

16

2 Ejem.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA INTEGRAL DE RECURSOS HUMANOS
SINRECH**

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N :
XOCHITL CHAVARRIA CRUZ
JORGE ARREOLA ORNELAS
ISRAEL LOPEZ AVILA

ASESOR: ING. ALBERTO TEMPLOS CARBAJAL



C. UNIVERSITARIA

1994

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis 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.

EN ULTIMA INSTANCIA, LA
SOLUCION DE LOS PROBLEMAS
NO CONSISTE EN HACER,
NI EN DEJAR DE HACER,
SINO EN COMPRENDER;
PORQUE DONDE HAY
VERDADERA COMPRESION,
NO HAY PROBLEMAS.

ANTHONY DE MELLO.

AGRADECIMIENTO.

Los que ahora suscribimos el siguiente trabajo, queremos dar las gracias a todas aquellas personas que participaron directa o indirectamente en la elaboración del mismo, y que sin su colaboración no hubiera sido posible llevarlo a su culminación.

Al Ing. Alberto Templos Carbajal, no solamente por su dirección y asesoramiento en la elaboración de esta Tesis, lo que nos llena de orgullo, sino también por considerarlo una de las personas más importantes en el desarrollo y culminación de nuestras carreras, por sus constantes consejos y el estar pendiente y dispuesto a brindarnos su ayuda cuando más la requerimos, por eso y más muchas gracias.

Al Ing. Sergio González Calderón por brindarnos todo su conocimiento, apoyo y confianza, por estar con nosotros en todos los momentos difíciles, por ser una persona excepcional como amigo y como jefe.

Al Ing. Gerardo Acosta Mancilla, por ayudarnos en todo lo que necesitamos, por brindarnos sus consejos ya que gracias a ellos hemos sabido decidir con mayor facilidad los problemas a los que nos hemos enfrentado, gracias por brindarnos tu amistad.

Al Ing. E. Salvador González Leyva, por el apoyo que nos brindo, mientras estuvimos a su cargo y por el impulso recibido para adentrarnos prácticamente en la carrera.

A toda la gente del CAPFCE que nos ha apoyado y nos sigue apoyando, de manera especial a la Srita. Ma. Cristina Domínguez Rodríguez, Subgerente de Nómina, por su apoyo decidido para la realización del proyecto. Al Lic. Roger Milton Rubio Paniagua, Gerente de Proceso de Datos, ya que sin su apoyo y decisión para la continuación del proyecto este no hubiera llegado a su culminación. Al Sr. Germán Hájera del Valle, Subgerente de producción, por los consejos brindados a lo largo del desarrollo del proyecto. Al Sr. Alejandro Pacheco por su colaboración y participación en el desarrollo del proyecto, también por el apoyo que siempre nos ha brindado.

A todos nuestros amigos de la Facultad de Ingeniería, por estar a nuestro lado a lo largo de la carrera, y por todos los momentos agradables que hemos pasado juntos.

Finalmente queremos agradecer a nuestra máxima casa de estudios, la Universidad Nacional Autónoma de México, por todos los conocimientos recibidos en sus aulas y por brindarnos la oportunidad de realizarnos como profesionistas útiles para nuestra sociedad. Gracias a todos nuestros profesores que de manera altruista y profesional han compartido con nosotros toda su experiencia y conocimientos.

A todos ellos muchas gracias.

ATENTAMENTE

Xóchitl Chavarría Cruz
Jorge Arreola Ornelas
Israel López Avila

A Dios, por darme el aliento que necesitaba a lo largo de mi vida, por darme la fe y el animo para continuar con esta meta ahora lograda.

A mi mamá, Catalina Cruz, que fue la que me dio la vida y a quien debo todo lo que soy, porque sin su apoyo, comprensión y consejos no hubiera podido concluir todo.

Gracias por ser mi amiga, por la confianza y alegría con la que me has hecho crecer, por tu forma de ser y sobre todo, gracias por estar siempre a mi lado. Esto es para ti mamá. Te amo.

A mi papá, Camilo Chavarria, por enseñarme el buen camino, por decirme como comportarme, por creer en mi y principalmente por apoyarme en todo, quiero que este logro también sea tuyo. Te amo papá.

A mis hermanos, Claudia y Rodrigo, por ser mis principales amigos, por ser como son, por hacerme ver las cosas de la manera más sencilla, por contar siempre con su cariño, por compartir los momentos felices y también por los momentos amargos que son los que nos han ayudado a unirnos cada día más, los quiero mucho.

A mi abuelita
Antonia: por los consejos dados, por la sinceridad de sus palabras, por ayudarme en todo, con mucho cariño. Gracias.

A mi abuelita
Delfina: por el apoyo brindado y por todos los momentos que hemos compartido juntas

A mis tios, por creer en mi, por darme el aliento necesario para continuar y sobre todo por los consejos recibidos. Gracias por todo lo bueno que me han brindado.

A Israel. Gracias por tu apoyo y comprensión, juntos hemos luchado para lograr la culminación de nuestra carrera, juntos compartiremos este triunfo, por eso y más, te dedico en forma muy especial esta tesis, y te doy gracias por todo el impulso que me brindaste en los momentos más difíciles de mi vida y sobre todo por el amor que me has demostrado. !!TE AMO!!

A mis primos, por el hecho de estar a mi lado y a pesar de todo, seguir unidos como hasta hoy, por compartir conmigo todos los momentos agradables, y principalmente por su cariño incondicional, gracias.

A Jorge, por ser compañero de Tesis y principalmente por ser un amigo incondicional que me ha demostrado la confianza que me tiene, también agradezco el apoyo brindado por la Familia Arreola Ornelas. Muchas gracias

A mis amigos, Víctor, América, Dolores, Guadalupe y Susana, por estar siempre conmigo, por ayudarme en todo lo que necesite, por su apoyo y comprensión, gracias.

A la Familia López Avila, por todo el apoyo que me han brindado, por la confianza que han depositado en mí, y principalmente por aceptarme como un miembro más de su familia. A todos Gracias.

A todas aquellas personas que me han apoyado a lo largo de mi estancia en la facultad, que por el momento sería muy largo mencionar, sin embargo también son importantes; un sincero agradecimiento.

SINCERAMENTE

Xóchitl Chavarría Cruz

Resulta realmente difícil poder agradecer, a todos y cada una de las personas que hacen posible hoy en día la elaboración de este trabajo. No solamente a las personas que me apoyaron a nivel de licenciatura y que por supuesto me siguen apoyando, sino también a aquellas personas que de una u otra forma me han ayudado para lograr mi formación desde mis profesores de primaria,, pasando por los de secundaria y bachillerato hasta llegar finalmente a los de licenciatura.

A todos mis amigos parientes y familiares, y a todas aquellas personas que siempre han sabido estar conmigo en los momentos agradables y difíciles de mi vida. A todos ellos un agradecimiento con todo el corazón.

Es por esta razón que quiero decirles que, este triunfo es tan suyo como mío, por lo cual no quisiera realizar menciones especiales a ninguna persona, ya que de hacerlo así incurriría en la grosería de omitir nombres. Sin embargo y de acuerdo a todo lo anterior yo se que ellos saben bien a quien me refiero, por esta razón muchas gracias.

Finalmente para terminar, y sin querer contradecir lo anterior, quisiera realizar algunas consideraciones por separado, por la importancia que estas tienen para mí. Yo se bien que me sabrán entender.

A mí Dios y a mi muy querida Virgencita de Guadalupe por recibir de ellos día a día las más grandes bendiciones.

A mi padre Sr. Gabriel Arreola Peralta por mostrarme siempre el camino a seguir a través de su ejemplo y saber impulsarme siempre en los momentos difíciles de mi vida.

A mi madre Sra. Reynalda Ornelas Alfaro, por sus constantes desvelos, cariño, dedicación y paciencia que siempre me ha tenido. Por saber apoyarme e impulsarme en todo momento para lograr ser mejor cada día.

A mis hermanos Martha, Verónica, Víctor y Héctor, así como a mi abuelita Soledad Alfaro, por que gracias a ellos y a mis padres he entendido lo importante que es contar con una bonita familia. Gracias por ser como son y por sus constantes apoyos y consejos.

A mis amigos y hermanos del alma, compañeros de Tesis Xóchitl e Israel, así como a sus respectivas familias por el constante apoyo recibido.

ATENTAMENTE

Jorge Arreola Ornelas.

Doy gracias a Dios por permitirme llegar a este momento, por darme la fe y la fuerza que necesito.

A mis Padres JUANITA y JOEL.
Porque gracias a su esfuerzo y dedicación es que he logrado siempre seguir adelante, por su apoyo, comprensión, cariño y amistad que siempre me han brindado.
Porque han compartido en todo momento conmigo los momentos felices pero mayormente los difíciles.
Porque con su ejemplo me han enseñado siempre el camino a seguir.
No podría decir todo lo que quisiera, solo quiero recordarles que este logro también es suyo y que los amo con todo mi ser.
Gracias por toda una vida de felicidad.

A mis hermanas LOLITA y VERO.
Porque han sido un ejemplo durante toda mi vida.
Porque me han enseñado a luchar por lo que quiero.
Por su cariño brindado en todo momento.
Por creer firmemente en mí.
Por oír con paciencia mis problemas y aconsejarme en su solución, por eso y muchas cosas más, las amo.

A mis hermanos JOEL Y TONHATIUH.
Porque han sido un apoyo importante en todo momento.
Por los momentos felices que hemos vivido.
Por todos los problemas que hemos compartido.
Por su cariño incondicional.
Porque he aprendido muchas cosas de ustedes, quiero que sepan que los quiero mucho.

A mi abuelita TRINIDAD AVILA"
como un homenaje póstumo a su memoria,
por todo el amor que me dio, con todo mi cariño.

A mi abuelita JULIA GRANADOS.
Por los buenos momentos vividos
con mucho cariño y afecto.

A mi tío RAUL.
Por ser más que un tío un amigo.
Por el cariño que siempre me has demostrado, quiero compartir contigo este logro.
te quiero mucho.

A mis tíos LEOBARDO y GUADALUPE.
A mis primos MEMO, ELI y LINO
por los grandes momentos que hemos pasado juntos, con todo cariño.

A KOCHITL.

Porque juntos hemos caminado una buena parte del camino.
Porque juntos alcanzamos las metas propuestas.
Porque juntos seguiremos luchando para conseguir mayores logros.
Porque juntos hemos compartido ya momentos felices y amargos.
Por el gran amor que me has demostrado.
Por tantas y tantas cosas, quiero que sepas que: ¡te amo!

A JORGE.

Por ser un gran amigo al que quiero y admiro,
por compartir juntos el reto de este trabajo,
porque siempre me ha apoyado, quiero agradecer
también el apoyo que siempre me ha brindado
su Familia, muchas gracias.

A la Familia CHAVARRIA CRUZ.

Por el apoyo brindado en los
últimos años, por la gran confianza
que me han tenido, con cariño,
muchas Gracias.

A mis grandes amigos de la adolescencia:

FELIPE, SERGIO e ISAIAS.

Porque siempre me han ayudado en momentos difíciles.
Porque hasta hoy tengo la dicha de gozar de su amistad.
Por brindarme siempre su apoyo, gracias.

No puedo dejar de mencionar a otros grandes amigos:

MOISES, JUAN CARLOS, ALBERTO, OMAR, JOSE LUIS Y JESUS G.
Porque a lo largo de nuestra estancia en la facultad
compartimos muchos logros, aventuras y fracasos.
Porque han sido un apoyo en todo momento.
Gracias a todos ellos.

Finalmente un agradecimiento sincero a los
familiares y amigos que no he mencionado
no por falta de ganas, sino por falta de espacio.
Los aprecio mucho, gracias por todo.

SINCERAMENTE

Israel López Avila

CONTENIDO

INTRODUCCION.....	iii
I.- GENERALIDADES.....	1
I.1. ANTECEDENTES.....	1
I.1.1. VENTAJAS DE LOS SISTEMAS ANTERIORES.....	1
I.1.2. DESVENTAJAS DE LOS SISTEMAS ANTERIORES.....	2
I.2. DEFINICION DEL PROBLEMA.....	4
I.3. ALTERNATIVAS DE SOLUCION.....	6
II.- ANALISIS.....	9
II.1. PLANEACION DEL DESARROLLO DEL SISTEMA INTEGRAL DE RECURSOS HUMANOS.....	11
II.2. SISTEMA DE TRANSACCIONES.....	14
II.3. SISTEMA DE CONTROL DE ASISTENCIA.....	21
II.4. SISTEMA DE PERCEPCIONES.....	26
II.5. SISTEMA DE DEDUCCIONES.....	37
II.6. SISTEMA DE FONDO DE AHORRO.....	44
II.7. DICCIONARIO DE DATOS.....	48
II.8. NIVELES Y CONSECUENCIAS DE LOS CAMBIOS QUE IMPLICARA EL NUEVO SISTEMA.....	55
III.- DISEÑO DEL ESQUEMA DE SOPORTE DE LAS APLICACIONES (BACK-END).....	57
III.1. DISEÑO LOGICO.....	57
III.1.1. DIAGRAMA ENTIDAD RELACION.....	57
III.1.2. DIAGRAMA DE TABLAS.....	61
III.1.3. DETERMINACION DE INDICES.....	63
III.1.4. NORMALIZACION.....	66
III.1.5. DESNORMALIZACION.....	73

III.1.6. INTEGRIDAD DE LOS DATOS.....	76
III.1.7. DICCIONARIO DE DATOS.....	90
III.2. DISEÑO FÍSICO.....	128
III.2.1. TAMAÑOS DE LAS BASES DE DATOS.....	128
III.2.2. ARQUITECTURA.....	131
III.2.2.1. CLIENTE - SERVIDOR.....	131
III.2.3. SEGURIDAD DEL SISTEMA.....	133
IV.- DISEÑO DE LAS APLICACIONES DE INTERFAZ CON EL USUARIO (FRONT-END).....	135
IV.1. DISEÑO DE FORMAS.....	135
IV.2. ALGORITMOS DE PROCEDIMIENTOS GENERALES.....	142
IV.3. DISEÑO DE REPORTES.....	145
IV.3.1. GENERADOR DE REPORTES.....	145
IV.3.2. REPORTES OPEN - CLIENT.....	146
V.- IMPLANTACION Y PRUEBAS.....	147
V.1. DISEÑO DE PRUEBAS.....	150
V.2. PRUEBAS EN PARALELO.....	153
VI.- CONCLUSIONES.....	155
BIBLIOGRAFIA.....	160
APENDICES.....	162
APENDICE A. TRIGGER.....	162
APENDICE B. VISTAS.....	193
APENDICE C. REGLAS.....	198
DEFAULTS.....	206
GLOSARIO.....	207

INTRODUCCION

ANTECEDENTES HISTORICOS.

El Comité Administrador del Programa Federal de Construcción de Escuelas, C.A.P.F.C.E., es el organismo descentralizado encargado de organizar, dirigir y llevar a cabo los programas federales de construcción, equipamiento y habilitación de instalaciones docentes.

Este organismo fue creado por decreto presidencial el 10 de abril de 1944 durante la administración del Presidente Manuel Avila Camacho, siendo entonces Secretario de Educación Pública el Doctor Jaime Torres Bodet. Es en este momento cuando a través del Comité se inicia la urgente tarea de edificar inicialmente instalaciones docentes destinadas al sistema de instrucción primaria en el medio urbano de la República, ya que hasta antes de este decreto, las escuelas de enseñanza básica del sistema federal (insuficientes e inadecuadas) venían funcionando en locales que distaban mucho de llenar las necesidades docentes, situación que fue agravándose con el transcurso del tiempo al aumentar considerablemente la demanda educativa derivada de la explosión demográfica.

Es así como después de superar numerosos problemas de carácter técnico, sociales y económicos, que en un principio tuvo que enfrentar, hoy en día el organismo ha alcanzado niveles de trabajo aceptables y la realización de proyectos adecuados, importantes para la vida educativa del país en los últimos 50 años.

ESTRUCTURA ORGANIZACIONAL.

La dirección del organismo está a cargo de un Director General del que dependen las siguientes áreas: Administrativa, Técnica, Auditoría Interna y la Dirección de Organización y Métodos.

El área técnica esta integrada por las siguientes gerencias: Planeación y Programación, Proyectos, Supervisión y Control de obras, Adquisiciones, Obras especiales, Ingeniería de costos y Proceso de datos.

Del área administrativa dependen las siguientes gerencias: Recursos Humanos, Tesorería, Contabilidad, Presupuestos y Asuntos Jurídicos.

Las Jefaturas de Zona son las encargadas de construir, equipar y entregar las escuelas para su operación, se encuentran en las 31 capitales estatales, y dependen funcionalmente de la Dirección General, manteniendo al mismo tiempo una estrecha relación con las diferentes gerencias de donde reciben los lineamientos generales y apoyos diversos para el cumplimiento de sus respectivos programas.

Por otra parte para el funcionamiento de cualquier empresa debe considerarse el aspecto relacionado con los Recursos Humanos y los diferentes sistemas de control que se puedan implementar para llevar a cabo la administración de los mismos, por lo que a continuación se explica la importancia de la gerencia de Recursos Humanos dentro de este organismo.

RECURSOS HUMANOS.

La Gerencia de Recursos Humanos se interrelaciona con todas las unidades administrativas del organismo (Gerencias y Jefaturas de Zona).

Esta Gerencia se desagrega orgánicamente de la siguiente manera:

- Subgerencia de personal.- A su vez esta subgerencia se

integra de las siguientes áreas de responsabilidad:

- > Selección y contratación de personal
 - > Prestaciones internas
 - > Prestaciones externas
 - > Licencias y períodos vacacionales
 - > Identificación de personal
 - > Capacitación y servicio social
 - > Relaciones laborales
 - > Fondo de ahorro
 - > Comisión nacional mixta de escalafón
 - > Comisión nacional mixta de higiene y seguridad
- Subgerencia de nómina.- Esta subgerencia se integra con las siguientes áreas:
- > Operación y control de nóminas
 - > Registro y control de personal
 - > Codificación y captura
 - > Hojas de servicios y constancias
 - > Archivo de personal
- Coordinación de presupuesto, sueldo y salarios.- Esta coordinación se integra con las siguientes áreas:
- > Presupuesto
 - > Tabuladores
 - > Sistema integral de información
 - > Control de personal empleado y plazas ocupadas

DESCRIPCION GLOBAL DEL SISTEMA.

Es importante mencionar de forma breve, los aspectos básicos que se pretenden lograr en la realización de cada capítulo de este documento.

En el capítulo I se muestran algunos antecedentes que se tienen sobre los sistemas actuales, en donde se ve de manera rápida las ventajas y desventajas de dichos sistemas, así como las necesidades que dan origen a la creación de un nuevo sistema que

abrará de forma adecuada los requerimientos presentes del Comité, es decir, se definirá el problema y por último se verán las diferentes alternativas de solución que se tienen para la realización del proyecto, indicando aquella que se habrá de adoptar para dar respuesta de una manera más adecuada a la problemática definida.

En los capítulos II al V, se realiza un análisis de los puntos a considerar en la realización del sistema, entre los que tenemos: Planeación, Diseño, Desarrollo, Implementación, Pruebas, Corrección de Errores, Puesta en Marcha y Mantenimiento. En el capítulo II, se hace un desglose de los subsistemas en que se ha dividido el sistema para su elaboración, los cuales son los siguientes:

- Sistema de Transacciones.
- Sistema de Control de Asistencia.
- Sistema de Percepciones.
- Sistema de Deducciones.
- Sistema de Fondo de Ahorro.

se incluye además un diccionario de datos inicial, así como una explicación de los niveles y consecuencias de los cambios que implicará el nuevo sistema.

En el capítulo III, se verá el diseño del sistema, tanto en su parte lógica como en la física, el diseño lógico trata el diagrama entidad-relación, el diagrama de tablas y vistas, la determinación de índices, la normalización, desnormalización, integridad de los datos y la definición del diccionario de datos. Por lo que se refiere al diseño físico, se toma en cuenta el tamaño de las bases de datos, así como la arquitectura a usar; que para nuestro caso es del tipo centralizada y la seguridad del sistema.

Por lo que corresponde al capítulo IV, se refiere a un diseño de las aplicaciones de interfaz con el usuario (FRONT-END). Para lo cual se explica el diseño de elaboración de las diferentes formas de captura, así como de los algoritmos de los procedimientos generales, y de los diversos reportes a generar.

En el capítulo V, se ve lo relacionado a la parte de implementación y pruebas. Con una explicación de la forma en que se realizan estas pruebas, así como las pruebas en paralelo con respecto al sistema anterior, para una correcta satisfacción y seguridad de los resultados obtenidos.

Por último el capítulo VI, contiene las conclusiones finales a las que se llegaron, después de la elaboración del sistema, conteniendo además una bibliografía completa que se utilizó en la elaboración de este documento, así como un glosario.

Es importante resaltar que el presente trabajo trata de integrar los sistemas de control ya existentes, además de agregar otros que se utilizan para la administración del área de Recursos Humanos, en uno solo denominado, *Sistema Integral de Recursos Humanos (SINRECH)*.

I - GENERALIDADES

I.1. ANTECEDENTES

El sistema que se pretende desarrollar sustituirá al que actualmente se encuentra operando en el Comité, el cual es un sistema híbrido que corresponde a una primera versión incompleta de SINRECH, que se desarrolló con una herramienta llamada RDB, la cual es un manejador de bases de datos para VAX, así como a un sistema de nómina llamado SYSNOM (Sistema de Nómina). Este sistema está dividido en tres partes:

- Registro en Nómina
- Control de Asistencia
- Pagos

Las partes Registro en Nómina y Control de Asistencia se trabajan en RDB y pascal, en una máquina VAX 4000, a través de SINRECH primera versión. La parte de pagos se trabaja en una computadora VAX 780, a través de SYSNOM.

I.1.1. VENTAJAS DE LOS SISTEMAS ANTERIORES.

La primera versión del sistema SINRECH, tenía la virtud de proporcionar un ambiente amigable para el usuario; ya que este contaba para su desarrollo con herramientas de software fabricadas en el Comité creando estándares y propiciando así un carácter de uniformidad, un fácil mantenimiento de la información y perspectivas para su crecimiento.

En cuanto al sistema de nómina SYSNOM, es un sistema que venía integrado con el equipo de cómputo PDP-11 el cual se encontró en funcionamiento en el Comité hasta 1990; se trata de un sistema comercial que está diseñado con características generales que

permiten su fácil adaptabilidad a algunos de los requerimientos específicos de cualquier empresa.

Con la implantación del nuevo sistema se desea continuar manteniendo las características anteriores, mejorando una gran cantidad de aspectos que se significaban por representar desventajas en los dos sistemas anteriores ya antes mencionados. Estas desventajas de ambos sistemas se habrán de explicar en el siguiente punto a tocar, enmarcando aquellas que se pretenden mejorar mediante la implantación del nuevo sistema.

I.1.2.- DESVENTAJAS DE LOS SISTEMAS ANTERIORES.

Uno de los puntos más importantes en todo sistema, tiene que estar relacionado indiscutiblemente, con el tiempo de proceso y administración de los datos. Y también proporcionar un mínimo de esfuerzo a todas aquellas personas que hagan uso del sistema en cuestión. Como se puede analizar, ambos puntos en muchas de las ocasiones no son tan fáciles de lograr, sin embargo, se pretende proporcionar las herramientas, tanto de software como de hardware adecuadas para reducir al máximo estos dos aspectos.

Pero no solamente esos puntos son los únicos que nos interesan, ya que existe una gran variedad de problemas con los sistemas anteriores, entre los cuales podemos mencionar los siguientes:

a) Desempeño y Consistencia.- Las partes Registro de Nómina y Control de Asistencia, se trabajan en RDB y pascal, en la máquina VAX 4000. Los problemas que tienen son en el diseño, desempeño (performance) y en la consistencia del mismo. La parte de pagos que se trabaja en la máquina VAX 780 tiene más problemas, ya que están programadas en BASIC Y COBOL, lo que genera conflictos en el desempeño y en su imposibilidad de crecimiento.

b) Redundancia y no estandarización de la información.- Al tener dos sistemas interactuando para el manejo de la información, se tiene el problema de la estandarización, en uno (SINRECH) se

procesan las Altas, Bajas y Cambios (ABC's), y el Control de Asistencia, y en el otro (SYSNOM) se procesa la nómina. El primer sistema obtiene toda la información necesaria para el funcionamiento del segundo sistema, dejándola en archivos secuenciales o indexados con las estructuras que requiere SYSNOM. Propiciando la redundancia de información, ya que la información de los archivos es la misma que contiene la base de datos, pero como no existe una estandarización de información es necesaria la creación de estos archivos.

c) Independencia de los sistemas y programas.- Del sistema SINRECH, quincenalmente se crea un archivo que contiene la información más importante de la base de datos, al cual se le llama fotografía. En base a este archivo funcionan otros programas o pequeños sistemas de SINRECH, tales como: SAR, ISSSTE, AVISOS DE ABC's, etc., siendo todos estos parcial o totalmente independientes del sistema en general.

d) Manipulación de datos en forma manual.- Actualmente existe una gran dependencia de la Gerencia de Procesos de Datos, en el manejo de los datos del sistema y en el proceso de información que el Comité envía a otras instituciones; es decir, se requiere de gente familiarizada con la manipulación directa de datos en la computadora, para atender un sin número de peticiones no formalizadas de la Gerencia de Recursos Humanos.

e) Inexistencia del Diccionario de datos.- No existe un diccionario de datos para SYSNOM, en el cual se pudiera hacer referencia a campos con el mismo nombre, o que por lo menos el mismo tipo de información existiera en los mismos archivos. En el caso de SINRECH, desde la versión 1.0, ya se contaba con un diccionario de datos.

f) Escasa documentación de los sistemas y falta de manuales.- Existe poca documentación tanto de diseño como del desarrollo del sistema SINRECH versión 1.0, además no se desarrollaron los manuales necesarios para la operación y mantenimiento del mismo,

así mismo las partes agregadas al sistema tampoco han sido documentadas.

g) **Falta de ayuda, para cada uno de los diferentes procesos de los sistemas.**- Para los sistemas SINRECH y SYSNOM no existen ayudas adecuadas para el uso interactivo en las pantallas, propiciando de esta manera que el usuario no tenga una buena orientación con el manejo de la información, aún cuando las pantallas no muestran un grado complejo para su manejo.

h) **Insuficientes sistemas de consultas.**- Esto se refiere a que, para tener acceso a la información de manera ordenada sólo se puede realizar en papel y únicamente en formatos ya establecidos, con lo cual se pierde la versatilidad de las consultas. Es decir, 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 verídica.

Cada uno de los puntos mencionados, pretenden ser mejorados, en base a la experiencia obtenida de los sistemas anteriores, reduciendo el tiempo y esfuerzo ocupados en los mismos y propiciando una mayor integridad y seguridad en el manejo de la información.

I.2 DEFINICIÓN DEL PROBLEMA

Se desea actualizar el sistema llamado " Sistema Integral de Recursos Humanos, SINRECH ", con la finalidad de mejorarlo y complementario, incrementando la flexibilidad en el manejo del mismo.

Elaborar un Sistema Integral de Recursos Humanos que sustituya al sistema actual y elimine la manipulación de datos en forma manual. Este sistema debe ser desarrollado con el objetivo de una independencia en el uso de la Gerencia de Proceso de Datos, al menos que se requiera mantenimiento o generación de nuevos módulos, permitiendo además el uso interactivo de varios usuarios a la vez, así como manejar transacciones para la actualización en forma

sencilla, estableciendo diferentes procedimientos de manipulación de datos entre los empleados de Mandos Medios y Superiores (MMYS) y los empleados de Base y Confianza. El sistema será centralizado y contendrá información a nivel Nacional, teniendo como objetivos los siguientes:

- 1.- El sistema cubrirá varios aspectos que intervienen en la elaboración del recibo de pago del trabajador. Además, sacará diversos reportes, preparando información para ser enviada a diferentes organismos tales como: ISSSTE, FOVISSSTE, SAR (Banco), Aseguradoras, etc.
- 2.- Una vez introducido el esquema relacional, y aprovechando la capacidad de consistencia que se ofrece por parte del manejador, se podrá permitir que el usuario manipule todos los programas que se hayan establecido para el uso del sistema.
- 3.- El manejador de bases de Datos, debe permitir de una manera eficiente el proceso multiusuario.
- 4.- A diferencia de la pasada versión de SINRECH, se contempla el manejo de transacciones que engloban el uso de más de una tabla.
- 5.- Se utiliza una manipulación de empleados en lo que respecta a su tipo de puesto. se debe tener seguridad en la manipulación de datos; separándose a los empleados de Mandos Medios y Superiores (MMYS) y a los de Base y Confianza.
- 6.- El Comité Administrador del Programa Federal de Construcción de Escuelas (C.A.P.C.E.) es una Institución Gubernamental a Nivel Nacional, por lo que sus Jefaturas de Zona (sucursales en los Estados) manipulan su información en las Oficinas Centrales.
- 7.- Se deberá evaluar al Sistema en funcionamiento; es decir, se pondrá en marcha en forma paralela con el Sistema actual, hasta que se compruebe que los resultados son los esperados.

I.3. ALTERNATIVAS DE SOLUCIÓN.

En base a los problemas planteados se requiere cubrir las necesidades informáticas del Comité de tal forma que la solución tenga las siguientes características:

- Integridad de los datos
- Interfase usuario-máquina amigable
- Tiempos de respuesta eficientes
- Herramientas de apoyo que permitan un desarrollo rápido y eficiente.

Se consideran las siguientes alternativas que son las que pueden cumplir en un momento dado las características anteriores:

1.- Adquisición de un sistema ya existente en el mercado.

En este caso se tiene la problemática de que los sistemas existentes en el mercado son planeados en forma general, por lo que no se cubrirían todas las necesidades particulares del Comité, por lo que se considera que esta alternativa no es la factible en la solución de nuestro problema, ya que no abarcaría todos los objetivos planteados al definirlo.

2.- Integración de los sistemas ya existentes.

Representa una alternativa que no es viable por los problemas ya descritos anteriormente, además, de que su adecuación requeriría de una fuerte inversión de tiempo que no contribuye a la corrección del sistema realmente, ya que se trata de integrar dos sistemas que fueron planeados en diferente forma y persiguiendo distintos objetivos.

3.- Desarrollo de una nueva versión del sistema SINRECH.

Esta alternativa requiere retomar el diseño de la versión 1.0 del sistema SINRECH, ajustarlo y en algunas partes hacer un diseño nuevo de ella, así como incrementar el campo de acción del mismo.

Esta opción representaría en primera instancia una inversión mayor en tiempo, pero considerando las ventajas que nos proporcionará el nuevo sistema y las desventajas mencionadas en los puntos anteriores es muy probable que esta opción se desarrolle en

menos tiempo.

Se plantea la reestructuración del diseño y finalización del sistema, debido a lo siguiente:

El sistema anterior de SINRECH, no contempla una actualización sencilla para los registros que se manipulan, por lo tanto para llevar a cabo una transacción, el usuario debe actualizar individualmente, esto es, a través de varios programas, cada una de las tablas para que dicha transacción se lleve a cabo.

Con el equipo de hardware actual, VAX 4000, el manejador de base de datos mejora en su tiempo de respuesta, sin embargo, no se cuenta con el software para continuar el desarrollo en dicho equipo, teniendo solo la parte de ejecución.

Actualmente se cuenta con una herramienta de desarrollo que permite hacer un diseño bajo el esquema relacional y ofrece mayores ventajas, para el aprovechamiento de nuestra base de datos.

Entre las ventajas que presenta se tienen:

- Una mayor velocidad de respuesta, tanto en proceso como en consultas.
- Facilidad en el desarrollo, desde la programación hasta la obtención de reportes.
- Disponibilidad ya instalada dentro del Hardware del Software con el que se desarrollará el sistema.
- Incorporación de una nueva tecnología al desarrollo de sistemas informáticos dentro del Comité.

Los recursos con los que se cuentan para el desarrollo del sistema, son los siguientes:

- Sistema Operativo
 - . VAX/VMS, versión 5.4-3.
- Compilador
 - . C.
- Manejador de Bases de Datos
 - . SYBASE.

SINRECH

- Herramientas para la generación de aplicaciones de SYBASE.
 - . DATA WORK BENCH.
 - . APLICATION PRODUCTIVITY TOOLS.
 - . DB LIBRARY.
- Hardware
 - . Minicomputadores VAX 11/780 y VAX 4000
 - . Terminales Wyse y H.P.
 - . Unidades de cinta Digital.
 - . Impresoras de alta velocidad Digital y H.P. LASERJET III y IIIp .
 - . Red Enthernet con PC's.

De acuerdo a los requerimientos antes vistos, consideramos que la alternativa que se apega a las necesidades de la elaboración del sistema es la última; ya que es la que cumple con todos los requerimientos actuales del Comité.

II - ANALISIS

En la elaboración de un sistema existe diversas técnicas para llegar a su realización, entre las que destacan:

- Ciclo de vida clásico del desarrollo del sistema.
- Método de desarrollo por análisis estructurado.
- Método del prototipo de sistemas.

Ciclo de vida clásico. Exige un enfoque sistemático, secuencial, del desarrollo del software el cual consiste de los siguientes pasos:

- 1.- Ingeniería y análisis del sistema.- Abarca los requerimientos globales a nivel del sistema con un poco de análisis y diseño a un nivel superior.
- 2.- Análisis.- El analista debe comprender el dominio de la información, así como la función, rendimiento e interfases requeridas.
- 3.- Diseño.- Se basa en tres pasos principales; estructura de datos, arquitectura del software, detalle de procedimientos.
- 4.- Desarrollo e implementación.- Es la traducción del diseño a una forma legible para la máquina.
- 5.- Pruebas.- Se realizan pruebas para asegurar que la entrada definida producirá los resultados que realmente se desean.
- 6.- Mantenimiento.- Son los cambios que requerirá el sistema una vez puesto en marcha.

Método de desarrollo por análisis estructurado. Se concentra básicamente en lo que se quiere que haga el sistema sin importar como y con que se implantará, sus elementos son:

SINRECH

- 1.- Descripción gráfica.- Es una forma de describir el sistema; es un bosquejo que indica sus características, identifica las funciones y además muestra como interactua con otros elementos.
- 2.- Diagrama de flujo de datos (DFD).- Es el modelo del sistema, la descripción completa del mismo estará formada por un conjunto de DFD.
- 3.- Diccionario de datos.- Contiene todas las definiciones de los elementos en el sistema; datos, procesos y almacenes.

Método del prototipo del sistema. Se realiza un sistema que funciona, no es solo una idea en papel, desarrollado con el fin de probar ideas y suposiciones relacionadas con el sistema.

En la práctica no se puede aplicar un método exactamente como se ve en teoría, sino que se usan variaciones o combinaciones de los mismos, ya que el éxito del método aplicado depende de cuando se emplea, cómo se aplica y de los participantes en el proceso de desarrollo.

Para la realización de este sistema se utiliza la combinación del ciclo de vida clásico y el desarrollo por análisis estructurado, con algunas pequeñas variantes en alguno de los casos.

El análisis como ya vimos es una fase del ciclo del vida tradicional del sistema, consiste en el entendimiento del problema separado del cómo y con qué se desarrollará e implementará. Los datos son analizados independientemente de las aplicaciones que puedan usarlos, además, debe existir una alta participación del usuario.

El análisis implica la planeación del proyecto y el análisis costo-beneficio.

II.1.- PLANEACION DEL DESARROLLO DEL SISTEMA INTEGRAL DE RECURSOS HUMANOS.

La planeación del desarrollo del sistema nos indica cuales son las actividades en orden prioritario que se deben realizar, en este punto es importante considerar la actividad-tiempo.

En la figura [2.1] se observa la idea general del flujo de datos, tanto de entrada como salida, que se pretende controlar a través del presente sistema, con lo que a partir de este momento se tiene ya un primer panorama genérico de lo que se pretende involucrar en el mismo.

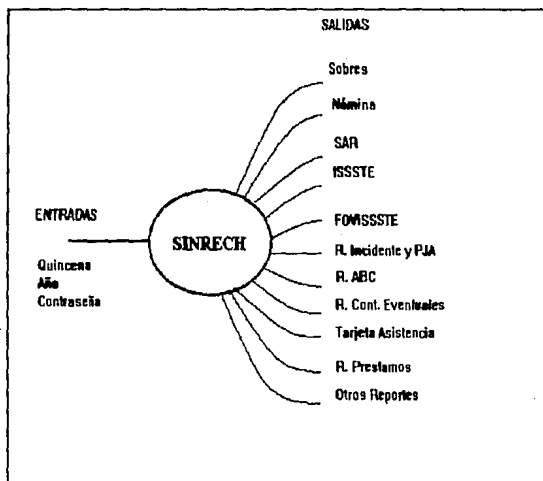


FIGURA 2.1 Nivel 0 del sistema SINRECH

Haciendo la descomposición funcional el sistema queda dividido en 5 diferentes módulos, para que a partir de estos se logre tener una comprensión más clara de cada parte que interviene en él, y para poder atacar los problemas más fácilmente. Estos módulos son los que se muestran en la siguientes figura (FIG. [2.2]):

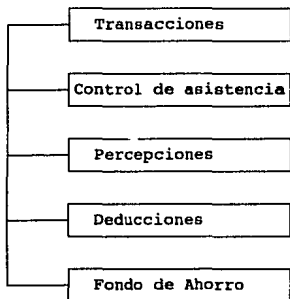


FIGURA 2.2 DESCOMPOSICION FUNCIONAL DE SINRECE.

La planeación se hará de acuerdo a las etapas del ciclo de vida clásico, como sigue (ver Tabla [2.1]):

SISTEMA	ANÁLISIS			DISEÑO			DESARROLLO E IMPLEMENTACIÓN			PRUEBAS		
	OPT.	TIP.	PES.	OPT.	TIP.	PES.	OPT.	TIP.	PES.	OPT.	TIP.	PES.
Transacciones	2	2.5	3	1.5	2	2.5	1	1.5	2	2.5	3	4
Percepciones	1	1.5	2	0.5	0.75	1	0.25	0.5	0.75	2.5	3	4
Deducciones	1	1.5	2	0.5	0.75	1	0.25	0.5	0.75	2.5	3	4
Ctrl. Asistencia	2	2.5	3	1.5	2	2.5	1	1.5	2	2.5	3	4
Totales	6	8	10	4	5.5	7	2.5	4	5.5	2.5	3	4

TABLA 2.1. MUESTRA LOS TIEMPOS ESPERADOS PARA CADA ETAPA DEL DESARROLLO DEL SISTEMA, Y ESTOS ESTAN DADOS EN MESES.

De acuerdo a la tabla [2.1], es importante aclarar que el inicio de cada etapa del desarrollo del sistema dependerá de la terminación de la anterior para cada subparte del sistema, de tal forma que la parte de diseño para transacciones comenzará inmediatamente después de haber concluido su análisis, y así sucesivamente. Podrán también traslaparse algunas actividades de los diferentes módulos según el avance que se vaya teniendo de cada uno de estos, como por ejemplo: puede ocurrir que se este trabajando en el diseño del sistema de percepciones, y a su vez estar terminando las etapas de análisis de los otros tres subsistemas ya mencionados.

II.2. SISTEMA DE TRANSACCIONES.

Transacción.- "Una transacción es cualquier suceso o actividad que afecta a toda la organización".¹

Esta definición de transacción nos da una primera idea acerca de lo que estas son y la importancia que tienen dentro de una organización y por lo tanto lo importante que son dentro del desarrollo del sistema.

Para el sistema, en las transacciones se contempla la afectación de las tablas de catálogo dentro de la base de datos, así como los principales movimientos de un empleado, por ejemplo altas, bajas y cambios; y los reportes y avisos derivados de los mismos.

El tipo de actualizaciones a realizar se efectúan como transacciones predefinidas y se tienen las siguientes:

ALTAS.

1.- *Alta general.* Actualiza la totalidad de los datos para un empleado.

¹ Análisis y diseño de sistemas de información, James A.Senn, 2a edición, Mc Graw Hill, pp. 25.

2.- Reingreso de un permiso sin goce de sueldo (PSGS). Da por terminado un permiso y modifica los registros necesarios para considerar nuevamente activo al trabajador.

3.- Reinstalación de suspensión. Reactiva al trabajador cuya causa de baja hubiese sido una suspensión.

BAJAS.

1.- PSGS. Pone a un trabajador inactivo por un permiso sin goce de sueldo.

- Se solicitan por un intervalo fijo.

- Se conceden al trabajador de base.

- Por razones de carácter particular hasta por seis meses, una sola vez cada año de calendario tomando como base la antigüedad del trabajador de la forma siguiente:

+ 30 días para quienes tengan de 6 meses a 1 año de servicio.

+ 90 días para quienes tengan de 1 a 3 años de servicio.

+ 180 días para quienes tengan más de 3 años.

Las licencias se solicitan por medio del sindicato, con una anticipación no menor de quince días procurando que los días de inicio y terminación del permiso correspondan al principio o al final de la quincena y que el trabajador liquide anticipadamente el pago de intereses derivados de préstamos del fondo de ahorro.

2.- Abandono. El trabajador deja de asistir a sus labores sin una causa que lo justifique.

Abandono de labores técnicas.- Se refiere al abandono de su actividad en las labores técnicas relativas al funcionamiento de maquinaria o equipo, o a la atención de personas; a la deficiencia de un servicio o que ponga en peligro la salud y vida de las personas.

3.- Cese. El trabajador incurre en una falta que le hace perder los derechos de su nombramiento.

4.- Incapacitación. Incapacidad permanente del trabajador

física o mental que le impida el desempeño de sus labores conforme al dictamen médico del ISSSTE.

5.- *Jubilación.* El trabajador es jubilado de acuerdo al reglamento del ISSSTE, para lo cual cuando comience sus trámites se le concede una licencia con goce de salario por el término de tres meses.

6.- *Defunción.* En caso de fallecimiento de un trabajador se genera un pago independiente del seguro de vida y equivalente a seis meses del salario total asignado a dicho trabajador.

7.- *Renuncia.* Separación voluntaria del trabajador en este caso se extiende una gratificación a razón de 13 días de salario total por cada año de servicios prestados siempre que se haya cumplido 15 años de servicio efectivo.

8.- *Terminación de contrato eventual.* Se considera que cuando un contrato eventual expira se renueva en forma automática excepto para aquellos casos en que se especifiquen como no recontractaciones.

9.- *Suspensión.* Suspensión en forma temporal del trabajador.

CAMBIOS.

1.- *Cambios sobre plaza.*

- *Promoción.* Cualquier cambio que repercute en un incremento de salario del trabajador, sin tomar en cuenta los aumentos de salario en forma general ni ajustes al tabulador ni el cambio de una plaza provisional a una definitiva.

- *Cambio de propietario.*

- *Cambio de Jefatura o Gerencia (JOG).*

2.- *Otros cambios.* Cambios de los campos que componen cualquier tabla excepto las llaves primarias.

En la siguiente tabla (TABLA [2.2]) se muestran los diferentes datos relacionados con cada una de las transacciones antes mencionadas.

TRANSACCION	DATOS RELACIONADOS	COMENTARIOS
ALTAS		
Alta General	<ul style="list-style-type: none"> - Numero empleado - Nombre(s) - Apellido paterno - Apellido materno - Registro Federal - Sexo - Calle y número - Colonia - Localidad - Estado - Código Postal - Teléfono - Nacionalidad - Estado Civil - Número de issste - Fecha de ingreso - Fecha de derechos laborales - Fecha de base - Sindicalizado - Antigüedad en el Gobierno Federal - Sueldo 	<p>El número de empleado es el número de identificación dentro de la misma empresa.</p> <p>Es importante decir que plaza ocupará el empleado.</p> <p>El domicilio y el número de ISSSTE son r e q u i s i t o s importantes.</p> <p>La antigüedad en el Gobierno Federal se toma en cuenta para aquellas personas que ya hayan laborado antes en el Gobierno.</p>
Reingreso de PSGS	<ul style="list-style-type: none"> - Número empleado - Fecha de inicio - Fecha terminación 	Se desea que este reingreso se realice automáticamente una vez terminado el período del mismo.
Reinstalación de suspensión	<ul style="list-style-type: none"> - Número empleado - Fecha de inicio - Fecha terminación 	Se desea que esta opción reinstale además de los empleados suspendidos también los de terminación de contrato eventual

TABLA [2.2] TRANSACCIONES Y SUS RELACIONES

CONTINUA...

...CONTINUACION

TRANSACCION	DATOS RELACIONADOS	COMENTARIOS
BAJAS		
Permiso Sin Goce de Sueldo	- Número empleado - Fecha de inicio - Fecha terminación	
Abandono	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una baja general. En el caso que se considere como baja general, se deberá pedir la causa de baja.
Cese	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una baja general. En el caso que se considere como baja general, se deberá pedir la causa de baja.
Incapacidad	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una suspensión.
Jubilación	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una baja general. En el caso que se considere como baja general, se deberá pedir la causa de baja.

TABLA [2.2] TRANSACCIONES Y SUS RELACIONES

CONTINUA...

... CONTINUACION

TRANSACCION	DATOS RELACIONADOS	COMENTARIOS
BAJAS		
Defunción	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una baja general. En el caso que se considere como baja general, se deberá pedir la causa de baja.
Renuncia	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una baja general. En el caso que se considere como baja general, se deberá pedir la causa de baja.
Terminación de contrato Eventual	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se puede considerar como una suspensión.
Suspensión	- Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Fecha de ingreso - Fecha derechos laborales - Plaza	Se considerar como una suspensión de labores.

TABLA [2.2] TRANSACCIONES Y SUS RELACIONES

CONTINUA...

...CONTINUACION

TRANSACCION	DATOS RELACIONADOS	COMENTARIOS
CAMBIOS		
Cambios sobre plaza	<ul style="list-style-type: none"> - Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Plaza - Categoría a la que pertenece - Jefatura o Gerencia 	
Otros cambios	<ul style="list-style-type: none"> - Número empleado - Nombre(s) - Apellido paterno - Apellido materno - Registro Federal - Sexo - Calle y número - Colonia - Localidad - Estado - Código Postal - Teléfono - Nacionalidad - Estado Civil - Número de issste - Fecha de ingreso - Fecha de derechos laborales - Fecha de base - Sindicalizado - Antigüedad en el Gobierno Federal - Sueldo 	Estos cambios se pueden dividir en dos partes, los cambios laborales y los cambios personales. Solo se pueden hacer cambios en campos que no sean llaves primarias.

TABLA [2.2] TRANSACCIONES Y SUS RELACIONES

II.3. SISTEMA DE CONTROL DE ASISTENCIA

Es la parte del sistema integral de recursos humanos, encargada de manejar la puntualidad y la asistencia de los empleados, considerando los permisos y justificaciones a que tienen derecho antes de deducir de éstos los descuentos o pagar los estímulos que se marcan en las condiciones generales de trabajo.

Incidente.- Evento del registro de entrada y/o salida diferente de la norma registrada en las condiciones generales de trabajo (y en algunos casos al horario especial autorizado a un empleado en particular); como son:

- Horas extra
- Cuota dominical
- Trabajo en día festivo
- Omisión de registro de salida
- Ausencia
- Omisión del registro de entrada
- Retardo
- Salida antes de horario

Permiso.- Solicitud y autorización previa al hecho registrado en la tarjeta de asistencia.

La solicitud deberá estar autorizada por la gerencia o jefatura de zona correspondiente, sin que esto implique necesariamente la aceptación del sistema (en la computadora) de acuerdo al registro de permisos previos, los permisos son:²

- P. Económico
- P. Por comisión de trabajo
- P. Por cuidados maternos
- P. A cuenta de vacaciones
- P. Por primer día de CLIDA
- P. Por segundo día de CLIDA
- P. Especial no registrar entrada

² Los permisos de retardo y/o salida antes de tiempo por periodo de lactancia se registrarán como autorización de horario especial.

- P. Especial no registrar salida
- P. Por fallecimiento de familiar

Justificación.- Presentación posterior al documento expedido por el médico del ISSSTE o la aceptación del gerente o jefe de zona que corresponda, de un hecho previamente registrado mecánicamente en la tarjeta de asistencia, las justificaciones son:

- Incapacidad médica por enfermedad
- Incapacidad médica por gravidez
- Incapacidad médica por intervención quirúrgica
- Autorización médica para salir
- Autorización de retardo
- Autorización de omisión de registro de entrada
- Autorización de omisión de registro de salida por motivos personales
- Autorización de omisión de registro de salida por motivos de trabajo

Anuencia.- Oficio del gerente o jefe de zona que autoriza el pago de un incidente registrado en la tarjeta de asistencia y que registrando un hecho positivo de asistencia o puntualidad deba reflejarse en un pago adicional en la nómina, las anuencias son:

- Anuencia de pago de horas extras
- Anuencia de trabajo en día festivo

Los objetivos de esta parte del sistema son:

- a) Registro mecánico de horas de llegada y partida de cada trabajador.
- b) Identificación de aquellos registros, de acuerdo al horario común o particular que signifiquen un incidente que deba ser captado porque afecte el pago quincenal o de los premios de puntualidad y asistencia.
- c) La captación de los oficios ya autorizados por parte de las autoridades que se relacionen con los incidentes, ya sea para inhibirlos, anularlos o autorizarlos. Los oficios pueden ser de permiso, justificación o anuencia (PJA) .

- d) La autorización deberá hacerse según las condiciones generales de trabajo y en base a los registros previos del semestre en curso.
- e) El sistema será capaz en cualquier momento que se le requiera de informar sobre estos acumulados de permisos, justificaciones y anuencias.
- f) La anulación, complementación y en su caso, aceptación de cada incidente con su correspondiente PJA, diferenciando aquellos que deban considerarse como usados o rechazados y acumulándolos para futuras consideraciones.
- g) Una vez que se han detectado los incidentes que deban considerarse para efectos de pago o deducción, se traducirán en la correspondiente orden de pago o descuento.
- h) Aquellos incidentes que afecten el pago de premios se registrarán para procesos semestrales.
- i) Todos aquellos permisos, justificaciones y anuencias que se hayan emparejado con el incidente al que hacen referencia se acumularán para efectos de consulta y aceptación de futuros oficios.
- j) Aquellos oficios que se capten fuera, o sea, posteriormente, de la quincena en proceso, se tomarán en cuenta para anular previos incidentes de asistencia y canalizarse como retribuciones por orden de pago especial.

INCIDENTE	DOCUMENTO COMPLEMENTARIO (PJA)	EFEECTO Y MODO DE OPERACION
Horas extra	Anuencia del pago de horas extra	- Para pagarse requiere existir el oficio de anuencia. - Se pagan tantas horas como se registre en la tarjeta y que sean igual o menos que el número autorizado en el oficio de anuencia.
Cuota Dominical		Se paga = salario diario + 25% del salario diario.
Trabajo en día festivo	Oficio de anuencia de trabajo en día festivo	- Requiere el correspondiente oficio de anuencia de trabajo en día festivo. - Se paga = pago del día + 2 veces el pago del día.
Omisión del registro de salida	- Permiso de no registrar la hora de salida por CLIDA segundo día. - Autorización médica de salir.	- Si existe la correspondencia se anulan los efectos en el pago quincenal y de permisos semestrales. - Si no hay se considera ausencia para pago y premio.
Ausencia	Una ausencia se anula por: -Permiso económico - Comisión de trabajo - Cuidados maternos	- Verificar que el permiso económico proceda de acuerdo a los permisos acumulados en el año (máximo 9). - Requiere permiso de comisión con el periodo de duración. - Para efectos del pago del premio no debe exceder de 6 días al semestre. En caso de ser más, se neutraliza el descuento en nómina pero se pierde el premio.

TABLA [2.3] RELACION ENTRE INCIDENTES DE ASISTENCIA Y LOS DOCUMENTOS COMPLEMENTARIOS (PJA)

CONTINUA...

...CONTINUACION

INCIDENTE	DOCUMENTO COMPLEMENTARIO (PJA)	EFFECTO Y MODO DE OPERACION
Ausencia	<ul style="list-style-type: none"> - Incapacidad por enfermedad - Permiso por fallecimiento de familiar. - Permiso de no registrar la entrada. - Justificación por aceptación de no registrar entrada (Autorización). <p>NOTA: El permiso por período de lactancia se canalizará como un horario especial y por lo tanto los retardos no se identificarán como tales en el paso de identificación de incidentes.</p>	<ul style="list-style-type: none"> - Requiere el correspondiente comprobante médico. - Asegurarse que no sean más de tres días por semestre. - Si no hay permiso correspondiente, se descuenta el día y se pierde el premio. Pues trabaja como una ausencia. - Posterior al hecho mismo si no existe correspondencia actúa como una ausencia, se descuenta el día y se pierde el premio. <p>Si no hay permiso ni justificación entonces:</p> <ul style="list-style-type: none"> - Almacenar retardo para efecto de pérdida de premio de puntualidad. - Sumarizar retardos por quincena. - Si total por quincena menos de 30 mins. Descuento = 1/2 día. - Si total por quincena esta entre 31 y 59. Descuento = 1 día. - Si total por quincena es mayor de 59. Descuento = 2 días.
Salida antes de tiempo	Igual que el retardo por período de lactancia no se identificarán como incidente ya que se manejará como horario especial.	Si no hay tal horario especial se maneja como omisión de salida.

TABLA [2.3] RELACION ENTRE INCIDENTES DE ASISTENCIA Y LOS DOCUMENTOS COMPLEMENTARIOS (PJA)

II.4 SISTEMA DE PERCEPCIONES.

El sistema de percepciones, comprende todas aquellas actividades encaminadas al pago de las prestaciones a las cuales un empleado tiene derecho, algunas prestaciones son reflejadas directamente en nómina y otras son enviadas a organismos externos para su manejo y cuidado por ejemplo el SAR (Sistema de Ahorro para el Retiro).

En la actualidad el cálculo de las percepciones se realiza por medio de los dos sistemas que están en operación, incluso se han agregado módulos nuevos que no pertenecen a ninguno de los dos para llevar a cabo el control de estas, un ejemplo de estos módulos es el de captación de incidentes.

El objetivo principal del sistema de percepciones es conjuntar todos los módulos existentes, en uno sólo y de esta forma tener mayor control sobre los mismos.

Existen tres formas distintas, por medio de las cuales se pueden generar las ordenes de percepción que son:

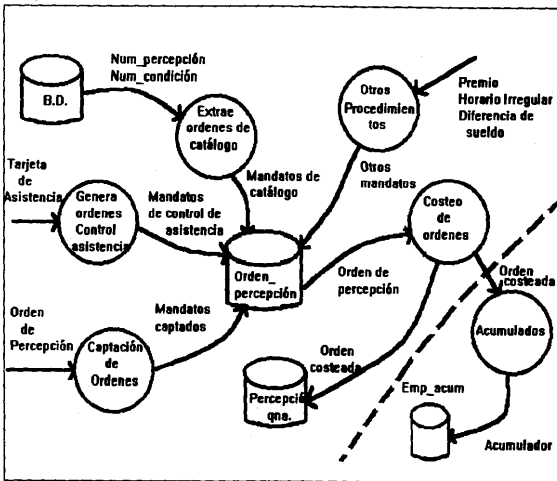
a) Por catálogo.- Son percepciones que se generan para aquellos empleados que cumplen con las condiciones que exige la percepción.

b) Por programas.- Son percepciones que se generan por medio de programas, los cuales generan las ordenes a partir del Control de Asistencias, del cálculo de premios, aguinaldos, SAR, FOVISSSTE, etc.

c) Por orden.- Son percepciones adicionales que no se pueden generar por catálogo, ni por programa, que no tienen un procedimiento fijo para generarlas, si no que requieren de un mandato para su elaboración.

En la figura [2.3] se puede observar el procedimiento general para la generación de ordenes.

FIGURA [2.3] PROCESO DE PERCEPCIONES.



ANÁLISIS

Para las ordenes que se generan por catálogo, por un lado el empleado cumple con ciertas condiciones específicas tales como:

- Que el empleado tenga nombramiento activo.
- Que su tipo de puesto sea mando medio o superior.
- Que tenga horario irregular quincenal, o bien horario irregular mensual.
- Que el tipo de puesto sea de base.
- Que el tipo de puesto sea de base o de confianza.
- Que el empleado tenga hijos menores de 6 años.
- Que exista beca.
- Que su nombramiento sea diferente de eventual.

Algunas percepciones para generarse necesitan cumplir con ciertas condiciones, las cuales se muestran en la tabla [2.4].

PERCEPCION	CONDICION
Sueldo	- Nombramiento activo
Prima quincenal	- Quincena par - Nombramiento activo - Nom. distinto de eventual
Horario irregular quincenal	- Nombramiento activo - Hor. irreg. quincenal
Prima vacacional B	- Quin. evento pago de vacac. - Nombramiento activo - Puesto Base o Confianza - Nom. distinto de eventual
Prima vacacional A	- Quin. evento pago de vacac. - Tipo de puesto Mando Medio o Superior - Nombramiento activo. - Nom. distinto de eventual
Compensación gerencial	- Compensación de fin de año - Nombramiento activo - Puesto igual a Confianza
Compensación especial	- Quin. evento comp. de abril - Puesto Base o Confianza

TABLA [2.4] PERCEPCION-CONDICION

CONTINUA...

...CONTINUACION

PERCEPCION	CONDICION
Asig. Servicios Especiales	- Quincena par - Nombramiento activo - Puesto Base o Confianza - Nom. distinto de eventual
Previsión Social Múltiple.	- Quincena non. - Nombramiento activo - Puesto Base o Confianza
Beca	- Quincena par - Quin. evento pago de beca - Nombramiento activo.
Despensa	- Quincena non - Nombramiento activo
Ayuda por serv. espec.	- Quincena par - Nombramiento activo - Puesto Base o Confianza. - Nom. distinto de eventual
Inscripción guardería	- Quin. pago inscrip. guardería - Nombramiento activo - Hijos menores de 6 años
Mensualidad guardería	- Quincena par - Nombramiento activo - Hijos menores de 6 años
Ajuste salarial	- Compensación fin de año - Nombramiento activo - Puesto Base o Confianza - Nom. distinto de eventual
Ayuda por servicio	- Quincena par - Nombramiento activo - Puesto Base o Confianza
Horario irreg. mensual	- Quincena par - Hor.irregular mensual - Nombramiento activo.

TABLA [2.4] PERCEPCION-CONDICION.

Quincenalmente hay condiciones que son llamadas generales ya que se utilizan para todos los empleados, estas son las siguientes:

- Si es quincena par.
- Si es quincena non.
- Si el evento corresponde al pago por concepto del pago de vacaciones.

SINRECH

- Si el evento corresponde al pago de la quincena por concepto de la compensación otorgada en abril.
- Si la quincena en cuestión corresponde al pago de la beca.
- Si la quincena en cuestión corresponde al pago de inscripción en guardería.
- Si la quincena en proceso corresponde a la compensación por fin de año.

De la lista anterior, tenemos que existen eventos que se cumplen para determinadas quincenas, como se observa en la tabla [2.5].

No. QUINCENA	EVENTO
13	Pago de vacaciones
23	Pago de vacaciones
7	Compensación de abril
10	Pago de beca
17	Pago de guardería
23	Pago de aguinaldo
23	Compensación de fin de año

TABLA [2.5] QUINCENA EVENTO.

Por lo que después de analizar cuales son las condiciones (tanto generales como particulares) con las cuales cumple el empleado, se compara con las condiciones de cada una de las percepciones y se generan las ordenes para aquellas que el empleado cumpla con las condiciones.

Existen también las percepciones que se generan por orden. Estas percepciones son de carácter especial ya que en estas no existe un costeo definido para las percepciones que se generan por este medio.

Este tipo de percepciones, se realizan cuando el departamento de Recursos Humanos lo considera conveniente, para lo cual se requiere que exista un incidente previo para algún empleado que realmente lo amerite. Las percepciones así generadas se realizan a partir de una orden especial llamada incidente, las cuales posteriormente son capturadas ya como una orden de percepción, misma que ya fue costeada de acuerdo a lo que el departamento de Recursos Humanos, considero como adecuado.

Entre este tipo de percepciones tenemos como ejemplo; dote matrimonial, gastos de funeral, prótesis, lentes, aparatos ortopédicos, etc.

Este tipo de percepciones solo tendrán interacción con el sistema una vez que se haya realizado su captura correspondiente.

El último tipo de percepciones que se generan se realizan a través de la activación de los programas correspondientes. Estos programas deberán ser realizados según las características de cada percepción. La activación de estos programas podrá realizarse de manera semiautomática, o bien de forma manual, en el tiempo en que el departamento de Recursos Humanos lo considere conveniente.

Este tipo de percepciones son aquellas que no se generan ni por catálogo, ni por ordenes, y que el costeo para la realización de las mismas requiere de condiciones especiales, las cuales son calculadas a partir de los programas encargados de la generación de éstas.

Se dice que algunas de éstas percepciones se realizan de manera semiautomática, ya que se generan a partir del subsistema de Control de Asistencias una vez que es activada la generación de la nómina, la cual emite ciertos mandatos de percepciones, sin que exista una tecla o un menú especial para su activación. Entre las percepciones así generadas tenemos:

Pago de horas extras, pago por trabajo en días festivos, pago de prima dominical.

SINRECH

Los programas que se realizan de manera manual, requieren de la opción de un menú, para su correspondiente elaboración. Algunas de las percepciones así generadas son:

Aguinaldo, premio de puntualidad, premio de asistencia, compensación de premio, notas buenas, etc.

En la tabla [2.6] se observan las percepciones que se manejan, su origen, la periodicidad con que se pagan, a que deducciones afectan y la forma en que se costean.

Los números que se muestran en la tabla indican lo siguiente:

- 1 = ISPT
- 2 = Fondo de ahorro
- 3 = ISSSTE
- 4 = FOVISSSTE
- 5 = Cuota sindical
- 6 = Pensión alimenticia ordinaria
- 7 = Sistema de ahorro para el retiro (SAR)

DESCRIPCION	ORIGEN	PERIODO								COSTEO
			1	2	3	4	5	6	7	
Sueldo	Catálogo	Quincenal	x	x	x	x	x	x	x	Cada plaza tiene asignado un sueldo mensual(SM), cuando el empleado es eventual o interino generalmente no tiene el 100% del valor de la plaza, por lo que el sueldo del empleado(SE) será $SM * \frac{SE}{SM}$, el sueldo quincenal es $SE/2$.
Prev. soc. múltiple	Catálogo	Mensual								La previsión social múltiple es una cantidad fija.
Horario Irreg.	Catálogo	Mensual o Quincenal	x						x	Esta percep. no es para todos los empleados, la G. de R.H. es la encargada de captar los empleados que la tienen y asignarle un monto fijo así como la periodicidad de pago.
Retroactivo compensa.	Programa	Excepcional	x						x	Si hay un cambio de sueldo y esté afecta la compensación, se debe bonificar a los empleados que tuvieron la comp. en el porcentaje de aumento del sueldo.
Asig. por serv. esp.	Catálogo	Mensual								Se trata de una cantidad fija.
Prima Quinquenal	Catálogo	Quincenal	x	x	x	x	x	x		Se le paga a los empleados de base, dependiendo del No. de años que tengan laborando para el Gob. Federal, 1 quinq. = 5 años.

TABLA [2.6] PERCEPCIONES

CONINDA...

SINBRECE

...CONTINUACION

DESCRIPCION	ORIGEN	PERIODO	1	2	3	4	5	6	7	COSTEO
Beca	Catálogo	Mensual								Es una cantidad fija, dependiendo del nivel de la beca.
Tiempo extra	Programa	Excepcional	x					x		Las horas extras se pagan al doble.
Prima dominical	Programa	Excepcional								Sueldo diario * 0.25.
Bonifica.	Orden	Excepcional								Costo de la orden.
Aguinaldo	Programa	Anual						x		Son 45 días de salario, o parte proporcional.
Compen. especial	Programa	Anual	x					x		Sueldo diario * 45, o parte proporcional (solo personal de confianza).
Premio puntualidad y asistencia	Programa	Semestral	x					x		Solo a personal de base.
Prima vacacional	Programa	Semestral	x					x		Son 3.5 días de salario para nómina "B" o 12 días para nómina "A".
Dote matrimonial	Orden	Excepcional								Son 35 días de salario.

TABLA [2.6] PERCEPCIONES

CONTINUA...

...CONTINUACION

DESCRIPCION	ORIGEN	PERIODO	1	2	3	4	5	6	7	COSTEO
Gastos de funeral	Orden	Excepcional								Es una cantidad fija según el reglamento.
Prest. Económicas	Orden	Excepcional								Costo de la orden.
Despensa	Catálogo	Mensual								Cantidad fija según el reglamento.
Ajuste salarial	Programa	Anual								Son 3 días de salario, solo base.
Otras Percep.	Orden	Excepcional	x					x		Costo de la orden.
Lentes	Orden	Excepcional								Costo de la orden.
Guardería	Catálogo	Mensual								Cantidad fija según reglamento.
Retroact. de sueldo	Programa	Excepcional	x	x	x	x	x	x	x	Porcentaje de aumento de sueldo.
Comp. X trabajo extraord.	Programa	Mensual						x		Cantidad fija según reglamento.
Aparatos Ortopédicos	Orden	Excepcional								Es una cantidad fija según la orden.

TABLA [2.6] PERCEPCIONES

CONTINUA...

SINRECH

...CONTINUACION

DESCRIPCION	ORIGEN	PERIODO	1	2	3	4	5	6	7	COSTEO
Prótesis	Orden	Excepcional								Es una cantidad fija según el costo de la orden.
Comp. Art. 76	Programa	Anual	x					x		Son 45 días de salario.
Retroactivo quinquenio	Programa	Excepcional	x	x	x	x	x	x		Porcentaje de aumento a los quinquenios.
Bonif. ISPT	Orden	Excepcional								Costo de la orden.
Retroactivo despensa	Programa	Excepcional								Porcentaje de aumento a la despensa.
Ayuda para licencia (Choferes)	Orden	Excepcional								Costo según la orden.
Diferencia de sueldo	Catálogo	Mensual o Quincenal	x					x		Esta percep. no es para todos los empleados, la G. de R.H. es la encargada de captar los empleados que la tienen y asignarle un monto fijo así como la periodicidad de pago.
Ayuda por Servicios	Catálogo	Mensual								Cuota fija según Gobierno.

TABLA [2.6] PERCEPCIONES

CONTINUA...

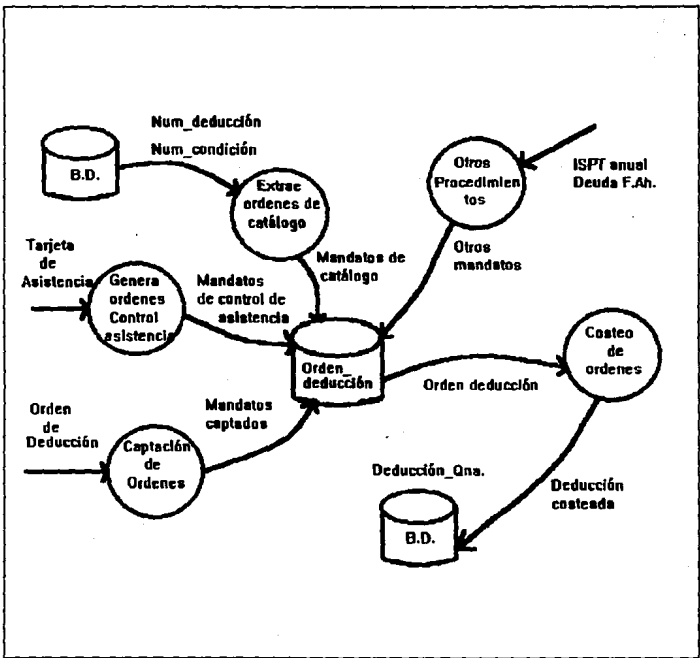


FIGURA [2.4] PROCESO DE DEDUCCIONES

Este sistema se encargará de generar y aplicar las deducciones que el empleado tiene obligación de pagar, ya sea al organismo o a dependencias externas, como son: Secretaría de Hacienda y Crédito Público, ISSSTE, FONAC, etc.

Al igual que las percepciones, el cálculo de las deducciones es realizado en la actualidad por más de un sistema, por lo que el objetivo de esta parte es unir los módulos existentes en uno solo.

Las formas por las que se genera una orden de deducción son:

a) Por catálogo.- Son aquellas deducciones para las cuales el empleado cumple con las condiciones que estas requieren.

b) Por programa.- Son aquellas que se generan por medio de programas a partir del Control de Asistencia, ISPT Anual, ISPT Complementario, etc.

c) Por orden.- Son aquellas que requieren un mandato para su generación.

En la figura [2.4] se observa el procedimiento general para la generación de una deducción.

Para las ordenes que se generan por catálogo se tiene que el empleado cumple con condiciones específicas como son:

- Que el empleado pague pensión alimenticia ordinaria
- Que el plan del fondo de ahorro del empleado sea A
- Que el plan del fondo de ahorro del empleado sea B
- Que el empleado sea sindicalizado o no lo sea
- Que el nombramiento del empleado no sea eventual
- Que el empleado esté activo

Para las deducciones solo existe una condición a la cual se le considera como general:

- Si el evento corresponde al cálculo del ISPT anual.

En la tabla [2.7] se observan las condiciones que deben cumplir las deducciones que se generan por catálogo:

DEDUCCION	CONDICION
I.S.P.T.	- Empleado activo
I.S.P.T. anual	- Si el evento corresponde al cálculo del ISPT anual
Aportación ISSSTE	- Empleado activo
Aportación Fondo de ahorro 'A'	- Que el plan del fondo de ahorro del empleado sea 'A' - Empleado activo
Aportación Fondo de ahorro 'B'	- Que el plan del fondo de ahorro del empleado sea 'B' - Empleado activo
Cuota sindical	- Empleado sindicalizado - Nombramiento del empleado no sea eventual - Empleado activo
Seguro colectivo de retiro	- Empleado activo
Aportación FONAC	- Empleado sindicalizado - Empleado activo

TABLA [2.7] DEDUCCION - CONDICION

Entonces tenemos que existe un evento que se cumple en una quincena determinada, el cual lo ponemos en la siguiente tabla:

No. QUINCENA	EVENTO
8	I.S.P.T. anual

El procedimiento que se sigue para la generación de las órdenes de deducción por catálogo es como sigue:

Se ve con cuales condiciones (tanto generales, como particulares) son con las que cumple el empleado, se compara con las condiciones que debe de cumplir las deducciones que se están

analizando y las que cumplan se generan.

Por otro lado tenemos las deducciones que se generan por orden, para este caso se genera la orden cuando se recibe un mandato especial, es decir, no hay condiciones con las que se deba cumplir, ni hay procedimientos que se repitan en forma algorítmica para su costeo.

Estas deducciones se generan, por ejemplo, cuando el empleado solicita un préstamo al ISSSTE y por lo tanto es esta institución la que manda la orden de en cuantas quincenas se debe de cubrir el préstamo y cuanto se amortiza con intereses en cada quincena, así como el ISSSTE existen otras deducciones más que también se generan por orden.

Estas deducciones entran a interactuar con el sistema hasta que se capturan.

Por último tenemos las deducciones que se generan por programa, estos programas toman en cuenta las características propias de cada una de las deducciones que se analizan; la activación de los programas se realiza en forma semiautomática o manual, en el tiempo en que la Gerencia de Recursos Humanos lo considera necesario.

Se dice que las deducciones se manejan en forma semiautomática por que se activan al detectar que otro proceso se esta llevando a cabo, por ejemplo, la generación de la nómina, o en forma manual, por que se necesita que a través de un menú se active el proceso y esto lo debe de realizar un operador.

Las deducciones por programa tienen que realizarse así, por que en algunos casos interactúan directamente con otros subsistemas como el de fondo de ahorro, o el de control de asistencia.

Entre las deducciones que se generan por programa tenemos las siguientes:

- Ausencias
- Retardos
- Amortización

- Préstamo Fondo de Ahorro 'A'
- Préstamo Fondo de Ahorro 'B'
- Intereses del Fondo de Ahorro, etc.

En la siguiente tabla [2.8] se ponen las deducciones que se procesan en el sistema en donde se ve su origen y la forma en que se costean, así como en el período en el que se descuentan.

DESCRIPCION	ORIGEN	PERIODO	COSTEO
Ausencias	Programa	Excepcional	Se descuenta un día por ausencia
I.S.P.T	Catálogo	Quincenal	Costeo quincena impar ³ Costeo quincena par ⁴
Aportación ISSSTE	Catálogo	Quincenal	El 8% de percepciones que afectan al ISSSTE
Aportación FAH 'A', aportación FAH 'B'	Catálogo	Quincenal	El 4% de percepciones que afectan al FAH para empleados de base y confianza. El 16% de la base de ahorro para mandos medios y superiores
Cuota sindical	Catálogo	Quincenal	El 5% de percepciones que afectan a la cuota sindical

TABLA [2.8] DEDUCCIONES

CONTINUA...

³ Se calcula un sueldo base del empleado (SB) en base a todas las percepciones gravables que el empleado tenga en la quincena, tomando en cuenta su sueldo mensual. Se ubica el SB dentro de algunos de los rangos que se marcan en la tabla de ISPT mensual proporcionada por la SHCP, se hace Excedente (E) = SB - Límite Inferior (L). $A = E * \% \text{impuesto}$, $B = A + \text{Cuota fija}$ donde A es una variable intermedia y B es el impuesto. Se calcula un subsidio al impuesto: Subsidio (S) = $(A * \% \text{subsidio}) + \text{Cuota fija}$ (el % del subsidio y la cuota fija se encuentran dentro de la tabla del subsidio mensual, también proporcionada por la SHCP), dependiendo del tipo de puesto que el empleado tenga, el subsidio se multiplica por 0.5 para empleados de base y confianza, y por 0.004 para empleados de mando medio o superior, por otro lado de la tabla del crédito al salario que proporciona la SHCP se ubica el SB dentro de alguno de los rangos y C = Cuota fija. Entonces $\text{ISPT mensual} = B - S - C$. Como es la primer quincena del mes, $\text{ISPT impar} = \text{ISPT mensual} / 2$ que corresponde al pago quincenal del impuesto mensual que todo empleado debe pagar a la SHCP.

⁴ Se calcula el SB en base a las percepciones gravables del empleado de la segunda quincena del mes más el SB de la primer quincena, con este SB se realiza el mismo procedimiento que se llevo a cabo para la quincena impar hasta obtener el ISPT mensual, el $\text{ISPT quincena par} = \text{ISPT mensual} - \text{ISPT quincena impar}$; con lo que el empleado esta pagando el ISPT que corresponde al mes que se procesa en base a las percepciones que recibió en el mismo (percepciones gravables).

...CONTINUACION

DESCRIPCION	ORIGEN	PERIODO	COSTEO
Días no laborados	Programa	Excepcional	Un día de salario por cada día no laborado
Incapacidad médica	Orden	Excepcional	Según lo que marque el ISSSTE
Amortización FAH 'A', amortización FAH 'B'	Programa	Quincenal	Cuando un empleado pide un préstamo, se le otorga un No. de quincenas para que amortice el mismo, entonces la amortización quincenal será: monto préstamo / No. quincenas a amortizar
Interés FAH 'A', interés FAH 'B'	Programa	Quincenal	Para sistema A: 0.5% sobre saldos insolutos, para sistema B: 1% sobre saldos insolutos
Préstamo corto plazo, mediano plazo hipotecario directo y fondo de garantía	Orden	Quincenal	El ISSSTE manda la orden de descuento indicando durante cuantas quincenas se realizará el descuento marcado
Renta multifamiliar e hipotecario FOVISSSTE	Orden	Quincenal	El FOVISSSTE manda la orden de descuento indicando durante cuantas quincenas se realizará el descuento marcado
Adeudo serv. médico	Orden	Excepcional	El ISSSTE manda la orden de descuento
Retardos	Programa	Excepcional	Hasta 10 min. no hay descuento, después de 10 min. hasta 15 min. medio día, de 15 min. hasta 30 min. un día y después de 30 min día y medio

TABLA [2.8] DEDUCCIONES

CONTINUA...

...CONTINUACION

DESCRIPCION	ORIGEN	PERIODO	COSTEO
Seguro colectivo de retiro	Catálogo	Quincenal	Cantidad fija, la marca el gob. federal
Pensión alimenticia ordinaria	Orden	Quincenal	Se aplica el porcentaje marcado en la orden al total de las percepciones del empleado
Pensión alimenticia extraord.	Orden	Quincenal	Se aplica el porcentaje marcado en la orden a las percepciones que afectan esta deducción
Pensión alimenticia fija	Orden	Quincenal	Se aplica el descuento marcado en la orden
Adeudo a CAFCE	Orden	Quincenal	El descuento marcado en la orden
Otras deducciones	Orden	Excepcional	Empleada como auxiliar para llevar a cabo descuentos no contemplados en esta lista, costo de la orden
Multiseguro (ASEMEX)	Orden	Quincenal	El costo de la orden, en el número de quincenas que marca la misma
Aportación FONAC	Catálogo	Quincenal	Es una cantidad fija marcada por el Gob. Fed.
ISPT anual	Catálogo	Anual	

TABLA [2.8] DEDUCCIONES

CONTINUA...

II.6 SISTEMA DE FONDO DE AHORRO.

OBJETIVO.

Los objetivos para los cuales se constituye el Fondo de Ahorro son:

- A) Fomentar el ahorro de los socios.
- B) Obtener recursos que ayuden a resolver las necesidades económicas de los socios a través de préstamos que serán de dos clases:
 - 1.- Préstamos personales
 - 2.- Préstamos de ayuda social.

POLITICAS.

Los sistemas del Fondo de Ahorro serán dos con las siguientes denominaciones:

A) Sistema "A".- Las aportaciones serán con fines de capitalización teniendo las siguientes características:

- i) Podrán optar por este sistema los socios que así lo deseen.
- ii) El socio que opte por este sistema formara parte de el con el total de sus aportaciones.

B) Sistema "B".- Tendrá vigencia por tres años, iniciando el día 16 de noviembre y concluyendo el día 15 de noviembre correspondiente, coincidiendo con el ejercicio presidencial.

En ningún caso los recursos del Fondo de Ahorro podrán ser concedidos para fines que no sean los establecidos en el reglamento.

El socio que decida por voluntad propia separarse del sistema "B" no podrá ingresar al sistema "A" sino hasta un nuevo ejercicio trianual.

NORMAS.

La administración del Fondo de Ahorro, utilizará los préstamos a los socios en los términos del reglamento y la reglamentación interna emitida por el organismo.

Las solicitudes de préstamo se presentaran a la gerencia de Recursos Humanos para que esta verifique los datos registrados en

la solicitud y la vigencia de derechos, turnándolas a la administración para su aprobación por riguroso número de folio.

La emisión de cheques se hará los días 15 y 30 de cada mes, para su entrega oportuna.

Los préstamos personales serán únicamente para los socios de los sistemas "A" y "B", con cargos a sus respectivos capitales.

Los préstamos de ayuda social solamente podrán otorgarse cuando exista acuerdo unánime de la administración y la comisión de vigilancia del fondo.

El socio que se encuentre disfrutando de licencia sin goce de sueldo, no podrá solicitar el otorgamiento de préstamo al Fondo de Ahorro.

En el sistema "B" los socios sin haber sido dados de baja en forma definitiva del C.A.P.F.C.E. podrán solicitar su retiro voluntario del Fondo de Ahorro.

ACTIV. No.	UNIDAD RESPONSABLE	DESCRIPCION DE LA ACTIVIDAD
1	Gerencia de Recursos Humanos	Recibe solicitud préstamo personal o ayuda social del trabajador, en original y dos copias. Registrándolo y asignándole folio.
2	Area Fondo de Ahorro	<ul style="list-style-type: none"> - Se registra y devuelve copia al trabajador debidamente requisitada. - Se analizan los datos del trabajador, así como los datos y firmas de los avales en su caso. - Se cuantifica monto a autorizar del préstamo, en estricto apego al reglamento del fondo de Ahorro vigente. - Se turnan al gerente de Recursos Humanos con el Vo.Bo. del subgerente de personal y del encargado del Fondo de Ahorro para su autorización correspondiente. - Una vez aprobados todos los préstamos en base a la disponibilidad bancaria, se turnan a la gerencia de tesorería con oficio y relación.

TABLA [2.9] RELACION DE LAS GERENCIAS Y EL FONDO DE AHORRO

CONTINUA...

... CONTINUACION.

ACTIV. No.	UNIDAD RESPONSABLE	DESCRIPCION DE LA ACTIVIDAD
3	Gerencia de Tesorería	Recibe oficio y relación de préstamos sellando de recibido, para la elaboración de los cheques y pagos de los mismos, quedándose con los originales del oficio, relación y del pagaré.
4	Subgerencia de Nómina	Se turna copia de los pagarés solicitados para su codificación y descuento en la quincena inmediata, a la fecha de pago del préstamo.
5	Area Fondo de Ahorro	<p>Se archiva copia de los préstamos autorizados en el consecutivo para cualquier aclaración del trabajador, revisión por parte de Contr. Interna o comisión de vigilancia del F.A. Recibe listados del estado de cuenta para su revisión de descuentos operados en esa quincena, así como listados de movimientos del Fondo de Ahorro, retardos ausencias y bonificaciones quincenalmente para llevar un control de ingresos del capital al Fondo.</p> <p>Recibe de la subgerencia de nómina relación de trabajadores que causaron baja, alta o P.S.G.S. En bajas se analiza su estado de cuenta personal para determinar su saldo en el Fondo de Ahorro, procediendo a devolver los saldos a favor o cargando al aval en caso contrario.</p> <p>En baja se hace oficio con el análisis correspondiente de devolución del saldo a favor, con el Vo. Bo. del subgerente de personal, encargado de Fondo de Ahorro y con la autorización del gerente de Recursos Humanos.</p> <p>Se turna el oficio a la gerencia de tesorería para el pago correspondiente, sellando de recibido copia del mismo.</p> <p>Se archiva la copia en el consecutivo de bajas para control y cualquier aclaración del interesado.</p>

TABLA [2.9] RELACION DE LAS GERENCIAS Y EL FONDO DE AHORRO

CONTINUA...

...CONTINUACION

ACTIV. No.	UNIDAD RESPONSABLE	DESCRIPCION DE LA ACTIVIDAD
5	Area Fondo de Ahorro	<p>En altas se revisa el estado de cuenta para cotejar que efectivamente haya sido incorporado al mismo, dando aviso a la subgerencia de nómina en caso contrario.</p> <p>En permisos sin goce de sueldo (P.S.G.S.) se revisa el estado de cuenta para verificar que el trabajador ya no aparezca en el, a partir de la fecha de inicio de la misma, haciendo la devolución de las quincenas que se hayan generado además, a través de un traspaso del F.Ahorro al programa regular CAPFCE y notificando a la subgerencia de nómina en caso no notificados a la misma.</p> <p>El traspaso se hace por medio de un oficio descriptivo de las cantidades que ingresaron al Fondo por concepto de: aportación del CAPFCE, aport. del emp., amortización, préstamo, intereses, préstamo, retardos y ausencias. Se turna el oficio a la gerencia de Tesorería para el traspaso correspondiente sellando de recibido copia del mismo.</p> <p>Se archiva la copia en el consecutivo de traspasos para control y cualquier aclaración.</p> <p>En el caso de PSGS el trabajador que tenga préstamo antes de disfrutar la licencia deberá liquidar el pago de intereses correspondientes por el tiempo que dure la misma. Procediendo a hacer una ficha de ingresos a caja en el momento en que el trabajador pase a pagar el importe correspondiente.</p> <p>Se elabora la ficha de caja original y dos copias con el artículo correspondiente de intereses, turnándola a la caja general con el Vo. Bo. del subgerente de personal, encargado del fondo de ahorro y la automatización del gerente de Recursos Humanos.</p>

TABLA [2.9] RELACION DE LAS GERENCIAS Y EL FONDO DE AHORRO

CONTINUA...

... CONTINUACION

ACTIV. No.	UNIDAD RESPONSABLE	DESCRIPCION DE LA ACTIVIDAD
5	Area Fondo de Ahorro	El cajero general se encarga de hacer el depósito correspondiente, devolviéndonos copia de nuestra ficha de caja conjuntamente con la copia de la ficha de ingreso del cajero general. Se archiva la copia en el consecutivo de fichas de caja para control y cualquier aclaración posterior.

TABLA [2.9] RELACION DE LAS GERENCIAS Y EL FONDO DE AHORRO

II.7 DICCIONARIO DE DATOS.

- Antigüedad en el gobierno federal: Indica la antigüedad del empleado trabajando para el Gobierno Federal.
- Anuencia: Oficio del Gerente o Jefe de Zona que autoriza el pago de un incidente registrado en la tarjeta de asistencia.
- Año de proceso: Es el año en el que se procesa la nómina.
- Apellido materno: Es el apellido materno del empleado.
- Apellido paterno: Es el apellido paterno del empleado.
- Aporta Fondo de Ahorro: Para los empleados de mandos medios y superiores es la base sobre la cual aportan al Fondo de Ahorro.
- Aporta FOVISSSTE n: Donde n=1,2,3,4; es la aportación quincenal al FOVISSSTE por el empleado, son 4 aportaciones ya que la entrega de este al banco es bimestral.
- Aporta SAR n: Donde n=1,2,3,4; es la aportación quincenal al SAR por el empleado, son 4 aportaciones ya que la entrega de este al banco es bimestral.
- Area: La plaza contiene también el área a la cual pertenece el puesto del empleado.
- Calle y número: Calle y número interior y/o exterior del domicilio del empleado.

- Cantidad: Se da en días, horas, minutos, etc., se utiliza para costear la percepción o la deducción.
- Cantidad incidente: Es la cantidad en días, horas, minutos con las que se registra un incidente.
- Cantidad PJA: Es la cantidad en días, horas, minutos con las que se registra un PJA.
- Categoría plaza: Un mismo puesto puede tener más de una categoría, estas pueden ser A, B, C, etc., la categoría A es la del sueldo más alto y en adelante va decreciendo.
- Causa de suspensión: Causas por las cuales el empleado es suspendido, estas pueden ser por terminación de contrato eventual, por incapacidad o por suspensión.
- Clave dependiente: Es una clave que se le asigna al dependiente económico del empleado ejemplo H para hijo.
- Código postal: Código postal del domicilio del empleado.
- Colonia: Colonia en la cual reside el empleado.
- Condición: Son condiciones con las cuales el empleado debe cumplir en algunas ocasiones para que se le pague una percepción.
- Condición deducción: Son condiciones con las que si el empleado cumple se le aplica un descuento.
- Condición general: Algunas condiciones son generales y se cumple para todos los empleados, pero estas solo se cumplen en ciertas quincenas específicas.
- Condición gral deducción: Son condiciones que se cumplen en una qna específica para todos los empleados, se aplica un descuento.
- Control de asistencia: Indica si el empleado tiene que checar tarjeta de asistencia y si tiene que sujetarse a los procedimientos de control de asistencia, empleados de base y confianza.
- Consecutivo contrato: Indica el número de contratos consecutivos que ha tenido un contrato eventual.
- Costo: Es el costo de la orden de percepción o deducción.
- Costo quinquenio: Es lo que se le paga a un empleado dependiendo del número de quinquenios que tenga.

SINRECH

- Cuota fija: Es la cuota fija de subsidio que asigna la SHCP para el empleado que caiga dentro de algún nivel de aplicación.
- Cuota fija crédito: Es la cuota fija de crédito al salario que asigna la SHCP para el empleado que caiga dentro de algún nivel de aplicación.
- Cuota fija ISPT quincenal: Es la cuota fija del ISPT quincenal que asigna la SHCP para el empleado que caiga dentro de algún nivel de aplicación.
- Dedución: Es un descuento que se le hace al empleado.
- Definitiva: Es un estado de la plaza, que indica que la plaza puede ser ocupada en forma definitiva.
- Disponible: Es un estado de la plaza, que indica que la plaza puede ser ocupada en cualquier momento ya que esta disponible.
- Estado: Es el estado en el cual se encuentra el domicilio del empleado.
- Estado civil: Es el estado civil del empleado, puede ser soltero, casado, viudo, divorciado, etc.
- Factores de costeo Fn: Son los factores numéricos que en conjunción con los factores Fnl nos dan el costo total de una orden de percepción.
- Factores de costeo Fnl: Donde $n = 1, 2, 3$, los factores pueden ser V, C, R donde V indica sueldo o costo de la orden, C indica constante y R indica rutina.
- Fecha actualización (Plaza): Es la fecha en que se actualizan los datos correspondientes a la plaza.
- Fecha de creación de la plaza: Es la fecha en que se crea la plaza.
- Fecha de base: Es la fecha en la cual el empleado deja de ser eventual o interino y adquiere su plaza en forma definitiva.
- Fecha de derechos laborables: Es la fecha que se toma de base para calcular la antigüedad de un empleado tomando en cuenta cuando adquiere sus derechos laborales.
- Fecha de ingreso: Es la fecha en la cual el empleado comienza a

laborar dentro del Comité.

- Fecha de inicio de beca: Es la fecha en que se comienza a pagar la beca.
- Fecha de inicio del contrato eventual: Es la fecha en que inicia el contrato eventual de un empleado.
- Fecha de inicio de un permiso sin goce de sueldo(PSGS): Es la fecha en que el empleado inicia un permiso sin goce de sueldo.
- Fecha de nacimiento: Es la fecha de nacimiento del dependiente económico.
- Fecha de reinstalación: Es la fecha en la cual el empleado es reinstalado después de haber sido suspendido.
- Fecha de suspensión: Es la fecha en la cual el empleado debe comenzar a cumplir una suspensión.
- Fecha de terminación de la beca: Es la fecha en que se termina la beca.
- Fecha de terminación de un contrato eventual: Es la fecha en la que termina el contrato eventual.
- Fecha de terminación de un PSGS: Es la fecha en que termina el PSGS del empleado, debiendo este incorporarse a sus labores al siguiente día hábil en que termina el permiso.
- FONAC: Indica si el empleado aporta o no aporta al FONAC, solo empleados de base y confianza.
- Fondo de Ahorro: Indica si el empleado pertenece al Fondo de Ahorro 'A' o 'B'.
- Función: Es la función que desempeña el área correspondiente.
- Grupo contable: Es el grupo contable al que pertenece el JOG.
- Grupo presupuestal: Es el grupo presupuestal al que pertenece el JOG.
- Incidente: Es el evento del registro de entrada y/o salida diferentes de la norma de registro en las condiciones generales de trabajo.
- JOG: Jefatura de Zona o Gerencia.
- Justificación: Presentación posterior al documento expedido por

SINRECH

el médico del ISSSTE, o la aceptación de un Gerente o Jefe de Zona que corresponda a un hecho previamente registrado.

- Límite inferior del crédito al salario: Es el límite inferior que marca la SHCP para aplicar una tarifa del crédito al salario al empleado que caiga dentro de los límites.

- Límite inferior ISPT mensual: Es el límite inferior que marca la SHCP para aplicar el impuesto al empleado que caiga dentro de los límites.

- Límite inferior subsidio mensual: Es el límite inferior que marca la SHCP para aplicar una tarifa de subsidio al empleado que caiga dentro de los límites.

- Límite superior del crédito al salario: Es el límite superior que marca la SHCP para aplicar una tarifa del crédito al salario al empleado que caiga dentro de los límites.

- Límite superior ISPT mensual: Es el límite superior que marca la SHCP para aplicar el impuesto al empleado que caiga dentro de los límites.

- Límite superior subsidio mensual: Es el límite superior que marca la SHCP para aplicar una tarifa de subsidio al empleado que caiga dentro de los límites.

- Localidad: Es la localidad del domicilio del empleado, en el Distrito Federal corresponde a la Delegación Política.

- Nacionalidad: Es la nacionalidad del empleado.

- Nombres: Es el nombre o nombres del empleado.

- Nombre del dependiente: Es el nombre completo del dependiente económico.

- Número de año: Es el año en que se aplican las percepciones o deducciones.

- Número de área: Es la clave asignada a un área.

- Número de beca: Un empleado puede tener más de una beca asignada.

- Número de empleado: Es el número de control que se asigna a un empleado cuando comienza a trabajar en el Comité.

- Número de folio: Cuando se genera una percepción se le asigna un

número consecutivo, este número es tanto para percepciones como para deducciones.

- Número de percepción: Es un número consecutivo el cual identifica cada una de las percepciones.

- Número de deducción: Es un número consecutivo el cual identifica cada una de las deducciones.

- Número del ISSSTE: Es el número de filiación que el ISSSTE le asigna al empleado cuando este se da de alta.

- Número de JOG: Es el número que identifica a las Jefaturas de Zona y Gerencias.

- Número de quincena: Una vez que se pagaron las percepciones y se dedujeron las deducciones se guardan y se les pone el número de quincena en que se aplicaron.

- Número de quinquenio: Es el número de quinquenios que tiene un empleado.

- Número de puesto: Es el identificador que se le asigna a un puesto.

- Número de zona económica: Es el número que identifica a las zonas económicas.

- Nivel beca: Es el nivel de la beca que se le paga al empleado, puede ser M(Medio), T(Terminal), S(Superior).

- Nivel de puesto: Es el nivel que se tiene en el tabulador del CAPFCE.

- Origen deducción: Las deducciones se asignan por orden, catálogo o por programa.

- Origen percepción: Las percepciones se asignan por orden, catálogo o por programa.

- Percepción: Es una prestación que tiene el empleado.

- Permiso: Solicitud y autorización previa al hecho registrado en la tarjeta de asistencia.

- Plaza: Nos define el puesto, el lugar, y de que tipo es el nombramiento del empleado, habiendo plazas que no pueden llegar a ocuparse en forma definitiva. Dependiendo de la plaza el

nombramiento puede ser eventual, interino, provisional o definitiva.

- Porcentaje: Es el porcentaje de subsidio que tiene derecho el empleado sobre la base (salario - límite inferior), dependiendo del rango en que haya caído el empleado dentro de las tablas de subsidio de la SHCP.

- Porcentaje ISPT: Es el porcentaje de impuesto que se le descuenta al empleado sobre la base (salario - límite inferior), dependiendo del rango en que haya caído el empleado dentro de las tablas de impuesto de la SHCP.

- Porcentaje de sueldo: Es el porcentaje que el empleado percibirá del total del sueldo de la plaza.

- Profesión: Es la profesión del empleado MMYS.

- Promoción confianza: Indica si la plaza puede ser promovida a confianza.

- Promoción escalafonaría: Indica si la plaza puede ser promovida por medio del escalafón.

- Promoción medios: Indica si la plaza puede ser promovida a mandos medios.

- Propietario: Es el número del empleado que es dueño de la plaza.

- Provisional: Indica si la plaza es del tipo provisional.

- Puesto: Es el puesto que el empleado ocupa dentro del Comité; esta definido dentro de la plaza, los tipos de puesto son; base, confianza, mandos medios y superiores.

- Quincena de proceso: Es la quincena en que se procesa la nómina.

- Registro Federal de Contribuyentes: Es el RFC que asigna la SHCP al empleado.

- Salario mínimo zona económica: Es el salario mínimo que se paga en cada zona económica.

- Sexo: Es el sexo del empleado.

- Sindicalizado: Indica si el empleado es sindicalizado.

- Sueldo: Es el sueldo que esta asignado a la plaza que el empleado ocupa.

- Teléfono: Es el teléfono del domicilio del empleado.
- Tipo de área: Es el tipo de área, la cual puede ser de producción, de desarrollo, técnica, etc.
- Tipo de puesto: Puede ser de base, confianza, mandos medios o superiores.
- Veces: Es el número de veces que se paga la percepción o se descuenta la deducción.
- Veces deducidas: Es el número de veces que se ha descontado una deducción.
- Veces pagadas: Es el número de veces que se ha pagado la percepción.
- Zona económica: Es la zona económica a la cual pertenece la Jefatura de Zona o Gerencia.

II.8 NIVELES Y CONSECUENCIAS DE LOS CAMBIOS QUE IMPLICARA EL NUEVO SISTEMA.

Niveles de información.

- Cada sistema realiza sus propias funciones, pero intervienen con las funciones de otros sistemas en forma indirecta.
- Cada sistema solo puede actualizar los catálogos y tablas que le corresponden.
- Las percepciones y deducciones así como el control de asistencia tienen interacción con el usuario, por lo que las áreas encargadas del control de estos deben tener cuidado al realizar sus actualizaciones.
- Para la generación de nómina el orden en que se operan los sistemas es importante, ya que si no se han generado las percepciones no se pueden generar las deducciones, esto es solo un ejemplo.
- Los módulos que se generan después de la nómina son SAR, FOVISSSTE, ISSSTE, FONAC, ASEGURADORAS, por mencionar algunas.

CONSECUENCIAS.

Desventajas:

- El error producido en algún sistema sería acarreado a todos los demás, en forma de cascada.

Ventajas:

- Los resultados finales para las salidas que se manejan, serán las mismas, al provenir de la misma fuente de información.

- La distribución de trabajo de registro de información se reduce al no ser necesario que la información sea duplicada para diferentes procesos que se llevan a cabo, esto es, la información captada para el módulo de transacciones podrá ser utilizada por los otros sistemas y la información restante la captaran diferentes áreas.

- La consistencia e integridad de la información será considerablemente alta, lo cual permitirá tomar decisiones con toda confiabilidad y seguridad.

III - DISEÑO DEL ESQUEMA DE SOPORTE DE LAS APLICACIONES (BACK - END)

III.1 DISEÑO LOGICO

Es importante primeramente dar la definición de lo que es el diseño lógico:

Definición:

Es un proceso que consiste en la definición de columnas para agruparlas en tablas, este a su vez es independiente de la estructura física de la base de datos y es independiente de la forma en que el usuario ve los datos finalmente.

Un buen diseño debe soportar:

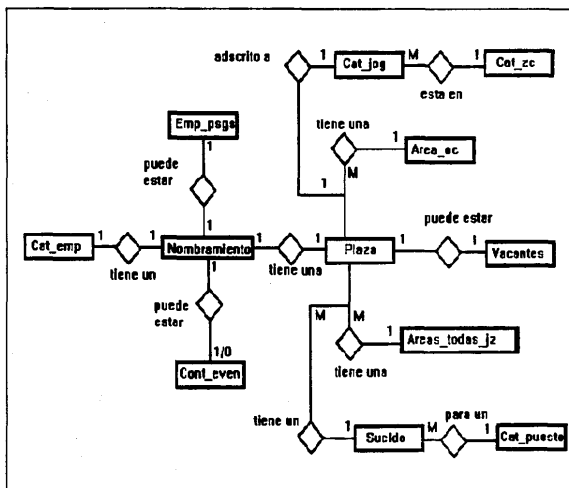
- Consistencia en la base de datos.
- Facilidad de explotación de la información.
- Flexibilidad a cambios.
- Fácil comprensión.

III.1.1 DIAGRAMA ENTIDAD RELACION (DER).

El diagrama entidad relación permite diseñar un fenómeno real para un sistema de base de datos, a través de diagramas con entidades y sus relaciones.

Una entidad es un objeto que contiene ciertos atributos que lo caracterizan, como son: debe tener existencia propia, debe ser un concepto genérico y debe contar con propiedades de interés.

Un atributo es un valor descriptivo o propiedad asociada a una entidad, sus características son: valor-asociado, rango de validación y estructura de datos asociado.



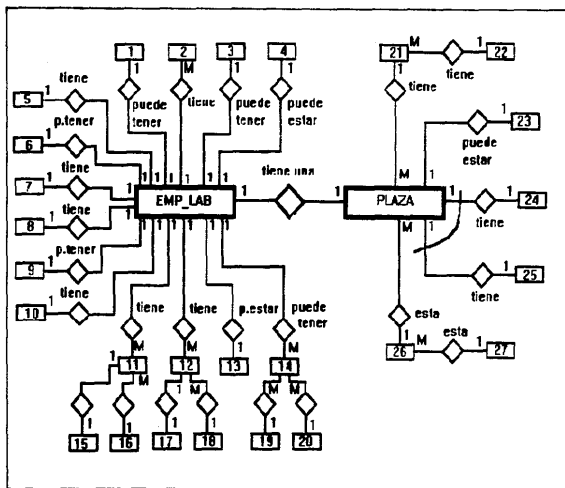
Una relación es una asociación entre entidades, es el número de ocurrencias que pueden enlazarse de una entidad con otra.

Para obtener nuestro DER actual, hubo necesidad de partir del DER que se muestra en la figura [3.1], para la nueva versión de SINRECH en donde ya se contemplan los sistemas de Control de Asistencia, Percepciones, Deducciones y el Fondo de Ahorro. Por otro lado las entidades Cat_emp y Nombramiento las transformamos en Emp_personal (datos personales del empleado) y Emp_lab (datos laborales del empleado) esto se hace por que en la versión 2.0 cuando un empleado se daba de baja se borra de la entidad nombramiento, quedando muchos de sus datos personales y laborales en Cat_emp, pero no todos los datos del empleado para la nueva versión serán dados de baja de todo el esquema, ya que se pretende guardar sus datos principales en la entidad Emp_bajas que será un histórico.

De lo anterior se tiene entonces que el DER para el sistema actual queda tal como se muestra en la figura [3.2], en donde los números que significan las entidades son:

- | | |
|-----------------------|---------------------|
| 1.- Cont_even | 15.- Deducción_gna |
| 2.- Dep_económicos | 16.- Cat_deducción |
| 3.- Emp_psgs | 17.- Percepción_gna |
| 4.- Emp_suspen | 18.- Cat_percepción |
| 5.- Emp_personal | 19.- Cat_incidente |
| 6.- Quinquenios | 20.- Cat_pja |
| 7.- FOVISSSTE | 21.- Sueldo |
| 8.- SAR | 22.- Cat_puesto |
| 9.- Fondo de ahorro | 23.- Vacantes |
| 10.- Porcen_trab | 24.- Area_oc |
| 11.- Orden_deducción | 25.- Area_todas_jz |
| 12.- Orden_percepción | 26.- Cat_jog |
| 13.- Emp_wkend | 27.- Cat_ze |
| 14.- Ctrl_asis | |

FIGURA 3.2 OBRAS SINERCE VERSION ACTUAL.



III.1.2 DIAGRAMA DE TABLAS.

En el diagrama de tablas se muestra el conjunto de las tablas de todos los sistemas que participan en SINRECH. Cada tabla está compuesta por los campos, indicando mediante el remarcado a la llave primaria, y en la parte superior de la tabla se indica el nombre de la misma.

NOTACION EMPLEADA.

La relación entre tablas se representa mediante una línea continua. El diagrama de tablas que se obtuvo se puede observar en la figura [3.3], el cual se consiguió partiendo del DER, de la figura [3.2], para saber cuales campos están relacionados con cada entidad.

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

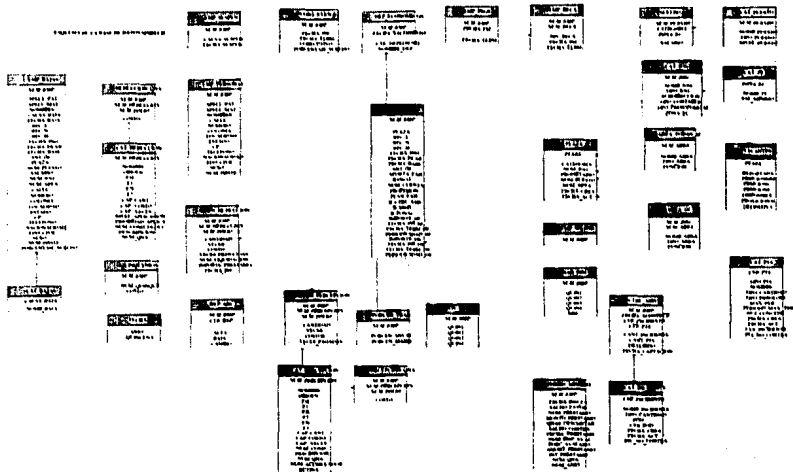


FIGURA [3.3] PRIMER DIAGRAMA DE TABLAS

III.1.3 DETERMINACION DE INDICES.

INDICE: Un indice es, un archivo donde cada entrada (registro) se compone de un valor del dato junto con uno o más apuntadores. El valor del dato es un valor para cierto campo del archivo indicado (el campo indicado), y los apuntadores identifican registros en el archivo indicado que tengan ese valor para ese campo. Un indice puede usarse de dos maneras. Primera, puede utilizarse para acceso secuencial al archivo indicado (en otras palabras, éste impone un ordenamiento sobre ese archivo indicado.) En segundo lugar, puede emplearse también para acceso directo a registros individuales en el archivo indicado sobre la base de un valor específico para ese mismo campo.

LLAVE: Es un atributo o conjunto de atributos que hacen distinguibles a una entidad dentro de un conjunto de entidades o a una relación dentro de un conjunto de relaciones.

LLAVE CANDIDATO(A): Es aquella que cumple con dos condiciones, unicidad y minimización.

LLAVE PRIMARIA: Es una llave candidato(a) elegida de manera arbitraria como identificador de una tabla.

LLAVE FORANEA: Conjunto de uno o más atributos que es llave primaria de otra relación o tabla y que sirve de acceso o conexión con otra relación o tabla.

En la siguiente tabla tenemos como están compuestas las llaves primarias y foráneas de las diferentes tablas que componen el sistema (ver tabla [3.1]).

ENTIDAD	ATRIBUTOS PRINCIPALES	LLAVES PRIMARIAS		LLAVES FORANEAS
		1a	2a	
Emp_lab	Num_emp	si	no	si
	Plaza	no	no	si
Emp_personal	Num_emp	no	no	si

TABLA [3.1] LLAVES PRIMARIAS Y FORANEAS

...CONTINUACION

ENTIDAD	ATRIBUTOS PRINCIPALES	LLAVES PRIMARIAS		LLAVES FORANEAS
		1a	2a	
Emp_psgs	Num_emp	si	no	si
Dep_económicos	Num_emp Fecha_nacimiento Cve_dependiente	no no no	no no no	si si si
Cont_event	Num_emp	si	no	si
Emp_susp	Num_emp	si	no	si
Orden_deducción	Num_emp Num_deducción Num_folio	si si si	no no no	si si no
Deducción_qna	Num_emp Num_deducción Num_folio	si si si	no no no	si si si
Cat_deducción	Num_deducción	si	no	si
Orden_percep- ción	Num_emp Num_percepción Num_folio	si si si	no no no	si si no
Percepción_qna	Num_emp Num_percepción Num_folio	si si si	no no no	si si si
Cat_percepción	Num_percepción	si	no	si
Quinquenio	Num_emp Num_quinque	si no	no no	no si
Sar_bim	Num_emp	si	no	si
Fov_bim	Num_emp	si	no	si
Porcen_trab	Num_emp	si	no	si
Fondo_ahorro	Num_emp Num_préstamo Num_emp_aval Num_qna Num_anio	si no no no no	no no no no no	si si si si si
Emp_wkend	Num_emp	si	no	si

TABLA [3.1] LLAVES PRIMARIAS Y FORANEAS

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

...CONTINUACION

ENTIDAD	ATRIBUTOS PRINCIPALES	LLAVES PRIMARIAS		LLAVES FORANEAS
		1a	2a	
Ctrl_asis	Num_emp Fecha_acontece Cve_incidente Cve_pja	si si si si	no no no no	si no si si
Cat_pja	cve_pja	si	no	si
Cat_incidente	cve_incidente	si	no	si
Plaza	Plaza Num_puesto Categoría Num_jog Num_área	si no no no no	no no no no no	si si si si si
Sueldo	Num_puesto Categoría Zona_ec	si si si	no no no	si no si
Cat_puesto	Num_puesto	si	no	si
Cat_jog	Num_jog Zona ec	si no	no no	si si
Cat ze	Zona ec	si	no	si
Vacantes	Plaza	si	no	si
Area_oc	Num_jog Num_área	si si	no no	si si
Area_todas_jz	Num_área	si	no	si

TABLA [3.1] LLAVES PRIMARIAS Y FORANEAS

III.1.4 NORMALIZACION.

Definición:

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

Beneficios:

- Bases de datos más pequeñas, lo que hace más rápida la entrada/salida.
- Las actualizaciones en la información son más rápidas.
- Las tablas y renglones son más pequeños.

Desventajas:

- Existe un gran número de tablas.
- Incrementa el número de joins.
- No permite sumarizar datos cuando se necesitan.

La teoría de la normalización esta basada en el concepto de formas normales. Se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones; por ejemplo, se dice que una relación está en primera forma normal (que se abrevia 1FN) si y sólo si satisface la restricción de contener únicamente valores atómicos (por tanto, toda relación normalizada está en 1FN, lo que explica el término <<primera>>).

Se han definido numerosas formas normales (véase fig.[3.4]). Codd definió originalmente la primera, la segunda y tercera formas normales (1FN, 2FN y 3FN). Ahora bien, se dice que todas las relaciones normalizadas están en 1FN; algunas relaciones en 1FN también están en 2FN; y algunas relaciones en 2FN también están en 3FN. La motivación subyacente a la definición era que la 2FN resultaba <<más deseable>> que la 1FN y la 3FN, a su vez más deseable que la 2FN. Es decir, el diseñador, por lo general, debe escoger relaciones en 1FN o en 2FN; aunque esta proposición no debe interpretarse como ley, el único requisito riguroso es que las relaciones estén al menos en primera forma normal.

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

Actualmente existe una 3FN nueva, llamada en ocasiones Forma Normal de Boyce/Codd (FNBC) para distinguirla de la forma antigua. Más tarde Fagin definió una <<cuarta>> forma normal (4FN) nueva y, más recientemente, otra forma normal que llamó <<forma normal de proyección-reunión>> (FN/PR, también conocida como 5FN). Como lo indica la figura [3.4], algunas relaciones en FNBC también están en 4FN, y algunas relaciones en 4FN también están en 5FN.¹

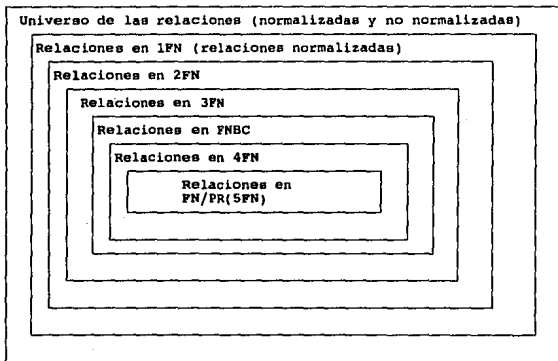


FIG. [3.4] FORMAS NORMALES

A continuación procedemos a definir las principales formas de normalización existentes, únicamente para tenerlas como información, de tal forma que si se quiere profundizar en estos

¹ En esta parte del capítulo únicamente se pretende dar a conocer las formas normales más conocidas que existen, para que de esta manera lo podamos enfocar a lo que nos interesa. Para una mayor referencia sobre estas formas normales se sugiere consultar el capítulo 14 del libro Introducción a los sistemas de bases de datos de C.J. Date, editorial Addison-Wesley iberoamericana.

temas refiérase a la bibliografía correspondiente.²

DEPENDENCIA FUNCIONAL.

Definición: Dada una relación R, el atributo Y de R es funcionalmente dependiente del atributo X de R si y sólo si cada valor de X en R tiene asociado a él exactamente un valor de Y en R (en cualquier instante).

Nótese que en la definición de dependencia funcional no existe el requisito de que un valor dado de X aparezca sólo si, siempre que dos tuplas de R coincidan en sus valores de X, también coincidan en sus valores de Y.

Para ejemplificar la forma en que se llevo a cabo la normalización se tomó la parte correspondiente al Fondo de ahorro:

*Una relación R está en 1FN si y sólo si todos los dominios subyacentes sólo contienen valores atómicos; es decir que cualquier relación normalizada está en 1FN.

Antes de nada se tiene la relación no normalizada, que es de la cual se parte:

Fondo ahorro
Num emp
Fecha_ing_fa
Saldo_favor
Num_préstamo
Monto_préstamo
Qnas_descontar
saldo_contra
Fecha_préstamo
Num_emp_aval
Porc_avalado
Num_qnas
Num_anio
Amort_préstamo
Int_préstamo

FIG. [3.5] ENTIDAD FONDO_AHORRO NO NORMALIZADA

² Introducción a los Sistemas de Bases de Datos de C. J. Date. Capítulo 14. editorial: Addison-Wesley Iberoamericana.

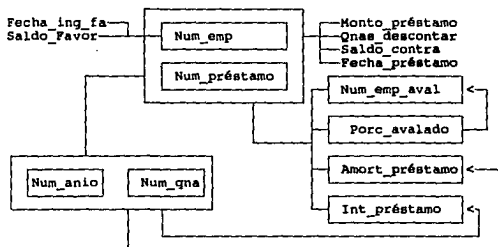


FIG. [3.6] DEPENDENCIAS FUNCIONALES DE LA ENTIDAD FONDO_AHORRO

De lo anterior tendremos la 1FN de Fondo_ahorro.

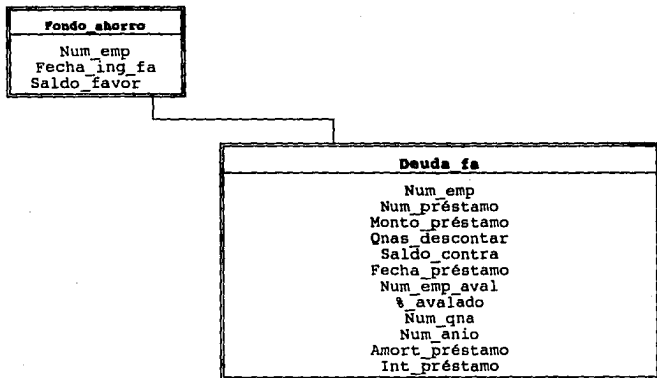


FIG. [3.7] FONDO_AHORRO EN 1FN

De aquí obtenemos la relación funcional de Deuda_fa:

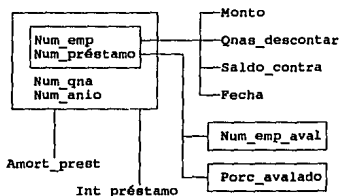


FIG. [3.8] DEPENDENCIAS FUNCIONALES DE DEUDA_FA

De lo anterior tenemos que las relaciones en 1FN quedan:

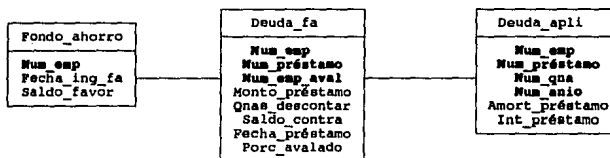


FIG. [3.9] ENTIDADES EN 1FN

Una relación R está en segunda forma normal (2FN), si y sólo si está en 1FN y cada atributo no es primo completamente dependiente de la llave primaria.

Un atributo es no primo si no participa en la llave primaria.

Si observamos el diagrama de dependencia funcional de Fondo_ahorro y deuda_apli, sus atributos no primos son completamente dependientes de la llave primaria.

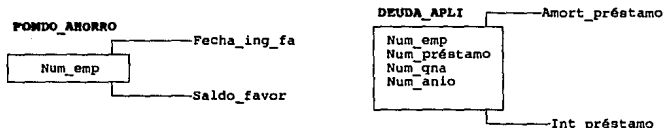


FIG. [3.10] DEPENDENCIAS FUNCIONALES DE FONDO_AHORRO Y DEUDA_APLI

El diagrama de dependencia funcional queda para deuda_fa

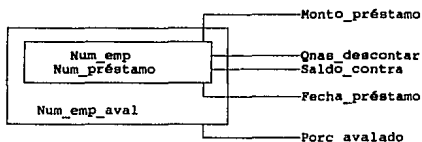


FIG. [3.11] DEPENDENCIA FUNCIONAL DE DEUDA_FA

El porcentaje avalado depende de Num_emp, Num_prestamo y Num_emp_aval. Finalmente las relaciones en 2FN quedarían:

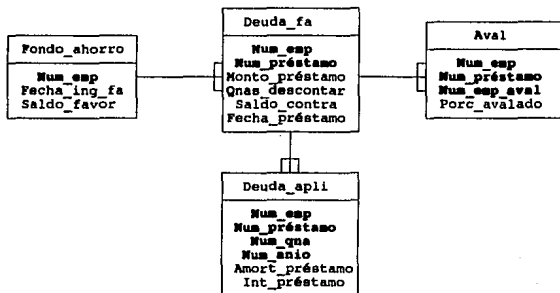


FIG. [3.12] ENTIDADES EN 2FN

Una relación R está en tercera forma normal (3FN) si y sólo si (para todo instante) cada tupla de R se compone de un valor de llave primaria que identifica alguna entidad, junto con un conjunto de valores de atributos mutuamente independientes que describen esa entidad de alguna manera.

De la definición anterior y en base a la figura [3.12] observamos lo siguiente:

- i) Las tuplas de la entidad Fondo_ahorro, están compuestas en

todo momento por una llave primaria (num_emp) que hace única a la tupla, junto con un conjunto de atributos que son mutuamente independientes (fecha_ing_fa, saldo_favor) que identifican la entidad; por lo tanto la entidad Fondo_ahorro se encuentra ya en 3FN.

ii) Para la entidad Deuda_fa la llave que describe la entidad esta formada por (num_emp, num_préstamo), esta llave compone las tuplas de la entidad en todo instante junto con un conjunto de valores de atributos mutuamente independientes (monto_préstamo, qnas_descontar, saldo_contra, fecha_préstamo), por lo que está en 3FN.

iii) De la misma forma para las entidades Aval y Deuda_apli, estas también se encuentran en 3FN.

La definición original de 3FN se sustituyó después por una definición más fuerte, que es la forma normal de Boyce/Codd (FNBC). Se acuerda llamar *determinante* a un atributo, tal vez compuesto, del cual depende funcionalmente en forma completa algún otro atributo. Entonces se define la FNBC como sigue:

Una relación R está en forma normal Boyce/Codd (FNBC) si y sólo si cada determinante es una llave candidata.

La motivación para introducir la FNBC estriba en que la definición original de 3FN no contempla satisfactoriamente el manejo de más de una llave candidata compuesta y traslapada en una relación.

Cada una de las relaciones Fondo_ahorro, Deuda_fa, Aval, Deuda_apli, que están en 3FN también lo están en FNBC, por que en cada caso la llave primaria es el único determinante en la relación.

Por otro lado tenemos que la 4FN siempre se pudo lograr, aunque no siempre es deseable en algunos casos llevar la descomposición hasta ese punto (o incluso hasta la FNBC), y por lo tanto no se llega a la 5FN.

Para nuestro caso todas las entidades del sistema se

encuentran en la forma normal de Boyce/Codd (FNBC), aunque no se asegura que todas estén en 4FN ya que para los requerimientos del sistema, se considera el grado final de descomposición la FNBC.³

III.1.5 DESNORMALIZACION.

Definición:

Es un proceso de regreso intencional de una normalización para proveer eficiencia. La normalización puede ocurrir a nivel columna o a nivel tabla. Para desnormalizar se requiere un conocimiento previo de como los datos van a ser usados.

Beneficios:

- Minimiza la necesidad de joins.
- Reduce el número de llaves foráneas.
- Reduce el número de índices.
- Puede reducir el número de tablas.
- Los agregados son precalculados.

Desventajas:

- Redundancia de datos lo que puede provocar anomalías a la hora de la inserción o borrado de renglones.
- Necesidad de mayor control en los datos volátiles.

Dado lo anterior, se podría pensar que es preferible evitar al máximo la redundancia, sin embargo, viene al caso mencionar que hasta la fecha no existe un manejador que emita con rapidez los resultados de una consulta, debido a la gran cantidad de joins que el manejador debe soportar, y a consecuencia de que se necesita eficiencia en el sistema, esto conduce a incluir columnas derivadas en las tablas, que constantemente se actualicen cada vez que se realice una inserción, baja o cambio, lo que puede provocar inconsistencia en los datos, sin embargo, si se cuenta con un manejador que provea una herramienta necesaria para estos fines,

³ Para el lector que quiera profundizar sobre las formas normales 4FN y 5FN refiérase al libro "Introducción a los sistemas de Bases de Datos" de C.J. Date edit. Addison-Wesley Iberoamericana Cap. 14 pp. 284-293.

ello se refiere a los trigger. de esta manera se mantienen consistentes los datos desde cualquier aplicación. Además algo que es muy importante, se debe contar con la suficiente experiencia en el conocimiento de los datos de cada sistema, lo que debe facilitar la definición adecuada de las columnas derivadas en cada tabla.

Nuestro sistema se desarrollara con un manejador que provee el manejo de triggers, sin embargo, en este caso no fue necesario realizar la desnormalización, por lo tanto, el diagrama de tablas que componen al sistema queda de la forma en que se muestra en la figura [3.13].

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

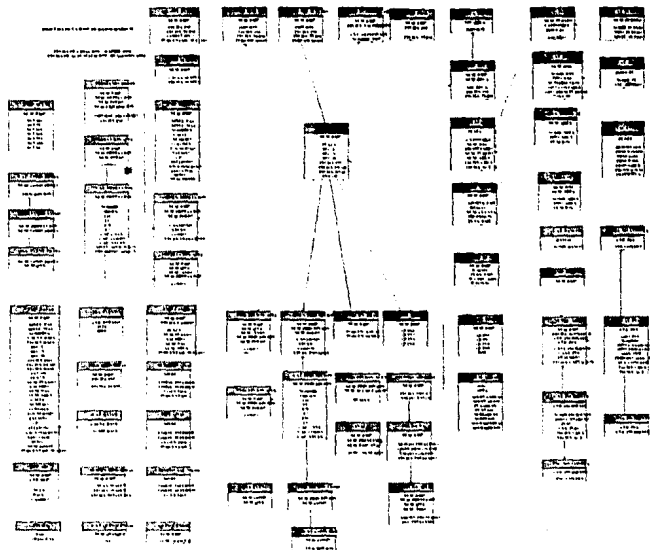


FIGURA [3.13] DIAGRAMA DE TABLAS (FINAL)

III.1.6 INTEGRIDAD DE LOS DATOS

El término integridad de datos se refiere a la corrección y completitud de la información en una base de datos.

Hasta ahora se ha considerado la llave primaria sólo desde un punto de vista formal, es decir, como identificador de tuplas en una relación, sin prestar ninguna atención a la forma de interpretar esas tuplas. Por lo general, sin embargo, esas tuplas representan entidades del mundo real, y la llave primaria, en realidad, sirve de identificador único para esas entidades. Esta interpretación conduce a imponer la regla siguiente.

Regla de integridad 1 (integridad de la entidad)

Ningún componente de un valor de una llave primaria puede ser nulo.

Las llaves primarias realizan la función de identificación única en una base de datos relacional. Un identificador que fuera totalmente nulo equivaldría a decir que hubo alguna entidad que no tuvo ninguna identificación única.

Es común que una relación incluya referencias a otra u otras; por ejemplo, la relación *Emp_beca* incluye referencias tanto a la relación *Beca_imp*te como a la relación *Emp_lab*, por medio de sus atributos *Num_emp* y *Num_beca*. Es claro que si una tupla de *Emp_beca* contiene un valor de *Num_emp*, por ejemplo 201, entonces una tupla para el empleado 201 debe existir en *Emp_lab* (de lo contrario, la tupla de *Emp_beca* se referiría, al parecer, a un empleado inexistente); y lo mismo vale para el importe de la beca (*Beca_imp*te; véase el diagrama de tablas de la figura [3.13]). Estas nociones se pueden precisar como sigue.

Se introduce la noción de *Dominio* ¹*Primario*. Un dominio específico puede designarse como *primario* si y sólo si existe alguna llave primaria de un solo atributo definida sobre ese dominio. Cualquier relación que incluya un atributo que se defina sobre un dominio *primario* debe obedecer la restricción siguiente.

Regla de integridad 2 (Integridad de referencia)

Sea D un dominio primario, y sea R_1 una relación con un atributo A que se define sobre D . Entonces, en cualquier instante dado, cada valor de A en R_1 debe ser o bien:

- a) Nulo
- b) igual a V ,

donde V es el valor de la llave primaria de alguna tupla de alguna relación R_2 con llave primaria definida sobre D .⁴

Integridad Referencial

Una columna (o grupo de columnas) que es usada para identificar en forma única un renglón en una tabla es llamada llave primaria de la tabla. Cuando una columna que es parte de la llave primaria existe en otra tabla está es llamada llave foránea. Al mantenimiento de la relación entre llaves primarias y llaves foráneas se le conoce como el mantenimiento de la integridad referencial en la base de datos.

Asegurando la Integridad Referencial

1.- Cuando la llave primaria de una tabla es cambiada o borrada y esta es usada como llave foránea en otra tabla, la otra tabla debe realizar también el cambio.

2.- Cuando se inserta o cambia una llave foránea, está debe ser validada respecto a la llave primaria a la cual se refiere.

El manejador SYBASE utiliza una herramienta que se llama triggers para cuidar la integridad referencial; un trigger es un tipo especial de procedimiento almacenado que es activado automáticamente por el servidor SQL cuando detecta que se esta realizando alguna de las siguientes actividades en cualquier tabla, inserción, actualización o borrado.

En la tabla {3.2} se muestra, la forma en que se implementaron los triggers para cuidar la integridad referencial.

⁴ R_1 debe de existir, por definición de dominio primario. Obsérvese también que la restricción se satisface de modo trivial si A es la llave primaria de R_1 .

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Area_Oc	num_jog debe estar en Cat_jog	num_jog y num area a borrar no deben de estar en Plaza	Actualizar en cascada Plaza en num_area	Insersión y act. en area_oc_tiu Borrado en area_oc_td
Area_todas_jz		Verificar que el área para jz no exista en Plaza		Borrado en area_todasjz_td
Beca_impte		El nivel de la beca no debe existir en Emp_beca	Actualizar niv_beca requerido en Emp_beca	Borrado en beca_impte_td y act. en beca_impte_tiu
Cat_bancos		Verificar que el banco no exista en Emp_MMYS	Actualiza banco en la tabla Emp_MMYS	Borrado en cat_bancos_td y act. en cat_bancos_tiu
Cat_jog	Asegurarse que en zona_ec exista en Cat_ze	num_jog no debe existir en Plaza ni en Area_oc	Si se act. num_jog se actualizan en cascada Plaza y Area_oc; si se act. zona_ec asegurarse que exista en Cat_ze	Borrado en Cat_jog_td; act. en cat_jog_tu; y inserción en cat_jog_tu
Cat_percepción		Se debe borrar en cascada en las tablas Percepción_cond y Percepción_acum	Se deben actualizar en cascada Percepción_acum, Orden_percepción, Percepción_cond, Percepción_apli y Percepción_qna	Borrado en Cat_percepción_td y act. en Cat_percepción_tu

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONITNUA...

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

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Cat_puesto		Verificar que el puesto a borrar no exista ni en Plaza ni en Sueldo	se actualizan en cascada Sueldo y Plaza	Borrado en cat_puesto_td y inserción y act. en Cat_puesto_tiu
Cat_ze		zona ec no debe de existir en Cat_jog, ni en Sueldo	se actualiza en cascada las tablas Cat_jog y Sueldo	Borrado en Cat_ze_td y para inserción y act. en Cat_ze_tiu
Cont_even	Aseguramos que el num_emp exista en Emp_lab.			Inserción en Cont_even_tiu
Ctrl_asis	Se debe asegurar que el empleado exista en la tabla Emp_bc y asegurarse que la clave de incidentes si no es nula exista en el catálogo de incidentes; y que si la clave de pja no es nula exista en el catálogo de Pja's.			Inserción: Ctrl_asis_ti
Dep_económicos	Asegurar que el empleado exista en la tabla de Emp_lab.			Inserción: en Emp_beca_tiu

TABLA (3.2) INTEGRIDAD REFERENCIAL

CONTINUA...

SINRECH

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Emp_bc	El empleado no debe de estar en la tabla de Emp_emys, además de verificar que éste exista sobre la tabla de Emp_lab.			Inserción en Emp_bc_tiu
Emp_beca	El empleado debe existir en la tabla Emp_lab, además de no tener más de 2 becas asignadas y que el nivel de la beca exista en la tabla de Beca_impte.			Inserción en Emp_beca_tiu
Emp_cambio	Registra la fecha cambio en Rep_abc			Inserción en Emp_cambio_ti
Emp_personal	La llave num_emp es única y debe de existir en la tabla de Emp_lab		Asegurarse que la llave num_emp no sea modificada.	Inserción en Emp_personal_tu y la actualización en Emp_personal_tu
Emp_paga	El empleado debe de existir en la tabla de Emp_lab y la llave de num_emp debe de ser única.			Actualizar en Emp_paga_tiu

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

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

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Emp_lab	El empleado no debe de existir en la tabla de Emp_bajas. La plaza debe de existir en el catálogo de plazas, debe de ser única y no estar asignada a otro empleado para proceder a insertar el empleado en la tabla de MMYS o de BC dependiendo del tipo del puesto de la tabla Cat_puesto y se hace un insert sobre Rep_abc, poniendo la alta como la fecha de ingreso del trabajador.	Se insertan los datos necesarios en las tablas siguientes antes de borrar: Emp_bajas, Rep_abc y posteriormente empieza la eliminación en cascada de las siguientes tablas: Emp_personal, Cont_even, Emp_hor_irr, Dep_económicos, Emp_paga, Emp_beca, Emp_mmys, Emp_bc, Emp_wkend, Orden_percepción, Deducción_apli, Emp_suspen, Emp_dif_sueldo, Ctrl_asia, Fondo_ahorro.	El número del empleado no se puede modificar y verificar que la plaza proporcionada exista en la tabla de catálogo de plazas y que esta sea única. Si no hubo errores se inserta en la tabla de Emp_cambios la plaza que el empleado deja y se actualiza la tabla Emp_cambio, en el porcentaje del sueldo si es que el empleado es del tipo eventual o interino.	Inserción en Emp_lab_tu, Borrar en Emp_lab_td, y actualizar en Emp_lab_tu
Emp_wkend	El empleado debe de existir en la tabla de Emp_bc.			Inserción en Emp_wkend_tiu.

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

SINRECH

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Emp_mmys	Se chequea que no existan cambios sobre el número de empleado y que éste no exista en la tabla de Emp_bc, y se debe de asegurar que el banco proporcionado exista.			Inserción en Emp_mmys_tiu
Emp_suspend	No están permitidos los cambios sobre el # de empleado, además de que el empleado debe existir en la tabla de Emp_lab. La causa de la suspensión sólo puede ser 3, 7 y 8.			Inserción en Emp_suspend_tiu
Fondo_ahorro	El empleado a dar de alta debe de estar en la tabla de Emp_lab.	Si el empleado debe al fondo de ahorro, no se puede dar de baja.		Inserción en Fondo_ahorro_ti y Borrado en Fondo_ahorro_td

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

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

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
PJA_incidente	Se chequea que la clave del incidente exista en el catálogo de incidentes, así como la clave del PJA exista en el catálogo de PJA's.			Inserción en PJA_incidente_tiu
Plaza	El # Jog debe existir en el catálogo de Jog y el # de puesto debe de existir en el catálogo de puestos y finalmente # de Área exista en el catálogo de Área_oc o en el catálogo de Área todas iz.		Es necesario actualizar la tabla de Emp_cambio.	Inserción y actualización en Plaza_tiu
Sueldo	Asegurar que la zona económica, exista en el catálogo de Zona económicas y que el puesto exista en el catálogo de puestos.		Se actualizan en cascada las tablas de Categoría y Zona_ec.	Inserción en Sueldo_ti y actualización en Sueldo_tu

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

SINRECH

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Vacantes	La plaza a insertar debe existir en la tabla de Plaza.			Insertión en vacantes_ti
PJA_no_contra	La clave pja y pja contra deben existir en Cat_pja.		El pja_contra debe existir en Cat_pja.	Insertión y actualización en pja_no_contra_tiu
Cat_pja		La clave pja no debe existir en Pja_no_contra, Ctrl_asis, Pja_incidente.		Borrado en cat_pja_td
Cat_incidente		La clave incidente no debe existir en Inc_no_contra, Ctrl_asis, pja_incidente.		Borrado en cat_incidente_td
Inc_no_contra	La clave del incidente y inc contra deben existir en Cat_incidente		El inc_contra debe existir en el Cat_incidente.	Insertión y actualización en inc_no_contra_tiu
Sar_bin	El empleado debe de existir en Emp_lab.			Insertión en sar_bin_ti

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

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

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Fov_bim	El empleado debe de existir en Emp_lab.			Inserción en fov_bim_ti
Porcen_trab	El empleado debe de existir en Emp_lab.			Inserción en porcen_trab_ti
Orden_percepción	El empleado debe de existir en Emp_lab y la percepción en Cat_percepción.			Inserción en orden_percepción_t i y actualización en orden_percepción tu
Percepción_acum	La percepción debe de existir en Cat_percepción.			Inserción en percepción_acum_ti
Deuda_fa	El empleado debe de tener fondo de ahorro.			Inserción en deuda_fa_ti
Aval	El empleado aval debe de estar en Emp_lab y debe existir el préstamo al que se hace referencia.			Inserción en aval_ti
Qna_evento	num_cond debe existir en Cat_cond_per.			Inserción en qna_even_ti

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

SINRECH

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Percepción_cond	La percepción debe de existir Cat_percepción y la condición en Cat_cond_per.			Inserción en Percepción_cond_ti
Cat_cond_per		El num_cond no debe existir en Percepción_cond ni en Qna evento.		Borrado en cat_cond_per_td
Quinquenios	El empleado debe de existir en Emp_lab y no debe de estar en Emp_rmys. El num_quingue debe existir en Cat_quinquenios.		El num_quingue debe de existir en Cat_quinquenios.	Inserción y actualización en quinquenios_tiu
Cat_quinquenios		El num_quingue no debe existir en Quinquenios.		Borrado en cat_quinquenios_td
Cat_bajas		La causa_baja no debe de existir en Emp_bajas ni en Emp_suspen.		Borrado en cat_bajas_td
Emp_bajas	La causa baja exista en cat_bajas.			Inserción en emp_bajas_ti

TABLA [3.2] INTEGRIDAD REFERENCIAL

CONTINUA...

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

...CONTINUACION

TABLA	INSERCIÓN	BORRADO	ACTUALIZAR	TRIGGERS
Cat_deducción		No deben de existir ordenes de deducción con la clave de num_deducción y se borra en cascada en Deducción_cond.	Se actualiza en Deducción_cond, en Deducción_apli, en Deducción_gna y en Orden_deducción la clave que se cambia.	Borrado esta en cat_deducción_td y actualización en cat_deducción_tu
Orden_deducción	El empleado debe de existir en Emp_lab, la deducción debe de existir en Cat_deducción.			Inserción en orden deducción_ti
Deducción_cond	El num_deducción debe existir en Cat_deducción y el num_cond_deduc debe existir en Cat_cond_deduc.			Inserción en deducción_cond_ti
Gna_event_deduc	El num_cond_deduc debe de existir en Cat_cond_deduc.			Inserción en gna_event_deduc_ti
Cat_cond_deduc		El num_cond_deduc no debe de existir en Deducción_cond ni en Gna_event_deduc.		Borrado en cat_cond_deduc_td

TABLA [3.2] INTEGRIDAD REFERENCIAL

Un trigger, por otra parte, es un tipo especial de procedimiento almacenado (store procedure), queda definido como sigue:

- Es una colección de sentencias de SQL, guardadas en la base de datos, las cuales son ejecutadas por el nombre con que fueron guardadas.
- Los procedimientos pueden aceptar y retornar parámetros, regresar valores y llamar a otros procedimientos.
- Es más rápida la ejecución de los comandos por medio de un procedimiento almacenado, que ejecutarlos interactivamente o desde un batch.

Para crear un procedimiento almacenado, se puede hacer de la siguiente manera:³

```
create procedure procedure_name
as SQL_statements
```

Ejemplo:

```
create procedure costo_quinquenio @num_quin tinyint
as
select costo from cat_quinquenios
where num_quinque = @num_quin
return
```

la forma de ejecutar el procedimiento es con la siguiente instrucción :

```
exec costo_quinquenio.
```

Entonces para crear un trigger se hace de la siguiente manera:

```
create trigger trigger_name
on table_name
for {insert|update|delete}
[, {insert|update|delete}...]
as
SQL_statements
```

³ Para mayor información acerca de los comandos de Sybase, favor de referirse al manual *commands reference*.

Ejemplo:

```

create trigger sar_bim_ti
on sar_bim
for insert
as
declare @num_renglón int
/* encuentra cuantos renglones se modifican */
select @num_renglón = @@rowcount
/* si ningún renglón fue afectado regresamos */
if @num_renglón = 0
return
/* si más de un renglón fue modificado */
if @renglón > 1
begin
print "solo se puede insertar un registro a la vez"
rollback transaction
return
end
/* verificamos que el empleado exista en Emp_lab */
if not exists(select * from emp_lab el, insert i
              where el.num_emp = i.num_emp)
begin
print "El empleado no existe"
rollback transaction
return
end

```

La integridad de la base de datos también es cuidada por medio de las reglas y los defaults que se pueden asociar a un campo.

Las reglas son rangos de valores o uno o varios valores específicos para un campo o tipo definido por el usuario.

Para crear una regla se hace mediante el comando `create rule`.

Ejemplo:

```
create rule anio
as
@anio like "[6-9][0-9]"
```

Para asignar una regla a una columna se hace con *sp_bindrule*.
sp_bindrule "anio", "percepción_apli.num_anio"

El default es un valor predefinido para la columna cuando el usuario no define la entrada a ese campo.

La forma en que se crea un default es con el comando *create default*.

Ejemplo:

```
create default no_propor
as
'NO PROPORCIONADO'
```

Para asignar el default a una columna se utiliza el comando siguiente:

```
sp_bindefault "no_propor", "emp_personal.calle"
```

En el siguiente tema, se trata el diccionario de datos, en el puede observarse que reglas y/o defaults están asociados a cada campo en cada una de las tablas que componen al sistema.

III.1.7. DICCIONARIO DE DATOS

"Contiene información de la información", es decir:

- * Ayuda a identificar y clasificar la información.
- * Son tablas con descripciones de los archivos, registros, programas y campos.

Tipos de diccionarios de datos:

- * Integrado:
 - Incluido en el manejador de bases de datos (DBMS).
 - Refleja los cambios a la Base de Datos de forma automática.

- No se puede usar con otro DBMS.

*** Stand-Alone**

- Es independiente del DBMS.

- Puede trabajar con varios DBMS.

- Requiere procesos de actualización.

- Puede ocasionar inconsistencia.

El diccionario de datos contiene información detallada acerca de los objetos que componen la base de datos tales como:

- Tablas: Las tablas son definidas usando la sentencia *create table*. En el comando se definen los siguientes puntos:

1.- Define el nombre de la tabla.

2.- Define el número y nombre de las columnas.

3.- Define el tipo de dato al que esta asignado cada columna.

4.- Define si la columna acepta o no nulos.

Ejemplo:

```
create table emp_lab
(num_emp      tipo_num_employed,
plaza        char4_num,
  rfc_l       char(4) not null,
  rfc_n       char(6) not null,
  rfc_h       char(3) null,
  fecha_ing   fecha,
  fecha_dlab  fecha,
  fecha_base  fecha)
```

- Reglas: Definidas en el tema anterior.

- Defaults: También vistos en el tema anterior.

- Tipos de datos: Existen dos tipos de datos, los tipos de datos del sistema, como son: int, smallint, float, money, text, datetime, bit, etc. y los tipos definidos por el usuario, los cuales son creados por la instrucción *sp_addtype* a estos tipos se les pueden asignar reglas y defaults directamente.

- **Indices:** Ya se definieron anteriormente. El DBMS maneja dos tipos de índices, los índices clustered y los nonclustered. Los primeros ordenan la página de datos por el orden de los índices y sólo puede haber uno por tabla. En el caso de los segundos no afectan el orden de los datos, simplemente proveen apuntadores a los renglones y puede haber hasta 249 índices nonclustered.

La sintaxis para crear un índice es la siguiente:

```
create [unique][clustered][nonclustered] index name
      on table (column,column,...)
```

Ejemplo:

```
create unique clustered index indice_lab
      on emp_lab (num_emp)
```

- **Vistas:** Una vista es una ventana con datos de una o más tablas, las vistas pueden ser tratadas como tablas.

Para crear una tabla esta puede ser descrita con una sentencia de select utilizando la siguiente estructura.

```
create view view_name
      [(column_name [,column_name]...)]
      as select_statement
```

- **Store procedure:** Fueron vistos en el tema anterior.

- **Triggers:** Tratados en el tema anterior.

A continuación presentamos el diccionario de datos del sistema, el cual esta dividido en dos partes. En la primera se presentan los tipos definidos por el usuario, y en la segunda las tablas que componen el sistema junto con las reglas y defaults de cada columna así como los triggers que cuidan la integridad referencial y los índices de cada tabla.

TIPOS

User_type	Storage_type	Length	Nulls	Default_name	Rule_name
A	char	1	0		
anio	char	2	0	def_anio	anio
booleano	bit	1	0	cero	
cadena20	char	20	0		
cadena20_null	char	20	1		
cadena60	char	60	0	no_propor	
cadena60_null	char	60	1	no_propor	
char1_alfa	char	1	0		char1_alfa
char1_num	char	1	0		char1_num
char2_alfa	char	2	0		char2_alfa
char2_num	char	2	0		char2_num
char3_alfa	char	3	0		char3_alfa
char3_num	char	3	0		char3_num
char4_num	char	4	0		char4_num
char5_num	char	5	0		char5_num
dinero	money	8	0	cero	
direccion	varchar	40	0	no_propor	
edificio	char	1	0	blanco	edificio
ep_prog	char	2	0		ep_prog
fecha	datetime	8	0		
hora	char	5	1		regla_hora
id_cons_art	char	5	0		
id_grupo_art	char	1	0		
id_grupo_guia	char	1	0		
id_obra	char	4	0		
id_opc_guia	char	3	0		
id_sub_art	char	2	0		
id_sub_guia	char	1	0		
id_tipo_art	char	1	0		tipo_art
id_tipo_guia	char	1	0		
intero	int	4	0	cero	
num_edo	char	2	0		num_edo
periodo	char	1	0		regla_periodo
sentero	smallint	2	0	cero	
telefono	varchar	12	0	no_tiene	
tentero	tinyint	1	0	cero	
tipo_cantidad	real	4	0		regla_mayorigus
tipo_cve_incide	smallint	2	0		
tipo_cve_pja	smallint	2	0		
tipo_factor	char	1	1		regla_factor
tipo_factor_1	char	1	1		regla_factor_1
tipo_num_deduc	tinyint	1	0		regla_numdeducc
tipo_num_emplea	int	4	0		
tipo_num_percep	tinyint	1	0		regla_numper
tipo_num_puesto	smallint	2	0		regla_numyacer
unidad_art	char	2	0		

TABLAS

AREA_OC

Name	Owner	Type	
area_oc	dbo	user table	
Data located on segment	When created		
default	Oct 8 1992 3:30PM		
Column name	Type	Length Nulls Default_name	Rule_name
num_jog	tentero	1 0 cero	regla_numjog
num_area	char1_num	1 0 NULL	char1_num
nomb_area	cadena60	60 0 no_propor	NULL
tipo_area	char	1 0 NULL	regla_tipoarea
funcion	char	1 0 NULL	regla_funcion

index_name: indice_area_oc
 index_description: clustered, unique located on default
 index_keys: num_jog, num_area
 triggers: area_oc_tiu inserción y actualización

AREA_TODAS_JZ

Name	Owner	Type	
area_todas_jz	dbo	user table	
Data located on segment	When created		
default	Oct 8 1992 3:30PM		
Column name	Type	Length Nulls Default_name	Rule_name
num_area	char1_num	1 0 NULL	char1_num
nomb_area	cadena60	60 0 no_propor	NULL
tipo_area	char	1 0 NULL	regla_tipoarea
funcion	char	1 0 NULL	regla_funcion

index_name: indice_area_jz
 index_description: clustered, unique located on default
 index_keys: num_area
 triggers: areas_todas_jz_td borrado

AUX_ACUM

Name	Owner	Type		
aux_acum	dbo	user table		
Data_located_on_segment	When_created			
default	Jul 8 1994 7:43PM			
Column_name	Type	Length Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4 0 NULL		NULL
acum4	dinero	8 0 NULL		NULL

AUX_DEDUCS

Name	Owner	Type		
aux_deduca	dbo	user table		
Data_located_on_segment	When_created			
default	Mar 29 1993 3:23PM			
Column_name	Type	Length Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4 0 NULL		NULL
vector	tinyInt	1 0 NULL		NULL

index_name: ind_aux_deduca
 index_description: Clustered located on default
 index_keys: num_emp

AUX_PERCEPS

Name	Owner	Type		
aux_perceps	dbo	user table		
Data_located_on_segment	When_created			
default	Mar 22 1993 12:18PM			
Column_name	Type	Length Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4 0 NULL		NULL
vector	int	4 0 NULL		NULL

index_name: ind_aux_perceps
 index_description: Clustered located on default
 index_keys: num_emp

AUXF

Name	Owner	Type			
auxf	dbo	user table			
Data_located_on_segment	When_created				
default	Jan 14 1994 8:18PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
fi	datetime	8	0	NULL	NULL
ff	datetime	8	0	NULL	NULL

AUXFSIG

Name	Owner	Type			
auxfsig	dbo	user table			
Data_located_on_segment	When_created				
default	Jan 18 1994 4:09PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
fisig	datetime	8	0	NULL	NULL
ffsig	datetime	8	0	NULL	NULL

AVAL

Name	Owner	Type			
aval	dbo	user table			
Data_located_on_segment	When_created				
default	May 2 1994 12:54PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_prestamo	varchar	8	0	NULL	NULL
num_emp_aval	tipo_num_emplea	4	0	NULL	NULL
porC_avalado	real	4	0	NULL	NULL

index_name: indice_aval
 index_description: nonclustered located on default
 index_keys: num_emp, num_prestamo, num_emp_aval
 triggers: aval_Ei inserción

BECA_IMPTE

Name	Owner	Type	
beca_imp	dbo	user table	
Data_located_on_segment	When_created		
default	Oct 8 1992 3:29PM		
Column_name	Type	Length Nulls Default_name	Rule_name
niv_beca	char	1 0 NULL	regla_nivbeca
importe	dinero	8 0 cero	NULL
index_name: indice_beca_imp			
index_description: clustered, unique located on default			
index_keys: niv_beca			
triggers: beca_imp_td borrado, beca_imp_tiu actualizacion			

CAT_BAJAS

Name	Owner	Type	
cat_bajas	dbo	user table	
Data_located_on_segment	When_created		
default	Nov 18 1992 2:15PM		
Column_name	Type	Length Nulls Default_name	Rule_name
causa_baja	char	1 0 NULL	NULL
nomb_baja	cadena20	20 0 NULL	NULL
index_keys: causa_baja			
triggers: cat_bajas_td borrado			

CAT_BANCOS

Name	Owner	Type	
cat_bancos	dbo	user table	
Data_located_on_segment	When_created		
default	Oct 8 1992 3:29PM		
Column_name	Type	Length Nulls Default_name	Rule_name
banco	char2_num	2 0 NULL	char2_num
nomb_banco	cadena60	60 0 no_propor	NULL
index_name: indice_bancos			
index_description: clustered, unique located on default			
index_keys: banco			
triggers: cat_bancos_td borrado, cat_bancos_tiu actualización			

CAT_COND_DEDUC

Name	Owner	Type			
cat_cond_deduc	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_cond_deduc	tentero	1	0	cero	regla_mayoracer
descripcion	cadena60	60	0	no_propor	NULL

index_name: indice_con_ded
 index_description: clustered, unique located on default
 index_keys: num_cond_deduc
 triggers: cat_cond_deduc_td borrado

CAT_COND_PER

Name	Owner	Type			
cat_cond_per	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_cond	tentero	1	0	cero	NULL
descripcion	cadena60	60	0	no_propor	NULL

index_name: indice_cond_per
 index_description: clustered, unique located on default
 index_keys: num_cond
 triggers: cat_cond_per_td borrado

CAT_DEDUCCION

Name	Owner	Type			
cat_deduccion	dbo	user table			
Data_located_on_segment	When_created				
default	Apr 2 1993 3:00PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num deduccion	tipo_num deducc	1	0	NULL	regla_numdeducc
nombre	cadena60	60	0	no propor	NULL
origen	char	1	0	NULL	NULL
f11	tipo_factor_1	1	1	NULL	regla_factor_1
f1	real	4	1	NULL	NULL
f21	tipo_factor_1	1	1	NULL	regla_factor_1
f2	real	4	1	NULL	NULL
cap_cant	booleano	1	0	cero	NULL
cap_costo	booleano	1	0	cero	NULL
cap_veces	booleano	1	0	cero	NULL
nivel_aplicacio	tentero	1	0	cero	NULL
prioridad_aplic	tentero	1	0	cero	NULL

index keys: num deduccion
 triggers: cat_deduccion_td borrado,
 cat_deducciob_tu actualización

CAT_INCIDENTE

Name	Owner	Type			
cat_incidente	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
cve incidente	tipo_cve_incide	2	0	NULL	NULL
nomb_incidente	cadena60	60	0	no propor	NULL
tipo_cantidad	char	1	0	NULL	NULL
pod	char	1	0	NULL	regla_pod
cve_pod	tentero	1	0	cero	NULL
fecha_crea	fecha	8	0	NULL	NULL
fecha_act	fecha	8	0	NULL	NULL

index_name: indice_incidente
 index_description: clustered, unique located on default
 index_keys: cve_incidente
 triggers: cat_incidente_td borrado

CAT_JOG

Name	Owner	Type			
cat_jog	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_jog	tentero	1	0	cero	regla_numjog
nomb_jog	cadena50	60	0	no_propor	NULL
tipo_jog	char	1	0	NULL	NULL
subdireccion	tentero	1	0	cero	NULL
gpo_contable	char	2	0	NULL	NULL
gpo_presupuesta	char	2	0	NULL	NULL
zona_ec	tentero	1	0	cero	regla_zonaec

index_name: indice_jog
 index_description: clustered, unique located on default
 index_keys: num_jog
 triggers: cat_jog_td borrado,
 cat_jog_tu actualización,
 cat_jog_ti inserción

CAT_PERCEPCION

Name	Owner	Type			
cat_percepcion	dbo	user table			
Data_located_on_segment	When_created				
default	Nov 24 1992 1:00PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_percepcion	tipo_num_percep	1	0	NULL	regla_numero
nombre	cadena50	60	0	no_propor	NULL
origen	char	1	0	NULL	regla_origenper
f11	tipo_factor	1	1	NULL	regla_factor
f1	real	4	1	NULL	NULL
f21	tipo_factor	1	1	NULL	regla_factor
f2	real	4	1	NULL	NULL
f31	tipo_factor	1	1	NULL	regla_factor
f3	real	4	1	NULL	NULL
cap_cant	booleano	1	0	cero	NULL
cap_costo	booleano	1	0	cero	NULL
cap_veces	booleano	1	0	cero	NULL

index_name: indice_percepcion
 index_description: clustered, unique located on default
 index_keys: num_percepcion
 triggers: cat_percepcion_td borrado
 cat_percepcion_tu actualización

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

CAT_PJA

Name	Owner	Type	
cat_pja	dbo	user table	
Data located on segment	When created		
default	Oct 8 1992 3:30PM		
Column_name	Type	Length Nulls Default_name	Rule_name
cve_pja	tipo_cve_pja	2 0 NULL	NULL
gpo_pja	char	1 0 NULL	NULL
nombre	cadena60	60 0 no propor	NULL
tipo_cantidad	char	1 0 NULL	NULL
tipo_desglose	char	1 0 NULL	NULL
max_per	tentero	1 0 cero	regla_max_per
periodo_max_per	char	1 1 NULL	NULL
aut_concede	char	1 0 NULL	regla_autoridad
fecha_crea	fecha	8 0 NULL	NULL
fecha_act	fecha	8 0 NULL	NULL

index name: indice_pja
 index description: clustered, unique located on default
 index keys: cve_pja
 triggers: cat_pja_td borrado

CAT_PUESTO

Name	Owner	Type	
cat_puesto	dbo	user table	
Data located on segment	When created		
default	Oct 8 1992 3:30PM		
Column_name	Type	Length Nulls Default_name	Rule_name
num_puesto	tipo_num_puesto	2 0 NULL	regla_mayoracer
nombr_puesto	cadena60	60 0 no propor	NULL
tipo_puesto	char	1 0 NULL	regla_tipopues
nivel_puesto	char	1 0 NULL	regla_nivelpues

index name: indice_puesto
 index description: clustered, unique located on default
 index keys: num_puesto
 triggers: cat_puesto_td borrado,
 cat_puesto_tiu inserción y actualización

CAT_QUINQUENIOS

Name	Owner	Type			
cat_quinquenios	dbo	user table			
Data_located_on_segment	When_created				
default	Jan 20 1993 12:31PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_quinque	tinyint	1	0	NULL	NULL
costo	dinero	8	0	cero	NULL

index_name: ind_cat_quin
 index_description: nonclustered located on default
 index_keys: num_quinque
 triggers: cat_quinquenios_td borrado

CAT_ZE

Name	Owner	Type			
cat_ze	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
zona_ec	tentero	1	0	cero	regla_zonasc
nomb_ze	cadena60	60	0	no_propor	NULL
sal_minimo	dinero	8	0	cero	NULL

index_name: indice_ze
 index_description: clustered, unique located on default
 index_keys: zona_ec
 triggers: cat_ze_td borrado
 cat_ze_tiu inserción y actualización

CONT_EVEN

Name	Owner	Type
cont_even	dbo	user table

Data_located_on_segment	When_created
default	Oct 8 1992 3:29PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
fecha_ini	fecha	8	0	NULL	NULL
fecha_term	fecha	8	0	NULL	NULL
consecutivo	tentero	1	0	cero	regla_consecuti
porcentaje_suel	real	4	0	NULL	NULL

index_name: indice_eventual

index_description: clustered, unique located on default

index_keys: num_emp

triggers: cont_even_tiu inserción

COS_BONIFIC

Name	Owner	Type
cos_bonific	dbo	user table

Data_located_on_segment	When_created
default	Jul 12 1994 4:05PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_jog	tentero	1	0	cero	NULL
num_emp	int	4	0	NULL	NULL
importe	int	4	0	NULL	NULL

index_name: ind_costo

index_description: nonclustered, unique located on default

index_keys: num_jog, num_emp

COS_EXTRAS

Name	Owner	Type
cos_extras	dbo	user table

Data_located_on_segment	When_created
default	Jul 13 1994 2:48PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_jog	tentero	1	0	cero	NULL
num_emp	int	4	0	NULL	NULL
horas_ext	float	8	0	NULL	NULL
importe	float	8	0	NULL	NULL

index_name: ind_costo

index_description: nonclustered, unique located on default

index_keys: num_jog, num_emp

CREDITO_SAL_MENS

Name	Owner	Type
credito_sal_mens	dbo	user table

Data_located_on_segment	When_created
default	Feb 8 1994 2:02PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
nivel	tentero	1	0	cero	NULL
limite_inferior	dinero	8	0	cero	NULL
limite_superior	dinero	8	0	cero	NULL
cuota_fija	dinero	8	0	cero	NULL

index_name: indice_credito

index_description: clustered, unique located on default

index_keys: nivel

CTRL_ASIS

Name	Owner	Type
ctrl_asis	dbo	user table

Data_located_on_segment	When_created
default	Feb 17 1993 1:08PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
fecha_acontece	fecha	8	0	NULL	NULL
cve_incidente	tipo_cve_incide	2	1	NULL	NULL
cve_pja	tipo_cve_pja	2	1	NULL	NULL
cant_incidente	tipo_cantidad	4	0	NULL	regla_mayorigua
cant_pja	tipo_cantidad	4	0	NULL	regla_mayorigua
detenido	booleano	1	0	cero	NULL
fecha_captacion	fecha	8	0	NULL	NULL

index_name: ind_ctrl_inc
 index_description: nonclustered located on default
 index_keys: num_emp, fecha_acontece, cve_incidente

index_name: ind_ctrl_pja
 index_description: nonclustered located on default
 index_keys: num_emp, fecha_acontece, cve_pja

trigger: ctrl_asis_ti inserción

DEDUCCION_APLI

Name	Owner	Type
deduccion_apli	dbo	user table

Data_located_on_segment	When_created
default	Jan 14 1994 1:39PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_qna	tentero	1	0	cero	regla_quincena
num_anio	anio	2	0	def_anio	anio
num_deduccion	tipo_num_deduc	1	0	NULL	regla_numdeducc
num_folio	int	4	0	NULL	NULL
costo	dinero	8	0	cero	NULL

index_name: ind_dedu_apli
 index_description: nonclustered located on default
 index_keys: num_emp, num_qna, num_anio, num_deduccion, num_folio

DEDUCCION_COND

Name	Owner	Type			
deduccion_cond	dbo	user table			
Data located on segment	When created				
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_deduccion	tipo_num_deduc	1	0	NULL	regla_numdeducc
num_cond_deduc	ten_tero	1	0	cero	regla_mayoracer
index_name: indice_deduccion_cond					
index_description: clustered, unique located on default					
index_keys: num_deduccion, num_cond_deduc					
triggers: deduccion_cond_ti inserción					

DEDUCCION_GNA

Name	Owner	Type			
deduccion_gna	dbo	user table			
Data located on segment	When created				
default	Jan 14 1994 1:45PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_deduccion	tipo_num_deduc	1	0	NULL	regla_numdeducc
num_folio	int	4	0	NULL	NULL
costo	dinero	8	0	cero	NULL
index_name: ind_deduc_gna					
index_description: nonClustered located on default					
index_keys: num_emp, num_deduccion, num_folio					

DEP_ECONOMICOS

Name	Owner	Type			
dep_economicos	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
fecha_nacimient	fecha	8	0	NULL	NULL
cve_dependientes	char	1	0	NULL	regla_dependien

index_name: indice_economicos

index_description: clustered, unique located on default

index_keys: num_emp, fecha_nacimiento

triggers: dep_economicos_tiu inserción

DEUDA_APLI

Name	Owner	Type			
deuda_apli	dbo	user table			
Data_located_on_segment	When_created				
default	May 2 1994 12:54PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_prestamo	varchar	8	0	NULL	NULL
num_qna	tentero	1	0	cero	NULL
num_anio	anio	2	0	def_anio	anio
amort_prestamo	dinero	8	0	cero	NULL
int_prestamo	dinero	8	0	cero	NULL

index_name: indice_deuda_apli

index_description: nonclustered located on default

index_keys: num_emp, num_prestamo, num_qna, num_anio

DEUDA_FA

Name	Owner	Type
deuda_fa	dbo	user table
Data_located_on_segment	When_created	
default	May 2 1994 12:54PM	

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
num_prestamo	varchar	8	0 NULL		NULL
monto_prestamo	dinero	8	0 cero		NULL
qnas_descontar	int	4	0 NULL		NULL
saldo_contra	dinero	8	0 cero		NULL
fecha_prestamo	fecha	8	0 NULL		NULL

index_name: indice deuda
 index_description: nonclustered located on default
 index_keys: num_emp, num_prestamo
 triggers: deuda_fa_ti inserción

DIA_EVENTO

Name	Owner	Type
dia_evento	dbo	user table
Data_located_on_segment	When_created	
default	Oct 8 1992 3:30PM	

Column_name	Type	Length	Nulls	Default_name	Rule_name
cve_evento	char	2	0 NULL		NULL
dia	tentero	1	0 cero		regla_dia
mes	tentero	1	0 cero		regla_mes

index_name: indice dia evento
 index_description: clustered, unique located on default
 index_keys: dia, mes, cve_evento

EMP_ACUM

Name	Owner	Type
emp_acum	dbo	user table
Data_located_on_segment	When_created	
default	Jul 21 1993 2:38PM	

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
acum1	dinero	8	0	cero	NULL
acum2	dinero	8	0	cero	NULL
acum3	dinero	8	0	cero	NULL
acum4	dinero	8	0	cero	NULL
acum5	dinero	8	0	cero	NULL
acum6	dinero	8	0	cero	NULL
acum7	dinero	8	1	cero	NULL
acum8	dinero	8	1	cero	NULL

index_name: indice_acum
 index_description: nonclustered, unique located on default
 index_keys: num_emp

EMP_BC

Name	Owner	Type
emp_bc	dbo	user table
Data_located_on_segment	When_created	
default	Oct 8 1992 3:29PM	

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
b_sind	booleano	1	0	cero	NULL
plan_fah	char	1	1	NULL	regla_plan_fah
b_ctFl_asis	booleano	1	0	cero	NULL
b_fonaC	booleano	1	0	cero	NULL

index_name: indice_bc
 index_description: clustered, unique located on default
 index_keys: num_emp
 triggers: emp_bc_tiu inserción

EMP_BAJAS

Name	Owner	Type
emp_bajas	dbo	user table

Data located on segment	When created
default	Feb 1 1994 2:41PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
apell_pat	cadena20	20	0 NULL		NULL
apell_mat	cadena20_null	20	1 NULL		NULL
nombres	cadena20	20	0 NULL		NULL
causa_baja	char	1	1 NULL		NULL
fecha_baja	fecha	8	0 NULL		NULL
rfc_l	char	4	1 NULL		NULL
rfc_n	char	6	0 NULL		NULL
rfc_h	char	3	1 NULL		NULL
fecha_ing	fecha	8	0 NULL		NULL
fecha_dlab	fecha	8	0 NULL		NULL
fecha_base	fecha	8	0 NULL		NULL
ant_gf	tentero	1	0 cero		NULL
plaza	char4_num	4	0 NULL		char4_num
num_puesto	tipo_num_puesto	2	1 NULL		regla_mayoracer
salario	dinero	8	0 cero		NULL
num_jog	tentero	1	0 cero		NULL
num_area	char1_num	1	1 NULL		char1_num
calle	cadena60_null	60	1 no_propor		NULL
numero	char	10	1 NULL		NULL
colonia	cadena60_null	60	1 no_propor		NULL
localidad	cadena60_null	60	1 no_propor		NULL
estado	num_edo	2	1 NULL		num_edo
cp	char	5	1 NULL		NULL
telefono	telefono	12	0 no tiene		NULL
nacionalidad	cadena20_null	20	1 NULL		NULL
edo_civil	char	1	1 NULL		NULL
sexo	char	1	0 NULL		NULL
num_issste	int	4	1 NULL		NULL
porcentaje_suel	real	4	1 NULL		NULL

index_name: indice_bajas
 index_description: clustered, unique located on default
 index_keys: num_emp
 triggers: emp_bajas_ti inserción

EMP_BECA

Name	Owner	Type			
emp_beca	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_beca	tentero	1	0	cero	NULL
niv_beca	char	1	0	NULL	regla_nivbeca
fecha_ini	fecha	8	0	NULL	NULL
fecha_term	fecha	8	0	NULL	NULL

index_name: indice_beca
 index_description: clustered, unique located on default
 index_keys: num_emp, num_beca
 triggers: emp_beca_tiu inserción

EMP_CAMBIO

Name	Owner	Type			
emp_cambio	dbo	user table			
Data_located_on_segment	When_created				
default	Feb 10 1993 3:50PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
fecha_cambio	fecha	8	0	NULL	NULL
plaza	char4_num	4	0	NULL	char4_num
num_puesto	tipo_num_puesto	2	0	NULL	regla_mayoracer
salario	dinero	8	0	cero	NULL
num_jog	tentero	1	0	cero	NULL
num_area	char1_num	1	0	NULL	char1_num
porcentaje_suel	real	4	1	NULL	NULL

index_name: indice_cambio
 index_description: clustered, unique located on default
 index_keys: num_emp, fecha_cambio
 triggers: emp_cambio_t1 inserción

EMP_DIF_SUELDO

Name	Owner	Type			
emp_dif_sueldo	dbo	user table			
Data located on segment	When created				
default	Mar 4 1994 5:33PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
importe	dinero	8	0 cero		NULL
fecha_ini	fecha	8	0 NULL		NULL
fecha_term	fecha	8	0 NULL		NULL
periodicidad	periodo	1	0 NULL		regla_periodo

index_name: indice_dif_sueldo
 index_description: clustered, unique located on default
 index_keys: num_emp

EMP_HOR_IRR

Name	Owner	Type			
emp_hor_irr	dbo	user table			
Data located on segment	When created				
default	Mar 4 1994 3:11PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
importe	dinero	8	0 cero		NULL
fecha_ini	fecha	8	0 NULL		NULL
fecha_term	fecha	8	0 NULL		NULL
periodicidad	periodo	1	0 NULL		regla_periodo

index_name: indice_hor_irr
 index_description: clustered, unique located on default
 index_keys: num_emp

EMP_LAB

Name	Owner	Type			
emp_lab	dbo	user table			
Data located on segment		When created			
default		Oct 8 1992 3:29PM			
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
plaza	char4_num	4	0	NULL	char4_num
rfc_l	char	4	1	NULL	regla_rfcl
rfc_n	char	6	0	NULL	regla_rfcn
rfc_h	char	3	1	NULL	NULL
fecha_ing	fecha	8	0	NULL	NULL
fecha_dlab	fecha	8	0	NULL	NULL
fecha_base	fecha	8	0	NULL	NULL
ant_gf	tentero	1	0	cero	regla_mayorigua

index name: indice_lab
 index description: clustered, unique located on default
 index keys: num_emp
 triggers: emp_lab_tu inserción,
 emp_lab_tu actualización,
 emp_lab_td borrado

EMP_MSYS

Name	Owner	Type			
emp_msys	dbo	user table			
Data located on segment		When created			
default		Oct 8 1992 3:29PM			
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
aporta_fah	dinero	8	0	cero	NULL
banco	char	2	1	NULL	NULL
num_cuenta	varchar	18	1	NULL	regla_numcuenta
profesion	varchar	6	1	NULL	NULL
plan_fah	char	1	1	NULL	regla_plan_fah

index name: indice_msys
 index description: clustered, unique located on default
 index keys: num_emp
 triggers: emp_msys_tiu inserción

EMP_PERSONAL

Name	Owner	Type			
emp_personal	dbo	user table			
Data_located_on_segment	When_created				
default	Nov 11 1992 2:08PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
apell_pat	cadena20	20	0	NULL	NULL
apell_mat	cadena20_null	20	1	NULL	NULL
nombres	cadena20	20	0	NULL	NULL
calle	cadena60_null	60	1	no_propor	NULL
numero	char	10	1	NULL	NULL
colonia	cadena60_null	60	1	no_propor	NULL
localidad	cadena60_null	60	1	no_propor	NULL
estado	num_edo	2	1	NULL	num_edo
cp	char	5	1	NULL	regla_cp
telefono	telefono	12	0	no_tiene	NULL
nacionalidad	cadena20_null	20	1	NULL	NULL
edo_civil	char	1	1	NULL	regla_edocivil
sexo	char	1	0	NULL	regla_sexo
num_issste	int	4	1	NULL	NULL

index_name: indice_personal
 index_description: Clustered, unique located on default
 index_keys: num_emp
 triggers: emp_personal_ti inserción,
 emp_personal_tu actualización

EMP_PSGS

Name	Owner	Type			
emp_psgs	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
fecha_ini	fecha	8	0	NULL	NULL
fecha_term	fecha	8	0	NULL	NULL

index_name: indice_psgs
 index_description: Clustered, unique located on default
 index_keys: num_emp, fecha_ini
 triggers: emp_psgs_tiu inserción

EMP_SUSPEN

Name	Owner	Type
emp_suspen	dbo	user table

Data_located_on_segment	When_created
default	Oct 8 1992 3:30PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
causa_suspen	char	1	0	NULL	NULL
fecha_suspen	fecha	8	0	NULL	NULL

index name: indice_suspen
 index description: clustered, unique located on default
 index keys: num_emp, fecha_suspen
 triggers: emp_suspen_tiu inserción

EMP_WKEND

Name	Owner	Type
emp_wkend	dbo	user table

Data_located_on_segment	When_created
default	Oct 8 1992 3:29PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL

index name: indice_wkend
 index description: clustered, unique located on default
 index keys: num_emp
 triggers: emp_wkend_tiu inserción

FECHA

Name	Owner	Type
fecha	dbo	user table

Data_located_on_segment	When_created
default	Nov 12 1992 2:42PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
anio	tentero	1	0	cero	NULL
quincena	tentero	1	0	cero	NULL

FONDO_AHORRO

Name	Owner	Type	
fondo_ahorro	dbo	user table	
Data_located_on_segment	When_created		
default	May 2 1994 12:54PM		
Column_name	Type	Length Nulls Default_name	Rule_name
num_emp	tipo_num_emplea	4 0 NULL	NULL
fecha_ing_fa	fecha	8 0 NULL	NULL
saldo_favor	dinero	8 0 cero	NULL

index name: indice_fond_ahorro
 index description: clustered, unique located on default
 index keys: num_emp
 triggers: fondo_ahorro_ti inserción,
 fondo_ahorro_td borrado

FOV_BIM

Name	Owner	Type	
fov_bim	dbo	user table	
Data_located_on_segment	When_created		
default	Jul 28 1993 2:00PM		
Column_name	Type	Length Nulls Default_name	Rule_name
num_emp	tipo_num_emplea	4 0 NULL	NULL
quin1	dinero	8 0 cero	NULL
quin2	dinero	8 0 cero	NULL
quin3	dinero	8 0 cero	NULL
quin4	dinero	8 0 cero	NULL

triggers: fov_bim_ti inserción

INC_NO_CONTRA

Name	Owner	Type	
inc_no_contra	dbo	user table	
Data_located_on_segment	When_created		
default	Nov 12 1993 8:14PM		
Column_name	Type	Length Nulls Default_name	Rule_name
cve_incidente	tipo_cve_incide	2 0 NULL	NULL
inc_contra	tipo_cve_incide	2 0 NULL	NULL

index_name: indice_inc_con
 index_description: clustered located on default
 index_keys: cve_incidente
 triggers: inc_no_contra_tiu inserción y actualización

ISPT_QNAL

Name	Owner	Type	
ispt_qnal	dbo	user table	
Data_located_on_segment	When_created		
default	Oct 22 1992 2:21PM		
Column_name	Type	Length Nulls Default_name	Rule_name
nivel	tentero	1 0 cero	NULL
limite_inferior	dinero	8 0 cero	NULL
limite_superior	dinero	8 0 cero	NULL
cuota_fija	dinero	8 0 cero	NULL
porcentaje	real	4 0 NULL	NULL

index_name: indice_ispt
 index_description: clustered, unique located on default
 index_keys: nivel

ORDEN_DEDUCCION

Name	Owner	Type
orden_deduccion	dbo	user table
Data_located_on_segment	When_created	
default	Mar 16 1994 2:02PM	

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_deduccion	tipo_num_deducc	1	0	NULL	regla_numdeducc
num_folio	int	4	0	NULL	NULL
cantidad	tipo cantidad	4	0	NULL	regla_mayorigua
veces	smallint	2	0	NULL	NULL
costo	dinero	8	0	cero	NULL
veces_deducidas	smallint	2	0	NULL	NULL

index_name: ind_ord_deduc

index_description: nonclustered located on default

index_keys: num_emp, num_deduccion, num_folio

triggers: orden_deduccion_ti inserción

ORDEN_PERCEPCION

Name	Owner	Type
orden_percepcion	dbo	user table
Data_located_on_segment	When_created	
default	Mar 16 1994 2:00PM	

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_percepcion	tipo_num_percep	1	0	NULL	regla_numero
num_folio	int	4	0	NULL	NULL
cantidad	tipo cantidad	4	0	NULL	regla_mayorigua
veces	smallint	2	0	NULL	NULL
costos	dinero	8	0	cero	NULL
veces_pagadas	smallint	2	0	NULL	NULL

index_name: ind_ord_percep

index_description: nonclustered located on default

index_keys: num_emp, num_percepcion, num_folio

triggers: orden_percepcion_ti inserción
orden_percepcion_tu actualización

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

PAGOS_SERV					
Name		Owner		Type	
pagos_serv		dbo		user table	
Data_located_on_segment		When_created			
default		Nov 27 1992 10:40AM			
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_bec	tentero	1	0	cero	NULL
fech_pago	fecha	8	0	NULL	NULL
num_mes	tentero	1	0	cero	NULL

PERCEPCION_ACUM					
Name		Owner		Type	
percepcion_acum		dbo		user table	
Data_located_on_segment		When_created			
default		Feb 24 1993 2:51PM			
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_percepcion	tipo_num_percep	1	0	NULL	regla_numero
num_acumulador	char	1	0	NULL	NULL
rutina	tentero	1	0	cero	NULL
index_name: indice_percep_acum					
index_description: clustered, unique located on default					
index_keys: num_percepcion, num_acumulador					
triggers: percepcion_acum_ti inserción					

PERCEPCION_APLI					
Name		Owner		Type	
percepcion_apli		dbo		user table	
Data_located_on_segment		When_created			
default		Jan 14 1994 1:23PM			
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_qna	tentero	1	0	cero	regla_quincena
num_anio	anio	2	0	def anio	anio
num_percepcion	tipo_num_percep	1	0	NULL	regla_numero
num_folio	int	4	0	NULL	NULL
costo	dinero	8	0	cero	NULL

index_name: ind_percep_apli
 index_description: nonclustered located on default
 index_keys: num_emp, num_qna, num_anio, num_percepcion, num_folio

PERCEPCION_COND

Name	Owner	Type			
percepcion_cond	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_percepcion	tipo_num_percep	1	0	NULL	regla_numero
num_condicion	texto	1	0	cero	NULL

index_name: indice_percep_con
 index_description: clustered, unique located on default
 index_keys: num_percepcion, num_condicion
 triggers: percepcion_cond_ti inserción

PERCEPCION_QNA

Name	Owner	Type			
percepcion_qna	dbo	user table			
Data_located_on_segment	When_created				
default	Jan 14 1994 1:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_percepcion	tipo_num_percep	1	0	NULL	regla_numero
num_folio	int	4	0	NULL	NULL
costo	dinero	8	0	cero	NULL

index_name: ind_percep_qna
 index_description: nonclustered located on default
 index_keys: num_emp, num_percepcion, num_folio

PJA_INCIDENTE

Name	Owner	Type			
pja_incidentes	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
cve_pja	tipo_cve_pja	2	0	NULL	NULL
cve_incidente	tipo_cve_incide	2	0	NULL	NULL

index_name: indice_pja_inc
 index_description: clustered, unique located on default
 index_keys: cve_pja, cve_incidentes
 triggers: pja_incidente_tiu inserción

PJA_NO_CONTRA

Name	Owner	Type			
pja_no_contra	dbo	user table			
Data_located_on_segment	When_created				
default	Nov 30 1993 6:00PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
cve_pja	tipo_cve_pja	2	0 NULL		NULL
pja_contra	tipo_cve_pja	2	0 NULL		NULL

index_name: indice_pja_con
 index_description: clustered located on default
 index_keys: cve_pja
 triggers: pja_no_contra_tiu inserción y actualización

PLAZA

Name	Owner	Type			
plaza	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
plaza	char4_num	4	0 NULL		char4_num
categoria	char	1	0 NULL		regla_categoria
num_jog	tentero	1	0 cero		regla_numjog
propietario	int	4	1 NULL		NULL
num_puesto	tipo_num_puesto	2	0 NULL		regla_mayoracer
num_area	char1_num	1	0 NULL		char1_num
fecha_crea	fecha	8	0 NULL		NULL
fecha_act	fecha	8	0 NULL		NULL

index_name: indice_plaza
 index_description: clustered, unique located on default
 index_keys: plaza

index_name: plaza_jog
 index_description: nonclustered located on default
 index_key: num_jog, num_area

trigger: plaza_tiu inserción y actualización

FORCEN_TRAB

Name	Owner	Type			
porcen_trab	dbo	user table			
Data_located_on_segment	When_created				
default	Oct 27 1993 7:43PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
porcen_anual	float	8	1	NULL	NULL
porcen_semes	float	8	1	NULL	NULL

index_name: ind_porcen_trab
 index_description: clustered located on default
 index_keys: num_emp
 triggers: porcen_trab_ti inserción

PRPSTAMO_ISSSTE

Name	Owner	Type			
prestamo_issste	dbo	user table			
Data_located_on_segment	When_created				
default	May 16 1994 2:43PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
num_deduccion	tipo_num_deducc	1	0	NULL	regla_numdeducc
num_folio	int	4	0	NULL	NULL
num_liquidacion	varchar	15	0	NULL	NULL
importe_prestam	dinero	8	0	cero	NULL
fecha_ini	fecha	8	0	NULL	NULL

index_name: ind_prestamo
 index_description: nonclustered located on default
 index_keys: num_emp, num_deduccion, num_folio, num_liquidacion

PGSG_HISTORICO

Name	Owner	Type			
page_historico	dbo	user table			
Data located on segment		When created			
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
fecha_ini	fecha	8	0 NULL		NULL
fecha_term	fecha	8	0 NULL		NULL

index_name: indice_pgshisto
 index_description: clustered, unique located on default
 index_keys: num_emp, fecha_ini

QNA_EVENT_DEDUC

Name	Owner	Type			
qna_event_deduc	dbo	user table			
Data located on segment		When created			
default	Oct 8 1992 3:29PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_cond_deduc	tentero	1	0 cero		NULL
num_qna	tentero	1	0 cero		regla_quincena

index_name: indice_event_deduc
 index_description: clustered, unique located on default
 index_keys: num_cond_deduc, num_qna
 triggers: qna_event_deduc_ti inserción

QNA_EVENTO

Name	Owner	Type			
qna_evento	dbo	user table			
Data located on segment		When created			
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_cond	tentero	1	0 cero		regla_numcond
num_qna	tentero	1	0 cero		regla_quincena

index_name: indice_qna_evento
 index_description: clustered, unique located on default
 index_keys: num_cond, num_qna
 triggers: qna_event_ti inserción

QUINQUENIOS

Name	Owner	Type			
quinquenios	dbo	user table			
Data_located_on_segment	When_created				
default	Jan 18 1993 1:55PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
num_quinque	tinyInt	1	0 NULL		NULL
index_name: ind quinque					
index_description: nonclustered located on default					
index_keys: num_emp					
triggers: quinquenios_tiu inserción y actualización					

REP_ABC

Name	Owner	Type			
rep_abc	dbo	user table			
Data_located_on_segment	When_created				
default	Apr 6 1993 12:26PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0 NULL		NULL
cve_rep	char	1	0 NULL		NULL
alta	fecha	8	1 NULL		NULL
baja	fecha	8	1 NULL		NULL
cambio	fecha	8	1 NULL		NULL
index_name: ind rep_abc					
index_description: nonclustered located on default					
index_keys: num_emp, cve_rep					

S SOCIAL

Name	Owner	Type			
s_social	dbo	user table			
Data_located_on_segment	When_created				
default	Nov 27 1992 10:40AM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_bec	tentero	1	0 cero		NULL
apell_pat	cadena20	20	0 NULL		NULL
apell_mat	cadena20_null	20	1 NULL		NULL
nombres	cadena20	20	0 NULL		NULL
num_jog	tentero	1	0 cero		NULL
monto	dinero	8	0 cero		NULL

SAR_BIM

Name	Owner	Type	
-----	-----	-----	
sar_bim	dbo	user table	
Data_located_on_segment	When_created		
-----	-----	-----	
default	Jul 9 1993 5:12PM		
Column_name	Type	Length Nulls Default_name	Rule_name
-----	-----	-----	-----
num_emp	tipo_num_emplea	4 0 NULL	NULL
quin1	dinero	8 0 cero	NULL
quin2	dinero	8 0 cero	NULL
quin3	dinero	8 0 cero	NULL
quin4	dinero	8 0 cero	NULL
bim	tinyint	1 0 NULL	regla_bim

triggers: sar_bim_ti inserción

SUBSIDIO_QNAL

Name	Owner	Type	
-----	-----	-----	
subsidio_qnal	dbo	user table	
Data_located_on_segment	When_created		
-----	-----	-----	
default	Dec 6 1993 3:42PM		
Column_name	Type	Length Nulls Default_name	Rule_name
-----	-----	-----	-----
nivel	tentero	1 0 cero	NULL
limite_inferior	dinero	8 0 cero	NULL
limite_superior	dinero	8 0 cero	NULL
cuota_fija	dinero	8 0 cero	NULL
porcentaje	float	8 0 NULL	NULL

index_name: indice_subsidio

index_description: clustered, unique located on default

index_keys: nivel

SUELDO

Name	Owner	Type			
sueldo	dbo	user table			
Data located on segment	When created				
default	Oct 8 1992 3:30PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_puesto	tipo_num_puesto	2	0	NULL	regla_mayoracer
categoria	char	1	0	NULL	regla_categoria
zona_ec	tentero	1	0	cero	regla_zonaec
salario	dinero	8	0	cero	NULL

index_name: indice_sueldo
 index_description: clustered, unique located on default
 index_keys: num_puesto, categoria, zona_ec
 triggers: sueldo_t1 inserción,
 sueldo_tu actualización

SUM_AREAS

Name	Owner	Type			
sum_areas	dbo	user table			
Data located on segment	When created				
default	Feb 15 1993 6:50PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
jog	cadena60	60	0	no_propor	NULL
area	cadena60	60	0	no_propor	NULL
sdoplaareas	money	8	0	NULL	NULL
sdoplajobs	money	8	0	NULL	NULL
plareas	int	4	0	NULL	NULL
plajogs	int	4	0	NULL	NULL
nomareas	int	4	0	NULL	NULL
nomjobs	int	4	0	NULL	NULL
sdoempareas	money	8	0	NULL	NULL
sdoempjobs	money	8	0	NULL	NULL

index_name: indice_sum_area
 index_description: Clustered located on default
 index_keys: jog, area

SUSPEN_HISTORICO

Name	Owner	Type			
suspen_historico	dbo	user table			
Data_located_on_segment	When_created				
default	Feb 22 1993 3:28PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
causa_suspen	char	1	0	NULL	NULL
fecha_suspen	fecha	8	0	NULL	NULL
fecha_reinstala	fecha	8	0	NULL	NULL

VACANTES

Name	Owner	Type			
vacantes	dbo	user table			
Data_located_on_segment	When_created				
default	Jan 27 1993 1:09PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
plaza	char4_num	4	0	NULL	char4_num
reportada	booleano	1	0	cero	NULL
prmcionmm	booleano	1	0	cero	NULL
prmcionc	booleano	1	0	cero	NULL
prmcione	booleano	1	0	cero	NULL
disponible	booleano	1	0	cero	NULL
provisional	booleano	1	0	cero	NULL
definitiva	booleano	1	0	cero	NULL

index name: ind_vacantes
 index description: nonclustered located on default
 index keys: plaza
 triggers: vacantes_ti inserción

Para más detalles de los triggers ver el apéndice A, para vistas apéndice B y para reglas y defaults ver el apéndice C.

III.2 DISEÑO FÍSICO.

III.2.1 TAMAÑO DE LAS BASES DE DATOS.

En el manejador de base de datos que se va a utilizar el tamaño de la base de datos debe estar en Megabytes, el tamaño mínimo es de 2 Megabytes, el tamaño por default es el tamaño que tenga la base de datos Model o el tamaño indicado en la tabla de configuración del sistema en la parte de "database size".

Para determinar el tamaño de la base de datos, es necesario incluir:

- Datos
- Indices
- Log de transacciones
- Un 25% más del calculado (para crecimiento).

Para calcular el tamaño de la tabla se hace lo siguiente:

- Estimar el tamaño en bytes del renglón
- Estimar la cantidad de registros que va a tener una tabla
- Agregar el overhead necesario

Para determinar el número de bytes por renglón, es necesario conocer la cantidad de bytes que utiliza cada tipo de datos, tal como se muestra en la tabla [3.3].

TIPO DE DATOS	BYTES
char	1-256
varchar	1-256
smallint	2
tinyint	1
int	2
float	4
text	2 M
bit	1 bit

TABLA [3.3] TAMAÑOS DE LOS TIPOS DE DATOS

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

Ejemplo:

Para la tabla Emp_lab el tamaño del registro es de 46 bytes, una tabla con aproximadamente 2,500 registros, se agregan 4 bytes de overhead, por lo tanto, el tamaño final de registro es de 50 bytes, finalmente se tiene:

50 bytes X 2,500 registros = 125,000 bytes = 125 Kb.

El tamaño en datos para la tabla emp_lab es de 125 Kb.

De la misma forma se realizó el cálculo para el resto de las tablas y se obtuvo el resultado que se muestra en la tabla [3.4].

TABLA	TAMAÑO	TABLA	TAMAÑO
EMP_ACUM	180K	CAT_COND_DEDUC	2K
DEDUCCION_COND	2K	QNA_EVENT_DEDUC	2K
EMP_BAJAS	1M	REP_ABC	132K
FECHA	2K	PRESTAMO_ISSSTE	45K
DEDUCCION_QNA	63K	CAT_DEDUCCION	6K
DIA_EVENTO	2K	PSGS_HISTORICO	4K
CAT_BAJAS	2K	SUSPEN_HISTORICO	4K
CAT_QUINQUENIOS	2K	CONT_EVEN	8K
EMP_SUSPEN	2K	EMP_PERSONAL	296K
ORDEN_DEDUCCION	54K	DEDUCCION_APLI	250K
EMP_CAMBIO	144K	SUBSIDIO_QNAL	2K
ISPT_QNAL	2K	CREDITO_SAL_MENS	2K
QUINQUENIOS	14K	EMP_HOR_IRR	12K
EMP_DIF_SUELDO	6K	DEP_ECONOMICOS	42K
EMP_PSGS	2K	EMP_LAB	125K
PERCEPCION_APLI	250K	PERCEPCION_QNA	63K
QNA_EVENTO	2K	ORDEN_PERCEPCION	50K
CAT_PERCEPCION	6K	PERCEPCION_COND	2K

TABLA [3.4] TAMAÑO DE TABLAS (DATOS).

CONTINUA...

...CONTINUACION

TABLA	TAMAÑO	TABLA	TAMAÑO
CAT COND PER	2K	PORCEN TRA	35K
PERCEPCION ACUM	2K	AVAL	29K
FOV BIM	100K	FONDO AHORRO	98K
DEUDA_FA	32K	DEUDA APLI	700K
BECA IMPTE	2K	EMP BECA	4K
PLAZA	98K	EMP MMYS	10K
EMP BC	38K	SAR BIM	103K
SUM AREAS	6K	CAT PUESTO	6K
CAT JOG	4K	AREA TODAS JZ	2K
AREA OC	4K	CAT BANCOS	2K
EMP WKEND	8K	CTRL ASIS	4.5M
CAT INCIDENTE	2K	INC NO CONTRA	2K
SUELDO	8K	CAT ZE	2K
VACANTES	10K	PJA NO CONTRA	2K
CAT PJA	2K	PJA INCIDENTE	2K

TABLA [3.4] TAMAÑO DE TABLAS (DATOS).

La suma de los tamaños de las tablas nos da: 8,597K. Este sería el tamaño de los datos, para los índices, suponemos que todos van a ser clustered y por lo tanto el espacio que requieren es el tamaño de la tabla mas un 20%, como el tamaño obtenido anteriormente es el tamaño de las tablas, el tamaño para los índices sería:

$$8,597K + (8,597K * 20\%) = 8,597 + 1719.4 \\ = 10,316.4K$$

Hasta este momento el tamaño de la base de datos sería de:

$$\text{Tamaño datos} + \text{tamaño índices} = \\ 8,597K + 10,316.4K = 18,913.4K$$

Le damos un 10% por error humano y un 25% para crecimiento futuro. Entonces el tamaño sería:

$$18,913.4K + 1,891.34K + 4,728.35K = 25,533.09K$$

Para el log de transacciones se calcula con el 10% del tamaño estimado de la base, pero como para el sistema se espera un gran número de transacciones se le asigna un 30%, entonces el tamaño del log de transacciones es de:

$$25,533.09K * 30\% = 7,659.927K$$

Por último el tamaño total de la base de datos será, el tamaño de datos e índices + el tamaño del log de transacciones:

$$25,533.09K \text{ aproximadamente } 26Mb$$

$$7,659.927K \text{ aproximadamente } 8Mb$$

Por lo que el tamaño será: $26Mb + 8Mb = 34Mb$.

III.2.2 ARQUITECTURA.

Una arquitectura en este tipo de contexto, se refiere a la forma en que trabaja el manejador de base de datos, como atiende a sus usuarios, como maneja los datos y la memoria, y otros varios aspectos.

En el siguiente punto se describe la arquitectura cliente servidor que es la que maneja la herramienta que se utilizó para la implantación del sistema.

III.2.2.1 ARQUITECTURA CLIENTE-SERVIDOR.

La arquitectura cliente-servidor se refiere al manejo de la información por medio de un servidor lógico, que es el que la administra físicamente, en espera de peticiones de los clientes que se conectan vía red al servidor, como se muestra en la figura [3.13].

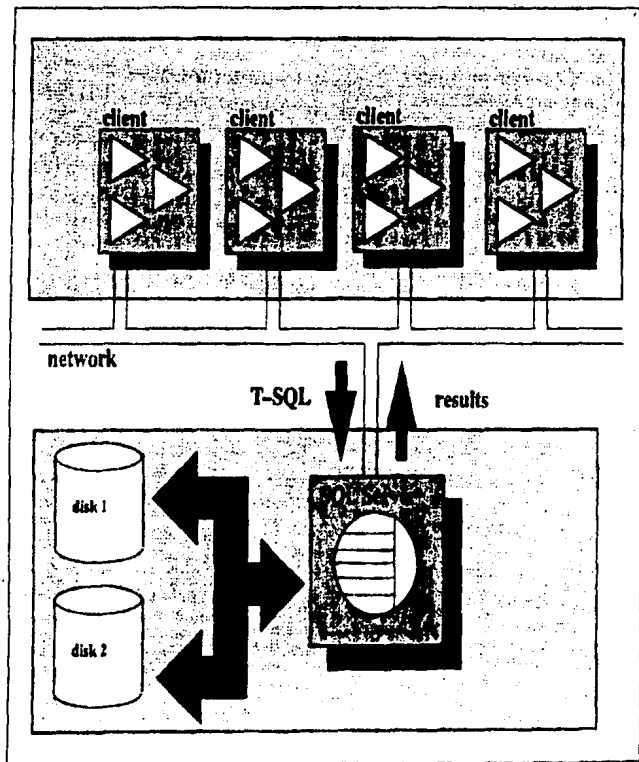


FIGURA 3.13 ARQUITECTURA CLIENTE SERVIDOR.

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

Los clientes no necesariamente deben de estar en la misma máquina, estos pueden estar en diferentes nodos de una red, existen clientes propios del DBMS como son: ISQL, APT_WB, DWB, BCP, que funcionan como front-end, existen los open client que son programas en lenguajes de tercera generación que accesan al servidor para obtener información de la base de datos.

El servidor maneja varias bases de datos así como varios usuarios, sabe exactamente la localización actual de los datos en el disco y que datos pertenecen a cada base de datos, mantiene la relación entre la parte lógica y la parte física del almacenamiento, mantiene los datos y procedimientos en memoria cache, compila y ejecuta las sentencias de SQL, regresa los resultados a los programas clientes, determina automáticamente la forma más eficiente de realizar las tareas en la base de datos, maneja una arquitectura multihilos, que quiere decir que por cada usuario establece un canal de comunicación diferente, diseñado en base a la tecnología de redes, reduce significativamente el tráfico de la red.

Algo importante es que el lenguaje de SQL proporciona una interfaz bien definida entre los sistemas frontales y de soporte, comunicando las peticiones de acceso a la base de datos de una manera eficiente.

III.2.3 SEGURIDAD DEL SISTEMA.

El DBMS provee procedimientos de sistema para el control de la seguridad en la base de datos, basándose en tres conceptos principalmente:

- 1.- Usuarios: son los que ocupan la base de datos, es decir los que consultan, insertan, suprimen o actualizan datos.
- 2.- Objetos: son los elementos que existen en una base de datos, tales como tablas, vistas, procedimientos almacenados, sobre los que se aplica la seguridad.
- 3.- Privilegios: son las acciones que se permiten sobre los

objetos a un usuario.

Existen dos tipos de privilegios:

a) Permisos de ejecución de comandos, tal como crear un default, crear un procedimiento, una tabla; en general la creación de objetos en la base de datos, así como el respaldo y recuperación de la información.

b) Permisos sobre objetos: que son los que se refieren a poder consultar, insertar, borrar, actualizar sobre algún objeto, así como permisos de ejecución sobre procedimientos almacenados.

El DBMS provee una jerarquía de usuarios para establecer las restricciones de acceso a los objetos de las bases de datos:

- SA: El System Administrator, es el nivel de usuario más alto, el cual puede crear bases de datos, controlar los permisos para todos los usuarios, en sí administra al manejador.

- DBO: Data Base Owner, es el dueño de la base de datos, decide que usuarios pueden hacer uso de la base de datos así como creación de grupos para control de los usuarios, es el que tiene todos los privilegios en su base de datos.

- Object Owner: Es el dueño de un objeto en la base de datos.

- Usuario: Es el que hace uso de los objetos de la base de datos.

- Grupo: Es un conjunto de usuarios, con los mismos privilegios.

- Public: Es el grupo con menos privilegios dentro de las bases de datos, al que pertenece el usuario invitado, el cual no tiene ningún privilegio más que el acceder la base de datos y saber que objetos tiene.

IV - DISEÑO DE LAS APLICACIONES DE INTERFAZ CON EL USUARIO (FRONT - END)

En éste capítulo se trata la forma en que se presentan las pantallas de captura, las pantallas de menús y los reportes, que se muestran al usuario, además de los procedimientos necesarios para llegar a ello.

IV.1 DISEÑO DE FORMAS

Se realiza un árbol de menús, el cual nos conducirá a establecer el orden en que se encuentran las formas, además nos ayudará a determinar las formas que se necesiten en cada caso, éste árbol incluye las opciones referentes a proceso, reportes y consultas además de las de captura.

Para el árbol de menús se trata de establecer las actividades principales en el manejo del sistema, quedando conformado de la siguiente forma:

1.- Control de Asistencia.- En esta parte del sistema se contempla la captura de los incidentes y sus respectivos pja's que cada empleado genere a partir de su tarjeta de asistencia y los comprobantes que se presenten, así como el mantenimiento del catálogo que relaciona que pja's son válidos para cada uno de los incidentes.

2.- Transacciones.- Se refiere a dos tipos de transacciones básicamente, las de catálogo, en donde se contempla el mantenimiento de todas las tablas de catálogo generales del sistema, y las transacciones del empleado, donde se manejan todo tipo de movimientos referentes al empleado.

3.- Percepciones.- Se refiere al mantenimiento de sus catálogos, además de la captura de incidencias que puedan provocar una percepción.

4.- Deducciones.- Mantenimiento de los catálogos relacionados con las deducciones, así como la captura de incidencias que generen una deducción.

5.- Nómina.- Contempla todo lo relativo al manejo y operación de la nómina.

6.- SAR y FOVISSSTE.- Se generan los pagos quincenales al SAR y al FOVISSSTE, se crean los archivos de información bimestral al banco, se tienen formas de consulta para cada uno y se generan los reportes correspondientes.

7.- ISSSTE (ABC).- Son los avisos de altas, bajas y cambios contenidos en reportes y archivos para informar al ISSSTE.

8.- Reportes.- Reportes varios, ya sean en papel o medios magnéticos.

9.- Consultas.- Consultas de nivel gerencial para la toma de decisiones, más rápida.

Del punto anterior tenemos que cada opción nos lleva a otro nivel más interno:

1. Control de asistencia

1.1. Relación PJA-Incidente

1.2. Captura de control de asistencia

2. Transacciones

2.1. Catálogos

2.1.1. Catálogo de bajas

2.1.2. Catálogo de bancos

2.1.3. Catálogo de incidentes

2.1.4. Catálogo de JOG

2.1.5. Catálogo de PJA

2.1.6. Catálogo de puestos

2.1.7. Catálogo de quinquenios

2.1.8. Catálogo de zonas económicas

2.1.9. Plaza

2.1.A. Sueldo

2.1.B. Areas oficinas centrales (O.C.)

2.1.C. Areas de Jefaturas de zona (J.Z.)

2.1.D. Importe de beca

2.2. Empleado

- 2.2.1. Alta general
- 2.2.2. Reinstala suspensión
- 2.2.3. P.S.G.S.
- 2.2.4. Termina contrato eventual
- 2.2.5. Cambios a plaza
- 2.2.6. Bajas
- 2.2.7. Reinstala bajas
- 2.2.8. Cambios Personales
- 2.2.9. Cambios Laborales
- 3. Percepciones
 - 3.1. Cat. condición percepción
 - 3.2. Cat. de percepciones
 - 3.3. Quincena evento
 - 3.4. Percepción condición
 - 3.5. Percepción acumulador
 - 3.6. Incidencias percepción
 - 3.7. Inc. Horario Irregular
 - 3.8. Inc. Diferencia de sueldo
 - 3.9. Inc. Beca
- 4. Deducciones
 - 4.1. Cat. condición deducción
 - 4.2. Cat. Deducción
 - 4.3. Deducción condición
 - 4.4. Día evento
 - 4.5. Quincena evento deducción
 - 4.6. I.S.P.T. quincenal
 - 4.7. Subsidio quincenal
 - 4.8. Crédito al salario mensual
 - 4.9. Incidencias deducción
 - 4.A. Deuda fondo ahorro
- 5. Nómina
 - 5.1. Cálculo de Nómina
 - 5.2. Impresión de nómina
 - 5.3. Impresión de sobres
 - 5.4. Impresión de tarjetas de asistencia
- 6.- SAR y FOVISSSTE
 - 6.1. Cálculo del SAR
 - 6.2. Cálculo del FOVISSSTE
 - 6.3. Impresión SAR Y FOVISSSTE
 - 6.4. Generación de disco para sucursal bancaria
 - 6.5. Consulta SAR (puede actualizar)
 - 6.6. Consulta FOVISSSTE (puede actualizar)
- 7. ISSSTE (ABC)
- 8. Reportes
 - 8.1. Catálogos
 - 8.1.1. Areas J.2.
 - 8.1.2. Areas O.C.
 - 8.1.3. Bajas
 - 8.1.4. Bancos
 - 8.1.5. Condición deducción

- 8.1.6. Condición percepción
- 8.1.7. Deducciones
- 8.1.8. Deducción condición
- 8.1.9. Día evento
- 8.1.A. Importe beca
- 8.1.B. Incidentes
- 8.1.C. ISPT quincenal
- 8.1.D. JOG
- 8.1.E. Percepciones
- 8.1.F. Percepción acumulador
- 8.1.G. Percepción condición
- 8.1.H. PJA
- 8.1.I. PJA-Incidentes
- 8.1.J. Puestos
- 8.1.K. Quincena evento
- 8.1.L. Quincena evento deducción
- 8.1.M. Quinquenios
- 8.1.N. Subsidio quincenal
- 8.1.Ñ. Sueldo
- 8.1.O. Zonas económicas
- 8.2. Plantillas
 - 8.2.1. Base J Z.
 - 8.2.2. Base O.C.
 - 8.2.3. Confianza J.Z.
 - 8.2.4. Confianza O.C.
 - 8.2.5. Mandos medios y superiores (MMYS) J.Z.
 - 8.2.6. MMYS O.C.
- 8.3. Reportes
 - 8.3.1. Altas, Bajas y Cambios
 - 8.3.2. Avisos de cambio en nómina
 - 8.3.3. Otros reportes
- 9. Consultas
 - 9.1. Consulta general de un empleado
 - 9.2. Consulta percepciones y deducciones de un empleado

Los niveles terminales nos pueden llevar a una forma de captura o a un proceso.

DISEÑO DE PANTALLAS

Para la realización del sistema se ocupo una pantalla estándar para todas las partes que interactuan con el usuario, esta pantalla se compone por:

a) **Encabezado.**- contiene las siglas de la empresa, el nombre del sistema y la fecha actual.

b) **Ventana.**- Contiene los menús con las opciones principales listadas en el árbol de menús, así como las formas de

captura.

c) **Teclas rápidas.**- Nos define el uso de las principales teclas rápidas usadas en el sistema.

En la figura [4.1] se muestra la estructura básica de las pantallas utilizadas en el sistema.

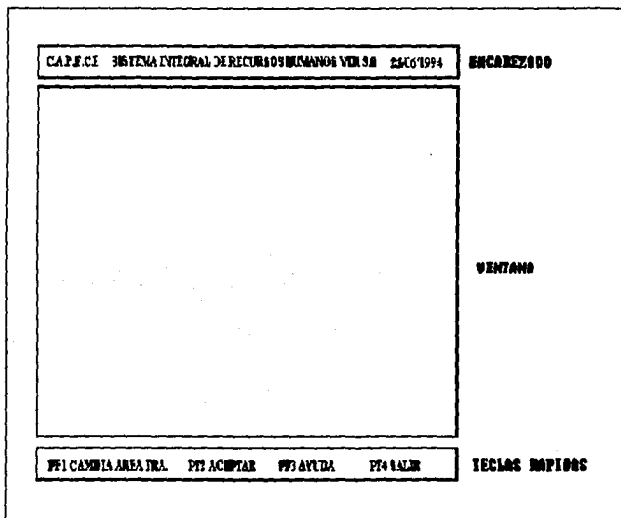


FIGURA [4.1] PANTALLA BASICA

Como ya se mencionó la parte de ventana contiene los menús del árbol de menús antes visto, en donde dependerá del valor que tome la opción para pasar a otro menú o a una forma de captura; para la

realización de las formas se diseñaron procedimientos estándares, los cuales pueden estar ligados al menú, a los campos, a la forma y a las teclas. Las formas tienen un diseño estándar, y están compuestas por un área de menú y un área de trabajo, como se muestra en la figura [4.2].

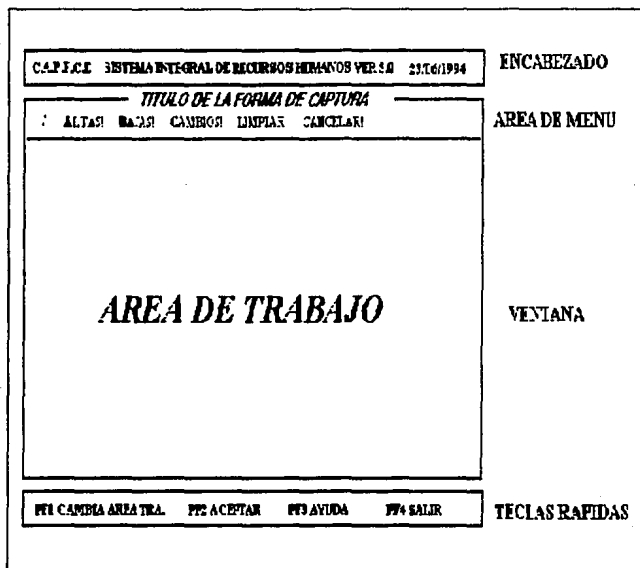


FIGURA [4.2] AREA DE MENU Y AREA DE TRABAJO

Las formas cuentan con los siguientes elementos:

- Opciones, se encuentran en el área de menú.
- Campos, se encuentran en el área de trabajo.

Como ya se mencionó, las formas tienen diferentes tipos de procedimientos:

a) Procedimientos ligados al menú.- estos procedimientos se activan cuando el usuario se encuentra en el área de menú y selecciona alguna de las opciones que en ella se encuentran, la mayoría de las formas en el área de menú presentan las opciones que se ven en la figura [4.2]. (Altas, Bajas, Cambios, Limpiar y Cancelar) Las cuales al ser seleccionadas llaman a un procedimiento que al ejecutarse realiza la transacción correspondiente.

b) Procedimientos ligados a los campos.- existen tres clase de procedimientos ligados a los campos, uno que se activa antes de la captura de la información llamado "preproceso", por ejemplo cuando una forma de captura es llamada por otra y necesita pasarle información, otro que se activa inmediatamente después de salir de la captura del campo llamado "postproceso", por ejemplo para la validación del contenido del campo, o para pasarlo de minúsculas a mayúsculas, otro tipo de procedimientos son los de "selección", por ejemplo para los menús principales se utilizó un tipo especial de campo llamado "campo single choice", el cual tiene los diferentes valores que pude tomar un campo, al ser seleccionado uno de los valores se ejecuta su procedimiento asociado.

c) Procedimientos ligados a la forma.- Existen dos clases de procedimientos ligados a las formas, los procedimientos de entrada a la forma , que son utilizados para inicializar los canales de comunicación y para inicializar los menús, y el otro tipo de salida de la forma, para cerrar el canal de comunicación.

d) Procedimientos ligados a las teclas. Estos procedimientos son ejecutados cuando se presiona cualquiera de las teclas rápidas.

En el siguiente tema se mostrarán los algoritmos de los procedimientos generales que son: Altas, Bajas y Cambios.

IV.2 ALGORITMOS DE PROCEDIMIENTOS GENERALES

Los algoritmos que se presentan a continuación son para las formas de captura, estos se implementaron en APT Workbench, el cual es un producto propio del manejador SYBASE, que se utiliza para la generación de front-ends.

ALGORITMO UTILIZADO EN LAS ALTAS

- 1.- Obtiene el nombre de la tabla a utilizar.
- 2.- Inicialización a la variable en donde se arma la instrucción de inserción con el nombre de la tabla.
- 3.- Mientras existan campos relacionados a la tabla haz
 - 4.- Obtén el valor del campo que se tecleo en la forma.
 - 5.- Agregarlo a la variable donde se arma la instrucción.
- 6.- Ejecuta la instrucción.
- 7.- Si no hubo error manda mensaje de "Alta efectuada"
- 8.- De lo contrario manda mensaje de error.

ALGORITMO UTILIZADO EN LAS BAJAS

- 1.- Obtener el nombre de la tabla a utilizar.
- 2.- Inicializar de la variable donde se arma la instrucción de borrado con el nombre de la tabla.
- 3.- Mientras existan campos llaves de la tabla haz
 - 4.- Obtén el valor del campo que se tecleo en la forma.
 - 5.- Agregarlo a la variable donde se arma la instrucción.
- 6.- Ejecuta la instrucción.
- 7.- Si no hubo error manda mensaje de "Baja efectuada"
- 8.- De lo contrario manda un mensaje de error.

ALGORITMO UTILIZADO EN UN CAMBIO

- 1.- Obtén el nombre de la tabla a utilizar.
- 2.- Inicializar la primer variable que contendrá la instrucción de cambio con el nombre de la tabla.
- 3.- Verificar que existan campos modificados.

DISEÑO DE LAS APLICACIONES DE INTERFAE CON EL USUARIO (FRONT - END)

- 4.- Si los hay entonces pase a 5 de lo contrario ir a 13.
- 5.- Verificar que los campos modificados sean válidos para modificar.
- 6.- Si son válidos ir a 7, si no pasar a 13.
- 7.- En la segunda variable obtén los valores de los campos y arma la parte de la modificación.
- 8.- En la tercer variable obtén los campos llaves de la tabla.
- 9.- La instrucción es armada a partir de la suma de las variables uno, dos y tres.
- 10.- Ejecutar la instrucción.
- 11.- Si no hubo error manda mensaje de "Un registro modificado"
- 12.- De lo contrario manda un mensaje de error.
- 13.- Manda mensaje de cambio no efectuado.

ALGORITMO UTILIZADO PARA LIMPIAR UNA FORMA

- 1.- Para los campos relacionados a la forma aquellos que sean llave y no se permita posicionarse en ellos, dejar posesionarse en ellos.
- 2.- Ejecuta el procedimiento de inicialización de menú.
- 3.- Deja en blanco los valores de los campos.
- 4.- Se posiciona en el campo inicial.

ALGORITMO UTILIZADO PARA INICIALIZAR UN CANAL DE COMUNICACION CON EL SERVIDOR

- 1.- Verificar que el canal no este abierto.
- 2.- Si esta abierto cerrarlo.
- 3.- Abrir el canal de comunicación.

ALGORITMO UTILIZADO PARA INICIALIZAR EL MENU

- 1.- Inhibir las opciones de altas, bajas y cambios.

ALGORITMO UTILIZADO PARA CERRAR EL CANAL DE COMUNICACION CON EL SERVIDOR

- 1.- Cerrar el canal de comunicación.

SINRECH

ALGORITMO UTILIZADO EN CAMPOS LLAVES

- 1.- Verificar que la información tecleada no sea nula.
- 2.- Si la información no es nula pasa a 3 de lo contrario pasa a 7.
- 3.- Si es el último campo llave pasa a 4 de lo contrario ve a 6.
- 4.- Busca en la tabla utilizada si existe la información de los campos llave dados.
- 5.- Si existe modifica el menú para poder utilizar Bajas y Cambios, de lo contrario modificarlo para usar solo Altas.
- 6.- Inhibir este campo y ve a 8.
- 7.- Manda un mensaje de "Introduzca dato por favor", ve a 1.
- 8.- Ve al campo siguiente.

Por otra parte utilizando el Open-Client de SYBASE, que son programas hechos en lenguaje 'C' que a través de librerías accesan al servidor, fueron hechos algunos procesos terminales, como son:

ALGORITMO PARA GENERAR ORDENES DE PERCEPCION DE CATALOGO

- 1.- Crea el vector de condiciones generales de la quincena.
- 2.- Para cada empleado, crea el vector de condiciones particulares.
- 3.- Genera la matriz de percepción-condición.
- 4.- Evaluar en base al vector de condiciones generales, a las condiciones del empleado y la matriz de percepción-condición, que percepciones se le pueden generar al empleado.
- 5.- Si la percepción se genera insertarla en la tabla de orden_percepción.
- 6.- Haz el paso 4 y 5 para todos los empleados.

Otros procesos elaborados en open-client, de los cuales no se pone su algoritmo son:

- Generación de ordenes de deducción de catálogo.
- Costeo de percepciones.
- Costeo de deducciones.

IV.3 DISEÑO DE REPORTES

Los reportes son una parte esencial en la aceptación de un sistema por parte del usuario final, ya que es donde identifica si el sistema le proporciona a su trabajo control de la información, organización, limpieza y la reducción de procesos manuales.

La generación de los reportes para el sistema se basó en las dos herramientas que nos proporciona el manejador SYBASE, que son:

- i) El generador de reportes
- ii) Open client

IV.3.1 GENERADOR DE REPORTES

El generador de reportes (Report Writer) es la herramienta que proporciona el manejador para la elaboración y manejo de reportes, éste se apoya en menús y consultas en SQL para su generación. La ventaja de utilizar éste consiste en que los reportes son elaborados en forma rápida y sencilla, apoyándose en ventanas y movimientos de campos y valores de campos (máscaras), las ventanas que maneja el reporteador son:

- Encabezados, Resultados y Pies.
 - * Encabezados página
 - * Encabezados reporte
 - * Resultados generales
 - * Agrupación de resultados
 - * Pies de página
 - * Pies de reporte
- Script de la consulta en SQL.
- Control de cortes de página.

Las desventajas que se tienen al trabajar con el reporteador, es la limitación en la presentación y el diseño del reporte, así como la limitación en cuanto al número de consultas que puede manejar (2).

Se diseñaron los reportes más sencillos en el reporteador, aquellos que sacan el contenido de las tablas de catálogo, reportes

que no requerían más de dos consultas para su elaboración, entre otros, los reportes sacados así son los mostrados en el punto 8.1 Reportes de catálogo y 8.2 Plantillas del árbol de menús.

IV.3.2 REPORTES CON OPEN-CLIENT.

Los reportes no elaborados con el reporteador fueron elaborados con los open-client del manejador, en otras palabras DB Library/C, que es un conjunto de rutinas y macros de C que permiten interactuar con el servidor.

El funcionamiento general de un programa escrito como open-client se define a partir de las siguientes funciones, el programa realiza el enlace con el servidor a partir de una llamada y se conecta con la base de datos a trabajar, la instrucción SQL es armada en un buffer y mandada al servidor a través de las DB-Library's, a través de éstas librerías el programa consulta el estado de sus peticiones, y también los estados de error que el servidor le manda, si la petición es una consulta los DB-Library's se encargan de mandar la información a buffers donde el programa toma la información para procesarla, la información es devuelta típicamente renglón a renglón o columna a columna, finalmente el programa realiza una llamada para desconectarse del servidor.

Los reportes elaborados en open-client, nos brindan mayores facilidades, como son el proceso de la información renglón a renglón, el poder manejar en un reporte más de una consulta SQL, la calidad de la presentación de la información es mayor, el control de los cortes es mayor, en otras palabras se pueden manejar reportes muy elaborados.

Algunos de los reportes manejados en open-client son: Impresión de la Nómina, Impresión de sobres, de tarjetas, de percepciones por concepto, de deducciones por concepto, reportes en papel y medios magnéticos al SAR, ISSSTE, FOVISSSTE, diversos reportes generados al fondo de ahorro, etc.

V - IMPLANTACION Y PRUEBAS

La implantación incluye todas aquellas actividades que tienen lugar para convertir el sistema anterior al nuevo. El nuevo sistema SINRECH esta reemplazando al anterior; la adecuada implantación es esencial para lograr un sistema confiable y que cumpla con las necesidades del Comité. Una implantación exitosa no garantiza el mejoramiento del Comité, pero su instalación inadecuada lo impedirá.

En la implantación existen tres aspectos importantes:

- 1.- Capacitación de personal.
- 2.- Procedimientos de conversión.
- 3.- Revisión después de la implantación.

CAPACITACION DE PERSONAL

La calidad de la capacitación del personal ayuda u obstruye la implantación exitosa del sistema. Aquellos que esten asociados con el sistema o afectados por el mismo deben conocer con detalle cuáles serán sus papeles, cómo pueden usar el sistema y qué hará o no hará el sistema. Tanto los operadores como los usuarios del sistema necesitan capacitación, en la tabla [5.1] se muestran las actividades de capacitación del sistema.

	CAPACITACION DEL PERSONAL DE SISTEMAS	CAPACITACION DEL PERSONAL USUARIO
Participantes	<ul style="list-style-type: none"> - Personal de preparación de datos - Personal de captura de datos - Operadores 	<ul style="list-style-type: none"> - Todos los usuarios directos o indirectos
Temas principales	<ul style="list-style-type: none"> - Uso de equipo - Identificación de los problemas del equipo - Procedimientos de ejecución - Procedimientos de trabajos - Programación de actividades - Mantenimiento del sistema 	<ul style="list-style-type: none"> - Uso del equipo (solo en caso necesario) - Familiarización con la aplicación - Captura y codificación de los datos - Manejo de datos, añadir, eliminar, editar, etc. - Recuperación de información - Uso de la información

TABLA [5.1] ACTIVIDADES DE CAPACITACION

PROCEDIMIENTOS DE CONVERSION

La conversión es el proceso de cambiar el sistema anterior al nuevo. Existen diferentes métodos para desarrollar la conversión de un sistema y los procedimientos empleados para asegurarse que se lleve a cabo adecuadamente, como se muestra en la tabla [5.2]

Existen cuatro métodos para llevar a cabo una conversión de sistemas. Cada método debe ser considerado a la luz de las ventajas que ofrece y los problemas que puede ocasionar. Pero muchas situaciones dictan el uso de un método en específico. En general, la conversión de sistemas debe realizarse lo más rápido posible. Los períodos de conversión largos aumentan la posible frustración y la dificultad de la tarea para las personas implicadas, incluyendo tanto a los analistas como a los usuarios.

METODO	DESCRIPCION	VENTAJAS	DESVENTAJAS
Sistemas paralelos	El sistema anterior se opera junto con el nuevo	Ofrece la máxima seguridad. Se puede recurrir al sistema anterior si se hallan errores en el nuevo o si ocurren problemas de uso.	Duplica los costos de operación. El nuevo sistema puede no ser juzgado justamente.
Conversión directa	El sistema anterior se reemplaza por el nuevo. La organización confía completamente en el nuevo	Obliga a los usuarios a que hagan trabajar el nuevo sistema. Hay beneficios inmediatos de los nuevos métodos y controles.	No hay otro sistema al cual recurrir si surgen dificultades con el nuevo. Requiere de la más cuidadosa planeación.
Enfoque piloto	Se implanta una versión de trabajo del sistema en una parte de la organización. Con base en la retroalimentación, se hacen cambios y el sistema se instala en el resto de la organización mediante uno de los demás métodos	Proporciona experiencia y prueba directa antes de la implantación.	Puede dar la impresión que el nuevo sistema no es confiable ni está libre de errores.
Por etapas	Se implanta el sistema de manera gradual a todos los usuarios	Permite a los primeros usuarios aprovechar las ventajas del sistema. Permite la capacitación y la instalación sin uso innecesario de recursos.	Un largo periodo de instalación provoca la duda en el usuario de si el proyecto marcha bien (demasiado entusiasmo) o mal (resistencia y falta de un juicio justo).

TABLA [5.2] METODOS PARA DESARROLLAR LA CONVERSION

Existe un plan de conversión que contenga la descripción de todas las actividades a realizar:

- 1.- Listar todos los archivo a convertir.
- 2.- Identificar todos los datos necesarios par construir los archivos nuevos durante la conversión.

3.- Listar todos los documentos nuevos y procedimientos que se usarán durante la conversión.

4.- Identificar todos los controles a usar durante la conversión. Establecer programas para verificación cruzada de los sistemas anterior y nuevo. Determinar cómo sabrán los miembros del equipo si algo no se ha llevado a cabo adecuadamente.

5.- Asignar responsabilidades para cada actividad.

6.- Verificar los tiempos para la conversión.

El plan de conversión debe anticipar los posibles problemas y la forma de solucionarlos.

REVISION DESPUES DE LA IMPLANTACION

Una vez que ya se tiene la implantación y también terminada la conversión, se hace una revisión del sistema conducida igualmente por los usuarios y los analistas. Esto no sólo es una práctica normal, sino que debe ser un proceso formal para determinar qué tan bien está funcionando el sistema, cómo ha sido aceptado y cuáles ajustes son necesarios. A continuación se nombran los diferentes tipos de pruebas que se diseñaron para verificar la confiabilidad del sistema.

V.1. DISEÑO DE PRUEBAS

Antes de empezar con las etapas finales de pruebas, es importante señalar que cada uno de los reportes, formas y soporte interno de las aplicaciones fueron probados individualmente al tiempo de su realización, por lo que en estas pruebas finales se verifica su funcionamiento en conjunto.

Para la realización de las pruebas estas se dividieron en varias etapas, las cuales son:

- a) Se llevó a cabo el plan de conversión.
- b) Capacitación de personal.
- c) Pruebas en paralelo.
- d) Revisión después de la implantación.

a) El plan de conversión (listados sus pasos anteriormente), se uso para pasar los datos del sistema anterior al nuevo, la idea de realizar el plan de conversión antes que la capacitación de los usuarios tuvo la finalidad de hacer algunas primeras corridas y pruebas generales del sistema, las cuales se procesaban con la información proveniente del sistema anterior, con lo cual se eliminaron los errores en las corridas ocasionados por la captura, asegurando así la confiabilidad en la parte de procesos terminales del sistema, al llevar a cabo la conversión se realizaron programas generales que pasaran automáticamente la información, con sólo correr dos procesos de carga, de esta manera se deja preparada la carga del nuevo sistema, para la siguiente fase de pruebas. En está fase se corren juntos los sistemas, y se comparan los resultados.

Los programas de carga deben generar los siguientes archivos, los cuales afectan a la tabla que tienen a su derecha.

ARCHIVO	TABLA QUE AFECTA
psgs_histórico.dat	psgs_histórico
cont_even.dat	cont_even
emp_suspen.dat	emp_suspen
emp_personal.dat	emp_personal
orden deducción.dat	orden_deducción
quinquenos.dat	quinquenos
emp_hor_irr.dat	emp_hor_irr
emp_dif_sueldo.dat	emp_dif_sueldo
emp_psgs.dat	emp_psgs
emp_lab.dat	emp_lab
orden percepción.dat	orden_percepción
fondo_ahorro.dat	fondo_ahorro
deuda_fa.dat	deuda_fa
deuda_apli.dat	deuda_apli
emp_beca.dat	emp_beca
plaza.dat	plaza
emp_mmys.dat	emp_mmys

ARCHIVO	TABLA QUE AFECTA
emp_bc.dat	emp_bc
sueldo.dat	sueldo
cat_jog.dat	cat_jog
área_todas_jz.dat	área_todas_jz
área_oc.dat	área_oc
ctrl_asis.dat	ctrl_asis
cat_puesto.dat	cat_puesto
cat_ze.dat	cat_ze
vacantes.dat	vacantes

Las demás tablas de catálogo deberán ser llenadas por el usuario y otras tablas son llenadas durante el proceso.

El script que realiza la carga del sistema nuevo contiene los siguientes pasos:

- 1.- De la base de datos del sistema anterior se crean los archivos antes mencionados con el formato de cada una de las tablas.
- 2.- Dichos archivos son subidos a la base de datos "carga" (es una base creada temporalmente para facilitar la carga del nuevo sistema) en donde no se verifica la consistencia de los datos.
- 3.- Corre el proceso que sube los datos de la base "carga" a la base "sinrech", el cual verifica la integridad referencial y consistencia de los datos por medio del soporte interno de las aplicaciones.

b) Para la capacitación del personal, se necesita de dos fases primordiales.

En la primera se capacitó al personal que dará el soporte técnico necesario para el mantenimiento del sistema ya en operación, en ésta fase se da capacitación sobre el hardware donde se implantó el sistema y además se capacitó a una persona (S.A.) sobre los aspectos técnicos internos del sistema.

En la segunda fase se requiere que la base de datos este llena con los datos reales del sistema anterior, estos datos corresponden

a una quincena anterior a la que se procesa, ya que vamos a capturar datos de la quincena a probar.

Por otra parte en esta fase se realizan dos pruebas al mismo tiempo, ya que la capacitación se va dando sobre las formas de captura y además se correrán los procesos de terminación, con lo que se verifica el buen funcionamiento de las formas de captura.

Esta fase se realizó por partes, es decir, se particionó al sistema y a los usuarios que realizan sus actividades con cada una de las partes, adiestrándolos en forma gradual, el sistema quedó dividido de la siguiente manera:

- 1.- Captura y control de incidencias.
- 2.- Control del personal.
- 3.- Control de asistencia.
- 4.- Control de movimientos de catálogo.
- 5.- Manejo general de la nómina.
- 6.- Manejo del SAR.
- 7.- Manejo del FOVISSSTE.
- 8.- Manejo del ISSSTE.
- 9.- Manejo del Fondo de Ahorro.

El orden en que aparecen listadas las partes, es el orden en que se realizaron las pruebas.

Una vez concluido el adiestramiento de todas las partes del sistema, se procede a llevar a cabo la última fase de pruebas la cual corresponde a las pruebas en paralelo.

V.2 PRUEBAS EN PARALELO

Para la realización de estas pruebas, se vuelven a correr los procedimientos de carga del sistema, capturando de esta forma la última información del sistema anterior, y de esta manera asegurar que los dos sistemas partirán conteniendo la misma información.

En esta última fase se lleva a cabo la captura y corrida de los sistemas al mismo tiempo, de esta manera se comienza a utilizar el nuevo sistema de manera global, y se asegura que si llegaran a

ocurrir dificultades se puede seguir trabajando con el sistema anterior, este método es muy seguro y garantiza que no se tendrán retrasos en el manejo de la información.

La fase anterior por razones políticas, y de seguridad se realizará hasta la segunda quincena de diciembre de 1994, quedando implantado y liberado el sistema definitivamente a partir de la primer quincena de enero de 1995.

Por último señalamos que la revisión de los errores se realizó inmediatamente después de la realización de cada una de las pruebas ejecutadas anteriormente, por lo que se espera no tener grandes problemas en la parte de pruebas en paralelo y solo agregar al sistema los módulos o reportes que el usuario detecte le hagan falta.

VI - CONCLUSIONES

Debido a la forma en que se desarrolló el sistema consideramos necesario dividir las conclusiones en tres partes principalmente:

a) Conclusiones Técnicas.

El presente trabajo proporciona una panorámica general en cuanto al diseño y desarrollo de sistemas utilizando el concepto de bases de datos relacionales y su implementación con DBMS que utiliza un lenguaje de cuarta generación en un ambiente cliente-servidor, incluyendo conceptos teórico y conjugándolos con los aspectos prácticos de manera general, intentando dar al lector un material que le sirva como base para la realización de otros sistemas.

Así pues, en teoría para tener un ambiente de bases de datos relacional es necesario cumplir con ciertas reglas, estas reglas en la práctica no todas se cumplen, tal es el caso de la normalización, ya que si se cumplía totalmente el rendimiento del sistema en general se ve afectado, incluyendo la capacidad de respuesta del equipo de cómputo, de tal forma que se tienen que crear herramientas de desarrollo como la utilizada en la implementación de este sistema, la cual ofrece opciones en el desarrollo de sistemas que sustituyen parcialmente las reglas ya mencionadas, entre estas se destacan los trigger's; los cuales proporcionan un manejo adecuado de la integridad referencial pese a que la normalización no sea satisfecha totalmente.

También, hemos podido observar que el desarrollo de un sistema

de software utilizando una herramienta de cuarta generación sea implementada más rápidamente teniendo como base un análisis ya definido.

La utilización de un reporteador proporcionado por el mismo manejador hizo más rápida la creación de los reportes sencillos necesarios, y aquellos más elaborados fueron creados a través de los open-client que también nos proporciona el manejador, a su vez los open-client fueron utilizados para la creación de procesos terminales.

En resumen, se procuró tener un sistema totalmente amigable al usuario y esto se logró debido a las facilidades que proporciona el manejador empleado, lo cual nos lleva a concluir lo importante que es contar con una buena herramienta de desarrollo.

b) Problemática del desarrollo.

Durante la elaboración del presente trabajo se encontraron dificultades de diversos tipos, entre las cuales podemos mencionar las siguientes:

- * Generalmente las necesidades de cualquier empresa por pequeñas que estas sean, van cambiando día a día de tal manera que lo que se necesitaba hace un año para satisfacer las necesidades de ese momento, quizás ya no sean las mismas que se requieren el día de hoy para el correcto cumplimiento de las funciones para la cual fue creada dicha empresa.

- * La falsa creencia por parte de la gente que la puesta en marcha de un sistema, va venir a sustituir en algunas ocasiones hasta en un 100% sus actividades laborales, por lo que se verían desplazados por el sistema. Este problema es muy grave ya que conlleva al ocultamiento de la información por parte de algunos departamentos, retrasando de esta forma el desarrollo del mismo. Aunque finalmente terminan aceptándolo, ya sea por convencimiento propio debido a la importancia del

mismo, o bien por que no se dan abasto en algunas ocasiones para el cumplimiento de su trabajo.

Sin embargo es importante mencionar que también existe gente que se da cuenta de la importancia que tendrá el desarrollo del sistema, aportando de esta forma todo cuanto este a su alcance para la pronta implementación del mismo.

* En algunas empresas no se cuenta con el personal necesario para el funcionamiento correcto del departamento de cómputo, por lo que en muchas ocasiones el personal encargado de un sistema se ve en la necesidad de compartir su tiempo en la implementación de otros sistemas o interviniendo en otras actividades propias del departamento que no corresponden a las actividades propias del sistema, por lo que se ve afectada la planeación original.

Así como la Gerencia de Proceso de Datos, igual papel de importancia tuvo la Gerencia de Recursos Humanos, ya que la información en su gran mayoría proviene de esta Gerencia, y es aquí en donde se nos indicaba como tenían que realizarse las cosas, que aspectos se tenían que considerar y cuales no.

Pese a todos los problemas anteriores el sistema se logró sacar adelante, por lo que es importante mencionar que en el desarrollo de todo sistema debe de existir una participación más completa por parte de todos los que de alguna u otra manera participan en el sistema y debe de ser visto como algo que más que traer un perjuicio va a realizarse como beneficio de la empresa y por consiguiente de todos cuantos laboran ahí.

Fue importante también observar lo complicado que resulta el poder manejar la información de tantas personas, por lo que se tuvo que cuidar que los datos devueltos por el sistema fueran los adecuados y que estos mostrarán la claridad adecuada para su posterior entendimiento, de acuerdo a los requerimientos de la Gerencia de Recursos Humanos.

c) Conclusiones finales.

Considerando todas las implicaciones que trae consigo un sistema es importante mencionar lo que nos deja esta experiencia después de su realización, enmarcando algunos puntos principales tales cómo, la importancia que tiene en el desarrollo un buen análisis y una buena base lógica, ya que es en este punto donde se ve el QUE y el COMO, y no el CON QUE, con esto nos referimos a la elaboración del diagrama relacional que podrá ser implementado en cualquier manejador de bases de datos.

Uno de los puntos más importantes para la elaboración de un sistema es la comunicación que se tiene entre los miembros del equipo de trabajo, así como el ambiente creado por el mismo grupo, ya que se pudo constatar que un equipo bien acoplado asegura que los resultados obtenidos se obtengan con mayor eficiencia y rapidez, y entonces el equipo sea por consiguiente más eficaz. Esto resulta vitalmente importante, si se considera que la formación actual de cualquier ingeniero independientemente del ámbito laboral del que se trate se realiza bajo una formación de trabajo en grupo, y si ya desde ahora se logra tener una buena adaptación y entendimiento con la gente los resultados se verían reflejados no sólo en la calidad, sino también en un buen ambiente de trabajo, beneficiando de esta forma a la empresa en general y por consiguiente a uno mismo.

Por otro lado para la elaboración del sistema es de suma importancia el contar con una herramienta de desarrollo de alta calidad, por lo que conocer los productos que nos faciliten de una alguna manera el desarrollo se deben de tener en un primer plano, considerando además de la calidad del producto los recursos económicos con los que se cuentan, pero no se debe de perder de vista, que un desarrollador no puede ni debe "casarse" con ningún

producto, ya que esto resultaría ser perjudicial para su crecimiento profesional, además se debe de tener en cuenta, que como profesionales el conocer y manejar productos nuevos dentro del mismo ámbito, no debe resultar muy difícil, por el contrario debe ser una experiencia muy agradable.

Finalmente podemos concluir que la elaboración de este trabajo nos proporcionó una satisfacción total ya que la implementación del mismo trajo consigo una gran cantidad de contrariedades mismas que fueron superadas totalmente en su momento, lo cual representa una primer experiencia para poner en marcha la realización de posteriores sistemas, ya que se puso además en práctica mucho de los puntos vistos esencialmente en los salones de clase, comprobando de esta forma que la formación recibida en la escuela resulta ser muy completa y de alta calidad, lo cual nos obliga a trabajar cada día más fuerte y a superarnos constantemente para llevar siempre en alto el orgullo de haber egresado de la Máxima Casa de Estudios y de manera muy especial de la Facultad de Ingeniería.

FOR MI RAZA HABLARA EL ESPIRITU
MEJICO, D.F. 23 DE OCTUBRE DE 1994

BIBLIOGRAFIA

LIBROS

- Ingeniería del Software, un enfoque práctico
Roger S. Pressman
Segunda Edición, McGraw-Hill

- Análisis y Diseño de Sistemas de Información
James A. Senn
Segunda Edición, McGraw-Hill

- Introducción a los Sistemas de Base de Datos
C.J. Date
Editorial Adison-Wesley Iberoamericana

- Introducción a los sistemas operativos
Harvey M. Deitel
Editorial Adison-Wesley Iberoamericana

- El lenguaje de programación 'C'
Brian W. Kernighan, Dennis M. Ritchie
Editorial Prentice Hall

- Lenguaje C, Programación avanzada
Herbert Schildt
McGraw-Hill

MANUALES

- System Administration Guide
Manual del manejador SYBASE

- APT-Workbench
Manual del manejador SYBASE

- Data-Workbench User's Guide
Manual del manejador SYBASE

- Command Reference
Manual del manejador SYBASE

- DB-Library
Manual del manejador SYBASE

- Manual de procedimientos de la Gerencia de Recursos Humanos
C.A.P.F.C.E., Febrero 1994.

APENDICE A - TRIGGER'S

AREA_OC_TD

```
create trigger area_oc_td
on area_oc
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos*/
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
    print "solo se puede borrar una area de oficinas centrales a la vez"
    rollback transaction
    return
  end
  /* verificamos que el numero de area y numero de jog no exista en
  la tabla plaza */
  if ((select count(*) from plaza, deleted
    where (plaza.num_area = deleted.num_area and
    plaza.num_jog = deleted.num_jog)) > 0)
  begin
    print "No se puede dar de baja una area que, esta asignado a una plaza."
    rollback transaction
  end
end
```

AREA_OC_TIU

```
create trigger area_oc_tiu
on area_oc
for insert, update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados
  sin ningun dato fue modificado o insertado, regresamos
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
    print "solo se puede dar de alta una area a la vez"
    rollback transaction
    return
  end */
```

```

if update(num_jog)
begin
/* aseguramos que el num_jog exista en el catalogo de JOG */
if ((select count(*) from cat_jog jog, inserted i
     where jog.num_jog = i.num_jog) = 0)
begin
print 'El numero de jog proporcionado no existe en el catalogo de JOG'
rollback transaction
return
end
end
if update(num_area)
begin
/* checamos si es una actualizacion para actualizar en cascada */
if ((select count(*) from deleted) > 0)
begin
/* realizamos una actualizacion en cascada sobre la tabla plaza */
update plaza
set plaza.num_area = inserted.num_area
from plaza, inserted, deleted
where plaza.num_area = deleted.num_area and
      plaza.num_jog > 32
end
end
end

```

AREA_TODAS_JZ_TIU

```

create trigger area_todas_jz_tiu
on area_todas_jz
for insert, update
as
declare @num_renglon int
select @num_renglon = @@rowcount
if @num_renglon = 0
return
/* Encuentra cuantos renglones fueron modificados
sin ningun dato fue modificado o insertado, regresamos
if @num_renglon > 1
begin
print "solo se puede dar de alta una area a la vez"
rollback transaction
return
end */
if update(num_area)
begin
/* checamos si es una actualizacion, si es asi la rechazamos */
if ((select count(*) from deleted) > 0)
begin
print "No estan permitidos los cambios, sobre este campo"
rollback transaction
return
end
end
end

```

AREA_TODASJE_TD

```
create trigger area_todasjz_td
on area_todas_jz
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos*/
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
    print "solo se puede borrar un banco a la vez"
    rollback transaction
    return
  end
  /* verificamos que el numero de area no exista en la tabla plaza */
  if ((select count(*) from plaza, deleted
    where (plaza.num_area = deleted.num_area and
    plaza.num_jog <= 32)) > 0)
  begin
    print "No se puede dar de baja una area que,
    esta asignada a una plaza."
    rollback transaction
  end
end
```

BECA_IMPTE_TD

```
create trigger beca_imppte_td
on beca_imppte
for delete
as
  declare @num_renglon int
  /* Encuentra Cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos*/
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
    print "solo se puede borrar un banco a la vez"
    rollback transaction
    return
  end
  /* verificamos que el nivel de beca a borra no exista en la tabla emp_beca */
  if ((select count(*) from emp_beca, deleted
    where emp_beca.niv_beca = deleted.niv_beca) > 0)
  begin
    print "No se puede dar de baja un nivel de beca desde la tabla beca_imppte,
    que este asigando a un empleado."
    rollback transaction
  end
end
```

BECA_IMPTE_TIU

```

create trigger beca_impnte_tiu
on beca_impnte
for insert, update
as
  declare @num_renglon int

/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount

/* sin ningun dato fue modificado o insertado, regresamos */
if @num_renglon = 0
  return

if update(niv_beca)
begin
  /* checamos si es una actualizacion para actualizar en cascada */
  if((select count(*) from deleted) > 0)
  begin
    /* realizamos una actualizacion en cascada sobre la tabla emp_beca */
    update emp_beca
    set emp_beca.niv_beca = inserted.niv_beca
    from emp_beca, inserted, deleted
    where emp_beca.niv_beca = deleted.niv_beca
    and
  end
end

```

CAT_BANCOS_TD

```

create trigger cat_bancos_td
on cat_bancos
for delete
as
  declare @num_renglon int

/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* si ningun renglon fue afectado, regresamos */
if @num_renglon = 0
  return
if @num_renglon > 1
begin
  print "solo se puede borrar un banco a la vez"
  rollback transaction
  return
end
/* verificamos que el banco a borra no exista en la tabla de mmys */
if ((select count(*) from emp_mmys, deleted
  where emp_mmys.banco = deleted.banco) > 0)
begin
  print "No se puede dar de baja un banco desde el cat_bancos,
    que este asigando a un empleado de mmys."
  rollback transaction
end

```

CAT_BANCOS_TIU

```

create trigger cat_bancos_tiu
on cat_bancos
for insert, update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* sin ningun dato fue modificado o insertado, regresamos */
  if @num_renglon = 0
    return
  if update(banco)
begin
  /* checamos si es una actualizacion para actualizar en cascada */
  if ((select count(*) from deleted) > 0)
  begin
    /* realizamos una actualizacion en cascada sobre la tabla emp_mmys */
    update emp_mmys
    set emp_mmys.banco = inserted.banco
    from emp_mmys, inserted, deleted
    where emp_mmys.banco = deleted.banco
  end
end
end

```

CAT_JOG_TD

```

create trigger cat_jog_td
on cat_jog
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos */
  if @num_renglon = 0
    return
  if @num_renglon > 1
begin
  print "solo se puede borrar un banco a la vez"
  rollback transaction
  return
end
  /* verificamos que el numero de jog no exista en la tabla plaza */
  if ((select count(*) from plaza, deleted
  where plaza.num_jog = deleted.num_jog) > 0)
begin
  print "No se puede dar de baja un JOG que,
  esta asignado a una plaza."
  rollback transaction
end
  /* verificamos que el numero de JOG no exista en la tabla area_oc */
  if ((select count(*) from area_oc, deleted
  where area_oc.num_jog = deleted.num_jog) > 0)
begin
  print "No se puede dar de baja un JOG, que esta siendo utilizada en la
  tabla de area_oc."
  rollback transaction
end
end

```

CAT_JOG_TI

```

create trigger cat_jog_ti
on cat_jog
for insert
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  /* sin ningun dato fue modificado o insertado, regresamos
  if @num_renglon > 1
  begin
    print "solo se puede dar de alta un JOG a la vez"
    rollback transaction
    return
  end */
  /* aseguramos que la zona_ec exista en el catalogo de zona economica */
  if(select count(*) from cat_ze ze, inserted i
  where ze.zona_ec = i.zona_ec) = 0
  begin
    print 'La zona economica proporcionada no existe
    en el catalogo de zona economica'
    rollback transaction
    return
  end
end

```

CAT_JOG_TU

```

create trigger cat_jog_tu
on cat_jog
for update
as
  declare @num_renglon int
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  /* Encuentra cuantos renglones fueron modificados */
  /* sin ningun dato fue modificado o insertado, regresamos
  if @num_renglon > 1
  begin
    print "solo se puede dar de alta un JOG a la vez"
    rollback transaction
    return
  end */
  if update(zona_ec)
  begin
    /* aseguramos que la zona_ec exista en el catalogo de zona economica */
    if(select count(*) from cat_ze ze, inserted i
    where ze.zona_ec = i.zona_ec) = 0
    begin
      print 'La zona economica proporcionada no existe
      en el catalogo de zona economica'
      rollback transaction
      return
    end
  end
end

```

```
if update(num_jog)
begin
/* realizamos una actualizacion en cascada */
update plaza
set plaza.num_jog = inserted.num_jog
from plaza, inserted, deleted
where plaza.num_jog = deleted.num_jog
update area_oc
set area_oc.num_jog = inserted.num_jog
from area_oc, inserted, deleted
where area_oc.num_jog = deleted.num_jog
end
```

CAT_PERCEPCION_TD

```
Creates trigger cat_percepcion_td
on cat_percepcion
for delete
as
begin
if @@rowcount = 0
return
if (select count(*) from cat_percepcion, orden_percepcion
where cat_percepcion.num_percepcion =
orden_percepcion.num_percepcion) != 0
begin
print 'existen ordenes de percepcion con esta clave'
rollback transaction
end
/* Se actualiza PERCEPCION_ACUM, PERCEPCION_CONDICION borrando el
registro con la clave que se elimino del catalogo de percepciones
*/
Delete percepcion_acum
from percepcion_acum,deleted
where percepcion_acum.num_percepcion = deleted.num_percepcion
Delete percepcion_cond
from percepcion_cond,deleted
where percepcion_cond.num_percepcion = deleted.num_percepcion
end
```

CAT_PERCEPCION_TU

```
Create trigger cat_percepcion_tu
on cat_percepcion
for update
as
begin
if @@rowcount = 0
return
/*Se actualiza PERCEPCION_ACUM, PERCEPCION_CONDICION, PERCEPS_APLI Y
* ORDEN_PERCEPCION sustituyendo la clave que se actualizo en
* el catalogo de percepciones */
Update percepcion_acum
```



```

set percepcion_acum.num_percepcion = inserted.num_percepcion
from percepcion_acum, inserted, deleted
where percepcion_acum.num_percepcion = deleted.num_percepcion
Update percepcion_cond
set percepcion_cond.num_percepcion = inserted.num_percepcion
from percepcion_cond, inserted, deleted
where percepcion_cond.num_percepcion = deleted.num_percepcion
Update percepcion_apli
set percepcion_apli.num_percepcion = inserted.num_percepcion
from percepcion_apli, inserted, deleted
where percepcion_apli.num_percepcion = deleted.num_percepcion
Update orden_percepcion
set orden_percepcion.num_percepcion = inserted.num_percepcion
from orden_percepcion, inserted, deleted
where orden_percepcion.num_percepcion = deleted.num_percepcion
end

```

CAT_PUESTO_TD

```

create trigger cat_puesto_td
on cat_puesto
for delete
as
declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* si ningun renglon fue afectado, regresamos */
if @num_renglon = 0
return
if @num_renglon > 1
begin
print "solo se puede borrar un banco a la vez"
rollback transaction
return
end
/* verificamos que el numero de puesto no exista en la tabla plaza */
if ((select count(*) from plaza, deleted
where plaza.num_puesto = deleted.num_puesto) > 0)
begin
print "No se puede dar de baja un puesto que,
esta asignado a una plaza."
rollback transaction
end
/* verificamos que el numero de puesto no exista en la tabla sueldo */
if ((select count(*) from sueldo, deleted
where sueldo.num_puesto = deleted.num_puesto) > 0)
begin
print "No se puede dar de baja un puesto,
que esta siendo utilizada en la tabla de sueldo."
rollback transaction
end
end

```

CAT_PUESTO_TIU

```

create trigger cat_puesto_tiu
on cat_puesto
for insert, update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* sin ningun dato fue modificado o insertado, regresamos */
  if @num_renglon = 0
    return
  /* if @num_renglon > 1
  begin
  print "solo se puede dar de alta un empleado a la vez"
  rollback transaction
  return
  end */
if update(num_puesto)
begin
  /* verificamos si es una actualizacion para actualizar en cascada */
  if ((select count(*) from deleted) > 0)
  begin
    /* actualizamos en cascada */
    update plaza
    set plaza.num_puesto = inserted.num_puesto
    from plaza, inserted, deleted
    where plaza.num_puesto = deleted.num_puesto
    update sueldo
    set sueldo.num_puesto = inserted.num_puesto
    from sueldo, inserted, deleted
    where sueldo.num_puesto = deleted.num_puesto
  end
end

```

CAT_ZE_TD

```

create trigger cat_ze_td
on cat_ze
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos */
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
  print "solo se puede borrar un banco a la vez"
  rollback transaction
  return
  end
  /* verificamos que la zona economica no exista en la tabla de cat_jog */
  if ((select count(*) from cat_jog, deleted

```

```

        where cat_jog.zona_ec = deleted.zona_ec) > 0)
begin
    print "No se puede dar de baja una zona economica, que esta siendo
    utilizada en el cat_jog."
    rollback transaction
end
/* verificamos que la zona economica no exista en la tabla sueldo */
if ((select count(*) from sueldo, deleted
    where sueldo.zona_ec = deleted.zona_ec) > 0)
begin
    print "No se puede dar de baja una zona economica,
    que esta siendo utilizada en la tabla de sueldo."
    rollback transaction
end
end

```

CAT_ZE_TIU

```

create trigger cat_ze_tiu
on cat_ze
for insert, update
as
    declare @num_renglon int,
            @cuenta int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* sin ningun dato fue modificado o insertado, regresamos */
if @num_renglon = 0
    return
/* if @num_renglon > 1
begin
    print "solo se puede dar de alta una zona economica a la vez"
    rollback transaction
    return
end
*/
if update(zona_ec)
begin
    /* checamos si es una actualizacion para actualizar en cascada */
    if((select count(*) from deleted) > 0)
    begin
        /* realizamos una actualizacion en cascada sobre la tabla cat_jog */
        update cat_jog
        set cat_jog.zona_ec = inserted.zona_ec
        from cat_jog, inserted, deleted
        where cat_jog.zona_ec = deleted.zona_ec
        update sueldo
        set sueldo.zona_ec = inserted.zona_ec
        from sueldo, inserted, deleted
        where sueldo.zona_ec = deleted.zona_ec
    end
end
end

```

CONT_EVEN_TIU

```

create trigger cont_even_tiu
on cont_even
for insert, update
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
  return
if update(num_emp)
  begin
/* aseguramos que el empleado exista en la tabla emp_lab */
if(select count(*) from emp_lab e, inserted i
  where e.num_emp = i.num_emp) = 0
  begin
    print "El numero de empleado no esta dado de alta"
    rollback transaction
    return
  end
end
/* si el porcentaje es cero no se permite la actualizacion o la alta */
if update(porcentaje_sueldo)
  begin
if exists(select * from inserted i
  where i.porcentaje_sueldo <= 0)
  begin
    print "El porcentaje del sueldo debe ser mayor que cero"
    rollback transaction
    return
  end
end
/* verificamos si hay actualizaciones sobre el campo porcentaje de sueldo, en
 caso afirmativo reportarlo como cambio */

```

CTRL_ASIS_TI

```

create trigger ctrl_asis_ti
on ctrl_asis
for insert
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* sin ningun dato fue modificado o insertado, regresamos */
if @num_renglon = 0
  return
/* aseguramos que el empleado exista en el catalogo de empleados de B o C */
if(select count(*) from emp_bc ebc, inserted i
  where ebc.num_emp = i.num_emp) = 0
  begin
    print 'El empleado no existe o no tiene control de asistencia'
  end

```

```
rollback transaction
return
end
/* aseguramos que si la clave de incidente no es nula esta exista en el
catalogo de incidentes */
if (select count(*) from inserted i
where i.cve_incidente is not null) = 1
begin
if(select count(*) from cat_incidente ci, inserted i
where ci.cve_incidente = i.cve_incidente) = 0
begin
print 'El incidente que se desa dar de alta
no existe en el catalogo de incidentes'
rollback transaction
return
end
end
/* aseguramos que si la clave del pja no es nula esta exista en el
catalogo de pja's */
if (select count(*) from inserted i
where i.cve_pja is not null) = 1
begin
if(select count(*) from cat_pja cp, inserted i
where cp.cve_pja = i.cve_pja) = 0
begin
print 'El pja que se desa dar de alta
no existe en el catalogo de pjas'
rollback transaction
return
end
end
end
```

DEP_ECONOMICOS_TIU

```
create trigger dep_economicos_tiu
on dep_economicos
for insert, update
as
declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
return
/* aseguramos que el empleado exista en la tabla emp_lab */
if(select count(*) from emp_lab e, inserted i
where e.num_emp = i.num_emp) = 0
begin
print 'El numero de empleado no esta dado de alta'
rollback transaction
return
end
```

DEUDA_APLI_TI

```

create trigger deuda_apli_ti
on deuda_apli
for insert
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
  return
if not exists(select * from deuda_fa dfa, inserted i
              where dfa.num_emp = i.num_emp)
begin
  print "El empleado no tiene un prestamo en el fondo de ahorro"
  rollback transaction
  return
end

```

DEUDA_FA_TI

```

create trigger deuda_fa_ti
on deuda_fa
for insert
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
  return
if not exists(select * from fondo_ahorro fa, inserted i
              where fa.num_emp = i.num_emp)
begin
  print "El empleado no esta en el fondo de ahorro"
  rollback transaction
  return
end

```

EMP_BC_TIU

```

create trigger emp_bc_tiu
on emp_bc
for insert, update
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
  return
/* verificamos que no existan cambios sobre la llave primaria */
if update(num_emp)
  if ((select count(*) from deleted) > 0)
  begin
    print "NO estan permitidos los cambios sobre este campo"
    rollback transaction
  end

```

```

        return
    end
    else
    begin
    /*aseguramos que el empleado no exista en la tabla de empleados de nmys */
    if(select count(*) from emp_nmys e, inserted i
        where e.num_emp = i.num_emp) > 1
        begin
        print "El numero de empleado proporcionado ya esta registrado como un
        empleado de mandos medios"
        rollback transaction
        return
        end
    end
    end
end

```

EMP_BECA_TIU

```

create trigger emp_beca_tiu
on emp_beca
for insert, update
as
    declare @num_renglon int
    /* Encuentra cuantos renglones fueron modificados */
    select @num_renglon = @@rowcount
    if @num_renglon = 0
        return
    if((update(num_emp) and update(num_beca)) or (update (num_emp)))
    begin
    /* aseguramos que el empleado exista en la tabla emp_lab */
    if(select count(*) from emp_lab e, inserted i
        where e.num_emp = i.num_emp) = 0
        begin
        print 'El numero de empleado no esta dado de alta'
        rollback transaction
        return
        end
    /* aseguramos que el empleado no tenga mas de dos becas*/
    if((select count(*) from emp_beca e, inserted i
        where e.num_emp = i.num_emp ) > 2)
        begin
        print 'El empleado tiene ya asignadas dos becas'
        rollback transaction
        return
        end
    end
end
else
    if update(num_beca)
    begin
    /* aseguramos que la llave primaria sea unica */
    if((select count(*) from emp_beca e, inserted i
        where e.num_emp = i.num_emp and
        e.num_beca = i.num_beca) > 1)
        begin
        print 'El empleado tiene ya asignada la beca proporcionada'
        rollback transaction
        return
        end
    end
end

```

```
/* aseguramos que el empleado no tenga mas de dos becas*/
if((select count(*) from emp_beca e, inserted i
   where e.num_emp = i.num_emp ) >= 2)
begin
  print 'El empleado tiene ya asignadas dos becas'
  rollback transaction
  return
end
end
if update(niv_beca)
begin
  /* aseguramos que el nivel de beca exista en el catalogo de beca importe */
  if((select count(*) from beca_impnte e, inserted i
     where e.niv_beca = i.niv_beca) = 0)
  begin
    print 'El nivel de beca no existe en el catalogo'
    rollback transaction
    return
  end
end
end
```

EMP_CAMBIO_TI

```
create trigger emp_cambio_ti
on emp_cambio
for insert
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  /* Se modifica la tabla rep_abc */
  if exists(select * from rep_abc ra, inserted i
     where ra.num_emp = i.num_emp)
  begin
    update rep_abc
    set cambio = i.fecha_cambio
    from rep_abc ra, inserted i
    where ra.num_emp = i.num_emp
  end
```

EMP_DIF_SUELDO_TIU

```
create trigger emp_dif_sueldo_tiu
on emp_dif_sueldo
for insert, update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  if update(num_emp)
  begin
```



```

/* aseguramos que el empleado exista en la tabla emp_lab */
if(select count(*) from emp_lab e, inserted i
   where e.num_emp = i.num_emp) = 0
  begin
    print "El numero de empleado no esta dado de alta"
    rollback transaction
    return
  end
end

```

EMP LAB_TD

```

create trigger emp_lab_td
on emp_lab
for delete
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* si ningún renglon fue afectado, regresamos*/
if @num_renglon = 0
  return
if @num_renglon > 1
begin
  print "solo se puede borrar un empleado a la vez"
  rollback transaction
  return
end
/* Procedemos a borrar toda la informacion del empleado que se da de baja
almacenando solo aquella que sea necesaria en la tabla emp_bajas */
/* insertamos los datos necesarios de la tabla emp_lab*/
/* insertamos los datos necesarios de la tabla emp_personal*/
insert into emp_bajas
(num_emp, apell_pat, apell_mat, nombres, calle, numero, colonia,
localidad, estado, cp, telefono, nacionalidad, edo_civil, sexo,
num_issste, rfc_l, rfc_n, rfc_h, fecha_ing, fecha_dlab, fecha_base, ant_gf,
plaza, fecha_baja)
select d.num_emp, p.apell_pat, p.apell_mat, p.nombres, p.calle, p.numero,
p.colonia, p.localidad, p.estado, p.cp, p.telefono, p.nacionalidad,
p.edo_civil, p.sexo, p.num_issste, d.rfc_l, d.rfc_n, d.rfc_h, d.fechaing,
d.fecha_dlab, d.fecha_base, d.ant_gf, d.plaza, getdate()
from emp_personal p, Deleted d where p.num_emp = d.num_emp
/* insertamos los datos necesarios de la tabla plaza*/
/* update emp_bajas set b.num_puesto = p.num_puesto, b.num_jog =
p.num_jog, b.num_area = p.num_area from emp_bajas b, plaza p, deleted d
where p.plaza = d.plaza and b.num_emp = d.num_emp */
/* insertamos los datos necesarios de la tabla plaza y el salario de la
tabla sueldo*/
update emp_bajas
set b.num_puesto = p.num_puesto, b.num_jog = p.num_jog,
   b.num_area = p.num_area, b.salario = s.salario,
   b.porcentaje_sueldo = 1
from emp_bajas b, sueldo s, plaza p, c
   at_jog j, deleted d
where b.num_emp = d.num_emp and
      p.plaza = d.plaza and
      p.num_jog = j.num_jog and

```

```
j.zona_ec = s.zona_ec and
p.categoria = s.categoria and
p.num_puesto = s.num_puesto
/* se actualiza el sueldo si es que el empleado es eventual */
if exists(select * from cont_even ce, deleted d
         where ce.num_emp = d.num_emp)
begin
  update emp_bajas
  set b.salario = b.salario * ce.porcentaje_sueldo,
      b.porcentaje_sueldo = ce.porcentaje_sueldo
  from emp_bajas b, cont_even ce, deleted d
  where b.num_emp = d.num_emp and
        ce.num_emp = b.num_emp
end
/* Se actualiza la tabla de rep_abc */
if exists(select * from rep_abc ra, deleted d
         where ra.num_emp = d.num_emp)
begin
  update rep_abc
  set baja = getdate()
  from rep_abc ra, deleted d
  where ra.num_emp = d.num_emp
end
/* empieza la eliminacion en cascada */
delete emp_personal /* se pone el nombre de la tabla ya que
                  se involucra otra tabla */
  from emp_personal e, deleted d
  where e.num_emp = d.num_emp
delete cont_even
  from cont_even e, deleted d
  where e.num_emp = d.num_emp
delete emp_hor_irr
  from emp_hor_irr e, deleted d
  where e.num_emp = d.num_emp
delete dep_economicos
  from dep_economicos e, deleted d
  where e.num_emp = d.num_emp
delete emp_psgs
  from emp_psgs e, deleted d
  where e.num_emp = d.num_emp
delete emp_beca
  from emp_beca e, deleted d
  where e.num_emp = d.num_emp
delete emp_mnys
  from emp_mnys e, deleted d
  where e.num_emp = d.num_emp
delete emp_bc
  from emp_bc e, deleted d
  where e.num_emp = d.num_emp
delete emp_wkend
  from emp_wkend e, deleted d
  where e.num_emp = d.num_emp
delete orden_percepcion
  from orden_percepcion e, deleted d
  where e.num_emp = d.num_emp
/*delete percepcion_apli
  from percepcion_apli e, deleted d
  where e.num_emp = d.num_emp */
delete orden_deducccion
  from orden_deducccion e, deleted d
```

```

where e.num_emp = d.num_emp
/*delete deduccion apli
from deduccion_apli e, deleted d
where e.num_emp = d.num_emp */
delete emp_suspen
from emp_suspen es, deleted d
where es.num_emp = d.num_emp
delete emp_dif_sueldo
from emp_dif_sueldo eds, deleted d
where eds.num_emp = d.num_emp
delete ctrl_asis
from ctrl_asis ca, deleted d
where ca.num_emp = d.num_emp
delete fondo_ahorro
from fondo_ahorro fa, deleted d
where fa.num_emp = d.num_emp

```

EMP_LAB_TI

```

create trigger emp_lab_ti
on emp_lab
for insert
as
declare @num_renglon int,
        @puesto      char(1),
        @num_emp     int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* si ningun renglon fue afectado, regresamos*/
if @num_renglon = 0
return
/* aseguramos que el num_emp no exista en la tabla de emp_bajas */
if (select count(*) from emp_bajas e, inserted i
     where e.num_emp = i.num_emp) > 0
begin
print "El numero de empleado proporciona no puede ser utilizado,
por estar asignado a un empleado en el historico de bajas"
rollback transaction
return
end
/* aseguramos que la plaza proporcionada exista en el catalogo de plazas */
if (select count(*) from plaza p, inserted i
     where p.plaza = i.plaza) = 0
begin
print "La plaza proporcionada no existe en el catalogo de plazas"
rollback transaction
return
end
/* aseguramos que la plaza sea unica, esto es no este asignada a otro empleado
, solo puede ser asignada si el propietario esta en la relacion
EMP_PSGS o en EMP_SUSPEND*/
select @num_renglon = count(*) from emp_lab e, inserted i
     where e.plaza = i.plaza
if @num_renglon > 2
begin
print "La plaza proporcionada ya esta asignada a otro empleado"
rollback transaction

```

```

return
end
if @num_renglon = 2
begin
select @num_emp = e1.num_emp
from emp_lab e1, inserted i
where e1.plaza = i.plaza and
      e1.num_emp <> i.num_emp
if not exists (select * from emp_paga
              where num_emp = @num_emp)
begin
if not exists(select * from emp_suspen
              where num_emp = @num_emp)
begin
print "La plaza proporcionada ya esta asignada a otro empleado"
rollback transaction
return
end
end
end
/* si no hubo ningun error procedemos a insertar el numero de empleado, ya sea
en la tabla de mmys o de bc, dependiendo del tipo de plaza, y de acuerdo al tipo
de puesto de la tabla CAT_PUESTO. Esto implica posteriormente realizar una
actualizacion en cualquiera de las dos tablas para completar la informacion
restante; en lugar de llevar a cabo una insercion en dichas tablas. Esto es
realizado con el fin de tener consistencia en la base de datos, ya que un empleo
que exista en la tabla emp_lab forzosamente debera existir ya sea en la tabla de
emp_mmys o en la tabla de emp_bc. */
select @puesto = (select tipo_puesto
                  from cat_puesto c, plaza p, inserted i
                  where i.plaza = p.plaza and
                        p.num_puesto = c.num_puesto)
if(@puesto = 'S' or @puesto = 'M')
begin
insert into emp_mmys (num_emp)
select i.num_emp
from inserted i
end
else
if(@puesto = 'B' or @puesto = 'C')
begin
insert into emp_bc (num_emp)
select i.num_emp
from inserted i
end
else
begin
print " La plaza proporcionada no tiene un tipo de puesto asociado,
verifique su informacion"
rollback transaction
return
end
/* Se hace un insert sobre rep_abc, poniendo a alta como la fecha de
ingreso que el trabajador tiene en emp_lab */
if not exists(select * from rep_abc ra, inserted i
              where ra.num_emp = i.num_emp)
begin
insert into rep_abc (num_emp, cve_rep, alta, baja, cambio)
select i.num_emp, 'A', i.fecha_ing, null, null
from inserted i

```

```
end
else
begin
  delete rep_abc
  from rep_abc ra, inserted i
  where ra.num_emp = i.num_emp
  insert into rep_abc (num_emp, cve_rep, alta, baja, cambio)
  select i.num_emp, 'A', i.fecha_ing, null, null
  from inserted i
end
```

EMP_LAB_TU

```
create trigger emp_lab_tu
on emp_lab
for update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos*/
  if @num_renglon = 0
  return
  if @num_renglon > 1
  begin
    print "solo se puede modificar un empleado a la vez"
    rollback transaction
    return
  end
  /* aseguramos que la llave primaria NUNCA sea modificada */
  if update(num_emp)
  begin
    print "El numero de empleado no puede ser modificado"
    rollback transaction
    return
  end
  if update(plaza)
  begin
    /* aseguramos que la plaza proporcionada exista en el catalogo de plazas */
    if (select count(*) from plaza p, inserted i
        where p.plaza = i.plaza) = 0
    begin
      print "La plaza proporcionada no existe en el catalogo de plazas"
      rollback transaction
      return
    end
    /* aseguramos que la plaza sea unica, esto es no este asignada a otro
    empleado */
    select @num_renglon = count(*) from emp_lab e, inserted i
    where e.plaza = i.plaza
    if @num_renglon > 2
    begin
      print "La plaza proporcionada ya esta asignada a otro empleado"
      rollback transaction
      return
    end
  end
```

```

if @num_renglon = 2
begin
  if (select count(*) from emp_pags e, emp_suspen s, plaza p, inserted i
      where i.plaza = p.plaza and
            (p.propietario = e.num_emp or
             p.propietario = s.num_emp)) = 0
      begin
        print "La plaza proporcionada ya esta asignada a otro empleado"
        rollback transaction
        return
      end
    end
  end
/* Si no hubo errores, insertamos en la tabla emp_cambios la plaza que el
empleado deja */
insert into emp_cambio
select d.num_emp, getdate(), p.plaza, p.num_puesto, s.salario, p.num_jog,
p.num_area, 1.0
from emp_lab l, plaza p, sueldo s, cat_jog j, deleted d
where (l.num_emp = d.num_emp and
      p.plaza = d.plaza and
      j.num_jog = p.num_jog and
      s.num_puesto = p.num_puesto and
      s.categoria = p.categoria and
      s.zona_ec = j.zona_ec )
/* actualizamos la tabla emp_cambio, en el porcentaje del sueldo
si es que el empleado es del tipo eventual o interino */
if exists(select * from cont_even ce, deleted d
          where ce.num_emp = d.num_emp)
begin
  update emp_cambio
  set ec.porcentaje_sueldo = ce.porcentaje_sueldo
  from emp_cambio ec, cont_even ce, deleted d
  where ce.num_emp = d.num_emp and
        ec.num_emp = ce.num_emp and
        ec.fecha_cambio = getdate()
end
end

```

EMP_MNYS_TIU

```

create trigger emp_mnys_tiu
on emp_mnys
for insert,update
as
declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
return
if update(num_emp)
/* checamos que no existan cambios sobre la llave primaria */
if((select count(*) from deleted) > 0) /* es una actualizacion */
begin
  print "No estan permitidos los cambios sobre este campo"
  rollback transaction
  return
end
end

```

```

else /* es una insercion */
begin
  /* aseguramos que el empleado no exista en la tabla de empleado de base
  o confianza */
  if(select count(*) from emp_bc e, inserted i
     where e.num_emp = i.num_emp) > 1
  begin
    print"El numero de empleado proporcionado ya esta registrado como un
    empleado de base o confianza"
    rollback transaction
    return
  end
end
if update(banco)
/* aseguramos que el banco proporcionado exista en el catalogo */
if(select count(*) from cat_bancos b, inserted i
   where b.banco = i.banco) = 0
begin
  print "La clave del banco proporcionado no existe en el catalogo"
  rollback transaction
  return
end
end

```

EMP_PERSONAL_TI

```

create trigger emp_personal_ti
on emp_personal
for insert
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
    print "solo se puede dar de alta un empleado a la vez"
    rollback transaction
    return
  end
  /* aseguramos que la llave primaria sea unica */
  if(select count(*) from emp_personal e, inserted i
     where e.num_emp = i.num_emp) > 1
  begin
    print 'El numero de empleado proporciona ya existe'
    rollback transaction
    return
  end
  /* aseguramos que el empleado exista en EMP LAB */
  if (select count(*) from emp_lab e, inserted i
     where e.num_emp = i.num_emp) = 0
  begin
    print 'Empleado sin nombramiento'
    rollback transaction
    return
  end
end

```

EMP_PERSONAL_TU

```
create trigger emp_personal_tu
on emp_personal
for update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  /* aseguramos que la llave primaria NUNCA sea modificada */
  if update(num_emp)
begin
  print 'El numero de empleado no puede ser modificado'
  rollback transaction
  return
end
```

EMP_PSGS_TD

```
create trigger emp_psgs_td
on emp_psgs
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos */
  if @num_renglon = 0
    return
  if @num_renglon > 1
begin
  print "solo se puede dar de baja un empleado a la vez"
  rollback transaction
  return
end
```

EMP_PSGS_TIU

```
create trigger emp_psgs_tiu
on emp_psgs
for insert, update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  if (update(num_emp) and update(fecha_ini))
  or (update(num_emp))
  /* es una alta la que se esta llevando a cabo o se actualizaron los dos
  campos */
begin
```



```

/* aseguramos que el empleado exista en la tabla emp_lab */
if(select count(*) from emp_lab e, inserted i
   where e.num_emp = i.num_emp) = 0
begin
  print "El numero de empleado no esta dado de alta"
  rollback transaction
  return
end
end
else
  if update(fecha_ini)
  begin
    /* aseguramos que la llave primaria sea unica */
    if((select count(*) from emp_psgs e, inserted i
       where e.num_emp = i.num_emp and
             e.fecha_ini = i.fecha_ini) > 1)
    begin
      print "El de empleado tiene ya registrado el permiso con la fecha de
            inicio proporcionada"
      rollback transaction
      return
    end
  end
end
end

```

EMP_SUSPEN_TD

```

create trigger emp_suspen_td
on emp_suspen
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  /* si ningun renglon fue afectado, regresamos*/
  if @num_renglon = 0
    return
  if @num_renglon > 1
  begin
    print "solo se puede dar de baja un empleado a la vez"
    rollback transaction
    return
  end
end

```

EMP_SUSPEN_TIU

```

create trigger emp_suspen_tiu
on emp_suspen
for insert, update
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0

```

```
return
if update(num_emp)
begin
  if((select count(*) from deleted) > 0)
  begin
    print "No estan permitidos los cambios sobre el numero de empleado"
    rollback transaction
    return
  end
else
begin
  /* aseguramos que el empleado exista en la tabla emp_lab */
  if(select count(*) from emp_lab e, inserted i
     where e.num_emp = i.num_emp) = 0
  begin
    print "El numero de empleado no esta dado de alta"
    rollback transaction
    return
  end
end
end
else
if update(causa_suspen)
  if ((select i.causa_suspen from inserted i, emp_suspen e
     where i.num_emp = e.num_emp) != '3' ) and
     ((select i.causa_suspen from inserted i, emp_suspen e
     where i.num_emp = e.num_emp) != '7' ) and
     ((select i.causa_suspen from inserted i, emp_suspen e
     where i.num_emp = e.num_emp) != '8' )
  begin
    print "La causa de suspension solo puede ser 3,7 u 8 "
    rollback transaction
    return
  end
end
```

EMP_WKEND_TIU

```
Create trigger emp_wkend_tiu
on emp_wkend
for insert, update
as
begin
  if @@rowcount = 0
  return
  if (select count(*) from emp_bc,inserted
     where emp_bc.num_emp = inserted.num_emp) = 0
  begin
    print 'Empleado inexistente'
    rollback transaction
  end
end
```

FONDO_AHORRO_TD

```

create trigger fondo_ahorro_td
on fondo_ahorro
for delete
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  /* si el empleado debe al fondo de ahorro no se puede dar de baja */
  if exists (select * from deuda fa df, deleted d
            where df.num_emp = d.num_emp
            having sum(saldo_contra) > 0)
begin
  print "El empleado debe al fondo de ahorro. Para poder dar de baja al
empleado tanto del fondo de ahorro como en forma definitiva debe saldar su
deuda o deudas que este tenga. ¡Verifique la informacion!."
  rollback transaction
  return
end

```

FONDO_AHORRO_TI

```

create trigger fondo_ahorro_ti
on fondo_ahorro
for insert
as
  declare @num_renglon int
  /* Encuentra cuantos renglones fueron modificados */
  select @num_renglon = @@rowcount
  if @num_renglon = 0
    return
  if not exists(select * from emp_lab el, inserted i
               where el.num_emp = i.num_emp)
begin
  print "El empleado que se desa registrar no esta dado de alta"
  rollback transaction
  return
end

```

PJA_INCIDENTE_TIU

```

Create trigger pja_incidente_tiu
on pja_incidente
for insert, update
as
begin
  if @@rowcount = 0
    return
  if (select count(*) from inserted, cat_incidente
     where inserted.cve_incidente = cat_incidente.cve_incidente) = 0
  begin

```

```

        print 'Error : Clave del incidente invalida'
        rollback transaction
    end
    if (select count(*) from inserted, cat_pja
        where inserted.cve_pja = cat_pja.cve_pja) = 0
    begin
        print 'Error : PJA invalido'
        rollback transaction
    end
end

```

PLAZA_TIU

```

create trigger plaza_tiu
on plaza
for insert, update
as
    declare @num_renglon int,
            @jog int,
            @zona_ec int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
if @num_renglon = 0
    return
if update(plaza)
begin
    /* checamos si fue una insercion que esta permitida, si es una
       actualizacion, la rechazamos */
    if((select count(*) from deleted) > 0) /* es una actualizacion */
    begin
        print 'NO esta permitido modificar este campo'
        rollback transaction
        return
    end
end
/* aseguramos que el num_jog exista en el catalogo de JOG */
if update(num_jog)
begin
    if(select count(*) from cat_jog jog, inserted i
        where jog.num_jog = i.num_jog) = 0
    begin
        print 'El JOG proporcionado no existe en el catalogo'
        rollback transaction
        return
    end
end
/* aseguramos que el num_puesto exista en el catalogo de puestos */
if update(num_puesto)
begin
    if(select count(*) from cat_puesto p, inserted i
        where p.num_puesto = i.num_puesto) = 0
    begin
        print 'El puesto proporcionado no existe en el catalogo'
        rollback transaction
        return
    end
end

```

```

end
/* aseguramos que exista un sueldo relacionado al puesto, la categoria y */
/* la zona economica */
if update(num_puesto)
begin
select @zona_ec = ( select cj.zona_ec from inserted i, plaza p, cat_jog cj
                    where p.plaza = i.plaza and cj.num_jog = i.num_jog)
if(select count(*) from sueldo s, inserted i
   where s.num_puesto = i.num_puesto and s.categoria = i.categoria
   and s.zona_ec = @zona_ec ) = 0
begin
print 'El puesto, categoria y zona_ec no estan relacionados a un sueldo'
rollback transaction
return
end
end

end
/* aseguramos que el num_area exista en el catalogo de area_oc o el catalogo de
area_todas_jz */
select @jog = ( select i.num_jog from inserted i, plaza p
               where p.plaza = i.plaza )
/* insert carga.dbo.gam (num_jog)
   select i.num_jog from inserted i, plaza p
   where p.plaza = i.plaza */
if update(num_area)
begin
if (@jog > 32)
begin
/* el jog es mayor que 32, pero checamos que exista en la tabla area_oc */
if(select count(*) from area_oc a, inserted i
   where a.num_jog = i.num_jog and
   a.num_area = i.num_area) = 0
begin
print 'La combinacion JOG-num_area proporcionados no existe en el
catalogo de oficinas centrales. Verifique el numero de JOG y el
numero de area.'
rollback transaction
return
end
end
else
if (@jog <= 32 )
begin
if(select count(*) from area_todas_jz a, inserted i
   where a.num_area = i.num_area) = 0
begin
print 'El area proporcionada no existe en el catalogo area_todas_jz'
rollback transaction
return
end
end
end
else
print ' @jog no tiene ningun valor'
end
end
/* actualizacion de la tabla EMP_CAMBIO */
if (select count(*) from deleted) > 0
begin
insert into emp_cambio
select l.num_emp, getdate(),d.plaza, d.num_puesto, s.salario, d.num_jog,
d.num_area, 1.0 from emp_lab l, plaza p, sueldo s, cat_jog j,
deleted d where l.plaza = d.plaza and

```

```

j.num_jog = d.num_jog and
s.num_puesto = d.num_puesto and
s.categoria = d.categoria and
s.zona_ec = j.zona_ec
if exists(select * from emp_lab el, cont_even ce, deleted d
where el.plaza = d.plaza and ce.num_emp = el.num_emp)
begin
update emp_cambio
set ec.porcentaje_sueldo = ce.porcentaje_sueldo
from emp_lab el, emp_cambio ec, cont_even ce, deleted d
where el.plaza = d.plaza and
ec.num_emp = ce.num_emp and
ce.num_emp = el.num_emp and
ec.fecha_cambio = getdate()
end
end

```

SUELDO_TI

```

create trigger sueldo_ti
on sueldo
for insert
as
declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* sin ningun dato fue modificado o insertado, regresamos */
if @num_renglon = 0
return
/* if @num_renglon > 1
begin
print "solo se puede dar de alta un sueldo a la vez"
rollback transaction
return
end */
/* aseguramos que la zona_ec exista en el catalogo de zona economica */
if(select count(*) from cat_zo ze, inserted i
where ze.zona_ec = i.zona_ec) = 0
begin
print 'La zona economica proporcionada no existe
en el catalogo de zona economica'
rollback transaction
return
end
/* aseguramos que el numero de puesto exista en el catalogo de puestos */
if(select count(*) from cat_puesto pu, inserted i
where pu.num_puesto = i.num_puesto) = 0
begin
print 'El numero de puesto proporcionado no existe
en el catalogo de puesto'
rollback transaction
return
end

```

SUELDO_TU

```

create trigger sueldo_tu
on sueldo
for update
as
  declare @num_renglon int
/* Encuentra cuantos renglones fueron modificados */
select @num_renglon = @@rowcount
/* sin ningun dato fue modificado o insertado, regresamos */
if @num_renglon = 0
  return
if @num_renglon > 1
begin
  print "solo se puede cambiar un sueldo a la vez"
  rollback transaction
  return
end
if update(num_puesto)
begin
  print 'NO estan permitidos los cambios sobre el campo numero de puesto, si
  desea hacerlo realice una baja de la informacion y realice una alta.'
  rollback transaction
  return
end
if update(categoria)
begin
  print 'NO estan permitidos los cambios sobre el campo categoria, si desea
  hacerlo realice una baja de la informacion y realice una alta.'
  rollback transaction
  return
end
if update(zona_ec)
begin
  print 'NO estan permitidos los cambios sobre el campo zona economica, si
  desea hacerlo realice una baja de la informacion y realice una alta.'
  rollback transaction
  return
end
end

```

SYB_TD_RPT

```

create trigger dbo.syb_td_rpt on dbo.syb_reports
for delete
as
  declare @rptid int
  if @@rowcount = 0
    return
  select @rptid = rptid from deleted
  begin tran
    delete dbo.syb_batches
      where rptid = @rptid
    delete dbo.syb_breaks
      where rptid = @rptid
    delete dbo.syb_variables
      where rptid = @rptid
  end tran

```

```
delete dbo.syb_layouts
  where rptid = @rptid
delete dbo.syb_fields
  where rptid = @rptid
delete dbo.syb_texts
  where rptid = @rptid
commit tran
return
```

SYB_TIU_PERM

```
create trigger dbo.syb_tiu_perm on dbo.syb_rptperms
for insert, update
as
declare @ownerid smallint
if @@rowcount = 0
  return
select @ownerid = ownerid from dbo.syb_reports r, inserted i
  where r.rptid = i.rptid
if (user_id() != 1 and user_id() != @ownerid)
begin
  rollback tran
  print "You must be dbo or report owner"
end
if not exists
(select r.rptid from dbo.syb_reports r, inserted i
  where r.rptid = i.rptid),
begin
  rollback tran
  print "This report id is not valid"
end
if not exists(select uid from sysusers, inserted
  where uid = ugid )
begin
  rollback tran
  print "This user/group id is not valid"
end
if not exists (select permcode from inserted
  where permcode = 205 or permcode = 206)
begin
  rollback tran
  print "Permission code is not valid"
end
return
```


APENDICE B - VISTAS

AREAS_JJDOS

Name	Owner	Type			
areas_jzdos	dbo	view			
Data_located_on_segment When_created					
not applicable	Nov 23 1992 8:02PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
nombre_emp	char	63	1	NULL	NULL
fecha_Base	fecha	8	0	NULL	NULL
plaza	char4_num	4	0	NULL	NULL

AREAS_JZUNO

Name	Owner	Type			
areas_jzuno	dbo	view			
Data_located_on_segment When_created					
not applicable	Jan 11 1993 7:56PM				
Column_name	Type	Length	Nulls	Default_name	Rule_name
jog	cadena60	60	0	NULL	NULL
areas_jz	cadena60	60	0	NULL	NULL
num_plaza	char4_num	4	0	NULL	NULL
num_propietario	int	4	1	NULL	NULL
sueldo_plaza	dinero	8	0	NULL	NULL
puesto	cadena60	60	0	NULL	NULL
sueldoemp	dinero	8	0	NULL	NULL

AREAS_OCUNO

Name	Owner	Type
areas_ocuno	dbo	view

Data_located_on_segment When_created
 not applicable Dec 4 1992 7:22PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
jog	cadena60	60	0	NULL	NULL
areas_oc	cadena60	60	0	NULL	NULL
num_plaza	char4_num	4	0	NULL	NULL
num_propietario	int	4	1	NULL	NULL
sueldo_plaza	dinero	8	0	NULL	NULL
puesto	cadena60	60	0	NULL	NULL
sueldoemp	dinero	8	0	NULL	NULL

DEDUCS_OC

Name	Owner	Type
deducs_oc	dbo	view

Data_located_on_segment When_created
 not applicable Jan 7 1994 9:00PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
nm_jog	tentero	1	0	NULL	NULL
nm_emp	tipo_num_emplea	4	0	NULL	NULL
n_anio	anio	2	0	NULL	NULL
n_gna	tentero	1	0	NULL	NULL
n_deduccion	tipo_num_deduc	1	0	NULL	NULL
nombre	cadena60	60	0	NULL	NULL
costo	dinero	8	0	NULL	NULL

EMP_SUELDO

Name	Owner	Type
emp_sueldo	dbo	view

Data_located_on_segment When_created
 not applicable Nov 13 1992 9:00PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
num_emp	tipo_num_emplea	4	0	NULL	NULL
puesto	tipo_num_puesto	2	0	NULL	NULL
categoria	char	1	0	NULL	NULL
tpuesto	char	1	0	NULL	NULL
sueldo	dinero	8	0	NULL	NULL

PERCEPS_OC

Name	Owner	Type
perceps_oc	dbo	view

Data_located_on_segment When_created

not applicable Jan 7 1994 8:57PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
nu_jcg	tentero	1	0	NULL	NULL
nu_emp	tipo_num_emplea	4	0	NULL	NULL
n_anio	anio	2	0	NULL	NULL
n_gna	tentero	1	0	NULL	NULL
n_percepcion	tipo_num_percep	1	0	NULL	NULL
nombre	cadena60	60	0	NULL	NULL
costo	dinero	8	0	NULL	NULL

PLAZA_SUELDO

Name	Owner	Type
plaza_sueldo	dbx	view

Data_located_on_segment When_created

not applicable Nov 13 1992 8:54PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
plaza	char4_num	4	0	NULL	NULL
salario	dinero	8	0	NULL	NULL

SPEROC

Name	Owner	Type
speroc	dbo	view

Data_located_on_segment When_created

not applicable Jan 5 1994 8:04PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
n_emp	tipo_num_emplea	4	0	NULL	NULL
pcosto	dinero	8	0	NULL	NULL
spcosto	moneyn	8	1	NULL	NULL

SUM_APORTAFOV

Name	Owner	Type
sum_aportafov	dbo	view

Data_located_on_segment	When_created
not applicable	Aug 17 1993 5:47PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
sumaquin1	moneyn	8	1	NULL	NULL
sumaquin2	moneyn	8	1	NULL	NULL
sumaquin3	moneyn	8	1	NULL	NULL
sumaquin4	moneyn	8	1	NULL	NULL
sumajogs	moneyn	8	1	NULL	NULL

SUM_APORTAS

Name	Owner	Type
sum_aportas	dbo	view

Data_located_on_segment	When_created
not applicable	Jul 5 1993 6:18PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
sumquin1	moneyn	8	1	NULL	NULL
sumquin2	moneyn	8	1	NULL	NULL
sumquin3	moneyn	8	1	NULL	NULL
sumquin4	moneyn	8	1	NULL	NULL
sumjogs	moneyn	8	1	NULL	NULL

SUM_AREASJE

Name	Owner	Type
sum_areasje	dbo	view

Data_located_on_segment	When_created
not applicable	Jan 26 1993 7:31PM

Column_name	Type	Length	Nulls	Default_name	Rule_name
jog	cadena60	60	0	NULL	NULL
area	cadena60	60	0	NULL	NULL
sdoplaareas	moneyn	8	1	NULL	NULL
sdoplajogs	moneyn	8	1	NULL	NULL
plaareas	int	4	1	NULL	NULL
plajogs	intn	4	1	NULL	NULL
nomareas	int	4	1	NULL	NULL
nomjogs	intn	4	1	NULL	NULL
sdoempareas	moneyn	8	1	NULL	NULL
sdoempjogs	moneyn	8	1	NULL	NULL

SUM_AREASOC

Name	Owner	Type
sum_areasoc	dbo	view

Data_located_on_segment	When_created
not applicable	Feb 13 1993 11:58AM

Column_name	Type	Length	Nulls	Default_name	Rule_name
jog	cadena60	60	0	NULL	NULL
area	cadena60	60	0	NULL	NULL
sdoplaareas	moneyn	8	1	NULL	NULL
sdoplaajogs	moneyn	8	1	NULL	NULL
plaareas	int	4	1	NULL	NULL
plajogs	intn	4	1	NULL	NULL
nomareas	int	4	1	NULL	NULL
nomjogs	intn	4	1	NULL	NULL
sdoempareas	moneyn	8	1	NULL	NULL
sdoempjogs	moneyn	8	1	NULL	NULL

APENDICE C - REGLAS

ANIO

```
create rule anio
as
  @anio like "[6-9][0-9]"
```

CHAR1_ALFA

```
create rule char1_alfa
as
  @char1 like "[A-Z,a-z]"
```

CHAR1_NUM

```
create rule char1_num
as
  @char1 like "[0-9]"
```

CHAR2_ALFA

```
create rule char2_alfa
as
  @char2 like "[A-Z,a-z][A-Z,a-z]"
```

CHAR2_NUM

```
create rule char2_num
as
  @char2 like "[0-9][0-9]"
```

CHAR3_ALFA

```
create rule char3_alfa
as
  @char3 like "[A-Z,a-z][A-Z,a-z][A-Z,a-z]"
```

CHAR3_NUM

```
create rule char3_num
as
  @char3 like "[0-9][0-9][0-9]"
```

CHAR4_NUM

```
create rule char4_num
as
  @char4 like "[0-9][0-9][0-9][0-9]"
```

CHAR5_NUM

```
create rule char5_num
as
  @char5 like "[0-9][0-9][0-9][0-9][0-9]"
```

EDIFICIO

```
create rule edificio
as
  @edif like "[A-Z]" or
  @edif like "[a-z]" or
  @edif like " "
```

EP_PROG

```
create rule ep_prog
as @prog in ( "5F", "5K", "AA", "5f", "5k", "aa" )
```

NUM_EDO

```
/****          REGLAS PARA LOS TIPOS          *****/
create rule num_edo
as
  @edo like "[0][1-9]" or
  @edo like "[1,2][0-9]" or
  @edo like "[3][0-2]"
```

REGLA AUTORIDAD

```
/* regla aplicada al campo aut_concede
   donde : C = Sindicato,      I = ISSSTE.
           J = Jefe Inmediato, S = Jefe Superior. */
create rule regla_autoridad
as
  @autoridad in ('C', 'I', 'J', 'S')
```

REGLA_BIM

```
create rule regla_bim
as
  @bim between 1 and 6
```

REGLA_CATEGORIA

```
/* regla aplicada al campo categoría */
create rule regla_categoria
as
  @categoria in ('A','B','C','D','E','F','G','H')
```

REGLA_CONSECUTIVO

```
/* regla aplicada al campo consecutivo */
create rule regla_consecutivo
as
  @consecutivo between 1 and 2
```

REGLA_CP

```
/* regla aplicada al código postal */
create rule regla_cp
as
  @cp like "[0-9][0-9][0-9][0-9][0-9]"
```

REGLA_DEPENDIENTES

```
/* regla para los dependientes económicas
donde: M = Madre, P = Padre, H = Hijo, E = Esposa */
create rule regla_dependientes
as
  @dependiente in ('M','P','H','E')
```

REGLA_DIA

```
/* regla aplicada al campo día */
create rule regla_día
as
  @día between 1 and 31
```

REGLA_EDOCIVIL

```
/* regla aplicada al campo estado civil */
create rule regla_edocivil
as
  @edo like "[0-7]"
```


REGLA_FACTOR

```
/* regla para los factores del catalogo de deducciones y
percepciones donde : C = Constante, V = Variable, R = Rutina.
*/
create rule regla_factor
as
    @factor in ('C','V','R')
```

REGLA_FACTOR_1

```
create rule regla_factor_1
as
    @factor in ('C','V','R','N')
```

REGLA_FUNCION

```
/* regla aplicada al campo función
donde : T = Técnica, A = Administrativa. */
create rule regla_función
as
    @función in ('T','A')
```

REGLA_HORA

```
create rule regla_hora
as
    @tiempo like "[0-1][0-9][:][0-5][0-9]" or
    @tiempo like "[2][0-4][:][0-5][0-9]"
```

REGLA_MAX_PER

```
create rule regla_max_per
as
    @max between 0 and 99
```

REGLA_MAYORACERO

```
/* regla para los campos cuyos valores sean mayores a cero */
create rule regla_mayoracero
as
    @campo > 0
```

REGLA_MAYORIGUALCERO

```
/* regla para los campos cuyos valores
 sean mayores-iguales a cero */
create rule regla_mayorigualcero
as
    @campo >= 0
```

REGLA_MES

```
/* regla aplicada al campo mes */
create rule regla_mes
as
    @mes between 1 and 12
```

REGLA_NIVAPLI

```
/* regla aplicada al campo nivel de aplicación*/
create rule regla_nivapli
as
    @nivel between 1 and 9
```

REGLA_NIVBECA

```
/* regla aplicada al campo nivel beca
 donde : S = Superior, M = Medio Superior, T = Medio Terminal. */
create rule regla_nivbeca
as
    @nivel in ('M', 'S', 'T')
```

REGLA_NIVELPUESTO

```
/* regla aplicada al campo nivel puesto
 donde: P = Profesional, T = Técnico */
create rule regla_nivelpuesto
as
    @nivel in ('P', 'T')
```

REGLA_NUMCOND

```
/* regla aplicada al campo num_cond */
create rule regla_numcond
as
    @cond between 1 and 15
```

REGLA_NUMCUENTA

```
create rule regla_numcuenta
as
  @cuenta like "[0-9][0-9][0-9][0-9][0-9][0-9]
[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]"
```

REGLA_NUMDEDUCCION

```
/* regla aplicada al campo numero de deducción */
create rule regla_numdeduccion
as
  @num between 1 and 49
```

REGLA_NUMJOG

```
/* regla aplicada al campo num_jog */
create rule regla_numjog
as
  @jog between 1 and 49
```

REGLA_NUMBER

```
/* regla aplicada al campo num_percepción */
create rule regla_number
as
  @num between 1 and 49
```

REGLA_ORIGENED

```
/* regla aplicada al campo origen del catalogo de deducciones
donde :C = Por Catalogo, O = Por Orden.
      P = Por Procedimiento, I = Por Orden Iterativo.
      K = Por Orden iterativa Kancelable. */
create rule regla_origened
as
  @origen in ('C','O','P','I','K')
```

REGLA_ORIGENPER

```
/* regla aplicada al campo origen del catalogo de percepciones.
donde :C = Por Catalogo, O = Por Orden.
      P = Por Procedimiento. */
create rule regla_origenper
as
  @origen in ('C','O','P')
```

REGLA_PERIODO

```
create rule regla_periodo
as
  @periodo in ('Q','M')
```

REGLA_PERIODOMAX

```
/* regla aplicada al campo periodo_max_per
   donde: A = Anual, D = Diario, S = Semestral, U = Unico.*/
create rule regla_periodomax
as
  @maximo in ('A','D','S','U')
```

REGLA_PLAN_FAH

```
/* regla aplicada al campo plan_fah */
create rule regla_plan_fah
as
  @pfh in ('A','B')
```

REGLA_POD

```
/* regla aplicada al campo pod
   donde : P = Percepción, D = Deducción. */
create rule regla_pod
as
  @pod in ('P','D')
```

REGLA_PRIOAPLI

```
/* regla aplicada al campo prioridad de aplicación */
create rule regla_prioapli
as
  @prioridad between 1 and 14
```

REGLA_QUINCENA

```
/* regla aplicada al campo quincena */
create rule regla_quincena
as
  @quincena between 1 and 24
```

REGLA_RFCL

```
create rule regla_rfcl
as
  @letra like "[A-Z][A-Z][A-Z][A-Z]"
```

REGLA_RFCN

```

/* regla aplicada al campo rfc numero */
create rule regla_rfcn
as
  @enum like "[1-9][0-9][0][1-9][0][1-9]" or
  @enum like "[1-9][0-9][0][1-9][1-2][0-9]" or
  @enum like "[1-9][0-9][0][1-9][3][0-1]" or
  @enum like "[1-9][0-9][1][0-2][0][1-9]" or
  @enum like "[1-9][0-9][1][0-2][1-2][0-9]" or
  @enum like "[1-9][0-9][1][0-2][3][0-1]"

```

REGLA_SEXO

```

create rule regla_sexo
as
  @sexo in ('M', 'F')

```

REGLA_TIPO_CANTIDAD

```

create rule regla_tipo_cantidad
as
  @tipo in ('H', 'M', 'V', 'D')

```

REGLA_TIPOAREA

```

/* regla aplicada al campo tipo_area
   donde :S = Subdireccion, C = Coordinacion, D = Duración. */
create rule regla_tipoarea
as
  @area in ('S', 'C', 'D')

```

REGLA_TIPOPUUESTO

```

/* regla aplicada al campo tipo_puesto
   donde: B = Base, C = Confianza, M = Mandos Medios,
          S = Mandos Superior, H = Honorarios */
create rule regla_tipopuesto
as
  @tpuesto in ('B', 'C', 'M', 'S', 'H')

```

REGLA_ZONAEC

```

/* regla aplicada al campo zona_económica */
create rule regla_zonaec
as
  @zona between 1 and 4

```

TIPO_ART

```

create rule tipo_art
as @art in ("M", "P", "A", "T", "m", "p", "a", "t")

```

DEFAULTS

BLANCOI

```
create default blancoi
as , ,
```

CERO

```
create default
as 0
```

DEF_ANIO

```
create default def_anio
as '92'
```

NO_PROPOR

```
create default no_propor
as 'NO PROPORCIONADO'
```

NO_TIENE

```
create default no_tiene
as 'NO_TIENE'
```

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".
- APT (Application Productivity Tools): Lenguaje de cuarta generación que nos sirve para el desarrollo de aplicaciones, propio del DBMS SYBASE.
- 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.
- C: Es un lenguaje de programación de tercera generación, utilizado ampliamente como software de workstations, minicomputadoras y microcomputadoras.
- CLIENTE: Cualquier elemento dentro de un ambiente de cómputo que solicita un servicio disponible dentro de cualquier parte del ambiente para realizar las tareas del usuario.
- CLIENTE-SERVIDOR: Modelo lógico utilizado para crear un procesamiento cooperativo que constituye el fundamento de las aplicaciones de software de un gran número de distribuidores y empresas. Fue originalmente creada durante las actividades de estandarización de la ISO/OSI en la década de los 70's, este concepto es independiente de la arquitectura del hardware y de los sistemas operativos y el término es utilizado para referirse a la filosofía de diseño más que para describir al software o a los componentes del hardware, prácticamente existen dos variaciones del término:

* Tanto el cliente como el servidor son parte del mismo software que pueden o no residir en la misma computadora, por ejemplo, programas de aplicación y la base de datos.

* El término cliente esta 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 otro. Este es un pre-requisito para procesamiento distribuido.

- **DBMS (DATA BASE MANAGMENT SYSTEM):** Sistema manejador de base de datos, es una herramienta utilizada para la creación e implementación de sistemas de bases de datos.

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

- **DICCIONARIO DE DATOS:** Depósito central de información de los sistemas de software o aplicaciones de procesamiento de datos, la cual puede ser vista como un modelo de datos, el diccionario incluye las siguientes clases de información:

* Qué datos son almacenados y dónde..

* Cuáles objetos de procesamiento trabajan con los datos.

* Quiénes crean los datos y los objetos que procesan los datos.

* 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.

- **DIAGRAMA ENTIDAD RELACION (DER):** Permite diseñar un fenómeno real para un sistema de base de datos, a través de diagramas con entidades y sus relaciones.

- **DIAGRAMA DE FLUJO DE DATOS (DFD):** Es el modelo del sistema, la descripción completa del mismo estará formada por un conjunto de

DFD.

- DESNORMALIZACION: Proceso de regreso intencional de una normalización para proveer eficiencia y puede ocurrir a nivel columna o a nivel tabla.
 - DWB (DATA WORK BENCH): Producto del DBMS SYBASE utilizado para la realización de consultas, manejo de tablas, manejo del diccionario de datos, que además incluye el reporteador.
 - FRONT-END: Es una parte del equipo y/o software que actúa como una interfase 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 sintáxis, 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 utilizado en la conexión de software y hardware, para indicar que ellos tienen diferentes arquitecturas.
 - HOMOGENEO: Es un término utilizado en sistemas múltiples de software y hardware para indicar que comparte una arquitectura común.
- 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, consiste en una serie de reglas, que gobiernan las actualizaciones. La integridad de datos asegura que los datos al principio y al final de cada transacción individual sean consistentes.
 - INTERACTIVO: Se refiere a un diálogo continuo entre el usuario y la aplicación. Dos programas también pueden interactuar entre sí, sí cooperan en un ambiente cliente-servidor.
 - MAESTRO/ESCLAVO: Es un tipo de procesamiento cooperativo en el cual un programa (maestro) llama a otro programa subordinado (esclavo) para satisfacer un requerimiento.
 - 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.

- MINICOMPUTADORA: Es un término utilizado para describir a las computadoras que se encuentran entre un mainframe y una workstation.

- NORMALIZACION: La normalización es un método que logra que una base de datos se encuentre en una consistencia lógica con un mínimo de redundancia.

- OPEN-CLIENT: Utiliza librerías para que un proceso programado en un lenguaje de tercera generación pueda acceder al servidor de la base de datos, este producto es propio de SYBASE.

- OPEN SQL-SERVER: Es un producto que proporciona el manejo de SQL estándar para acceder a una gran variedad de manejadores heterogéneos de SQL (IBM, Oracle, SYBASE, Ingres, RDB, etc). De esta forma permite que cualquiera de estos manejadores actúe como servidor.

- PC (o Personal Computer): Es la más pequeña de las clases de hardware, y es comúnmente utilizada en ambientes administrativos.

- PLATAFORMA: Es un término genérico utilizado para describir el hardware del sistema operativo en la cual una pieza dada del 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.

- 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.

- **RIGHTSIZING:** El proceso de ajustar el tamaño de 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.
- **SERVIDOR:** 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 interfase programada para que los sistemas 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 las bases de datos.
- **SQL-DB SERVER:** Es un manejador puramente relacional, que es particularmente flexible para trabajar en plataformas de minicomputadoras.
- **STORE PROCEDURE:** SYBASE es un manejador de bases de datos de alto rendimiento, para aplicaciones OLTP (on line transactions procesing). 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 ejecución), en tiempo de ejecución.

Primero las transacciones típicas de un programa OLTP utilizan las sentencias INSERT, DELETE, UPDATE y SELECT, que acceden a una única fila de datos basándose en la clave primaria de la fila. Es relativamente fácil optimizar este tipo de sentencias 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.

Segundo y más importante, los procedimientos almacenados proporcionan un modo de analizar, optimizar y compilar una secuencia entera de programas, 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 pide a SQL server que ejecute el procedimiento almacenado. Por tanto los procedimientos almacenados proporcionan al SQL server la flexibilidad de un interfase SQL dinámico y los beneficios de rendimiento de las sentencias precompiladas encontradas en SQL estático.

Los procedimientos almacenados también juegan un papel importante en la mejora del rendimiento de RED, de la arquitectura cliente-servidor, propuesta inicialmente por SYBASE y SQL SERVER. En esta arquitectura, el programa de aplicación reside en un sistema cliente, mientras el DBMS reside en un sistema servidor diferente, conectado a los sistemas clientes mediante una red de área local, para obtener buenos rendimientos de la arquitectura cliente-servidor, el tráfico sobre la red debe ser minimizado y los procedimientos almacenados ayudan a lograr ese objetivo.

- TRANSACCION: Una unidad lógica de trabajo, que modifica una o más bases de datos, sin perder la consistencia de los mismos.

- TWO PHASE COMMIT: La parte de manejo de datos que garantiza la integridad de la información en un ambiente multibases de datos. El two phase commit es un método que coordina las actividades de transacción, asegurando que la actualización de cada base de datos sea completada o desechada.

-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.