

34  
2ej.



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA INTEGRAL DE INFORMACION DE ADMINISTRACION DE OBRAS DEL COMITE ADMINISTRADOR DEL PROGRAMA FEDERAL DE CONSTRUCCION DE ESCUELAS

## T E S I S

QUE, PARA OBTENER EL TITULO DE INGENIERO EN COMPUTACION PRESENTAN:  
ALEJANDRO GARCIA PALACIOS  
GUSTAVO ADOLFO LOPEZ MANJARREZ  
JORGE LUIS GUEVARA GONZALEZ  
NORMA PATRICIA MUÑOZ VILLALVA  
RUBEN RIVERA MARTINEZ

DIRECTOR DE TESIS:

ING. HECTOR ARMANDO LOPEZ PINEDA

MEXICO, D. F.

AGOSTO 1992



TESIS CON FALLA DE ORIGEN



## **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.

## CONTENIDO

I.- INTRODUCCION.	1
II.- GENERALIDADES.	3
II.1. ANTECEDENTES.	7
II.1.1. VENTAJAS DE LOS SISTEMAS ANTERIORES.	10
II.1.2. DESVENTAJAS DE LOS SISTEMAS ANTERIORES.	12
II.2. DEFINICION DEL PROBLEMA.	15
II.3. ALTERNATIVAS DE SOLUCION.	16
III.- ANALISIS.	19
III.1. PLANEACION DEL DESARROLLO DEL SISTEMA INTEGRAL DE ADMINISTRACION DE OBRAS.	20
III.2. SISTEMA DEL PROGRAMA GENERAL DE OBRA.	23
III.3. SISTEMA DE CONCURSO DE OBRA.	29
III.4. SISTEMA DE ADQUISICIONES Y ABASTECIMIENTO.	36
III.5. SISTEMA DEL ALMACEN CENTRAL.	47
III.6. DICCIONARIO DE DATOS.	51
III.7. NIVELES Y CONSECUENCIAS DE LOS CAMBIOS QUE IMPLICARA EL NUEVO SISTEMA.	60
IV.- DISEÑO DEL ESQUEMA DE SOPORTE DE LAS APLICACIONES (BACK-END).	63
IV.1. DISEÑO LOGICO.	64
IV.1.1. DIAGRAMA ENTIDAD RELACION.	65
IV.1.2. DIAGRAMA DE TABLAS Y VISTAS.	67
IV.1.2.1. NORMALIZACION.	69
IV.1.2.2. DESNORMALIZACION.	69
IV.1.3. INTEGRIDAD DE LOS DATOS.	70
IV.1.4. DISEÑO DE BASE MULTIPLES.	86
IV.1.5. DICCIONARIO DE DATOS.	91

IV.2. DISEÑO FISICO.	95
IV.2.1. DETERMINACION DE INDICES.	95
IV.2.2. TAMAÑOS DE LAS BASES DE DATOS.	96
IV.2.3. ARQUITECTURAS.	104
IV.2.3.1. CENTRALIZADA.	104
IV.2.3.2. CLIENTE-SERVIDOR.	106
IV.2.3.3. MULTISERVIDOR.	108
IV.2.4. SEGURIDAD DEL SISTEMA.	111
V.- DISEÑO DE LAS APLICACIONES DE INTERFAZ CON EL USUARIO (FRONT-END).	115
V.1. DISEÑO DE FORMAS DE CAPTURA.	116
V.2. ALGORITMOS DE PROCEDIMIENTOS GENERALES.	128
V.3. DISEÑO DE REPORTES.	133
V.3.1. GENERADOR DE REPORTES.	133
V.3.2. REPORTES CON CLIENTE ABIERTO.	133
VI.- IMPLANTACION Y PRUEBAS.	137
VI.1. FASE CENTRALIZADA.	138
VI.2. FASE CLIENTE-SERVIDOR.	141
VI.3. FASE MULTISERVIDOR Y SISTEMAS ABIERTOS.	143
VII.- CONCLUSIONES.	145
BIBLIOGRAFIA.	
APENDICES.	
GLOSARIO.	

# I.-INTRODUCCION.

En México, para que la educación impartida por el Gobierno Federal se ajuste a los procesos sociales, debe enfrentarse a una amplia problemática: La construcción y mejoramiento de la infraestructura instalada, la implementación de programas de actualización académica, la reorientación de recursos financieros y humanos destinados al apoyo de la educación, etc., son actividades encaminadas a dotar al país de los recursos humanos con la capacidad necesaria para afrontar los retos de un contexto mundial cada vez más competitivo.

Dentro de este contexto, el Comité Administrador del Programa Federal de Construcción de Escuelas (C.A.P.F.C.E.), como entidad dependiente de la Secretaría de Educación Pública (S.E.P.) y en apoyo a los programas de egresos autorizados por la Secretaría de Hacienda y Crédito Público (S.H.C.P.), tiene como objetivo organizar, dirigir y llevar a cabo los programas federales de construcción, equipamiento y habilitación de inmuebles e instalaciones destinadas al servicio de la educación en general.

En el marco de las políticas de modernización y descentralización de la educación pública, el C.A.P.F.C.E. ha impulsado la integración informática de sus áreas funcionales con la finalidad de cumplir con el objetivo básico para el cual fué creado.

Para llevar a cabo sus funciones, el C.A.P.F.C.E. debe realizar un adecuado seguimiento de las actividades inherentes a la administración de las obras desde su planeación hasta su terminación. Así pues, considerando esta problemática, el presente trabajo pretende ser un apoyo computarizado, capaz de mantener de forma confiable y eficiente la relación de información entre las gerencias que participan en la Administración de Obras.

## **II.- GENERALIDADES.**

El objetivo del C.A.P.F.C.E., es atender las necesidades de construcción, reparación y equipamiento de escuelas a nivel nacional, basados en las especificaciones acordadas con autoridades de la S.E.P. y de acuerdo a los programas de egresos autorizados por la S.H.C.P.

El C.A.P.F.C.E. está dividido principalmente en Oficinas Centrales y Jefaturas de Zona (una por cada estado). Para las Oficinas Centrales se tienen dos subdirecciones: la técnica y la administrativa, cada una conformada a su vez de varias Gerencias.

La Subdirección Administrativa, como su nombre lo indica, es la encargada de todos los aspectos jurídicos, contables, administrativos que conciernen al C.A.P.F.C.E. Por otro lado, la Subdirección Técnica participa en la programación, evaluación, abastecimiento, realización y supervisión de las obras. De esta última, uno de los procesos más importantes que existen es el de la Administración de las Obras.

La Administración de Obras inicia cuando en las diferentes localidades de cada uno de los estados de la República se reciben peticiones de construcción, equipamiento o reparación de inmuebles escolares. De esta manera cada Jefatura de Zona recaba y envía a las Oficinas Centrales tales peticiones, las cuales llegan a la Gerencia de Planeación y Programación. En ésta, se analiza cada petición y en base a aspectos financieros y técnicos se realiza una propuesta costeadada oficial que se presenta a la S.H.C.P., para que más tarde ésta autorice el presupuesto requerido por cada obra.

La Gerencia de Ingeniería de Costos recibe de las Jefaturas de Zona el conjunto de obras que han sido autorizadas y concursadas para que mediante los procedimientos técnicos indicados proponga el dictamen a la Subdirección Técnica y así se asigne al contratista ganador.



En base al análisis técnico de los requerimientos de construcción y equipamiento de las obras, se determinan los materiales necesarios (mobiliario, equipo y prefabricados) para cumplir con las metas propuestas en la fase de planeación de cada obra.

Una vez determinadas las necesidades de cada obra, éstas son reportadas a la Gerencia de Adquisiciones, la cual se encarga de investigar, analizar, organizar y efectuar los procedimientos de compra de materiales y equipo para, posteriormente, planear y determinar la forma en la cual se distribuirá el material adquirido.

Después los materiales que fueron adquiridos y pedidos se entregan por parte de los proveedores en el almacén central, en el almacén de jefatura de zona o directamente en la obra. En el caso de que la entrega se realice en el almacén central (o Gerencia de Almacenes), éste se encargará de recibir, custodiar y enviar el material a cada jefatura de zona de acuerdo a las instrucciones indicadas por la Gerencia de Adquisiciones. La estructura orgánica del C.A.P.F.C.E. se presenta en la figura I.1.

**ESTRUCTURA ORGANICA DEL  
C.A.P.F.C.E**

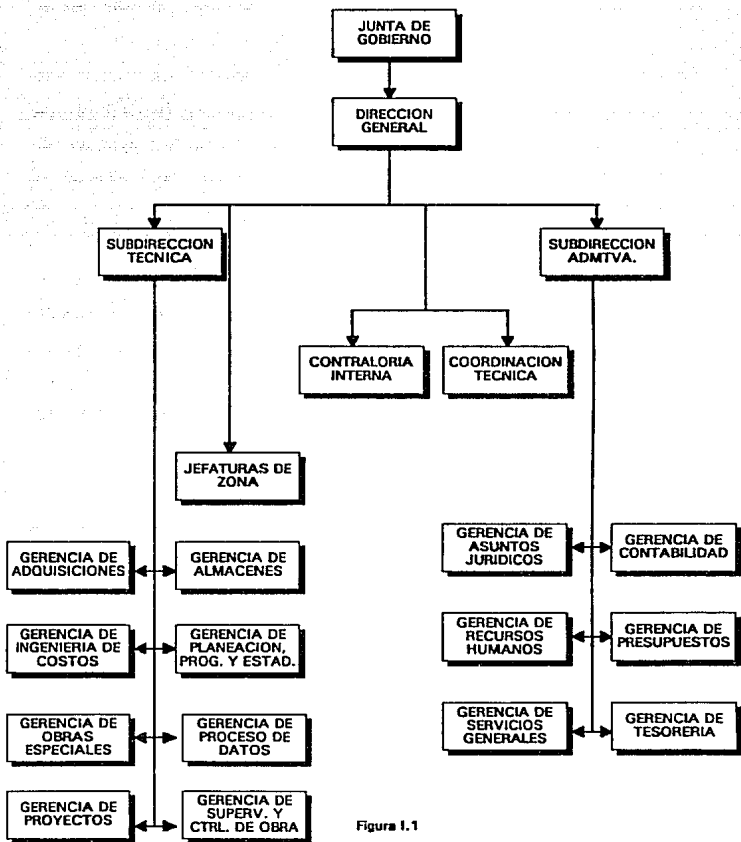


Figura 1.1

## II.1.- ANTECEDENTES

El C.A.P.F.C.E. ha evolucionado gradualmente en la incorporación de actividades y procedimientos manuales al uso del equipo de cómputo. Así, las diversas áreas funcionales que intervienen en la Administración de Obras, han desarrollado sistemas de información independientes que solucionan temporalmente sus necesidades y que carecen de los elementos necesarios para integrarlos modularmente en un sistema general. No obstante las restricciones anteriormente mencionadas, este tipo de sistemas ofrecen una gran versatilidad en cuanto al manejo de información de cada Gerencia; rapidez de respuesta a nuevas necesidades, un entorno aislado donde cada área conoce de su información y solo se preocupa de la misma sin tener presente la interrelación entre las diversas áreas, facilidad de mantenimiento cuando cambian los procedimientos administrativos, etc, constituyen las principales ventajas de este tipo de sistemas aunque su costo de desarrollo y mantenimiento es alto.

Desde el punto de vista informático, la Administración de Obras se inicia desde el momento en que se autoriza la construcción de una obra hasta que se entrega a la Entidad solicitante. De esta manera, la Gerencia de Planeación con el sistema del Programa General de Obra (P.G.O.) reporta a la Gerencia de Adquisiciones las necesidades de cada obra, asimismo, genera la información requerida para determinar la composición de los elementos que constituyen dicha construcción, la cantidad de recursos erogados en un ejercicio (Finiquito) y los recursos consumidos en un período determinado del mismo (Transferencia).

La Gerencia de Ingeniería de Costos registra los datos inherentes a los concursos de obras realizados: obras, costos y contratistas a nivel nacional.

La Gerencia de adquisiciones basada en el Sistema Integral de Control de Adquisiciones y

Almacenes (SINCAA), cuya finalidad original fue proporcionar un sistema que integrara las gerencias de Ingeniería de Proyectos, Adquisiciones y Almacenes, se encarga de registrar las licitaciones y a los ganadores de las mismas para posteriormente, realizar los pedidos correspondientes e informar al almacén central del material que deberá recibir. En el área de abastecimiento, habiendo ya determinado las necesidades de las obras, se procede a realizar órdenes de envío por obra a cada Jefatura de Zona, tomando en cuenta las disponibilidades de las mismas. Esta información también se envía al Almacén Central para que éste realice los envíos correspondientes y retroalimente a Adquisiciones con la información pertinente que le permita estimar correctamente los envíos a realizar.

El Almacén Central apoyado en la computadora, realiza las actividades de control del material que está por enviar, así como del material que se ha enviado; registrar y controlar las entradas por pedidos, registro de las salidas efectuadas por la coordinación de almacenes y de las existencias físicas.

Estas actividades le permiten retroalimentar a la Gerencia de Adquisiciones de los elementos necesarios para determinar el adeudo de proveedores y el adeudo a obra.

En la figura II.1 se muestra esquematizado el flujo de información entre las gerencias implicadas en el proceso de la Administración de Obras.

DIAGRAMA GENERAL DE FLUJO DE INFORMACION

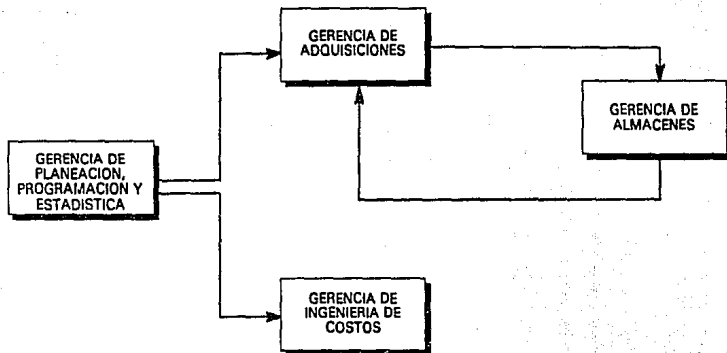


Figura II.1

## II.1.1.- VENTAJAS DE LOS SISTEMAS ANTERIORES.

Dentro de todo el proceso que implica la Administración de Obras, existen algunas actividades que de alguna forma se apoyan en la computadora. Estos apoyos están implementados en diferentes herramientas tanto de *hardware* como de *software*. Es difícil distinguir las ventajas que nos proporcionen un panorama para percibir los antecedentes de un intento de elaboración de un sistema integral. Sin embargo conviene destacar las ventajas que tiene cada uno de los ambientes en los que se desarrollaron los sistemas, las desventajas que se mencionaron con anterioridad nos proporcionan un punto de referencia para poder describir varios sistemas que en un sólo ambiente tienen varios elementos en común.

El sistema de P.G.O. (Programa General de Obra), el cual tiene la virtud, desde el punto de vista técnico, de tener para su desarrollo herramientas de software fabricadas en el C.A.P.F.C.E. para formar estándares tanto en su interfase con el usuario (es decir pantallas y formas de captura), como en la misma estructura de los ABC's (Altas, Bajas y Cambios), dándole así un carácter de uniformidad y por lo tanto un fácil mantenimiento y perspectivas sencillas para su crecimiento. Desde el punto de vista usuario, presenta las ventajas de proporcionar un ambiente amigable.

Para el abastecimiento de mobiliario, equipo, y material prefabricado se cuenta con uno de los intentos más cercanos a crear un sistema integral de información en lo que respecta la mayoría de los procedimientos principales de la Administración de Obras, ya que a pesar de que su enfoque estaba más orientado al control de compras, adquisiciones y almacén se tiene un diseño más o menos abierto que permite, de alguna forma, tener un control de las obras autorizadas, ya que su esquema esta implementado en un modelo relacional apoyado en un manejador de bases de datos relacional.

En conclusión, este último sistema es un intento formal de integración de información. Sin

embargo. en la actualidad su función se ha reducido a proporcionar órdenes de envío de materiales y de controlar el abastecimiento a nivel obra en el cual no sólo se generan reportes importantes sino que también contiene algunas consultas por pantalla. Desde el punto de vista técnico también hay que destacar que se desarrollaron herramientas parecidas a las que se tenían para el sistema de P.G.O. tales como generadores de ABC's, generadores de formas de captura y de menús.

Como se mencionó anteriormente, el sistema de cómputo dejó de utilizarse para fases tales como la adquisición del mobiliario y equipo, así como para el control del almacén debido a la lentitud de su operación. Es por ello que para proporcionar apoyo a estas gerencias se desarrollaron sistemas que están implementados en computadoras personales, los cuales presentan la ventaja de tener un mantenimiento relativamente sencillo, además de que atienden a las necesidades del usuario de manera inmediata, ya que el desarrollo de dichos sistemas se ha ido ajustando a las necesidades conforme se van presentando. Esto último puede considerarse de alguna forma una ventaja, ya que el usuario se siente satisfecho por el apoyo recibido.

## **II.1.2.-DESVENTAJAS DE LOS SISTEMA ANTERIORES.**

Todos los sistemas que apoyan de una u otra forma a que el proceso de Administración de Obras se reduzca en tiempo y esfuerzo, en algunas de sus fases presentan desventajas comunes a pesar de que están desarrolladas en las más variadas herramientas.

### **a) Redundancia en la información.**

Esto se debe básicamente a la relativa independencia que presentan los sistemas. Por ejemplo, para el caso del P.G.O., se tienen procedimientos en donde se cargan las necesidades de las obras a niveles más bajos de información (es decir la información a detalle que se proporciona acerca de una obra) cuando por otro lado se tienen procesos en los que se carga la obra a su nivel más global y de los cuales se desglosa con ayuda de los catálogos para obtener el detalle de cada una de las obras. Es decir, se duplica la información y en algunos casos no concuerda con la información a nivel global. También en el caso del sistema de concursos se carga información correspondiente a la obra la cual puede tomarse de la información que previamente se ha cargado en el catálogo de obras en ejercicio, lo mismo sucede para muchos otros catálogos. Toda esta redundancia produce que la información proporcionada por los sistemas a un nivel gerencial tenga un alto grado de incertidumbre, obligando así a que los procedimientos manuales se lleven a cabo para darle seguridad a la información que más se acerque a la obtenida.

### **b) Ambiente monousuario.**

La mayoría de los sistemas que existen se utilizan en una forma monousuario. Esto ha provocado que muchas de las tareas apoyadas en computadora se hayan vuelto insuficientes para toda la necesidad de información que se requiere en un momento dado. Esto ocurre, incluso en los sistemas que están implementados en un ambiente multiusuario, y esto es debido a que fueron diseñados de una forma que no se aprovecharon las ventajas que existen en el sistema operativo y que



quizá debido a una mala información no se implementó; éste es el caso concreto para los sistemas en manejadores de archivos en VAX. Para el sistema que esta implementado en el manejador de bases de datos existe la posibilidad de trabajar en multiusuario, sin embargo, eso ocasiona que el sistema se degrade en tiempo de respuesta, ya sea debido a una mala implementación técnica del manejador, a que el *hardware* donde estaba implementado era insuficiente, o a que su diseño detallado no fue terminado de una forma cuidadosa.

Además, el diseño transaccional de las aplicaciones multiusuario que se ejecutan en la base de datos es deficiente lo que motiva frecuentes conflictos de acceso para poder hacer uso de la información y en consecuencia altos tiempos de respuesta.

**c) Manejo de *batch*.**

Al degradarse la eficiencia de los sistemas, se tuvo que depender del analista para convertir los procesos de actualización y recurrir al procesamiento en *batch*. Esta problemática se presenta principalmente en los sistemas de concursos, adquisiciones y abastecimiento, debido a que las transacciones en línea tienen un alto costo en tiempo. Es por ello que se dejaron de utilizar en muchas de sus funciones volviéndose así inoperante.

**d) Inexistencia del Diccionario de Datos.**

No existe un diccionario de datos (a pesar de que se tenía la herramienta), en el cual todos los sistemas pudieran referirse a los mismos campos con el mismo nombre, o que por lo menos el mismo tipo de información existiera en los mismos archivos, perdiendo así, una de las formas normales básicas.

**e) Escasa documentación de los sistemas.**

En todos los sistemas no existe ningún tipo de documentación actualizada, además de que la

poca documentación existente no sigue un estándar definido, con lo cuál el análisis de éstos se vuelve una tarea complicada para considerar perspectivas de crecimiento, e incluso de mantenimiento.

**f) Dependencia a nivel datos.**

Esta característica la presentan todos los sistemas basados en archivos así como también los que están en herramientas más avanzadas, ya que el hecho de modificar algo de la estructura del diseño del sistema convierte en un caos el actualizar todos los programas.

**g) Ineficiente sistema de consultas.**

Esto se refiere que para tener acceso a la información de manera ordenada sólo se puede realizar en papel y sólo en formatos ya establecidos, con lo cuál se pierde la versatilidad de las consultas. Es decir que para realizar consultas inesperadas se tiene siempre que programar la aplicación, repercutiendo esto en las decisiones gerenciales, que generalmente claman por una respuesta ágil y veraz.

## II.2.- DEFINICION DEL PROBLEMA.

Los sistemas computarizados del C.A.P.F.C.E. padecen de grandes deficiencias. La deficiencia de mayor importancia es que se mantiene en uso un sistema para cada gerencia involucrada en la Administración de Obras; es por ello que cada una de las mismas maneja información particular, lo que convierte al sistema en un conjunto de programas sin la adecuada relación entre ellos, originando grandes deficiencias en la administración del C.A.P.F.C.E. además de que para el adecuado funcionamiento de los programas existe una alta dependencia del usuario hacia el personal del área de cómputo.

Es por este motivo que se hace presente la necesidad de la realización de un sistema computarizado que permita un manejo de información homogénea, que le proporcione a cada área usuaria una herramienta de apoyo, organización y simplificación de los trabajos realizados por ellos teniendo una interfase máquina-usuario amigable y sencilla tanto para el trabajo cotidiano como para la toma de decisiones y consultas a nivel gerencial; además de compartir recursos de información común entre dos o más áreas funcionales con una mínima participación y dependencia del personal especializado en donde el propio usuario sea responsable del manejo de su información.

### **II.3.- ALTERNATIVAS DE SOLUCION**

De acuerdo a la problemática presentada anteriormente, se pretende cubrir los siguientes requerimientos informáticos del C.A.P.F.C.E., de tal forma que la solución contenga las siguientes características :

- Información común a las diferentes áreas del C.A.P.F.C.E.
- Tiempos de respuesta eficientes.
- Integridad de la información.
- Una interfase usuario-máquina amigable.
- Herramientas que permitan un desarrollo rápido y eficiente.

Las alternativas que pudiesen cubrir las características antes mencionadas son las siguientes:

- 1) La compra de un sistema que ya exista en el mercado que satisfaga las necesidades del C.A.P.F.C.E.

Por las características de la información que genera el C.A.P.F.C.E. y su manejo no hacen factible esta alternativa ya que en el mejor de los casos las aplicaciones existentes sólo cubren parcialmente la necesidad de alguna gerencia del C.A.P.F.C.E. Las hojas de cálculo electrónicas son un ejemplo notable del uso de este tipo de herramientas que ofrecen una gran versatilidad, pero no los elementos suficientes que permitan la integración de las áreas del C.A.P.F.C.E.

- 2) La adaptación de los sistemas existentes para lograr la integración informática del C.A.P.F.C.E.

La depuración de los sistemas de información actuales representa otro camino

a seguir, sin embargo, por los problemas ya descritos en secciones previas, su adecuación representaría una fuerte inversión en tiempo que no contribuiría a corregir diseños ya rebasados por las nuevas necesidades de información en la Institución.

### 3) El desarrollo de un sistema integral.

Esta opción implicaría retomar desde los diseños de los sistemas actuales para ajustarlos a uno que dé como resultado un sistema integral; o bien, diseñar completamente algún de ellos.

Esta alternativa representa en primera instancia un costo mayor en tiempo, sin embargo, considerando las desventajas en puntos anteriores, es muy probable que esta opción sea implementada en un tiempo menor, ya que se cuenta ahora con una nueva herramienta que tiene características que facilitan las etapas del desarrollo. Además, en *hardware* se cuenta con una infraestructura que proporciona ventajas para resolver la problemática planteada de una manera eficaz. Es decir, al escoger esta alternativa, se utilizarían los recursos informáticos que posee el C.A.P.F.C.E., de tal forma, que además de proporcionar una solución a los requerimientos planteados, se incorporaría nueva tecnología al desarrollo de sistemas informáticos del C.A.P.F.C.E.

A continuación se listan los recursos mencionados:

#### ▪ Sistemas operativos.

. SCO UNIX V.3.2.0.

. VMS, versiones 5.02 y 5.43.

#### ▪ Compiladores.

. C.

\* **Manejadores de bases de datos.**

. Sybase.

\* **Herramientas para la generación de aplicaciones de Sybase.**

. Data work bench.

. Application productivity tools.

. DB library.

\* **Equipo de *hardware*.**

. Minicomputadores VAX: 11/780 y 4000.

. Equipos WYSE 80386.

. Terminales WYSE y DEC configurables.

. Unidades de cinta DIGITAL, además de unidades de cartucho TK70.

. Impresoras de línea EPSON, de alta velocidad y HP Laserjet III.

La evaluación financiera de cualquier sistema es sin duda necesaria, sin embargo, en este sistema no es determinante, debido a que ya existen factores humanos, materiales y técnicos que permiten su realización sin que represente erogación alguna.

### III.- ANALISIS

### III.1. PLANEACION DEL DESARROLLO DEL SISTEMA INTEGRAL DE ADMINISTRACION DE OBRAS.

La adecuada planeación en el desarrollo de cualquier proyecto conlleva el estudio de las actividades a desarrollar. En el área de sistemas las tareas se dividen de acuerdo al ciclo de vida del *software*: Planeación, análisis de requerimientos, análisis de sistemas, diseño, implementación, pruebas y mantenimiento.

La planeación del sistema obedece a factores operativos y está determinada por el adecuado conocimiento de las herramientas a emplear tanto en *hardware* como en *software*. De este planteamiento se desprenden las siguientes premisas:

- . Es prioritario mantener en operación a los sistemas actuales.
- . Es necesario compenetrarse con el *hardware* y *software*.
- . En la instrumentación del sistema no se contempla como meta principal la propuesta de procedimientos administrativos mejores sino su adaptación al uso de la computadora, partiendo entonces, del análisis de los sistemas en producción.
- . El sistema no tendrá una planificación temporal sólo una planificación organizacional que distribuya los esfuerzos necesarios para su consecución.

Se muestra en la figura III.1. el diagrama 0, donde se presenta al proceso central SIAO (Sistema Integral de Información de Administración de Obras).

El análisis partirá del diagrama de la figura III.2, el cual es el primer desglose del nivel cero, que describe de una manera gráfica los procesos más importantes en los que se ha decidido dividir el análisis del sistema completo.



## DIAGRAMA DE CONTEXTO

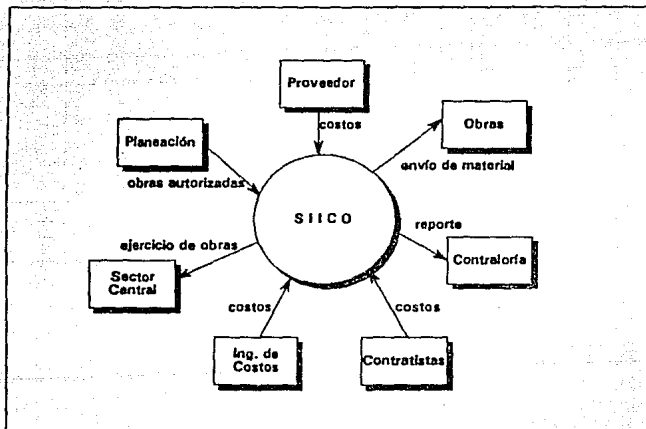


Figura III.1

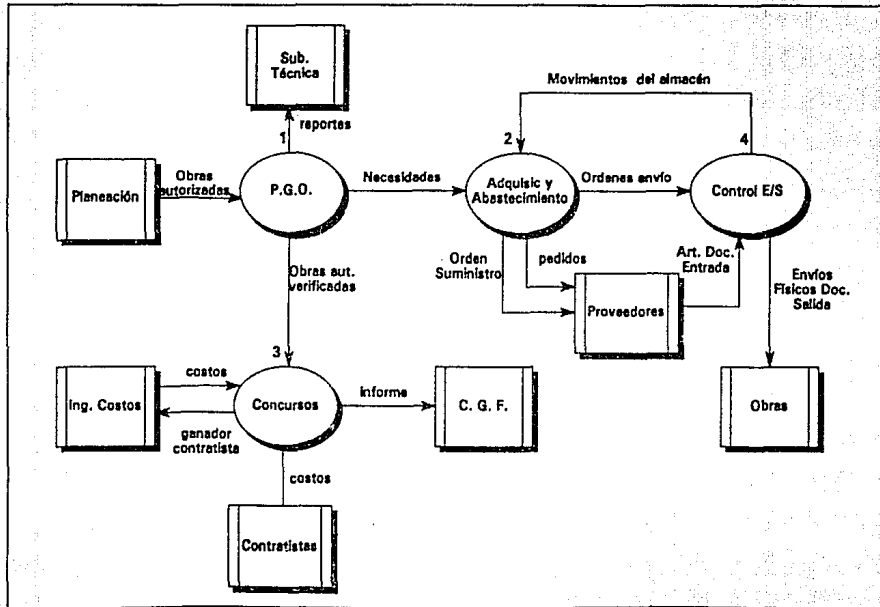


Figura III.2

### III.2. SISTEMA DEL PROGRAMA GENERAL DE OBRA.

La Gerencia de Planeación, al igual que otras gerencias del C.A.P.F.C.E., ha incorporado gradualmente sus sistemas de información manual al uso de la computadora.

La formulación del Programa General de Obra por parte de la Gerencia de Planeación y las Jefaturas de Zona dan pie al inicio de actividades del C.A.P.F.C.E. Esta Gerencia también esta encargada de dar seguimiento al presupuesto asignado por la S.H.C.P. mediante evaluaciones e informes en algunos períodos del ejercicio, (Transferencias y Finiquito).

Las actividades de esta Gerencia se dividen en tres áreas principales, a saber :

- Planeación.
- Programación.
- Estadística.

La realización de obras requiere, por una parte, de la participación de las Jefaturas de Zona las cuales formulan sus propuestas de realización de obras las cuales son turnadas a la S.H.C.P. para su análisis y aprobación del presupuesto correspondiente.

En este renglón, el sistema del Programa General de Obra (P.G.O.) ofrece un informe respecto a los recursos financieros asignados a cada obra en los rubros de construcción, mobiliario, equipo y prefabricado además de información inherente al tipo de inmueble que se construirá o habilitará. Este sistema también ofrece la posibilidad de estimar los egresos que se realizarán en el siguiente ejercicio después de que se han registrado las obras propuestas por las Jefaturas de Zona.

Por otra parte, el seguimiento y evaluación de la realización de obras, requiere de informes

periódicos a la S.H.C.P. con objeto de verificar el adecuado cumplimiento de las metas del C.A.P.F.C.E. sobre el ejercicio presupuestal.

El sistema de P.G.O. mediante información proporcionada por las Jefaturas de Zona detalla sobre los recursos erogados en cada entidad federativa por cada nivel educativo y programa de financiamiento, con la finalidad de evaluar el avance en las metas de construcción del C.A.P.F.C.E.

Otra de las áreas importantes atendidas por el sistema P.G.O. es la que se refiere a las necesidades de materiales prefabricados, mobiliario y equipo, las cuales más tarde son registradas en el sistema de adquisiciones.

## **ARCHIVOS Y PROCESOS**

### **- Catálogos:**

**Obras.**

**Municipios.**

**Nivel educativo.**

**Estructuras.**

**Capítulos.**

**Programas de Financiamiento.**

**Autorizaciones de Obra.**

**Elementos.**

**Gufas.**

**Artículos.**

**Estados.**

**- Transacciones:**

Elementos por Obra.

Gufas por Obra.

Artículos por Obra.

Modelos por elemento.

Gufas por elemento.

**- Procesos:**

Finiquito.

Transferencia.

Explosión de obras.

Prorrato.

**- Reportes:**

Combinatorio de P.G.O. ( por ejemplo TESPO ).

Explosión de mobiliario.

TESPO interno.

Finiquito.

Transferencia final.

Catálogos ( gufa de mobiliario y equipo)

En las figuras III.3, III.4.a y III.4.b. se muestra el diagrama de flujo operativo del procedimiento descrito anteriormente y el diagrama de flujo de datos, respectivamente.

**DIAGRAMA DE FLUJO OPERATIVO PLANEACION**

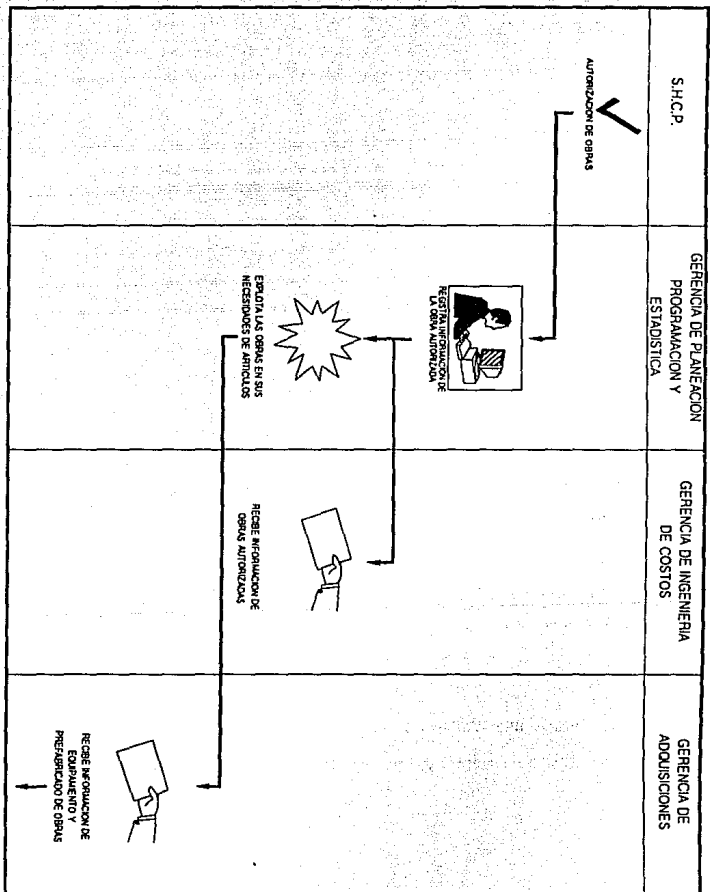


Figura III.3.

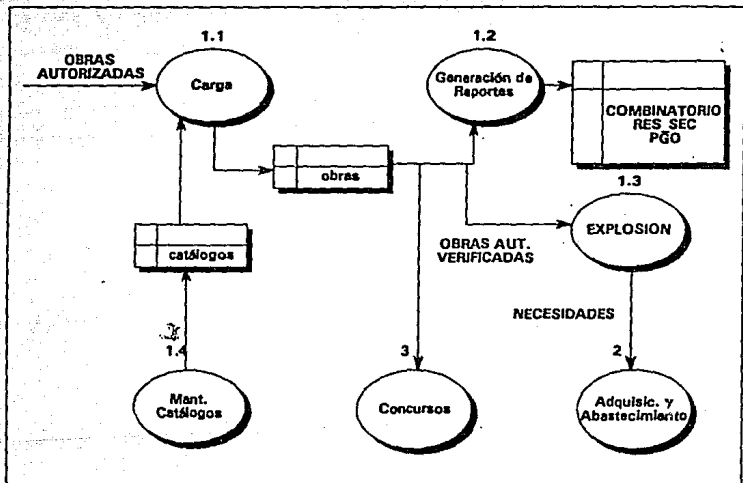


Figura III.4.a

# EXPLOSION

Proceso 1.3  
Nivel 2

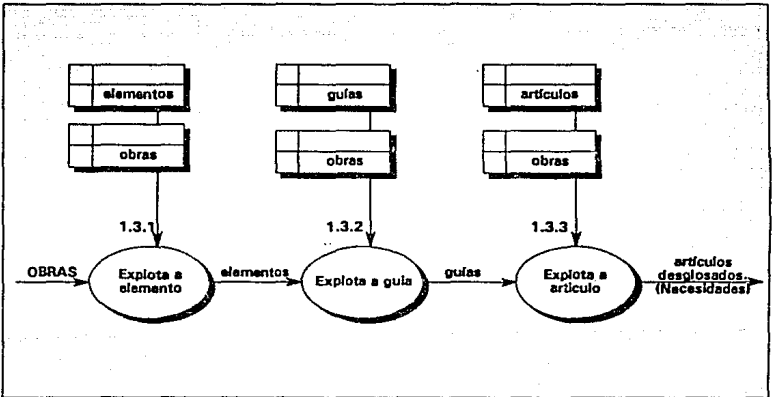


Figura III.4.b



### **III.3. SISTEMA DE CONCURSO DE OBRA.**

La Gerencia de Planeación, Programación y Estadística informa las obras que ya fueron autorizadas a cada Jefatura de Zona y a las demás gerencias. Una vez autorizadas, cada Jefatura de Zona procede a realizar el Concurso de Obras bajo la autorización de la Gerencia de Ingeniería de Costos.

Existen dos tipos de Concursos: Públicos y Directos. El tipo de concurso depende del monto total de construcción asignado. En ellos, se establecen los límites de montos para cada tipo de asignación de obras, lo cual está determinado por la Ley de Obras Públicas.

El Concurso Directo o de Invitación conforma la mayor cantidad de la información total de Concursos. Mientras que solo una parte mínima de concursos son Públicos; en éstos participan todas las constructoras, ya que por ser Público debe de realizarse una Convocatoria en los diarios. A diferencia del Concurso Público, en el Concurso Directo, ya se tiene una lista de contratistas propuestos por la Jefatura de Zona.

Procedimiento de ejecución de concursos de obras.

#### **1.- Formación de Paquetes.**

En cada Jefatura de zona se establecen los paquetes de obras. Estos paquetes constan de una o más obras y generalmente obedecen a sus características y a su localización geográfica.

#### **2.- Elaboración de la propuesta.**

Los paquetes formados en cada Jefatura de Zona son enviados a Oficinas Centrales, específicamente a la Gerencia de Ingeniería de Costos como una propuesta de concursos.

### 3.- Asignación del número de concurso.

Conforme van llegando los paquetes a la Gerencia de Ingeniería de Costos, se les va asignando un número consecutivo de concurso, lo cual le da un carácter de autorización al concurso.

### 4.- El aviso a la Secretaría de la Contraloría General de la Federación de Concursos Programados.

La Gerencia de Ingeniería de Costos tiene que avisar a la Secretaría de la Contraloría General de la Federación los concursos que van a realizarse en cada Jefatura de Zona.

Una vez autorizado el paquete, cada Jefatura de Zona lleva a cabo el Concurso. En el concurso generalmente se inscriben las constructoras o contratistas, pero solo algunas de ellas participan de manera activa, es decir, la constructora asiste al Concurso presentando el monto total a nivel concurso y a niveles más detallados de información (pago de salarios (peón, etc), costos unitarios de materiales (acero, cemento, etc), y conceptos varios, etc). En caso de que no se presentaran los contratistas, el concurso se declara desierto o cancelado y se avisa a la Gerencia de Ingeniería de Costos.

De esta manera la Jefatura de Zona revisa cada oferta de los contratistas, verificando totales, y que todos los niveles de información pedidos a los contratistas estén completos. Así la Jefatura de Zona ordena de menor a mayor los importes de las ofertas, enviando a la Gerencia de Ingeniería de Costos los tres primeros lugares y último.

### 5.- Fallo del Concurso.

La Gerencia de Ingeniería de Costos al recibir la información antes mencionada, procede a realizar varios cálculos como son:

- a) Costo C.A.P.F.C.E. (Costo por metro cuadrado de estructura por edificio).

- b) Porcentaje de desviación entre el Costo C.A.P.F.C.E. y el Costo propuesto a nivel concurso por el Contratista.
- c) Puntuación total obtenida por contratista.
- d) Determinación del lugar ocupado en esta última evaluación.

Estos datos obtenidos son depositados en el Esquema de Dictamen. El resto de la información es proporcionada por la Jefatura de Zona. Con los parámetros estimados para la evaluación del contratista, se procede a fallar el concurso, es decir se determina al ganador o bien se rescinde el concurso (no existe ganador).

Generalmente el ganador es el que ocupa el primer lugar en cuanto al presupuesto menor, sin embargo el ganador puede ser cualquier otro, si es que el primer lugar tuviera algunas otras obras asignadas y que el monto de ellas fuese muy elevado.

#### 6.- Aviso a la Secretaría de la Contraloría General de la Federación de los concursos fallados.

Una vez que se ha determinado al ganador se avisa a la Jefatura de Zona para que proceda a realizar el contrato de obra y además se avisa a la Secretaría de la Contraloría General de la Federación a qué contratista o constructora se le asignó la(s) obra(s) o bien que concursos se cancelaron.

#### ARCHIVOS Y PROCESOS

##### - Catálogos:

Contratistas.

Estructuras.

Estados.

**- Transacciones:**

Obras y edificios.

Concurso y ganador.

**- Procesos:**

Carga de lote de obras falladas.

**- Reportes:**

Contratistas ganadores en orden alfabético.

Contratistas ganadores por estado.

Costo por metro cuadrado de estructuras por estado.

Costo por metro cuadro de estructuras a nivel nacional.

En las figuras III.5.a., III.5.b. y III.6 se muestran los diagramas de flujo operativo y el diagrama de flujo de datos, respectivamente.

### DIAGRAMA DE FLUJO OPERATIVO DE COSTOS

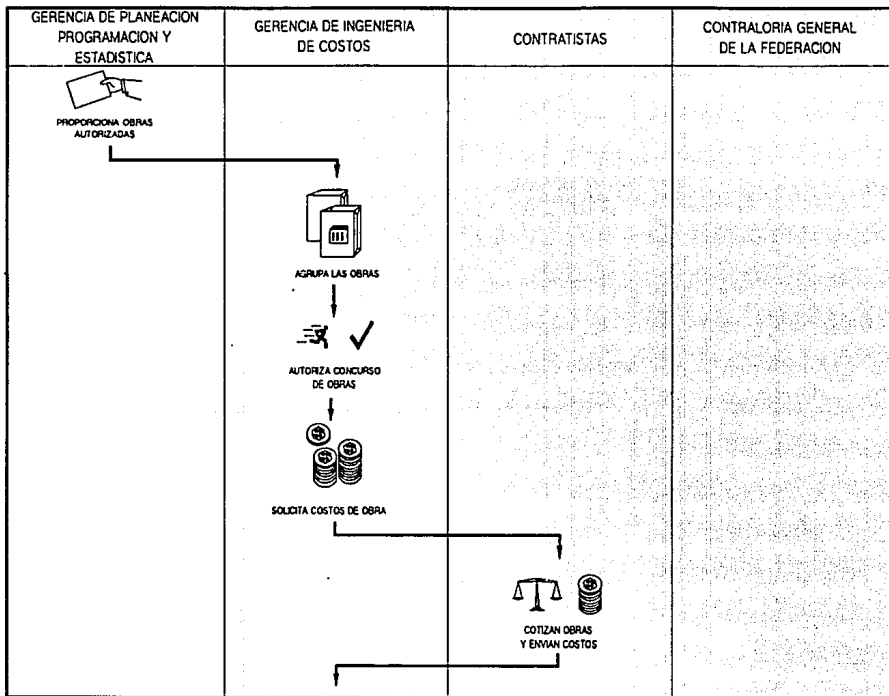


Figura III.5.a.

### DIAGRAMA DE FLUJO OPERATIVO DE COSTOS

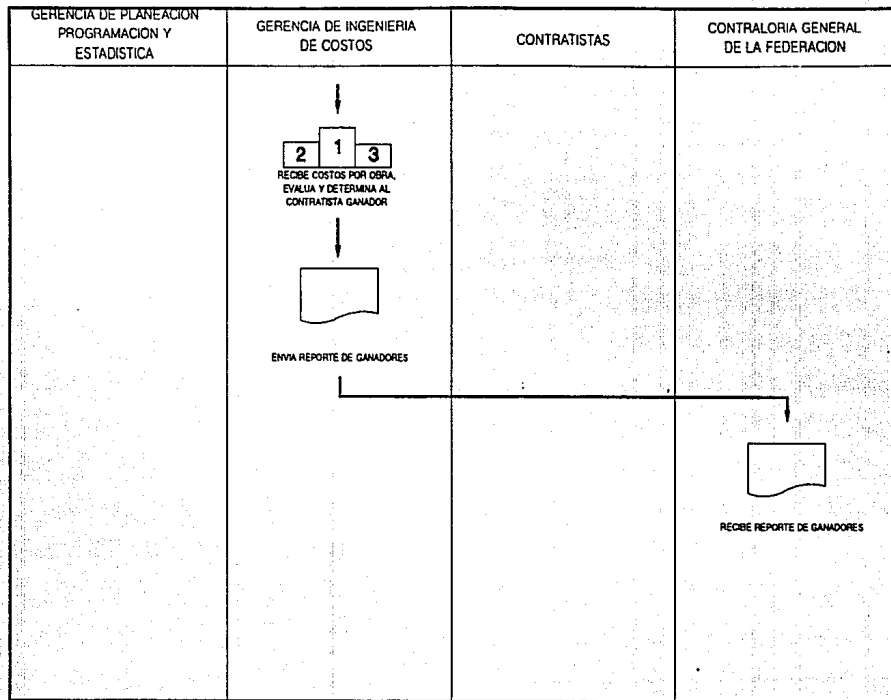


Figura III.5.b.

CONCURSOS

Proceso 3  
Nivel 1

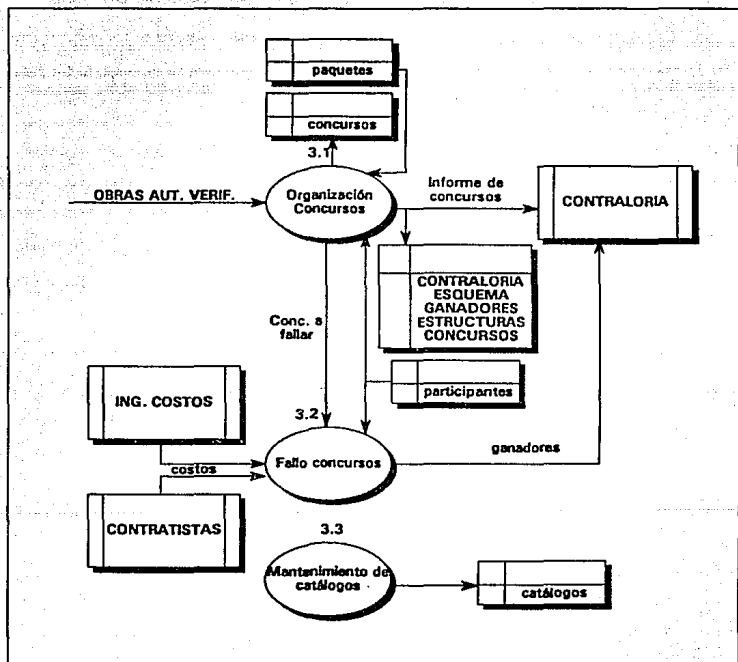


Figura III.6

### III.4. SISTEMA DE ADQUISICIONES Y ABASTECIMIENTO.

La Gerencia de Adquisiciones para llevar a cabo sus funciones de abastecimiento de materiales a las obras ha evolucionado en sus métodos de trabajo utilizando desde sistemas manuales de registro y flujo de información hasta los sistemas computarizados. La transición ha contemplado dos fases sistematizadas fundamentales:

- Operación manual.

En esta etapa, el intercambio de información entre las diferentes áreas funcionales involucradas en el abastecimiento de materiales (Planeación, Adquisiciones y Almacenes) se realiza mediante escritos mecanografiados o a través de formas diseñadas para cada función.

Así, la oficina representante del C.A.P.F.C.E. en cada Jefatura de Zona, formula la lista de elementos (prefabricado y equipamiento) requeridos por el conjunto de obras que se realizan en su jurisdicción. Posteriormente, la Jefatura de Zona solicita a la Gerencia de Adquisiciones que autorice al Almacén Central el envío de materiales mediante un documento denominado requisición, en el cual se especifican los elementos necesarios a enviar y su cantidad.

Más tarde, a través del conocimiento de las unidades disponibles en las Jefaturas de Zona, se procede a realizar, en caso de ser necesario, la adquisición de materiales o el abastecimiento.

La Gerencia de Adquisiciones publica en los principales diarios del país la convocatoria a la licitación, en la cual se plantean los requerimientos a los posibles participantes. Dichos participantes, una vez inscritos, concursan por los artículos que ofertan, proporcionando para ello todos los datos requeridos por el C.A.P.F.C.E. Cada oferta deberá especificar su contenido, características y



condiciones. Cabe mencionar que ningún proveedor podrá inscribirse si no reúne las características que el C.A.P.F.C.E. necesita, lo cual a su vez se apega a la normatividad señalada por la Ley de Adquisiciones y Servicios.

El procedimiento de licitación concluye con el fallo de la licitación que determina al proveedor ganador. Una de las características de este fallo es que no puede ser sistematizado, debido a que no existe un patrón definido que determine a un ganador, pues se manejan parámetros no determinados, por ejemplo, la calidad de un artículo va en relación a un precio, por lo cual no siempre el menor precio es el aceptado.

Determinados los ganadores, se procede a formular los pedidos. Para tener un estricto control y debido a que por lo general no se hace un pedido por la cantidad total ganadora, se contabilizan los pedidos anteriores, hasta completar la cantidad total.

Otra forma de hacer adquisición de material, es mediante la designación directa de un proveedor específico, lo cual se hace en base al historial que se mantiene en el C.A.P.F.C.E. de los pedidos anteriores hechos a su nombre.

El abastecimiento de materiales se realiza bajo dos modalidades: a) Ordenes de abastecimiento y b) Ordenes de Suministro. En la primera de ellas se consideran los elementos disponibles en cada uno de los almacenes de Jefatura de Zona, en tanto que la segunda tiene como finalidad el facilitar el manejo y la entrega de materiales voluminosos directamente a la obra que los requiere ya que en este caso es el proveedor el que realiza la entrega en cualquiera de las Entidades de la República. En ambos casos, la Gerencia de Adquisiciones autoriza al Almacén Central a enviar en forma total o parcial el material que requieren el conjunto de obras de la Jefatura de Zona.

- Operación computarizada.

Generación de reportes de sumarización de necesidades para la Gerencia de Planeación. La Gerencia de Planeación apoyada por el equipo de cómputo produce una relación de los artículos que necesita el conjunto de obras que requiere una Entidad Federativa; ésta se envía al Almacén Central el cual a su vez, envía los materiales a la Jefatura de Zona en la medida de sus existencias, esta operación se realiza mediante la captura de la información propia de cada jefatura.

Los esfuerzos por llevar a cabo un suministro más eficiente de materiales a las obras han conducido a implementar un esquema de abastecimiento por obra. En él, la Gerencia de Planeación informa a la Gerencia de Adquisiciones sobre las necesidades de cada obra que integran al Programa General de Obra (P.G.O). Más tarde, habiendo determinado las sustituciones de artículos debidas a la obsolescencia de los mismos se formulan órdenes de envío y suministro por obra las cuales consideran la disponibilidad de elementos en cada jefatura de zona. La determinación de disponibilidades de artículos en este contexto es fundamental y tiene como fuente de alimentación en caso del Almacén Central las entradas de artículos originadas por los pedidos realizados anteriormente, el inventario inicial y los artículos que han dejado de formar parte del abastecimiento a una obra debido a la cancelación o modificación de ésta.

La integración de las áreas de Planeación, Almacenes y Adquisiciones constituye uno de los logros más importantes del sistema SINCAA (Sistema Nacional de Control de Adquisiciones y Almacenes) el cual tiene como finalidad distribuir los elementos disponibles en cada almacén de acuerdo a las necesidades de cada obra y de redistribuir los materiales en caso de que exista alguna variación de las obras.

El C.A.P.F.C.E. construye y habilita alrededor de 10,000 inmuebles escolares en cada ejercicio de operación y cada obra tiene asociados varios artículos que están registrados en diversas fases de la construcción, desde su determinación como necesidad hasta su entrega en jefatura de zona por el Almacén Central, todas estas fases involucran el manejo masivo de información que tiene que ser validada por el manejador de base de datos y por el sistema desarrollado.

## ARCHIVOS Y PROCESOS.

### a) Sistema de Abastecimiento de obras.

#### - Catálogos:

Artículos.

Precios de los artículos.

Artículos de Suministro.

Paquetes de artículos.

Artículos equivalentes.

Inventario de almacenes.

Estados.

#### - Transacciones:

Necesidad de Obra.

Disponibilidad de artículos.

Ordenes de envío.

Ordenes de suministro.

Ordenes de envío de artículos.

Ordenes de suministro de artículos.

Concentrado de artículos.

Concentrado de envíos de artículos.

**- Procesos:**

Generación de órdenes de envío de material prefabricado, y mobiliario y equipo por necesidad de obras.

Actualización de disponibilidades.

**- Reportes:**

Artículos por enviar a cada obra.

Concentrado de artículos por enviar.

Existencia de disponibilidad en Jefatura de Zona.

Necesidades, existencia en el almacén, pedidos, salidas de almacén y disponibilidad.

Concentrado de Necesidades, envíos, disponibilidad en Jefaturas de Zona.

Existencias en el Almacén Central.

Necesidad, ordenado, salida física de Almacén Central.

**b) Sistema de adquisiciones.**

**- Catálogos:**

Artículos.

Proveedores.

Paquetes de artículos.

Descripción larga del artículo.

**- Transacciones:**

Licitaciones.

Licitación de artículos.

Licitación de paquetes.

Composición de paquetes.  
Proveedores por licitación.  
Ganadores.  
Pedidos.  
Pedidos asignados.  
Pedidos asignados por artículo.

**- Procesos:**

Pedidos por licitación.  
Pedidos por asignación.

**- Reportes**

Fallo final de ganadores.  
Composición de licitaciones.  
Proveedores por partida.  
Por partida con los cinco primeros lugares.  
Partidas por proveedor.  
Proveedores por partida con primeros lugares.  
Acta segunda.  
Adeudo de proveedores

En las figuras III.7. y III.8, se tiene el diagrama de flujo operativo y diagrama de flujo de datos respectivamente, de las actividades conjuntas de adquisiciones y abastecimiento.

**DIAGRAMA DE FLUJO OPERATIVO DE ADQUISICIONES**

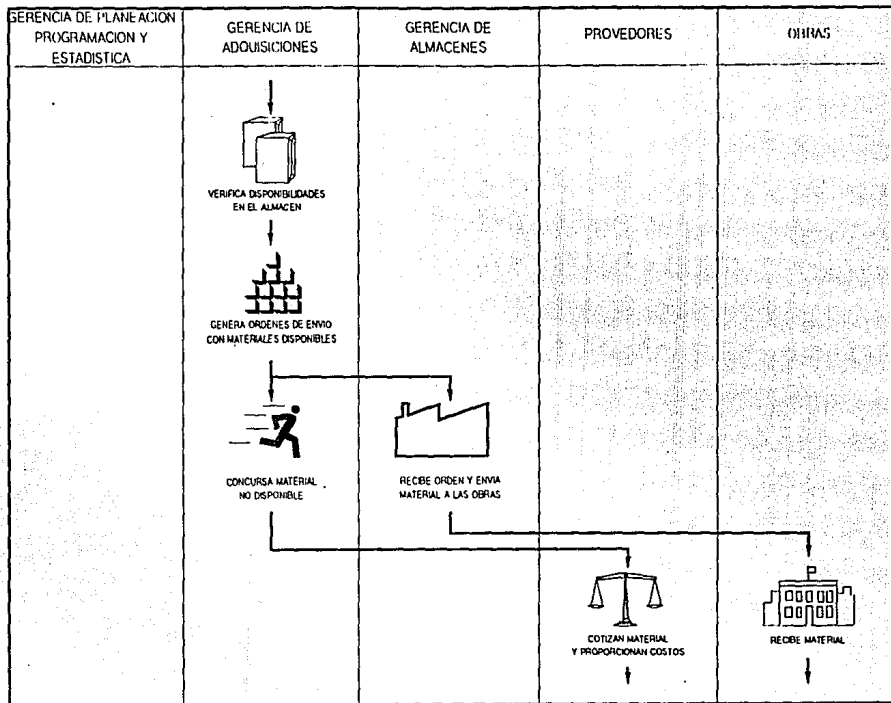


Figura III.7.a.

**DIAGRAMA DE FLUJO OPERATIVO DE ADQUISICIONES**

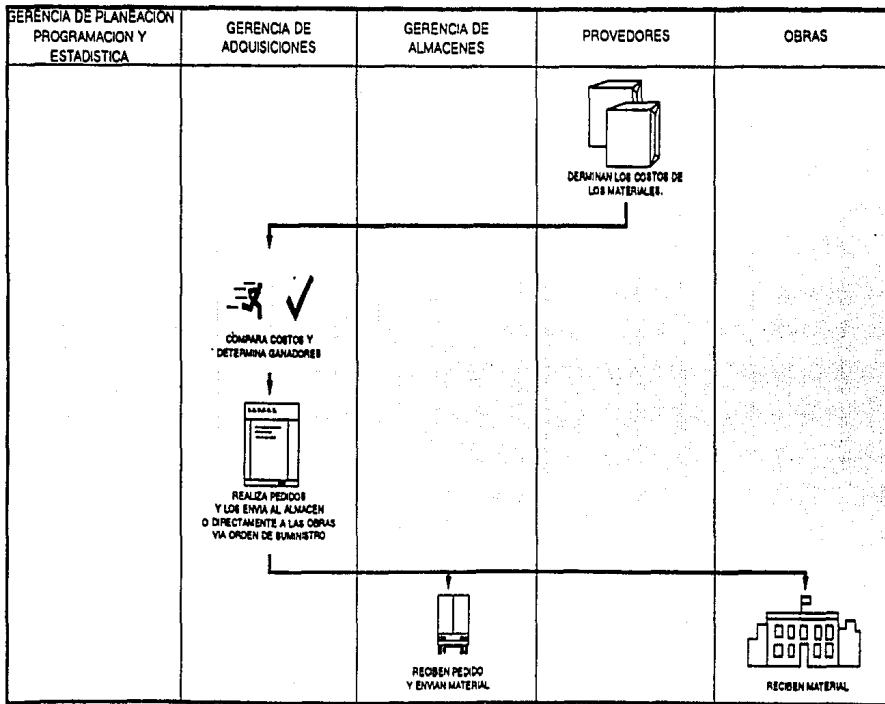


Figura III.7.b.

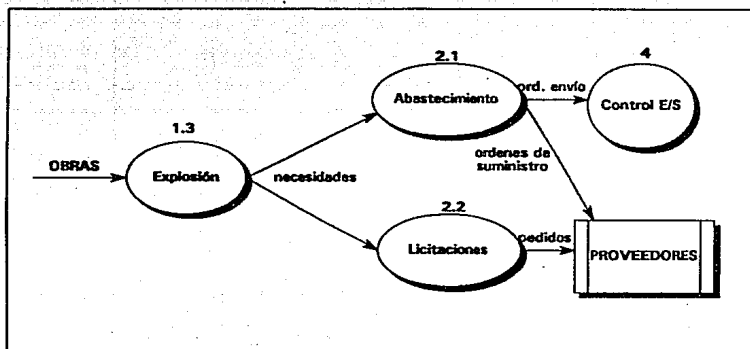


Figura III.8.a



ABASTECIMIENTO

Proceso 2.1  
Nivel 2

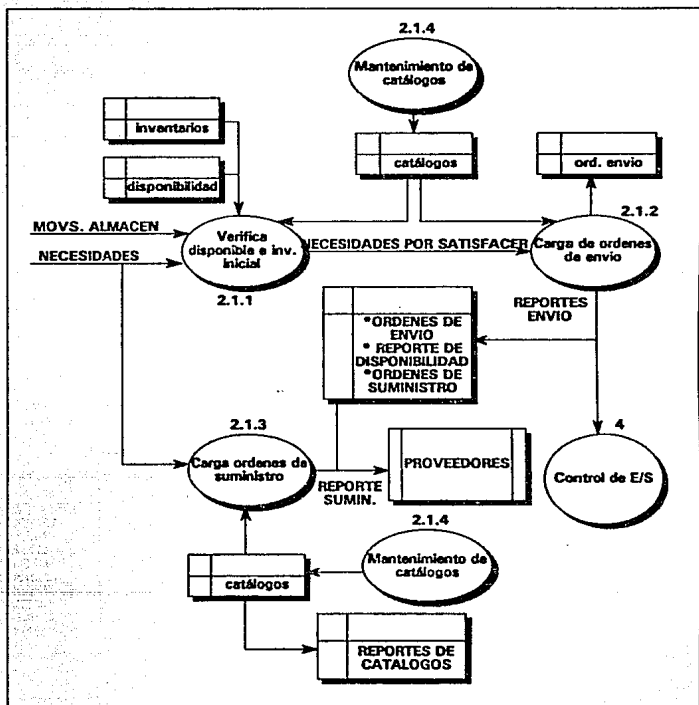


Figura III.8.b

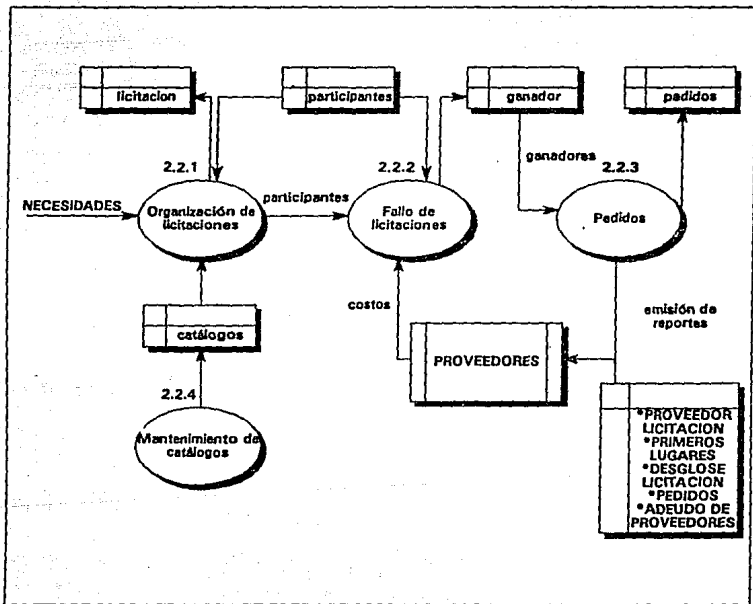


Figura III.8.c

### III.5. SISTEMA DEL ALMACEN CENTRAL.

El Almacén Central para su operación se divide en las siguientes tres áreas fundamentales:

- i).- Supervisión y control de calidad.
- ii).- Coordinación de Almacenes.
- iii).- Registro y control.

El Almacén Central tiene dos entradas de datos fundamentalmente. Por una parte está el material entregado por los proveedores que contrataron con la Gerencia de Adquisiciones, lo cual se traduce en entradas al almacén; y por otro lado están las órdenes de envío emitidas por la misma Gerencia de Adquisiciones y que contiene los destinos asignados al material recibido por el Almacén, esto se transforma en salidas del almacén.

Cuando un proveedor se presenta al Almacén a entregar material, el departamento de supervisión y control de calidad recibe y examina el material de acuerdo a las especificaciones establecidas en el pedido respectivo, si el material no cumple con dichas especificaciones se le devuelve al proveedor y se le notifica a la Gerencia de Adquisiciones para que trate el asunto directamente con el proveedor.

Una vez realizada la revisión del material y habiendo cumplido con las especificaciones requeridas, el proveedor acude a la Sección de Pedidos para entregar su remisión o factura aprobada por el departamento de Control de Calidad para que el pedido del cual se trate sea revisado y se compruebe que el proveedor entrega una cantidad menor o igual a la marcada en el pedido y que se verifique que el pedido no ha sido cancelado para el caso de que el proveedor entregue fuera de los plazos indicados en el mismo. Una vez revisados estos conceptos se le asigna un número de entrada

(entrada oficial) y el proveedor procede a entregar el material en la bodega respectiva de acuerdo al tipo de material de que se trate (Prefabricado, Maquinaria, Equipo de Laboratorio, Mobiliario; etc).

Una vez registrada se envía la entrada oficial a la Sección de Registro para que ésta proceda a actualizar las existencias. Este documento oficial se le envía también a la bodega para que registre oficialmente la entrada en su kardex interno.

Todas estas operaciones de entrada se le reportan a la Gerencia de Adquisiciones para que ésta proceda a programar el abastecimiento requerido en cada estado de la república, emita las órdenes de envío respectivas y el almacén proceda a enviar el material especificado en las mismas. Asimismo esta información sirve para elaborar el llamado adeudo de proveedores en donde se tiene el registro de entregas de cada proveedor y para el caso de que no haya entregado o cumplido con los términos y cláusulas del pedido se proceda a la cancelación y cobro de la fianza correspondiente. Si esto sucede se le notifica al almacén para que en caso de que el proveedor se presente a entregar el material no proceda a darle entrada a dicho material.

Las órdenes de envío son remitidas a la coordinación de almacenes, la cual se encarga de programar los envíos y destinos en función del transporte disponible y de la cantidad de artículos por enviar a un destino determinado. Tratando también de optimizar los recursos de transporte al asignar en un solo embarque varios destinos que se encuentren en una ruta común.

Una vez programados los destinos, éstos se envían a la bodega respectiva para que proceda al embarque de los artículos para lo cual la bodega elabora un documento oficial de salida y le asigna un número o talón de embarque al transporte de que se trate para que sea entregado en la Jefatura de Zona destino. Dichos documentos servirán para la elaboración de la póliza de seguros que ampare al material transportado. Una copia del documento de salida se envía a la Sección de Registro para

la correspondiente actualización de las existencias y otra copia se le entrega al transportista.

La información referente a las salidas realizadas se envían al Gerencia de Adquisiciones para que ésta conozca como han sido atendidos los diversos estados de la república y tome las acciones que correspondan para futuras emisiones de órdenes de envío.

#### **ARCHIVOS Y PROCESOS**

##### **- Catálogos:**

Artículos.

Proveedores.

Estados.

##### **- Transacciones:**

Recepción de pedidos por bodega.

Recepción de artículos por bodega.

Existencias por bodega.

Entradas y salidas de artículos por bodega.

##### **- Procesos:**

Control de Existencias.

Entradas y salidas del almacén.

##### **- Reportes:**

Estado de entradas al almacén.

Movimiento entre almacenes.

Movimiento de entrada al almacén.

Salidas del almacén.

Existencias por bodega.

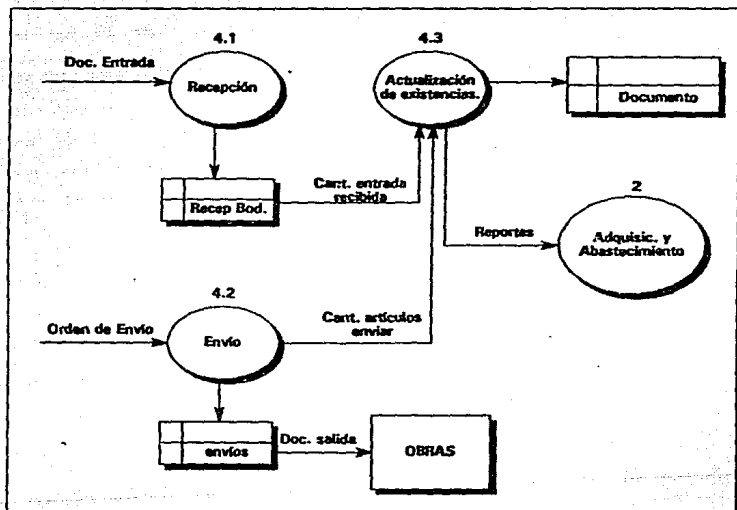


Figura III.9

la correspondiente actualización de las existencias y otra copia se le entrega al transportista.

La información referente a las salidas realizadas se envían al Gerencia de Adquisiciones para que ésta conozca como han sido atendidos los diversos estados de la república y tome las acciones que correspondan para futuras emisiones de órdenes de envío.

## ARCHIVOS Y PROCESOS

### - Catálogos:

Artículos.

Proveedores.

Estados.

### - Transacciones:

Recepción de pedidos por bodega.

Recepción de artículos por bodega.

Existencias por bodega.

Entradas y salidas de artículos por bodega.

### - Procesos:

Control de Existencias.

Entradas y salidas del almacén.

### - Reportes:

Estado de entradas al almacén.

Movimiento entre almacenes.

Movimiento de entrada al almacén.

Salidas del almacén.

Existencias por bodega.

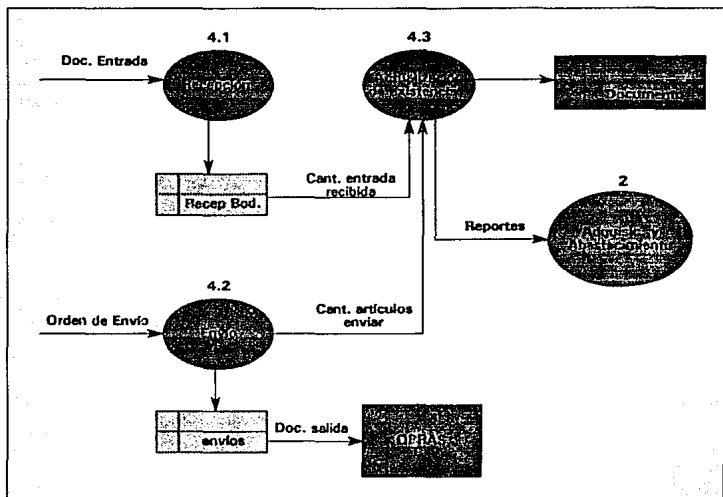


Figura III.9



### **III.6. DICCIONARIO DE DATOS**

**analista:** Es la persona encargada de proporcionar los costos de construcción de un inmueble.

**anexos:** Es el número total de construcciones anexas a una escuela que deben ser consideradas; estas pueden ser por ejemplo, direcciones, servicios sanitarios, etc.

**aulas:** Es la cantidad de aulas escolares que presenta el diseño y construcción de una escuela.

**autorización:** Es la clave de la autorización oficial entregada por S.H.C.P. para la construcción de la obra.

**cantidad del pedido:** Se tendrá que considerar la cantidad de material que se tenga que recibir de los proveedores y para llevar un control, en base a esta cantidad, para no sobrepasarse en las cantidades enviadas.

**cantidad ganadora:** Es la cantidad con la cual el C.A.P.F.C.E. declara ganador a un proveedor.

**cantidad licitada:** Es la cantidad de artículos o paquetes de artículos que se están licitando.

**cantidad pedida:** Es la cantidad de artículos o de paquetes que se están asignando a un proveedor en un pedido.

**cantidad recibida:** Es la cantidad que el almacén deberá de contabilizar en las entregas que realicen los proveedores.

**capital contable:** Es el capital con el que un concursante puede respaldar su oferta de construcción.

**capítulo:** Es la fase en la que se encuentra la construcción evolutiva de una obra.

**cantidad ofertada:** Es la cantidad que un proveedor se compromete a entregar al momento de inscribirse en una licitación.

**clase del pedido:** Es la forma de identificar a un pedido respecto a la manera en que se hará la entrega.

**clave de la licitación:** Se mantiene un control de las licitaciones mediante una clave, la cual contempla un consecutivo y un tipo de licitación.

**clave de la obra:** Es la llave para poder identificar a cada obra en específico.

**clave de las estructuras:** De acuerdo al tipo de construcción y al medio ambiente en el cual se construye un inmueble.

**clave de movimientos:** Es el tipo de movimiento del almacén, esto es, entradas, salidas, etc.

**clave del artículo:** Está constituido por los siguientes atributos: tipo, grupo, subgrupo, y un número consecutivo del artículo. Se guardará esta relación debido a que se mantiene una liga estrecha con la bodega en la cual se almacenan los diversos materiales.

**clave del concurso:** Es la clave con la cual se identifica a cada concurso de obra.

**clave del estado:** De acuerdo al orden alfabético de los Estados de la República Mexicana, se designa a cada uno de ellos un número consecutivo que forma la clave para identificarlo.

**clave del nivel educativo:** Es la clave con la cual se identifica al programa educativo de acuerdo al nivel para el cual se desarrolla la construcción de la escuela.

**clave del pedido:** Está conformada por tres principales entidades : año del pedido, tipo del pedido y un número consecutivo.

**concepto:** Es la descripción del movimiento en el almacén.

**consecutivo de proveedor:** Es la clave con la cual se identifica al proveedor al cual se haga referencia en los distintos procesos que se lleven a cabo.

**costo estimado por el C.A.P.F.C.E.:** Es el costo que el C.A.P.F.C.E. calcula por la construcción de un inmueble.

**costo por metro cuadrado:** Es el costo por metro cuadrado que el C.A.P.F.C.E. estima por cada obra.

**cubiertas:** Es el material que se utiliza en el recubrimiento del techo de una obra.

**descripción larga:** Es la descripción detallada y precisa de un artículo o de un paquete de artículos de un pedido.

**domicilio:** Es el domicilio oficial de un proveedor.

**elemento:** Es la agrupación de gufas de artículos que se deberán entregar a una construcción.

**fecha de apertura:** Es la fecha con la cual se abrirá oficialmente la licitación o concurso.

**fecha de embarque:** Se lleva un control de fechas para indicar al departamento correspondiente, si el proveedor esta cumpliendo satisfactoriamente con lo establecido en el contrato.

**fecha de entrada:** El uso de este tipo esta basado en que se debe seguir una política PEPS (Primeras Entradas, Primeras Salidas).

**fecha de fallo:** Es la fecha en la cual el C.A.P.F.C.E. dará el resultado de sus evaluaciones sobre una licitación o concurso.

**fecha de la fianza:** Es el día en el que el proveedor proporciona al C.A.P.F.C.E. la fianza.

**fecha de salida de material:** Indica cuando es que ha salido el material del almacén, y se lleva para fines de control de entrega de materiales.

**fecha del pedido:** Es la fecha con la cual se hace oficial el pedido a un proveedor.

**fianza:** Es el monto que un proveedor deja al C.A.P.F.C.E. para garantizar que cumplirá con lo contratado.

**gufas:** Es la unión de artículos, que por estar comunmente asociados a una obra, se agrupan para un mejor control.

**inventario:** Es el número o cantidad de artículos desglosados que reporta cada jefatura de zona, y que puede ser tanto material comprometido para una obra, como material disponible en sus almacenes.

**jefatura de zona:** Es la oficina estatal que representa al C.A.P.F.C.E. en una entidad federativa.

**kit:** Es un conjunto de artículos agrupados y considerados por su volumen como un solo artículo.

**laboratorios:** Es el número de laboratorios escolares que presenta la construcción de un inmueble escolar.

**localidad:** Es la comunidad en donde se está desarrollando la construcción de una escuela.

**material comprometido:** Es el material, ya sea de mobiliario o de prefabricado, que el C.A.P.F.C.E. se ha comprometido a entregar a una obra.

**material enviado:** Es el material que el C.A.P.F.C.E. se había comprometido a entregar y que ya ha enviado.

**modalidad:** Es la característica principal de la Obra

**municipio:** Es el municipio perteneciente al estado en donde actualmente se desarrolla la construcción de un inmueble escolar.

**nombre de la licitación:** Es el nombre con el cual se identifica a las licitaciones.

**nombre de la obra:** Es el nombre con el cual se identifica a una obra en especial.

**nombre de la persona que formula:** Es el nombre de quién formula un documento para control dentro del almacén.

**nombre del chofer:** Es el nombre de la persona que está a cargo de la unidad en que se transporta el material.

**número de bodega:** Para la Gerencia de Almacenes, este campo indicará en cual de las bodegas existentes está localizado cierto artículo. Se contempla para agilizar los envíos a las Jefaturas de Zona.

**número de comprador:** Es el número con el cual el C.A.P.F.C.E. identifica a la persona que a nombre suyo ha sido la encargada de comprar material a un proveedor específico.

**número de descripción:** Es el número de la descripción del artículo o del paquete del pedido.

**número de documento:** Es el número con el cual se identifica un movimiento en el almacén.

**número de estado a entregar:** Es el número de estado al cual mediante un pedido se hará la entrega de material.

**número de modelo:** Es un número asociado al nivel educativo que indica que tipo de material será utilizado.

**número de modificación:** Es el número con el cual se identifican las modificaciones que ha sufrido un pedido durante toda su vida.

**número de orden de envío:** Es un número consecutivo que indica el número total de veces que se

**ha tenido que mandar material a una obra en específico.**

**número de orden de suministro:** Es un número consecutivo que indica las veces que un proveedor ha tenido que enviar material a una obra.

**número de partida:** Se entiende por partida un aspecto de la licitación en el cual está involucrado un solo artículo. Entonces, el número de partida esta enfocado a representar a dicho artículo en la licitación de manera que se pueda identificar fácilmente su asociación.

**número de programa de financiamiento:** Es la clave con la cual se determina el tipo de programa de financiamiento (aspectos económicos).

**número de proveedor:** Este se deberá conocer debido a la importancia que guarda con respecto al pedido de artículos.

**número de remisión:** Será el número de factura que proporciona un proveedor al momento de entregar el material.

**obras inscritas:** Es el número de obras inscritas a cada concurso.

**participantes:** Es el número de participantes inscritos en un concurso.

**precio del artículo:** Es el valor unitario de cada artículo.

**precio del inventario:** Es el precio que reporta cada jefatura de zona al momento de hacer su inventario.

**precio del pedido:** Es el precio al cual se ha pedido un artículo o paquete.

**precio ganador:** Es el precio que el C.A.P.F.C.E. y el proveedor se comprometen a garantizar una vez que se ha designado al ganador.

**precio ofertado:** Es el precio que el proveedor se compromete a respetar (como máximo) en caso de que se le declare ganador.

**puntuación:** Es la calificación que otorga el C.A.P.F.C.E. a cada participante de cada concurso.

**razón social:** Es la razón social de un proveedor, es decir, el nombre con el cual se le identifica.

**registro shcp:** Es el número de registro con el cual se identifica a un proveedor o razón social, y que ha sido otorgado por la Secretaría de Hacienda y Crédito Público.

**superficie:** Es el área en metros cuadrados de cada construcción.

**talleres:** Es el número de talleres que se contemplan en la construcción de una escuela.

**tipo de impuesto:** Es el tipo de impuesto que se tiene al momento de hacer una compra, por ejemplo IVA, PECE, etc.

**tipo de modificación:** Es el campo que nos indica en palabras que tipo de modificación se ha hecho al pedido.



**tipo de partida:** Es para identificar que tipo de partida es la que se está evaluando.

**tipo de transporte:** Para fines de control en el transporte de materiales, se tendrá que manejar el tipo de vehículo de transporte.

**unidad de licitación:** Es para identificar que es lo que se está licitando, es decir, si es artículo suelto o es un paquete de artículos.

**unidad de medida:** Es la unidad con la cual se miden o contabilizan los artículos.

### **III.7. NIVELES Y CONSECUENCIAS DE LOS CAMBIOS QUE IMPLICARA EL NUEVO SISTEMA.**

#### **Nivel de información:**

- Cada Gerencia puede actualizar solo los catálogos que le correspondan.
- Cada sistema debe realizar las funciones propias, sin interferir en las funciones de las otras.
- Cada Gerencia debe estar preocupada por tener la información a tiempo.
- La Gerencia de Planeación debe tener mucho cuidado en el manejo de las obras, ya que una vez autorizadas se actualizan de forma automática a los sistemas de Adquisiciones y Concursos principalmente.
- La Gerencia de Adquisiciones cuando realiza una compra de artículos debe tener cuidado en el manejo de las autorizaciones de los pedidos, ya que afecta al Sistema de Abastecimiento.

#### **Consecuencias:**

##### **Desventajas:**

- Debido a la interconexión de los sistemas, las Gerencias se ven obligadas a tener actualizada la información con la que participan, es decir, todas las Gerencias van a depender una de otra.

**Ventajas:**

- Todas las gerencias van a obtener las mismas salidas para iguales entradas.
- La distribución de trabajo de registro de información se reduce, ya que, no va a ser necesario que cada gerencia introduzca la información que otra gerencia ya produjo previamente.
- La consistencia e integridad de la información será considerablemente alta, lo cual permitirá tomar decisiones con toda la confiabilidad y seguridad necesaria.

**IV.- DISEÑO DEL ESQUEMA DE  
SOPORTE DE LAS  
APLICACIONES  
(BACK-END).**

En esta fase de diseño, es conveniente establecer la división clásica en cuanto al diseño de una base de datos: el diseño lógico y el diseño físico.

#### **IV.1. DISEÑO LOGICO.**

El diseño lógico es el proceso de definir columnas y agruparlas en tablas siendo independiente de la estructura física de la base de datos así como de la presentación al usuario final.

Un buen diseño debe caracterizarse por:

- Consistencia en la base de datos.
- Fácil comprensión.
- Alta eficiencia.

Sin embargo ningún diseño puede satisfacer completamente los requerimientos de estos tres objetivos. Por ello debido a que lo que se persigue en la realización de este sistema es la eficiencia, el diseño lógico está fundamentado en el mecanismo de la desnormalización. Mas adelante se hará una comparación entre la normalización y desnormalización.

En esta parte del diseño lógico, se presentan los diagramas de: entidad-relación, tablas y vistas, así como cuadros de integridad referencial, y se justificará el uso de una arquitectura de bases de datos múltiples, finalizando con un diccionario de datos.

#### IV.1.1. DIAGRAMA ENTIDAD-RELACION.

En base a lo planteado en el capítulo de análisis, es posible determinar el diagrama entidad relación, el cual está formado por rectángulos, que representan a las entidades y los rombos que representan las relaciones existentes entre las entidades. Dentro de ellos se coloca el nombre de la entidad o relación. La manera en que se conectan dichas entidades y rombos es mediante las líneas, las cuales pueden terminar en flecha o no, dependiendo de la cardinalidad con que se establezca las relaciones.

En la figura IV.1 podemos observar la implementación de los diagramas de flujo de datos en una forma entidad-relación.

##### Notación.

En el diagrama de la página siguiente, la punta de flecha indica uno (1) y la cola ene (N), es decir uno a muchos. Una línea con sus dos extremos de punta de flecha, y una línea sin puntas de flecha, nos representan la cardinalidad uno a uno y muchos a muchos, respectivamente.



#### **IV.1.2. DIAGRAMA DE TABLAS Y VISTAS.**

A partir del diagrama entidad-relación se diseñó el siguiente diagrama de tablas donde se muestra el conjunto de las tablas de todos los subsistemas que participan en el S.I.I.A.O. de C.A.P.F.C.E. Cada tabla está compuesta por sus atributos respectivos indicando en la parte superior, con letras mayúsculas, el nombre de la misma.

Notación empleada.

La relación entre tablas se representa mediante una línea continua. Para aquellas tablas que tengan una tabla sobrepuesta indica que los campos son idénticos.

En el diagrama de tablas y vistas que se muestra en la figura IV.2, se puede apreciar una desnormalización. El hecho de tener campos derivados, y repetición de columnas, así como la réplica completa de una tabla tiene una justificación. Se tratará de aclarar esta situación en los siguientes puntos.



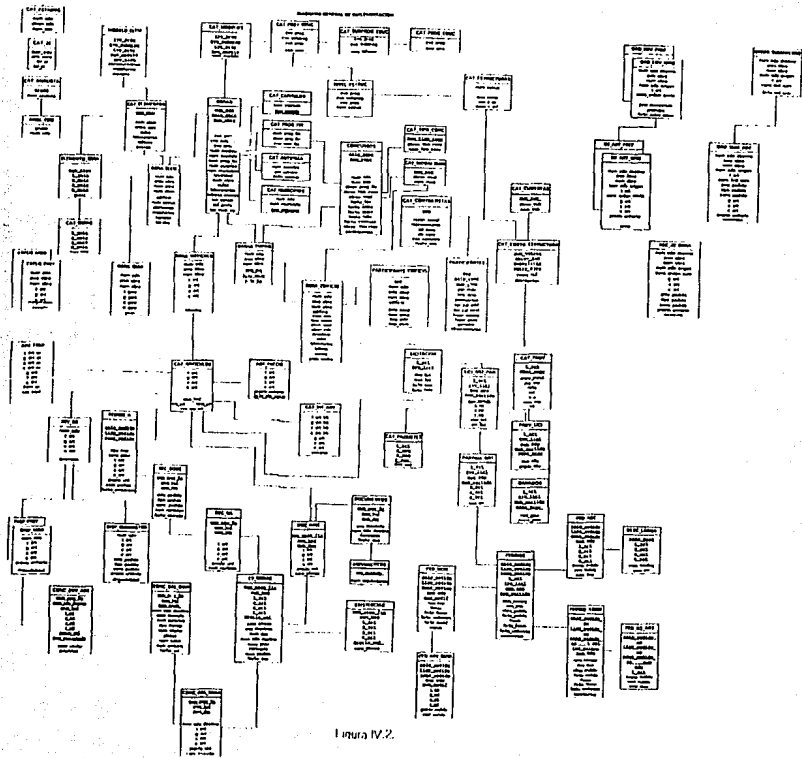


Figure IV.2

#### IV.1.2.1. NORMALIZACION.

La normalización es un método que logra que una base de datos se encuentre en una consistencia lógica con mínimo de redundancia. Posee las siguientes características:

- Incrementa el número de *joins*.
- Las actualizaciones en la información son menos complejas.
- Los renglones son más pequeños.
- Existe un gran número de tablas.

Un punto importante entre las características antes mencionadas, es el hecho de incrementar *joins* ya que esto ocasiona en la mayoría de los *DBMS* que el rendimiento se vea degradado, por ello en contraposición apareció una metodología llamada desnormalización.

#### IV.1.2.2. DESNORMALIZACION.

Es una metodología<sup>1</sup> que contrapone a la normalización para proveer eficiencia. La desnormalización puede ocurrir a nivel columna o a nivel tabla. Para desnormalizar se requiere un conocimiento previo del uso de los datos, esto significa que se pueda controlar la redundancia. Las características con las que cuenta la desnormalización se muestran a continuación:

- Minimiza la necesidad de *joins*.
- Reduce el número de llaves foráneas.
- Reduce el número de índices.
- Puede reducir el número de tablas.

---

<sup>1</sup> Relational Database Design, Paul Winsberg 1990

- Los agregados son precalculados.
- El romper alguna de las formas normales puede provocar anomalías al momento de la inserción o borrado de renglones.
- Se requiere un mayor control en los datos volátiles, es decir aquellos datos que tienen una mayor probabilidad de sufrir modificaciones.

La normalización y su contrapartida tienen sus pros y sus contras y la mayoría de los diseñadores de sistemas optan por aquellos sistemas que no presenten desnormalización, sin embargo, en la actualidad no existe un manejador que proporcione con rapidez los resultados de una consulta, debido a la gran cantidad de *joins* que el manejador debe soportar (el manejador utilizado en este proyecto sólo soporta 16 *joins*.), y a consecuencia de que se necesita eficiencia en el sistema, esto conduce a incluir columnas derivadas en las tablas, que constantemente se deben actualizar para cada una de las transacciones básicas, lo que podría provocar inconsistencia en los datos, pero, cabe aclarar que se cuenta con un manejador que provee una herramienta apropiada para estos fines: los *triggers*. De esta manera, difícilmente se puede perder la consistencia entre los datos independientemente de que sea programada en la aplicación. Además algo que es muy importante, se tiene un previo conocimiento de la manipulación de los datos de cada sistema, lo que facilita la definición y control adecuados para llevar a cabo la desnormalización.

#### IV.1.3. INTEGRIDAD DE LOS DATOS.

Como se vió el manejador posee herramientas que mantienen a los datos en términos de integridad. Ya que precisamente, una de las funciones importantes de un *DBMS* es preservar la integridad de los datos almacenados en la mayor medida posible. Es decir, estas funciones restringen los valores que pueden ser insertados en la base de datos mediante una actualización. Varios tipos diferentes de restricciones de integridad de datos suelen encontrarse en las bases de datos.

#### a) Integridad Referencial.

La integridad referencial se refiere a que una llave foránea en una base de datos relacional enlaza cada renglón de la tabla hija que contiene la llave foránea con el renglón de la tabla padre que contiene el valor de llave primaria correspondiente. Una herramienta para mantener la integridad referencial se lleva a cabo mediante *triggers*. En los *triggers* se crea toda la lógica de programación del sistema e incluso se amarra la integridad entre los diferentes subsistemas aún cuando estos se encuentren ubicados en su propia base de datos.

Los *triggers*, desde el punto de vista del manejador utilizado, son un tipo particular de procedimientos que están almacenados como un objeto en la base de datos. Existen tres tipos diferentes de *triggers*, dependiendo del tipo transacción que se realice: *insert*, *update* o *delete*, por ejemplo cada vez que se hace una inserción desde cualquier cliente el *trigger* se activa automáticamente.

En las tablas IV.1, se pueden apreciar los tipos de *triggers* que se aplican a cada una de las tablas del S.I.I.A.O. en sus transacciones básicas, también se puede apreciar la integridad referencial entre las bases de datos, apoyándose en la idea central de conseguir la integración de todos los subsistemas previo análisis en el capítulo anterior donde se identificó la afectación de información de la conexión entre ellos.

#### b) Chequeo de validez.

Algunas columnas de las tablas tienen un dominio, es decir un conjunto de valores que son legales para esa columna, por ello, para mantener la inserción o modificación del dato, fue necesario aplicar reglas proporcionadas por el manejador utilizado.

c) Integridad de identidad.

Para poder definir llaves primarias en las tablas, éstas deben contener un valor único en cada renglón, diferente de los valores de todos los renglones restantes. Para llevar a cabo este concepto se utilizaron los índices únicos (*Index unique*) que proporciona el manejador utilizado, de esta manera se evita que se duplique la llave primaria, aun cuando la inserción de los renglones se haga de manera masiva.

d) Datos requeridos.

Algunas columnas de las tablas forzosamente deben tener un valor en cada renglón, es decir, no se permiten valores nulos, por ello cuando se definen las columnas de una tabla o bien se definen los tipos, estos deben ser declarados como no nulos (*not null*). En algunos casos fué necesario hacer uso de los *defaults* los cuales son creados con valores determinados y se asocian a las columnas, de tal manera que si el usuario no proporciona el valor, el *default* será asignado.

## INTEGRIDAD REFERENCIAL - ALMACEN

TABLA	ALTA	BAJA	CAMBIO
<b>doc_arti</b>	DEBE EXISTIR EN documentos PARA DAR DE ALTA EN es_bodar CON doc_arti $\theta$ documentos $\langle$ NULO	DAR DE BAJA EN documentos CUANDO doc_arti $\theta$ documentos = NULO es_bodar	SI CAMBIA EL CAMPO cant_piezas es_bodar.cant_piezas = cant_piezas
<b>documentos</b>	DEBE EXISTIR EN movimientos Y cve_movimie $\langle$ "IM"	DAR DE BAJA EN doc_arti	SI CAMBIA EL CAMPO num_edo_destino es_bodar.num_edo_destino = num_edo_destino concepto es_bodar.concepto = concepto fecha_doc es_bodar.fecha_doc = fecha_doc
<b>es_bodar</b>	EN CASO DE cve_movimie "1":cant_piezas = cant_piezas "2":cant_piezas = -cant_piezas		SI CAMBIA EL CAMPO cant_piezas EN CASO DE cve_movimie "1":cant_piezas = cant_piezas "2":cant_piezas = -cant_piezas
<b>conc_sal_enar</b>	DEBE EXISTIR EN sal_envi conc_envar DAR ALTA EN es_bodar CON sal_envi $\theta$ conc_sal_enar	DAR DE BAJA EN sal_envi CUANDO conc_sal_enar $\theta$ sal_envi = NULO es_bodar	SI CAMBIA EL CAMPO cant_enviada es_bodar.cant_piezas = cant_enviada

Tabla IV.1.a

## INTEGRIDAD REFERENCIAL - ALMACEN

TABLA	ALTA	BAJA	CAMBIO
<b>pedido_a</b>		NO DEBE EXISTIR EN rec_bode	
<b>rec_ba</b>	DEBE EXISTIR EN rec_bode pedido_a PARA DAR DE ALTA EN es_bodar CON rec_ba $\theta$ rec_bode	DAR DE BAJA EN rec_bode CUANDO rec_ba $\theta$ rec_bode = NULO	SI CAMBIA EL CAMPO cant_piezas es_bodar.cant_piezas = rec_ba.cant_piezas
<b>rec_bode</b>	DEBE EXISTIR EN pedido_a	DAR DE BAJA EN rec_ba	SI CAMBIA EL CAMPO cant_piezas EN CASO DE cve_movimie *1':cant_piezas = cant_piezas *2':cant_piezas = -cant_piezas
<b>sal_envi</b>	DEBE EXISTIR conc_envar	DAR DE BAJA EN conc_sal_enar sal_envi PARA TODO num_bode num_doc	

Tabla IV.1.b

## INTEGRIDAD REFERENCIAL - ADQUISICIONES

TABLA	ALTA	BAJA	CAMBIO
<b>art_equl</b>	DEBE EXISTIR EN cat_articulos		
<b>art_precio</b>	DEBE EXISTIR EN cat_articulos	NO DEBE EXISTIR EN disp_suministro disp_mob CUANDO t_art="M" oe_art_mob disp_pref ce_art_pref ord_sum_art inv_ini ped_art	
<b>cat_articulos</b>		NO DEBE EXISTIR EN inv_ini art_precio lici_art_paq cat_kit_art art_equl ped_art ped_as_art EN CASO DE t_art "P":NO DEBE EXISTIR EN explo_pref "M":NO DEBE EXISTOR EN explo_mob	
<b>cat_kit_art</b>	DEBE EXISTIR EN cat_articulos		
<b>cat_paquetes</b>	NO DEBE EXISTIR EN cat_articulos	NO DEBE EXISTIR EN lici_art_paq	NO DEBE EXISTIR EN lici_art_paq

Tabla IV.1.c



## INTEGRIDAD REFERENCIAL - ADQUISICIONES

TABLA	ALTA	BAJA	CAMBIO
<b>cat_prpv</b>		NO DEBE EXISTIR EN prov_lici	
<b>desc_larga</b>	DEBE EXISTIR EN ped_art ped_as_art	NO DEBE EXISTIR ped_art ped_as_art	
<b>ganador</b>	DEBE EXISTIR EN prov_lici DAR DE ALTA ganador  CUANDO prov_lici.cant_ofer >= ganador.cant_ofer Y prov_lici.precio_ofer >= ganador.precio_ofer	NO DEBE EXISTIR EN pedidos	SI CAMBIA EL CAMPO cant_gana suma( cant_gana ) > lici_art_paq  precio_gana precio_gana > prov_lici.precio_ofer
<b>lici_art_paq</b>	DEBE EXISTIR EN licitacion cat_articulos CUANDO uni_lici = "A" cat_paquetes CUANDO uni_lici = "P"	NO DEBE EXISTIR EN prov_lici DAR DE BAJA EN partida_art	SI CAMBIA EL CAMPO cant_lici prov_lici.cant_ofer = cant_lici CUANDO cant_lici < prov_lici.cant_ofer  uni_lici, g_art, s_art, c_art NO DEBE EXISTIR EN prov_lici uni_lici EN CASO DE uni_lici *P*: DEBE EXISTIR EN cat_paquetes DAR DE BAJA EN partida_art  *A*: DEBE EXISTIR EN cat_articulos DAR DE BAJA EN partida_art

## INTEGRIDAD REFERENCIAL - ADQUISICIONES

TABLA	ALTA	BAJA	CAMBIO
<b>licitacion</b>		DAR DE BAJA EN lici_art_paq	
<b>ped_as_art</b>	DEBE EXISTIR EN pedido_asigna cat_articulos	DAR DE BAJA EN desc_larga CUANDO Y CUENTA( ped_as_art ) = 0 Y CUENTA( ped_art ) = 0	SI CAMBIA EL CAMPO cons_desc DEBE EXISTIR EN desc_larga
<b>ped_art_mod</b>	DEBE EXISTIR EN ped_mod		
<b>partida_art</b>	DEBE EXISTIR EN lici_art_paq cat_articulos	NO DEBE EXISTIR EN prov_lici	NO DEBE EXISTIR EN prov_lici
<b>pedido_asigna</b>	DEBE EXISTIR EN cat_prov	DAR DE BAJA EN ped_as_art	

Tabla IV.1.e

# INTEGRIDAD REFERENCIAL - ADQUISICIONES

TABLA	ALTA	BAJA	CAMBIO
<b>ped_art</b>	DEBE EXISTIR EN pedidos DAR DE ALTA EN ped_art CUANDO ganador.precio_gana = Y pedidos.precio_pedido ganador.cant_gana >= Y pedidos.cant_pedido ( ganador θ lici_art_paq θ O pedidos θ partida_art ganador θ lici_art_paq θ pedidos )	DAR DE BAJA EN pedidos CUANDO CUENTA( ped_art ) = 0  desc_larga CUANDO CUENTA( ped_art ) = 0	SI CAMBIA EL CAMPO cant_pedido DEBE ganador.cant_gana >= ped_art.cant_pedido  precio_uni DEBE ganador.precio_gana >= ped_art.precio_pedido
<b>pedidos</b>	DEBE EXISTIR EN ganador	DAR DE BAJA EN ped_art	
<b>ped_mod</b>	DEBE EXISTIR EN pedidos DAR DE ALTA EN ped_art_mod CON ped_art		
<b>prov_licl</b>	DEBE EXISTIR EN cat_prov lici_art_paq Y lici_art_paq.cant_licl >= prov_licl.cant_ofer	NO DEBE EXISTIR EN ganador	SI CAMBIA EL CAMPO cant_ofer DEBE lici_art_paq.cant_licl >= prov_licl.cant_ofer NO DEBE EXISTIR EN ganador precio_ofer NO DEBE EXISTIR EN ganador

Tabla IV.1f

# INTEGRIDAD REFERENCIAL - CONCURSOS

TABLA	ALTA	BAJA	CAMBIO
<b>analista_estado</b>	DEBE EXISTIR EN cat_analistas		
<b>cat_analistas</b>		DAR DE BAJA EN analista_estado	
<b>cat_contratistas</b>		DAR DE BAJA EN participantes	SI CAMBIA EL CAMPO spp participantes.spp = cat_contratistas.spp
<b>cat_costo_estructura</b>	DEBE EXISTIR EN cat_estructuras cat_cubiertas		SI CAMBIA EN CAMPO num_estruc DEBE EXISTIR EN cat_estructuras abrev_cub DEBE EXISTIR EN cat_cubiertas
<b>cat_cubiertas</b>		NO DEBE EXISTIR EN obra_edificio DAR DE BAJA EN cat_costo_estructura	SI CAMBIA EL CAMPO abrev_cub: obra_edificio.abrev_cub = cat_cubiertas.abrev_cub
<b>cat_estructuras</b>		DAR DE BAJA EN cat_costo_estructura	
<b>cat_modalidad</b>		NO DEBE EXISTIR EN concursos	SI CAMBIA EL CAMPO abrev_mod concursos.abrev_mod = cat_modalidad.abrev_mod

Tabla IV.1.g

# INTEGRIDAD REFERENCIAL - CONCURSOS

TABLA	ALTA	BAJA	CAMBIO
<b>cat_tipo_conc</b>	NO DEBE EXISTIR EN concursos		SI CAMBIA EL CAMPO abrev_tipo_conc ACTUALIZAR EN concursos CON abrev_tipo_conc = abrev_tipo_conc
<b>concursos</b>	DEBE EXISTIR EN obras_empaq pgo.cat_prog_fin cat_tipo_conc cat_modalidad CUANDO concursos.abrev_mod <> NULO  ACTUALIZAR EN concursos.obras = CUENTA(obras_empaq)	DAR DE BAJA EN participantes	SI CAMBIA EN CAMPO abrev_prog_fin DEBE EXISTIR EN pgo.cat_prog_fin  abrev_prog_fin DEBE EXISTIR EN obras_empaq  abrev_mod DEBE EXISTIR EN cat_modalidad
<b>obra_edificio</b>	DEBE EXISTIR EN obras_empaq cat_modalidad CUANDO abrev_mod <> NULO  cat_cubiertas CUANDO abrev_cub <> NULO  cat_estructuras CUANDO num_estruc <> NULO	DAR DE BAJA EN participante_edificio	SI CAMBIA EN CAMPO: abrev_cub DEBE EXISTIR EN cat_cubiertas CUANDO abrev_cub <> NULO  num_estruc DEBE EXISTIR EN cat_estructuras CUANDO num_estruc <> NULO

# INTEGRIDAD REFERENCIAL - CONCURSOS

TABLA	ALTA	BAJA	CAMBIO
<b>obras_empa</b>	<p>DEBE EXISTIR EN pgo.obras</p> <p>ACTUALIZAR EN concursos</p> <p>CON obras=obras+1</p>	<p>DAR DE BAJA EN concursos</p> <p>CUANDO CUENTA( obras_empa )=0 obra_edificio</p> <p>ACTUALIZAR EN concursos</p> <p>CON obras= obras -1</p>	<p>SI CAMBIA EN CAMPO num_paq</p> <p>ACTUALIZAR EN concursos</p> <p>obras = obras - 1 obras = obras + 1</p>
<b>participante_ edificio</b>	<p>DEBE EXISTIR EN participantes concursos 0 obras_empa 0 obra_edificio</p> <p>ACTUALIZAR EN participantes</p> <p>CON tot_cal_oext = SUMA(imp_oext) tot_cal_edif = SUMA(imp_edif)</p>	<p>ACTUALIZAR EN participantes</p> <p>CON tot_cal_oext = SUMA(imp_oext) tot_cal_edif = SUMA(imp_edif)</p>	<p>SI CAMBIA EL CAMPO imp_edif, imp_oext</p> <p>ACTUALIZAR EN participantes</p> <p>CON tot_cal_oext = SUMA(imp_oext) tot_cal_edif = SUMA(imp_edif)</p>
<b>participantes</b>	<p>DEBE EXISTIR EN cat_contratistas concursos</p> <p>CUENTA(participantes) = 1</p>	<p>BAJA EN participante_edificio</p>	<p>SI CAMBIA EL CAMPO ganador</p> <p>CUENTA(participante) &lt;&gt; 1</p>

Tabla IV.1j

# INTEGRIDAD REFERENCIAL - PROGRAMA GENERAL DE OBRA

TABLA	ALTA	BAJA	CAMBIO
<b>cat_autorisa</b>		NO DEBE EXISTIR EN obras	
<b>cat_capitulos</b>		NO DEBE EXISTIR EN obras	
<b>cat_elementos</b>		NO DEBE EXISTIR EN obra_lem DAR DE BAJA EN modelo_lem elemento_guia	
<b>cat_guias</b>		NO DEBE EXISTIR EN obra_guia elemento_guia	
<b>cat_modelos</b>	DEBE EXISTIR EN cat_proy_educ	NO DEBE EXISTIR EN obras DAR DE BAJA EN modelo_lem	
<b>cat_municipios</b>		NO DEBE EXISTIR EN obras	
<b>cat_prog_educ</b>		NO DEBE EXISTIR EN obras DAR DE BAJA EN cat_subprog_educ	

Tabla IV.1.)

# INTEGRIDAD REFERENCIAL - PROGRAMA GENERAL DE OBRA

TABLA	ALTA	BAJA	CAMBIO
<b>cat_prog_fin</b>		NO DEBE EXISTIR EN obras	
<b>cat_proy_educ</b>	DEBE EXISTIR EN cat_subprog_educ	NO DEBE EXISTIR EN obras DAR DE BAJA EN cat_modelos nivel_estruc	
<b>cat_subprog_educ</b>	DEBE EXISTIR EN cat_prog_educ	NO DEBE EXISTIR EN obras DAR DE BAJA EN cat_proy_educ	
<b>elemento_guia</b>	DEBE EXISTIR EN cat_elementos cat_guias		
<b>guia_articulo</b>	DEBE EXISTIR EN adquisic.cat_articulos cat_guias	NO DEBE EXISTIR EN obra_guia elemento_guia	
<b>modelo_elem</b>	DEBE EXISTIR EN cat_elementos cat_modelos		
<b>nivel_estruc</b>	DEBE EXISTIR EN cat_proy_educ cat_estructuras		



# INTEGRIDAD REFERENCIAL - PROGRAMA GENERAL DE OBRA

TABLA	ALTA	BAJA	CAMBIO
<b>obra_articulo</b>	<p>DEBE EXISTIR EN obras <b>adquisic.cat_articulos</b></p>		
<b>obra_elem</b>	<p>DEBE EXISTIR EN obras cat_elementos obra_elem <math>\theta</math> modelo_elem <math>\theta</math> obras cat_estructuras obra_elem <math>\theta</math> nivel_estruc <math>\theta</math> obras</p> <p>ACTUALIZAR EN obras</p> <p>CON</p> <p>aulas = SUMA (obra_elem.aulas * elementos.aulas)</p> <p>laboratorios = SUMA (obra_elem. laboratorios * elementos.laboratorios)</p> <p>talleres = SUMA (obra_elem.talleres * elementos.talleres)</p> <p>anexos = SUMA(obra_elem.anexos * elementos.anexos)</p>	<p>ACTUALIZAR EN obras</p> <p>CON</p> <p>aulas = aulas - cat-elementos * elementos.aulas</p> <p>laboratorios = laboratorios - cat_elementos.laboratorios * elementos.laboratorios)</p> <p>talleres = talleres - cat_elementos. talleres * elementos.talleres)</p> <p>anexos = anexos - cat_elementos. anexos * elementos.anexos)</p>	<p>SI CAMBIA EL CAMPO num_estruc DEBE EXISTIR EN obra_elem <math>\theta</math> modelo_ elem <math>\theta</math> obras cat_estructuras elementos</p> <p>ACTUALIZAR EN obras</p> <p>CON</p> <p>aulas = SUMA (obra_elem.aulas * elementos.aulas)</p> <p>laboratorios = SUMA (obra_elem.la boratorios * elementos.laboratorios)</p> <p>talleres = SUMA (obra_elem.talleres * elementos.talleres)</p> <p>anexos = SUMA(obra_elem.anexos * elementos.anexos)</p>
<b>obra_guia</b>	<p>DEBE EXISTIR EN cat_guias obras</p>		

# INTEGRIDAD REFERENCIAL - PROGRAMA GENERAL DE OBRA

TABLA	ALTA	BAJA	CAMBIO
<b>obras</b>	DEBE EXISTIR EN cat_modelos cat_prog_fin cat_capitulos cat_autoriza cat_municipios	NO DEBE EXISTIR EN concurso.obras_empaq adquisic.exple_mob adquisic.exple_prof  DAR DE BAJA EN obra_elem obra_articulo obra_guia	SI CAMBIA EL CAMPO cve_prog, cve_subprog, cve_proy, num_modelo DEBE EXISTIR EN cat_modelos DAR DE BAJA EN obra_elem obra_articulo obra_guia  num_prog_fin DEBE EXISTIR EN cat_modelos  num_capitulo DEBE EXISTIR EN cat_autoriza  num_municipio DEBE EXISTIR EN cat_municipios

#### IV.1.4. DISEÑO DE BASES DE DATOS MÚLTIPLES.

Varios productos *DBMS* utilizan enfoques muy diferentes para organizar las(s) base(s) de datos que manejan, y estas diferencias afectan el modo de diseñar las bases de datos y de obtener acceso a ellas.

La figura IV.3 muestra una arquitectura de bases de datos múltiples en donde cada base de datos tiene asignado un nombre único y está dedicada generalmente a una aplicación particular. Cuando se añade una nueva aplicación, lo más común es crear una nueva base de datos.

La ventaja principal de la arquitectura de bases de datos múltiples con respecto a la arquitectura de bases de datos única es que se divide a las tareas de manejo de base de datos en partes más pequeñas y más manejables. Cada persona responsable puede ahora ser administrador de su propia base de datos, con menos preocupaciones con respecto a la coordinación global. Cuando llega el momento de añadir una nueva aplicación, puede desarrollarse en su propia base de datos sin afectar a las bases de datos existentes. También, es probable que los usuarios y los programadores puedan recordar la estructura general de sus propias bases de datos.

Con una arquitectura de base de datos múltiples, para obtener acceso a una base de datos debe informarse al *DBMS* cual es la base de datos a la cual se desea entrar o consultar sus objetos.

Esto es posible hacerlo mediante una ruta definido como:

base de datos.dbo.tabla

Donde:

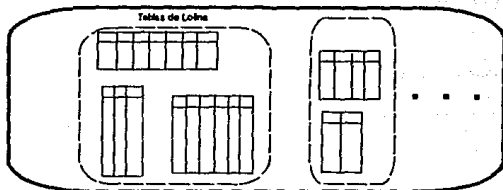
base de datos = es el nombre de la base de datos a la cual se desea acceder.

dbo (Data Base Owner) = es el dueño de la base de datos.

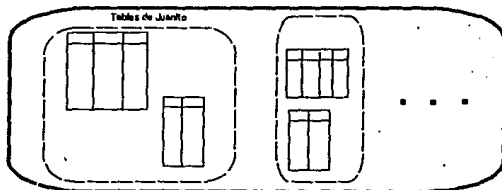
tabla= nombre de la tabla que se desea acceder.

De acuerdo a lo explicado anteriormente, existen ventajas al utilizar bases de datos múltiples, por ello, desechamos la idea de usar una sola base de datos. Además, cada gerencia cuenta con su propio equipo de cómputo, lo cual apoya en gran medida la decisión tomada.

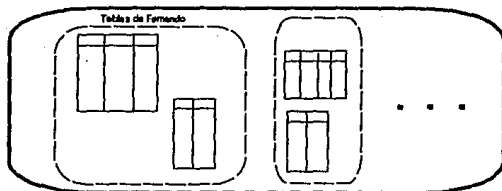
## ARQUITECTURA DE BASES DE DATOS MULTIPLES



BASE DE DATOS PGO



BASE DE DATOS CONCURSO



BASE DE DATOS ALMACEN

Figura IV.3.

La forma en que se van a crear las bases de datos es por gerencia. Cada gerencia debe tener su propia base de datos, de acuerdo a ello, se identificará a cada base de datos por el nombre del subsistema. En la parte de la Gerencia de Adquisiciones, existen dos subsistemas: El subsistema de licitaciones o adquisiciones, y el de abastecimiento, los cuales van a quedar en una sola base de datos. En la siguiente tabla se muestran las bases de datos correspondientes:

SISTEMA POR GERENCIA	NOMBRE DE LA BASE DE DATOS
Sistema del Almacén Central	almacén
Sistema de Adquisiciones y Abastecimiento	adquisic
Sistema de Concurso de Obra	concurso
Sistema del Programa General de Obra	pgo

Además de tener presentes estas cuatro bases de datos, existe una más, la base de datos modelo, la cual es proporcionada por el propio manejador y sirve para estandarizar el uso de tipos comunes a los sistemas.

En el análisis realizado en el capítulo anterior, se identificaron los archivos por cada sistema y se pudo apreciar que varios catálogos se repiten por sistema. Ahora en el diseño presentado, se considera una arquitectura de bases de datos múltiples, en donde cada catálogo será compartido entre las diferentes bases de datos. De este modo, la distribución de los catálogos se hizo como se muestra a continuación:

#### **BASE DE DATOS: ADQUISICIONES**

Artículos.

Proveedores.

#### **BASE DE DATOS: CONCURSOS**

Estructuras.

#### **BASE DE DATOS: PGO**

Obras.

Municipios.

Nivel Educativo.

Estados.

Programa de financiamiento.

Gufas

#### IV.1.5. DICCIONARIO DE DATOS.

El diccionario de datos es un almacenamiento o depósito central de información que permite identificar cuales y donde están los diferentes tipos de datos de un sistema. En el presente sistema, se construyeron dos diccionarios de datos, uno para la base de datos modelo, el cual contempla todos los tipos de datos comunes a los sistemas, a los cuales se les pueden asignar reglas o *defaults*; otro diccionario para cada una de las bases de datos mencionadas anteriormente que se basan en el diccionario de datos modelo.

#### DEFINICIONES DE LA BASE DE DATOS MODELO

El manejador de base de datos SYBASE posee una base de datos llamada modelo, la cual puede contener tipos, reglas, *defaults*, tablas, índices, llaves primarias y llaves foráneas y procedimientos almacenados, todos ellos se denominan objetos de la base de datos. Estos objetos son definidos previamente por el analista de sistemas, el cual de antemano conoce lo suficiente de todos los sistemas para que determine los datos comunes entre ellos; de esta manera cada vez que se crea una base de datos, en ésta se copian los objetos que previamente existen en la base de datos modelo.

En las tablas IV.2. se presentan el diccionario de datos modelo y en el apéndice se indica el diccionario por cada subsistema.



DEFINICION DE REGLAS.

NOMBRE	DEFINICION
anio	"69"- "99"
char1_alfa	"A-Z" y "a-z"
char1_num	"0"- "9"
char2_alfa	"AA-ZZ" y "aa-zz"
char2_num	"00"- "99"
char3_alfa	"AAA-ZZZ" y "aaa-zzz"
char3_num	"000"- "999"
char4_num	"0000"- "9999"
edificio	"A-Z", "a-z" y " "
ep_prog	"5F", "5K", "AA", "5f", "5k", "aa"
num_edo	"01"- "32"
t_art	"M", "P", "A", "m", "p", "a"

Tabla IV.2.a.

DEFAULTS.

NOMBRE	VALOR
blanco1	" "
cero	0
def_anio	(el año en curso)
no_propor	"NO PROPORCIONADO"
no_tiene	"NO TIENE"

Tabla IV.2.b.

TIPOS GENERALES DEFINIDOS POR EL USUARIO

NOMBRE	DESCRIPCIÓN	TIPO	REGLAS	DEFAULTS
anio	Año	char(2)	anio	def_anio
char1_alfa	Caracter alfabético de una posición	char(1)	char1_alfa	
char1_num	Caracter numérico de 1 posición	char(1)	char1_num	
char2_alfa	Caracter alfabético de 2 posiciones	char(2)	char2_alfa	
char2_num	Caracter numérico de 2 posiciones	char(2)	char2_num	
char3_num	Caracter numérico de 3 posiciones	char(3)	char3_num	
char4_num	Caracter numérico de 4 posiciones	char(4)	char4_num	
char5_num	Caracter numérico de 5 posiciones	char(5)	char5_num	
dinero	Dinero formateado.	money		cero
direccion	Dirección	char(40)	no_tiene	
edificio	Edificio.	char(1)	edificio	blanco1
ep_prog	Programa educativo	char(2)	ep_prog	
fecha	Fecha	datetime		
id_c_art	Consecutivo artículo	char(5)		
id_g_art	Grupo artículo	char(1)		
id_g_guia	Grupo guía	char(1)		
id_num_bod	Número de bodega	char(1)	id_numbod	
id_o_guia	Opción guía	char(3)		
id_s_art	Subgrupo artículo	char(2)		
id_s_guia	Subgrupo guía	char(1)		
id_t_art	Tipo artículo	char(1)		
id_t_guia	Tipo guía	char(1)		

Tabla IV.2.c.

TIPOS GENERALES DEFINIDOS POR EL USUARIO

NOMBRE	DESCRIPCION	TIPO	REGLAS	DEFAULTS
ientero	Entero normal	int		cero
num_edo	Número de estado	char(2)	num_edo	
real	Real	float		cero
s_entero	Entero pequeño	smallint		cero
telefono	Teléfono	char(12)		no_tiene
t_entero	Entero más pequeño	tinyint		cero
unidad_art	Unidad de artículo	char(2)		

Tabla IV.2.c.

## IV.2. DISEÑO FÍSICO.

El diseño físico se refiere a la forma en la cual los datos serán almacenados, el espacio físico requerido en disco y su administración.

### IV.2.1. DETERMINACION DE LOS INDICES.

Un aspecto importante en el diseño físico de una base de datos es pensar como los datos serán almacenados físicamente, ya que de ello depende la eficiencia del sistema.

Uno de los puntos importantes para la eficiencia de los sistemas, en cuanto a sus funciones de transacciones, es el manejo de índices. Los índices son una estructura de almacenamiento separado creada para complementar a las tablas de datos, y poseen las siguientes características:

Contienen apuntadores que permiten un acceso más rápido a los datos reales.

Sirven para forzar la unicidad de los datos.

Proveen una trayectoria más rápida para el acceso de los datos.

Evitan la búsqueda exhaustiva de datos en una tabla.

Evita un ordenamiento.

El criterio manejado para determinar la utilidad de un índice en una tabla está basado en el número de entradas y salidas, tomando en cuenta que es posible disminuir, en algún momento, el número de éstas.

Existen dos tipos de índices los *clustered* y los *no\_clustered*. Para los *clustered*, en un árbol binario, el nivel más bajo de los índices es el dato. De ésta manera los datos son físicamente

almacenados en ese orden. En el *DBMS* utilizado permite uno por tabla; mientras que para los *no\_clustered*, el orden de los índices es independiente del orden físico de los datos. Se permiten más de uno en el *DBMS* utilizado.

De lo anterior, se deduce que para cada una de las tablas, es necesario determinar la existencia o el tipo de índice que se puede utilizar. Para determinarlo se tienen que considerar las siguientes características :

- La cantidad estimada de renglones.
- La frecuencia de transacciones que se realicen.
- La monotonía o aleatoriedad de las transacciones sobre la llave primaria.
- La estructura de las consultas más solicitadas.

Considerando lo anterior, se identificaron aquellos catálogos que tienen el menor número de renglones, los cuales no necesitarán índices. De la misma forma, aquellas tablas que presentan mayor frecuencia de transacciones se determinó el uso de índices *no\_clustered*. El resto de las tablas poseen índices *clustered* para sus llaves primarias.

#### IV.2.2. TAMAÑO DE LAS BASES DE DATOS.

Una vez realizado el diseño lógico de las bases de datos, es necesario conocer el espacio en disco que cada una de ellas va a ocupar.

Para determinar el tamaño de una base de datos, es necesario obtener los tamaños de cada tabla; para ello es importante definir el *fillfactor* por página, el cual es expresado de un valor de 0% hasta un 100%, además de saber cuáles de los campos que conforman a la tabla son variables, y

conocer el número aproximado de renglones de esa tabla.

El factor de llenado es un parámetro importante para mejorar la eficiencia en los sistemas, permite decidir que tan densamente se desea que las páginas sean llenadas (tanto índices como datos). Si se usa un factor de llenado pequeño los datos e índices ocuparán mayor número de páginas. Esto se puede observar en los resultados obtenidos en la tabla IV.4.

El cálculo se realizó en base a tres factores de llenado, uno del 0%, otro del 50% y un tercero de 70%.

Se presenta a continuación las fórmulas empleadas para calcular el espacio en disco de una tabla.

tamaño de la página = 2048 - 32 bytes.

overhead/renglón = 2 bytes .

(mayor si existen campos variables)

renglones por página = tamaño página / (tamaño renglón + overhead)

página = renglones / (renglones por página \* factor de llenado)

Ejemplo:

Se tiene una tabla que contiene 1,000,000 de renglones y 100 bytes por renglón, y considerando un factor de llenado del 75 % se obtiene que:

renglones por página =  $(2048 - 32) / (100 + 2) = 19$  reng/pag

páginas =  $1,000,000 / (19 * 0.75) = 70,175$  páginas

Para determinar el número de *bytes* por renglón, es necesario conocer la cantidad de *bytes* que utiliza cada tipo de dato. De acuerdo a ello en la siguiente tabla se indica el número de *bytes* que utiliza los tipos de datos empleados en la creación de las tablas:

TIPO DE DATO	BYES		
char	1-256	varchar	1-256
smallint	2		
tinyint	1		
int	2		
float	4		
text	2 Mb		
bit	1 bit		

# TAMAÑOS DE LAS BASES DE DATOS

## ALMACEN

TABLAS	CAMPOS VARIABLES	NUMERO DE REGISTROS	FACTOR DE LLENADO (Kb)		
			0 %	50 %	70 %
doc_arti		12,000	372	754	536
documentos	concepto	4,000	228	462	328
es_bodar	concepto	20,000	1,706	3,436	2,446
existencias		3,000	70	138	100
movimientos		10	16	16	16
pedido_a		3,400	142	288	204
rec_ba		13,600	394	798	568
rec_bode	num_remision	3,400	132	268	190
con_sal_enar		28,000	1,804	3,634	2,586
sal_envi		4,800	478	966	686
TAMAÑO			5,342	10,760	7,660



# TAMAÑOS DE LAS BASES DE DATOS

## ADQUISICIONES

TABLAS	CAMPOS VARIABLES	NUMERO DE REGISTROS	FACTOR DE LLENADO (Kb)		
			0 %	50 %	70 %
art_equi		50	16	16	16
art_precio		27,000	728	1,470	1,046
cat_articulos	nom_art	9,000	974	1,956	1,394
cat_kit_art		1,400	30	60	42
cat_paquetes	desc_paq	50	16	16	16
cat_prov	razon_social	4,000	748	1,502	1,072
	calle				
	col				
	tel				
conc_env_art		60,000	2,228	4,484	3,192
desc_larga		5,000	210	422	302
disp_mob		2,000	50	100	72
disp_pref		2,000	50	100	72
disp_suministro		500	16	32	24
explo_mob		85,000	1,790	3,600	2,562
explo_pref		85,000	1,790	3,600	2,562
ganador		100	16	28	20
inv_ini		38,000	876	1,762	1,254
lici_art_paq		2,500	74	126	90

# TAMAÑOS DE LAS BASES DE DATOS

## ADQUISICIONES

TABLAS	CAMPOS VARIABLES	NUMERO DE REGISTROS	FACTOR DE LLENADO (Kb)		
			0 %	50 %	70 %
licitaciones	nom_licí	50	16	16	16
oe_art_mob		55,600	2,178	4,390	3,122
oe_art_pref		102,100	3,998	8,058	5,732
ord_env_mob		11,000	352	712	508
ord_env_pref		11,300	362	732	522
ord_sum_art		16,000	678	1,364	972
orden_suministro		2,300	62	124	90
partida_art		5,000	116	232	164
ped_art		5,000	170	346	244
ped_as_art		250	16	18	16
ped_as_mod		4,000	128	258	184
ped_mod		800	52	102	72
pedido_asigna		50	16	16	16
pedidos		1,000	58	114	82
prov_licí		1,000	28	54	40
rec_jz_obra		16,000	9,708	19,592	13,922
sal_conc_enar		60,000	2,046	4,122	2,934
TAMAÑO			29,596	59,524	42,372

Tabla IV.3.c

# TAMAÑOS DE LAS BASES DE DATOS

## CONCURSOS

TABLAS	CAMPOS VARIABLES	NUMERO DE REGISTROS	FACTOR DE LLENADO (Kb)		
			0 %	50 %	70 %
anal_edo		32	16	16	16
cat_analista	nom_analista	4	16	16	16
cat_contratistas	razon_social representante tel_cont dir_cont	5,000	868	1,746	1,244
cat_costo_estructura	descripcion	300	24	48	34
cat_cubiertas	nom_cub	4	16	16	16
cat_jz	jefe_zona dir_jz tel_jz	32	16	16	16
cat_modalidad	nom_mod	10	16	16	16
cat_tipo_conc	nom_tipo_conc	2	16	16	16
concursos		2,500	200	402	288
obra_edificio		6,500	330	668	474
obras_empaq		4,000	148	298	212
participante_edificio		20,000	902	1,812	1,292
participantes	observaciones	5,000	650	1,308	932
TAMAÑO			3,218	6,378	4,572

Tabla IV.3.d

**TAMAÑOS DE LAS BASES DE DATOS**  
**PROGRAMA GENERAL DE OBRA**

TABLAS	CAMPOS VARIABLES	NUMERO DE REGISTROS	FACTOR DE LLENADO (Kb)		
			0 %	50 %	70 %
cat_autoriza	cve_autoriza	25	16	16	16
cat_capitulos	nom_capitulo	5	16	16	16
cat_elementos	nom_elem	5,000	250	500	356
cat_estados	nom_edo	32	16	16	16
cat_estructuras	nom_estruc	20	16	16	16
cat_guias	nom_guia	540	52	104	74
cat_modelos	nom_modelo	60	16	16	16
cat_municipios	nom_municipio	5,000	230	462	328
cat_prog_educ	nom_prog	3	16	16	16
cat_prog_fin	nom_prog_fin	11	16	16	16
cat_proy_educ	nom_proy	26	16	16	16
cat_subprog_educ	nom_subprog	11	16	16	16
elemento_guia		7,000	140	282	200
modelo_elem		5,000	80	160	114
nivel_estruc		1,000	16	22	16
obra_articulo		10,000	268	546	388
obra_elem		50,000	1,102	2,212	1,578
obra_guia		5,000	120	242	172
obras	nom_obra localidad	10,000	768	1,548	1,104
<b>TAMAÑO</b>			<b>3,170</b>	<b>6,222</b>	<b>4,474</b>

Tabla IV.3.e

### **IV.2.3. ARQUITECTURAS.**

Una arquitectura desde el punto de vista de base de datos, se refiere a la ubicación, conexión y relación de éstas, de las aplicaciones y sus servidores. En los siguientes puntos se describen los diferentes tipos de arquitecturas.

#### **IV.2.3.1. CENTRALIZADA.**

En esta arquitectura el DBMS y los datos físicos residen en una computadora central, junto con el programa de aplicación que acepta entradas desde la terminal del usuario y muestra los datos en la pantalla.

Tanto el procesamiento de la aplicación como el procesamiento de la base de datos se producen en la computadora central, y debido a que el sistema es compartido por varios usuarios, cada usuario experimenta una degradación del rendimiento cuando el sistema tiene una carga fuerte. Esta arquitectura se muestra en la figura IV.4.

## ARQUITECTURA CENTRALIZADA

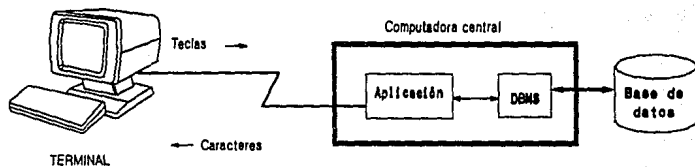


Figura IV.4

#### IV.2.3.2. CLIENTE-SERVIDOR.

La arquitectura cliente-servidor presenta varias características innovadoras en la industria del *software* entre las cuales se pueden mencionar:

**Servidor Programable:** Esta característica permite el manejo de datos y transacciones en forma independiente a las aplicaciones. Es decir, basta con desarrollar una sola vez la programación organizada de reglas y transacciones en el servidor para no tener que repetir éstas en cada aplicación. Esto proporciona un mejor control de los datos, aplicaciones y herramientas, además de simplificar el mantenimiento y aumentar el rendimiento del sistema. Todo esto se basa en que SQL es un vehículo natural para implementar aplicaciones utilizando una arquitectura cliente/servidor distribuida. En este papel, SQL sirve como enlace entre los *front-end* optimizados para interacción con el usuario y los *back-end* especializados para la manipulación de los datos, permitiendo que cada sistema rinda lo mejor posible, sobre todo con el concepto de *stored procedures*. La figura IV.5. muestra la emergente arquitectura cliente/servidor para manejo de bases de datos. En esta arquitectura, las computadoras personales están combinadas en una red de área local junto con un servidor de base de datos compartidas. Las funciones del *DBMS* están divididas en dos partes. Los *front-ends* de base de datos, tales como herramientas de consulta interactiva (ISQL), escritores de informe (Report Writer) y programas de aplicación (APT Edit, APTSQL), que se ejecutan en la computadora personal. La máquina de soporte (*back-end*) de la base de datos que almacena y manipula los datos se ejecuta en el servidor. SQL se ha convertido en el lenguaje de base de datos estándar para comunicaciones entre las herramientas *front-end* y la máquina de soporte en esta arquitectura.

# ARQUITECTURA CLIENTE-SERVIDOR

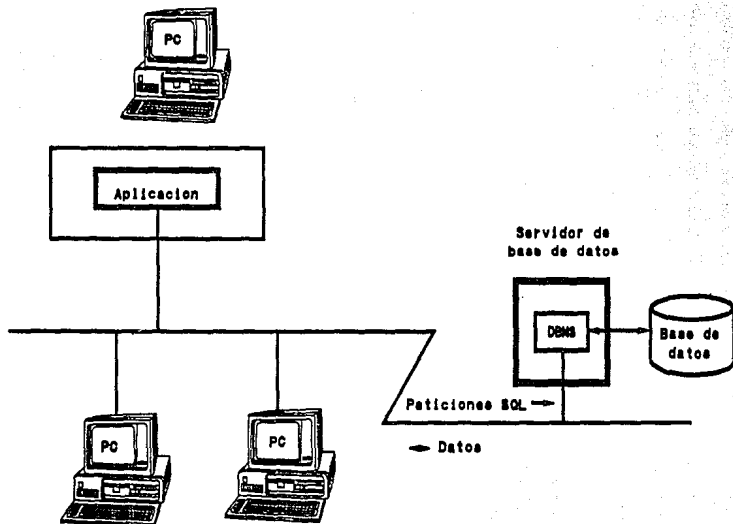


Figura IV.5.



En esta arquitectura cliente-servidor, la consulta viaja a través de la red hasta el servidor de la base de datos como una petición SQL. La máquina de base de datos en el servidor procesa la petición y explora la base de datos, que también reside en el servidor. Cuando calcula el resultado, la máquina de base de datos envía de vuelta a través de la red una única respuesta a la petición inicial, y la aplicación *front-end* la muestra en la pantalla de la PC.

La arquitectura cliente-servidor reduce el tráfico de red y divide la carga de trabajo de base de datos. Las funciones en proceso de datos, tales como entrada/salida de archivos, y el procesamiento de consultas se concentran en el servidor de la base de datos, lo que es importante el lenguaje de SQL proporciona una interfaz bien definida entre los sistemas *front-end* y de soporte, comunicando las peticiones de acceso a la base de datos de una manera eficiente.

**Procesamiento Distribuido de Alto Rendimiento:** Esto se logra al distribuir las diferentes funciones de una aplicación en tareas más pequeñas realizadas por los componentes del manejador; por ejemplo, el uso de procedimientos previamente compilados reduce el tiempo de procesamiento de una consulta y libera recursos rápidamente.

#### IV.2.3.3. MULTISERVIDOR.

##### PROCEDIMIENTOS ALMACENADOS Y EL RENDIMIENTO EN DBMS

Sybase es un manejador de base de datos de alto rendimiento para aplicaciones *OLTP* (*on line transactions processing*). Los procedimientos almacenados son una parte crítica de este reclamo de rendimiento. Cada sentencia que un programa envía a SQL server recorre los cinco pasos (análisis, validación, optimización, compilación y finalmente ejecución) en tiempo de proceso. Primero las transacciones típicas de un programa *OLTP* utilizan las sentencias *INSERT*, *DELETE*, *UPDATE* y

*SELECT* que acceden a un único renglón de datos basándose en la llave primaria del renglón. Es relativamente fácil optimizar este tipo de sentencia SQL. Por lo tanto, el recargo de la optimización y la compilación en tiempo de ejecución es relativamente bajo para los programas *OLTP* sencillos de alto uso que necesitan el rendimiento más elevado.

En contraparte, los procedimientos almacenados proporcionan un modo de analizar, optimizar y compilar una secuencia entera de sentencias SQL. Para obtener el mejor rendimiento posible de un programa, el programador toma las sentencias SQL a ejecutar, define un procedimiento que las contenga y almacena el procedimiento compilado en la base de datos. En tiempo de ejecución, el programa simplemente solicita al SQL server que ejecute el procedimiento almacenado. Por tanto, los procedimientos almacenados proporcionan al SQL Server la flexibilidad de una interfaz SQL dinámico y los beneficios de rendimiento de las sentencias precompiladas encontradas en SQL estático. Esto además permite la existencia de *Remote Procedure Calls* el cual es el medio por el cual se pueden comunicar varios servidores, dando paso así a los esquemas de datos distribuidos y a los sistemas abiertos.

Cliente Abierto e Interfaces del Servidor: Esto permite que herramientas de programación, aplicaciones y fuentes de datos que no son propias del manejador puedan ser integrados a éste de manera transparente sin que afecten el rendimiento de las aplicaciones nativas del manejador. La figura IV.6. esquematiza la arquitectura anterior.

# ARQUITECTURA MULTISERVIDOR

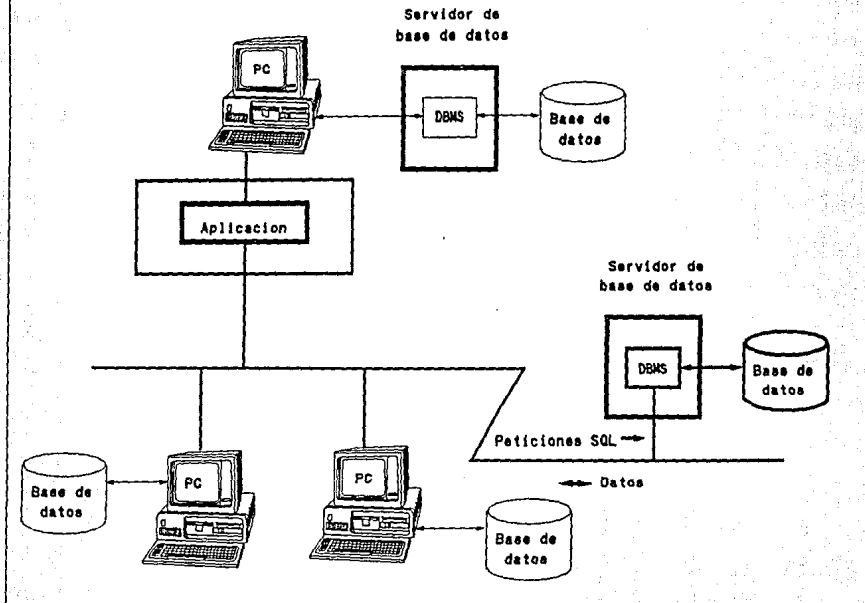


Figura IV.6.

Una arquitectura o sistema abierto facilita la portabilidad y manejo de diferentes componentes tanto de *software* como de *hardware* dentro de un mismo ambiente de trabajo.

La independencia que pueda presentar el manejador de la base de datos con respecto al *hardware* y al *software* permite que el manejador sirva como medio de enlace entre aplicaciones residentes en diferentes tipos de máquinas de manera que puedan operar conjuntamente.

#### IV.2.4. SEGURIDAD DEL SISTEMA.

El servidor-SQL provee procedimientos del sistema para el manejo de la cuenta de los usuarios. El administrador del sistema o dueño de la base de datos puede crear *logins* en el servidor-SQL y agregar usuarios a una base de datos específica. Además los procedimientos son usados para cambiar la identificación del usuario y sus permisos.

La implementación de un esquema y el reforzamiento de las restricciones de seguridad son responsabilidad del *software* del DBMS. El lenguaje SQL define un panorama general para la seguridad de la base de datos, y las sentencias SQL se utilizan para especificar restricciones de seguridad. El esquema de seguridad SQL se basa en tres conceptos principales:

- Los usuarios, los cuales son los actores de las bases de datos. Cada vez que el DBMS recupera, inserta, suprime o actualiza datos, lo hace a cuenta de algún usuario. El DBMS, permitirá o restringirá la acción dependiendo de que usuario esté efectuando la petición.

- Los objetos de la base de datos son los elementos a los cuales se puede aplicar la protección de seguridad SQL. La seguridad se aplica generalmente a tablas y vistas pero otros objetos tales como procedimientos almacenados, *triggers*, reportes pueden también ser protegidos. La mayoría de los

usuarios tendrán permiso para utilizar ciertos objetos de la base de datos, pero tendrán prohibido el uso de otros.

- Los privilegios son los permisos que un usuario tiene permitido efectuar para un determinado objeto de la base de datos. Un usuario puede tener permiso para insertar o consultar un renglón determinado, por ejemplo, pero puede carecer de permiso para borrar o actualizar el renglón. Un usuario diferente puede tener un conjunto diferente de privilegios.

El *DBMS* utilizado provee una jerarquía para establecer las restricciones de acceso a los objetos de las bases de datos. Existe como cuenta privilegiada (para fines de administración del propio manejador) al *System Administrator* ("sa") desde el cual se crean los permisos de entrada a las bases de datos mediante un *login* determinado. En un segundo nivel se encuentra el dueño de la base de datos, el cual decide que usuarios va a permitir accederla, así como crear los grupos para fines de administración de usuarios. En tercer nivel existen los usuarios o dueños de los objetos. En cuarto nivel están los usuarios invitados (*guest*) o públicos (*public*) los cuales forman parte de otras bases de datos.

En la figura IV.7. se muestra la jerarquía de privilegios de acceso.

La distribución de *logins*, grupos y usuarios está determinado por el conjunto de bases de datos y el número de usuarios que accesan cada una de ellas.

## SEGURIDAD DEL SISTEMA

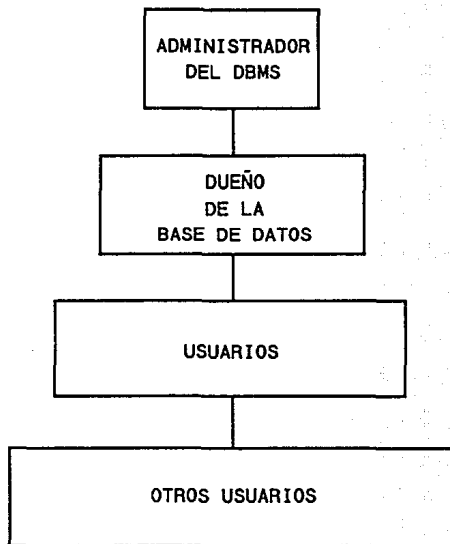


Figura IV.7.



**V. DISEÑO DE LAS APLICACIONES DE  
INTERFAZ CON EL USUARIO  
(FRONT-END).**



El diseño de las aplicaciones de interfaz con el usuario, se refiere, al diseño de las formas de captura y el diseño de los reportes.

#### V.1. DISEÑO DE FORMAS DE CAPTURA.

En una forma existen básicamente dos componentes principales que deben ser diseñados:

##### ARBOL DE MENUS

Para el árbol de menús se estableció una forma estándar, es decir, que generalmente el orden de las opciones estuviera formado de la siguiente manera:

- a) Catálogos.
- b) Proceso(s) principal(es) de primer nivel del sistema.
- c) Consultas (sumarizadas).
- d) Reportes.

a) En la parte de catálogos, se proporciona todo el mantenimiento a los catálogos propios de cada sistema.

b) En esta parte existen las opciones que generan los procesos más significativos del sistema. Estos pueden estar basados en una forma de mantenimiento que a su vez 'dispare' procesos de segundo nivel.

c) En la sección de consultas sumarizadas, están aquellas consultas que el usuario generalmente realiza, o que son más importantes para él, inclusive pueden ser utilizadas a nivel gerencial.

d) Los reportes son las salidas a las cuales tienen que llegar los procesos importantes de los sistemas, así como también permiten llevar un control más específico sobre el mantenimiento de los catálogos.

Algunos árboles de menús no se acoplan a la estandarización antes mencionada, debido a la condiciones propias del sistema. En la figura V.1 se muestran los árboles de menús que dan origen a las pantallas en los diferentes sistemas.

# ARBOL DE MENUS

## ADQUISICIONES

LICITACIONES

ABASTECIMIENTO

Figura V.1a

# ARBOL DE MENUS

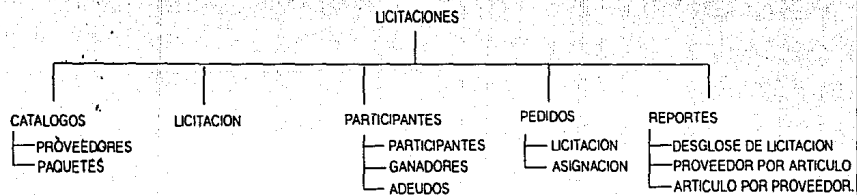


Figura V.1.b

# ARBOL DE MENUS

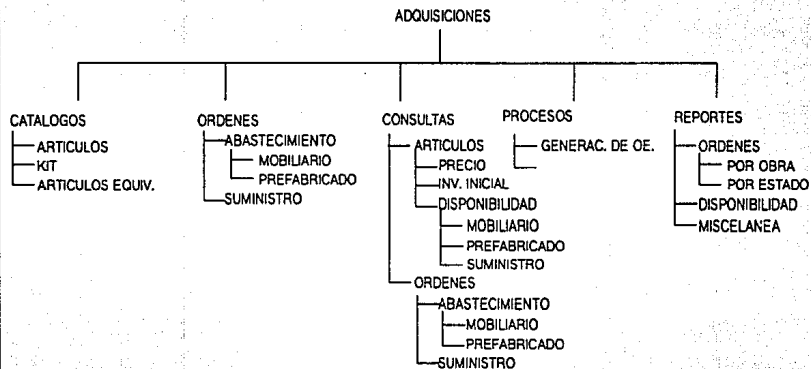
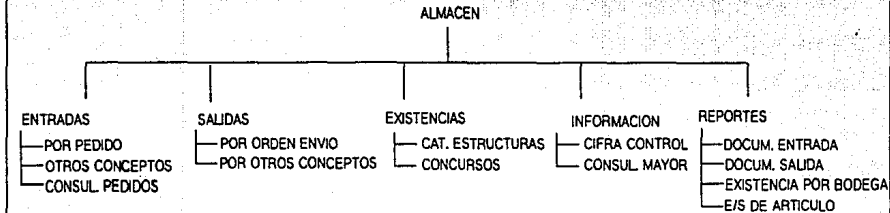


Figura V.1.c.

# ARBOL DE MENUS



# ARBOL DE MENUS

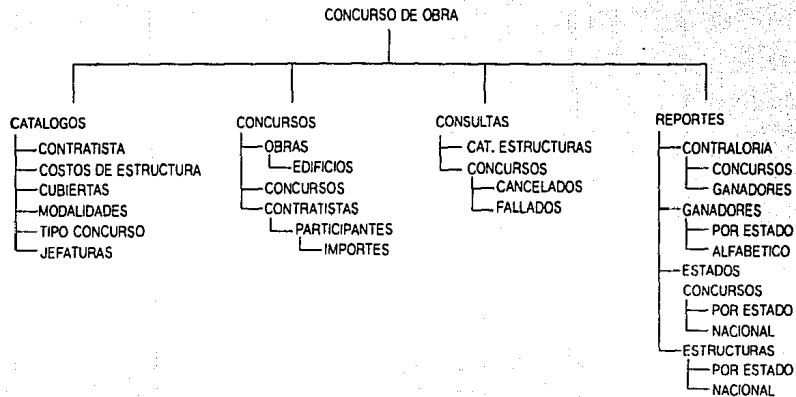


Figura V.1.e.

# ARBOL DE MENUS

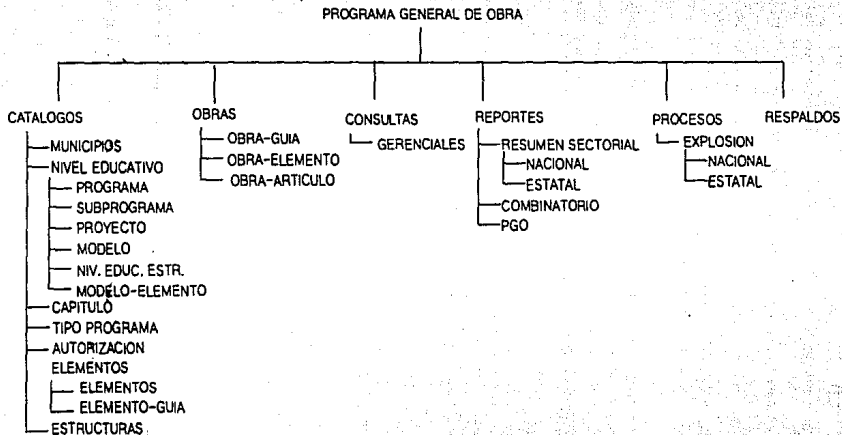


Figura V.1.f



## DISEÑO DE PANTALLAS

El objetivo de realizar rápidamente sistemas, nos lleva a establecer pantallas estándares, que tienen ligados procedimientos; esto se refiere a que la pantalla al ser llamada, ya contiene procedimientos de limpieza de pantallas, distinción de la fecha, y del usuario, e inclusive impresión de la misma pantalla. Esta pantalla tiene como nombre 'formabas'.

La forma base esta compuesta por un encabezado, opciones y procesos ligados que se utilizan cada vez que se crea una nueva pantalla de captura. Al momento de llamar a la forma base es necesario salvarla con otro nombre apegado a la aplicación. De este modo se crea una réplica, en la cual ya se pueden depositar los campos de la tabla que se desee realizar altas, bajas o cambios.

La forma base cuenta con los siguientes elementos:

A) TITULO

B) OPCIONES

C) ICONOS

D) AREA DE TRABAJO

1.- Encabezado.

2.- Campos.

E) PROCESOS LIGADOS A LA FORMA BASE

Existen cuatro tipos de procedimientos involucrados en una forma de trabajo:

a) Procedimientos ligados a la forma

La forma para su presentación al usuario necesita tener procedimientos que se activen antes de entrar a la forma, los cuales tienen como función limpiar los campos, obtener la fecha actual y el

nombre del usuario.

b) Procedimientos ligados al menú

Estos procedimientos se activan cuando alguna de las opciones del menú es escogida por el usuario. La mayoría de las formas de los sistemas presentan las siguientes opciones dentro de su área de menú:

1.-Altas 2.-Bajas 3.-Cambios 4.-Borra ...

las cuales como su nombre lo indican llaman a un procedimiento que al ejecutarse realizan la transacción correspondiente. En el caso de los tres puntos, es un indicador de continuación del menú, es decir, que al escoger esta opción, se abre una pantalla donde se presentan más opciones. La opción borra, llama a un procedimiento que se encarga de borrar la información contenida en los campos de captura de la forma.

También en la sección de menús de la forma existe un conjunto de íconos, los cuales se localizan en la parte derecha de la sección de menús. Se muestran a continuación:

^            X  
<   +   >   \*  
v            ?

Para las cuatro puntas de flecha, se tienen procedimientos ya proporcionados por la misma forma que sirven para hacer *scroll* de la pantalla. El símbolo de + es una opción que combinándose con cualquiera de las flechas, nos permite llegar a los extremos. La X sirve para salir de la forma.

El \* al ser escogido proporciona una ventana de utilerfas necesarias para el mantenimiento de la forma, tales como imprimirla, cambiarla de color, accionarla para que trabaje con *mouse*, etc. El signo de interrogación ? presenta texto que ayuda al manejo de la misma, en caso de que se requiera.

c) Procedimientos ligados a los campos

Existen dos tipos de procedimientos principalmente, uno que se activa antes de capturar la información del campo (preproceso) y otro que se activa inmediatamente después de salir de la captura del campo (postproceso).

En la mayoría de las formas existe un campo que como postproceso controla las opciones del menú automáticamente, es decir, al realizar la búsqueda en las tablas por medio del índice, se active la opción de alta si no encuentra el renglón, o bien se activa baja y cambio, si el renglón ya está contenido en la tabla.

Además de este procedimiento, existen otros procedimientos que sirven para convertir los textos a mayúsculas.

Para el encabezado de la pantalla, existen procedimientos que traen el nombre del usuario y la fecha del sistema.

d) Procedimientos ligados a las teclas.

Es posible definir funciones específicas en las teclas. La tecla F2 se tomó como tecla estándar para realizar consultas en base a los datos que el usuario proporcionará en la forma de captura. Estos datos pasan como parámetros definidos por la propia forma, y la forma a ser llamada también contiene estos mismos parámetros, de tal suerte, que de manera de lista aparezcan los datos en la siguiente forma. La forma base se muestra en la figura V.2.

TITULO					
1.-Altas	2.-Bajas	3.-Cambios	4.-Borra	...	
C.A.P.F.C.E.   Usuario:			Fecha:		

Forma base. (FORMABAS)

Figura V.2.

## V.2. ALGORITMOS DE PROCEDIMIENTOS GENERALES.

A continuación se presenta los algoritmos de procedimientos generales como son: altas, bajas, cambios, consultas y búsquedas, que son útiles para los sistemas involucrados en el SIAO. La finalidad de hacer estos procedimientos generales es para proporcionar una serie de ventajas como son:

- Evitar la repetición de código.
- Independizar los programas de alguna modificación de la bases de datos.
- Agilizar la realización de los sistemas futuros.

### Algoritmo del Procedimiento de Alta General.

- 1.- Obtener el nombre de la tabla que está accedando la forma de captura.
- 2.- Inicializar la variable que contendrá la instrucción de inserción con el nombre de la tabla obtenido anteriormente.
- 3.- Realizar una consulta que obtenga los nombres de todos los campos de la tabla proporcionada.
- 4.- Obtener el valor de cada uno de los campos que se teclaron en la forma de captura y concatenarlos en la variable de caracteres que contiene la instrucción de inserción en el orden en que fueron definidos.
- 5.- Una vez que la variable que contiene la instrucción de inserción ha quedado completa, se somete a ejecución.

6.- En caso de no existir error en la ejecución se manda un mensaje que indique el éxito de la transacción, de lo contrario se manda un mensaje de error.

#### Algoritmo del Procedimiento de Baja General.

- 1.- Obtener el nombre de la tabla que está accedando la forma de captura.
- 2.- Inicializar la variable que contendrá la instrucción de borrado con el nombre de la tabla obtenido anteriormente.
- 3.- Realizar una consulta que obtenga los nombres de todos los campos que compongan la llave primaria de la tabla proporcionada.
- 4.- Obtener el valor de cada uno de los campos, que forman la llave primaria, que se teclearon en la forma de captura y concatenarlos en la variable de caracteres que contiene la instrucción de borrado.
- 5.- Una vez que la variable que contiene la instrucción de inserción ha quedado completa, se somete a ejecución.
- 6.- En caso de que no haya error en la ejecución se manda un mensaje que indique el éxito de la transacción, de lo contrario se manda un mensaje de error.

#### Algoritmo del Procedimiento de Cambio General.

- 1.-Obtener el nombre de la tabla que está accedando la forma de captura.

- 2.-Revisar que campos se modificaron en la forma de captura.
- 3.-En caso de que no se haya cambiado ninguno se termina el procedimiento.
- 4.-En caso de que si se haya modificado al menos uno de los campos se realizan los siguientes puntos.
- 5.-Inicializar dos variables. Una de ellas contendrá la instrucción de actualización junto con el nombre de la tabla obtenido anteriormente; mientras que la otra se inicializar con la palabra reservada que indique que condiciones debe cumplir el registro que ser actualizado.
- 6.-Realizar una consulta que obtenga los nombres de todos los campos que forman la llave primaria de la tabla proporcionada y aparte de ellos obtener los valores que tienen en la forma de captura y concatenarlos en la variable de caracteres que indicarán las condiciones del renglón a modificar.
- 7.-Solo a los campos que fueron modificados se les concatena en la variable de caracteres que contendrá todos los campos a modificar.
- 8.-Una vez checados todos los campos de la forma de captura, se concatenan las dos variables anteriormente mencionadas.
- 9.-Se ejecuta la variable que contiene la instrucción de actualización.
- 10.-En caso de no existir error en la ejecución se manda un mensaje que indique el éxito de la transacción, de lo contrario se manda un mensaje de error.

### Algoritmo del Procedimiento de Consulta General.

- 1.- Obtener el nombre de la tabla que está accedando la forma de captura.
- 2.- Inicializar la variable de caracteres que contendrá la instrucción de consulta con el nombre de la tabla obtenido anteriormente.
- 3.- Obtener el valor de cada uno de los campos que se llevan tecleados hasta el momento y concatenarlos en la variable de caracteres que contiene la instrucción de consulta.
- 4.- Abrir una ventana sobre la pantalla, en la cual se desplieguen los renglones obtenidos de la consulta delimitada.
- 5.- Una vez que la variable que contiene la instrucción de inserción ha quedado completa, se ejecuta.
- 6.- En caso del que tipo de dato del campo a consultar sea de tipo cadena de caracteres, este se buscará como un patrón que podrá estar incluido en cada uno de los datos que estén en la base de datos para dicho campo.
- 7.- En caso de que la consulta no encuentre el o los registros determinados se envía un mensaje de datos no encontrados. Mientras que para el caso en que no se proporcionen datos que marquen condiciones de búsqueda se enviará un mensaje de no hay valores para consultar.



### **Algoritmo del Procedimiento de Búsqueda General.**

- 1.- Obtener el nombre de la tabla que está accedando la forma de captura.**
- 2.- Inicializar la variable de caracteres que contendrá la instrucción de consulta con el nombre de la tabla obtenido anteriormente.**
- 3.- Obtener el valor de cada uno de los campos que se llevan tecleados hasta el momento y concatenarlos en la variable de caracteres que contiene la instrucción de consulta.**
- 4.- Una vez que la variable que contiene la instrucción de inserción ha quedado completa, se somete a ejecución.**
- 5.- En caso de que la consulta no encuentre el registro determinado se activa unicamente la opción de alta en el menú superior, en caso contrario se coloca toda la información devuelta por la consulta en sus campos respectivos de la forma de captura y se activan unicamente las opciones de baja y cambio en el menú superior.**

### **V.3. DISEÑO DE REPORTE.**

Los reportes constituyen una de las áreas de gran importancia en la aceptación de un sistema por parte del usuario final, ya que es la parte del sistema en donde el usuario identifica si la utilización del equipo de cómputo le proporciona a su trabajo limpieza, organización, control de la información, reducción de procesos manuales, etc.

Se utilizaron dos maneras para realizar reportes, una de ellas es el generador de reportes propio del DBMS, y otra mediante el *DB library/C*, la cual es una herramienta conocida como API (*Application Programmer Interface*).

#### **V.3.1. GENERADOR DE REPORTE.**

El generador de reportes (*Report Writer*) es la herramienta propia del DBMS, la cual consiste en la generación de reportes ayudados por menús y consultas apoyadas en SQL. La ventaja que proporciona esta opción es que el diseño de reportes se realiza en forma rápida y sencilla. Su desventaja es que las opciones para modificar el formato de salida no es muy amplia y limita mucho la presentación y el diseño del mismo. Se diseñaron varios reportes de este tipo en cada una de los sistemas implantados, principalmente para los catálogos de artículos, proveedores, contratistas, inventarios, etc.

#### **V.3.2. REPORTE CON CLIENTE ABIERTO.**

Los reportes que no pudieron implementarse en el reporteador se desarrollaron usando *DB Library/C* que es un conjunto de rutinas y macros de C que permiten interactuar con el servidor. Esta herramienta es en sí una interfaz de programación de aplicaciones (*API*). La operación básica de un

*API* es la siguiente:

- a) El programa comienza el acceso a la base de datos con una llamada que conecta el programa con el manejador y también a una base de datos específica.
  
- b) Para enviar una sentencia de SQL al manejador, el programa construye la sentencia como una cadena de texto en un *buffer* y luego hace una llamada *API* para transferir el contenido del *buffer* al manejador.
  
- c) El programa hace llamadas *API* para comprobar el estado de su petición al manejador así como también para manejar errores.
  
- d) Si la petición es una consulta, el programa utiliza llamadas *API* para recuperar los resultados en los *buffers* del programa. Típicamente, las llamadas devuelven datos renglón a renglón o columna a columna.
  
- e) El programa termina su acceso a la base de datos con una llamada *API* que le desconecta del manejador.

En una configuración cliente-servidor el código para las funciones *API* está localizado en el cliente, donde se ejecuta el programa de aplicaciones. El *software* del manejador está localizado en el servidor, donde reside la base de datos. Las llamadas desde el programa de aplicación al *API* tiene lugar localmente dentro del cliente, pero la comunicación entre el *API* y el manejador tiene lugar sobre una red. Un *API* de este tipo en una arquitectura cliente servidor ofrece la ventaja de minimizar la cantidad de tráfico de red entre la aplicación y el *DBMS*.

La metodología comunmente utilizada por un *API SQL* para acceder al *DBMS* se muestra en la figura V.3.

## EL ACCESO DE UN API A UN DBMS

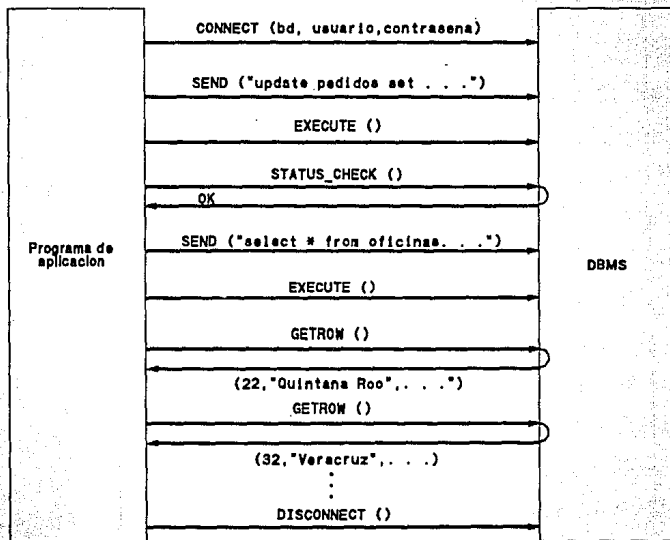


Figura V.3.

## **VI. IMPLANTACION Y PRUEBAS.**

Una de las primicias de la implantación del SIAO fue la liberación paulatina de los subsistemas a cada una de las gerencias involucradas. Esto es, a medida que se terminaran cada uno de los módulos de cada subsistema, se fueran integrando al subsistema como un todo. De esta manera, además de que los módulos empezaban su ciclo productivo, también se terminaba la fase de pruebas.

Después de ser desarrollados cada uno de los módulos que componen a un subsistema, se sometía a una etapa de pruebas, las cuales consistían en ejecutar los procesos críticos, entendiéndose a éstos como los procesos de mayor importancia en el objetivo del subsistema. Este proceso fue llevado a cabo por parte nuestra y del personal gerencial para que de esta manera el módulo fuera puesto en funcionamiento. Es así como podemos distinguir tres fases principales de implantación.

#### **VI.1. FASE CENTRALIZADA.**

Como se mencionó anteriormente, la liberación paulatina de módulos para cada una de las gerencias, nos condujo a implementar los subsistemas en cada uno de sus equipos de cómputo, generando de esta manera que los subsistemas tuvieran una arquitectura centralizada e independiente a los demás subsistemas que intervienen en el SIAO.

Es así como se pusieron en marcha todos los subsistemas, teniendo como consecuencia los siguientes puntos:

- Las pruebas a nivel detalle fueron solucionadas a medida que la gente operativa las iba detectando.
  
- La eficiencia a nivel diseño fue corregida en algunos de sus

aspectos, tales como la modificación de índices, la redistribución de las tablas, etc.

- La eficiencia a nivel servidor fue corregida en algunos de sus aspectos, tales como el *fillfactor*, *memory*, que al fin y al cabo son parámetros que nos sirven para sintonizar el rendimiento del servidor.

En la figura VI.1 se esquematiza la arquitectura implementada en ésta fase.



# IMPLANTACION FASE I ARQUITECTURA MULTIUSUARIO

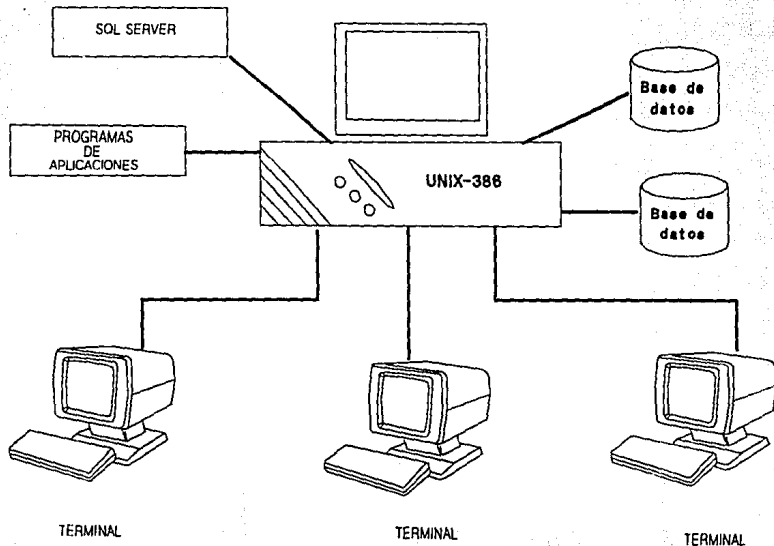


Figura VI.1.

## **VI.2. FASE CLIENTE-SERVIDOR.**

Una vez implementada completamente la fase anterior, se procedió a desarrollar las pruebas para un ambiente cliente servidor, de las cuales se obtuvieron las siguientes observaciones:

- El tiempo de respuesta fué mejorado considerablemente, comprobando la teoría.
  
- La integridad entre los subsistema fué probada exitosamente.
  
- Se comprobó que el rendimiento seguía siendo favorable.

Cabe hacer la aclaración que estas pruebas se realizaron a nivel local, es decir, utilizando dos de las máquinas dedicadas al desarrollo y que la implementación final (figura VI.2) utilizando la VAX 4000 como *server* no ha sido implementada aún debido a que no se tiene el cableado de la red.

## IMPLANTACION FASE II ARQUITECTURA CLIENTE-SERVIDOR

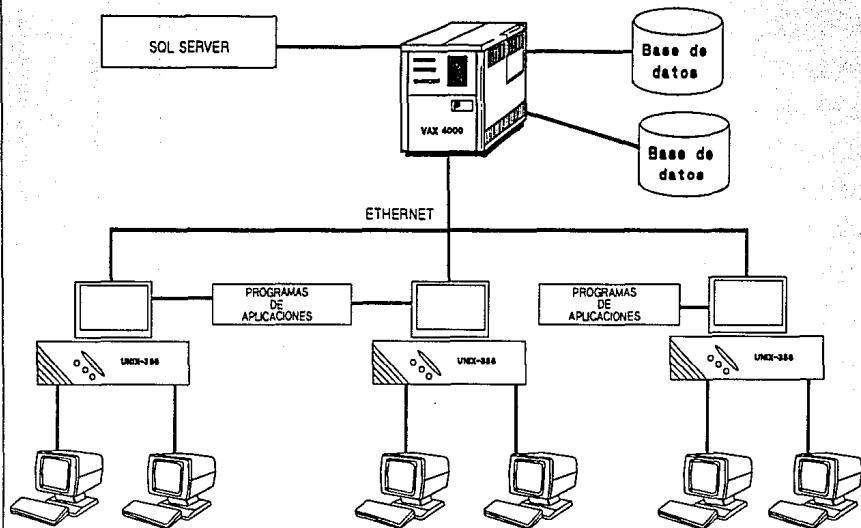


Figura VI.2.

### VI.3. FASE MULTISERVIDOR Y SISTEMAS ABIERTOS.

Afortunadamente y a pesar de que no se cuenta con una infraestructura de *hardware* instalada, el diseño del SIIAO está listo para implementarse en una arquitectura cliente servidor, y no solo eso, sino que además el diseño y el desarrollo quedan abiertos para que fácilmente se tome como base para un sistema de información distribuida. Esto se debe principalmente a que las llamadas entre las diferentes bases de datos sea sustituidas por *stored procedures*. Esta arquitectura se esquematiza en la figura VI.3.

### IMPLANTACION FASE III ARQUITECTURA MULTISERVIDOR

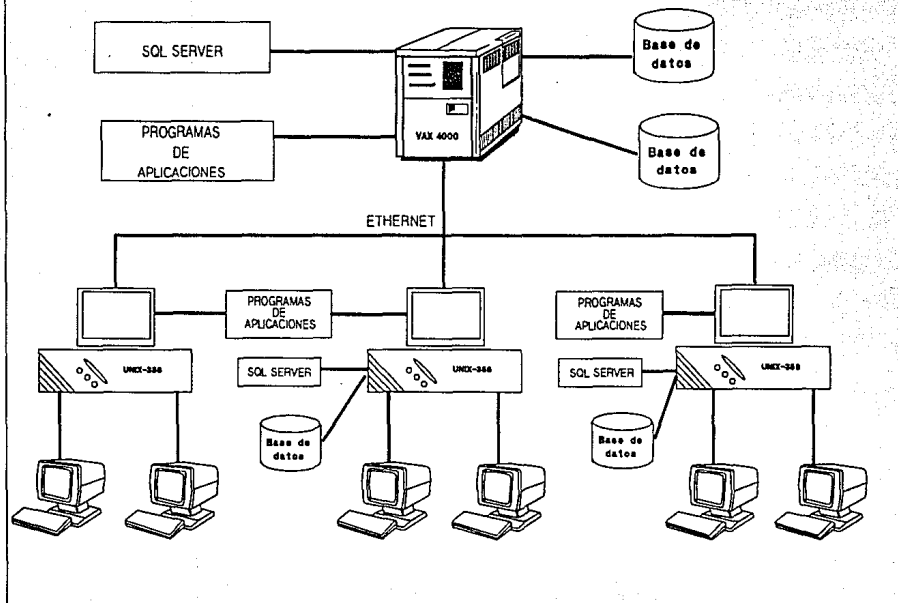


Figura VI.3.

## VII. CONCLUSIONES

La realización del presente sistema surge como columna vertebral para que a partir de él se complementen todas las funciones operativas y técnicas que desempeña el C.A.P.F.C.E. El sistema no contempla a todas las áreas pertenecientes al C.A.P.F.C.E. sino solo aquellas que son básicas para el adecuado control y administración de lo que es el ciclo de construcción de una obra. Sin embargo, el sistema desarrollado presenta un diseño abierto desde el punto de vista del acoplamiento de sistemas, con lo cual la incorporación de nuevas áreas operativas al presente sistema representaría un trabajo de bajo costo financiero y bajo esfuerzo humano, toda vez que la arquitectura básica de diseño y operación quedó claramente definida.

Esta apertura presenta características como la de formar una estandarización de todos los conceptos empleados para la implementación de sistemas así como de todos los formatos de interfaz con el usuario. Otra característica es la de proporcionar información única en catálogos perfectamente delimitados para cada área, teniendo esto como ventaja el proceso distribuido sobre estos datos centralizados.

El presente trabajo proporciona una panorámica general del diseño de procesamiento distribuido en bases de datos en un ambiente cliente-servidor.

Por todo esto, podemos concluir que a partir del planteamiento del problema inicial, la solución implementada resulta ser innovadora debido a que actualmente son pocos los sistemas desarrollados con estas características. Así pues, es de notar que en la teoría, para tener un ambiente de base de datos relacional es indispensable cumplir con ciertas reglas. Dichas reglas en la práctica pierden parte de su valor; tal es el caso de las reglas de normalización; esto es debido a que si se cumplen al pie de la letra todas las tareas que exigen estos esquemas, el rendimiento que puede ofrecer cualquier equipo de cómputo se ve degradado considerablemente. Es por ello que se han desarrollado herramientas como la empleada, que ofrecen opciones en el desarrollo de sistemas que

pueden sustituir parcialmente las reglas antes mencionadas. Una de estas opciones son los *triggers*, los cuales ayudan a mantener la integridad referencial a pesar de que la normalización se vea aparentemente violada en alguna de sus partes.

También, hemos podido observar que el desarrollo de un sistema de *software* utilizando una herramienta de cuarta generación hace posible que su implementación sea realizada en un tiempo menor teniendo como base un análisis ya establecido. Esto confirma que el rediseño de los sistemas ya existentes con herramientas de este tipo son de mayor eficiencia que su sola conversión.



## BIBLIOGRAFIA

### Manuales del Manejador SYBASE:

- .System Administration Guide.
- .APT-Workbench.
- .Data-Workbench User's Guide.
- .Command Reference.
- .DB-Library.

### Libros:

- .Apilque SQL  
James R. Groff, Paul N. Weynberg  
Mc Graw Hill, 1991.
- .Database Management  
Ozkarahan, Esen  
Iberoamericana, 1990.
- .Database Management  
Baisjald, Ralph  
Mc Graw Hill, 1991.
- .Ingenieria de Software  
Pressman  
Prentice-Hall, 1988.
- .Introducción a los sistemas de bases de datos  
C. J. Date  
Addison-Wesley Iberoamericana, 1986.
- .Introducción a los sistemas operativos  
Harvey M. Deitel  
Addison-Wesley Iberoamericana, 1987 .
- .Ingenieria de Software  
Richard Fairley  
Mc Graw Hill, 1988.
- .El lenguaje de programación C  
Brian W. Kernighan, Dennis M. Ritchie  
Prentice Hall, 1985.
- .Engineering Software  
Jag Shodi  
Addison-Wesley Iberoamericana, 1990.

**.Principles of relational database systems**  
Sitansu S. Mittra  
Prentice Hall, 1991.

**.Unix for VMS users**  
Phillip E. Bourne  
Digital Press, 1990.

**.Relational Database Design**  
Paul Winsberg  
Winsberg, 1990.

**Revistas:**

**.BYTE, Product Focus**  
Multiserver Databases, The SQL  
Volumen 15, número 5, p.p. 136-150  
Mc Graw Hill, mayo 1990.

**Artículos:**

**Software AG**  
Entire Technology Glosary  
Enero de 1992, Darmstadt, Alemania.

**SYBASE SQL Server**  
SYBASE Inc.  
Enero de 1991, Emeryville, CA USA.



# APENDICES

## A. SCRIPT PARA LA CREACION DEL BACK-END.

Creación del dispositivo para la base de datos P.G.O.:

```
disk init
name = "pgo_conc",
physname = "/dev/ry",
vdevno = 28,
size = 12000
go
```

Creación de la Base de Datos P.G.O.:

```
create database pgo
on pgo_conc= 5
go
```

Creación del login para la base de Datos P.G.O.:

```
sp_addlogin pgo,pgo,pgo
go
```

Definición del dueño:

```
use pgo
go
sp_changedbowner pgo
go
```

Definición de usuarios:

```
use pgo
go
sp_adduser almacen
go
sp_adduser adquisic
go
sp_adduser concurso
go
```

Creación de una tabla:

```
create table obras
( num_edo      num_edo,
  anoobra     anoobra,
  numobra     char4_num,
  cve_prog    ep_prog,
  cve_subprog char2_num,
  cve_proy    char2_num,
  num_modelo  char1_num,
  num_capitulo char1_num,
  num_prog_fin char2_num,
  num_autoriza char2_num,
  num_municipio char3_num,
  localidad    varchar (16) not null,
  nomobra     varchar (16) not null,
  aulas       tentero,
  laboratorios tentero,
  talleres    tentero,
  anexos      tentero,
  tot_constr  dinero,
  tot_prefa   dinero,
  tot_mob_eq  dinero
)
```

go

Creación de un índice :

```
create unique clustered index ind_obras
on obras ( num_edo,anoobra,numobra)
go
```

Asignación de defaults:

```
sp_bindefault no_propor,"obras.localidad"
go
create default sin_nombre as "SIN NOMBRE"
go
sp_bindefault sin_nombre,"obras.nomobra"
go
```

Definición de un trigger para altas:

```
create trigger alta_obras
on obras
for insert as
begin
if @@rowcount = 0 return
if not exists ( select * from cat_modelos cm,inserted i
                where cm.cve_prog + cm.cve_subprog +
                       cm.cve_proy + cm.num_modelo =
                       i.cve_prog + i.cve_subprog +
                       i.cve_proy + i.num_modelo)
begin
print "El nivel educativo proporcionado no existe"
rollback tran
return
end
if not exists ( select * from cat_prog_fin cpf, inserted i
                where cpf.num_prog_fin = i.num_prog_fin )
begin
print "El tipo de programa proporcionado no existe"
rollback tran
return
end
if not exists ( select * from cat_capitulos cc, inserted i
                where cc.num_capitulo = i.num_capitulo )
begin
print "El capitulo proporcionado no existe"
rollback tran
return
end
if not exists ( select * from cat_autoriza ca, inserted i
                where ca.num_autoriza = i.num_autoriza )
begin
print "La autorizacion proporcionada no existe"
rollback tran
return
end
if not exists (select * from cat_municipios cm, inserted i
                where cm.num_edo = i.num_edo and
                       cm.num_municipio = i.num_municipio )
begin
print "El municipio proporcionado no existe"
rollback tran
return
end
end
go
```

Definición de un trigger para bajas:

```
create trigger baja_obras
on obras
for delete as
begin
if @@rowcount = 0 return
delete obra_elem from obra_elem oe,deleted d
    where oe.num_edo = d.num_edo and
    oe.anio_obra = d.anio_obra and
    oe.num_obra = d.num_obra
delete obra_articulo from obra_articulo oa,deleted d
    where oa.num_edo = d.num_edo and
    oa.anio_obra = d.anio_obra and
    oa.num_obra = d.num_obra
delete obra_guia from obra_guia og,deleted d
    where og.num_edo = d.num_edo and
    og.anio_obra = d.anio_obra and
    og.num_obra = d.num_obra
end
go
```



### Definición de un trigger para cambios:

```
create trigger cambia_obras on obras
for update as
begin
if @@rowcount = 0 return
if update ( cve_prog ) or update ( cve_subprog ) or update
( cve_proy ) or update ( num_modelo )
begin
if not exists ( select * from cat_modelos cm, inserted i
where cm.cve_prog + cm.cve_subprog +
cm.cve_proy + cm.num_modelo =
i.cve_prog + i.cve_subprog +
i.cve_proy + i.num_modelo)*1
begin
print *El nivel educativo proporcionado no existe*
rollback tran
return
end
else
begin
delete obra_elem from obra_elem oe,deleted d
where oe.num_edo = d.num_edo and
oe.anio_obra = d.anio_obra and
oe.num_obra = d.num_obra
delete obra_articulo from obra_articulo oa,deleted d
where oa.num_edo = d.num_edo and
oa.anio_obra = d.anio_obra and
oa.num_obra = d.num_obra
delete obra_guia from obra_guia og,deleted d
where og.num_edo = d.num_edo and
og.anio_obra = d.anio_obra and
og.num_obra = d.num_obra
end
end
if update ( num_prog_fin ) and
not exists ( select * from cat_prog_fin cpf, inserted i
where cpf.num_prog_fin = i.num_prog_fin )
begin
print *El tipo de programa proporcionado no existe*
rollback tran
return
end
if update ( num_capitulo ) and
not exists ( select * from cat_capitulos cc, inserted i
where cc.num_capitulo = i.num_capitulo )
begin
print *El capitulo proporcionado no existe*
rollback tran
return
end
if update ( num_autoriza ) and
not exists ( select * from cat_autoriza ca, inserted i
```

```

        where ca.num_autoriza = i.num_autoriza )
begin
    print "La autorizacion proporcionada no existe"
    rollback tran
    return
end

if update ( num_edo ) or update( num_municipio ) and
    not exists ( select * from cat_municipios cm, inserted i
        where cm.num_edo = i.num_edo and
              cm.num_municipio = i.num_municipio )
begin
    print "El municipio proporcionado no existe"
    rollback tran
    return
end

if update ( num_edo ) or update ( anio_obra )
    or update ( num_obra )
begin
    update obra_elem set oe.num_edo = i.num_edo,
        oe.anio_obra = i.anio_obra,
        oe.num_obra = i.num_obra
from obra_elem oe, inserted i
    update obra_guia set og.num_edo = i.num_edo,
        og.anio_obra = i.anio_obra,
        og.num_obra = i.num_obra
from obra_guia og, inserted i
    update obra_articulo set oa.num_edo = i.num_edo,
        oa.anio_obra = i.anio_obra,
        oa.num_obra = i.num_obra
from obra_articulo oa, inserted i
end
end
go

```

**B. DICCIONARIO DE DATOS**

**ALMACEN**

DATO	DESCRIPCION	TIPO DATO		TABLAS
anio_pedido c_art	Año del pedido. Consecutivo artículo.	anio, c_art,	no nulo no nulo	pedido_a. rec_ba,es_bodar, existencias, conc_env_art, sal_env_art.
cant_enviada cant_enviar cant_pedida cant_piezas cant_recibida concepto cons_pedido cons_prov fecha_embarque fecha_entrada fecha_doc fecha_sal g_art	Cantidad de artículo enviada. Cantidad de artículos a enviar. Cantidad de artículos pedidos. Cantidad de artículo en existencia. Cantidad de artículos recibida. Concepto de entradas y salidas. Consecutivo pedido. Consecutivo proveedor. Fecha de embarque de artículos. Fecha de entrada a bodega. Fecha de alta a la bodega. Fecha de salida de los artículos. Grupo artículo.	sentero sentero sentero sentero sentero varchar(20) no nulo no nulo fecha fecha fecha fecha g_art	no nulo no nulo no nulo no nulo no nulo no nulo no nulo no nulo no nulo no nulo no nulo no nulo no nulo	pedido_a. rec_ba,es_bodar, existencias, conc_env_art, sal_env_art. sal_env_art. conc_envio_art. pedido_a existencias. rec_ba es_bodar. pedido_a pedido_a pedido_a rec_bode es_bodar. sal_envi. pedido_as,rec_ba, es_bodar, existencias, conc_envio_art, sal_env_art.
nom_autorizo nom_formulo nombre_chofer	Nombre del autorizante. Nombre del que formuló. Nombre del chofer.	varchar(30) varchar(30) varchar(30)	nulo nulo nulo	sal_envi. sal_envi. sal_envi.
num_bod	Número de bodega.	num_bod	no nulo	rec_bode,rec_ba es_bodar, existencias, conc_envio_art, sal_envi, sal_envi_art.

ALMACEN (Cont...)

DATO	DESCRIPCION	TIPO DATO		TABLAS
num_concentrado num_edo_destino	Número de concentrado. Número de estado destino.	char4_num num_edo	no nulo no nulo	conc_envio_art conc_envio_art, sal_envi sal_env_art
num_embarque num_prog_fin num_remision num_talon placas precio_unitario	Número de embarque. Número de programa de financiam. Número de remisión. Número de talón. Placas del transporte. Precio unitario del artículo.	char4_num char2_num varchar(10) char4_num varchar(6) dinero	no nulo no nulo no nulo no nulo nulo no nulo	sal_envi. rechode. sal_envi. sal_envi. pedido_a rec_ba,es_bodar, existencias, conc_envio_art, sal_env_art.
prioridad s_art	Prioridad de salida del artículo. Subgrupo artículo.	char1_num s_art	no nulo no nulo	conc_envio_art. pedido_as,rec_bode rec_ba,es_bodar, existencias, conc_envio_art, sal_env_art.
t_art	Subgrupo artículo.	s_art	no nulo	pedido_as,rec_bode rec_ba,es_bodar, existencias, conc_envio_art, sal_env_art.
tipo_imp tipo_pedido tipo_transp	Tipo impuesto. Tipo pedido. Tipo transporte.	char1_alfa tipo_pedido varchar(10)	no nulo no nulo nulo	pedido_a pedido_as sal_envi.

**ADQUISICIONES.**

<b>DATO</b>	<b>DESCRIPCION</b>	<b>TIPO DATO</b>		<b>TABLAS</b>
anio_obra	Año de la obra.	anio	no nulo	explo_mob, explo_pref, batch_obras_mob, batch_obras_pref, orden_suministro, ord_sum_art, ord_env_mob, oe_art_pref, oe_art_mob.
anio_pedido	Año del pedido.	anio	no nulo	pedidos, ped_art disp_suministro ord_sum_art.
anio_pedido_as	Año del pedido asignado.	anio	no nulo	pedido_asig, ped_as_art.
autorizacion	Autorización.	bit	no nulo	ord_env_mob. pedidos, pedido_asign.
c_art	Consecutivo Artículo.	c_art	no nulo	partida_art, ped_art,ped_as_art ped_art mod.
c_art	Consecutivo articulo.	c_art	no nulo	cat_articulos, cat_kit_art, art_precio,inv_ini art_equi,disp_mob, disp_pref, disp_suministro, ord_sum_art, oe_art_pref, oe_art_mob, conc_env_art, cat_kit_art.
c_art_kit	Consecutivo articulo kit.	c_art	no nulo	cat_kit_art.
c_art_or	Consecutivo articulo or	c_art	no nulo	art_equi.

**ADQUISICIONES (Cont...)**

<b>DATO</b>	<b>DESCRIPCION</b>	<b>TIPO DATO</b>	<b>TABLAS</b>
c_p	Código Postal	char5_num	cat_prov.
c_pag	Consecutivo paquete.	c_art	cat_paquetes.
calle	Calle.	varchar(10)	cat_prov.
cant_art	Cantidad de artículos.	tentero	partida_art.
cant_enviada	Cantidad enviada.	sentero	conc_env_art,
			sal_con_enar.
cant_equi	Cantidad equivalente.	sentero	art_equi.
cant_gana	Cantidad de artículos al ganador.	tentero	ganador.
cant_lici	Cantidad de artículos licitacitada.	tentero	lici_art_paq.
cant_ofer	Cantidad de artículos ofertada.	tentero	prov_lici.
cant_or	Cantidad Or	sentero	art_equi.
cant_pedida	Cantidad de artículos pedida.	tentero	ped_art,
			ped_art_mod,
			ped_as_art
cant_transferida	Cantidad transferida	sentero	conc_env_art.
clase_pedido	Clase pedido.	clase_pedido	pedidos.
col	Colonia.	varchar(30)	cat_prov.
compromiso	Cantidad de artículos necesarios.	sentero	explo_mob,
			explo_pref.
cons_desc	Consecutivo descripción.	c_art	ped_art,
			ped_as_art,
			desc_larga
cons_ord_sum	Consecutivo orden de suministro		orden_sumministro,
			ord_sum_art
cons_orden_envio	Consecutivo orden de envío.		ord_env_mob,
			ord_env_pref,
			oe_art_pref,
			oe_art_mob.
cons_pedido	Consecutivo de pedido.		disp_sumministro,
			ord_sum_art.
cons_pedido	Consecutivo pedido.	c_art	pedidos, ped_art
			ped_mod,
			ped_art_mod

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

**ADQUISICIONES (Cont...)**

<b>DATO</b>	<b>DESCRIPCION</b>	<b>TIPO DATO</b>	<b>TABLAS</b>
cons_pedido_as	Consecutivo pedido asignado.	c_art no nulo	pedido_asign ped_as_art
cons_prov	Consecutivo proveedor.	c_art no nulo	cat_prov, prov_lic ganador,
cve_asigna	Clave asignación.	char4_num no nulo	pedido_asign.
cve_licl	Clave licitación.	char4_num no nulo	licitacio, lici_art_paq, partida_art, pedidos. cat_articulos.
cve_spp_art	Clave spp del artículo.		
desc_art	Descripción del artículo.	text	desc_larga.
desc_paq	Descripción del paquete.	varchar(67) no nulo	cat_paquetes.
disponibilidad	Disponibilidad de artículos.	sentero	disp_suministro, disp_mob, disp_pref.
enviado	Cantidad de artículos enviados.	sentero	explo_pref, explo_mob
envio	Cantidad de artículos de envío.	sentero	ce_art_pref, ce_art_mob. art_precio.
fecha_alta_ precio	Fecha de alta del precio	fecha	
fecha_apert	Fecha de apertura.	fecha	licitacion.
fecha_embarque	Fecha de embarque.	fecha	pedidos, pedido_asign.
fecha_fallo	Fecha de fallo.	fecha	licitacion.
fecha_fianza	Fecha fianza.	fecha	pedidos, pedidos_asig.
fecha_modif	Fecha de modificación.	fecha	ped mod.
fecha_ord_sum	Fecha de orden de suministro.	fecha	orden_suministro.
fecha_orden_ envio	Fecha de orden de suministro.	fecha	ord_env_mob, ord_env_pref.
fecha_pedido	Fecha del pedido.	fecha no nulo	pedidos, pedidos_asign.

ADQUISICIONES (Cont...).

DATO	DESCRIPCION	TIPO DATO		TABLAS
fianza	Fianza.	dinero	no nulo	pedidos
g_art	Grupo artículo.	g_art	no nulo	pedidos asign. lici_art_paq, partida_art, ped_art,ped_as_art
g_art	Grupo artículo.	g_art	no nulo	cat_articulos, cat_kit_art, art_precio,inv_ini art_equi,disp_mob, disp_pref, disp_suministro, ord_sum_art, oe_art_pref, oe_art_mob, conc_env_art, sal_conc_enar.
g_art_kit	Consecutivo artículo kit.	g_art	no nulo	cat_kit_art.
g_art_or	Consecutivo artículo or	g_art	no nulo	art_equi.
g_paq	Grupo paquete.	g_art	no nulo	lici_art_paq partida_art ped_art,ped_as_art inventario
modificacion	Bandera de modificación.	sentero char1_num		inv_ini ord_env_mob ord_env_pref. cat_articulos. cat_articulos, conc_env_art, sal_conc_enar.
nom_articulo	Nombre del artículo.	varchar(67)	no nulo	pedidos.
num_bod	Número de bodega.			ord_env_mob, ord_env_pref, conc_env_art, sal_conc_enar.
num_compra	Número de comprador.	char2_num	no nulo	
num_concentrado	Número de concentrado.			



**ADQUISICIONES (Cont..)**

DATO	DESCRIPCION	TIPO DATO	TABLAS
num_edo	Número de estado.	num_edo no nulo	explo_mob, explo_pref,inv_ini disp_pref, disp_mob, disp_suministro.
num_edo_destino	Número de estado destino.	num_edo no nulo	batch_obras_mob, batch_obras_pref, ord_env_pref, ord_env_mob, oe_art_mob, oe_art_pref, orden_suministro, ord_sum_art, conc_env_art, sal_conc_enar, rec_jz_obra.
num_edo_origen	Número de estado origen	num_edo no nulo	ped_mod, ped_mod_art.
num_modif	Número de modificación.	char4_num no nulo	batch_obras_mob, batch_obras_pref, ord_env_pref, ord_env_mob, oe_art_mob, oe_art_pref, orden_suministro, ord_sum_art, conc_env_art, sal_conc_enar, rec_jz_obra.
num_obra	Número de obra.	char4_num no nulo	batch_obras_mob, batch_obras_pref, ord_env_pref, ord_env_mob, oe_art_mob, oe_art_pref, orden_suministro, ord_sum_art, conc_env_art, sal_conc_enar, rec_jz_obra.

ADQUISICIONES (Cont...)

DATO	DESCRIPCION	TIPO DATO	TABLAS
num_partida	Número de partida.	char4_num no nulo	lici art pag, partida art prov lici, ganador, pedidos, pedido asign.
precio_ precio_gana precio_ofer precio_pedido	Precio de inventario. Precio del ganador. Precio ofertado. Precio pedido.	dinero dinero no nulo dinero no nulo dinero no nulo	inv ini. ganador. prov lici. ped art, ped as art, ped art mod.
precio_unitario	Precio unitario del articulo.	dinero	art precio, disp suministro, disp mob, disp pref, ord sum art, oe art mob, oe art pref, conc env art, sal conc enar.
prioridad	Prioridad	char1_num	batch obras mob, batch obra pref, ord env pref, ord env mob.
razon_social recepcion reg_spp	Razón social. Bandera de recepción. Registro SPP.	varchar(60) no nulo char1_num reg_spp	cat prov rec jz obra. cat prov.

ADQUISICIONES (Cont...)

DATO	DESCRIPCION	TIPO DATO	TABLAS
s_art	Subgrupo articulo.	s_art no nulo	cat_articulos, cat_kit_art,d art_precio,inv_ini art_equi,disp_mob, disp_pref, disp_suministro, ord_sum_art, oe_art_pref, oe_art_mob, conc_env_art, sal_conc_enar.
s_art	Subgrupo articulo.	s_art no nulo	lic_art_paq partida_art ped_art_mod ped_art,ped_as_art
s_art_kit	Subgrupo articulo kit.	s_art no nulo	cat_kit_art.
s_art_or	Subgrupo articulo or	s_art no nulo	art_equi.
s_paq	Subgrupo paquete.	s_art no nulo	lici_art_paq partida_art ped_art_mod ped_art,ped_as_art ord_sum_art,
suministro	Cantidad de articulos de suministro	entero	
t_art	Tipo articulo.	t_art no nulo	cat_articulos, cat_kit_art, art_precio,inv_ini art_equi,disp_mob, disp_pref, disp_suministro, ord_sum_art, oe_art_pref, oe_art_mob, conc_env_art, sal_conc_enar.

**ADQUISICIONES (Cont...)**

DATO	DESCRIPCION	TIPO DATO	TABLAS
t_art_kit t_art_or tel tipo_imp	Tipo artículo kit. Tipo artículo or Teléfono. Tipo impuesto.	t_art no nulo t_art no nulo telefono tipo_imp no nulo	lici art_paq, partida art, ped_art_mod ped_art,ped_as_art cat_kit_art. art_equi. cat_prov.
tipo_licí tipo_partida tipo_pedido	Tipo licitación. Tipo partida. Tipo de pedido.	tipo_licí no nulo tipo_partida no nulo	pedidos pedidos asign. licitación. lici art_paq.
tipo_pedido	Tipo pedido.	id_tipo_pedido no nulo	disp_sumnistro, ord_sum_art. pedidos, ped_mod ped_art_mod, ped_art
tipo_pedido_as	Tipo pedido asignado.	id_tipo_pedido no nulo	pedido_asign, ped_as art.
transac um_articulo uni_licí	Tipo de movimiento en mod.Pedidos Unidad de medida del artículo. Unidad de licitación.	transac char2 alfa uni_licí no nulo	ped_mod. cat_articulos. lici art_paq.

## CONCURSO DE OBRA

DATO	DESCRIPCION	TIPO DATO		TABLAS
abrev_cub	Abreviatura cubierta.	char1_alfa	no nulo nulo	cat_cubiertas, obra_edificio.
abrev_mod	Abreviatura modalidad.	char2_alfa	no nulo nulo nulo nulo	cat_modalidad, concursos, bitacora_conc, obra_edificio.
abrev_tipo_conc	Abreviatura tipo concurso.	char1_alfa	no nulo no nulo no nulo	cat_tipo_conc concursos, bitacora_conc.
anexos	Anexos.	tentero	no nulo	obra_edificio.
anio_conc	Año del concurso.	anio	no nulo	concursos, bitacora_conc participantes.
anio_obra	Año de la obra.	anio	no nulo	obras_empaq, obra_edificio, participante_edificio.
aulas	Aulas.	tentero	no nulo	obra_edificio.
cap_contable	Capital contable.	dinero	no nulo	cat_contratistas.
costo_capfca	Costo Capfca.	dinero	no nulo	obra_edificio.
costo_m2	Costo de metro cuadro de estructura	dinero	no nulo	cat_costo_estructura.
descripcion	Descripción de la estructura.	varchar(40)	no nulo	cat_costo_estructura.
dir_cont	Dirección del contratista.	direccion	no nulo	cat_contratista.
dir_jz	Dirección de la Jefatura de Zona.	direccion	no nulo	cat_jz.
edificio	Edificio.	edificio	no nulo	obra_edificio, participante_edificio.
entre_ejes	Cantidad de entre ejes.	real	no nulo	obra_edificio.
fecha_alta	Fecha del sistema.	fecha	no nulo	cat_contratistas.
fecha_apert	Fecha de apertura del concurso.	fecha	no nulo	concursos, bitacora_conc.
fecha_contrato	Fecha de contrato del contratista.	fecha	no nulo	concursos.

**CONCURSO DE OBRA (Cont..)**

<b>DATO</b>	<b>DESCRIPCION</b>	<b>TIPO DATO</b>		<b>TABLAS</b>
fecha_entre	Fecha de entrega de documentos.	fecha	no nulo	concursos,
fecha_fallo	Fecha de fallo del concurso.	fecha	no nulo	bitacora_conc. concursos
fecha_fax	Fecha llegada del esquema dictamen.	fecha	no nulo	bitacora conc. concursos,
fecha_fin	Fecha de terminación obra.	fecha	no nulo	bitacora conc.
fecha_inicio	Fecha de inicio de la obra.	fecha	no nulo	obras_empaq.
fecha_recibido	Fecha de recibido de la propuesta.	fecha	no nulo	obras_empaq. participantes.
grupo	Grupo al que pertenece un analista.	char1_alfa	no nulo	cat_analistas.
imp_acta	Importe total del acta de apertura	money	no nulo	participantes.
imp_edif	Importe por edificio.	dinero	no nulo	participantes.
imp_oext	Importe por obra exterior.	dinero	no nulo	participantes.
inscritis	Empresas inscritas al concurso.	uno	nulo	concursos, bitacora_conc
jefe_zona	Nombre del Jefe de Zona.	varchar(40)	no nulo	cat_jz.
laboratorios	Cantidad de Laboratorios.	tentero.	no nulo	obra edificio.
lugar_monto	Lugar ocupado por monto.	uno	no nulo	participantes.
lugar_punt	Lugar ocupado por puntuación.	uno	nulo	participantes.
nom_analista	Nombre del analista.	varchar(40)	no nulo	cat_analistas.
nom_cub	Nombre de la cubierta	varchar(20)	no nulo	cat_cubiertas.
nom_mod	Nombre de la modalidad.	varchar(25)	no nulo	cat_modalidad.
nom_tipo_conc	Nombre del tipo de concurso.	varchar(7)	no nulo	cat_tipo conc.
num_conc	Número de concurso.	num_conc	no nulo	concursos, bitacora_conc, participantes.
num_cub	Número de cubierta.	char2 num	no nulo	
num_edo	Número de estado.	num_edo	no nulo	obras_empaq obra edificio, participante edi- ficio, cat_jz, concursos.

## CONCURSO DE OBRA (Cont...)

DATO	DESCRIPCION	TIPO DATO		TABLAS
num_estruc	Número de estructura.	char3_num	no nulo	obra edificio, cat_costo_estruc- tura.
num_mod	Número de modalidad	char2_num	no nulo	cat_modalidad.
num_obra	Número de obra.	char4_num	no nulo	obras empaq participante_edi- ficio.
num_paq	Número de paquete de obra.	char3_num	no nulo	obra empaq, concursos, bitacora_conc.
num_tipo_conc obras	Número del tipo de concurso. Cantidad de obras por concurso.	char1_num uno	no nulo no nulo	cat_tipo_conc concursos bitacora_conc.
observaciones participantes	Observaciones al contratista. Cant. de participantes por concurso	varchar(60) uno	nulo	participantes. concursos, bitacora_conc.
por_ind	Porcentaje de indirectos.	float	no nulo	participantes.
puntuacion	Puntuación obtenida por contratista	real	no nulo	participantes.
razon_social	Nombre de la razón social.	varchar(40)	no nulo	cat_contratistas.
representante	Nombre del representante.	varchar(40)	no nulo	cat_contratistas.
spp	RFC de la razón social.	rfc	no nulo	cat_contratistas, participantes participante_edi- ficio.
superficie	Superficie por estructura.	real	no nulo	obra edificio,cat_ costo estructura. obra edificio.
talleres	Cantidad de Talleres.	tentero	no nulo	cat_contratistas.
tel_cont	Teléfono del contratista.	telefono	no nulo	cat_jz
tel_jz	Teléfono de la Jefatura de Zona.	telefono	no nulo	participantes.
tot_cal_edif	Total de importe de los edificios.	dinero	no nulo	participantes.
tot_cal_oext	Total de importe de las obras exts.	dinero	no nulo	participantes.

## PROGRAMA GENERAL DE OBRA

DATO	DESCRIPCION	TIPO DATO	TABLAS
abrev_edo	Abreviatura del estado.	char2_num no nulo	cat_estados, obras, obra_elem, obra_guia, obra_articulo, cat_municipios. cat_prog_fin.
abrev_prog_fin anexos	Abreviatura del Programa de Finac. Cantidad de anexos.	char3_alfa no nulo sentero no nulo	obras, cat_elementos. obras, obra_elem, obra_guia, obra_articulo.
anio_obra	Año de la obra.	anio no nulo	obra_articulo. obras, cat_elementos. obra_articulo.
articulos aulas	Cantidad de artículos. Cantidad de aulas.	sentero no nulo sentero no nulo	obras, cat_elementos. obra_articulo.
c_art c_guia	Consecutivo artículos. Consecutivo guía.	t_art no nulo c_art no nulo	cat_guias, obra_guia, elemento_guia. modelo_elem
construcciones costo_x_m2 cve_autoriza cve_prog	Cantidad de construcciones. Costo por metro cuadrado de estruc. Clave de autorización. Clave de Programa Educativo.	sentero dinero char(2) no nulo	cat_estructuras. cat_autoriza. cat_prog_educ, cat_subprog_educ, cat_proy_educ, cat_modelos, obras, modelo_elem
cve_proy	Clave del Subproyecto Educativo.	char1_num no nulo	cat_proy_educ, cat_modelos, obras, modelo_elem.
cve_subprog	Clave del Subprograma Educativo	cha2_num no nulo	cat_subprog_educ, cat_proy_educ, cat_modelos,obras, modelo_elem.



**PROGRAMA GENERAL DE OBRA (Cont...)**

<b>DATO</b>	<b>DESCRIPCION</b>	<b>TIPO DATO</b>	<b>TABLAS</b>
edificio elementos equipos	Edificio Cantidad de elementos. Cantidad de equipos.	edificio sentero no nulo sentero no nulo	obra_elem obra_elem. obra_elem, modelo_elem cat_elementos. cat_autoriza. obra_guia, elemento_guia. obra_articulo. obra_guia, elemento_guia.
entre_ejes fecha_autoriza guias	Cantidad de entre ejes. Fecha de autorización de la Obra. Cantidad de guías	real fecha sentero no nulo	obra_articulo. obra_guia, elemento_guia. obras, cat_eleemntos. obra_elem obras obra_elem, modelo_elem. cat_capitulos. cat_estados. cat_estructuras.c cat_guias. cat_municipios.
g_art g_guia	Grupo artículo. Grupo guía.	g_art no nulo g_guia no nulo	obras. cat_prog_educ. cat_prog_fin. cat_proy_educ. cat_subprog_educ. cat_capitulos obras.
laboratorios	Cantidad de laboratorios.	sentero no nulo	cat_estados, obras, obra_elem, obra_guia, obra_articulo.
locales localidad mobiliarios	Cantidad de Locales Nombre de la localidad. Cantidad de mobiliarios.	sentero no nulo varchar(16) sentero no nulo	
nom_capitulo nom_edo nom_estruc nom_guia nom_municipio nom_obra nom_prog nom_prog_fin nom_proy nom_subprog num_capitulo	Nombre del capítulo. Nombre del estado. Nombre de la estructura. Nombre de la guía Nombre del municipio. Nombre de la obra. Nombre del programa educativo. Nombre del programa de financiam. Nombre del proyecto. Nombre del subprograma educativo. Número de capítulo.	varchar(13) varchar(20) no nulo varchar(20) varchar(60) varchar(32) varchar(16) no nulo varchar(25) no nulo varchar(16) no nulo varchar(25) varchar(25) char2_num no nulo	
num_edo	Número de estado.	char2_num no nulo	

**PROGRAMA GENERAL DE OBRA (Cont..)**

<b>DATO</b>	<b>DESCRIPCION</b>	<b>TIPO DATO</b>	<b>TABLAS</b>
num_elemento num_autoriza num_estruc	Número de elemento. Número de autorización. Número de estructura.	char4_num char2_num char3_num	cat_elementos. cat_autoriza. cat_estructuras, nivel_estruc, obra_elem. cat_modelos,obras. cat_municipios, obras.
num_modelo num_municipio	Número de modelo. Número del municipio	char1_num no nulo char3_num no nulo	obras, obra_elem, obra_guia, obra_articulo. cat_prog_fin obras.
num_obra	Número de obra.	char4_num no nulo	cat_guias, elemento_guia, obra_guia. obra_articulo. cat_guias, obra_guia, elemento_guia. cat_estructuras. obra_articulo. cat_guia, obra_guia, elemento_guia. obras, cat_elementos. obras.
num_prog_fin	Número del programa de financiam.	char2_num no nulo	obras.
opcion_guia	Opción guía.	char1_num	cat_guias, elemento_guia, obra_guia. obra_articulo. cat_guias, obra_guia, elemento_guia. cat_estructuras. obra_articulo. cat_guia, obra_guia, elemento_guia. obras, cat_elementos. obras.
s_art s_guia	Subgrupo artículo. Subgrupo guía.	s_art no nulo no nulo	cat_guias, obra_guia, elemento_guia. cat_estructuras. obra_articulo. cat_guia, obra_guia, elemento_guia. obras, cat_elementos. obras.
sup_x_es t_art t_guia	Superficie por entre ejes. Tipo artículo. Tipo guía.	real t_art no nulo	cat_guia, obra_guia, elemento_guia. obras, cat_elementos. obras.
talleres	Cantidad de talleres	entero	obras, cat_elementos. obras.
tot_constr	Total para construcción.	dinero. no nulo	obras.
tot_mobeq	Total para mobiliario y equipo.	dinero. no nulo	obras.
tot_pref	Total para prefabricado.	dinero. no nulo	obras.

# GLOSARIO

**AGENTE.** En términos de relaciones cliente servidor, un server puede asumir el papel de un cliente, para hacer uso de los servicios de otro server, con el objeto de satisfacer los requerimientos del cliente original. Este proceso es llamado "Efecto cascada".

**API (Application Programming Interface).** Es un punto específico de entrada de un producto que permite a los programas invocar la funcionalidad o comunicarse con otros programas. Por ejemplo, un programa escrito en C puede comunicarse con un mainframe utilizando las conexiones API de DB-library.

**BACK-END.** 1. Es la parte lógica de la aplicación la cual típicamente reside en el Host. 2. Es el Host mismo. En términos de una arquitectura cliente servidor, el servidor es el Back End.

**BASE DE DATOS REMOTA.** Es una base de datos que cumple con una de las formas más antiguas de la computación cliente servidor, en la cual la lógica de aplicaciones y el manejo de los datos es dividida. La interface con el usuario y la aplicación están en una plataforma y la base de datos en otra. Actualmente, esto es normalmente llevado a cabo utilizando gateways.

**BASES DE DATOS DISTRIBUIDAS.** Es una base de datos dentro de una red cuyos componentes (archivos, vistas, tablas, etc.) residen en mas de un sistema. Puede ser accesada o actualizada por cualquier otro nodo dentro de la red; cuando es hecha de manera sincronizada se requiere tecnología Two Phase Commit.

**BROCKER.** Es parte del software que recibe peticiones de los clientes y los conecta con los servers apropiados. En contraste con un agente, el broker actúa como un acoplador independiente.

C. Es un lenguaje de programación completo desarrollado originalmente por desarrolladores de

sistemas operativos. (UNIX está implementado en C), pero hoy en día es ampliamente utilizado como software de workstations y minicomputadoras.

**CLIENTE.** Cualquier elemento dentro de un ambiente de cómputo que solicita un servicio disponible desde cualquier parte del ambiente para realizar las tareas del usuario. Por ende, el cliente actúa como una interface entre los servicios disponibles y los usuarios del sistema.

**CLIENTE SERVIDOR.** Es un modelo lógico usado para crear un procesamiento cooperativo que constituye el fundamento de las aplicaciones de software de un gran número de distribuidores y empresas.

Originalmente creado en la década de los 70's durante las actividades de estandarización de la ISO/OSI, el concepto de cliente servidor es independiente de la arquitectura de hardware y de los sistemas operativos. Este término es utilizado para referirse a la filosofía de diseño, más que para describir al software o a los componentes de hardware. En la práctica existen dos variaciones del término :

- 1.- En la primera variación, tanto el cliente como el servidor son parte del software que pueden o no residir en la misma computadora, por ejemplo, programas de aplicación y la base de datos.
- 2.- En la segunda variación, el término cliente está restringido a una estación de trabajo en ambiente monousuario, como una combinación de software y hardware, y el servidor es una combinación de hardware multiusuario, software de la base de datos y otros servicios.

**CONECTIVIDAD.** La habilidad de dos o más elementos de software y hardware dentro de un ambiente de cómputo heterogéneo para ligarse uno con el otro. Este es un pre-requisito para procesamiento distribuido.

**DBMS (Data Base Management System).** Es un sistema de administración de bases de datos.

**DEFAULT.** Es un valor proporcionado por el manejador en caso de ser omitidos.

**DESPACHADOR.** Es parte del software que inicia los procesos del server en un ambiente cliente servidor.

**DICCIONARIO DE DATOS.** Es un depósito central de información de los sistemas de software o aplicaciones de procesamiento de datos usados por una empresa esta información puede ser vista como un modelo de los datos. En particular, incluye las siguientes clases de información acerca de un sistema de procesamiento de información :

- Que datos son almacenados y donde. Esto datos reflejan la estructura de información de una organización.
- Cuales objetos de procesamiento trabajan con los datos. Estos datos reflejan la estructura funcional de una organización.
- Quienes crean los datos y los objetos que procesan los datos. Esto datos reflejan al personal que utiliza los sistemas de procesamiento.
- Reglas de procesamiento a ser aplicadas cuando se hace referencia a los elementos de datos con el propósito de garantizar la consistencia e integridad de los datos.

**DOWNIZING.** En el aspecto de hardware, se refiere al reemplazo de sistemas grandes con pequeños y/o menos costosos. En consideración, este concepto está siendo ampliamente utilizado para reducir costos de producción. Esto ha derivado en el uso de la palabra rightsizing para expresar la idea de vender hardware y software.

**ESCALABILIDAD.** Es la habilidad de modificar un ambiente de proceso de datos por medio de un

downsizing o un upsizing.

**FILLFACTOR.** Es el porcentaje con el cual se llenan las páginas de datos e índices.

**FRONT-END.** Es una parte del equipo y/o software que actúa como una interfaz entre el usuario y el back end. En términos de una arquitectura cliente-servidor, el cliente es el front-end.

**GATEWAY.** Es un nodo físico o lógico que ejecuta el protocolo de comunicación u operaciones de sintaxis, y de esta manera conectar redes o dispositivos incompatibles. Dos ejemplos comunes son los gateway de comunicaciones y los gateway de SQL.

**HETEROGENEO.** Es un término usado en la conexión de hardware y software para indicar que ellos tienen diferentes arquitecturas. Estos sistemas son más comunes en el mercado actual; la habilidad del software para correr sin modificaciones representa un mayor costo y factor de eficiencia.

**HOMOGENEO.** Es un término utilizado en sistemas múltiples de hardware y software para indicar que comparten una arquitectura común. En la práctica, las configuraciones homogéneas son raras de encontrar, pero pueden ser simuladas para usuarios finales a través de una imagen única del sistema.

**HOST.** Es en esencia una computadora. Típicamente un mainframe que comparte sus recursos simultáneamente con usuarios múltiples.

**INTEGRIDAD DE DATOS.** Es la corrección, validación y consistencia de datos dentro de una base de datos, consistente en reglas que gobiernan las actualizaciones y distribución. La integridad de datos asegura que los datos al principio y al final de cada transacción individual sea consistente. Esto es un pre-requisito para un proceso distribuido.

**INTERACTIVO.** Se refiere a un diálogo continuo entre el usuario y la aplicación. Dos programas también pueden interactuar si cooperan en una relación cliente-servidor.

**INTEROPERABILIDAD.** Es la habilidad de que dos o mas sistemas de software heterogéneos puedan trabajar juntos basados en el uso de un protocolo común,

**LAN (Local Area Network).** Es el número de piezas de equipo, (típicamente workstations o PC'y sus periféricos) que han sido unidos y que intercambian información y parte de sus recursos. En contraste con las WAN, las LAN son de área reducida

**MAINFRAME.** Es una computadora de gran tamaño, que posee las características de multiusuario expansivo, típicamente con arquitectura de IBM S/700 o S/390. En ambiente cliente-servidor, un mainframe juega el papel de 'super server' el cual es capaz de manejar datos en masa, o un alto número de funciones.

**MEMORY.** Parámetro estático del server de SYBASE que indica el espacio de memoria RAM que utiliza el proceso del servidor.

**MINICOMPUTADORA.** Es un término usado para describir a las computadoras que se encuentran entre un mainframe y workstations.

**NODO.** Es una unidad funcional (de hardware y/o software) dentro de una red. Los nodos de hardware son a menudo computadoras, pero también pueden ser dispositivos, tales como controladores, impresoras u otros equipos periféricos. Un ejemplo de un nodo de software en una base de datos.



**OLTP (Online Transaction Processing).** Es una manera de manejar recursos y tareas los cuales permiten a múltiples usuarios comunicarse con varios servers simultáneamente. Está diseñado para llevar a cabo la optimización de la velocidad, confiabilidad e integridad.

**OPEN SQL SERVER.** Es un producto que proporciona el manejo de SQL Standard para acceder a una gran variedad de manejadores heterogéneos de SQL (IBM, ORACLE, SYBASE, INGRES, RDB). De esta forma, se permite que cualquiera de estos manejadores actúen como servidores.

**PLATAFORMA.** Es un término genérico utilizado para describir el hardware del sistema operativo en la cual una pieza dada de software corre.

**PORTABILIDAD.** Un término utilizado ambiguamente. La portabilidad verdadera describe la característica de los programas de software y datos que se permiten sean migrados a diferentes plataformas.

**PRESENTACION DISTRIBUIDA.** Es una forma de procesamiento cooperativo en el cual la función de manejo de la presentación de una aplicación es implementada en dos o mas componentes de software que interactúan a través de la red. Está caracterizado por aplicaciones de un mainframe. En este caso, la lógica del manejo de las presentaciones existe en ambos ambientes (pc y mainframe), en contraste con la presentación remota.

**PROCESAMIENTO COOPERATIVO.** Es un modelo de ambiente de cómputo que consiste de dos o más elementos de software que interactúan para realizar tareas específicas. Estos elementos pueden residir en una o más computadoras. Existen cinco formas diferentes de procesamiento cooperativo : presentación distribuida, presentación remota, aplicación distribuida, base de datos remotas y base de datos distribuidas.

**PROCESAMIENTO DISTRIBUIDO.** Una forma de procesamiento cooperativo en el cual los recursos de son compartidos a través de la red de componentes de hardware discretos de manera que es transparente al usuario. Generalmente, el procesamiento distribuido se refiere a una situación en donde los componentes individuales dentro de una aplicación están distribuidos o corren en una o más computadoras.

**PROPIETARIO.** Un ambiente computacional se dice que es propietario si restringe al usuario de utilizar productos del mismo proveedor.

**PROTOCOLO.** Una colección predefinida de reglas procedurales utilizadas para configurar y manejar la comunicación entre dispositivos o programas discretos.

**RED.** Es una serie de unidades funcionales (nodos de hardware y/o software) y mecanismos de comunicación conectados entre sí. El objeto de unirlos es permitir el intercambio de información y de recursos.

**RIGTHSIZING.** El proceso de ajustar el tamaño de un hardware de una empresa y sistemas de software a sus requerimientos de negocios. Generalmente, los rightsizing incluyen una reducción en el tamaño del hardware en particular. Esto es también conocido como downsizing.

**RPC (Remote Procedure Call).** Son medios de comunicación a nivel aplicación, las cuales permiten a los programas acceder a subrutinas o procedimientos situados en diferentes plataformas y de esta manera invocar su funcionalidad. Estas son vitales para aplicaciones cliente servidor, en las cuales la funcionalidad de aplicación será distribuida.

**RULE.** Es un modelo de sintaxis para cada uno de los campos.

**SERVER.** Es cualquier elemento dentro del ambiente computacional que provee servicios a múltiples clientes.

**SISTEMAS ABIERTOS.** Es un término utilizado en una amplia variedad de conceptos. Un sistema puede ser llamado abierto si cumple todas o cualquiera de las siguientes características:

- Es capaz de aceptar estándares.
- Permite la interoperabilidad con productos de terceras partes.
- Proporciona interfaces cuya documentación es publicada.
- Permite la independencia del usuario (portabilidad).

**SISTEMAS OPERATIVOS.** Es un sistema de software que toma y maneja recursos de hardware disponibles para los usuarios y programas. Sus servicios incluyen programación, control de entrada-salida, almacenamiento y manejo de datos.

**SQL (Structured Query Language).** Es una interfaz programada para que los sistemas de manejadores de bases de datos relacionales ofrezcan facilidades para definir, manipular y controlar los datos. Aunque su funcionalidad todavía no ha sido finalizada, SQL se está volviendo un estándar para el acceso a la bases de datos.

**STORED PROCEDURE.** Es un procedimiento de SQL compilado y almacenado en la base de datos.

**TRANSACCION.** Una unidad lógica de trabajo que modifica una o más bases de datos sin perder la consistencia de los datos.

**TRIGGER.** Es un stored procedure que se activa automáticamente al realizar una transacción básica (insert, delete, update) sobre una tabla.

**TWO PHASE COMMIT.** La parte de manejo de datos que garantiza la integridad de los datos en un ambiente de multibases de datos. El two phase commit es un método de coordinar las actividades de transacción asegurando que la actualización de cada base de datos sea completada o deshechada.

**UPGRADING.** Es el reemplazo de un sistema existente con uno mejorado.

**UPSIZING.** Es el proceso de reemplazar un sistema existente con uno mas grande.

**WAN (Wide Area Network).** Es un número de piezas de equipo de hardware y sus canales de comunicación que utilizan datos públicos que se encuentran a grandes distancias.

**WORKSTATION.** Es una computadora de alta categoría entre las personales. Originalmente tales máquinas corren aplicaciones especiales tales como el CAD/CAM, pero debido a que ahora se ha mejorado su precio se utilizan para múltiples aplicaciones. Puede o no ser un sistema de un solo usuario.