3

DISEÑO DE UN SISTEMA ALTERNATIVO

Con base en el estudio sobre el sistema manual, en los procedimientos que sigue el área de proyectos para la realización de sus funciones y en la problemática que se expuso en el Capítulo 2, a continuación, se explicará el sistema que se pretende maneje las actividades que se desarrollan para el Control de Obra y las estimaciones.

Este sistema computarizado se creará de acuerdo con las necesidades de la Gerencia de Proyectos y tomando en cuenta el equipo con el que se cuenta en la empresa para aprovechar al máximo sus recursos, previendo que posteriormente no se presenten problemas en su operación (por incompatibilidad con el equipo), en particular en la parte gráfica que manejará.

3.1. DESCRIPCION DEL SISTEMA PROPUESTO

Debido a la complejidad y volumen de los datos a manejar por el sistema, se decidió dividirlo en 5 módulos. El diseño modular es conveniente ya que nos permite reducir la complejidad de un problema a través de su descomposición en problemas más simples, facilitando los cambios y dando como resultados una más fácil implementación, lo cual posibilita el desarrollo paralelo de diferentes partes. Esta división considera el concepto de independencia funcional mencionado en el punto 1.3.2. del capítulo 1.

Así, los módulos de este sistema realizarán sólo la función que les corresponde y tendrán una interacción mínima con los demás, cumpliendo con dicha independencia funcional.

- OBJETIVOS DEL SISTEMA

Almacenar y procesar toda la información generada por la Gerencia de Proyectos en relación con el Control de Obra y Estimaciones, con el fin de proporcionar la información necesaria para controlar una obra, las estimaciones correspondientes, y facilitar la toma oportuna de decisiones y el control efectivo de las obras que realicen la Dirección de la empresa y otros responsables.

- METAS DEL SISTEMA

Procesar la información de Control de Obra y Estimaciones de manera eficiente, confiable, integral, oportuna y rápida; es decir, la información que el sistema almacene y genere debe ser no sólo verídica sino útil y necesaria, obtenerse en períodos de tiempo cortos y su manejo debe ser sencillo y rápido.

3.1.1. ESPECIFICACION DE REQUERIMIENTOS

El sistema deberá ser diseñado para funcionar en una red Novell Arcnet en estrella de tal forma que pueda ser accedido por todas las estaciones de trabajo conectadas. Esta característica es importante debido a que la empresa cuenta con una red de este tipo ya instalada y al manejar varias obras en diferentes estaciones de trabajo se puede trabajar a un mismo tiempo con más información, compartiendo tanto los recursos de cómputo (computadoras, impresoras, etc), como el software (en este caso, el Sistema de Control de Obras). Al compartir los recursos en esta forma, los resultados que genere el sistema serán más eficientes.

La estación de trabajo deberá constar del siguiente equipo:

Una PC con al menos 1 Mb en memoria RAM, 1 Drive de 5 1/4, monitor VGA monocromático y tarjeta para red. (El sistema también puede trabajar en PC's no conectadas a una red de área local).

El monitor VGA será necesario para el manejo del ambiente gráfico que utiliza el sistema en la presentación de la Gráfica de Gantt (partidas) y la Gráfica de avances (Real, Estimado y Presupuestado).

- OPCIONES CONTEMPLADAS

Las funciones que realiza el Sistema de Control de Obras se distribuyen en los 5 módulos mencionados anteriormente, y estos son:

A) Obras:

- Obras en Proceso.- en él se manejan todos los datos generales de la obra; estos datos se capturan una sola vez, al inicio de la obra, y se utilizan para la emisión de reportes (Actividad 3 Diagrama 2.1.).
- Partidas.- aquí se capturan los datos para controlar el tiempo de ejecución, la cantidad de unidades de unidades programadas por partida y el cálculo de su importe con base en los precios unitarios establecidos (Actividad 4 Diagrama 2.1.).
- Datos Comparativos.- en esta opción se dan de alta los datos de cada empresa concursante en la licitación para poder imprimir el cuadro comparativo (Cuadro Frío). (Actividad 14 Diagrama 2.1.).
- Reporte Comparativo.- esta opción imprime los datos capturados en la opción anterior en el "Cuadro Frío" para poder comparar las ofertas presentadas por las constructoras participantes. (Actividad 14 Diagrama 2.1.).

B) Avance Real:

- Aquí se capturan los datos del avance real de la obra recogidos en campo para ser utilizados en la consulta por pantalla de la Gráfica de Gantt o en impresión. Esta última se utilizará para archivar en el expediente de la obra y será posible obtenerla todas las veces que sea necesario por cambios en el plan de la obra. (Actividades 27 y 28 Diagrama 2.1.).

C) Estimaciones:

- Normales y Adicionales.- en esta opción se capturan los datos de las estimaciones normales y adicionales (en caso de que existan) para ser graficados y llevar el control de pagos a la constructora (Actividad 24 Diagrama 2.1.).

- Extraordinarias.- cuando se presenta alguna partida extra que no había sido contemplada originalmente, se capturará como extraordinaria en este módulo. Posteriormente se calcularán los montos totales de los pagos realizados y se imprimirá el reporte correspondiente (Actividades 24 y 25 Diagrama 2.1.).

D) Catálogos:

- Catálogo de Conceptos.- en esta opción se manejan las altas, bajas y modificaciones al catálogo de conceptos que se está utilizando para el desarrollo de la obra. (Actividad 8 Diagrama 2.1.).

- Reporte de Catálogo.- aquí se imprimirá el catálogo de conceptos para el expediente de la obra y el control de la misma (Actividad 8 Diagrama 2.1.).

E) Utillerías:

- Respaldo.- en esta opción se respaldan los datos más importantes con los que trabaja el sistema, como son catálogo de conceptos, datos de estimaciones y avance real, así como los datos generales de la obra. Este respaldo es importante porque si existe algún problema (daño de equipo, borrado accidental, etc.) con la información que maneja el sistema es posible recuperarla completa o con muy pocos datos faltantes.

- Claves.- aquí se controlan las prioridades de acceso de los usuarios que pueden hacer uso del sistema; también es posible realizar modificaciones (altas, bajas o cambios) sobre las claves de acceso existentes. Estas claves se utilizarán para restringir el acceso a puntos importantes del sistema (como son las partes donde se pueda borrar información), y así mantendremos segura la información.

- PLAN DEL PROYECTO

Se utilizará el "Ciclo de Vida por fases" mencionado en el capítulo 1 punto 1.3.

Los requisitos preliminares para el desarrollo del Sistema de Control de Obra son:

SOFTWARE

- A) Foxbase
- B) Clipper
- C) Clipx:1.98e
- D) Editor Side Kick
- E) Librería DGE V.4

Las utilerías o software mencionado en los primeros 4 incisos se utilizarán para todos los módulos que contendrá el sistema y el inciso E) se utilizará para los módulos de graficación (Gráfica de Gant y Gráfica de Avances).

Durante las pruebas se deberán:

- Prever todos los errores que el usuario pueda cometer en el uso del Sistema de Control de Obra;
- Validar todos los datos contra documentación existente;
- Generar todos los reportes que emita el sistema y verificar su contenido;
- Probar la concurrencia en uso de archivos (funcionamiento en red);
- Verificar el funcionamiento de todos los módulos;
- Validar permisos y restricciones de todos los usuarios;
- Validar resultados finales arrojados por el Sistema de Control de Obra contra documentación existente;
- Verificación de tiempos de respuesta.

3.2. DISEÑO

3.2.1. CARTA ESTRUCTURADA

Esta carta es un diagrama de árbol que nos permite observar la estructura jerárquica y las interconexiones dentro del sistema a desarrollar presentando sus opciones principales y en las que se subdivide cada una de ellas (Diagrama 3.1.).

Como se puede apreciar este diagrama sólo muestra la estructura del sistema y la dependencia de módulos sin presentar estructuras de control ni el orden secuencial de las tareas inherentes en un diagrama de flujo, por ello se utilizan los D.F.D.

Para complementar la visión general que presenta la carta estructurada, presentamos el Cuadro 3.1., el cual contiene por cada uno de los módulos principales, sus entradas y salidas sin especificar muy concretamente los datos que entran ya que se mencionan en forma general. Estos datos se refinarán posteriormente en los Diagramas de Flujo de Datos.

DIAGRAMA ESTRUCTURADO

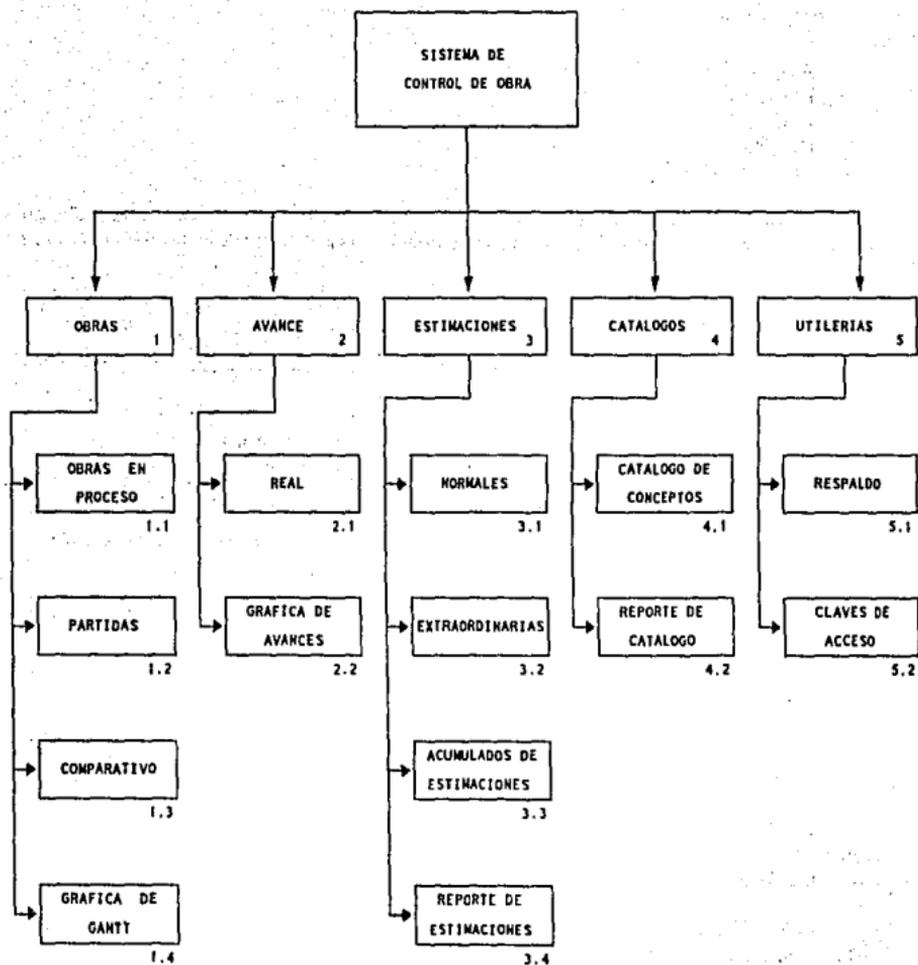


Diagrama 3.1.

MODULO	ENTRADA	SALIDA
1	<ul style="list-style-type: none"> - Datos de obras en proceso - Datos de partidas - Datos cuadro frio (Datos de Oferentes) 	<ul style="list-style-type: none"> - Datos actualizados de obras en proceso - Datos actualizados de partidas - Datos de oferentes (Cuadro Frio, reporte) - Grafica de Gantt
2	<ul style="list-style-type: none"> - Datos de avance real - Datos de obras en proceso - Datos de estimaciones 	<ul style="list-style-type: none"> - Datos actualizados de avance real - Grafica de avances (Presupuestado, Estimado y Real)
3	<ul style="list-style-type: none"> - Datos de estimaciones normales - Datos de estimaciones extraordinarias 	<ul style="list-style-type: none"> - Datos actualizados de estimaciones normales - Datos actualizados de estimaciones extraordinarias - Reporte de estimaciones
4	<ul style="list-style-type: none"> - Datos de catalogo de conceptos 	<ul style="list-style-type: none"> - Datos actualizados de catalogo de conceptos - Reporte de catalogo de conceptos
5	<ul style="list-style-type: none"> - Archivos del sistema - Datos de los usuarios del sistema 	<ul style="list-style-type: none"> - Datos actualizados de claves de acceso

DATOS DE ENTRADA Y SALIDA (CARTA ESTRUCTURADA)

Cuadro 3.1.

3.2.2. DIAGRAMAS DE FLUJO DE DATOS (D.F.D.)

Estos diagramas nos sirven para mostrar el dominio de información del sistema, así como el conjunto de datos que lo alimentan y aquéllos que son arrojados por el mismo. La simbología que se emplea en él se muestra en la Figura 3.1.

El diagrama 3.2 comienza en un nivel "0" en el que se muestran en forma general todas las entradas al sistema de Control de Obras, el cual en este momento se representa como una caja negra ya que se desconocen las transformaciones que sufre la información hasta producir las salidas del sistema. Este diagrama posteriormente se va refinando o subdividiendo en otros niveles hasta llegar a un grado de detalle que nos permita traducirlo más fácilmente a un lenguaje de programación.

En el D.F.D. no se da ninguna indicación explícita de la secuencia de sucesos, pero sí se especifica el flujo de datos del sistema.

En el nivel Intermedio se presenta otro grado de abstracción del sistema, en el cual ya se incluyen los 5 módulos que lo conformarán, sus respectivas entradas y salidas, los archivos que utilizará y cómo fluye la información de un módulo a otro (Diagrama 3.3.).

Posteriormente, en el Diagrama 3.4., se presenta el módulo I "Obras" el cual contempla 4 subdivisiones: Obras en Proceso, Partidas, Datos Cuadro Comparativo y Gráfica de Gantt.

En la primera opción se reciben como entrada los datos de obra, marcado con el "I" y estos datos son:

- N° de contrato
- Frente
- Localidad
- Delegación o Municipio
- Estado
- Programa
- Inicio Programado
- Inicio Real
- Días de Obra y,
- Presupuesto

En la segunda opción se reciben como entrada los datos de partidas marcados con el "II":

- N° de Contrato
- Clave
- Cantidad Programada
- Fecha Inicial y,
- Fecha Final

PROCESO



Es utilizado cuando se ejecuta alguna transformación de sus datos de entrada, produciendo sus datos de salida

FLUJO DE DATOS



Se utiliza para conectar los procesos entre sí indicando la dirección de transferencia de los datos

ALMACEN DE DATOS



Es un depósito de datos; la cabeza de las flechas indican las entradas y salidas del almacén

FIGURA 3.1. SIMBOLOGIA DE LOS D.F.D.

Fuente: Pressman R., Ingeniería del Software (1988; p. 188)

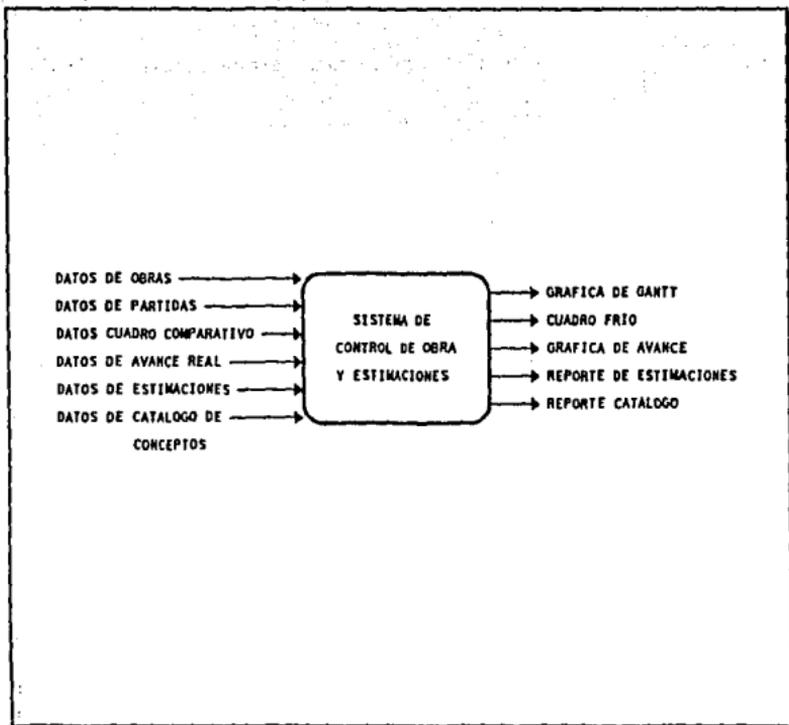


DIAGRAMA DE FLUJO DE DATOS

Diagrama 3.2.

Nivel 0

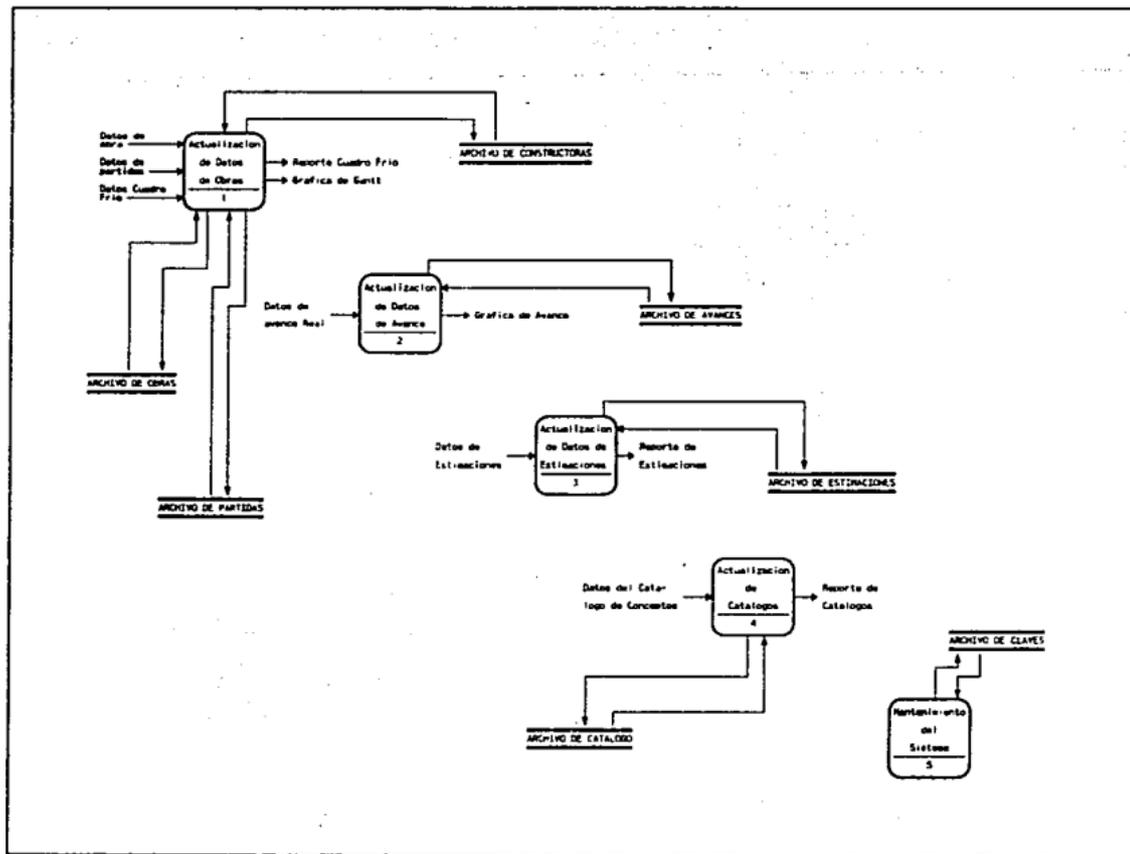


Diagrama 3.3. DIAGRAMA DE FLUJO DE DATOS Nivel Intermedio

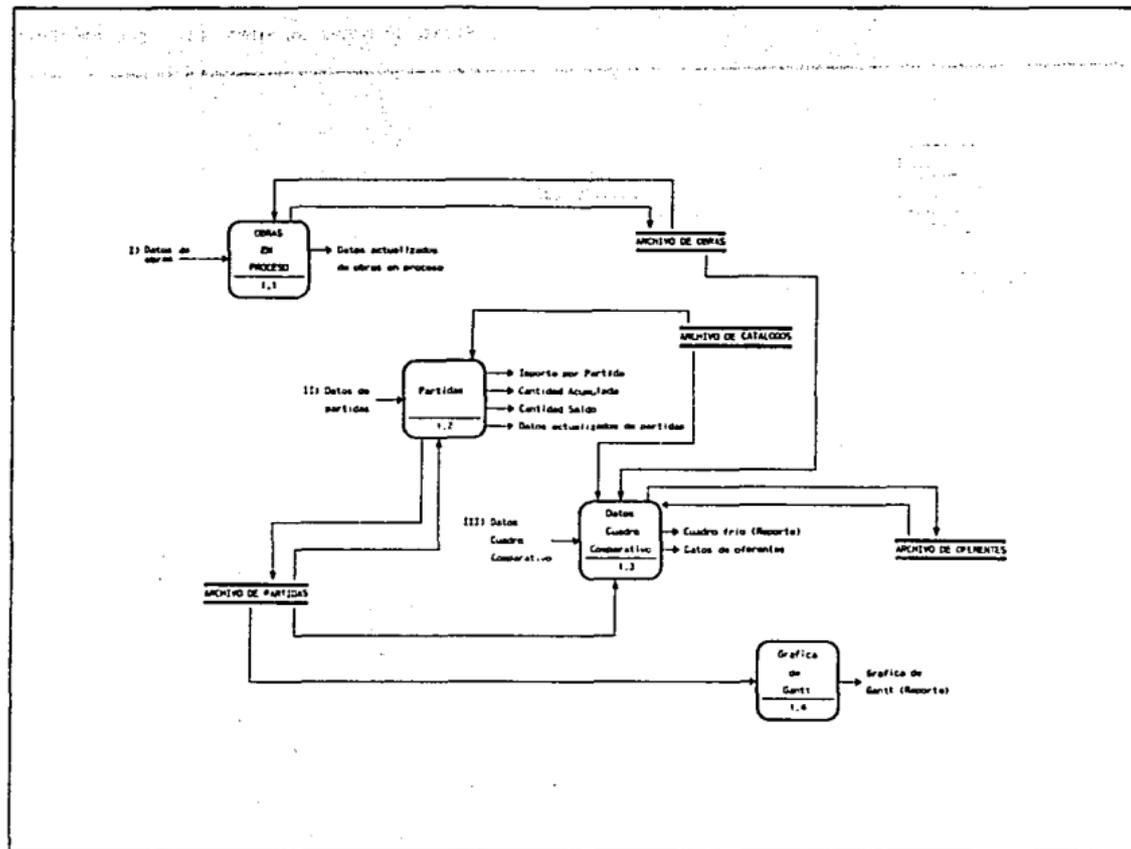


Diagrama 3.4. DIAGRAMA DE FLUJO DE DATOS
Nivel 1

La tercera opción recibe como entrada los datos del cuadro comparativo, marcado con el III y estos datos son:

- N° de Licitación
- N° de Compañía
- Importe por partida
- Porcentaje de anticipo requerido
- Tiempo de ejecución de la obra
- Interés en la licitación
- Antecedentes con la compañía
- Calidad en la presentación de la oferta
- Si retornó o no documentos y planos de la licitación
- Si presentó análisis básicos
- Si presentó matrices de precios unitarios
- Si presentó dudas por escrito sobre la licitación, y
- Vigencia de la oferta

La opción 4 se alimenta de los datos contenidos en el archivo de partidas para presentar la gráfica de Gantt.

Este diagrama presenta las relaciones que existirán entre los módulos, sus entradas, salidas y los archivos que utilizará.

El Diagrama 3.5. muestra el módulo 2 (Avance Real), el cual se subdividirá en dos opciones, avance real y gráfica de avances, también sus entradas, salidas y archivos con los que trabajará.

El Diagrama 3.6. muestra el módulo 3 (estimaciones), el cual se subdividirá en 4 opciones, que serán: estimaciones normales, estimaciones adicionales, estimaciones extraordinarias y reporte de estimaciones.

En las primeras tres opciones, se recibirá como entrada los datos de estimaciones, los cuales son:

- N° de Contrato
- N° de estimación
- Factura
- Fecha de factura, y
- Período de ejecución

En el mismo diagrama se muestra la interrelación entre estas opciones y los archivos que utilizarán (Diagrama 3.6.).

El módulo 4 (Catálogos) se presenta en el Diagrama 3.7. y se subdivide en dos opciones: Catálogo de conceptos y reporte de catálogo. En este diagrama se muestran las entradas, salidas y los archivos que utilizará.

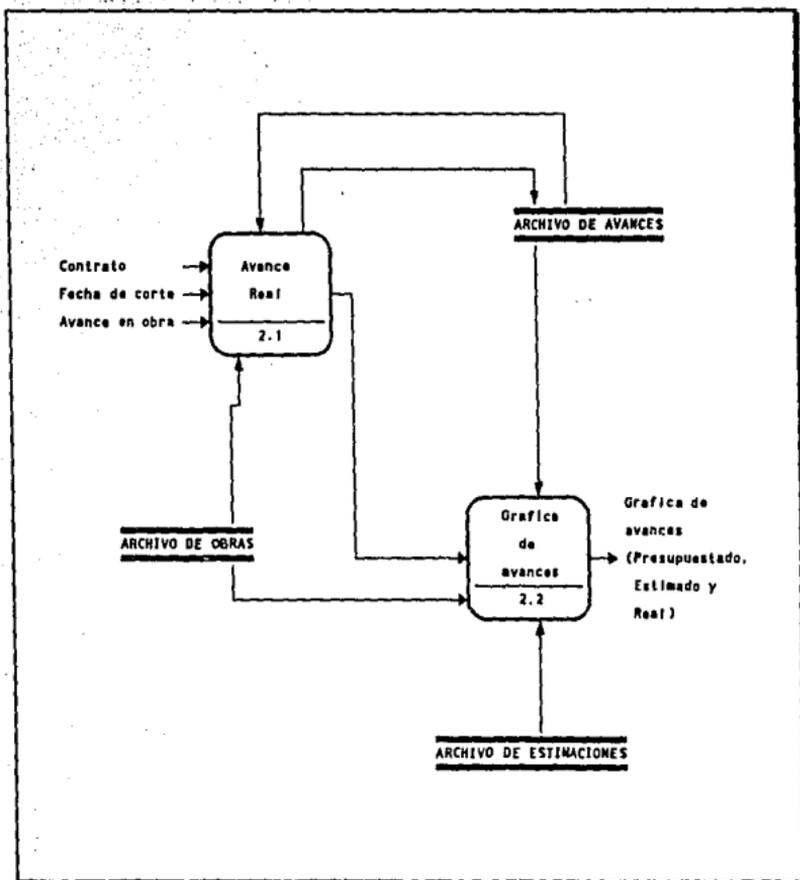


Diagrama 3.5. DIAGRAMA DE FLUJO DE DATOS

Nivel 1

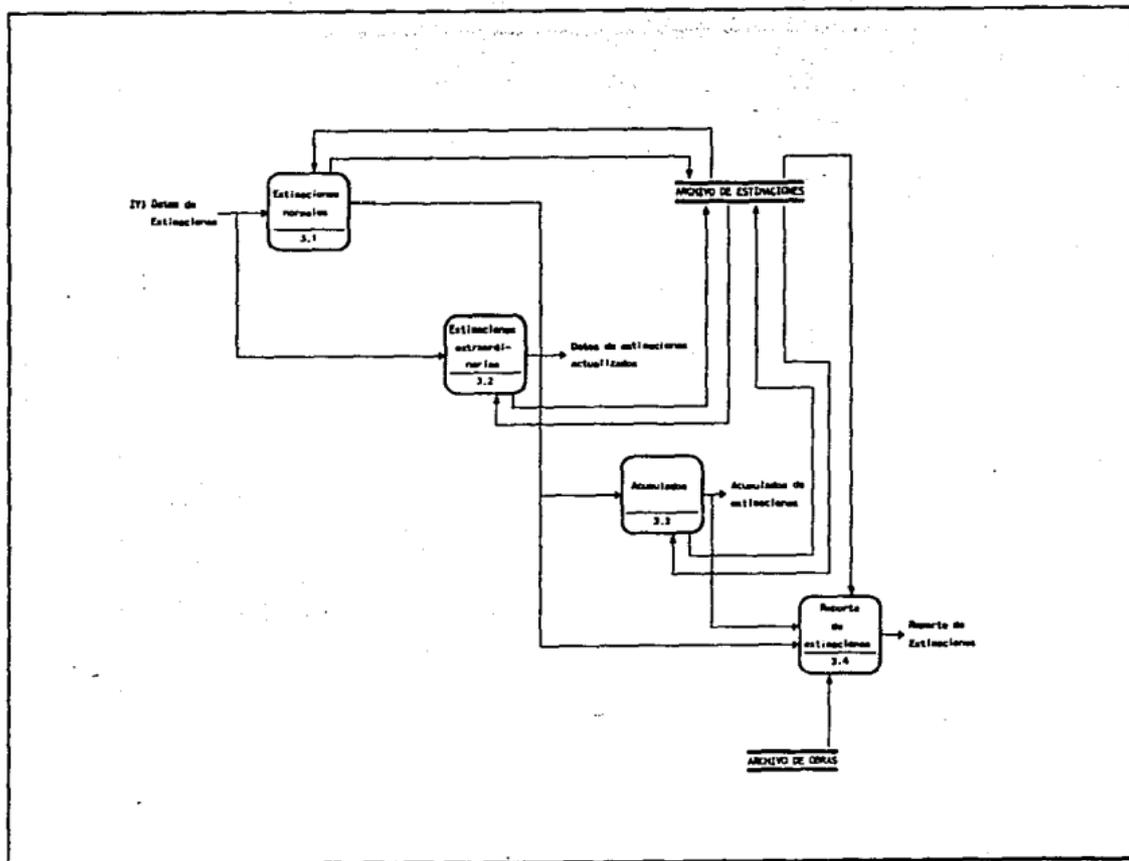


Diagrama 3.6. DIAGRAMA DE FLUJO DE DATOS Nivel 1

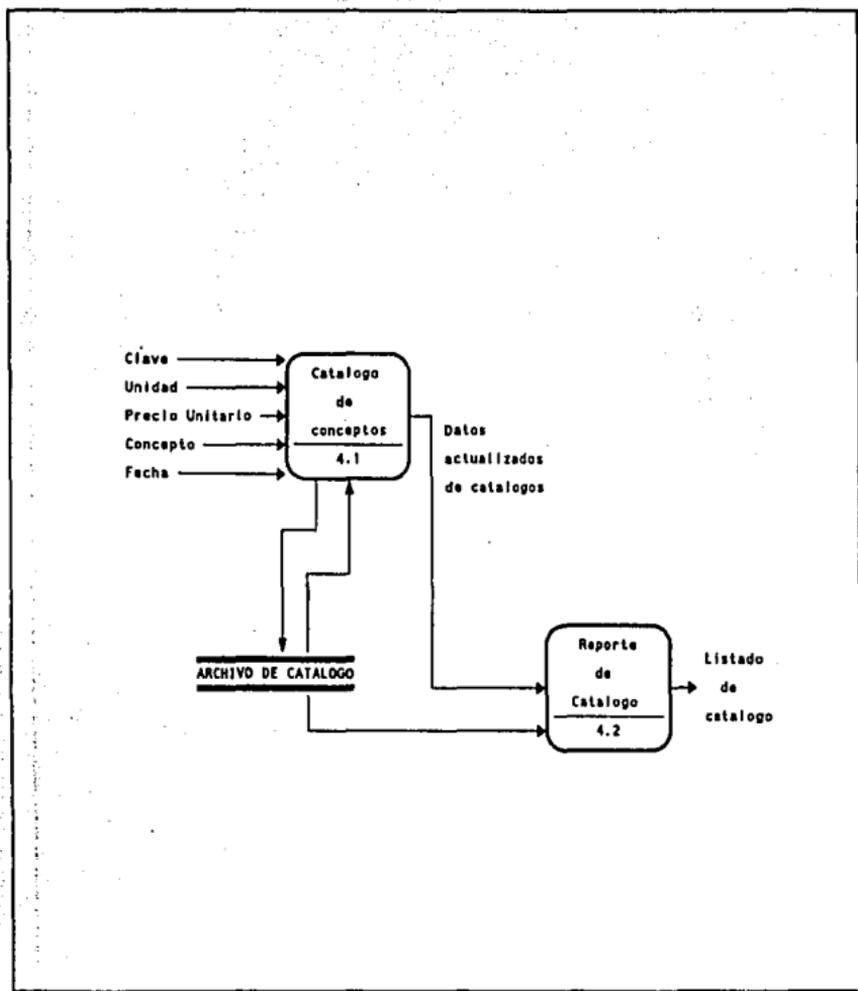


Diagrama 3.7. DIAGRAMA DE FLUJO DE DATOS

Nivel 1

El módulo 5 (Utilerías) se subdividirá en 2 opciones, respaldo y claves de acceso (Diagrama 3.8.), en la primera opción se respaldarán los archivos con que trabaja el sistema a un diskette y en la segunda se dará mantenimiento a las claves de acceso al sistema.

Estos diagramas muestran un esquema más refinado de los datos de entrada presentados en el Cuadro 3.1., diferenciando los datos de entrada más explícitamente y ya no en una forma tan general como se presentó en el cuadro de la carta estructurada.

En el nivel 1 de los D.F.D. que abarca del diagrama 3.4. al 3.8. ya nos aporta una visión más completa de las opciones que manejará el sistema, sus entradas, salidas y archivos que utilizará, con lo cual se nos facilitará el paso de la programación basándonos en este diseño. Pero aún falta por definir explícitamente los archivos que se utilizarán y los datos que contendrá cada uno para lo cual realizaremos a continuación el diccionario de datos.

3.2.3. DICCIONARIO DE DATOS

Un análisis del dominio de la información puede ser incompleto si sólo se considera el flujo de datos. Cada flecha de un Diagrama de Flujo de Datos representa uno o más elementos de información, por lo tanto debe representarse el contenido de cada flecha de un D.F.D. y para esto se utiliza el Diccionario de Datos.

Una definición de Diccionario de Datos nos la da Page-Jones, M. y dice :

"El diccionario de datos 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. Los datos compuestos (datos que pueden además ser divididos) se definen en términos de sus componentes; los datos elementales (datos que no pueden ser divididos) se definen en términos del significado de cada uno de los valores que puede asumir. Por tanto, el diccionario de datos esta compuesto de definiciones de flujo de datos, archivos (datos almacenados) y datos usados en los procesos (transformaciones)." (Pressman, R. ;193)

Para la creación del diccionario de datos se utiliza la siguiente notación:

Construcción de datos	Notación	Significado
	=	Está compuesto de
Secuencia	+	y
Selección	[]	Uno u otro
Repetición	{ } ()	n repeticiones de Datos opcionales

El diccionario de datos se expande hasta que todos los datos compuestos han sido

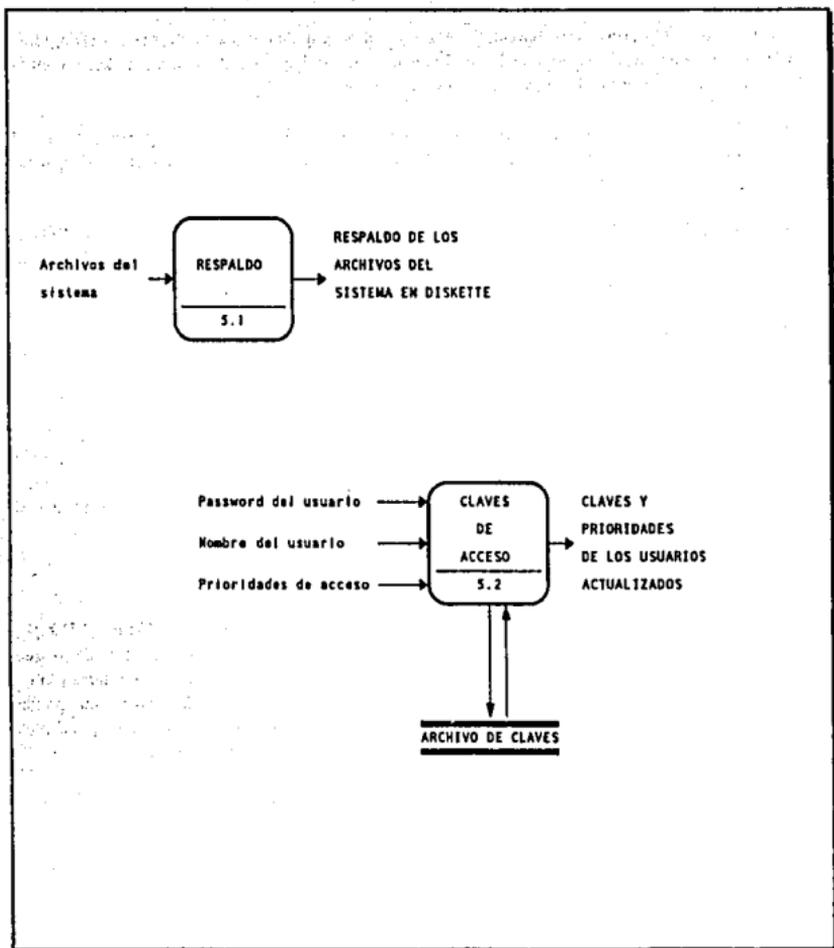


Diagrama 3.8. DIAGRAMA DE FLUJO DE DATOS

Nivel 1

representados como componentes elementales o hasta que todos los elementos compuestos están representados en términos que pueden ser bien conocidos y de forma no ambigua para todos los lectores. Esta descomposición de todos los datos nos facilitará posteriormente la creación de las estructuras de datos que utilizará el sistema ya que nos proporciona una visión completa de los datos que se utilizarán y los elementos que componen cada uno de ellos.

El diccionario de datos para el Sistema de Control de Obra se presenta en las Figuras 3.2., 3.3. y 3.4.. De estos datos y su descomposición partiremos para la descripción de los archivos que se utilizarán.

3.2.4. DESCRIPCION DE ARCHIVOS

La descripción de archivos es otra parte muy importante dentro del diseño de cualquier sistema ya que una vez definidas las estructuras y los datos que utilizará, se partirá para la creación de los programas fuente con datos como: nombre de cada una de las variables, tipo de información que contienen, longitud y en que archivo se encuentra cada una.

Para presentar los archivos y su estructura se utilizará el formato mostrado en la Fig. 3.5., en la cual se presentan los siguientes datos: Sistema del cual se trata, Clave del sistema, Archivo (que es un nombre descriptivo del archivo que se describirá), Nemónico (nombre por el que se reconocerá el archivo dentro de los programas del sistema) y a continuación se listará los campos necesarios para cada archivo con su nombre descriptivo, nemónico, tipo, longitud y posiciones que utilizará dentro del archivo.

En los tipos de campo se utilizarán diferentes letras para identificar en cada uno de ellos el tipo específico de datos que contendrá, y estos son:

- N Numérico
- C Caracter
- M Memo
- F Fecha

Esta descripción de archivos además de facilitarnos el paso del diseño a la codificación, nos ayuda a crear un estándar para los nombres de los campos de cada uno de los archivos y de esta forma referirnos a ellos en todos los programas en que sean utilizados.

Aquí sólo presentaremos la descripción de los archivos más importantes del Sistema de Control de Obras y éstos se presentan en los cuadros del 3.2. al 3.14.

DICCIONARIO DE DATOS

SISTEMA: CONTROL DE OBRA	
ELEMENTO	DESCRIPCION
DATOS DE OBRA	• No. DE CONTRATO + FRENTE + LOCALIDAD + DELEGACION + ESTADO + PROGRAMA + INICIO PROGRAMADO + INICIO REAL + DIAS DE OBRA + PRESUPUESTO
No. DE CONTRATO	• No. CONSECUTIVO + AÑO DEL CONTRATO
No. CONSECUTIVO	• [001 002 003 . . . 999]
AÑO DEL CONTRATO	• [92 93 94 . . .]
FRENTE	• PARTE DE LA OBRA QUE SE DESARROLLA
LOCALIDAD	• LUGAR DONDE SE DESARROLLA LA OBRA
DELEGACION O MUNICIPIO	• [ATIZAPAN NAUCALPAN TLALPAM . . . COYOACAN]
ESTADO	• [EDO. DE MEXICO . . . D.F.]
PROGRAMA	• AÑO DE LOS TRABAJOS
AÑO DE LOS TRABAJOS	• [1992 1993 1994 . . .]
INICIO PROGRAMADO	• FECHA PROGRAMADA DE INICIO DE TRABAJOS
INICIO REAL	• FECHA REAL DE INICIO DE TRABAJOS
DIAS DE OBRA	• DIAS QUE DURARA LA REALIZACION DE LA OBRA
PRESUPUESTO	• CANTIDAD DE DINERO ASIGNADA PARA EL DESARROLLO DE LA OBRA
DATOS DE PARTIDAS	• No. DE CONTRATO + CLAVE DE PARTIDA + CANTIDAD PROGRAMADA + FECHA INICIAL DE PARTIDA + FECHA FINAL DE PARTIDA
CLAVE DE PARTIDA	• [1.01 1.02 1.03 . . . 99.99]
CANTIDAD PROGRAMADA	• No. DE UNIDADES A UTILIZAR POR CADA PARTIDA
FECHA INICIAL DE PARTIDA	• FECHA EN QUE SE DEBERA INICIAR A TRABAJAR EN LA PARTIDA EN CUESTION
FECHA FINAL DE PARTIDA	• FECHA EN QUE SE DEBERA TERMINAR CON LA PARTIDA EN CUESTION
DATOS CUADRO COMPARATIVO	• No. DE CONTRATO + No. DE LICITACION + No. DE COMPANIA + IMPORTE POR PARTIDA + PORCENTAJE DE ANTICIPO + TIEMPO DE EJECUCION DE LA OBRA + INTERES EN LA LICITACION + ANTECEDENTES CON LA COMPANIA + CALIDAD EN LA PRESENTACION DE LA OBRA + RETORNA DOCUMENTOS Y PLANOS DE LA LICITACION + PRESENTA ANALISIS BASICOS + PRESENTA MATRICES DE PRECIOS UNITARIOS + PRESENTA DUDAS POR ESCRITO SOBRE LA LICITACION + VIGENCIA DE LA OFERTA
No. DE LICITACION	• No. CONSECUTIVO
No. CONSECUTIVO	• [01 02 03 . . . 99]
No. DE COMPANIA	• [1 2 3 . . . 9]
IMPORTE POR PARTIDA	• COSTO DE CADA PARTIDA
PORCENTAJE DE ANTICIPO REQUERIDO	• PORCENTAJE DEL COSTO DE LA OBRA QUE REQUERIDO PARA INICIAR LA OBRA
TIEMPO DE EJECUCION DE LA OBRA	• TIEMPO (EN DIAS) PARA EL DESARROLLO DE LA OBRA
INTERES EN LA LICITACION	• [MB B M]

Figura 3.2.

DICCIONARIO DE DATOS

SISTEMA: CONTROL DE OBRA	
ELEMENTO	DESCRIPCION
MB	- MUY BUENO
B	- BUENO
M	- MALO
ANTECEDENTES CON LA COMPANIA	- [MB B M]
CALIDAD EN LA PRESENTACION DE LA OFERTA	- [MB B M]
RETORNA DOCUMENTOS Y PLANOS DE LA LICITACION	- [SI NO]
PRESENTA ANALISIS BASICOS	- [SI NO]
PRESENTA MATRICES DE PRECIOS UNITARIOS	- [SI NO]
PRESENTO DUDAS POR ESCRITO SOBRE LA LICITACION	- [SI NO]
VIGENCIA DE LA OFERTA	- No. DE DIAS QUE RESPETARAN LOS PRECIOS PRESENTADOS EN LA OFERTA
DATOS DE AVANCE REAL	- No. DE CONTRATO + FECHA DE CORTE + AVANCE DE OBRA
FECHA DE CORTE	- FECHA EN QUE SE LEVANTAN LOS DATOS DE AVANCE
AVANCE DE OBRA	- PORCENTAJE DE AVANCE REAL DE LOS TRABAJOS DE LA OBRA
PORCENTAJE DE AVANCE REAL DE LOS TRABAJOS DE LA OBRA	- [10 % 20 % 30 % . . . 100 %]
DATOS DE ESTIMACIONES	- No. DE CONTRATO + No. DE ESTIMACION + No. DE FACTURA + FECHA DE FACTURA + PERIODO DE EJECUCION
No. DE ESTIMACION	- No. CONSECUTIVO DE ESTIMACION
No. CONSECUTIVO DE ESTIMACION	- [001 002 003 . . . 999]
No. DE FACTURA	- No. CONSECUTIVO DE FACTURA
No. CONSECUTIVO DE FACTURA	- [0001 0002 0003 . . . 9999]
FECHA DE FACTURA	- FECHA DE LA FACTURA QUE SE EXTIENDE PARA EL PAGO
PERIODO DE EJECUCION	- FECHA INICIAL + FECHA FINAL
FECHA INICIAL	- FECHA DE INICIO DEL PERIODO DE LA ESTIMACION
FECHA FINAL	- FECHA DE TERMINO DEL PERIODO DE LA ESTIMACION
DATOS DE CATALOGO	- No. DE CONTRATO + CLAVE + UNIDAD + PRECIO UNITARIO + CONCEPTO + FECHA

Figura 3.3.

DICCIONARIO DE DATOS

SISTEMA: CONTROL DE OBRA	
ELEMENTO	DESCRIPCION
CLAVE	- [No. DE PARTIDA No. DE CONCEPTO]
No. DE PARTIDA	- [1.00 2.00 3.00 . . . 99.00]
No. DE CONCEPTO	- [1.01 1.02 1.03 . . . 99.99]
UNIDAD	- UNIDAD DE MEDIDA UTILIZADA
UNIDAD DE MEDIDA UTILIZADA	- [PIEZA M ² M ³ KG . . . TON]
PRECIO UNITARIO	- PRECIO POR CADA UNIDAD QUE SE UTILIZE
CONCEPTO	- DESCRIPCION DE LA PARTIDA O EL CONCEPTO
FECHA	- FECHA DE LOS DATOS DEL CATALOGO

Figura 3.4.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB		
ARCHIVO:			NEMONICO :		
NOMBRE	CAMPO	NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A

Figura 3.5.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB		
ARCHIVO: Datos de estimaciones normales, adicionales y extraordinarias			NEMONICO : Estimac.Dbf		
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A	
1) Clave de la obra (1)	Obra	C	3	1	3
2) Clave de la obra (2)	Obra1	C	2	4	5
3) Tipo de estimacion	Tipo	N	1	6	6
4) Numero de estimacion	N_est	N	3	7	9
5) Numero de factura	Fact	N	4	10	13
6) Fecha de la factura	Fec_fac	F	8	14	21
7) Fecha de inicio del periodo	Per_ini	F	8	22	29
8) Fecha final del periodo	Per_fin	F	8	30	37
9) Monto de estimacion normal o adicional	M_estn	N	11 2	38	48
10) Acumulado de estimacion normal o adicional	A_estn	N	13 2	49	61
11) Amortizacion de estimacion normal o adicional (parcial)	Am_estnp	N	11 2	62	72
12) Amortizacion de estimacion normal o adicional (acum.)	Am_estna	N	11 2	73	83
13) Parcial de IVA estimacion normal o adicional	Iv_estnp	N	11 2	84	94
14) Acumulado de IVA estimacion normal o adicional	Iv_estna	N	11 2	95	105

Cuadro 3.2.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA				CLAVE : SISCOB	
ARCHIVO: Datos de estimaciones normales, adicionales y extraordinarias				NEMONICO : Estimac.Db1	
NOMBRE	CAMPO	NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
15) Neto pagado en factura estimacion normal o adicional (parcial)		Npf_enp	N	11 2	106 116
16) Neto pagado en factura estimacion normal o adicional (acumulado)		Npf_ena	N	11 2	117 127
17) Monto de estimacion extraordinaria		M_este	N	11 2	128 138
18) Acumulado estimacion extraordinaria		A_este	N	11 2	139 149
19) IVA estimacion extraordinaria (parcial)		Iv_estep	N	11 2	150 160
20) IVA estimacion extraordinaria (acumulado)		Iv_estea	N	11 2	161 171
21) Neto pagado en factura estimacion extraordinaria (parcial)		Npf_eep	N	11 2	162 172
22) Neto pagado en factura estimacion extraordinaria (acum.)		Npf_eea	N	11 2	173 183
23) Saldo por ejercer		Sdo_pe	N	17 2	184 194

Cuadro 3.3.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB	
ARCHIVO: Avance real en obras			NEMONICO : Avan_ob.Dbf	
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
1) Clave de la obra (1)	Obra	C	3	1 3
2) Clave de la obra (2)	Obral	C	2	4 5
3) Fecha de corte	Fec_cor	F	8	6 13
4) Avance real en obra	Avan_re	N	6 2	14 19

Cuadro 3.4.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB		
ARCHIVO: Partidas de obra			NEMONICO : Subcat.Dbf		
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A	
1) Clave de la obra (1)	Obra	C	3	1	3
2) Clave de la obra (2)	Obra1	C	2	4	5
3) Clave de la partida	Clave1	N	5 2	6	10
4) Descripción de la partida	Desc	M	10	11	20
5) Unidad en que se mide la partida	Unidad	C	4	21	24
6) Precio unitario de la partida	Pre_uni	N	16 2	25	40

Cuadro 3.5.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA		CLAVE : SISCOB		
ARCHIVO: Partidas ordenadas para creacion de grafica de Gantt		NEMONICO : Ordpar.Dbf		
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
1) Clave de la partida	Clavel	N	5 2	1 5
2) Fecha de inicio programado	Fec_inp	F	8	6 13
3) Fecha de terminacion programada	Fec_tep	F	8	14 21

Cuadro 3.6.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB	
ARCHIVO: Informacion de partidas en obra			NEMONICO : Partidas.Dbf	
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
1) Clave de la obra (1)	Obra	C	3	1 3
2) Clave de la obra (2)	Obra1	C	2	4 5
3) Clave de la partida	Clave1	N	5 2	6 10
4) Cantidad programada	Can_prog	N	10 2	11 20
5) Importe de la partida	Importe	N	15 2	21 35
6) Cantidad acumulada	Can_acu	N	15 2	36 50
7) Cantidad saldo	Can_sal	N	15 2	51 65
8) Fecha de inicio programado	Fec_inp	F	8	66 73
9) Fecha de terminacion programada	Fec_tep	F	8	74 81

Cuadro 3.7.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA				CLAVE : SISCOB	
ARCHIVO: Datos de obras en proceso				NEMONICO : Obras.Dbf	
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A	
1) Clave de la obra (1)	Obra	C	3	1	3
2) Clave de la obra (2)	Obra1	C	2	4	5
3) Nombre de la obra	Frente	C	30	6	35
4) Localidad de la obra	Local	C	30	36	65
5) Delegacion o municipio de la obra	Deleg	C	20	66	85
6) Estado de la obra	Estado	C	20	86	105
7) Programa de trabajo de la obra	Prog	N	4	106	109
8) Fecha de inicio programado	Fec_inip	F	8	110	117
9) Fecha de inicio real	Fec_inir	F	8	118	125
10) Dias de duracion de la obra	Dias_ob	N	3	126	128
11) Presupuesto asignado a la obra	Presup	N	17 2	129	145
12) Amortizacion en pagos	Amortiza	N	2	146	147
13) Costo real de obra	Cto_real	N	17 2	148	164

Cuadro 3.8.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB		
ARCHIVO: Partidas de constructoras participantes en licitacion			NEMONICO : Cons_par.Dbf		
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL		POSICION DE A
1) Numero de licitacion	No_lic	C	2		1 2
2) Numero de compañía	No_cla	C	2		3 4
3) Numero de partida	No_par	N	5	2	5 9
4) Importe de la partida	Importe	N	17	2	10 26

Cuadro 3.9.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA		CLAVE : SISCOB			
ARCHIVO: Temporal de partidas para creacion de grafica de Gantt		NEMONICO : Tempar.DbF			
HOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A	
1) Numero de partida	Clavel	N	5 2	1	5
2) Fecha de inicio programada	Fec_inp	F	8	9	13
3) Fecha de terminacion programada	Fec_tep	F	8	14	21

Cuadro 3.10.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB		
ARCHIVO: Datos de constructoras para licitacion			NEMONICO : Construc.Dbf		
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A	
1) Clave de la obra (1)	Obra	C	3	1	3
2) Clave de la obra (2)	Obra1	C	2	4	5
3) Numero de licitacion	No_lic	C	2	6	7
4) Numero de compañía	No_cia	C	2	8	9
5) Nombre de la compañía (1)	Nom_cia1	C	18	10	27
6) Nombre de la compañía (2)	Nom_cia2	C	18	28	45
7) Porcentaje de anticipo requerido	Por_ant	N	2	46	47
8) Tiempo de ejecucion de la obra	Tiempo	N	3	48	50
9) Interes en la licitacion	Interes	C	2	51	52
10) Antecedentes con la compañía	Anteced	C	2	53	54
11) Calidad en la presentacion de la oferta	Calidad	C	2	55	56
12) Retorna documentos y planos de la licitacion	Doc_y_pla	C	1	57	57
13) Retorna los documentos y planos firmados	Ret_dyp	C	1	58	58
14) Presenta analisis basicos	Ana_bas	C	1	59	59
15) Presenta matrices de precios unitarios	Mat_pun	C	1	60	60

Cuadro 3.11.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA		CLAVE : SISCOB		
ARCHIVO: Datos de constructoras para licitacion		NEMONICO : Construc.Dbf		
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
16) Presenta dudas por escrito	Dud_x_esc	C	1	61 61
17) Vigencia de la oferta	Vigencia	N	2	62 63

Cuadro 3.12.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA			CLAVE : SISCOB	
ARCHIVO: Conceptos de obra			NEMONICO : Catalogo.Dbf	
NOMBRE	CAMPO NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
1) Clave de la obra (1)	Obra	C	3	1 3
2) Clave de la obra (2)	Obra1	C	2	4 5
3) Numero de concepto	Clave	N	5 2	6 10
4) Descripcion del concepto	Concepto	C	30	11 40
5) Precio unitario	Precio_uni	N	16 2	41 56

Cuadro 3.13.

DESCRIPCION DE ARCHIVOS

SISTEMA: CONTROL DE OBRA				CLAVE : SISCOB	
ARCHIVO: Temporal alta de partidas				NEMONICO : Altpar.Dbf	
NOMBRE	CAMPO	NEMONICO	TIPO	LONGITUD ENTERO DECIMAL	POSICION DE A
1) Numero de partida	Partida	Partida	N	5 2	1 5
2) Descripcion de la partida	Desc	Desc	C	30	6 35
3) Importe de la partida	Importe	Importe	N	17 2	36 52

Cuadro 3.14.

DESARROLLO DEL SISTEMA (SISCOB)

CAPITULO 4

DESARROLLO DEL SISTEMA (SISCOB)

4.1. INSTRUMENTACION

4.1.1. CREACION DE PROGRAMAS FUENTE

En la creación de los programas fuente se trató de optimizar el número de líneas y para esto se realizó un estudio con el objeto de definir cuáles funciones se utilizarían en diferentes módulos del sistema y así crear un programa con aquellas funciones y procedimientos comunes.

Estas funciones y procedimientos se reúnen en un sólo programa y en él podremos encontrar las funciones para el control de los archivos en su funcionamiento para red tales como:

- Net_use (abre un archivo para uso exclusivo o compartido)
- Fil_lock (asegura un archivo para uso exclusivo y poderlo modificar)
- Rec_lock (asegura un registro de un archivo para modificarlo)
- Add_rec (permite la inserción de un nuevo registro en forma exclusiva)

y además, el control de impresión para los reportes, el cual será igual en todos los que genere el sistema.

En los párrafos anteriores, hemos hablado del uso de archivos en forma exclusiva o compartida. Un archivo se abrirá en forma compartida en aquellos casos en los que únicamente vamos a consultar la información que contiene, sin modificarla, y de esta forma exista la posibilidad de que otro usuario pueda utilizar el mismo archivo. Por otro lado, cuando lo abrimos en forma exclusiva solamente un usuario podrá trabajar con ese archivo y los demás deberán esperar a que sea liberado al terminar sus modificaciones.

Otro punto a considerar en la creación de los programas fuente es el uso de un programa por cada parte integrante del sistema y no programas monolíticos, ya que esto acarrearía muchos problemas en la revisión de los mismos y haría del mantenimiento una labor muy complicada y lenta.

Así mismo, la indentación de las líneas del programa deberá ser utilizada para que nos permita apreciar con facilidad la estructura del mismo y donde comienza y donde termina el cuerpo principal y las funciones y procedimientos que contenga.

La documentación del código fuente es una parte muy importante que nos servirá para facilitar su posterior revisión, y para esto cada uno de ellos contendrá un encabezado en el que se precisen los

siguientes datos:

- Nombre del programa
- Explicación breve de la(s) función(es) que realiza
- Sistema al que pertenece
- Fecha de creación
- Nombre del programador
- Fecha de última actualización, y
- Nombre de la persona que realizó la actualización.

Y en las partes importantes de cada programa insertar una línea de comentario haciendo notar esta parte. (P.e. definición de variables, cuerpo principal, etc.)

En el Apéndice 2 se muestran partes del código que muestran que se cumplió con estos requisitos.

4.1.2. PLAN DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Este plan tiene como objetivos:

- Verificar que el software satisface todos los requerimientos especificados por el usuario, y
- Verificar que el software cumple con las especificaciones de diseño.

Para comprobar que cumplimos con estos dos puntos será necesario contemplar dos aspectos : Primero, en el momento de la creación de los programas fuente, cumplir con las especificaciones definidas en el diseño y seguir los estándares especificados, con lo cual se avanza por buen camino hacia un software de alta calidad, tomando en cuenta lo expuesto en el punto 1.3.4. Y segundo, diseñando y aplicando pruebas al sistema para confirmar el buen funcionamiento de todos los módulos o detectar errores y corregirlos, para posteriormente aplicar nuevamente las pruebas necesarias.

El diseño de pruebas, su aplicación y resultados se tratará más adelante, en el punto 4.2.

4.1.3. INTERFACE CON EL USUARIO

La interface con el usuario, que es la interacción entre el usuario o usuarios con un sistema de cómputo, es una parte que en ocasiones es olvidada en el diseño de los sistemas.

Para la creación del Sistema de Control de Obra, se trató de crear esta interface lo más amigable posible, es decir, de fácil manejo y con una distribución que cumpliera incluso requisitos de carácter estético, de los datos que se presentan por pantalla para poder visualizarlos sin problema.

siguientes datos:

- Nombre del programa
- Explicación breve de la(s) función(es) que realiza
- Sistema al que pertenece
- Fecha de creación
- Nombre del programador
- Fecha de última actualización, y
- Nombre de la persona que realizó la actualización.

Y en las partes importantes de cada programa insertar una línea de comentario haciendo notar esta parte. (P.e. definición de variables, cuerpo principal, etc.)

En el Apéndice 2 se muestran partes del código que muestran que se cumplió con estos requisitos.

4.1.2. PLAN DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Este plan tiene como objetivos:

- Verificar que el software satisface todos los requerimientos especificados por el usuario, y
- Verificar que el software cumple con las especificaciones de diseño.

Para comprobar que cumplimos con estos dos puntos será necesario contemplar dos aspectos : Primero, en el momento de la creación de los programas fuente, cumplir con las especificaciones definidas en el diseño y seguir los estándares especificados, con lo cual se avanza por buen camino hacia un software de alta calidad, tomando en cuenta lo expuesto en el punto 1.3.4. Y segundo, diseñando y aplicando pruebas al sistema para confirmar el buen funcionamiento de todos los módulos o detectar errores y corregirlos, para posteriormente aplicar nuevamente las pruebas necesarias.

El diseño de pruebas, su aplicación y resultados se tratará más adelante, en el punto 4.2.

4.1.3. INTERFACE CON EL USUARIO

La interface con el usuario, que es la interacción entre el usuario o usuarios con un sistema de cómputo, es una parte que en ocasiones es olvidada en el diseño de los sistemas.

Para la creación del Sistema de Control de Obra, se trató de crear esta interface lo más amigable posible, es decir, de fácil manejo y con una distribución que cumpliera incluso requisitos de carácter estético, de los datos que se presentan por pantalla para poder visualizarlos sin problema.

Para la creación de las pantallas de captura se realizaron reuniones con los usuarios del sistema para diseñarlas y, posteriormente, presentarles las ya creadas y en caso necesario realizar los cambios que se requirieran para que estas sean de fácil manejo para ellos.

De la misma forma que las pantallas, se diseñaron los reportes, poniéndolos a consideración de los usuarios para que estos sean realmente funcionales y les ayuden en el desempeño de su trabajo (una muestra de ellos se presenta en el Apéndice 3).

4.1.4. SEGURIDAD DEL SISTEMA

Para asegurar la integridad de la información que contendrá el Sistema de Control de Obras, se pensó en utilizar dos niveles de seguridad.

El primero nos lo proporciona el manejo del sistema en la Red Novell, la cual cuenta con un sistema de "password" para los usuarios dados de alta en la red, mediante el cual podemos controlar el acceso de quienes designemos, permitiéndoles o no el acceso al subdirectorío que contendrá el sistema y todos sus archivos de datos.

El segundo es a través del mismo sistema, el cual contiene un archivo de claves de acceso. Estas claves se almacenan utilizando un algoritmo de encriptamiento y sólo uno de los usuarios (el que ellos mismos designen), tendrá acceso a él para modificar claves, dar de alta nuevos usuarios o borrar alguno de los ya existentes.

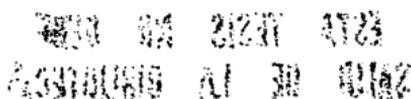
Este nivel de seguridad trabajará de la siguiente manera: el archivo de claves contendrá todas las opciones con que trabajará el sistema y cada una de ellas (que serán de tipo booleano), tendrá una bandera lógica, cierto o falso, y dependiendo de ésta, tendrán o no acceso a la opción solicitada. De esta manera podremos controlar, por cada usuario que tenga acceso al sistema, las prioridades que tendrá en cada opción y así trabajar con un grado mayor de certeza de que aquellos usuarios que así se determine no tendrán acceso a ciertas opciones del sistema.

Para determinar estas prioridades se platicará con el encargado del área de proyectos y en base a esto serán dados de alta los usuarios con los accesos y prohibiciones que se les hayan asignado.

4.2. PRUEBA

Una vez desarrollados los programas del Sistema de Control de Obras, procedimos a probar por separado cada uno de los módulos que lo integran y posteriormente, todos de manera simultánea, a través de un menú principal o programa principal del sistema.

Estas pruebas tienen como objeto encontrar, en caso de que existan, errores en la operación del sistema o incongruencias con las especificaciones de diseño. Para este efecto, utilizamos distintos tipos de pruebas, enfocados a "pulir" el sistema y poder entregar al usuario un producto confiable y eficiente.



4.2.1. TIPOS DE PRUEBAS

Las pruebas que se utilizaron para comprobar la funcionalidad y eficiencia del Sistema de Control de Obras podemos agruparlos en tres tipos generales:

1) Prueba de unidad

La prueba de unidad es algo que va de la mano con la codificación, ya que como su nombre lo indica, en ella se va probando cada unidad o programa que se genere. Su forma de trabajo está orientada hacia el tipo de prueba de la caja blanca (punto 1.3.5.). A continuación describiremos las actividades y puntos importantes a considerar dentro de esta prueba.

Esta prueba se aplica a todos los programas una vez que hayan sido desarrollados, revisados y verificados en su sintaxis. Los puntos a comprobar son los siguientes:

- recepción correcta de los datos de entrada;
- inicialización correcta de variables;
- prueba de todos los caminos independientes (caminos básicos);
- prueba de todas las decisiones lógicas en sus caras verdadera y falsa;
- el funcionamiento de los ciclos con sus límites y condiciones establecidos; y
- la correcta apertura y cierre de archivos.

2) Prueba de integración

Como se expresó anteriormente, una vez que se ha probado cada programa en forma independiente, se procederá a integrarlos en un solo sistema a través de un programa principal.

El tipo de prueba de integración que se utilizó fue descendente, es decir se comenzó con el programa principal y posteriormente se integraron los módulos que dependen de él, utilizando la forma "primero-en-profundidad".

Una vez comprobado el funcionamiento independiente de cada programa, esta prueba tiene como finalidad comprobar su correcto funcionamiento, interactuando con los demás programas que conforman el sistema. Para este efecto se diseñó un plan de prueba en el cual se especifica el orden en que cada programa se integra al sistema.

3) Prueba de validación

Al finalizar la prueba de integración, el sistema ya está integrado en un solo software, se han encontrado y corregido los errores de interfaces y se está en condiciones de proceder a las pruebas finales del software.

Este tipo de validación está orientado a la prueba de la "caja negra" (punto 1.3.5), para

demonstrar la conformidad con los requerimientos mediante la aplicación de pruebas alfa y beta.

La prueba alfa consiste en que el usuario final del sistema trabaje con él bajo la supervisión de un miembro del equipo de desarrollo y exprese si cumple o no con los requerimientos pedidos; en caso necesario, se corrigen las desviaciones en que se haya incurrido. El encargado de supervisar la prueba con el usuario deberá tomar nota de todos aquellos puntos en que se presenten problemas.

La prueba beta es muy similar a la prueba alfa, la única diferencia es que en esta prueba el usuario trabaja con el sistema sin la supervisión del equipo de desarrollo y es él mismo quien registra los problemas que encuentra (reales o imaginarios) e informará de esto.

4.2.2. APLICACION DE PRUEBAS Y RESULTADOS

Las pruebas de unidad se fueron aplicando paralelamente a la creación de todos los programas del sistema, corrigiendo los errores que se presentaron en las áreas mencionadas en el punto 4.2.1.

Algunos de los errores en los programas al aplicar las pruebas de unidad fueron los siguientes:

- errores en los nombres de variables
- inicialización errónea de variables (por omisión o tipo)
- manipulación errónea de las condiciones de salida para ciclos
- olvido de apertura y cierre de archivos
- llamados erróneos a funciones o procedimientos

Al terminar estas pruebas y corregidos todos los errores encontrados, todos los programas funcionaron correctamente en forma independiente.

Para la prueba de integración se siguió el siguiente orden a fin de acoplar todos los programas al módulo o programa principal:

1) Módulo de Ohrs

- actualización de datos de obras en proceso
- actualización de datos de partidas
- actualización de datos para cuadro comparativo "Cuadro Frío"
- impresión de reporte comparativo
- creación e impresión de gráfica de gantt

2) Módulo de avance real

- actualización de datos de avance real
- creación e impresión de gráfica de avances

3) Módulo de estimaciones

- actualización de datos de estimaciones normales
- actualización de datos de estimaciones adicionales
- actualización de datos de estimaciones extraordinarias
- impresión de reporte de estimaciones

4) Módulo de catálogo de obra

- actualización de datos de catálogos de obra
- impresión del catálogos de obra

5) Módulo de utilerfas

- respaldo de información del sistema
- actualización y mantenimiento de claves de acceso

Con esta prueba se espera que al ir acoplando cada uno de los programas al funcionamiento global, no existan errores y su interacción con el resto del sistema sea correcta. En los párrafos siguientes, sólo se presentarán los detalles de los errores o problemas encontrados al integrar cada programa; aquellos que no presentaron problemas, no serán mencionados.

Al compilar el módulo 1 (programa de impresión del cuadro comparativo), con los demás programas se presentó un función duplicada. Esta función fué la de control de impresión que, como se mencionó anteriormente, al ser común a varios programas, se incluyó en el menú principal. Sin embargo dicha función también fué incluida en este reporte para su prueba de unidad y no fué removida. Una vez eliminada del programa, éste funcionó correctamente.

El mismo problema se presentó para los demás programas de impresión (creación e impresión de la gráfica de gantt, creación e impresión de gráfica de avances, impresión de reporte de estimaciones e impresión de reporte de catálogos). En relación con los programas restantes, no presentaron ningún problema en su integración al sistema.

Para las pruebas de validación se utilizó la forma mostrada en la figura 4.1., y en ella se presenta el programa a validar, el tipo de prueba a aplicar y la descripción de la prueba, los resultados esperados en la prueba y los resultados obtenidos. En el Apéndice 4 se presentan algunas de estas formas llenas con la información de las pruebas realizadas al sistema.

En los casos en que la aplicación de estas pruebas revelen errores o problemas con el funcionamiento de los programas, en la forma de prueba también se incluirá una breve descripción del problema, su(s) causa(s) y la solución al mismo.

4.3. MANTENIMIENTO

El mantenimiento de cualquier software es una actividad que lleva mucho tiempo y que en

DESCRIPCION DE PRUEBA

SISTEMA: CONTROL DE OBRA	CLAVE: SISCOB
MODULO:	
PROGRAMA:	
TIPO DE PRUEBA:	
DESCRIPCION DE LA PRUEBA:	
ENTRADAS:	
RESULTADOS ESPERADOS:	
RESULTADOS OBTENIDOS:	

Figura 4.1. FORMATO PARA PRUEBA DE VALIDACION

ocasiones se descuida por quien lo desarrolla. Sin embargo, es algo que debe realizarse con mucho detenimiento y con una buena organización para que los esfuerzos dedicados sean útiles y exitosos.

4.3.1. PLAN DE MANTENIMIENTO

Elaborar un plan de mantenimiento específico que prevea tanto posibles errores como problemas que pudieran presentarse en el uso de un sistema es prácticamente imposible. Lo que sí podemos hacer es crear los instrumentos necesarios para el momento en que ocurra algún cambio en el software (ya sea por algún error que no descubrieron las pruebas aplicadas al sistema o por mejoras en el mismo) y mantener una documentación histórica que nos permita posteriormente consultar los cambios o mejoras aplicados al sistema, la fecha en que se realizaron, cómo se atacaron los problemas, quién se encargó de este trabajo, etc.

Para que toda esta documentación se maneje dentro de un estándar, se crearon unas formas que contienen estos datos. Estos formatos deberán ser anexados al documento que contiene el análisis y diseño del sistema para ser consultados en el momento que se requieran.

4.3.2. FORMAS DE MANTENIMIENTO

La documentación del mantenimiento se integra en 4 formas diseñadas para estandarizarla. A continuación se presenta cada una de ellas explicando cada campo.

La primera de ellas, Fig. 4.2. (Resumen de mantenimiento), contiene los datos del mantenimiento que recibe el sistema en forma general. Su llenado estará a cargo del responsable del mismo. Los datos son:

- 1.- Sistema. Nombre del sistema al que se le da mantenimiento.
- 2.- Fecha. Fecha en que se realizó el mantenimiento.
- 3.- Clave. Esta clave permite llevar un orden en los diferentes trabajos que se realicen y está formada por un número de 5 dígitos.
- 4.- Tipo de mantenimiento: correctivo o perfectivo.
- 5.- Módulo afectado. Nombre del módulo cambiado.
- 6.- Programa(s). Nombre del programa o programas que sufrieron cambios.

La segunda forma, Fig. 4.3. (Solicitud de cambio de software), se utilizará para el mantenimiento perfectivo, es decir, se realizarán cambios al sistema por mejoras que el usuario desee o por modificaciones en la forma en que se maneja la información. Esta forma será llenada por el usuario del sistema que solicita el cambio, a excepción de la clave. Los datos que contiene son:

- 1.- Sistema. Nombre del sistema al que se le harán cambios.
- 2.- Fecha. Fecha en que se solicitan estos cambios.

RESUMEN DE MANTENIMIENTO

SISTEMA : 1				
FECHA	CLAVE	TIPO DE MANTENIMIENTO	MODULO AFECTADO	PROGRAMA(S)
2	3	4	5	6

Figura 4.2. FORMA PARA RESUMEN DE MANTENIMIENTO

SOLICITUD DE CAMBIO DE SOFTWARE

SISTEMA : 1

FECHA : 2

MODULO : 3

CLAVE : 4

DESCRIPCION DEL CAMBIO SOLICITADO :

5

SOLICITADO POR :

6

Figura 4.3. FORMA PARA SOLICITUD DE CAMBIO DE SOFTWARE

- 3.- Módulo. Nombre del módulo en el cual se desean los cambios.
- 4.- Clave. Número consecutivo que le corresponda.
- 5.- Descripción del cambio solicitado. Texto explicativo del cambio que se requiere.
- 6.- Solicitado por. Nombre del usuario que solicita el cambio.

La tercera forma, Fig. 4.4. (Cambio de software), será usada para el mantenimiento correctivo, cuando se presente algún error o problema con el funcionamiento del sistema y será llenada por el responsable del mantenimiento del sistema. Los datos que maneja son:

- 1.- Sistema. Nombre del sistema en el que se corrigieron las fallas encontradas.
- 2.- Fecha. Fecha en que se reportaron los problemas en el sistema.
- 3.- Módulo. Nombre del módulo en el que se detectaron los problemas.
- 4.- Clave. Número consecutivo que le corresponda.
- 5.- Descripción del problema. Texto explicativo del problema o problemas que se detectaron.
- 6.- Reportado por. Nombre de la persona que reportó la falla.

La última forma, Fig. 4.5. (Control de cambios), guardará las actividades que se desarrollen para el arreglo del sistema o su mejoramiento. Los datos que controla son:

- 1.- Sistema. Nombre del sistema en que se trabajó.
- 2.- Fecha. Fecha en que se realizan los cambios necesarios.
- 3.- Clave. Número consecutivo que le corresponda.
- 4.- Módulo. Nombre del módulo en el que se trabajó.
- 5.- Programa(s). Nombre del programa o programas que se cambiaron.
- 6.- Descripción de cambios realizados. Texto explicativo de los cambios realizados al sistema.
- 7.- Resultados de los cambios. Reporte de cómo funciona el sistema después de realizado el mantenimiento.
- 8.- Cambios realizados por. Nombre de la persona o personas encargadas de los cambios realizados.

Esta forma se utilizará tanto para el mantenimiento correctivo como para el perfectivo, esto es, por cada forma que se llene de solicitud de cambio de software, se deberá llenar una de control de cambios la cual será archivada en la documentación del sistema.

CAMBIO DE SOFTWARE	
SISTEMA : 1	
FECHA : 2	MODULO : 3
CLAVE : 4	
DESCRIPCION DEL PROBLEMA :	
5	
REPORTADO POR : 6	

Figura 4.4. FORMA PARA REPORTAR PROBLEMAS CON EL SOFTWARE

CONTROL DE CAMBIOS

SISTEMA : 1

FECHA : 2

CLAVE : 3

MODULO : 4

PROGRAMA(S) AFECTADO(S) :

5

DESCRIPCION DE CAMBIOS REALIZADOS :

6

RESULTADO DE LOS CAMBIOS :

7

CAMBIOS REALIZADOS POR :

8

Figura 4.5. FORMATO PARA CONTROL DE CAMBIOS DE SOFTWARE

El presente informe tiene como finalidad informar a la Junta de Gobierno de la Universidad de los Andes sobre el desarrollo de las actividades de la Oficina de Asesoría Jurídica durante el periodo comprendido entre el 1 de enero de 1970 y el 31 de diciembre de 1970.

En el presente informe se detallan las actividades realizadas en materia de asesoría jurídica, representación legal, defensa de los intereses de la Universidad y en materia de control de la legalidad de los actos administrativos.

El presente informe se divide en tres partes: la primera parte trata de las actividades de asesoría jurídica, la segunda parte trata de las actividades de representación legal y la tercera parte trata de las actividades de control de la legalidad de los actos administrativos.

En materia de asesoría jurídica se han prestado servicios de asesoría a los diferentes departamentos de la Universidad, así como a los funcionarios de la misma, en materia de derecho administrativo, derecho civil, derecho penal y derecho procesal.

En materia de representación legal se han representado a la Universidad en los juicios que se han seguido en materia de responsabilidad administrativa de los funcionarios de la misma.

En materia de control de la legalidad de los actos administrativos se han emitido dictámenes de legalidad sobre los actos administrativos de los diferentes departamentos de la Universidad.

En materia de control de la legalidad de los actos administrativos se han emitido dictámenes de legalidad sobre los actos administrativos de los diferentes departamentos de la Universidad.

CONCLUSIONES

CAPITULO 5

CONCLUSIONES

En el mercado existen muchos productos de software para desarrollar diferentes tareas en diferentes áreas, pero por lo general estos paquetes trabajan ya de una forma a la cual se deben apegar los usuarios. Al desarrollar sistemas como SISCOB, el proceso se revierte, ya que aquí el sistema es el que se ajusta exactamente a las necesidades del usuario, lo cual proporciona ventajas como: no se requieren modificaciones mayores en los procedimientos, la forma en que se controla la información es prácticamente la misma (el único cambio es que ahora se maneja a través de un sistema por computadora), el entrenamiento del personal solamente es necesario en términos del manejo del sistema y no en nuevos esquemas de trabajo, etc.

Esto no quiere decir que siempre sea mejor la creación de un "sistema a la medida", pues en ocasiones existen puntos tales como costos y tiempos que obligan la utilización de software comercial, que si no se ajusta exactamente a las necesidades de una empresa, si cubre una parte importante de ellas.

En el desarrollo del sistema que aquí se presenta se tomo muy en cuenta el aspecto de seguridad, del cual ya se habló más a fondo en el capítulo 4. Sin embargo, cabe recalcar que la forma en que se manejó ha dado buenos resultados y hasta el momento no se han encontrado fisuras por las que pueda perderse información o que personal no autorizado tenga acceso a información confidencial.

Podemos afirmar que el desarrollo de este sistema contó con una base sólida para su creación, ya que el diseño del mismo se encuentra totalmente documentado, siguiendo las metodologías y técnicas expuestas en el Capítulo 2, lo cual nos permite hablar de un alto grado de confiabilidad y solidez en la operación del mismo.

La documentación creada para este sistema, conlleva un esfuerzo mínimo para su mantenimiento y repercute en que las mejoras que se requieran implementar al sistema en un futuro, se puedan realizar sin grandes contratiempos.

El desarrollo de esta tesis pretende no sólo utilizar una de las metodologías modernas para el desarrollo de software, sino aportar un ingrediente adicional a este desarrollo. Esta aportación engloba dos aspectos: la documentación de mantenimiento y la correspondiente a pruebas.

Las formas que contendrán estos datos (en ambas áreas) y el plan para realizar el seguimiento del mantenimiento realizado, se crearon específicamente para el Sistema de Control de Obras, tratando de considerar la información necesaria para poder crear un expediente histórico, siguiendo guías de que datos considerar en cada caso de la bibliografía consultada, en la cual no existen formas ya creadas para este efecto.

Este expediente con la documentación de mantenimiento y pruebas será de gran utilidad para realizar consultas posteriores y poder tomar datos de las dos áreas cuando se presenten casos similares que resolver (principalmente en mantenimiento).

El Sistema de Control de Obras tendrá una vida útil mínima de 5 años ya que su diseño modular le permite aceptar modificaciones y adiciones para adecuarse a los posibles cambios del entorno para el cual fué diseñado. sin que esto traiga consigo grandes problemas en su adaptación.

El sistema SISCOB, podrá ser utilizado fuera de la empresa para la cual fué diseñado pero se verá limitado por las diferencias en la forma en que se maneja este tipo de información ya que aunque en el diseño del sistema se trató de seguir una forma general para la realización del control de obras y las estimaciones, existen diferencias sobre todo en lo concerniente a los formatos para la presentación de información (reportes) los cuales se crearon de acuerdo a las necesidades y especificaciones presentadas por la empresa.

El diseño del sistema le asegura una gran portabilidad ya que puede trabajar en cualquier terminal de una red Novell, o si así se requiere, en una PC autónoma que cumpla con los siguientes requerimientos de equipo: 1 Mb mínimo de RAM y monitor VGA. Si se cumplen estas dos condiciones, el sistema operará sin presentar problemas en su funcionamiento.

Una limitación importante que presenta el sistema reportado en esta tesis es que sólo puede manejar 25 partidas por cada obra que se agregue al sistema. Esta limitación se presenta al dibujar la Gráfica de Gantt pues al presentarla en la pantalla, el máximo número de barras que pueden presentarse sin que se pierdan o presenten una visión antiestética es de 25. Afortunadamente para la empresa, este número es suficiente ya que revisando el desarrollo de obras anteriores el número máximo de partidas utilizadas fué de 20. Este último número de partidas se presenta porque aunque se han realizado obras de gran tamaño, por lo general éstas se dividen en partes más pequeñas para de ésta forma lograr un mejor control de las mismas.

En el caso de que se requiriera un número mayor a 25 partidas, la obra a manejar en el sistema podría dividirse en dos partes y de esta forma salvar el problema visual de pantalla ya que la información de ambas partes se complementaría.

Por otra parte, este sistema puede servir como base para realizar otros sistemas más grandes en la misma área de ingeniería civil, como por ejemplo el control de precios unitarios. lo cual va muy ligado al control de obras y las estimaciones

APENDICES

APENDICE 1

GLOSARIO DE TERMINOS

ACUMULADOS: es la cantidad que hasta determinado momento se ha pagado a la constructora encargada de la obra en cada una de las estimaciones.

AVANCE REAL: porcentaje de la obra que ya ha sido concluido y se maneja de acuerdo a información recogida en la obra misma.

CATALOGO DE OBRA: documento que contiene los diferentes conceptos a realizar dentro de una obra, con su unidad de medida y precio unitario .

CONCEPTO: es la descripción específica de materiales y trabajos a realizar en una obra. Uno o varios conceptos conforman una partida.

DESVIACION: cantidad que existe de diferencia entre los gastos reales realizados y el presupuesto asignado a una obra. Esta cantidad se expresa en cifras porcentuales.

ESPECIFICACIONES: son el conjunto de disposiciones, requisitos, condiciones e instrucciones que se establecen para la contratación y ejecución de una obra.

ESTIMACION: es el control administrativo que se lleva a cabo en una obra para realizar los pagos a las constructoras de la parte terminada de una obra.

ESTIMACION NORMAL: es el control administrativo de los pagos a las constructoras por los conceptos especificados en el contrato inicial.

ESTIMACION ADICIONAL: es el control administrativo de los pagos realizados a las constructoras cuando existen cambios en las cantidades especificadas por concepto en el contrato inicial.

ESTIMACION EXTRAORDINARIA: es el control administrativo de los pagos efectuados a las constructoras por conceptos que se contratan posteriormente a la firma del contrato inicial.

FRENTE: nombre que recibe cada una de las partes en que puede dividirse una obra para facilitar su manejo.

LICITACION: es el proceso mediante el cual se presenta a varias constructoras las especificaciones de una obra para que presenten una oferta de servicios y en base a ella elegir una de ellas para realizar los trabajos de la misma.

APENDICE I

GLOSARIO DE TERMINOS

OBRA: conjunto de operaciones y materiales que de acuerdo a las especificaciones respectivas, integran cada una de las partes en que una construcción se divide convencionalmente para fines de medición y pagos.

PARTIDA: es la clasificación que se utiliza para subdividir una obra y a su vez se compone de varios conceptos. Por ejemplo: preliminares, acabados, etc.

PRECIOS UNITARIOS: son los importes que se paga por cada unidad de material, equipo o mano de obra utilizada para una obra.

UNIDAD DE OBRA: es la unidad de medición que se señala en las especificaciones como base para cuantificar cada concepto de trabajo para fines de medición y pago.

APENDICE 2

CODIGO FUENTE

Este Apéndice contiene muestras del código fuente creado para el funcionamiento del Sistema de Control de Obras así como comentarios acerca de los estándares utilizados en su creación.

Este programa contiene las funciones necesarias para la operación del sistema en un ambiente de red.

```
.....
* Programa: Red. prg   Fecha de creación:
* Control de operaciones de red
* Sistema de Control de Obras
* Autor: Roberto Gonzalez D.
* Última Actualización :          Por : RGD
.....
* Encabezado
* del
* Programa
```

- * Función NET_USE
- * Abre un archivo para uso Exclusivo o Compartido.
- * SET INDEXes en el procedimiento si es necesario.
- * Pasa los siguientes parametros
- * 1. Caracter - Nombre de la Base de Datos a Abrir
- * 2. Logico - Realiza la apertura (exclusivo/no exclusivo)
- * 3. Numerico - Tiempo de espera en segundos (0 = Indefinidamente)

FUNCTION NET_USE

PARAMETER(S file, ex_use, wait

PRIVATE forever

forever = (wait = 0)

DO WHILE (forever OR wait > 0)

IF ex_use && apertura exclusiva

USE &file EXCLUSIVE

ELSE

USE &file && apertura compartida

ENDIF

IF .NOT. NETERR() && No hubo ningun error

RETURN (T)

ENDIF

INKEY(1) && espera 1 segundo

wait = wait - 1

ENDDO

RETURN (F) && Fallo la apertura de la Base de Datos

APENDICE 2

CODIGO FUENTE

- Función FIL_LOCK
- Trata de asegurar el actual archivo compartido
- Recibe el siguiente parametro
- 1. Numerico - segundos de espera (0 = espera hasta que lo pueda asegurar)

```
FUNCTION FIL_LOCK
PARAMETERS wait
PRIVATE forever

IF FLOCK()
    RETURN (T) && asegurado
ENDIF
forever = (wait = 0)
DO WHILE (forever OR wait > 0)
    INKEY(.5) && espera 1/2 segundo
    wait = wait - .5
    IF FLOCK()
        RETURN (.T.) && asegurado
    ENDIF
ENDDO
RETURN ( F ) && no asegurado
```

- Función REC_LOCK
- Trata de asegurar el registro actual
- Recibe el siguiente parametro
- 1. Numerico - segundos de espera (0 = espera hasta que lo pueda asegurar)

```
FUNCTION REC_LOCK
PARAMETERS wait
PRIVATE forever
IF RLOCK()
    RETURN ( T ) && registro asegurado
ENDIF
forever = (wait = 0)
DO WHILE (forever OR wait > 0)
    IF RLOCK()
        RETURN ( T ) && registro asegurado
    ENDIF
    INKEY(.5) && espera 1/2 segundo
    wait = wait - .5
ENDDO
RETURN ( F ) && no asegurado
```

APENDICE 2

CODIGO FUENTE

- Función ADD_REC
- Regresa .T. si un registro es agregado. El nuevo registro es el actual y esta asegurado
- Recibe el siguiente parámetro
- 1. Numerico - segundos de espera (0 = espera hasta que lo agrega)

```
FUNCTION ADD_REC
PARAMETERS wait
PRIVATE forever

APPEND BLANK
IF .NOT. NETERR()
    RETURN (.T)
ENDIF
forever = (wait = 0)
DO WHILE (forever .OR. wait > 0)
    APPEND BLANK
    IF .NOT. NETERR()
        RETURN .T.
    ENDIF
    INKEY(.5) && espera 1/2 segundo
    wait = wait - .5
ENDDO
RETURN (.F.) && no asegurado
```

Fragmento del programa principal del Sistema de Control de Obras.

- ```
.....
* Sistema de Control de Obras
* Programa : SCO_MP.PRG Fecha : Marzo 3, 1992
* MENU PRINC. SISTEMA DE CONTROL DE O.
* Autor : Roberto Gonzalez D.
* Ultima revision : Mayo 31, 1993 POR: RGD.
.....
```

```
SET DATE BRITISH
SET PROCEDURE TO FINCC
PUBLIC LIN, TIPO, OBRA_M {4}, OBRA_B {4}, AVAN_M {2}, AVAN_B {2}
PUBLIC UTIL_M {2}, UTIL_B {2}, ESTI_M {4}, ESTI_B {4}, CAT_M {1}, CAT_B {1}
PRIVATE PANT
TIPO=0
PARAMETERS NBC
```

## APENDICE 2

### CODIGO FUENTE

---

#### \* ARREGLOS

```
DECLARE OBRA_M [5], OBRA_B [5]
OBRA_M [1] = {OBRAS EN PROCESO}
OBRA_M [2] = {PARTIDAS}
OBRA_M [3] = {DATOS COMPARATIVO}
OBRA_M [4] = {REPORTE COMPARATIVO}
OBRA_M [5] = {GRAFICA DE GANTT}
```

```
DECLARE AVAN_M [2], AVAN_B [2]
AVAN_M [1] = 'REAL'
AVAN_M [2] = 'GRAFICA'
```

```
DECLARE ESTI_M [4], ESTI_B [4]
ESTI_M [1] = 'NORMALES'
ESTI_M [2] = 'ADICIONALES'
ESTI_M [3] = 'EXTRAORDINARIAS'
ESTI_M [4] = 'REPORTE'
```

```
DECLARE CAT_M [2], CAT_B [2]
CAT_M [1] = {CATALOGO DE CONCEPTOS}
CAT_M [2] = {REPORTE}
```

```
DECLARE UTIL_M [2], UTIL_B [2]
UTIL_M [1] = 'RESPALDO'
UTIL_M [2] = 'CLAVES'
```

```
LN24=CHR(27)+CHR(25)+CHR(24)+CHR(26)+' Movimientos del cursor '+CHR(186)+'
'+CHR(17)+CHR(196)+CHR(217)+' Selección '+CHR(186)+' <ESC> para terminar'
OPC = 0
```

```
AsignaPrioridad (OBRA_B ,3 ,7)
AsignaPrioridad (AVAN_B ,8 ,9)
AsignaPrioridad (ESTI_B ,10,13)
AsignaPrioridad (CAT_B ,14,15)
AsignaPrioridad (UTIL_B ,16,17)
```

Q=.T.

DO WHILE Q

SET COLOR TO {W+...BG}

CLEAR

@ 01,01 SAY NBC

@ 01,53 SAY "Sistema de Control de Obra"

@ 02,01 TO 04,79

## APENDICE 2

### CODIGO FUENTE

---

```
@ 02,70, 04,79 BOX [AA;UAA']
@ 03,71 SAY DATE()
@ 23,01 SAY REPLICATE(-,79)
@ 24,0 SAY LN24
@ 03,02 PROMPT [OBRAS]
@ 03,14 PROMPT [AVANCE]
@ 03,27 PROMPT [ESTIMACIONES]
@ 03,45 PROMPT [CATALOGOS]
@ 03,60 PROMPT [UTILERIAS]
MENU TO OPC
DO CASE
CASE OPC = 1
PANT = SAVESCREEN(4,1,10,21)
@ 04,01,10,21 BOX [AAA'UAA']
MENUCHOICE = ACHOICE(5,2,09,20,OBRA_M,OBRA_B)
SET CURSOR ON
DO CASE
CASE MENUCHOICE = 1
LIN = SAVESCREEN(3,1,3,14)
DO SCO_OBRA
CASE MENUCHOICE = 2
DO SCO_PART
CASE MENUCHOICE = 3
DO SCO_COMP
CASE MENUCHOICE = 4
DO SCO_CUFR
CASE MENUCHOICE = 5
DO GANTT
CASE MENUCHOICE = 0
RESTSCREEN(4,1,10,21,PANT)
TECLAS ()
ENDCASE
RESTSCREEN(4,1,10,21,PANT)
CASE OPC = 2
PANT = SAVESCREEN(4,13,7,21)
@ 04,13, 7,21 BOX [AAA'UAA']
OPC2 = ACHOICE(5,14,6,20,AVAN_M,AVAN_B)
SET CURSOR ON
DO CASE
CASE OPC2 = 1
DO SCO_AVAN
CASE OPC2 = 2
DO GRA_AVA
```

## APENDICE 2

### CODIGO FUENTE

---

```
CASE OPC2 = 0
 RESTSCREEN(4,13,7,21,PANT)
 TECLAS ()
ENDCASE
RESTSCREEN(4,13,7,21,PANT)
CASE OPC = 3
PANT1 = SAVESCREEN(4,26,9,42)
@ 4,26, 9,42 BOX [ÁÁÁ'ÚÁÁ']
OPC3 = ACHOICE(5,27, 8,41, ESTI_M,ESTI_B)
SET CURSOR ON
DO CASE
 CASE OPC3 = 1
 DO SCO_ESTN
 CASE OPC3 = 2
 DO SCO_ESTA
 CASE OPC3 = 3
 DO SCO_ESTE
 CASE OPC3 = 4
 DO SCO_REST
 CASE OPC3 = 0
 RESTSCREEN(4,26,9,42,PANT1)
 TECLAS ()
ENDCASE
RESTSCREEN(4,26,9,42,PANT1)
CASE OPC = 4
PANT = SAVESCREEN(4,45,7,67)
@ 04,45, 7,67 BOX [ÁÁÁ'ÚÁÁ']
OPC4 = ACHOICE(5,46,6,66,CAT_M,CAT_B)
SET CURSOR ON
DO CASE
 CASE OPC4 = 1
 DO SCO_CAT
 CASE OPC4 = 2
 DO SCO_RCAT
 CASE OPC4 = 0
 RESTSCREEN(4,45,7,67,PANT)
 TECLAS ()
ENDCASE
RESTSCREEN(4,45,7,67,PANT)
CASE OPC = 5
PANT = SAVESCREEN(4,59,7,68)
@ 04,59, 7,68 BOX [ÁÁÁ'ÚÁÁ']
```

## APENDICE 2

### CODIGO FUENTE

---

```
OPC5= ACHOICE(5,60,6,67,UTIL_M,UTIL_B)
SET CURSOR ON
DO CASE
 CASE OPC5 = 1
 DO SCO_RESP
 CASE OPC5 = 2
 @ 4,59 CLEAR TO 4,69
 @ 4,59 SAY REPLICATE ('X',11)
 DO CLAVES
 CASE OPC5 = 0
 RESTSCREEN(4,59,7,68,PANT)
 TECLAS ()
ENDCASE
RESTSCREEN(4,59,6,68,PANT)
CASE OPC = 0
 FINCC ()
ENDCASE
ENDDO
SET COLOR TO
CLEAR
RETURN
```

CUADRO FRIO

| CONCEPTOS                                           | COSTA                | CONSTRUCCION         | ICA                  | REPARACION COSTAS    |  |  |  |  |
|-----------------------------------------------------|----------------------|----------------------|----------------------|----------------------|--|--|--|--|
|                                                     |                      | IMP                  |                      | REC. IMPUESTOS       |  |  |  |  |
| <b>POSTEROS</b>                                     |                      |                      |                      |                      |  |  |  |  |
| 1 TRABAJOS PRELIMINARES                             | 54,240.00            | 4,560.00             | 4,562.00             | 158,404.22           |  |  |  |  |
| 2 ALERQUELERIA                                      | 3,215.00             | 3,420.00             | 434.00               | 22,310.21            |  |  |  |  |
| 3 ACABADOS                                          | 15,404.00            | 620.00               | 2,654.00             | 254,541.50           |  |  |  |  |
| 4 REJERERIA                                         | 4,824.00             | 3,907.00             | 3,364.00             | 454,214.63           |  |  |  |  |
| 5 ESTRUCTURAS                                       | 92,254.00            | 70,560.00            | 4,564.00             | 454,821.98           |  |  |  |  |
| 6 INSTALACION ELECTRICA                             | 70,254.00            | 2,634.00             | 89,774.00            | 222,454.78           |  |  |  |  |
| 7 MOBILIARIO                                        | 70,254.00            | 4,321.00             | 70,740.00            | 120,234.23           |  |  |  |  |
| 8 LIMPIEZAS Y ACABOS                                | 12,000.00            | 51,000.00            | 92,546.00            | 342,456.63           |  |  |  |  |
| IMPORTE DE LA OBRA ANTES DE IVA                     | 399,110.00           | 133,250.00           | 302,982.00           | 2,774,282.37         |  |  |  |  |
| APLICACION DE IVA                                   | 20,911.00            | 15,225.00            | 20,200.30            | 227,428.33           |  |  |  |  |
| IMPORTE TOTAL DE LA OFERTA C/IVA                    | 510,021.00           | 148,475.00           | 323,182.30           | 3,001,710.70         |  |  |  |  |
| PORCENTAJE DE PARTICIPACION                         | 30.00 %              | 30.00 %              | 25.00 %              | 25.00 %              |  |  |  |  |
| TIEMPO DE EJECUCION DE OBRA                         | 120 Dias Calendarios | 200 Dias Calendarios | 120 Dias Calendarios | 180 Dias Calendarios |  |  |  |  |
| IMPONES EN LA LICITACION                            | NINGUNO              | NOY                  | NINGUNO              | NINGUNO              |  |  |  |  |
| ANTECEDENTES CON LA EMPRESA                         | NINGUNO              | NOY                  | NINGUNO              | NINGUNO              |  |  |  |  |
| CALIDAD EN LA PRESTACION DE LA OFERTA               | NINGUNO              | NOY                  | NINGUNO              | NINGUNO              |  |  |  |  |
| RECURSOS INGRESOS Y PLANOS DE LA LICITACION         | SI                   | SI                   | SI                   | NOY                  |  |  |  |  |
| RECORRER LOS ANTES DE INICIAR EL CONSTRUCCION       | SI                   | SI                   | SI                   | SI                   |  |  |  |  |
| PRESENTAR ANALISIS ANALISIS: PROYECTO, F. SENSIBLE  | SI                   | SI                   | SI                   | SI                   |  |  |  |  |
| PRESENTAR INFORMES DE P.A.                          | SI                   | NO                   | SI                   | SI                   |  |  |  |  |
| PRESENTAR INFORME POR CREDITO DURANTE LA LICITACION | NO                   | SI                   | NO                   | SI                   |  |  |  |  |
| VERIFICACION DE LA OFERTA                           | 30 Dias Calendarios  | 15 Dias Calendarios  | 24 Dias Calendarios  | 34 Dias Calendarios  |  |  |  |  |

CONCEPTO DE ESTIMACIONES  
DEPARTAMENTO DE COSTOS

Contrato : 001-01 PLAZA LA HACIENDA

HOJA : 1  
FOLIO : 12/01/94

| EST. NO. | CANT. M <sup>2</sup> | FECHA    | PERIODO DE EJECUCION | ESTIMACIONES NOMINALES |           | AVANTAJIZACION DE ANTICIPO |           | NETO PAGADO EN FACTURA |           | ESTIM. ESTIMOPROHIBIDAS |           | NETO PAGADO EN FACTURA |           | SOLDO POR EJERCER |              |
|----------|----------------------|----------|----------------------|------------------------|-----------|----------------------------|-----------|------------------------|-----------|-------------------------|-----------|------------------------|-----------|-------------------|--------------|
|          |                      |          |                      | PARCIAL                | ACUMULADO | PARCIAL                    | ACUMULADO | PARCIAL                | ACUMULADO | PARCIAL                 | ACUMULADO | PARCIAL                | ACUMULADO | PARCIAL           | ACUMULADO    |
| 001-A    | 250                  | 26-05-92 | 15-05-92             | 26-05-92               | 80,000.00 | 100,000.00                 | 20,000.00 | 20,000.00              | 77,000.00 | 77,000.00               |           |                        | 77,000.00 |                   | 4,757,666.45 |
| 001-B    | 750                  | 18-08-92 | 01-08-92             | 18-08-92               | 80,711.22 | 200,711.22                 | 60,215.27 | 160,215.27             | 24,547.64 | 231,267.64              |           |                        | 24,547.64 |                   | 4,331,118.81 |
| 001-C    | 150                  | 15-11-92 | 01-11-92             | 15-11-92               | 80,000.00 | 600,711.22                 | 90,000.00 | 190,215.27             | 31,000.00 | 462,267.64              |           |                        | 31,000.00 |                   | 4,290,118.81 |
| 001-D    | 250                  | 15-11-92 | 01-11-92             | 15-11-92               | 50,000.00 | 650,711.22                 | 17,670.00 | 197,885.27             | 45,252.00 | 507,500.04              |           |                        | 45,252.00 |                   | 4,244,716.81 |
| A. I. E. | 250                  | 15-01-92 | 01-01-92             | 15-01-92               |           | 629,411.22                 |           | 197,885.27             |           | 507,500.04              | 15,000.00 | 15,000.00              | 16,200.00 | 16,200.00         | 4,244,716.81 |
| 001-E    | 150                  | 15-01-92 | 17-01-92             | 15-01-92               |           | 629,411.22                 |           | 197,885.27             |           | 507,500.04              | 84,000.00 | 99,000.00              | 97,000.00 | 100,000.00        | 4,244,716.81 |
| 001-F    | 250                  | 15-02-92 | 01-02-92             | 15-02-92               | 1,200.00  | 660,611.22                 | 247.97    | 198,246.24             | 921.62    | 508,822.26              |           |                        | 99,000.00 | 100,000.00        | 4,243,824.19 |

## CATALOGO DE OBRA

DEPARTAMENTO DE COSTOS

CARRANZO 140191 PLAZA LA HACIENDA

HOJA : 1

### 1.00 TRABAJOS PRELIMINARES

- 1.01 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 50,254.00  
 DESMONTAJE Y DEMOLICION DE FALDA PLAFON DE METAL DESPLEGADO CON APLAFADO DE YESO Y ACABADO CON TIROL RUSTICO EN PASILLOS DE ACCESO A LOCALES COMERCIALES (ESTANDO EL AREA LIBRE DE ELEMENTOS DE SOTOPERTERA). INCLUYE CARGA Y ACARREO DE
- 1.02 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 68,525.00  
 DEMOLICION DE FIMPE DE CONCRETO SIMPLE EN BANQUETAS Y GUARNICION. INCLUYE CARGA Y ACARREO DE ESCOMBRO FUERA DE LA
- 1.03 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 63,485.00  
 DEMOLICION DE MURO DE TABIQUE PISO RECOCIDO 7 X 14 X 28 A UNA ALTURA NO MAYOR DE 5M. INCLUYE CARGA Y ACARREO DE ESCOMBRO FUERA DE LA OBRA.
- 1.04 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 65,258.00  
 DEMOLICION DE LA LOSA DE CONCRETO DE 18 CM. DE ESPESOR A UNA ALTURA NO MAYOR DE 5.00 MTS. INCLUYE TEJA, CARGA Y ACARREO DE ESCOMBRO FUERA DE LA OBRA.
- 1.05 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 726.50  
 PREPARACION DE PISO EXISTENTE PARA RECIBIR LOSETA DE CERAMICA. INCLUYE PICADO, LIMPIEZA, CARGA Y ACARREO DE ESCOMBRO FUERA DE LA OBRA.
- 1.06 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 1,645.00  
 PREPARACION DE MURO DE TABIQUE PISO RECOCIDO DE 7 X 14 X 28 CM. PARA RECIBIR PANEL CONVITEC, CONSISTENTE EN RETIRO DE TEJA DE BAPTELIZADO + PERFORACION DE 1A Y 1A EN MURO.
- 1.07 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 491.00  
 TRAZO Y NIVELACION PARA DESPLANTE DE ESTRUCTURAS ESTABLECIENDO REFERENCIAS DEFINITIVAS CON MEDIOS MANUALES.
- 1.08 UNIDAD : M<sup>3</sup> PRECIO UNITARIO : 5,987.60  
 PREPARACION DE COLUMNAS PARA RECIBIR NUEVA ESTRUCTURA, A BASE DE DEMOLICION DE CONCRETO PARA ENCONTRAR ARMADO, INCLUYE ALARRES DE ESCOMBRO FUERA DE LA OBRA.
- 1.09 UNIDAD : M PRECIO UNITARIO : 15,584.00  
 PROTECCION EN ANDADONES A BASE DE TABLALES DE MADERA DE PINO A UN ALTURA DE 3.00 M. S.N.P.T... INCLUYE MATERIALES, MONTAJE.
- 1.10 UNIDAD : PZA PRECIO UNITARIO : 216,529.12  
 CIMENTACION PARA ESCALERA A BASE DE CONCRETO f'c=250kg/cm<sup>2</sup> ARMADO CON ACEPO DE REFUERZO INDICADO EN PLANOS, INCLUYE FUNDACION, HERRAMIENTAS DE ACERO Y DESPERDICIOS, CIMENTADO, Y COMPACTACION.
- 1.11 UNIDAD : M<sup>2</sup> PRECIO UNITARIO : 12,291.00  
 RETIRO DE HERRAMIENTAS EN ACIEROS, INCLUYE, MANO DE OBRA, HERRAMIENTA, LIMPIEZA DE ENGRILLADO Y ACARREO FUERA DE LA OBRA.

## DESCRIPCION DE PRUEBA

|                                                                                                                                                                                                              |                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| SISTEMA: CONTROL DE OBRA                                                                                                                                                                                     | CLAVE : SISCOB |
| MODULO: OBRAS EN PROCESO                                                                                                                                                                                     |                |
| PROGRAMA: ACTUALIZACION DE DATOS DE OBRAS EN PROCESO                                                                                                                                                         |                |
| TIPO DE PRUEBA: FUNCIONAL                                                                                                                                                                                    |                |
| DESCRIPCION DE LA PRUEBA:<br><br>Se probaran las opciones de alta, baja, modificacion y consulta de diferentes datos de obras en proceso, revisando su correcta actualizacion en el archivo correspondiente. |                |
| ENTRADAS:<br><br>Datos generales de 10 obras en construccion.                                                                                                                                                |                |
| RESULTADOS ESPERADOS:<br><br>Se espera que la actualizacion de datos en la base de datos sea correcta en cada una de las opciones a probar.                                                                  |                |
| RESULTADOS OBTENIDOS:<br><br>Las altas, bajas y modificaciones trabajaron correctamente. Esto se verifico con la opcion de consulta del mismo programa y contra el archivo correspondiente.                  |                |

## DESCRIPCION DE PRUEBA

|                                                                                                                                                                                                   |                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| SISTEMA: CONTROL DE OBRA                                                                                                                                                                          | CLAVE : SISCOB |
| MODULO: CATALOGOS DE OBRA                                                                                                                                                                         |                |
| PROGRAMA: ACTUALIZACION DE DATOS DE CATALOGOS DE OBRA                                                                                                                                             |                |
| TIPO DE PRUEBA: FUNCIONAL                                                                                                                                                                         |                |
| DESCRIPCION DE LA PRUEBA:<br><br>Se daran de alta y de baja conceptos y subconceptos de catalogos de obra y se realizaran algunas modificaciones sobre los datos capturados.                      |                |
| ENTRADAS:<br><br>Se alimentara el sistema con dos catalogos de obra con todos sus conceptos y subconceptos.                                                                                       |                |
| RESULTADOS ESPERADOS:<br><br>El programa debera actualizar correctamente las altas y bajas del catalogo, permitiendo las modificaciones que le sean solicitadas.                                  |                |
| RESULTADOS OBTENIDOS:<br><br>La actualizacion de datos en altas, bajas y modificaciones fue correcta. La informacion dada de alta fue almacenada correctamente en el archivo que le correspondia. |                |

## DESCRIPCION DE PRUEBA

|                                                                                                                                                                                                                                                                                                                                          |                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| SISTEMA: CONTROL DE OBRA                                                                                                                                                                                                                                                                                                                 | CLAVE : SISCOB |
| MODULO: OBRAS EN PROCESO                                                                                                                                                                                                                                                                                                                 |                |
| PROGRAMA: ACTUALIZACION DE DATOS DE OBRAS EN PROCESO                                                                                                                                                                                                                                                                                     |                |
| TIPO DE PRUEBA: TENSION Y DESEMPEÑO                                                                                                                                                                                                                                                                                                      |                |
| DESCRIPCION DE LA PRUEBA:<br><br>Se accedera en una terminal el archivo de datos de obras en proceso mientras que en otras dos se intenta el acceso al mismo archivo para comprobar el funcionamiento en red y que los mensajes que informan al usuario que el archivo no esta disponible por el momento sean desplegados correctamente. |                |
| ENTRADAS:                                                                                                                                                                                                                                                                                                                                |                |
| RESULTADOS ESPERADOS:<br><br>Que el sistema bloquee el acceso al archivo de obras en proceso despues de que el primer usuarios lo haya accedido y no les permita a los demas el acceso enviandoles el mensaje correspondiente.                                                                                                           |                |
| RESULTADOS OBTENIDOS:<br><br>El funcionamiento en red fue correcto, los usuarios que intentaron acceder posteriormente al primero no tuvieron acceso al archivo y recibieron correctamente el mensaje de aviso.                                                                                                                          |                |

## DESCRIPCION DE PRUEBA

|                                                                                                                                                                                                                                                                                                                                         |                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| SISTEMA: CONTROL DE OBRA                                                                                                                                                                                                                                                                                                                | CLAVE : SISCOB |
| MODULO: OBRAS EN PROCESO                                                                                                                                                                                                                                                                                                                |                |
| PROGRAMA: ACTUALIZACION DE DATOS DE PARTIDAS                                                                                                                                                                                                                                                                                            |                |
| TIPO DE PRUEBA: FUNCIONAL                                                                                                                                                                                                                                                                                                               |                |
| DESCRIPCION DE LA PRUEBA:<br><br>Se daran de alta, baja y modificaran datos de partidas para verificar su correcta actualizacion.                                                                                                                                                                                                       |                |
| ENTRADAS:<br><br>Partidas de trabajo de dos obras diferentes.                                                                                                                                                                                                                                                                           |                |
| RESULTADOS ESPERADOS:<br><br>El programa debiera actualizar correctamente la informacion en la base de datos correspondiente en las diferentes opciones para su manejo (alta, baja, modificacion).                                                                                                                                      |                |
| RESULTADOS OBTENIDOS:<br><br>Operacionalmente el programa trabajo bien pero en uno de los reemplazos de la informacion no lo hizo dejando ese campo vacio.<br><br>El problema se presento debido a un error en una asignacion a una variable se corrigio este problema y se volvio a aplicar la prueba obteniendo resultados positivos. |                |

## DESCRIPCION DE PRUEBA

|                                                                                                                                                                                                                                                                                                              |                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| SISTEMA: CONTROL DE OBRA                                                                                                                                                                                                                                                                                     | CLAVE : SISCOB |
| MODULO: OBRAS EN PROCESO                                                                                                                                                                                                                                                                                     |                |
| PROGRAMA: IMPRESION DE CUADRO COMPARATIVO (CUADRO FRIO)                                                                                                                                                                                                                                                      |                |
| TIPO DE PRUEBA: TENSION Y DESEMPEÑO                                                                                                                                                                                                                                                                          |                |
| DESCRIPCION DE LA PRUEBA:<br><br>Se imprimira este reporte para comprobar la exactitud de los datos que proporciona y su correcto funcionamiento.                                                                                                                                                            |                |
| ENTRADAS:<br><br>Datos de cinco empresas que participen en una licitacion.                                                                                                                                                                                                                                   |                |
| RESULTADOS ESPERADOS:<br><br>La impresion del reporte debera proporcionar correctamente los datos capturados y la distribucion de la informacion debera ser de acuerdo a los lugares que le correspondan.                                                                                                    |                |
| RESULTADOS OBTENIDOS:<br><br>La informacion que presento el reporte fue correcta. En la distribucion se presento un error en las coordenadas del renglon del total de la obra.<br><br>Una vez corregidas estas coordenadas se imprimio nuevamente el reporte y en esta ocasion la distribucion fue correcta. |                |

---

## **BIBLIOGRAFIA**

## Bibliografía

- Lientz B., E. Swason, Software Maintenance Management  
Addison-Wesley, 1980
- S. Pressman, Roger. Ingeniería del Software un enfoque práctico  
McGraw Hill, 1988
- Swason E. B., The Dimensions of Maintenance  
Conf. Software Engineering, IEEE, Octubre 1976, pp. 492-497.
- Freedman D., G. Weinberg, Techniques of Program and System Maintenance  
G. Parikh, Ed., Winthrop Publishers, 1981
- Fairley, Richard, Ingeniería de Software  
McGraw Hill, 1989
- Jackson, M. A., Principles of Program Design  
Academic Press, 1975
- Warnier J. D., Logical Construction of Programs  
Van Nostrand Reinhold, 1974
- Wirth Niklaus, Program Development by Stepwise Refinement  
Cacm, Vol. 14, No. 4, 1971
- Myers G. Composite Structured Design  
Van Nostrand, 1978
- McCall J., P. Richards and G. Walters, Factor in Software Quality  
NTIS, 1977
- Myers G., The Art of Software Testing  
Wiley, 1979