UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

"SISTEMA PARA EL CONTROL DE LA INFORMACIÓN DE ESTACIONES Y EQUIPOS DE LA RED ACELEROGRÁFICA DEL INSTITUTO DE INGENIERÍA"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A N:

DIANA CHÁVEZ ESPINOZA DE LOS MONTEROS
MARÍA DE LOS ÁNGELES HERRERA CORRO

DIRECTOR DE TESIS:
ING. CITLALI PÉREZ YÁÑEZ.

MÉXICO, D. F. 1999.

"SISTEMA PARA EL CONTROL

DE LA INFORMACIÓN

DE ESTACIONES Y EQUIPOS

DE LA RED ACELEROGRÁFICA

DEL INSTITUTO DE INGENIERÍA"

Damos nuestro agradecimiento a:

La Universidad Nacional Autónoma de México
Nuestra Alma Mater.

A la Facultad de ingeniería y a los profesores
Que contribuyeron en nuestra formación profesional
para servir a la comunidad.

Al Instituto de Ingeniería
Por permitirnos formar parte de su personal.
Por la oportunidad de realizar este trabajo.

A todo el personal de la coordinación de Instrumentación sísmica:
En especial al Ing. David Almora Mata por su apoyo y cooperación.

Y de una manera muy especial
a nuestra directora Ing. Citlali Pérez Yáñez
Por su dedicación, ayuda y paciencia en la
elaboración de este trabajo.

"El presente proyecto es el resultado del esfuerzo y la dedicación no sólo de dos personas, sino de todos aquellos que nos brindaron su apoyo y su cariño.... en lo particular, y de una manera muy especial quiero expresar mi agradecimiento a

Rosa y Jorge
Por darme la vida y guiarme
por el camino correcto.

Alejandra y Jorge
Por compartir sus experiencias.

Gloria Chávez
Por quererme y darme consejos.

José Guadalupe Chávez
Por ese cariño que incondicional
hasta la muerte.

Socorro Vargas
Por su ternura.

Sara y Angeles
Porque a través de ellas
conocí la verdadera amistad.

Atte. Diana Chávez Espinoza de los M.

Con este trabajo quiero dar un sincero agradecimiento:

A Dios:
Por no abandonarme nunca
e iluminar el camino de mi vida.

A mi madre:
Por el amor, la confianza, los sacrificios
y el apoyo para superarme. Porque si ti
jamas lo hubiera logrado,
éste trabajo te lo dedico Gracias.

A mi abuelita:
Por su amor, dedicación
y cuidados hacia mi.

A mi hermano:
Por su cariño.

A Diana Chávez y Rocío Barajas
Porque me han enseñado
lo que vale la amistad.

A mis amigos y amigas:
Quienes han estado conmigo
dándome su apoyo, en especial a
Eloisa, Gloria, José Luis, Arturo,
Mauricio A., Jorge G. L. y Ricardo.

Atte. Ángeles Herrera C.

ÍNDICE

Capítulo 1 Introducción 2

Capítulo 2 Planteamiento y análisis del sistema 6

2.1 Manejo actual de la información 6

2.2 Presentación del problema 7

2.3 Análisis y estrategia de solución 8

2.3.1 Arquitectura cliente/servidor 10

2.3.2 Planteamiento de la solución 12

2.3.2.1 Software 13

2.3.2.2 Hardware 17

2.3.2.3 Usuario final 17

2.4 Análisis del sistema 19

2.4.1 Diagrama de flujo de datos 22

2.4.2 Diagrama entidad/relación 34

2.4.3 Normalización de la base de datos 37

2.4.4 Diccionario de datos 41

Capítulo 3 Diseño del sistema 52

3.1 Tablas y valores nulos 52

3.2 Reglas, tipo de datos, defaults. 53

3.3 Procedimientos almacenados 54

3.4 Triggers 58

3.5 Vistas 61

3.6 Tamaño de la base de datos 73

Capítulo 4 Interfaz gráfica del usuario 76
4.1 Diagrama de transición de estados 80
87
Capitulo 5 Operación del sistema
5.1 Inventario 89
5.2 Actualizaciones 92
5.3 Consultas 95
5.4 Reportes 102
5.5 Ayuda 104
5.6 Salir 105

Capítulo 6. Conclusiones 107

Bibliografía 111

Apéndice Manual del usuario 114
A-1 Instalación del sistema 114
A-2 Trabajando con el sistema 116

Anexo Programa fuente (SQL) 128

Capítulo 1

Introducción

CAPÍTULO 1

1. INTRODUCCIÓN

A más de 35 años de iniciada la instrumentación sísmica en México, el Instituto de Ingeniería ha sido la institución pionera en el registro de eventos sísmico. La cual cuenta con una extensa red acelerográfica localizada principalmente en las costas de Guerrero, los estados de Michoacán, Puebla, Oaxaca y el Distrito Federal, lugares de mayor sismicidad del país, contribuyendo de esta manera al desarrollo de la ingeniería sísmica encargada de establecer una normatividad para la construcción de estructuras y el uso de suelo.

En la actualidad se cuenta con diversas tecnologías de registro de datos (cassette, memoria RAM, memoria flash) y aproximadamente 100 estaciones con equipos acelerográficos. Con el desarrollo tecnológico y la modernización de los instrumentos, se han generado enormes cantidades de información, misma que debería ser manejada con gran control de una forma flexible y automatizada.

Recientemente se ha hecho un esfuerzo por tener concentrados los registros acelerográficos de las Instituciones (Cenapred, ICA, I. de I., CICESE, CFE) en una base de datos a nivel nacional, reforzando la idea de tener relacionados los datos de estaciones e instrumentos para una fácil consulta y acceso a los mismos, además de permitir llevar un riguroso control de dicha información.

Debido a los avances en las últimas décadas en el área de computación, los sistemas han sido de gran utilidad para facilitar el desarrollo y desempeño de muchas empresas. Estos se encuentran presentes en una gran parte de las

actividades cotidianas, por ello es conveniente la creación de un sistema para el control de la información de instrumentos y estaciones de la red acelerográfica del Instituto de Ingeniería, mismo que tendrá como objetivo principal integrar en una base de datos las rutinas de instalación, mantenimiento y procesamiento de la información generada por dicha red, de tal manera que optimice y facilite la captura, almacenamiento, consulta y generación de reportes, así mismo que permita tener actualizado el historial de parámetros y características de cada uno de los equipos y estaciones acelerográficas.

En el capítulo 2 de este trabajo, se plantea la forma actual en que es controlada la información, se hace la presentación del problema, se analiza la estrategia de solución basándose en la arquitectura cliente/servidor, así como los modelos de bases de datos, se plantea la solución con la evaluación de los requerimientos del usuario final, de software y hardware y dentro del análisis del sistema se presentan los diagramas de flujo de datos, normalización de la base de datos, diagrama entidad relación y diccionario de datos.

El capítulo 3 muestra las definiciones y diseños de las tablas, las reglas, los defaults, los procedimientos almacenados, los triggers y las vistas, también se señala el tamaño de la base de datos y la proporción en la que crecerá.

El capítulo 4 explica el ambiente en que se desarrolló la interfaz, los diagramas de transición de estados y las consideraciones que se tomaron en cuenta para el diseño de las diferentes pantallas que conforman los módulos.

El capítulo 5 presenta la operación del sistema a los usuarios con base en los módulos de inventario, actualización, consultas, reportes y ayuda, mismos que se describen de forma amplia.

El capítulo 6 presenta los comentarios finales a los que se llega desde un inicio hasta la puesta en operación el sistema, así como la seguridad.

Finalmente, se hace mención a la bibliografía que nos sirvió de referencia para la realización de la presente Tesis Profesional.

En el apéndice se ha colocado el manual del usuario y en el anexo se encuentra el programa fuente del sistema.

Capítulo 2

Planteamiento y Análisis del Sistema.

CAPÍTULO 2

2. PLANTEAMIENTO Y ANÁLISIS DEL SISTEMA.

2.1. MANEJO ACTUAL DE LA INFORMACIÓN.

En la coordinación de instrumentación sísmica se maneja información sobre estaciones, equipos acelerográficos y sensores.
Esta información generada desde hace algunos años, se tiene organizada de la siguiente manera:

- El historial de los equipos acelerográficos se encuentra en hojas de papel y libretas.
- Se tienen las hojas de especificación de los equipos con datos originales acerca de la fabricación de los equipos y sensores, así como de los cambios que han ido sufriendo en algunos de sus parámetros.
- Carpetas donde se lleva el control de lectura de cassettes y archivos, que son los lugares en que se almacenan temporalmente los registros acelerográficos.
- Libretas organizadas por equipo y número de serie donde se lleva el historial de los cambios realizados.
- Se cuenta también con bitácoras de revisión del estado de operación de estaciones y equipos (algunos de los datos son: fecha de revisión, personal encargado de la revisión, tiempos de referencia, estado de la alimentación de la estación, baterías, celdas solares, reguladores y en su caso línea AC. y cargadores).

En base a lo anterior se puede ver que la información no se encuentra organizada en un solo lugar, por lo tanto la consulta de algunos datos de equipos o de estaciones resulta deficiente.

Esto ocasiona que el acceso a dicha información se realice de una forma poco práctica, incluso para el personal que está a cargo del correcto funcionamiento de la red y que desea obtener reportes que le faciliten y ayuden en el mantenimiento correctivo y preventivo de alguna estación, equipo y/o sensor.

2.2. PRESENTACIÓN DEL PROBLEMA.

Observamos que algunos de los problemas que se manifiestan al manejar manualmente la información de la red acelerográfica, repercuten especialmente en el procesamiento de los datos, ya que la información final que se entrega al usuario debe ser precisa en todos sus valores, si esto no ocurre, podrían cambiar drásticamente los valores de la información proporcionada.

Debido a la diversidad de modelos de equipos acelerográficos y sensores instalados en las estaciones, el problema se complica ya que puede existir un cambio continuo a otras estaciones, ya sea por reparación, pruebas o estrategias especiales y dado que las actividades se llevan a cabo en forma manual, la información que se obtiene en la consulta no siempre es exacta, ni se obtiene con la prontitud deseada a causa de que existe un gran manejo de documentos.

Por lo que se requiere tener una constante actualización de la información de los instrumentos y estaciones acelerográficas, que su acceso y consulta sea de manera eficaz, que se relacionen los parámetros y constantes de cada equipo y estaciones, como rangos, números de serie, orientaciones, tipos de instrumentos, coordenadas geográficas (latitud y longitud), cambios, etc. esto es, se requiere de una infraestructura que permita reorganizar, actualizar y conservar la información disponible para facilitar la consulta y acceso al usuario, llevando un riguroso control de dicha información.

2.3. ANÁLISIS Y ESTRATEGIA DE SOLUCIÓN.

Dado lo anterior, es conveniente el desarrollo de un sistema que permita tener un control organizado y centralizado de los datos, pero además que cada grupo de trabajo pueda manejar sus propios datos desde su lugar de trabajo; una aplicación que opere independientemente en diferentes máquinas, compartiendo la misma información, incluso, con posibilidad a futuro de que otras instituciones tengan acceso a ella.

Las necesidades que se mencionaron anteriormente han llevado a realizar el planteamiento de una base de datos (BD), misma que se define como la unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier redundancia entre los mismos, es decir, que los datos no se repitan y es individual si puede compartirse entre usuarios distintos, en el sentido de que cada uno de ellos pueda tener acceso a la misma parte de la base de datos y utilizarla con propósitos diferentes.

Existen tres modelos de bases de datos, el modelo jerárquico, el de red y el relacional, en los que puede basarse un *sistema de manejo de base de datos* (DBMS) para definir la estructura fundamental de los mismos.

Estos modelos ofrecen ventajas y desventajas, por ejemplo, el *modelo Jerárquico* se construye y organiza en una estructura de árbol, donde cada nivel es llamado nodo y el nivel más alto es el nodo raíz; cada relación es una rama y cada nodo terminal recibe el nombre de hoja, la ventaja mayor que ofrece este modelo es la existencia de sistemas de manejo de base de datos probados que usan el modelo como estructura básica, su diseño es relativamente sencillo, además, permite crear nuevos campos en cada nivel de la estructura, con sólo cambiar un apuntador, pero su principal desventaja ocurre en su tipo de relaciones (muchos a muchos) que pueden ocasionar redundancia de los datos almacenados, y dado el estricto ordenamiento jerárquico, la inserción y supresión de registros se vuelve extremadamente compleja.

El *modelo de Red* consiste en una colección de registros conectados entre sí por ligas, donde el *registro* es un conjunto de campos (atributos) y una *liga* es una asociación entre dos registros, por lo que se considera que este modelo es fácil de implementar.
Su principal desventaja radica en el hecho de que el diseño inicial de la base de datos es fundamental, porque cuando ya ha sido creado, cualquier cambio en algún grupo de datos implica realizar una nueva estructura.

En cambio, en el *modelo Relacional* los datos se perciben por los usuarios en forma de relaciones o tablas bidimensionales (renglones y columnas) y los operadores disponibles para interactuar con la información, manipulan estas tablas sin crear dependencia entre ellas.

Este modelo ofrece las siguientes ventajas:

- Reducir la redundancia de datos que ocasionaría un desperdicio de espacio de almacenamiento.
- Evitar la inconsistencia en los datos que daría a los usuarios información incorrecta o contradictoria.
- Mantener la integridad del sistema para asegurar que los datos en la base sean exactos.
- Compartir los datos entre las diferentes aplicaciones.
- Tener facilidad en el desarrollo de aplicaciones.
- Aplicar reglas de seguridad para cada tipo de acceso a la información.
- La independencia de datos que ofrece este modelo constituye el objetivo principal de cualquier sistema ya que elimina la consideración de restricciones impuestas por dispositivos físicos.

Además, cubre las características y necesidades con que debe contar la base de datos:

- Representación de datos por medio de tablas.
- Flexibilidad en el mantenimiento de las estructuras y datos en las consultas.
- Tener integrado su diccionario de datos.
- Simplicidad de uso y entendimiento.
- Permitir obtener respuestas a partir de consultas no planeadas.
- Inserción y fácil actualización de datos.
- Independencia de datos, es decir, la capacidad para utilizar la base de datos sin conocer los detalles de representación y localización de los mismos.
- Las peticiones de usuarios para accesar los datos son manejadas por el DBMS (Sistema Manejador de Bases de Datos).

2.3.1. ARQUITECTURA CLIENTE/SERVIDOR.

Se ha mencionado la conveniencia de tener la información organizada y centralizada y que cada usuario pueda manejar los datos desde su lugar de trabajo de manera independiente; la forma en que se pretende cumplir con estas características es con una arquitectura cliente/servidor(C/S). Esta arquitectura nos permitirá al tener la base de datos en el servidor y la aplicación corriendo en el cliente, un mejor rendimiento y tiempo de respuesta.

Dentro de la arquitectura cliente/servidor intervienen las siguientes partes: el cliente, el servidor y el elemento que los enlaza a la red de cómputo. Tanto el cliente como el servidor son combinaciones de software y hardware. Se puede observar como una plataforma que maneja un ambiente de paso de mensajes: el cliente que inicia la comunicación, direcciona ciertas tareas y da los servicios a un nivel local, y el servidor, estando a la espera de requerimientos, manipulación de datos y seguridad e integridad de los mismos, responde (Fig. 2.1.).

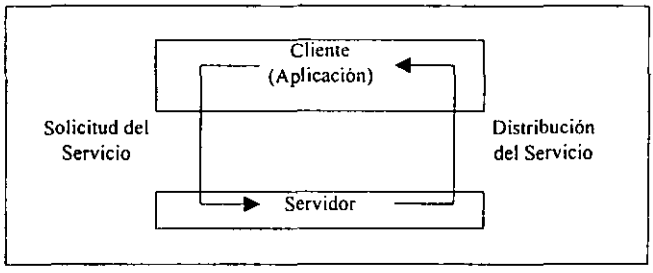


Fig.2.1. Paso de mensajes en modelo cliente/servidor.

También existen elementos que se distribuyen entre el cliente y el servidor, los cuales son: el manejo de datos, la aplicación y la presentación.

- El manejo y almacenamiento de datos se refiere al sistema de archivos o al manejador de bases de datos (DBM),
- la aplicación es el software que emplea tales datos para el propósito específico del usuario, y
- la presentación es el software que establece la forma en que los datos se visualizan en la pantalla del cliente.

Las ventajas más importantes de esta arquitectura son:

- Permite el acceso de varios clientes a un servidor en forma simultánea (Fig. 2.2).
- Reduce el tráfico en la red, ya que por ella sólo viajan los requerimientos y las atenciones a ellos.
- Puede operar bajo sistemas abiertos, lo que significa capacidad de cambio de plataformas con un mínimo de problemas y riesgos.
- Permite interactuar con interfaces gráficas de usuario versátiles a los clientes.

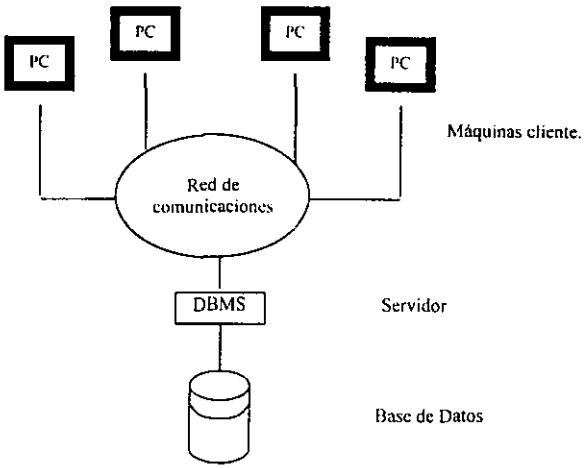


Fig. 2.2. Diagrama de una arquitectura cliente/servidor.

2.3.2. PLANTEAMIENTO DE LA SOLUCIÓN.

Hasta este punto se han analizado las ventajas que representaría integrar nuestro sistema en una base de datos relacional y con la filosofía de que la aplicación pueda correr independientemente en diferentes máquinas como en un sistema multiusuario.
Sin embargo, para evaluar un sistema de base de datos es importante tener en cuenta las siguientes consideraciones: la interoperabilidad, la adaptabilidad y escalabilidad.

- La *interoperabilidad* permite elegir los mejores componentes, tanto en hardware como en software, y ofrece la certeza de que ambos funcionarán bien.
- La *adaptabilidad* garantiza que el sistema aceptará avances tecnológicos.
- La *escalabilidad* asegura que el sistema proveerá siempre la satisfacción de las necesidades comerciales. Es decir, que si la empresa crece el sistema también crece.

Así mismo, como requerimientos para la selección de herramientas que llevan a la solución del sistema, se consideraron al software, hardware y usuario final como factores básicos, mismos que se describen a continuación:

2.3.2.1. SOFTWARE.

El *Software* son los programas en la computadora que interactúan y realizan los procesos requeridos por el usuario. Dicho software puede ser de aplicación como son los programas que se utilizan con fines específicos (p.e. procesadores de texto, hojas de cálculo, programas de diseño, y de contabilidad entre otros) y el software de sistema operativo (p.e. MS-DOS, UNIX, OS/2 etcétera) el cual controla y administra los procesos del equipo de cómputo.

Los elementos con los que disponemos en la Coordinación de Instrumentación Sísmica son básicamente Sistema operativo y software de bases de datos como herramientas de trabajo, y el uso de computadoras PC conectadas a través de una red de comunicación bajo el ambiente de Microsoft Windows.

Además, el Instituto de Ingeniería por medio de la coordinación de servicios de cómputo tiene a su cargo la administración del servidor "Tonatiuh" en el que está instalado el Manejador de Bases de Datos SYBASE versión 11; esto aunado a la evaluación de diversos productos de bases de datos como Paradox, Oracle, Informix, Sybase, DbaselV, presentados en la tabla 1, así como otras herramientas de desarrollo y protocolos de comunicación como TCP/IP, NETBEUI, que se tenían en existencia, nos llevó a considerar un sistema DBMS montado en arquitectura cliente/servidor eligiendo a SYBASE como DBMS para correr en el servidor.

PRODUCTO	ORACLE	INFORMIX
REQUERIMIENTOS EN HARWARE	PC's con procesador 386 o mayor, 32 Mb en RAM, de 25 a 50 Mb en disco, unidad CD-ROM IBM RS/6000, Sun 4/SPARC Apple Mac OS, A UX	Mainframes Sistemas abiertos
CLIENTES QUE SOPORTA	MS-Windows, MACINTOSH, OS/2, MS-DOS, VMS, Sun OS, Solaris	UNIX, Windows Macintosh
SISTEMA OPERATIVO RED	Novell, SCO UNIX Windows NT, OS/2	Novell Windows NT
INTERFAZ GRAFICA DEL SISTEMA OPERATIVO DE RED	Windows Macintosh	Windows Macintosh Caracter
SISTEMA MANEJADOR DE BASES DE DATOS Y/O LENGUAJE	RDBMS(relacional) SQL, C++, FORTRAN COBOL, Net TCP/IP	RDBMS(relacional) ODBC, C COBOL
HERRAMIENTAS PARA DESARROLLO DE APLICACIÓN	ORACLE FORM4.5 ORACLE GRAPHICS 2.5 ORACLE REPORTS 2.5 Soporta objetos multiples	INFORMIX-NewEra INFORMIX-Menus INFORMIX-SQL INFORMIX-ESQL para C
MÓDULOS DE CONECTIVIDAD PREESTABLECIDOS	DB2, SQL Rdb Server de Microsoft	Bases de datos de IBM

Tabla 1. Evaluación de algunas herramientas.

PRODUCTO	PARADOX 5.0	dBASE IV 2.1	SYBASE
REQUERIMIENTOS EN HARWARE	24 Mb eb disco	5 Mb en disco duro; 2 Mb adicionales para cada usuario	De 72 a 111 Mb EN DISCO, 32 Mb en RAM Corre en las plataformas: IBM, HP, Intel, Digital, Sequent, Data General, AT&T, Silicon, Grapics, Unysis, y Stratus
CLIENTES QUE SOPORTA	MS-DOS	SCO UNIX AT&T UNIX SUN Solaris	UNIX, VMS Solaris, OS/2 Windows, MS-DOS
SISTEMA OPERATIVO RED	Novell, Windows NT, DecNET Pathworks IBM LAN Server, Microsoft LAN Manager, UNIXWARE		UNIX VMS OS/2 Windows NT
INTERFASE GRAFICA DEL SISTEMA OPERATIVO DE RED	Windows		Windows, Macintosh OS/2
SISTEMA MANEJADOR DE BASES DE DATOS Y/O LENGUAJE	RDBMS(relacional) dBMS, DDE, DLLs C, C++ Pascal, SQL, ODBC		RDBMS(relacional) ANSI SQL Transact SQL
HERRAMIENTAS PARA DESARROLLO DE APLICACIÓN	Crystal Reports for dBASE Borland Database Engine 2.0 ReportSmith 2.5		Las aplicaciones se pueden desarrollar con más de 125 herramientas de desarrollo distintas.
MODULOS DE CONECTIVIDAD PREESTABLECIDOS	dBASE,Paradox Oracle, Sybase MS/SQL Server Informix		Sybase tienen más de 1000 asociaciones en aplicaciones y herramientas de desarrollo, para una gran variedad de mercados.

Tabla 1. (Continuación)

Como se observa en la tabla 1, Sybase se encuentra corriendo bajo el sistema operativo UNIX, a su vez es compatible con ciertos protocolos que permiten la conexión desde DOS, Windows, u OS/2 al servidor SQL. Las características complementarias de Sybase se mencionan a continuación:

- Soporta diferentes plataformas de comunicación:
  - ✓ HP9000
  - ✓ IBM(R) RS/6000
  - ✓ NEC
  - ✓ NOVELL(TM) Net Ware(R)
  - ✓ PC OS/2 (R)
  - ✓ PC WINDOWS NT
  - ✓ SUN-4/SUNos(TM)
- Consultas en línea.
- Procesador central de transacciones.
- SQL Structured Query Languaje.
- Ofrece control de concurrencia, procedimientos almacenados e integridad referencial.
- Diccionario de datos en línea.
- Soporta bases de datos distribuidas.
- En bases de datos soporta:
  - ✓ 32,767 BD para el servidor SQL Server.
  - ✓ Abre 16 bases de datos para una consulta.
  - ✓ Abre 16 tablas en una sola consulta.
  - ✓ 2 billones de tablas por base de Datos.
  - ✓ 250 Columnas por tabla.
  - ✓ 251 índices por tabla.
  - ✓ El número de renglones es limitado solo por espacio en disco.
  - ✓ 16 columnas para un índice compuesto.
  - ✓ 30 caracteres máximo para el nombre de la Bases de Datos.
  - ✓ 30 caracteres máximo para nombre de las tablas y columnas.

En general, Sybase en un manejador de bases de datos de alto rendimiento con un amplio conjunto de características, pero también tiene sus desventajas, como es una implementación limitada en SQL, y el costo del producto.

2.3.2.2. HARDWARE.

El *Hardware* son los dispositivos físicos y tangibles, volúmenes de almacenamiento secundario (discos duros, flexibles, o CD ROM), donde reside la base de datos, junto con unidades de control, canales, etc.

Dado que la información de la base de datos es amplia y va a ser accesada por diferentes usuarios a la vez, es necesaria una excelente cobertura en el servidor, misma con la que cumple "Tonatiuh".

"Tonatiuh" es una máquina SUN Solaris con sistema operativo SUN en su versión OS 3.5.1, cuenta con dos procesadores con una velocidad de 50 Mhz. cada uno, cuenta con 94 Mb en RAM para SQL Server, dos discos internos con capacidad de 500 Mb cada uno, y dos discos externos con una capacidad de 3 Gb cada uno.

Los equipos utilizados como terminales para el usuario, los clientes, serán computadoras personales PC conectadas a la red de comunicación a través de Windows y deben cumplir necesariamente con los siguientes requisitos mínimos: 8 MB en RAM para SQL Server, procesador 486DX2 a 66 Mhz, disco duro de 240 Mb, monitor VGA a color, drive 3 ½ HD, teclado y Ratón.

2.3.2.3. USUARIO FINAL.

Las necesidades del usuario son el punto de partida para el desarrollo de nuestro sistema, por lo que se consideran tres clases de usuarios:

- *El programador de aplicaciones*: es el encargado de escribir programas de aplicación que utilice bases de datos. Los programas en sí pueden ser aplicaciones convencionales de procesamiento por lotes o programas en línea diseñados para apoyar a un usuario final, que interactúa con el sistema desde una terminal en línea.

- *Usuario final* es el que accesa a la base de datos desde una terminal, puede emplear un lenguaje de consulta proporcionado como parte integral del sistema o recurrir a un programa de aplicación escrito por un usuario programador que acepte órdenes desde la maquina cliente o a su vez formule solicitudes al DBMS.
- *Administrador de Bases de Datos* (DBA) es la persona encargada del control y seguridad general del sistema de Bases de Datos.

Para el éxito del sistema tenemos como prioridad las necesidades del usuario, mismo que pide un sistema entrenador que le resulte fácil de aprender, de utilizar, que sea directo y no demasiado estricto para su mejor y máximo uso, esto a través de menús y botones de selección agradables a la vista.

El programa por debajo del cual todas las operaciones son invisibles al usuario se conoce como: *interfaz gráfica del usuario*.

Se eligió la herramienta *Microsoft Visual Basic 3.0 - Edición Profesional* para la programación de la aplicación, la cual por medio de su librería para servidores SQL permite el desarrollo de una interfaz gráfica al usuario en ambiente Windows. Se basa en las librerías con extensión .DLL de "Base de Datos", que permite que el programa realice la conexión a Sybase, se formulen comandos SQL, los envíen y procesen los resultados.

Visual Basic utiliza el protocolo ODBC *(Open Database Connectivity)* de Microsoft para que la aplicación en Windows se comunique fácilmente con el sistema manejador de bases de datos. Esta interfaz de aplicación (API) como se le conoce en el mercado, ofrece gran compatibilidad con Sybase.

2.4. ANÁLISIS DEL SISTEMA.

Después de haber establecido la presentación del problema y el planteamiento de la solución, se establece de forma general, un sistema integral como el mostrado en la figura 2.3.

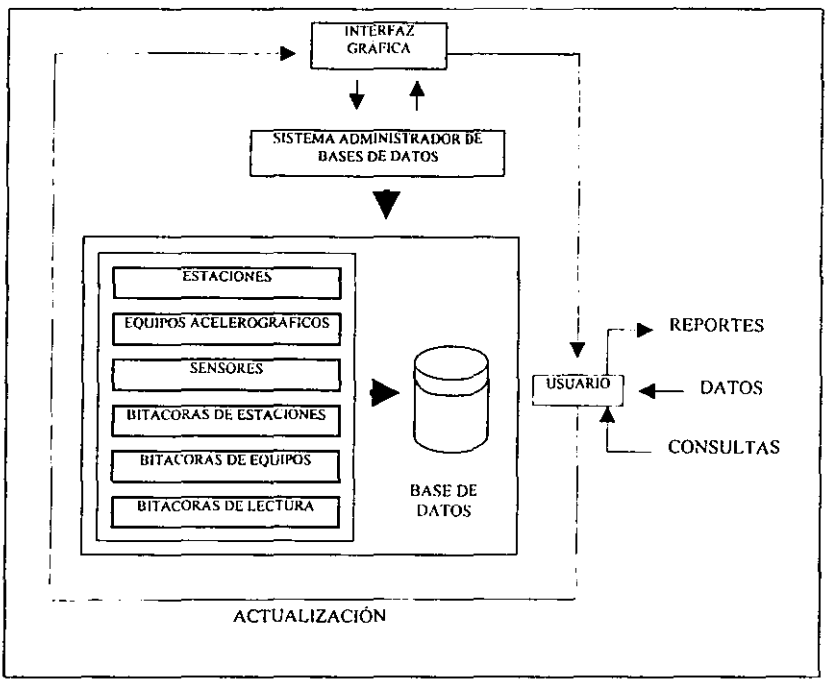


Fig. 2.3. Diagrama general del sistema de instrumentos y estaciones

De acuerdo al diagrama anterior se menciona la información requerida. Se necesitan conocer las características generales de las estaciones:

- Nombre de la estación.
- Clave de la estación.
- Lugar de la estación (se encuentra en campo libre, estructura o pozo).
- Ubicación (dirección precisa, calle, colonia, estado, etc.).

- Localización, coordenadas geográficas (Lat. N y Long. W).
- Altitud aproximada sobre el nivel del mar (msnm).
- Tipo de suelo en el que está instalada (lugar, en el caso de estructuras o edificios).
- Institución responsable (P.e.: I. de I, Cenapred, CFE, etc.).
- Red a la que pertenece (P.e.: I. de I., Guerrero, Puebla, etc.).
- Fecha de instalación.
- Fecha de la última visita.
- Estado de operación (Ej.: actualmente en operación; fuera de operación en definitivo; prevista para su instalación; transferida a otra institución; fuera de operación temporalmente; estado no especificado).
- Fecha de retiro y causa del retiro.
- Acceso a la estación (instrucciones para llegar al lugar).
- Problemas más comunes (comentarios hechos en bitácora).
- Tipo de alimentación (energía eléctrica o energía solar y sus características propias del tipo: capacidad de carga, número de celdas, tipo de regulador, tipo y número de baterías, etc.).
- En caso de existir celdas solares se desea conocer su capacidad, y la altura de la torre.
- Se desea conocer el historial de los equipos instalados en la estación por rango de fechas, haciendo referencia por canal, por rango, orientación, rumbo, umbral de disparo, memoria de preevento y memoria de postevento.

También se necesitan conocer las características de cada equipo acelerográfico, así como el historial de los cambios hechos a sus parámetros, en qué estaciones ha estado operando y comentarios generales como fallas que ha tenido y pruebas que se le han hecho. Por lo que se requiere de lo siguiente:

- Modelo del equipo acelerográfico.
- Número de serie.
- Fabricante.

- Propietario (Institución a cargo).
- Fecha de recepción en el laboratorio.
- Datos generales del equipo como número de canales, velocidad de muestreo por canal, intervalo de muestreo por canal, etc. cabe mencionar que para algunos equipos, estos parámetros son fijos, y para otros, son parámetros programables.
- Tipo de sensores (externos, internos).

Para los sensores se deben incluir las características de:

- Modelo.
- Número de serie.
- Rango por canal.
- Fecha de modificación del rango.
- Ganancia por canal.
- Frecuencia natural por canal.
- Amortiguamiento por canal.

Las bitácoras de estaciones, equipo y lectura, son combinación de los datos anteriores, propios de estación, equipo acelerográfico y sensor.

Una vez considerados los requisitos del sistema y tomado las decisiones sobre el software y hardware a utilizar, es necesario realizar un análisis más profundo sobre la manera en que van a ser manejados los datos.
Para esto es necesario utilizar diversas técnicas, como son:

✓Diagramas de flujo de datos.

✓Diagrama entidad/relación.

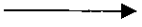
La ventaja principal de la utilización de estas herramientas, es que permiten visualizar detalladamente como se encuentran los datos en una primera instancia, y como a través de estos se obtiene información.

2.4.1. DIAGRAMA DE FLUJO DE DATOS.

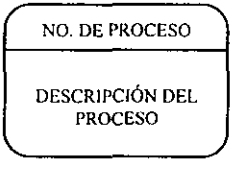
Una técnica para representar el flujo de la información a través del sistema, son los llamados diagramas de flujo de datos (DFD), que permiten visualizar al sistema como una red de procesos funcionales, conectados entre sí por conductos y tanques de almacenamiento.

Los componentes de un DFD son:

a) *Flujo de datos.* Se utiliza para describir el movimiento de datos de una parte del sistema a otra. Se identifica gráficamente como una flecha que entra o sale de un proceso:

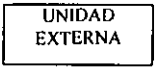


b) *Proceso.* Es una parte del sistema que transforma de datos de entrada a datos de salida, su representación gráfica es la siguiente:

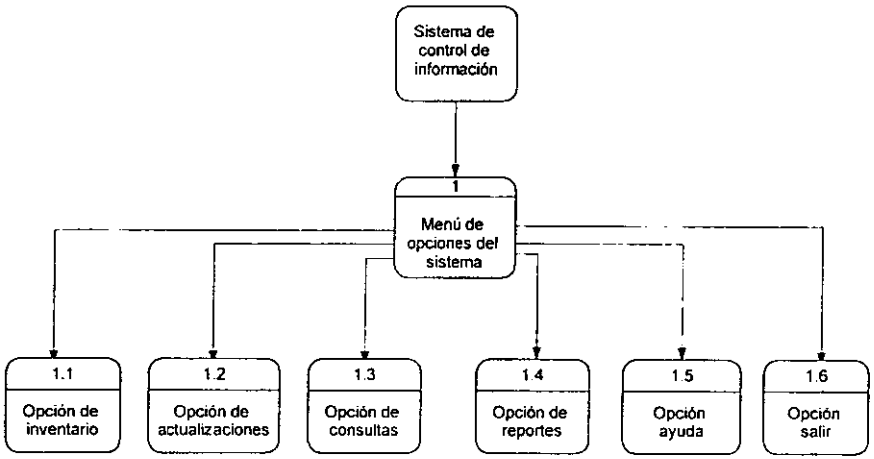


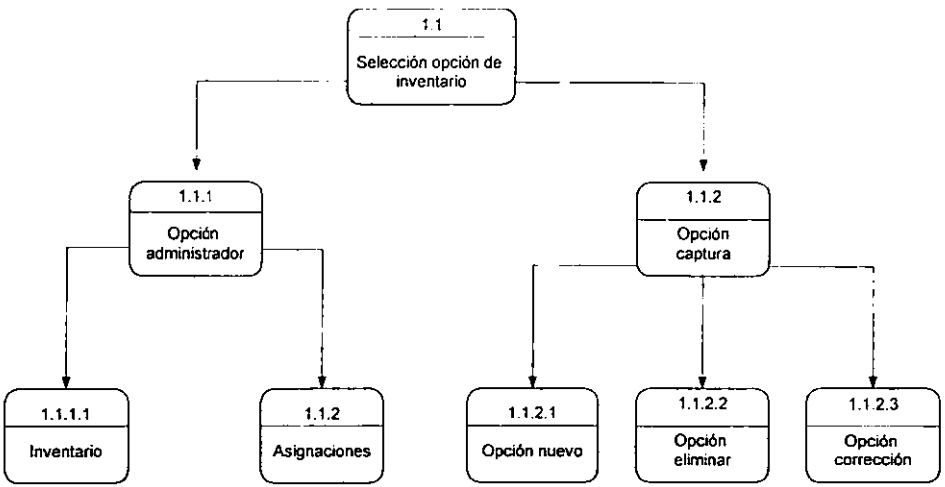
c) *Almacenamiento.* Es donde se almacena un conjunto de datos en reposo, su representación gráfica es la siguiente.

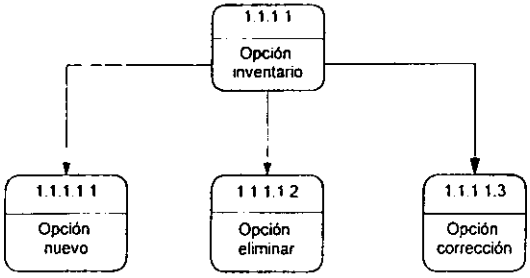
d) *Unidad externa o terminal.* Son unidades externas con las cuales se comunica el sistema, su representación gráfica es la siguiente:

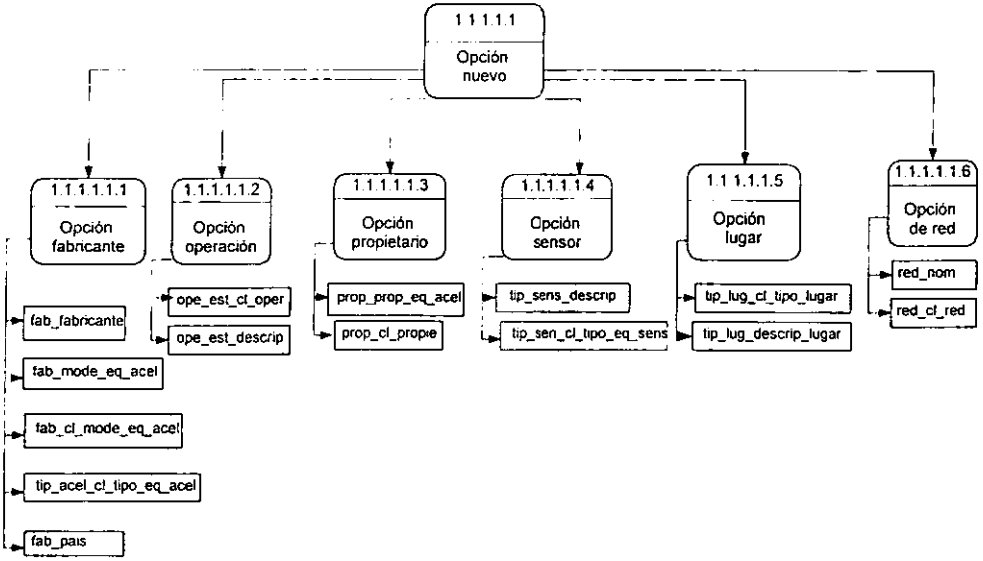


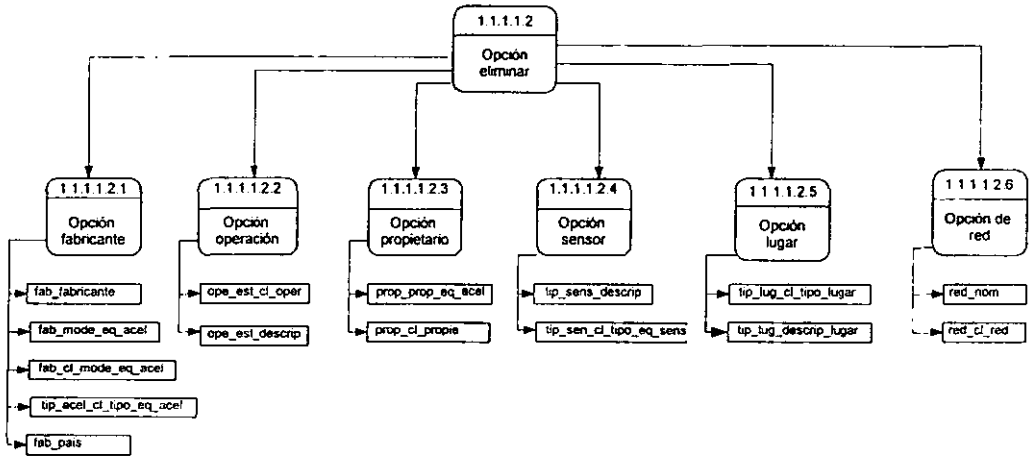
Con ayuda de las herramientas antes mencionadas se realizaron los diagramas de flujo de datos de nuestro sistema, mismos que se muestran a continuación.

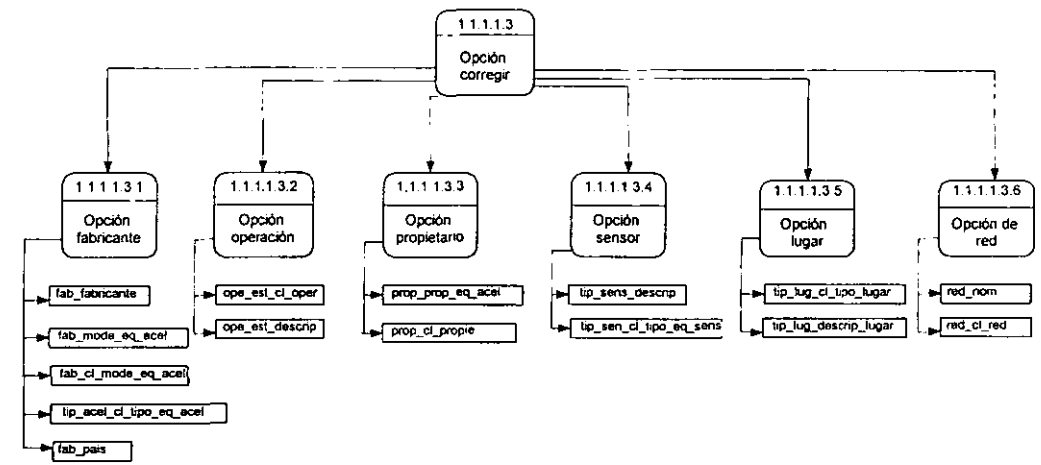

Sistema de control de información
1
Menú de opciones del sistema
1.1
Opción de inventario
1.2
Opción de actualizaciones
1.3
Opción de consultas
1.4
Opción de reportes
1.5
Opción ayuda
1.6
Opción salir

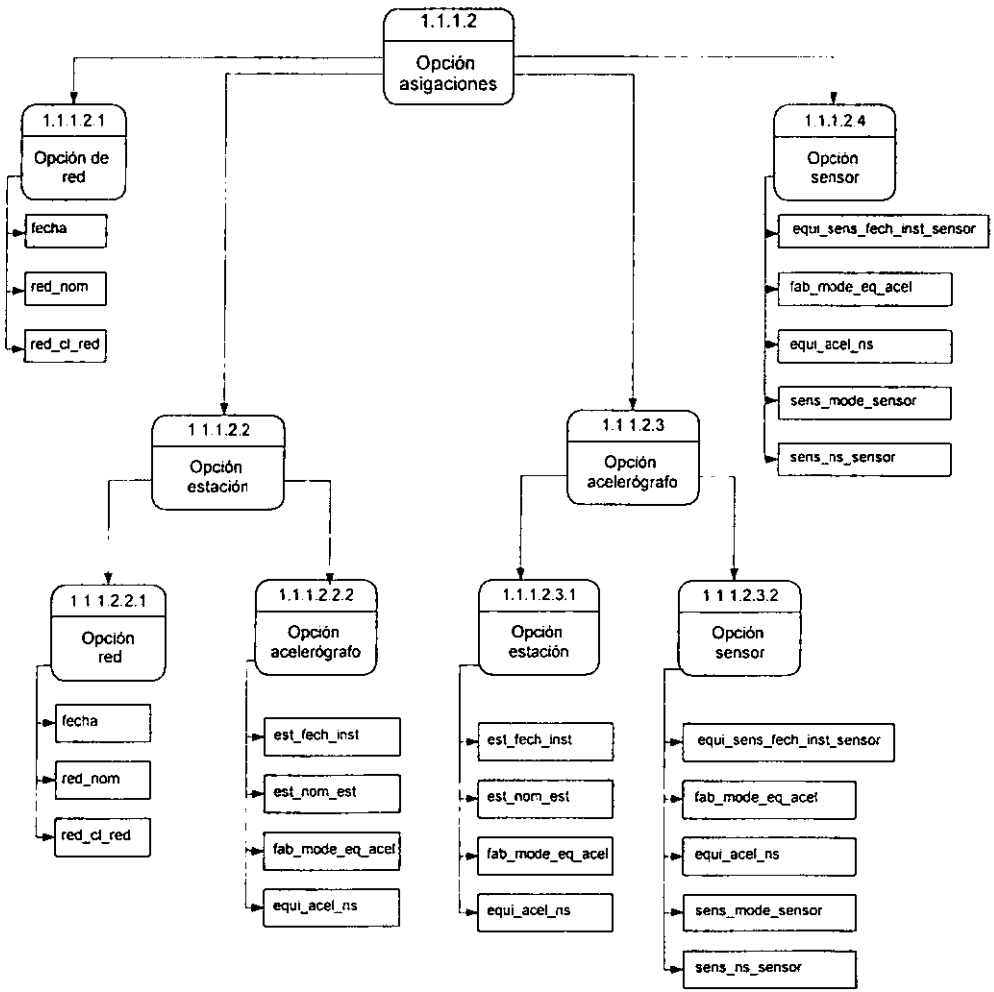

1.1
Selección opción de inventario
1.1.1
Opción administrador
1.1.2
Opción captura
1.1.1.1
Inventario
1.1.2
Asignaciones
1.1.2.1
Opción nuevo
1.1.2.2
Opción eliminar
1.1.2.3
Opción corrección

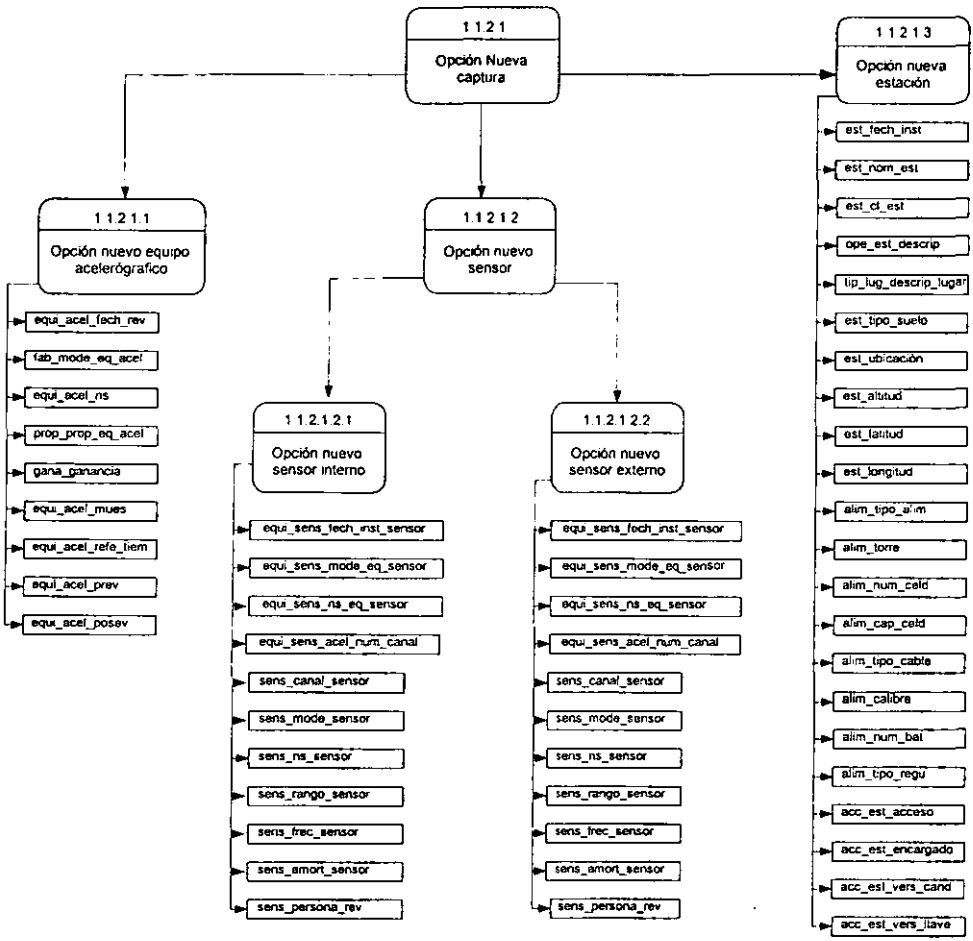

1.1.1.1
Opción inventario
1.1.1.1.1
Opción nuevo
1.1.1.1.2
Opción eliminar
1.1.1.1.3
Opción corrección

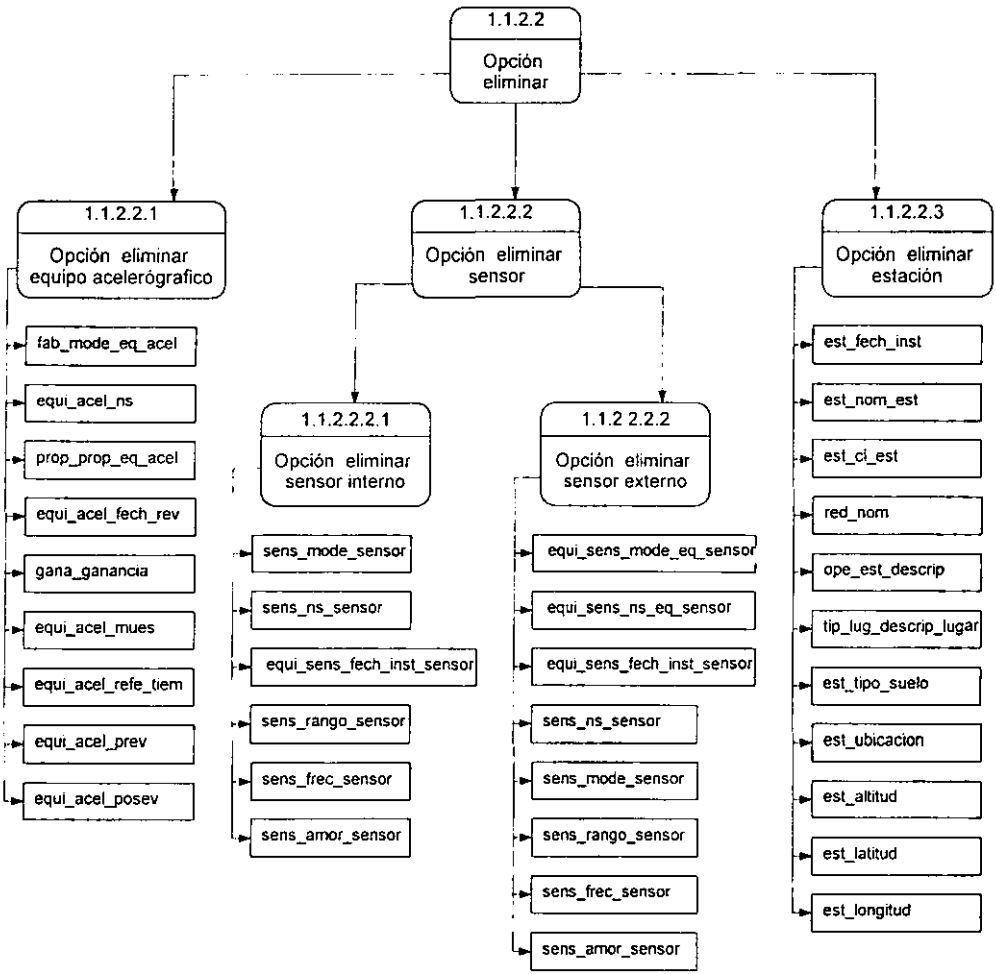

1.1.1.1.1
Opción nuevo
1.1.1.1.1.1
Opción fabricante
1.1.1.1.1.2
Opción operación
1.1.1.1.1.3
Opción propietario
1.1.1.1.1.4
Opción sensor
1.1.1.1.1.5
Opción lugar
1.1.1.1.1.6
Opción de red
fab_fabricante
fab_mode_eq_acel
fab_cl_mode_eq_acel
tip_acel_cl_tipo_eq_acel
fab_pais
ope_est_cl_oper
ope_est_descrip
prop_prop_eq_acel
prop_cl_propie
tip_sens_descrip
tip_sen_cl_tipo_eq_sens
tip_lug_cl_tipo_lugar
tip_lug_descrip_lugar
red_nom
red_cl_red

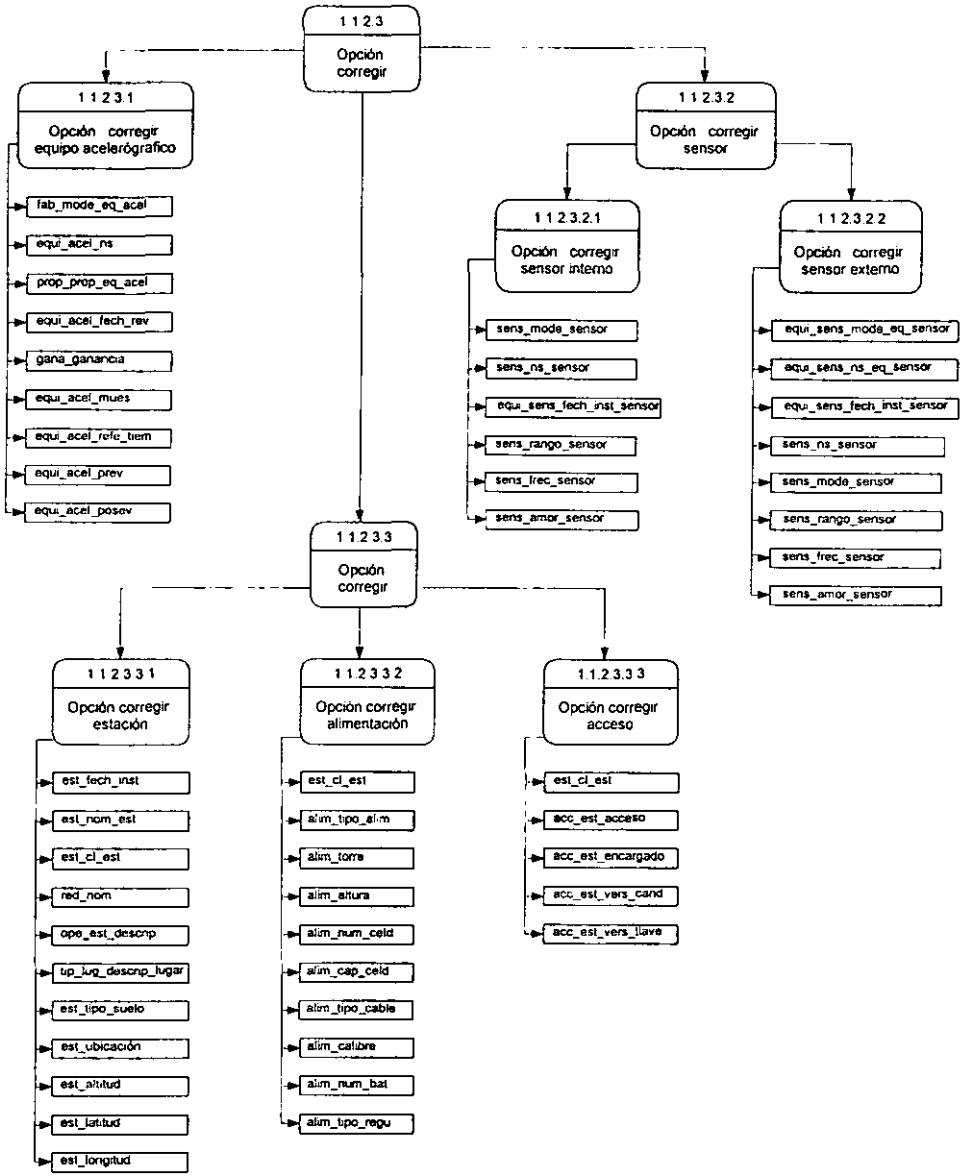

1.1.1.1.2
Opción eliminar
1.1.1.1.2.1
Opción fabricante
1.1.1.1.2.2
Opción operación
1.1.1.1.2.3
Opción propietario
1.1.1.1.2.4
Opción sensor
1.1.1.1.2.5
Opción lugar
1.1.1.1.2.6
Opción de red
fab_fabricante
fab_mode_eq_acel
fab_cl_mode_eq_acel
tip_acel_cl_tipo_eq_acel
fab_pais
ope_est_cl_oper
ope_est_descrip
prop_prop_eq_acel
prop_cl_propie
tip_sens_descrip
tip_sen_cl_tipo_eq_sens
tip_lug_cl_tipo_lugar
tip_lug_descrip_lugar
red_nom
red_cl_red

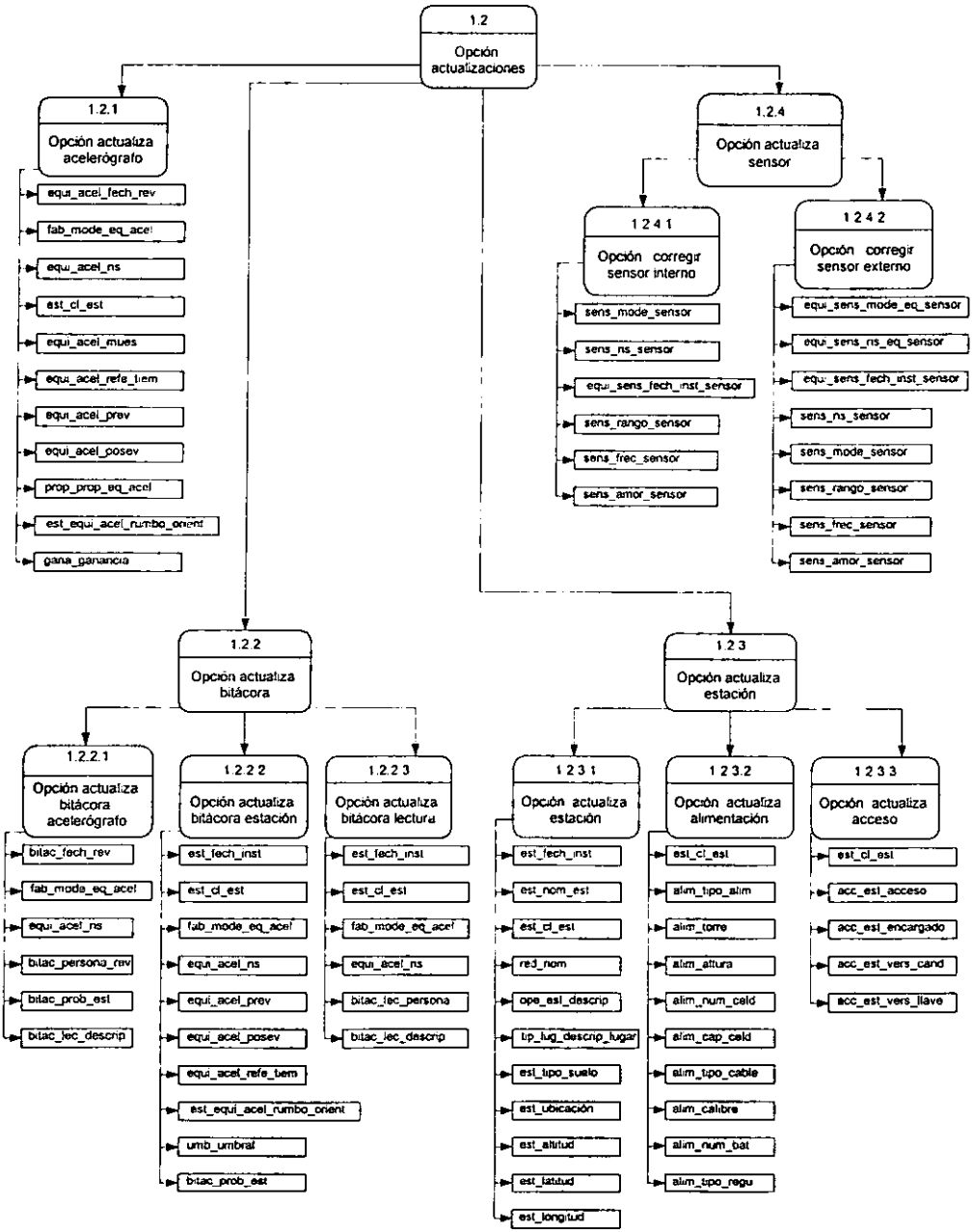

1.1.1.1.3
Opción corregir
1.1.1.1.3.1
Opción fabricante
1.1.1.1.3.2
Opción operación
1.1.1.1.3.3
Opción propietario
1.1.1.1.3.4
Opción sensor
1.1.1.1.3.5
Opción lugar
1.1.1.1.3.6
Opción de red
fab_fabricante
fab_mode_eq_acel
fab_cl_mode_eq_acel
tip_acel_cl_tipo_eq_acel
fab_pais
ope_est_cl_oper
ope_est_descrip
prop_prop_eq_acel
prop_cl_propie
tip_sens_descrip
tip_sen_cl_tipo_eq_sens
tip_lug_cl_tipo_lugar
tip_lug_descrip_lugar
red_nom
red_cl_red

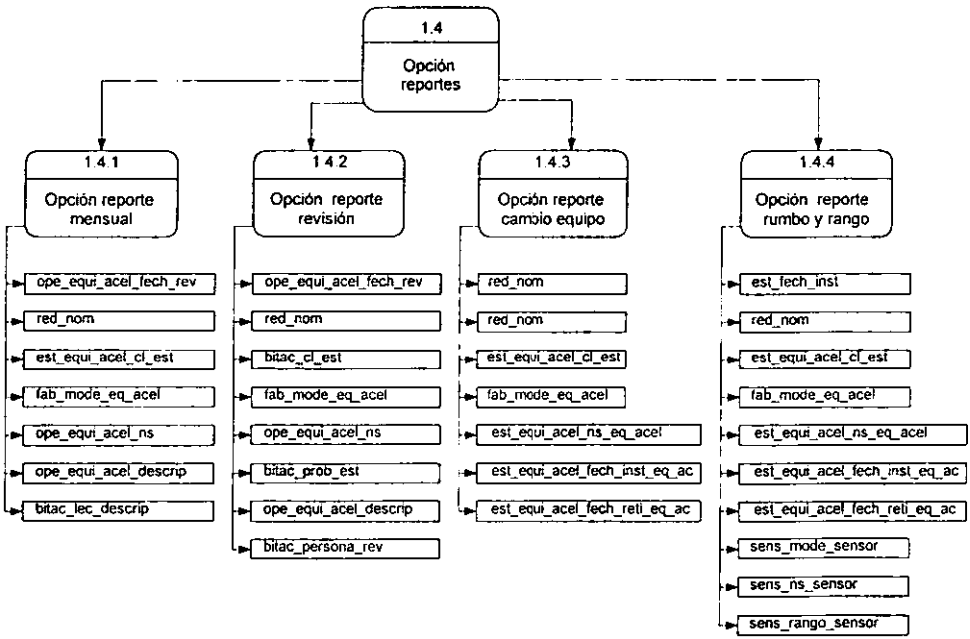

1.1.1.2
Opción asigaciones
1.1.1.2.1
Opción de red
fecha
red_nom
red_cl_red
1.1.1.2.2
Opción estación
1.1.1.2.2.1
Opción red
fecha
red_nom
red_cl_red
1.1.1.2.2.2
Opción acelerógrafo
est_fech_inst
est_nom_est
fab_mode_eq_acel
equi_acel_ns
1.1.1.2.3
Opción acelerógrafo
1.1.1.2.3.1
Opción estación
est_fech_inst
est_nom_est
fab_mode_eq_acel
equi_acel_ns
1.1.1.2.3.2
Opción sensor
equi_sens_fech_inst_sensor
fab_mode_eq_acel
equi_acel_ns
sens_mode_sensor
sens_ns_sensor
1.1.1.2.4
Opción sensor
equi_sens_fech_inst_sensor
fab_mode_eq_acel
equi_acel_ns
sens_mode_sensor
sens_ns_sensor

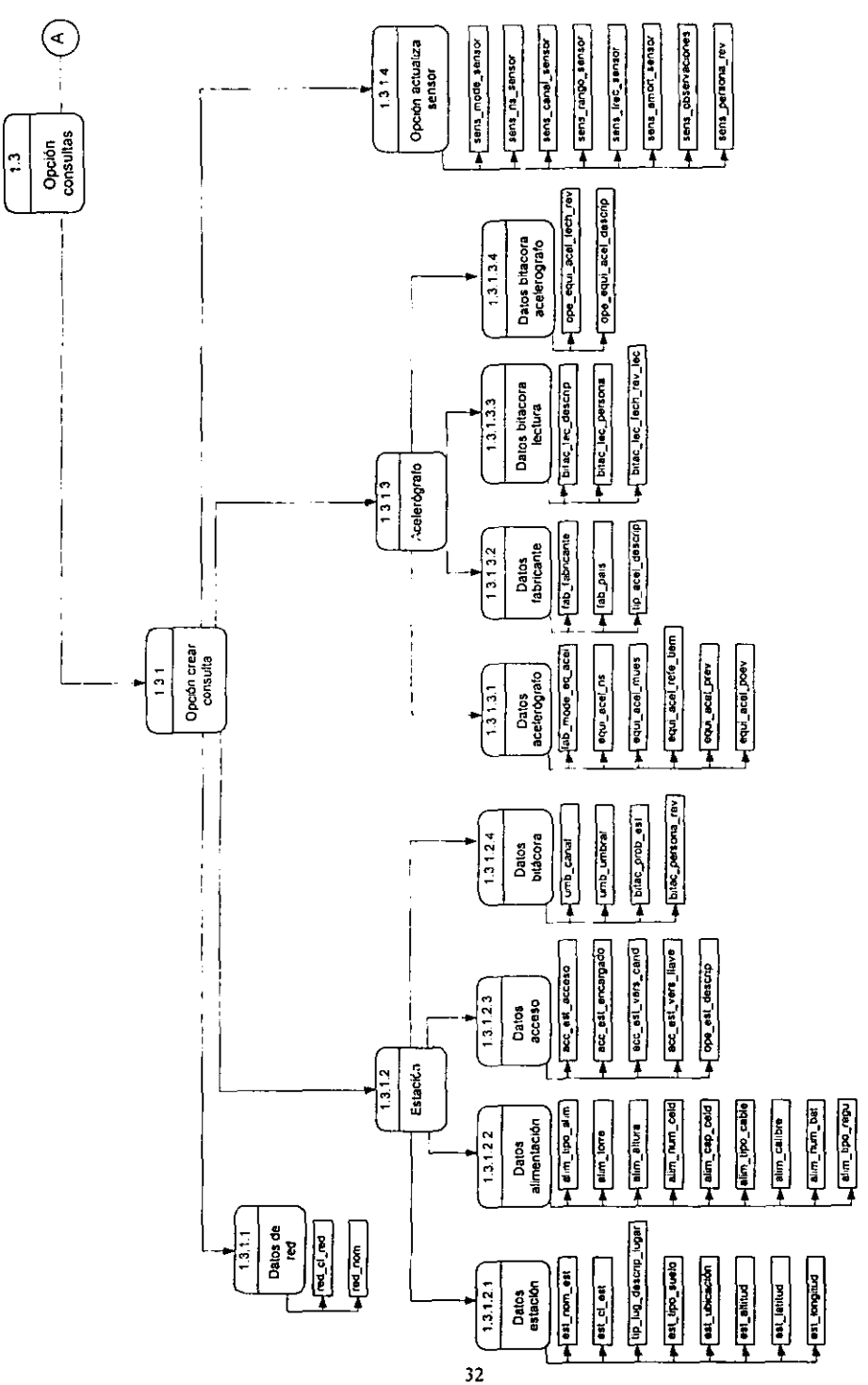

1.1.2.1
Opción Nueva captura
1.1.2.1.3
Opción nueva estación
est_fech_inst
est_nom_est
est_cl_est
ope_est_descrip
tip_lug_descrip_lugar
est_tipo_suelo
est_ubicación
est_altitud
est_latitud
est_longitud
alim_tipo_alim
alim_torre
alim_num_celd
alim_cap_celd
alim_tipo_cable
alim_calibre
alim_num_bat
alim_tipo_regu
acc_est_acceso
acc_est_encargado
acc_est_vers_cand
acc_est_vers_llave
1.1.2.1.1
Opción nuevo equipo acelerógrafico
equi_acel_fech_rev
fab_mode_eq_acel
equi_acel_ns
prop_prop_eq_acel
gana_ganancia
equi_acel_mues
equi_acel_refe_tiem
equi_acel_prev
equi_acel_posev
1.1.2.1.2
Opción nuevo sensor
1.1.2.1.2.1
Opción nuevo sensor interno
equi_sens_fech_inst_sensor
equi_sens_mode_eq_sensor
equi_sens_ns_eq_sensor
equi_sens_acel_num_canal
sens_canal_sensor
sens_mode_sensor
sens_ns_sensor
sens_rango_sensor
sens_frec_sensor
sens_amort_sensor
sens_persona_rev
1.1.2.1.2.2
Opción nuevo sensor externo
equi_sens_fech_inst_sensor
equi_sens_mode_eq_sensor
equi_sens_ns_eq_sensor
equi_sens_acel_num_canal
sens_canal_sensor
sens_mode_sensor
sens_ns_sensor
sens_rango_sensor
sens_frec_sensor
sens_amort_sensor
sens_persona_rev

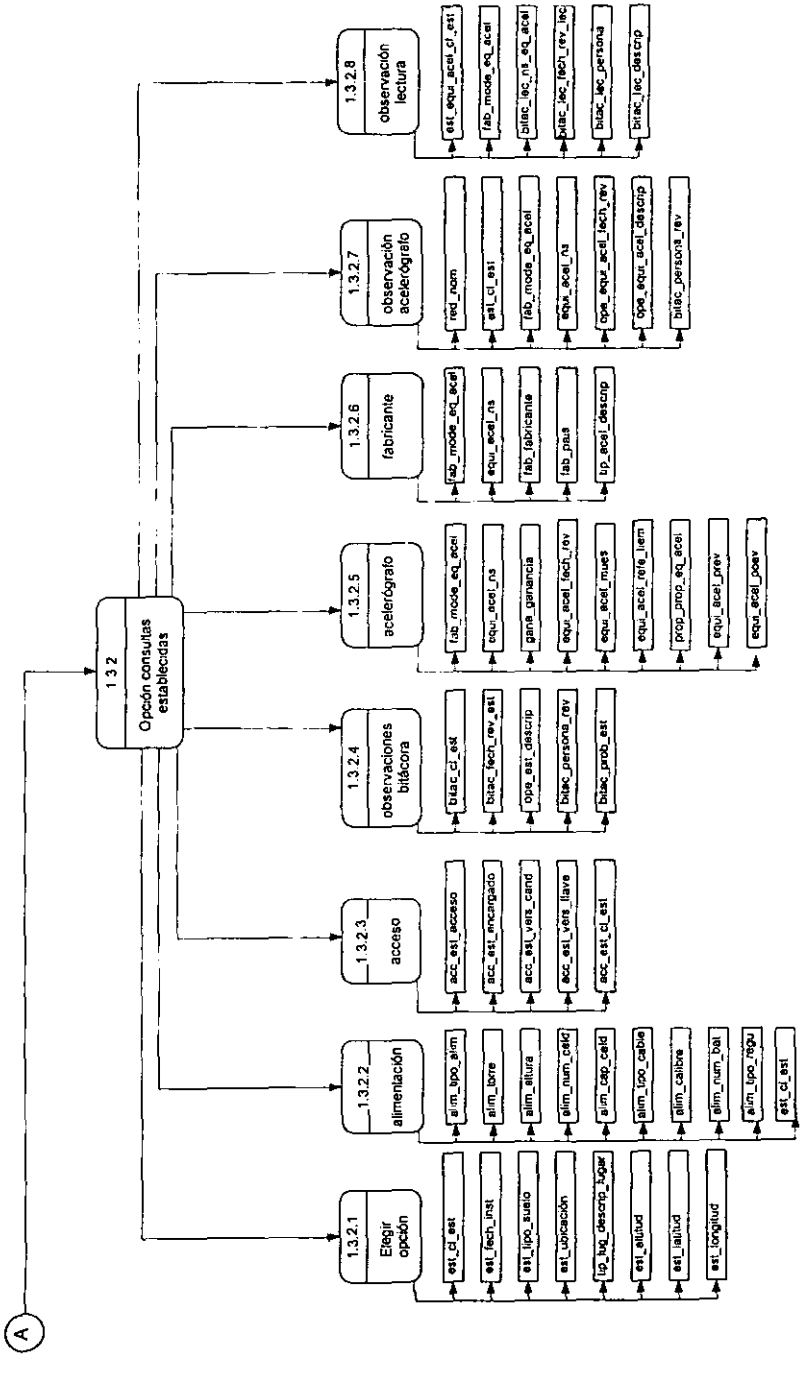

1.1.2.2
Opción eliminar
1.1.2.2.1
Opción eliminar equipo acelerógrafico
1.1.2.2.2
Opción eliminar sensor
1.1.2.2.3
Opción eliminar estación
fab_mode_eq_acel
equi_acel_ns
prop_prop_eq_acel
equi_acel_fech_rev
gana_ganancia
equi_acel_mues
equi_acel_refe_tiem
equi_acel_prev
equi_acel_posev
1.1.2.2.2.1
Opción eliminar sensor interno
1.1.2 2.2.2
Opción eliminar sensor externo
sens_mode_sensor
sens_ns_sensor
equi_sens_fech_inst_sensor
sens_rango_sensor
sens_frec_sensor
sens_amor_sensor
equi_sens_mode_eq_sensor
equi_sens_ns_eq_sensor
equi_sens_fech_inst_sensor
sens_ns_sensor
sens_mode_sensor
sens_rango_sensor
sens_frec_sensor
sens_amor_sensor
est_fech_inst
est_nom_est
est_ci_est
red_nom
ope_est_descrip
tip_lug_descrip_lugar
est_tipo_suelo
est_ubicacion
est_altitud
est_latitud
est_longitud


1.1.2.3
Opción corregir
1.1.2.3.1
Opción corregir equipo acelerógrafico
fab_mode_eq_acel
equi_acel_ns
prop_prop_eq_acel
equi_acel_fech_rev
gana_ganancia
equi_acel_mues
equi_acel_refe_tiem
equi_acel_prev
equi_acel_posev
1.1.2.3.2
Opción corregir sensor
1.1.2.3.2.1
Opción corregir sensor interno
sens_mode_sensor
sens_ns_sensor
equi_sens_fech_inst_sensor
sens_rango_sensor
sens_frec_sensor
sens_amor_sensor
1.1.2.3.2.2
Opción corregir sensor externo
equi_sens_mode_eq_sensor
equi_sens_ns_eq_sensor
equi_sens_fech_inst_sensor
sens_ns_sensor
sens_mode_sensor
sens_rango_sensor
sens_frec_sensor
sens_amor_sensor
1.1.2.3.3
Opción corregir
1.1.2.3.3.1
Opción corregir estación
est_fech_inst
est_nom_est
est_cl_est
red_nom
ope_est_descrip
tip_lug_descrip_lugar
est_tipo_suelo
est_ubicación
est_altitud
est_latitud
est_longitud
1.1.2.3.3.2
Opción corregir alimentación
est_cl_est
alim_tipo_alim
alim_torre
alim_altura
alim_num_celd
alim_cap_celd
alim_tipo_cable
alim_calibre
alim_num_bat
alim_tipo_regu
1.1.2.3.3.3
Opción corregir acceso
est_cl_est
acc_est_acceso
acc_est_encargado
acc_est_vers_cand
acc_est_vers_llave


1.2
Opción actualizaciones
1.2.1
Opción actualiza acelerógrafo
equi_acel_fech_rev
fab_mode_eq_acel
equi_acel_ns
est_cl_est
equi_acel_mues
equi_acel_refe_tem
equi_acel_prev
equi_acel_posev
prop_prop_eq_acel
est_equi_acel_rumbo_orient
gana_ganancia
1.2.4
Opción actualiza sensor
1.2.4.1
Opción corregir sensor interno
sens_mode_sensor
sens_ns_sensor
equi_sens_fech_inst_sensor
sens_rango_sensor
sens_frec_sensor
sens_amor_sensor
1.2.4.2
Opción corregir sensor externo
equi_sens_mode_eq_sensor
equi_sens_ns_eq_sensor
equi_sens_fech_inst_sensor
sens_ns_sensor
sens_mode_sensor
sens_rango_sensor
sens_frec_sensor
sens_amor_sensor
1.2.2
Opción actualiza bitácora
1.2.2.1
Opción actualiza bitácora acelerógrafo
bitac_fech_rev
fab_mode_eq_acel
equi_acel_ns
bitac_persona_rev
bitac_prob_est
bitac_lec_descrip
1.2.2.2
Opción actualiza bitácora estación
est_fech_inst
est_cl_est
fab_mode_eq_acel
equi_acel_ns
equi_acel_prev
equi_acel_posev
equi_acel_refe_tem
est_equi_acel_rumbo_orient
umb_umbral
bitac_prob_est
1.2.2.3
Opción actualiza bitácora lectura
est_fech_inst
est_cl_est
fab_mode_eq_acel
equi_acel_ns
bitac_lec_persona
bitac_lec_descrip
1.2.3
Opción actualiza estación
1.2.3.1
Opción actualiza estación
est_fech_inst
est_nom_est
est_cl_est
red_nom
ope_esl_descrip
tip_lug_descrip_lugar
est_tipo_suelo
est_ubicación
est_altitud
est_latitud
est_longitud
1.2.3.2
Opción actualiza alimentación
est_cl_est
alim_tipo_alim
alim_torre
alim_altura
alim_num_celd
alim_cap_celd
alim_tipo_cable
alim_calibre
alim_num_bat
alim_tipo_regu
1.2.3.3
Opción actualiza acceso
est_cl_est
acc_esl_acceso
acc_esl_encargado
acc_est_vers_cand
acc_est_vers_llave


1.4
Opción reportes
1.4.1
Opción reporte mensual
1.4.2
Opción reporte revisión
1.4.3
Opción reporte cambio equipo
1.4.4
Opción reporte rumbo y rango
ope_equi_acel_fech_rev
red_nom
est_equi_acel_cl_est
fab_mode_eq_acel
ope_equi_acel_ns
ope_equi_acel_descrip
bitac_lec_descrip
ope_equi_acel_fech_rev
red_nom
bitac_cl_est
fab_mode_eq_acel
ope_equi_acel_ns
bitac_prob_est
ope_equi_acel_descrip
bitac_persona_rev
red_nom
red_nom
est_equi_acel_cl_est
fab_mode_eq_acel
est_equi_acel_ns_eq_acel
est_equi_acel_fech_inst_eq_ac
est_equi_acel_fech_reti_eq_ac
est_fech_inst
red_nom
est_equi_acel_cl_est
fab_mode_eq_acel
est_equi_acel_ns_eq_acel
est_equi_acel_fech_inst_eq_ac
est_equi_acel_fech_reti_eq_ac
sens_mode_sensor
sens_ns_sensor
sens_rango_sensor


1.3
Opción consultas
A
1.3.1
Opción crear consulta
1.3.1.1
Datos de red
red_cl_red
red_nom
1.3.1.2
Estación
1.3.1.2.1
Datos estación
est_nom_est
est_cl_est
tip_lug_descrip_lugar
est_tipo_suelo
est_ubicación
est_altitud
est_latitud
est_longitud
1.3.1.2.2
Datos alimentación
alm_tipo_alm
alm_torre
alm_altura
alm_num_celd
alm_cap_celd
alm_tipo_cable
alm_calibre
alm_num_bat
alm_tipo_regu
1.3.1.2.3
Datos acceso
acc_est_acceso
acc_est_encargado
acc_est_vers_candi
acc_est_vers_llave
ope_est_descrip
1.3.1.2.4
Datos bitácora
umb_canal
umb_umbral
bitac_prob_esi
bitac_persona_rev
1.3.1.3
Acelerógrafo
1.3.1.3.1
Datos acelerógrafo
lab_mode_ac_acel
equi_acel_ns
equi_acel_mues
equi_acel_refe_tiem
equi_acel_prev
equi_acel_posev
1.3.1.3.2
Datos fabricante
fab_fabricante
fab_pais
tipo_acel_descrip
1.3.1.3.3
Datos bitacora lectura
bitac_lec_descrip
bitac_lec_persona
bitac_lec_fech_rev_lec
1.3.1.3.4
Datos bitacora acelerografo
ope_equi_acel_fech_rev
ope_equi_acel_descrip
1.3.1.4
Opción actualiza sensor
sens_mode_sensor
sens_na_sensor
sens_canal_sensor
sens_rango_sensor
sens_ires_sensor
sens_amort_sensor
sens_observaciones
sens_persona_rev


A
1.3.2
Opción consultas establecidas
1.3.2.1
Elegir opción
est_cl_est
est_fech_inst
est_tipo_suelo
est_ubicación
tip_hig_descrip_lugar
est_altitud
est_latitud
est_longitud
1.3.2.2
alimentación
alm_tipo_alm
alm_libre
alim_altura
alim_num_celd
alim_cap_celd
alim_tipo_cable
alim_calibre
alim_num_bat
alim_tipo_regu
est_cl_est
1.3.2.3
acceso
acc_est_acceso
acc_est_encargado
acc_est_vers_cand
acc_est_vers_llave
acc_est_cl_est
1.3.2.4
observaciones bitácora
bitac_cl_est
bitac_fech_rev_est
opa_est_descrip
bitac_persona_rev
bitac_prob_est
1.3.2.5
acelerógrafo
fab_mode_eq_acel
equ_acel_ns
gane_ganancia
equ_acel_fech_rev
equ_acel_mues
equ_acel_refe_liem
prop_prop_eq_acel
equ_acel_prev
equ_acel_poev
1.3.2.6
fabricante
fab_mode_eq_acel
equ_acel_ns
fab_fabricante
fab_pais
tip_acel_descrip
1.3.2.7
observación acelerógrafo
red_nom
est_cl_est
fab_mode_eq_acel
equ_acel_ns
ope_equ_acel_fech_rev
ope_equ_acel_descrip
bitac_persona_rev
1.3.2.8
observación lectura
est_equ_acel_cl_est
fab_mode_eq_acel
bitac_lec_ns_eq_acel
bitac_lec_fech_rev_lec
bitac_lec_persona
bitac_lec_descrip

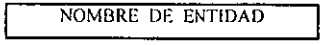
2.4.2. DIAGRAMA ENTIDAD / RELACIÓN.

Los *diagramas entidad/relación* (E-R) constituyen una técnica para representar la estructura lógica de una base de datos. Permiten ver los detalles y contenido de la base de datos al definir los datos y sus relaciones lógicas. Los objetivos de este son:

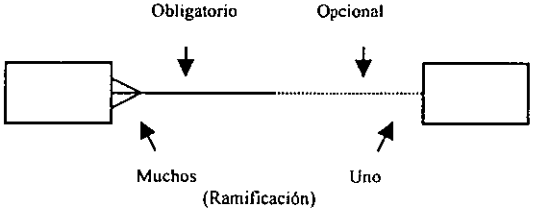
- Proporcionar un modelo preciso de las necesidades de información de la organización, que actuarán como un marco de trabajo para el desarrollo de sistemas nuevos o mejorados.
- Proporcionar un modelo independiente de cualquier almacenamiento de datos y método de acceso, que permita tomar decisiones objetivas de las técnicas de implementación del sistema.

Los componentes de este diagrama son:

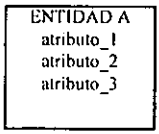
a) *Tipo de objetos (entidades).* Es una colección de objetos del mundo real y se representa por un rectángulo con un nombre en singular, con letras mayúsculas.



b) *Relaciones.* Los objetos se conectan entre sí por medio de relaciones, que son un conjunto de conexiones entre objetos. Se puede dar el caso de que existan múltiples relaciones entre objetos. Cada relación tiene dos extremos para cada uno de los cuales tiene un nombre, grado o cardinalidad (cuántos) y opcionalidad (opcional u obligatorio). Su representación es la siguiente:



c) *Atributo.* Es cualquier detalle que sirve para calificar, identificar, clasificar, cuantificar o expresar una entidad (no es necesario mostrar atributos en el diagrama entidad/relación). Su representación es:



La construcción de un diagrama E-R es un proceso que requiere múltiple refinamiento, que considera ampliaciones y reducciones a las entidades presentes o modificación a la definición de la misma.

A continuación se muestra el diagrama entidad/relación del sistema.

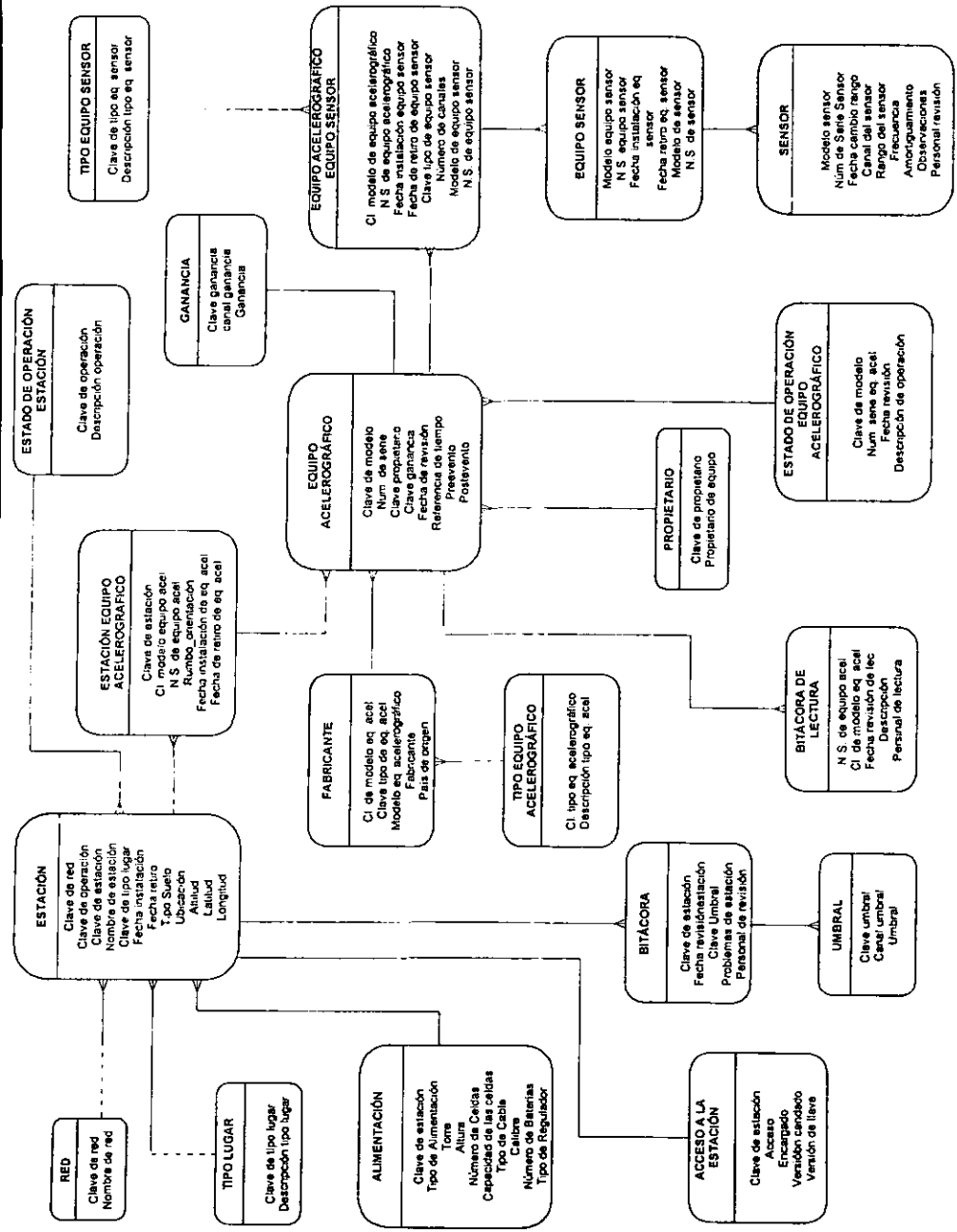


Diagrama Entidad - Relación.

2.4.3. NORMALIZACIÓN DE LA BASE DE DATOS.

Una vez modelada la información se procede a la normalización de la base de datos, cuyo objetivo es construir relaciones evitando la redundancia de datos, donde el problema que se origina es la inconsistencia de datos y no el espacio que se causa por redundancia.

Existen dos razones para convertir una base de datos en un conjunto de relaciones. Primero, tal conversión es un paso apropiado para ir del modelo a un conjunto de relaciones. Al final, cada tabla se convierte en una relación del sistema. La otra razón es más importante, ya que existe una teoría basada en fundamentos matemáticos que especifica cómo construir tablas evitando la redundancia de datos, lo que será después un conjunto de relaciones "normales". Si se organizan los datos en un conjunto de tales relaciones se asegura un buen diseño eliminando las anomalías que pudieran ocasionarse con las operaciones para insertar, borrar o actualizar datos.

En la teoría relacional, en una tabla, las columnas generalmente se llaman *atributos*, y las filas, *tuplas*. Cada relación tiene un único nombre en el sistema y cada columna o atributo tienen un único nombre en la relación.

Primera Forma Normal

Existen lo que se llaman formas normales, podemos decir que una relación esta en forma normal si todos los valores de sus columnas son simples. Una relación que posee esa propiedad está, al menos en la primera forma normal. Para poder describir la segunda y la tercera formas normales, es necesario definir una *clave de relación* como un conjunto de columnas cuyos valores seleccionan una única fila de relación, y pueden estar formadas por más de una columna.

De esta primera forma normal tenemos el siguiente ejemplo:

Clave de red	Clave de estación	Acelerógrafo	N. de serie acel.	Clave sensor	Fecha instalación.
GU	ATYC	DCA-333	110	E	10 enero 1997
GU	ATYC	DCA-310	124	E	25 enero 1998
II	DFVG	DCA-310	150	E	16 abril 1990
II	DFVG	ETNA	160	I	27 mayo 1991
II	DFVG	DCA-310	140	E	15 mayo 1994
OA	PNTP	DCA-333	185	E	14 diciembre 1997

Segunda Forma Normal.

Las relaciones donde subconjuntos de atributos clave determinan valores de atributos no clave puede estar en primera forma normal, pero puede no estar en segunda forma normal. Para estar en *segunda forma normal,* una relación no debe tener atributos no clave que dependan funcionalmente de parte de la clave de la relación.

Un ejemplo de las relaciones en segunda forma normal es el siguiente:

Redes

Clave de red	Clave de estación
GU	ATYC
II	DFVG
OA	PNTP

Equipos

Clave de estación	Acelerógrafo	N. de serie de acel.	Clave sensor	Fecha instalación
ATYC	DCA-333	110	E	10 enero 1997
ATYC	DCA-310	124	E	25 enero 1998
II	DCA-310	150	E	16 abril 1990
II	ETNA	160	I	27 mayo 1991
II	DCA-310	140	E	15 mayo 1994
II	DCA-333	185	E	14 diciembre 1997

Tercera Forma Normal.

Para estar en tercera forma normal, una relación debe encontrarse primero en segunda forma normal. Además, no debe tener ninguna dependencia funcional entre sus atributos no clave. Una *dependencia funcional* determina si un valor particular de un atributo (X) en una relación determina un valor particular de otro atributo (Y) para esa relación, es una forma de decir que si se conoce el valor de X se puede determinar un único valor de Y.

Aplicando la tercera forma normal a las relaciones se verán de la siguiente manera:

Redes

Clave de red	Clave de estación
GU	ATYC
II	DFVG
OA	PNTP

Estaciones

Clave de estación	Acelerógrafo
ATYC	DCA-333
ATYC	DCA-310
DFVG	DCA-310
DFVG	ETNA
DFVG	DCA-310
PNTP	DCA-333

Equipo

Acelerógrafo	N. de serie de acel.	Clave sensor	Fecha instalación
DCA-333	110	E	10 enero 1997
DCA-310	124	E	25 enero 1998
DCA-310	150	E	16 abril 1990
ETNA	160	I	27 mayo 1991
DCA-310	140	E	15 mayo 1994
DCA-333	185	E	14 diciembre 1997

Cuarta Forma Normal.

Las relaciones en cuarta forma normal no deben contener más de una dependencia multivaluada independiente o una dependencia multivaluada independiente junto con una dependencia funcional. Existe una dependencia multivaluada en una relación

cuando un valor de una columna o conjunto de columnas, X determina un conjunto de valores de otra columna, Y.

La cuarta forma se ejemplifica a continuación:

Redes

Clave de red	Clave de estación
GU	ATYC
II	DFVG
OA	PNTP

Estaciones

Clave de estación	Acelerógrafo
ATYC	DCA-333
ATYC	DCA-310
DFVG	DCA-310
DFVG	ETNA
DFVG	DCA-310
PNTP	DCA-333

Equipos

Acelerógrafo	N. de serie de acel.	Fecha instalación
DCA-333	110	10 enero 1997
DCA-310	124	25 enero 1998
DCA-310	150	16 abril 1990
ETNA	160	27 mayo 1991
DCA-310	140	15 mayo 1994
DCA-333	185	14 diciembre 1997

Sensores

Clave sensor	Sensor
E	Externo
I	Interno

Quinta Forma Normal.

La quinta forma normal se puede considerar como una extensión de la cuarta forma normal en el sentido de que ahora las dependencias multivaluadas no son independientes.

Nuestra base de datos fue llevada hasta la cuarta forma normal, misma en la que se realizaron pruebas y se encontró buen funcionamiento.

2.4.4 DICCIONARIO DE DATOS.

El *diccionario de datos* que se realizó es un catálogo del sistema que contiene información de la base de datos y de cómo está estructurada.

La estructura de los datos se presentan en 20 tablas que cumplen con lo preestablecido por el sistema, separadas a su vez en tablas principales (aquellas que reciben los datos y están activas) tabla 2 y auxiliares (aquellas que nos sirven para hacer referencias y consultas que nos faciliten el análisis de datos) tabla 3.

NOMBRE DE LA TABLA	DESCRIPCIÓN
T_ACCESO_ESTACION	Acceso a la estación
T_ALIMENTACION	Alimentación que mantiene a la estación
T_BITACORA	Bitácora de revisión de la estación
T_BITACORA_LECTURA	Problemas de lectura de los equipos
T_EDO_OPERACION_ESTACION	Edo de operación en el que se encuentra una estación
T_EDO_OPERA_EQ_ACELEROGRAFICO	Bitácora de operación del equipo acelerográfico
T_EQUIPO_ACELEROGRAFICO	Datos generales del equipo acelerográfico
T_ESTACION	Datos de la estación
T_FABRICANTE	Datos de los diferentes tipos de acelerógrafos
T_GANANCIA	Datos de la ganancia de equipos acelerográficos
T_PROPIETARIO	Datos del propietario del equipo
T_RED	Red acelerográfica a la que esta asociada una estación
T_SENSOR	Datos de los diferentes sensores
T_TIPO_LUGAR	Lugar donde se puede instalar una estación acelerográfica
T_TIPO_EQ_SENSOR	Tipo de equipo sensor
T_TIPO_EQ_ACELEROGRAFICO	Tipo de equipo acelerográfico
T_UMBRAL	Datos de umbral de disparo de la estación

Tabla 2. Tablas principales.

NOMBRE DE LA TABLA	DESCRIPCIÓN
TA_EQUIPO_SENSOR	Datos del equipo sensor y de los sensores
TA_EQ_ACELEROGRAFICO_EQ_SENSOR	Datos del equipo acelerográfico y del equipo sensor
TA_ESTACION_EQ_ACELEROGRAFICO	Datos de la estación y del equipo acelerográfico

Tabla 3. Tablas auxiliares.

A continuación se presenta la descripción de los campos de cada una de las tablas (principales y auxiliares), indicando el nombre del campo dentro de la base de datos y el tipo de dato que contiene.

T_ACCESO_ESTACION

Contiene la información concerniente a la ruta que se debe seguir para llegar a la estación y la persona con quien debe dirigirse al llegar, además de la versión del candado y llave que debe utilizar.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
acc_est_cl_est	char (4)	Clave única de la estación
acc_est_acceso	text	Acceso a la estación
acc_est_encargado	varchar (30)	Encargado de la estación en campo
acc_est_vers_cand	varchar (20)	Versión del candado
acc_est_vers_llave	char (2)	Versión de la llave

T_ALIMENTACION

Contiene información sobre las características de la fuente de alimentación que es instalada para cada estación, especificando cada una de las características asociadas.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
alim_cl_est	char (4)	Clave de la estación
alim_tipo_alim	varchar (15)	Alimentación que recibe el equipo
alim_torre	char (2)	Cuenta o no cuenta con torre
alim_altura	float	Altura en metros de la torre
alim_num_celd	smallint	Núm. de celdas que tiene la torre de una estación
alim_cap_celd	text	Capacidad de las celdas: amperes o watts
alim_tipo_cable	varchar (15)	Tipo de cable utilizado.
alim_calibre	smallint	Calibre del cable utilizado
alim_num_bat	smallint	Número de baterías instaladas por estación
alim_tipo_regu	varchar (20)	Tipo de regulador que requiere la alimentación para operar.

T_BITACORA

Contiene la información relacionada con los problemas que se hayan presentado en una estación durante la revisión. Por otro lado, aunque el umbral de disparo esta asociado al equipo acelerográfico, quien determina el umbral es la estación en una fecha determinada.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
bitac_cl_est	char (4)	Clave de la estación
bitac_fech_rev_est	datatime	Fecha de revisión de la estación
bitac_cl_umbral	char(12)	Clave del umbral de disparo
bitac_prob_est	text	Problemas relacionados con la estación
bitac_persona_rev	char(25)	Personal encargado de la revisión

T_BITACORA_LECTURA

Contiene la información concerniente a las observaciones realizadas en el preprocesamiento de los registros, además de datos adicionales para llevar un control.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
bitac_lec_ns_eq_acel	smallint	Número de serie del equipo acelerográfico
bitac_lec_cl_mode_eq_acel	char (3)	Clave de modelo de eq. acelerográfico
bitac_lec_fech_rev_lec	datatime	Fecha de revisión del equipo acelerográfico
bitac_lec_descrip	text	Problemas de lectura de eq. acelerográfico
bitac_lec_persona	char(25)	Personal encargado del procesamiento del registro.

T_EDO_OPERACION_ESTACION

Esta tabla contiene información sobre los diferentes tipos de estados de operación en el que se puede encontrar una estación (fuera de operación, próxima a instalar, etc.).

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
ope_est_cl_oper	char(1)	Clave de operación de la estación
ope_est_descrip	varchar(35)	Descripción del edo. de operación actual de la estación

T_EDO_OPERA_EQ_ACELEROGRAFICO

Literalmente, esta es la bitácora de equipo. Contiene la información relacionada con los problemas que se hayan presentado en un acelerógrafo.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
ope_equi_acel_cl_mode	char(3)	Clave de modelo de equipo. acelerográfico
ope_equi_acel_ns	smallint	Número de serie del equipo acelerográfico
ope_equi_acel_fech_rev	datatime	Fecha de revisión de la estación
ope_equi_acel_descrip	text	Descripción de operación del eq. acelerográfico

T_EQUIPO_ACELEROGRAFICO

Contiene información sobre los parámetros que permiten identificar un equipo acelerográfico.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCION
equi_acel_cl_mode	char (3)	Clave de modelo de equipo acelerográfico
equi_acel_ns	smallint	Número de serie del equipo acelerográfico
equi_acel_cl_propie	char(2)	Clave de propietario del equipo acelerográfico
equi_acel_cl_gana	varchar(16)	Clave de la ganancia del equipo acelerográfico
equi_acel_fech_rev	datatime	Fecha de revisión de la estación
equi_acel_mues	smallint	Velocidad de muestreo
equi_acel_refe_tiem	varchar(25)	Referencias de tiempo
equi_acel_prev	float	Memoria de preevento
equi_acel_posev	float	Memoria de posevento

T_ESTACION

Contiene información sobre las características y parámetros relacionados a una estación acelerográfica.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
est_cl_red	char (2)	Clave de la red a la que pertenece la estación.
est_cl_oper	char (1)	Clave del edo. de operación de la estación.
est_cl_est	char (4)	Clave única para cada estación
est_nom_est	varchar (35)	Nombre completo de la estación
est_cl_tipo_lugar	smallint	Clave de lugar de la estación
est_fech_inst	datetime	Fecha de instalación de la estación
est_fech_reti	datetime	Fecha de retiro de la estación
est_tipo_suelo	varchar (100)	Tipo de suelo en el que se encuentra la estación
est_ubicación	text	Ubicación de la estación: Dirección
est_altitud	smallint	Altitud de la estación en metros sobre el nivel del mar
est_latitud	float	Latitud donde se encuentra la estación
est_longitud	float	Longitud donde se encuentra la estación

T_FABRICANTE

Contiene información básica sobre los diferentes tipos de instrumentos que se han operado en la red del I. de I.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
fab_cl_mode_eq_acel	char (3)	Clave de modelo de eq. acelerográfico
fab_tipo_eq_acel	char(1)	Tipo del equipo acelerográfico (Analógico, Digital, etc.)
fab_mode_eq_acel	varchar(11)	Modelo del equipo acelerográfico
fab_fabricante	varchar (35)	Nombre del fabricante
fab_pais	varchar(10)	País de fabricación del equipo acelerográfico

T_GANANCIA

Información de la ganancia y el número de canal de los equipos acelerográficos

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
gana_cl_gana	varchar (16)	Clave de la ganancia
gana_canal	char (2)	Número de canal
gana_ganancia	smallint	Ganancia del equipo acelerográfico

T_PROPIETARIO

Catálogo cuya información se refiere a los distintos propietarios de los equipos acelerográficos.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
prop_cl_propie	char (2)	Clave de propietario de equipo acelerográfico
prop_prop_eq_acel	varchar (75)	Propietario del equipo acelerográfico

T_RED

Catálogo que contiene los nombres y claves de las redes acelerográficas que se manejan en el Instituto.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
red_cl_red	char (2)	Clave de la red
red_nom	varchar (25)	Nombre completo de la red

T_SENSOR

Contiene la información sobre las características de los sensores. Se maneja cada sensor individualmente, no importando si es externo o interno.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
sens_mode_sensor	varchar (10)	Modelo del sensor.
sens_ns_sensor	int	Número de serie del sensor.
sens_fech_camb_rango	datetime	Si existe, fecha de cambio de rango del sensor
sens_canal_sensor	char(2)	Canal del sensor
sens_rango_sensor	float	Rango del sensor por canal.
sens_frec_sensor	float	Frecuencia natural del sensor por canal.
sens_amort_sensor	float	Amortiguamiento del sensor por canal.
sens_observaciones	text	Observaciones del personal de revisión
sens_persona_rev	char(25)	Personal de revisión

T_TIPO_LUGAR

Catálogo que contiene información sobre los tipos de lugares donde se ubica o instala una estación.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
tip_lug_cl_tipo_lugar	smallint	Clave de acceso al tipo de lugar
tip_lug_descrip_lugar	varchar(40)	Lugar donde se encuentra la estación (campo libre, presa, edificio, etc.)

T_TIPO_EQ_SENSOR

Catálogo que contiene la descripción sobre los tipos de equipo sensor que se tienen (interno o externo).

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
tip_sens_cl_tipo_eq_sens	char(1)	Clave del tipo de equipo sensor
tip_sens_descrip	char(7)	Tipo de equipo sensor que se tiene

T_TIPO_EQ_ACELEROGRAFICO

Catálogo que contiene la descripción del tipo de equipo acelerográfico que se maneja (analógico o digital).

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCION
tip_acel_cl_tipo_eq_acel	char(1)	Clave de tipo de equipo acelerográfico
tip_acel_descrip	varchar(10)	Tipo de equipo acelerográfico

T_UMBRAL

Datos del umbral de disparo de la estación.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCION
umb_cl_umbral	char (12)	Clave de umbral
umb_canal	char (2)	Número de canal
umb_umbral	smallint	Umbral de la estación

✓ TABLAS AUXILIARES

TA_EQUIPO_SENSOR

Contiene información que permite relacionar un equipo sensor con cada uno de los sensores que contiene. Como equipo sensor se está haciendo referencia a la caja, chasis, placa o tarjeta que identifica al conjunto de sensores, ya sean internos o externos.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCION
equi_sens_mode_eq_sensor	varchar(10)	Modelo del equipo sensor
equi_sens_ns_eq_sensor	smallint	Número de serie del equipo sensor
equi_sens_fech_inst_sensor	datatime	Fecha de instalación del sensor en el eq. sensor
equi_sens_fech_reti_sensor	datatime	Fecha de retiro del sensor del equipo sensor
equi_sens_mode_sensor	varchar(10)	Modelo del sensor
equi_sens_ns_sensor	int	Número de serie del sensor

TA_EQ_ACELEROGRAFICO_EQ_SENSOR

Contiene información que permite relacionar un equipo acelerográfico con su respectivo equipo sensor.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
equi_sens_acel_cl_mode_eq_acel	char(3)	Clave del modelo del equipo acelerográfico
equi_sens_acel_ns_eq_acel	smallint	Número de serie del equipo acelerográfico
equi_sens_acel_fech_inst_eq_s	datatime	Fecha de instalación de equipo sensor
equi_sens_acel_fech_reti_eq_s	datatime	Fecha de retiro de equipo sensor
equi_sens_acel_cl_tipo_eq_sens	char(1)	Clave de tipo de equipo sensor
equi_sens_acel_num_canal	smallint	Número de canales del equipo sensor
equi_sens_acel_mode_eq_sensor	varchar(10)	Modelo del equipo sensor
equi_sens_acel_ns_eq_sensor	smallint	Número de serie del equipo sensor

TA_ESTACION_EQ_ACELEROGRAFICO

Contiene información que permite relacionar una estación con un equipo acelerográfico en determinada fecha, y que rumbo-orientación tenía.

NOMBRE DEL CAMPO	TIPO DE DATO	DESCRIPCIÓN
est_equi_acel_cl_est	char(4)	Clave de la estación
est_equi_acel_cl_mode_eq_acel	char(3)	Clave del modelo del equipo acelerográfico
est_equi_acel_ns_eq_acel	smalint	Número. de serie del equipo acelerográfico
est_equi_acel_rumbo_orient	varchar(98)	Orientación y rumbo del equipo acelerográfico
est_equi_acel_fech_inst_eq_ac	datatime	Fecha de instalación del equipo acelerográfico
est_equi_acel_fech_reti_eq_ac	datatime	Fecha de retiro del equipo acelerográfico

Capítulo 3

Diseño del Sistema

CAPÍTULO 3

3. DISEÑO DEL SISTEMA.

Una vez que se tienen todos los elementos que permiten visualizar, analizar e implementar la base de datos, se comienza con su diseño desde el punto de vista físico. El DBMS que se utilizó es SYBASE que realiza las funciones de actualización, consulta, verificación de integridad y administración de recursos y para realizar dichas funciones requiere de los siguientes objetos:

- Tablas.
- Reglas.
- Triggers.
- Procedimientos almacenados.
- Vistas.

3.1 TABLAS Y VALORES NULOS.

El primer objeto en la creación de la base de datos son las tablas; una *tabla* es un conjunto de filas que tienen columnas asociadas con elementos de datos individuales. Con la creación de estas podemos agrupar conforme un mismo interés los datos relacionados entre sí, de esta forma se puede lograr independencia de datos y la correcta relación entre tablas.

Cuando se crea una tabla, se asignan nombres a sus columnas y un tipo de datos a cada columna. También puede especificarse si una columna concreta puede contener valores nulos, o indicarse cualquier restricción de integridad para las columnas de la tabla.

Ejemplo de cómo se crea una tabla T_RED

```
create table T_RED
    (red_cl_red char(2) not null,
    red_nom varchar(25) not null,
    primary key clustered(red_cl_red ))
```

3.2 REGLAS, TIPOS DE DATOS, DEFAULTS.

Basándose en las características de cada entidad o tabla cabe hacer notar que algunas de ellas: estación, alimentación, red, etc., tienen restricciones que deben cumplirse en la base de datos, por lo que se crearon algunas reglas.
Una *regla* es la estructura estricta que restringe las características de captura de datos o valores determinados dentro del campo de una tabla.

Un *default* o *valor predeterminado* es un valor para la columna de una tabla o un tipo de dato definido por el usuario que el servidor inserta automáticamente si no se introduce explícitamente un valor para dicha columna.
Los valores predeterminados y las reglas pueden usarse para mantener la integridad de los datos en toda la base de datos.

Para la base de datos se crearon cinco reglas, mismas que se asignaron a las tablas convenientes, a continuación se describe el funcionamiento de cada una de ellas.

1. La creación de esta regla es con el propósito de restringir la captura de la clave de estación. La regla r_cl_est permite que se inserte un nuevo registro sólo si la clave de la estación se encuentra formada por cuatro caracteres, donde las tres primeras son letras mayúsculas y no necesariamente la cuarta que puede ser un número. La creación de esta regla es:

```
create rule r_cl_est as @cl_est like "[A-Z][A-Z][A-Z]%"
```

2. La clave de las redes se restringe al uso de letras, por lo tanto la regla r_cl_red permite que se inserte un nuevo registro sólo si la clave de la red son dos letras mayúsculas. La regla es:

```
create rule r_cl_red as @cl_red like "[A-Z][A-Z]"
```

3. La clave de operación esta restringida al uso de una letra, por ello la regla r_cl_oper permite que se inserte un nuevo registro sólo si la clave de operación de la estación es una letra mayúscula. La regla es:

```
create rule r_cl_oper as @cl_oper like "[A-Z]"
```

4. La versión de la llave de la estación se conoce con números por tal razón la regla r_vers_llav permite que se inserte el dato correspondiente a la versión de la llave sólo si es un número y va del 1 al 9. La regla es:

```
create rule r_vers_llav as @vers_llav like "[1-9]"
```

5. El tipo de lugar es identificado por un número, así la regla r_tipo_lugar permite que solo se inserte un nuevo registro del tipo de lugar de la estación con un número del 1 al 9. La regla es:

```
create rule r_tipo_lugar as @tipo_lugar like "[1-9]"
```

3.3 PROCEDIMIENTOS ALMACENADOS.

Una de las características más importantes y más ampliamente utilizadas del servidor SQL es su soporte de procedimientos almacenados. Un *procedimiento almacenado* es una secuencia de sentencias de SQL a las que se les asigna un nombre, se compilan y se almacenan en el servidor. Una vez que el procedimiento almacenado ha sido definido en la base de datos, un programa de aplicación puede llamarlo por su nombre, utilizando la interfaz dblib (database library).

Debido a que los *procedimientos almacenados* se encuentran previamente analizados, compilados y optimizados, el servidor los puede manejar eficientemente. Además, el tráfico entre el servidor y el cliente se reduce porque el cliente puede enviar un solo mensaje que a su vez llama a una serie de sentencias SQL, en vez de enviar por separado cada una de ellas.

Por las ventajas que ofrecen los procedimientos almacenados se han realizado algunos que nos permitan traer información de las tablas a la aplicación de una manera más rápida. La descripción de algunos de los procedimientos almacenados se presenta a continuación.

- Procedimiento acceso(nombre varchar(25))

Este procedimiento envía la información necesaria para llegar al lugar donde se ubica la estación y recibe como parámetro la clave de la estación.

Ejemplo del código del procedimiento acceso:

```
create proc acceso(@clave char(4)) as
begin
    select
        acc_est_acceso, acc_est_encargado,
        acc_est_vers_cand,acc_est_vers_llave
    from
        T_ACCESO_ESTACION, T_ESTACION
    where
        acc_est_cl_est = est_cl_est and
        acc_est_cl_est = @clave
end
```

- Procedimiento adm_fabricante((@fabricante varchar(35), @modelo varchar(11), @cl_modelo char(3), @tipo varchar(10), @pais varchar(10))

Este procedimiento da de alta un nuevo registro de la tabla de fabricante y recibe como parámetros los datos de la misma.

- Procedimiento alimentacion(@clave char(4),@tipo varchar(15))

Este procedimiento se encarga de mostrar las características de alimentación de la estación y recibe como parámetros la clave de la estación y el tipo de alimentación.

- Procedimiento equipo(@modelo varchar(11), @nserie smallint)

Este procedimiento se encarga de mostrar los datos del equipo acelerográfico al usuario y recibe como parámetros el modelo del equipo y su número de serie.

- Procedimiento estacion_ec(@clave char(4))

Este procedimiento muestra al usuario los datos de la estación y basándose en esto recibe como parámetro la clave de la estación.

- Procedimiento fabricante(@modelo varchar(11))

Este procedimiento selecciona los datos del fabricante de acuerdo al modelo que se le pide.

- Procedimiento man_eq_acel(@fecha datetime, @modelo varchar(11), @nserie smallint, @propietario varchar(75), @gana_ganancia smallint, @velmues smallint, @refetiem varchar(10), @prevento float, @posevento float)

Este procedimiento da de alta los datos del equipo acelerográfico, checando a la vez que no este duplicado el registro y que el modelo haya sido registrado con anterioridad.

- Procedimiento man_estacion(@fecha datetime, @nombre varchar(35), @cl_est char(4), @operacion varchar(35), @lugar varchar(40), @suelo varchar(100), @ubicacion varchar(255), @altitud smallint, @latitud float, @longitud float)

Con este procedimiento es posible guardar los datos de la estación y verificar que no existan datos repetidos desde la interfaz gráfica.

- Procedimiento man_estacion_acc(@cl_est char(4), @acceso varchar(255), @encargado varchar(30), @candado varchar(20), @llave char(2))

Este procedimiento da de alta los datos de acceso a la estación, verifica que exista la estación y que no estén duplicados los datos.

- procedimiento man_estacion_alim(@cl_est char(4), @tipo_alim varchar(15), @torre char(2), @altura float, @num_celd smallint, @cap_celd varchar(255), @tipo_cable varchar(15), @calibre smallint, @num_bat smallint, @alim_tipo_regu varchar(20))

Este procedimiento verifica la existencia de una estación para dar de alta los datos referentes a la alimentación y que no se dupliquen.

- procedimiento man_estacion_red(@red varchar(25), @cl_est char(4))

Este procedimiento asocia una nueva estación con la red a la que pertenece, en caso de no existir asociación le asigna NP que significa que no pertenece a ninguna red.

3.5 TRIGGERS.

El *trigger* es otra herramienta muy útil en la construcción de aplicaciones, éste es un tipo especial de procedimiento almacenado que se activa automáticamente, su objetivo es garantizar la integridad de los datos por el servidor SQL.

Para cualquier evento que ocasione un cambio en la base de datos, se puede especificar una acción que ejecutará el DBMS en ese momento. Los casos que pueden disparar un *trigger* son intentos de insertar, borrar o actualizar registros en una tabla. Al conjunto de sentencias SQL que la conforma se le denomina trigger y se utilizan para implementar las reglas de integridad referencial.

La manera como funciona un *trigger* consiste en definir un conjunto de instrucciones SQL, que al ocurrir un evento se ejecutan, y abortan un movimiento a la base de datos en caso de que no se cumplan las reglas en él establecidas. Esto nos permitirá que ninguna tabla contenga registros "huérfanos" o que no tengan la referencia apropiada hacia otros registros.
Así por ejemplo, si se quieren insertar datos en la tabla T_ESTACION, deberá existir o estar dado de alta lo siguiente:

- Red acelerográfica.
- Estado de operación de la estación.
- Tipo de lugar donde se ubica o instala una estación.

En caso contrario, no dejará completar la transacción, enviando mensaje de que falta dar de alta algún dato.

La descripción de algunos de los triggers desarrollados s encuentra a continuación.

- Trigger: tr_acceso_estacion.

Este *trigger* no permite introducir en la tabla de acceso_estacion una clave de estación no dada de alta en la tabla de estación.

Ejemplo del código del trigger tr_acceso_estacion:

```
create trigger tr_acceso_estacion
on T_ACCESO_ESTACION
for insert, update as
if @@rowcount=0
   return
if(select count(*) from T_ESTACION, inserted, T_ACCESO_ESTACION
  where T_ESTACION.est_cl_est =inserted.acc_est_cl_est)=@@rowcount
begin
   print "Tratando de insertar una clave de estacion en la tabla acceso
          a estacion no valida en la tabla de estacion"
   rollback transaction
   return
end
commit transaction
return
```

- Trigger tr_alimentacion.

Este *trigger* no permite introducir en la tabla de alimentación una clave de estación no dada de alta en la tabla de estación.

- Trigger tr_bitacora.

Este *trigger* no permite introducir en la tabla de bitácora una clave de estación no dada de alta en la tabla de estación.

- Trigger tr_bitacora_lectura.

Este *trigger* no permite introducir en la tabla de bitacora_lectura una clave de equipo y número de serie de equipo acelerográfico que no hayan sido dados de alta en la tabla de equipo acelerográfico.

- Trigger tr_edo_opera_eq_acelerografico.

Este *trigger* no permite introducir en la tabla de estado de operación una clave y número de serie de equipo acelerográfico que no hayan sido dados de alta en la tabla de equipo acelerógrafico.

- Trigger tr_eq_acelerografico_eq_sensor.

Este *trigger* no permite introducir en la tabla auxiliar de equipo_acelerografico una clave y número de serie de equipo acelerográfico, ni insertar en la tabla auxiliar de equipo_acelerografico_equipo_sensor una clave del tipo de equipo sensor no valido en la tabla de tipo de equipo sensor.

- Trigger tr_equipo_acelerografico.

Este *trigger* no permite introducir en la tabla de equipo_acelerografico una clave de equipo acelerográfico que no haya sido dado de alta en la tabla de fabricante o insertar en la tabla de equipo acelerográfico una clave de propietario no dado de alta en la tabla de t_propietario.

- Trigger tr_estacion.

Este *trigger* no permite introducir en la tabla de estación una clave de red no válida en la tabla de red o insertar en la tabla de red una clave de operación no valida en la tabla de estado de operación de estación o insertar en la tabla de estación una clave de tipo de lugar no valida en la tabla tipo_lugar.

- Trigger tr_estacion_eq_acelerografico.

Este *trigger* no permite introducir en la tabla auxiliar de estacion_equipo_acelerografico una clave de estación no válida en la tabla de estación o insertar en la tabla auxiliar de estación_equipo_acelerografico una clave de modelo y un número de serie de equipo acelerógrafico no válidos en la tabla de equipo_acelerografico.

- Trigger tr_fabricante

Este *trigger* no permite introducir en la tabla fabricante una clave del tipo de equipo acelerógrafico no válida en la tabla de tipo de equipo acelerográfico.

3.6 VISTAS.

Una *vista* es una forma alternativa de ver los datos de una o más tablas. Se puede pensar en una *vista* como un marco a través del cual pueden verse los datos en los que se tienen interés (esta es la razón por la que se habla de ver los datos o cambiar los datos "mediante" una vista).

Una *vista* se deriva de una o más tablas reales cuyos datos están almacenados físicamente en la base de datos. Las tablas que originan la vista se denominan tablas base o tablas subyacentes. Una vista también se puede derivar de otra vista.

Las *vistas* se pueden utilizar para centrar, simplificar y personalizar la percepción de cada usuario proporcionando un mecanismo de seguridad al permitir que los usuarios sólo tengan acceso a los datos que necesitan.

Mediante una *vista,* los usuarios pueden consultar y modificar sólo los datos que pueden ver, de tal manera que el resto de la base de datos no es visible ni accesible. Por las ventajas que nos ofrecen, se han realizado algunas que no permitan al usuario modificar la información, solo fueron diseñadas para uso exclusivo de consultas con fines de seguridad para limitar el acceso a un subconjunto de filas o columnas de una tabla base, es decir, un subconjunto dependiente del valor.

La descripción de algunas de las vistas se presentan a continuación.

- Vista est_red1

Esta vista se encarga de mostrar los siguiente datos: nombre de la red, nombre y clave de las estaciones que conforman la red acelerográfica.

Ejemplo del código de la vista est_red1:

```
create view est_red1 as
select
    red_nom, est_nom_est, est_cl_est
from
    T_RED, T_ESTACION
where
    red_cl_red = est_cl_red and
    red_cl_red = 'II'
```

- Vista dat_acel2edificios

Esta vista muestra la clave de estación, el modelo y número de serie del equipo acelerográfico, estado de operación, tipo de sensor, modelo y número de serie del sensor.

- Vista dat_estacion3

Esta vista regresa la clave de la estación, la fecha de instalación, el tipo de suelo, la ubicación, tipo de lugar, altitud, latitud y longitud.

- Vista alim_estacion4

Esta vista contiene la clave de la estación, el tipo de alimentación, la torre, altura, número y capacidad de las celdas, tipo y calibre de cable, número de baterías y el tipo de regulador.

- Vista acceso_estacion5

Esta vista contiene el nombre y clave de la estación, el acceso, personal encargado, versión del candado y de la llave.

- Vista prob_estacion6

Esta vista contiene el nombre y clave de la estación, la fecha de revisión, el estado de operación, el personal de revisión y problemas de la estación.

- Vista dat_acelerografo7

Esta vista contiene la clave de la estación, modelo y número de serie del acelerógrafo, ganancia por canal, fecha de revisión, velocidad de muestreo, referencia de tiempo, preevento, postevento y propietario.

- Vista fab_acelerografo8

Esta vista contiene el modelo, número de serie y tipo del acelerógrafo, el fabricante y su país.

- Vista prob_acelerografo9

Esta vista contiene el nombre de la red, clave de la estación, el modelo, número de serie y comentarios o problemas del acelerógrafo, fecha de revisión y el personal de revisión.

- Vista prob_lectura10

Esta vista contiene la clave de estación, modelo y número de serie del equipo acelerográfico, la fecha el personal y los comentarios de la lectura.

- Vista dat_sensor11

Esta vista contiene nombre y número de serie del equipo sensor, fecha de instalación, modelo y número de serie del sensor, rango, número de canal, frecuencia, amortiguamiento, personal de revisión y comentarios.

- Vista dat_redestacion12

Esta vista contiene el nombre de la red, clave de la estación, altitud, latitud, longitud, tipo de alimentación, acceso, fecha de revisión y observaciones de la estación.

- Vista dat_estacionaceler13

Esta vista contiene la clave de estación, tipo de alimentación, acceso, modelo y número de serie del equipo acelerográfico, referencia de tiempo, preevento y postevento.

- Vista dat_acelsensor14

Esta vista contiene modelo y número de serie del equipo acelerográfico, referencia de tiempo, preevento, postevento, tipo de sensor, modelo y número de serie del sensor, número de canales y fecha de revisión.

CARACTERÍSTICAS ASOCIADAS A LAS TABLAS PRINCIPALES

A continuación se muestran las tablas y algunos de los objetos anteriores asociados a las mismas.

T_ACCESO_ESTACION

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLAS	NULOS	TRIGGERS
acc_est_cl_est	char (4)	*	1		*
acc_est_acceso	text				
acc_est_encargado	varchar (30)			*	
acc_est_vers_cand	varchar (20)				
acc_est_vers_llave	char (2)		4		

T_ALIMENTACION

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
alim_cl_est	char (4)	*	1		*
alim_tipo_alim	varchar (15)	*			
alim_torre	char (2)			*	
alim_altura	float			*	
alim_num_celd	smallint			*	
alim_cap_celd	text			*	
alim_tipo_cable	varchar (15)			*	
alim_calibre	smallint			*	
alim_num_bat	smallint			*	
alim_tipo_regu	varchar (20)			*	

T_BITACORA

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
bitac_cl_est	char (4)	*	1		*
bitac_fech_rev_est	datatime	*			
bitac_cl_umbral	char(12)				
bitac_prob_est	text				
bitac_persona_rev	char(25)				

T_BITACORA_LECTURA

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
bitac_lec_ns_eq_acel	smallint	*			*
bitac_lec_cl_mode_eq_acel	char (3)	*			*
bitac_lec_fech_rev_lec	datatime	*			
bitac_lec_descrip	text				
bitac_lec_persona	char(25)				

T_EDO_OPERACION_ESTACION

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
ope_est_cl_oper	char(1)	*	3		
ope_est_descrip	varchar(35)				

T_EDO_OPERA_EQ_ACELEROGRAFICO

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
ope_equi_acel_cl_mode	char(3)	*			*
ope_equi_acel_ns	smallint	*			*
ope_equi_acel_fech_rev	datatime	*			
ope_equi_acel_descrip	text				

T_EQUIPO_ACELEROGRAFICO

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
equi_acel_cl_mode	char (3)	*			*
equi_acel_ns	smallint	*			
equi_acel_cl_propie	char(2)				*
equi_acel_cl_ganancia	char(16)				
equi_acel_fech_rev	datatime	*			
equi_acel_mues	smallint				
equi_acel_refe_tiem	varchar (25)				
equi_acel_prev	float				
equi_acel_posev	float				

T_ESTACION

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
est_cl_red	char (2)	*	2		*
est_cl_oper	char (1)		3		*
est_cl_est	char (4)		1		
est_nom_est	varchar (35)				
est_cl_tipo_lugar	smallint		5		*
est_fech_inst	datetime				
est_fech_reti	datetime			*	
est_tipo_suelo	varchar (100)				
est_ubicacion	text				
est_altitud	smallint				
est_latitud	float				
est_longitud	float				

T_FABRICANTE

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
fab_cl_mode_eq_acel	char (3)	*			
fab_tipo_eq_acel	char(1)				*
fab_mode_eq_acel	varchar(11)				
fab_fabricante	varchar (35)				
fab_pais	varchar(10)				

T_GANANCIA

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
gana_cl_ganancia	varchar (16)	*			*
gana_canal	char (2)	*			
gana_ganancia	smallint				

T_PROPIETARIO

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
prop_cl_propie	char (2)	*			
prop_prop_eq_acel	varchar (75)				

T_RED

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
red_cl_red	char (2)	*	2		
red_nom	varchar (25)				

T_SENSOR

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
sens_mode_sensor	varchar (10)	*			
sens_ns_sensor	int	*			
sens_fech_camb_rango	datetime	*			
sens_canal_sensor	char(2)				
sens_rango_sensor	float				
sens_amort_sensor	float				
sens_frec_sensor	float				
sens_observaciones	text				
sens_persona_rev	char(25)				

T_TIPO LUGAR

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
tip_lug_cl_tipo_lugar	smallint	*	5		
tip_lug_descrip_lugar	varchar(40)				

T_TIPO_EQ_SENSOR

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
tip_sens_cl_tipo_eq_sens	char(1)	*			
tip_sens_descrip	char(7)				

T_TIPO_EQ_ACELEROGRAFICO

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
tip_acel_cl_tipo_eq_acel	char(1)	*			
tip_acel_descrip	varchar(10)	*			

T_UMBRAL

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
umb_cl_umbral	char (12)	*			*
umb_canal	char (2)	*			
umb_umbral	smallint				

✓ CARACTERÍSTICAS ASOCIADAS A TABLAS AUXILIARES

TA_EQUIPO_SENSOR

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLAS	NULOS	TRIGGERS
equi_sens_mode_eq_sensor	varchar(10)				*
equi_sens_ns_eq_sensor	smallint				*
equi_sens_fech_inst_sensor	datatime				
equi_sens_fech_reti_sensor	datatime			*	
equi_sens_mode_sensor	varchar(10)				*
equi_sens_ns_sensor	int				*

TA_EQ_ACELEROGRAFICO_EQ_SENSOR

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
equi_sens_acel_cl_mode_eq_acel	varchar(3)				*
equi_sens_acel_ns_eq_acel	smallint				*
equi_sens_acel_fech_inst_eq_s	datatime			*	
equi_sens_acel_fech_reti_eq_s	datatime				
equi_sens_acel_cl_tipo_eq_sens	char(1)				*
equi_sens_acel_num_canal	smallint				
equi_sens_acel_mode_eq_sensor	varchar(10)				
equi_sens_acel_ns_eq_sensor	smallint				

TA_ESTACION_EQ_ACELEROGRAFICO

NOMBRE DEL CAMPO	TIPO DE DATO	LLAVE PRIMARIA	REGLA	NULOS	TRIGGERS
est_equi_acel_cl_est	char(4)				*
est_equi_acel_cl_mode_eq_acel	char(3)				*
est_equi_acel_ns_eq_acel	smalint				*
est_equi_acel_rumbo_orient	varchar(98)				
est_equi_acel_fech_inst_eq_ac	datatime				
est_equi_:acel_fech_reti_eq_ac	datatime			*	

3. 7 TAMAÑO DE LA BASE DE DATOS

Dentro de la implementación del sistema algo de suma importancia es determinar el tamaño de la base de datos y calcular su crecimiento con el tiempo basándose en los datos que se modifican e incrementan.

Para esto se utilizó un procedimiento almacenado propio del manejador de bases de datos cuya estructura es:

sp_estspace table_name, no_of_rows, fill_factor, cols_to_max, textbin_len, iosec

donde:

- table_name.- es el nombre de la tabla,
- no_of_rows.- es el número de renglones en la tabla (>0),
- fill_factor.- es el factor de llenado del índice. El rango de valores van de 1 a 100. (por default = 0, usado en caso de que este lleno el factor interno)
- cols_to_max.- es una lista de la longitud variable de las columnas por la cual es usada la máxima longitud en lugar del promedio (por default = null)
- textbin_len es la longitud por renglón de todos los campos textos y binarios (por default = 0).
- iosec.- es el número de entradas y salidas por segundo de la máquina (por default = 30)

Ejemplo: sp_estspace T_RED, 10

name	type	idx_level	Pages	Kbytes
T_RED	data	0	8	16
T_RED_7812458381	clustered	0	1	2
T_RED_7812458381	clustered	1	1	2

Total_Mbytes
0.02

name	type	total_pages	time_mins
T_RED_7812458381	clustered	10	0

Haciendo cálculos para cada tabla obtenemos en total 0.46 Kb de la base sin datos, ahora si consideramos el crecimiento de la base por año tenemos 7.6 Mb y considerando un crecimiento lineal en 5 años tendríamos 38 Mb.

El historial de las redes acelerográficas a lo largo de 18 años no ha sido lineal porque no siempre se realiza el mismo número de revisiones al año, y el número de estaciones ha ido creciendo, aún así, para calcular el tamaño de la base lo consideramos lineal, de esta manera el tamaño aproximado que la base de datos ocupará será de 132 MB, que sumado a cinco años dará un total de 170 MB.

Capítulo 4

Interfaz Gráfica del Usuario

CAPÍTULO 4

4. INTERFAZ GRÁFICA DEL USUARIO.

La interfaz gráfica de usuario es el mecanismo a través del cual se establece un diálogo entre el programa y el humano.

En el diseño de la interfaz es necesario tomar en cuenta tanto los aspectos de tecnología, como los factores humanos. Algunas de las preguntas que deben ser planteadas y respondidas como parte del diseño de la interfaz del usuario son las siguientes: ¿Quién es el usuario? ¿Cómo aprende a interactuar con un sistema nuevo basado en computadora? ¿Qué espera el usuario del sistema? Esto con el fin de generar una presentación consistente de la interfaz.

Un buen diseño de los formatos y las pantallas de entrada de la interfaz debe satisfacer los objetivos de eficacia, precisión, facilidad de uso, consistencia, sencillez y atracción.

La eficacia significa que las formas y las pantallas de entrada satisfagan propósitos específicos del sistema de información de la administración, mientras que la precisión se refiere a un diseño tal que asegure una realización satisfactoria. La facilidad de uso implica que tanto las formas como las pantallas serán explícitas y no requerirán de tiempo adicional para descifrarse. La consistencia, en este caso, significa que las formas y las pantallas ordenen los datos de manera similar de una aplicación a otra, mientras que la sencillez se refiere a mantener en un mínimo los elementos indispensables que centren la atención del usuario. La atracción implica que el usuario disfrutará del uso o tránsito a través de las formas y las pantallas cuyos diseños les sean más atractivos.

Además es necesario que el sistema ofrezca al usuario una serie de ventajas como por ejemplo:

- Se puedan visualizar diferentes tipos de información simultáneamente, con el fin de que el usuario examine resultados y/o utilice un texto sin perder la conexión entre ambos.

- La utilización de iconos gráficos, menús desplegables, botones y técnicas de presentación continua que reduzcan el número de pulsaciones en el teclado y de esta manera aumentar la eficiencia en usuarios con poca experiencia en mecanografía o con la computadora.

Para el desarrollo de la interfaz gráfica del sistema se utilizó la herramienta Visual Basic versión 3.0 de Microsoft, la cual presenta las siguientes características:

- Permite el desarrollo de aplicaciones con los elementos estándar de cualquier producto bajo ambiente Windows, como la creación de menús, ventanas, botones, cajas de edición, cajas de opciones, gráficos, etc.

- Cada pantalla o menú que es creado en Visual Basic es un objeto que cuenta con *propiedades* que son los elementos que podemos modificar de un objeto, los *eventos* que son las acciones que realiza cuando sucede éste y las *funciones* que van a cumplir ese objeto.

- Acceso a diferentes bases de datos como son las bases creadas en Access, Foxpro 2.0, Foxpro 2.5, dBase III, dBase IV, Paradox y RTRIEVE, por medio del ODBC (Open Data Connectivity) que es el estándar de conectividad de bases de datos de Microsoft, el cual utiliza manejadores para cada DBMS trabajando así en un ambiente cliente/servidor.

- A diferencia de otras herramientas que trabajan con bases de datos, Visual Basic permite la creación del archivo ejecutable (extensión .exe) de la aplicación que se ha desarrollado, lo que se vuelve un beneficio al requerir menor espacio en disco que cuando se carga en memoria el Visual Basic, agilizando de esta manera el sistema.

A continuación se muestra de manera breve el funcionamiento general de Visual Basic y las herramientas que ofrece en el diseño y presentación del sistema (interfaz gráfica del usuario).

Visual Basic ofrece una pantalla en la que incluye cinco objetos.

- *La ventana principal* formada por el menú que contiene gran cantidad de opciones para el diseño de aplicaciones.
- *La barra de herramientas* cuenta con los comandos o controles que pueden agregarse a una forma, para darle vida a la aplicación.
- *La ventana de propiedades* tiene las propiedades que caracterizan a una forma o a un comando, por ejemplo: colores, tipo y tamaño de letra, etc.
- *La ventana de proyecto* permite visualizar las formas o módulos que integran el proyecto, así como ver su código o su estructura.
- *La forma* es el espacio de diseño y tiene la siguiente estructura.

  Botones de minimización y maximización de pantallas.

  El título de la forma está situado en la parte superior central

  Área de diseño en la cual son agregados los objetos (cajas de texto, listas, etc.).

Dada las facilidades que ofrece la herramienta se generaron pantallas diseñadas con la idea de permitir a los usuarios introducir datos, consultarlos y generar reportes, bajo un mismo formato (Fig. 4.1), basándose en una programación flexible y adaptable a las necesidades

Dentro de lo que se considera el formato del sistema se tienen los siguientes elementos:

- Área de captura y resultados, que es donde se presentan los datos que requiere el sistema así como los resultados que podemos obtener.
- Barra de comentarios, que ofrece indicarle al usuario una referencia sobre los datos que se requieren.
- Opciones para poder cancelar, menús de ayuda y si así se desea, salir del sistema sin necesidad de regresar al menú principal donde también se presentan dichas opciones.

De esta manera se le permite al usuario el control total del sistema y por ende que este le sea mucho más accesible y sencillo de utilizar, no importando el nivel de conocimiento que el usuario tenga del sistema.

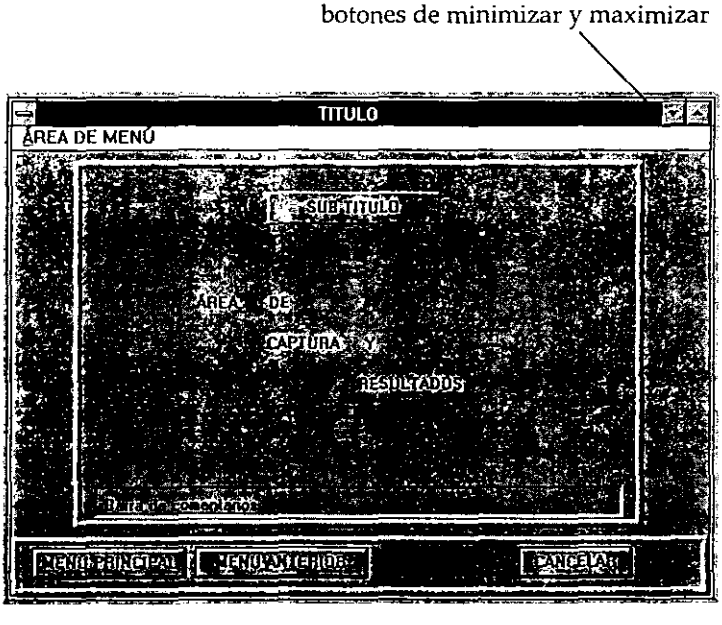


Fig. 4.1. Diseño general de las pantallas del sistema.

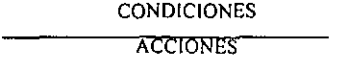
4.1. DIAGRAMA DE TRANSICIÓN DE ESTADOS.

Cuando se tiene una secuencia del comportamiento o flujo que siguen los datos, el diseño del sistema se vuelve relativamente sencillo. Una de las herramientas que nos ayuda a esto es el diagrama de transición de estados, que es la especificación secuencial del comportamiento en el tiempo del sistema. Sus elementos son:

a) *Estados.* Representan el estado en que se puede encontrar un sistema en un momento dado, su representación gráfica es la siguiente:

b) *Cambios de estado.* Se muestran como flechas y plasman los cambios de estado que se pueden dar y el orden en que sucede.

c) *Condiciones y acciones.* Muestran las condiciones que causan un cambio de estado y las acciones que el sistema efectúa cuando se cambia de estado, gráficamente es una línea horizontal que contiene la condición en la parte superior y la acción en la parte inferior.



Al construirse un diagrama de transición de estados, se recomienda empezar por identificar todos los estados y proseguir con la conexión de los mismos de manera iterativa.

Para el desarrollo de esta aplicación se manejaron diagramas de transición de estados, que permitieron seguir una secuencia en la construcción de la interfaz como se muestra a continuación.

Diagrama de transición de estados de Inventario

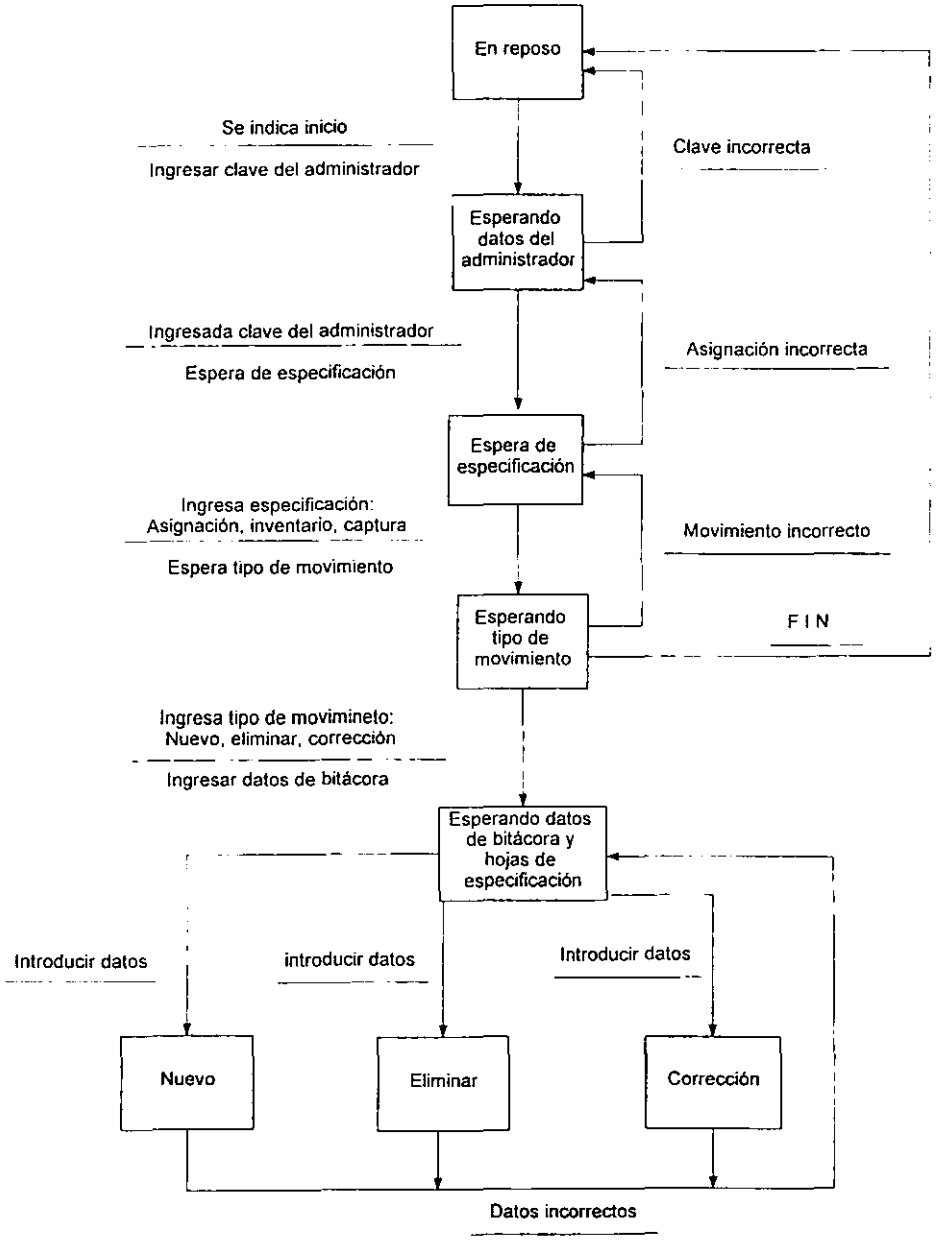


Diagrama de transición de estados de Asignaciones

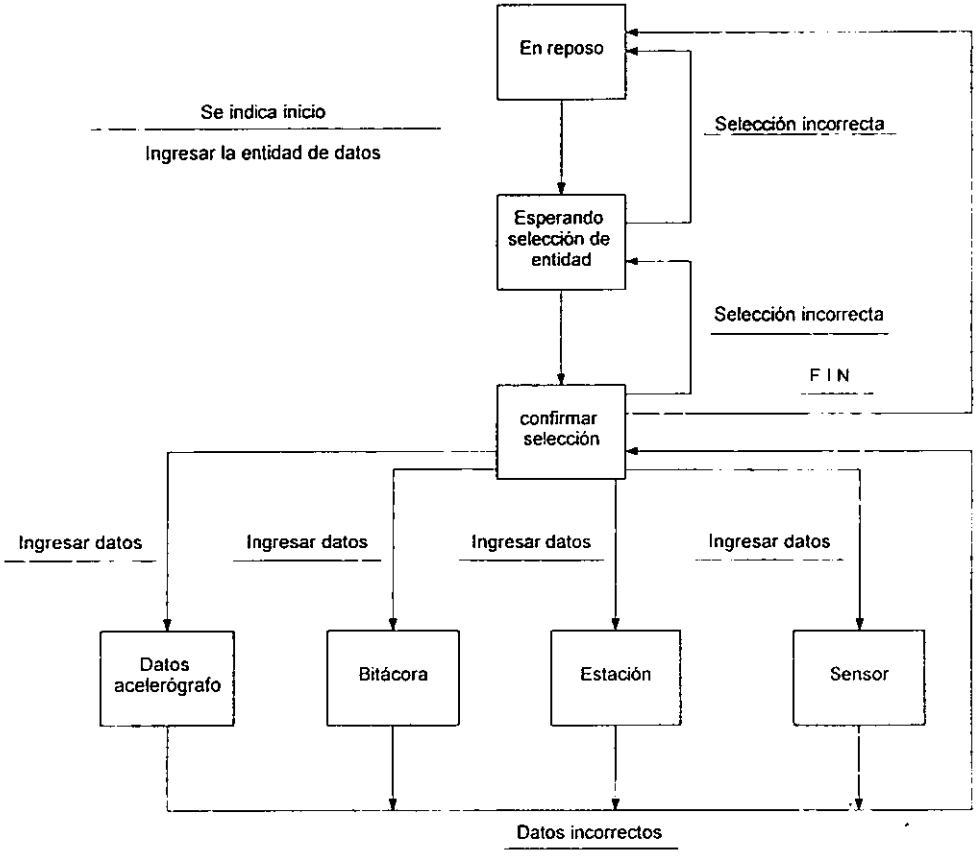


Diagrama de transición de estados de Reportes.

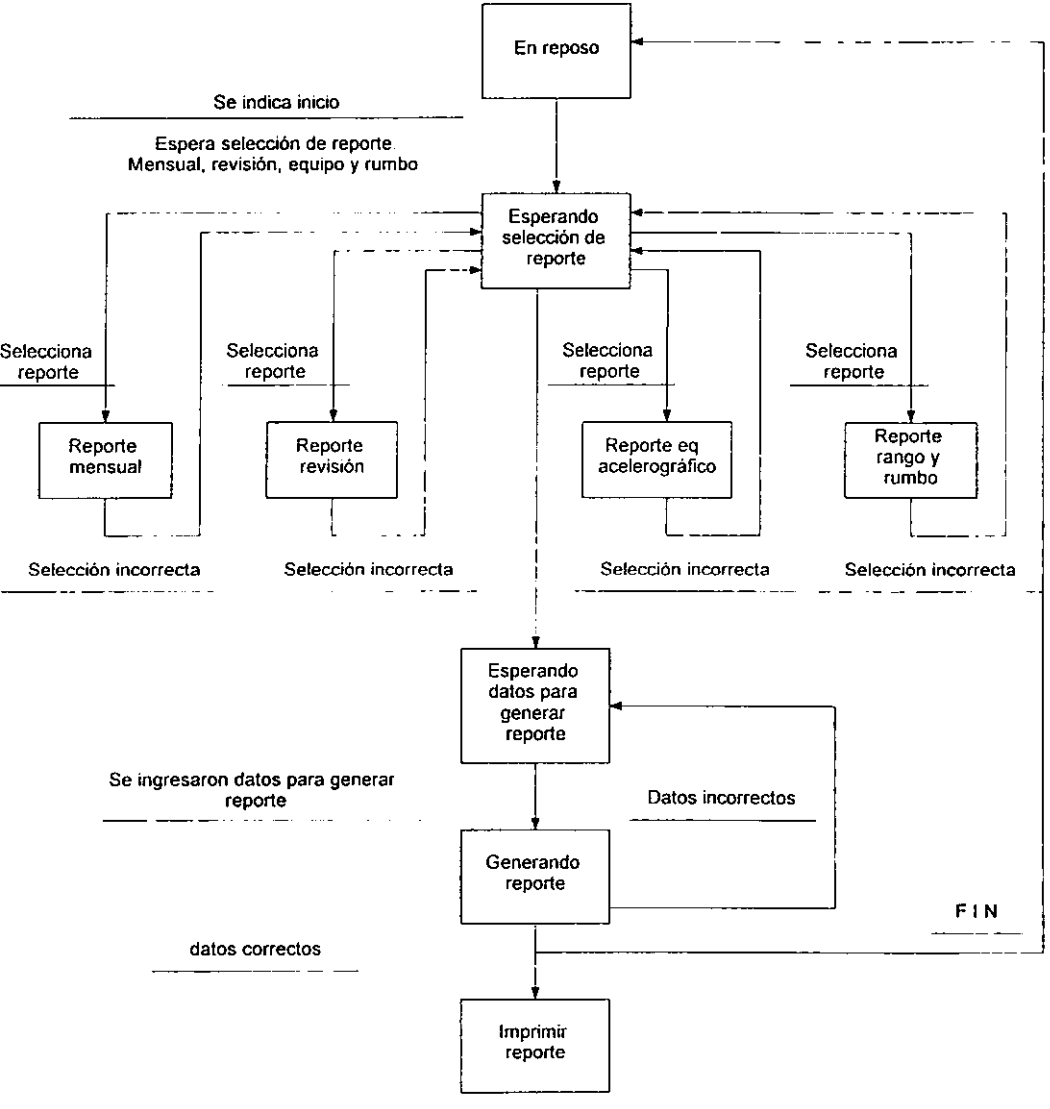


Diagrama de transición de estados de Consultas.

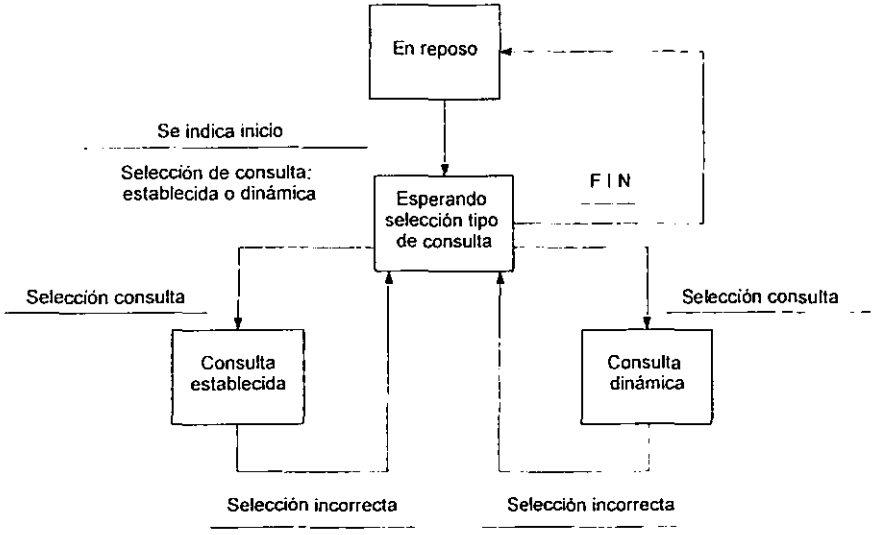


Diagrama de transición de estados de Consulta establecida

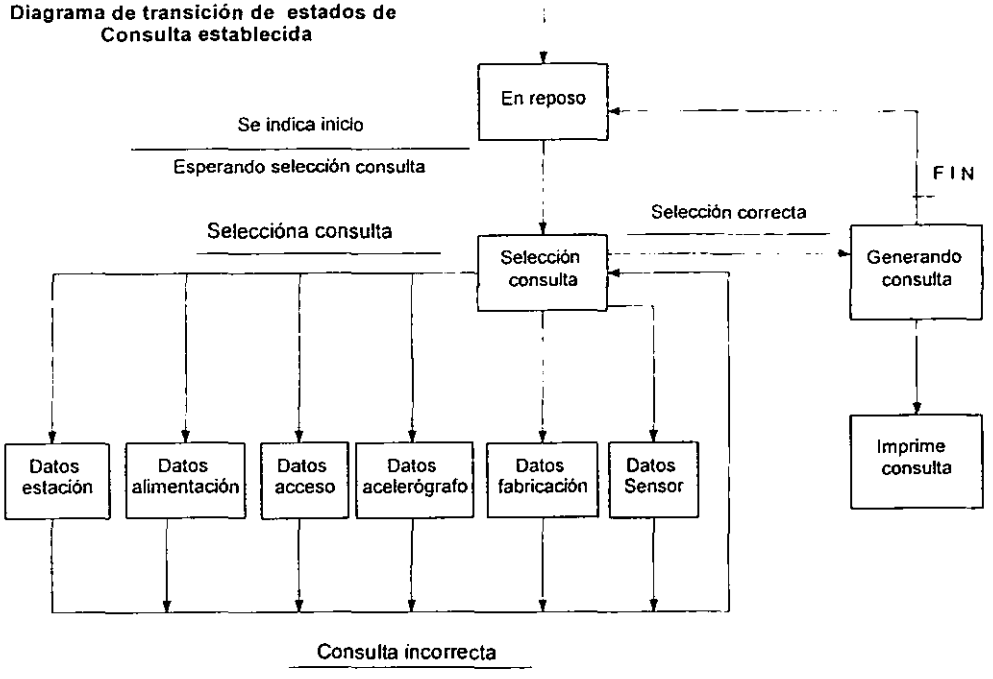
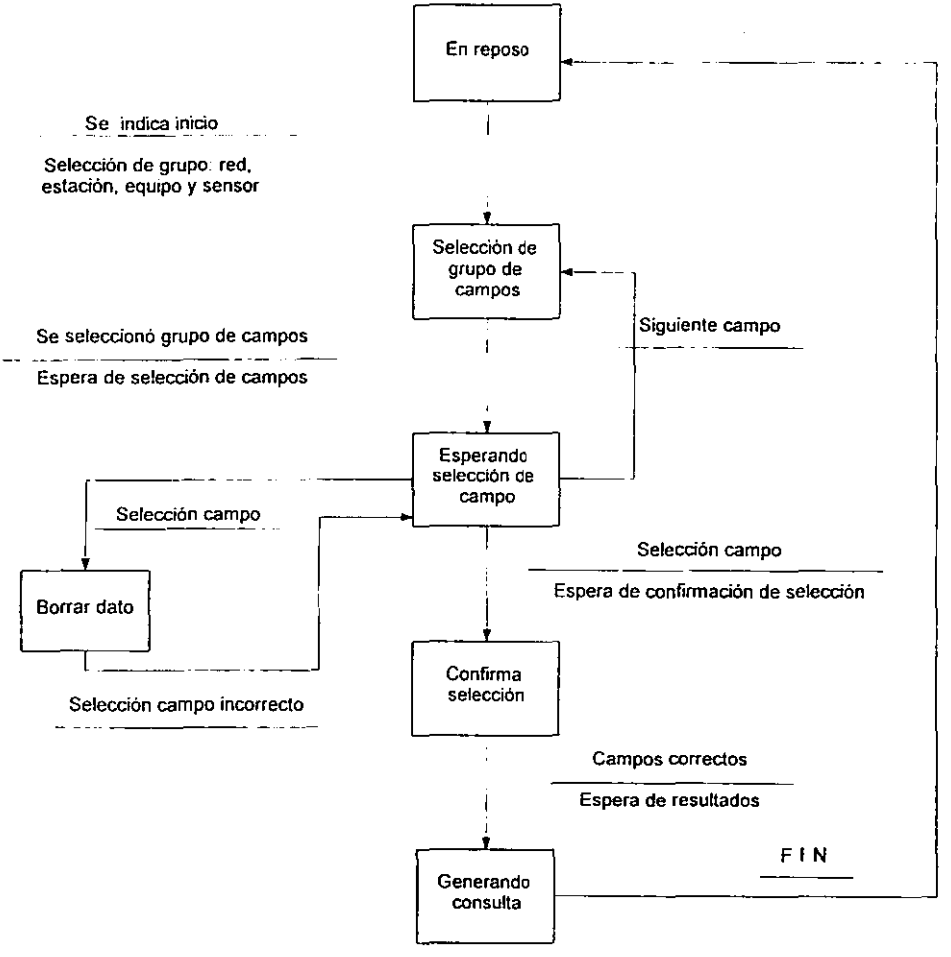


Diagrama de transición de estados de Consulta dinámica



Capítulo 5

Operación del Sistema.

CAPÍTULO 5

5. OPERACIÓN DEL SISTEMA.

El sistema esta compuesto de los siguientes módulos: inventario, actualizaciones, consultas, reportes y ayuda.

- Los primeros cuatro módulos permiten automatizar la búsqueda de información así como la consulta y reporte de datos.
- El módulo de ayuda permite aclarar las dudas que pudieran originarse en el momento de llenar los espacios en blanco, al obtener un reporte o al generar una consulta.

El sistema comienza con una pantalla de presentación (Fig. 5.1), seguida de un menú principal (Fig. 5.2) que muestra seis botones de opción. Los dos primeros, el de inventario y actualizaciones, cuentan con acceso restringido, mientras que los siguientes son para todo tipo de usuario.

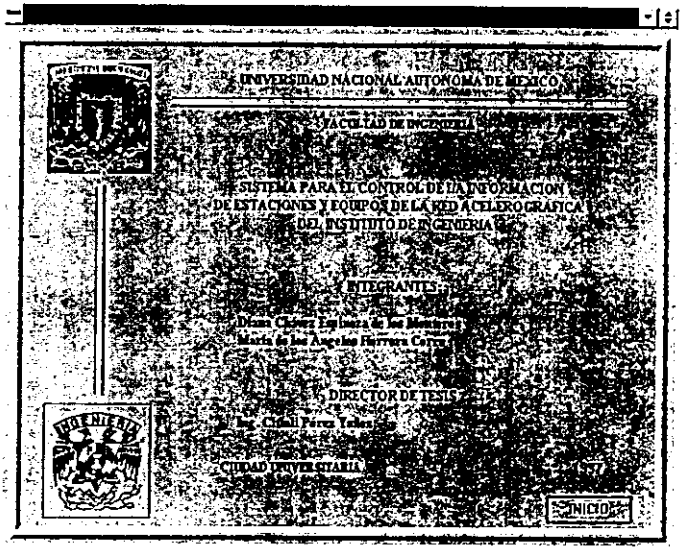


Fig. 5.1. Pantalla de presentación.

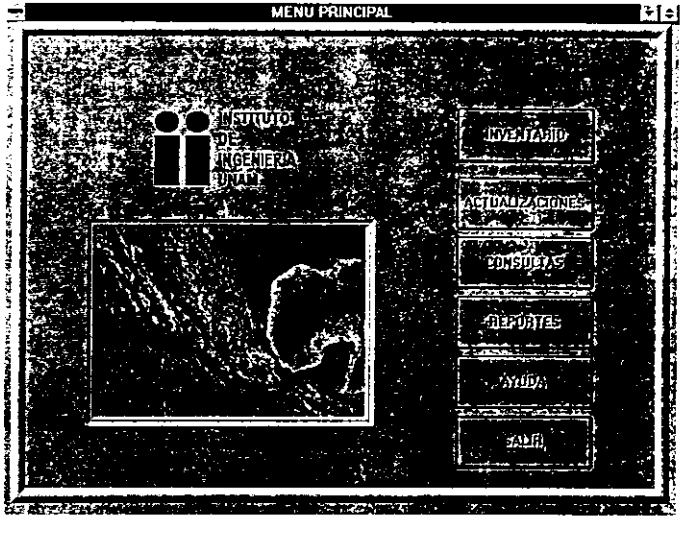


Fig. 5.2. Pantalla de menú principal.

5.1. INVENTARIO.

El módulo de inventario está integrado por 10 pantallas de presentación y captura de la información, las cuales se encuentran bajo el mismo formato.

Las tareas que se realizan dentro de este módulo son las siguientes:

- Dar de alta por primera vez un registro.
- Eliminar y corregir la información sobre el primer registro.

Las tablas que se utilizan son la de red, estación, tipo de lugar, alimentación de la estación, acceso a la estación, edo. de operación de la estación, equipo acelerógrafico, su propietario, tipo de equipo acelerógrafico, fabricantes, tipo de equipo sensor y sensores.

Por motivos de seguridad, los primeros registros de cualquiera de las tablas o relaciones las da de alta el dueño de la base de datos o el encargado de la operación del sistema. La captura se hace con una pantalla pequeña que registra un identificador y una palabra clave (Fig. 5.3), mismas que solo serán conocidas por él, con el fin de evitar que se realicen modificaciones de la información sin ningún control.

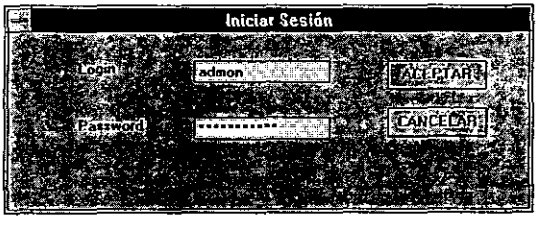


Fig. 5.3. Pantalla de control de acceso.

Dentro de este módulo también se maneja el eliminar o corregir la información sobre el primer registro en cualquiera de las tablas antes mencionadas.

Además se tiene un rubro en el que se asignan estaciones, equipos y sensores debido a que existen casos especiales como los siguientes:

- Cuando se construye una nueva estación y no se sabe a que red pertenecerá.
- Cuando llega un nuevo equipo acelerográfico y no se sabe en que estación quedará instalado.
- Los sensores que se tienen como repuesto (en caso de falla o mantenimiento de los que están en funcionamiento) o para agregarle uno nuevo a un equipo acelerográfico.

De esta manera, la opción de asignaciones (Fig. 5.4) nos permite indicar posteriormente en donde estará una estación, un equipo o un sensor, y además la operación se puede dar en forma iterativa.

En la base de datos, las tablas de red, tipo de lugar de la estación, estado de operación de la estación, propietario del equipo acelerográfico, fabricante, tipo de equipo acelerográfico y tipo de equipo sensor, contienen pocos datos y no crecen en registros por ello son capturados en un principio.

La figura 5.5 corresponde a inventario y en ella se muestran algunas de las opciones descritas anteriormente; la información se registra a través de pantallas de captura como en la figura 5.6.

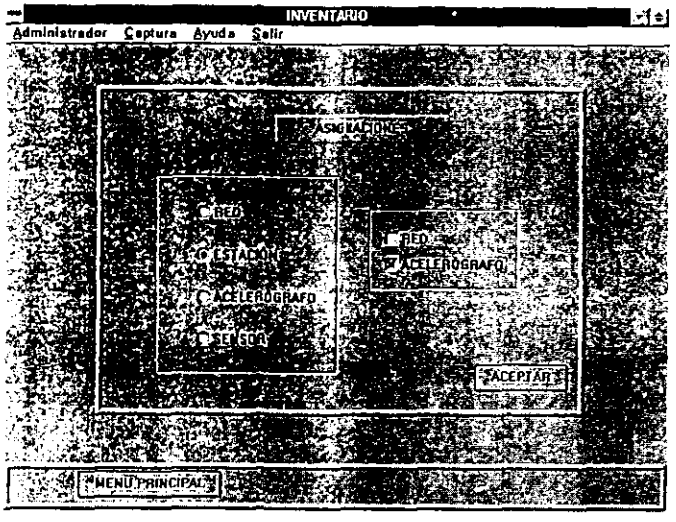


Fig. 5.4. Pantalla de inventario (opción asignaciones).

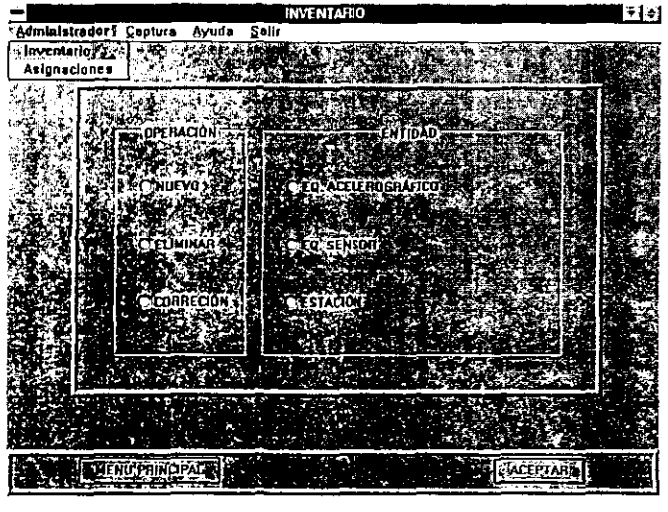


Fig. 5.5. Ejemplo de algunas opciones en la pantalla de inventario.

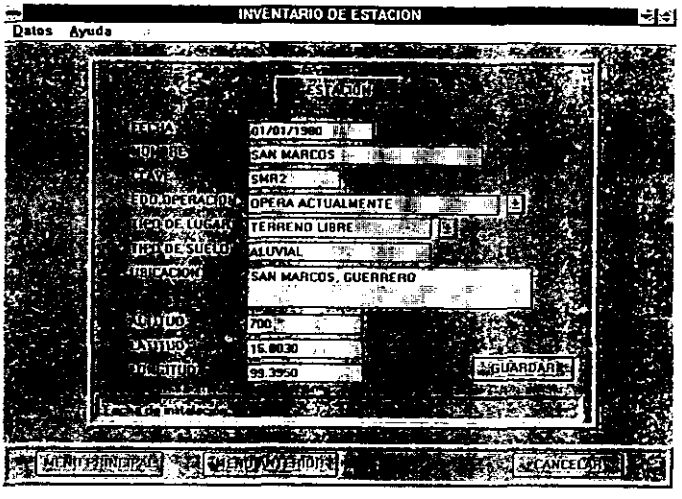


Fig. 5.6. Pantalla de captura de estación nueva.

5.2. ACTUALIZACIONES.

Dentro de éste módulo se realizan las tareas de actualización de la base de datos aproximadamente cada mes con los datos de las revisiones hechas a las estaciones.

El acceso a éste módulo estará restringido para dos o tres personas encargadas de realizar las actualizaciones, ya que el que muchas personas tengan acceso ocasionaría fallas en la integridad de la base, o incluso la pérdida de algunos datos.

Las actualizaciones se pueden realizar por acelerógrafo, estaciones, sensor (interno o externo), y por bitácoras.

La opción de bitácoras será la más utilizada porque ésta actualiza a las tablas de estación, bitácora de estación, umbral, bitácora de lectura, equipo acelerográfico, edo. de operación del equipo acelerográfico, ganancia, sensores, estación-equipo-acelerográfico, equipo acelerográfico-eq. sensor y equipo sensor. A su vez la opción esta dividida en:

* Actualización de acelerógrafo.
* Actualización de bitácora de estación.
* Actualización de lectura.

La actualización de bitácora de estación se realiza en dos partes, una con datos de entrada y otra con datos de salida, debido a la forma en que se llenan las bitácoras en el lugar de la revisión y de esa manera facilitarle al usuario su captura.

En la figura 5.7 se observa la pantalla de actualizaciones con las opciones que se especificaron anteriormente; el panel que contiene las opciones de aceleración, estación y lectura, y el panel con opciones de interno y externo que se activan al escoger la opción de bitácoras o sensor respectivamente, cualquiera de las combinaciones anteriores lleva a una pantalla muy parecida a la que se muestra en la figura 5.8 que solicita los datos de la estación.

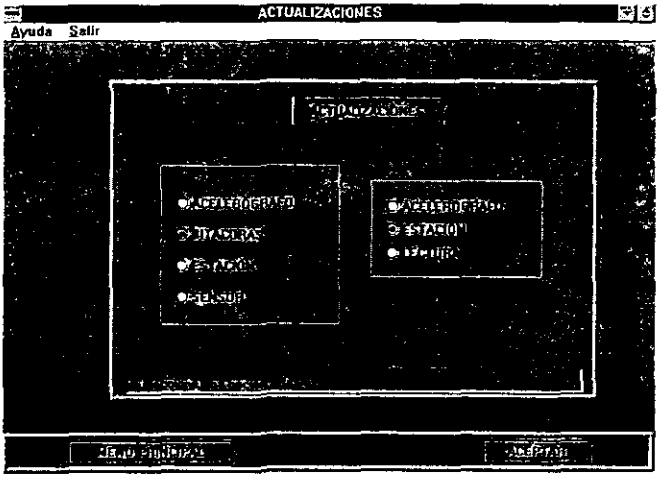


Fig. 5.7. Pantalla de actualizaciones.

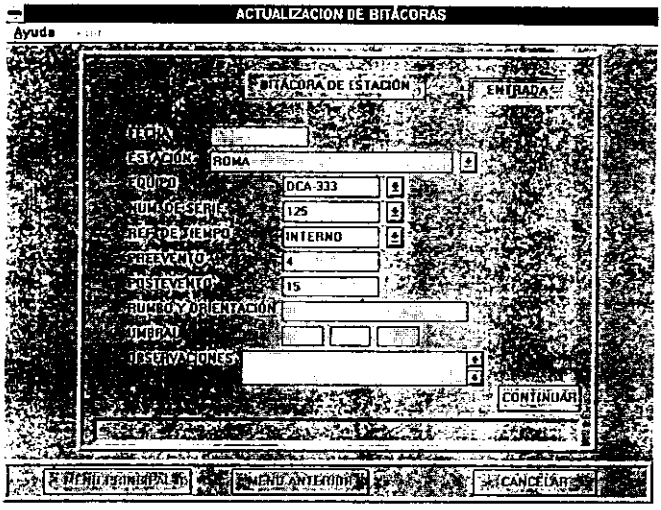


Fig. 5.8. Pantalla de captura de datos de estación.

Después de guardar los datos de la actualización, la operación se puede repetir de manera iterativa con el siguiente cuadro de texto con pregunta (Fig. 5.9), la mayoría de las formas trae un botón de cancelar, con la que se puede interrumpir y abandonar la operación que se esta realizando (Fig. 5.10).

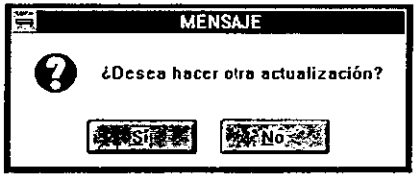


Fig. 5.9. Pantalla de mensaje.

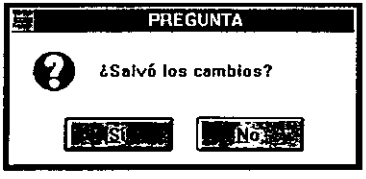


Fig. 5.10. Pantalla de mensaje.

5.3. CONSULTAS.

La entrada a este módulo no está restringida, por lo tanto podrá ser utilizada por todo el personal de la coordinación de instrumentación sísmica, permitiendo así la consulta de los datos que cada usuario necesite para realizar su trabajo.

El tiempo y la experiencia en el trabajo de la coordinación, han ayudado a conocer las necesidades que cada usuario tiene en sus consultas, por ello se sabe que la mayoría de veces que éste pide un reporte lleva los mismos datos, teniendo a la fecha como única variable. Aunque ocasionalmente se requiere conocer mayor información, es vital contemplar otros tipos de consulta:

- ✓ Las consultas establecidas
- ✓ Las consultas dinámicas.

CONSULTAS ESTABLECIDAS.

Este tipo de consultas ya han sido definidas y cada una cuenta con datos agrupados en base a las necesidades más elementales de los usuarios.

A continuación se observa en la figura 5.11, cómo el usuario puede elegir alguna de las consultas establecidas con que se cuenta, desplegándose los datos en forma de lista.

De igual manera, dentro de algunas de las consultas establecidas, el usuario puede seleccionar la red de la que se desea la información con el propósito de aprovechar al máximo la consulta en su propio beneficio.

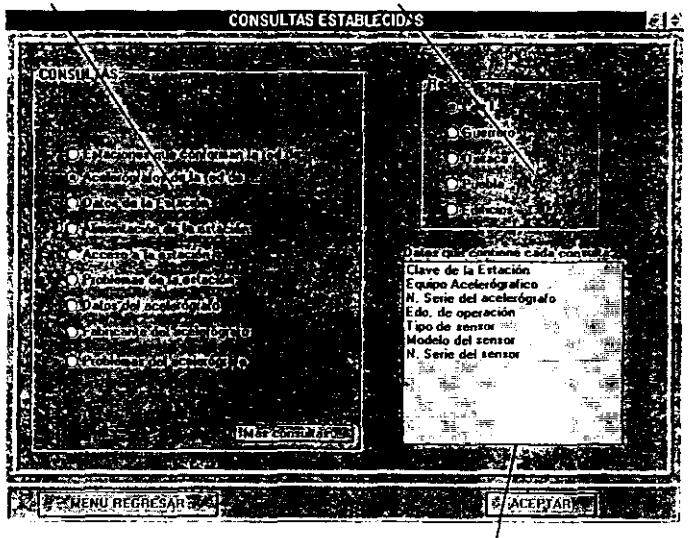


Fig. 5.11. Pantalla de consultas establecidas.

La tabla 4 muestra los datos que contienen algunas de las consultas establecidas:

Nombre de la consulta.	Datos que contiene la consulta.
Acelerografos de la red de:	Clave de la Estación
I. de I., Guerrero	Equipo y N. de serie del acelerógrafo
Oaxaca, Puebla	Edo. de operación, Tipo de sensor
Edificios	Modelo y N. de serie del sensor
Datos de la estación.	Clave de la Estación
	Fecha de Instalación
	Tipo de suelo, Tipo de lugar
	Ubicación, Altitud
	Latitud, Longitud
Datos del Acelerógrafo.	Clave de la Estación
	Modelo y N. de serie del acelerógrafo
	Ganancia por canal
	Fecha de revisión
	Velocidad de muestreo
	Referencia de tiempo
	Preevento y Postevento
Fabricante del acelerógrafo.	Modelo y N. de serie del acelerógrafo
	Fabricante, Tipo de acelerógrafo, País
Datos del sensor.	Modelo y N. de serie del eq. sensor
	Fecha de instalación
	Modelo y N. de serie del sensor
	Numero de canal y Rango
	Frecuencia, Amortiguamiento
	Personal de revisión y observaciones

Tabla 4. Datos mostrados para algunas consultas establecidas.

La presentación de los resultados de estas consultas se realiza con "Crystal Report", una de las principales herramientas de Visual Basic para el diseño de reportes. En ella se especifican previamente los datos que contiene cada consulta (obtenidos directamente de las tablas de la base de datos), así como los detalles que implica el diseño de cada reporte (tipo y tamaño de letra, gráficos, orden, tamaño y orientación del papel, etc.) (Fig. 5.12).

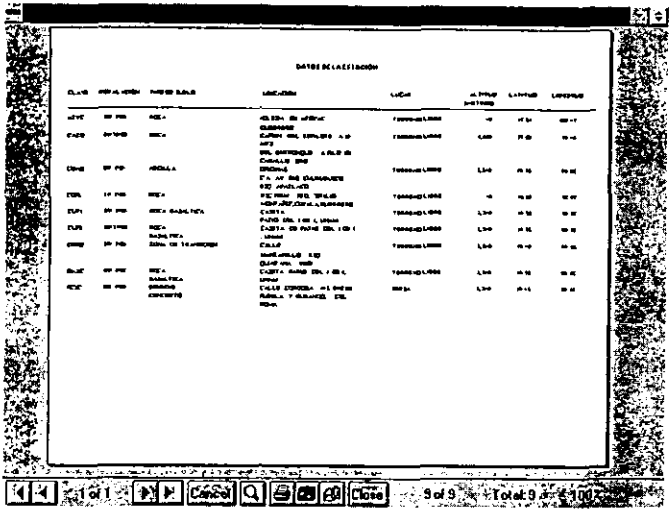


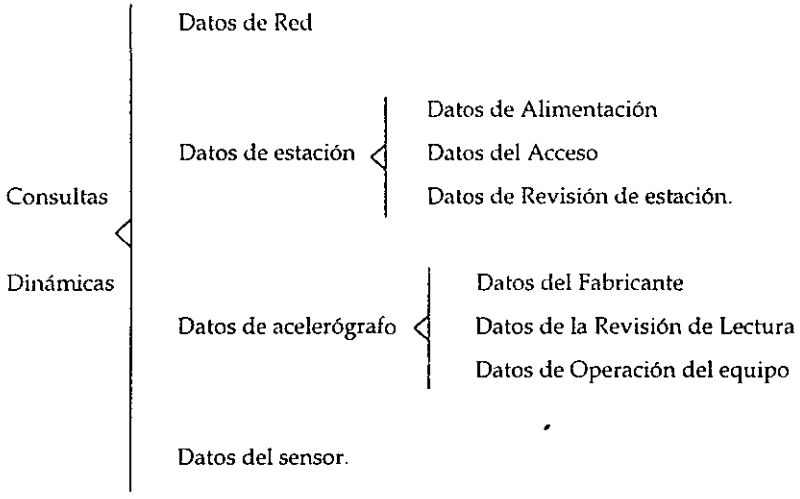
Fig. 5.12. Resultado de una consulta establecida.

La presentación final de cada consulta tiene automáticamente integradas las siguientes opciones:

- Ver los datos por páginas.
- Ver la página previa a su impresión.
- Mandar a imprimir por páginas o todo.

CONSULTAS DINÁMICAS.

El otro tipo de consultas es el dinámico, en el cual el usuario puede seleccionar los datos que requiera. Cada consulta tendrá como mínimo dos columnas y un máximo de 15 columnas o campos de interés. La información la puede consultar de acuerdo a la clasificación de datos ya establecida que es la siguiente:



Como en una consulta dinámica la cantidad de datos y combinaciones que existen son especificadas por el usuario, la presentación se realiza en una tabla integrada por filas y columnas que aumentan o disminuyen conforme se requiera. La selección de los datos se realiza de una lista de datos agrupados (Fig. 5.13), en caso de no haber datos para la consulta aparecerá un mensaje (Fig. 5.14).

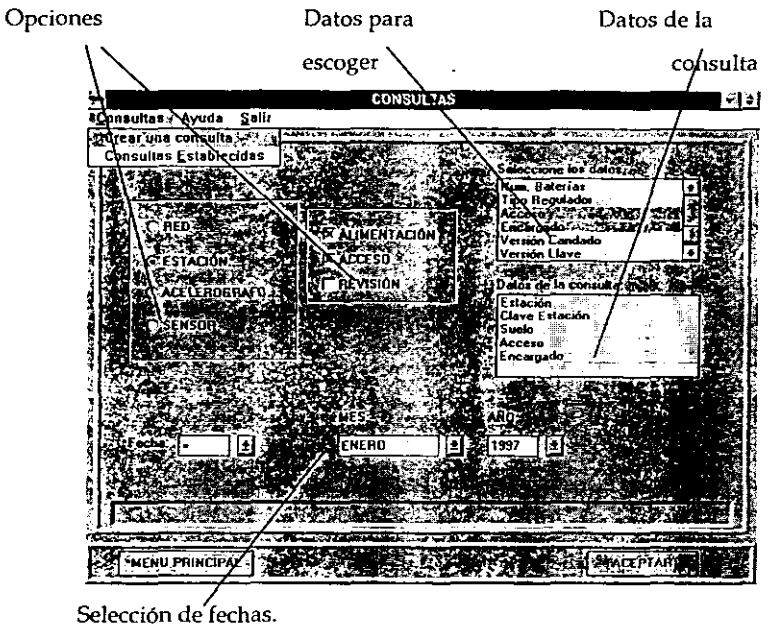


Fig. 5.13. Pantalla para crear una consulta.

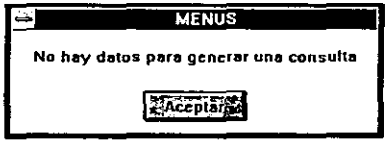


Fig. 5.14. Pantalla de mensaje.

El formato de presentación de los resultados de una consulta dinámica se observa en la figura 5.15.

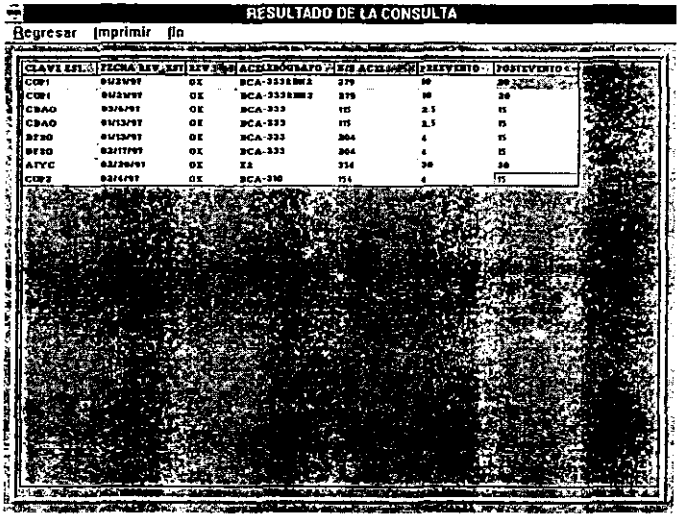


Fig. 5.15. Pantalla del resultado de una consulta.

5.4. REPORTES.

Al igual que el módulo de consultas, el módulo de reportes es para uso de todo el personal, en el que se contemplan los reportes solicitados con mayor frecuencia por los ingenieros (tabla 5). Estos contienen datos generales del estado de las estaciones y equipos, que les ayudarán y facilitarán sus revisiones periódicas a las estaciones en su campo de trabajo (Fig. 5.16). La figura 5.17 muestra la pantalla de resultados de un reporte.

Nombre del reporte	Datos que contiene el reporte
Reporte mensual	Fecha, red, estación, equipo y N/S del acelerógrafo, observaciones del estado del equipo y de la lectura
Reporte de revisión	Fecha, red, estación, equipo y n/s del acelerógrafo, observaciones del estado de la estación y del equipo, personal de revisión.
Reporte de cambio de equipo	Red, estación, modelo y n/s del acelerógrafo, fecha de instalación y retiro del equipo.
Reporte por rumbo y orientación	Red, estación, modelo y n/s del acelerógrafo, fechas de instalación y retiro del equipo, rumbo y orientación, modelo y n/s del equipo sensor, rango.

Tabla 5. Datos que contienen los reportes.

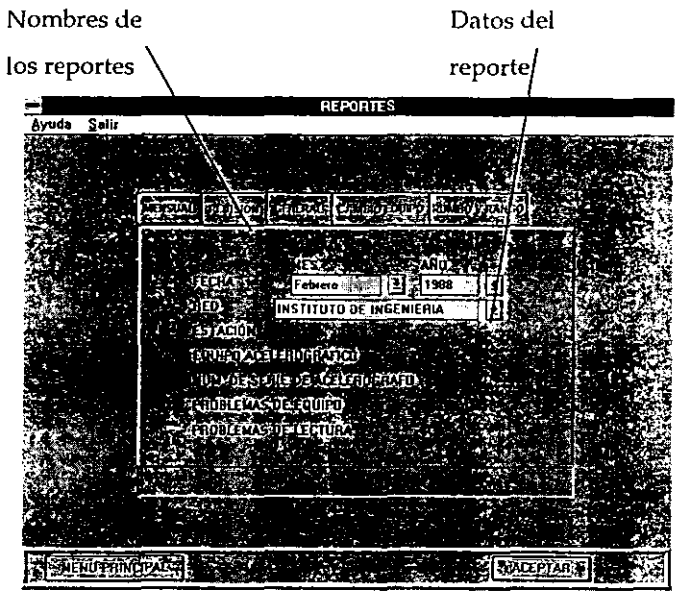


Fig. 5.16. Pantalla de reportes.

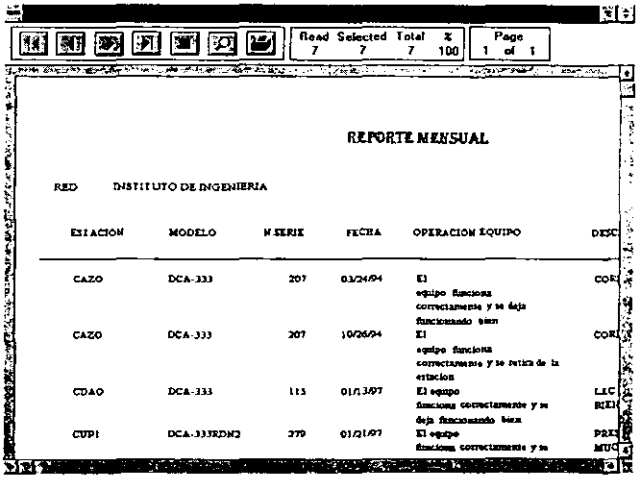


Fig. 5.17. Pantalla de resultados de un reporte.

5.5. AYUDA.

El diseño del sistema esta contemplado de una forma muy sencilla y fácil de comprender, pero aun así, se pensó en la posibilidad de personal recién integrado a la coordinación que no esté tan familiarizado con el funcionamiento del sistema por lo que se le pueda llegar a dificultar el uso del mismo o no comprenda la manera de consultar o llenar los espacios. Pensando en esto se realizó la ayuda de forma independiente que permita una mejor comprensión del sistema.

La ayuda presenta un botón por cada letra del alfabeto, que a su vez activa una lista que muestra los temas y al seleccionarlo despliega el texto con la explicación de interés (Fig. 5.18). El sistema muestra también información referente al nombre, versión, lugar de desarrollo, etc. (Fig. 5.19).

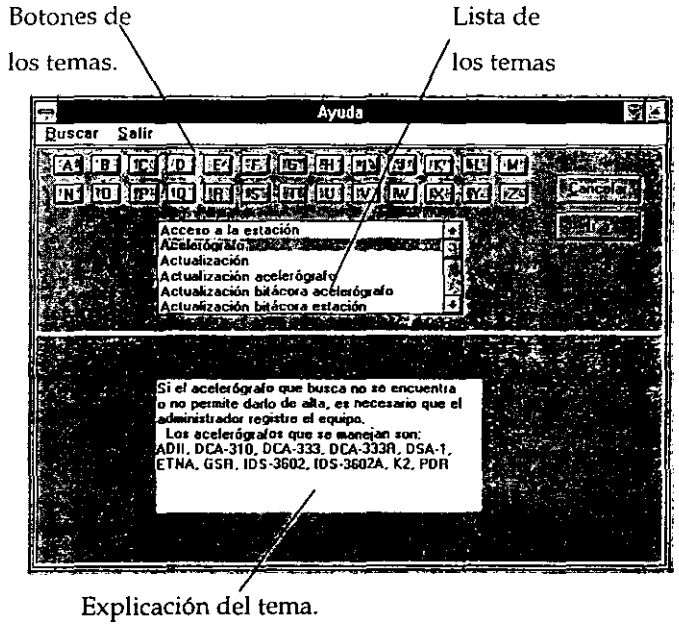


Fig. 5.18. Pantalla de ayuda.

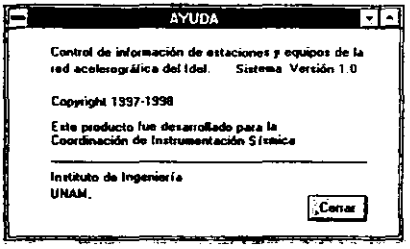


Fig. 5.19. Pantalla de información del sistema.

5.6 SALIR.

El módulo de salir permite realizar las siguientes acciones:

- Cerrar la base de datos.
- Terminar la conexión al servidor.
- Liberar la memoria de la maquina que utilizan las formas, variables y controles.
- Terminar la aplicación.

El menú principal y la mayoría de las pantallas en el sistema ofrecen la opción de salir, misma que se confirma con un aviso (Fig. 5.20):

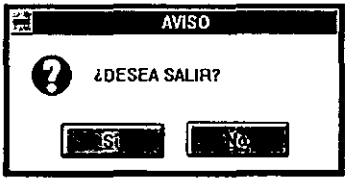


Fig. 5.20. Pantalla de mensaje.

Capítulo 6

Conclusiones.

CAPÍTULO 6

6. CONCLUSIONES

Al término de este trabajo podemos darnos cuenta que estando a principios del nuevo milenio la información es uno de los elementos más importantes, ya que la calidad al obtener la información influye notablemente en el desarrollo de las organizaciones.

La Coordinación de Instrumentación Sísmica del I. de I. requiere del manejo eficiente de la información, por ello se integra el historial de estaciones y equipos de la red acelerográfica en una base de datos.

Este trabajo permite aplicar los conocimientos adquiridos sobre el desarrollo de un sistema de base de datos con un enfoque relacional, permitiendo de ésta manera, cubrir las necesidades actuales de integridad y seguridad de la bases de datos y ajustarse a los requerimientos en el futuro.

La filosofía que se tomó en cuanto a la arquitectura seleccionada fue que la aplicación debería correr independientemente en diferentes máquinas conectadas en una red de comunicación, es decir, un sistema multiusuario para compartir datos. De aquí la selección de un manejador de bases de datos (DBMS) montado en arquitectura cliente/servidor.

Sybase como el DBMS seleccionado, representa una limitante para el usuario al contarse únicamente con la herramienta "isql" (utilería rudimentaria para proceso en línea), ya que seria necesario que se tuvieran conocimientos de bases de datos y detalles de la implementación. Por esto se decidió diseñar una interfaz gráfica que permita capturar y manejar la información de una manera mucho más sencilla, en lugar de realizarlo con comandos SQL.

El sistema que se ha desarrollado se encarga de realizar la organización y almacenamiento de los datos, generando información a partir de los reportes y consultas establecidas, y dadas sus características en forma de lista permite analizar rápidamente la información, para el trabajo de mantenimiento de las redes acelerográficas.

El tiempo es un factor importante en todo sistema, por lo que al realizar la mayoría de los procedimientos de actualización de la base en el servidor se logra reducir el tiempo de espera y respuesta en el cliente al mínimo.

Si bien es cierto que el sistema ofrece muchas ventajas, también es cierto que presenta algunos inconvenientes, como los problemas que pudieran ocasionarse en la red de comunicación, el depender de personal eficiente que mantenga al servidor y bases de datos en buen estado, y que los respaldos se utilicen adecuadamente en el servidor.

Podemos resumir que los objetivos del sistema se cumplieron de manera amplia, creando así un sistema que tuviera el control de los parámetros de las estaciones y equipos, y que cada usuario pudiera tener la información necesaria desde su lugar de trabajo.

Además de sugerir algunos cambios a futuro, que de ninguna manera afectarán el rendimiento del sistema, estos podrían ser:

- Utilizar el sistema para alimentar a los programas que se utilizan para el procesamiento de registros sísmicos, generando los archivos con el formato que estos requieren para su funcionamiento y con la información que se tiene almacenada en la base de datos.
- Crear un módulo automático de respaldo.
- La posibilidad de que otras Instituciones puedan utilizar los datos de la base.

- El sistema puede crecer cubriendo nuevas necesidades como es la creación de nuevas tablas, procedimientos, campos, módulos, etc. sin que estas afecten al diseño planteado en esta primera versión.

La base de datos en conjunto con la interfaz permitirán ser migrados en caso necesario a versiones más recientes con un mínimo de cambios, sin que estos causen problemas.

Capítulo 7

Bibliografía

BIBLIOGRAFÍA

* Ingeniería de software un enfoque practico
  Presman S. Roger
  editorial Mc Graw Hill, 1993

* Introducción a los sistemas de bases de datos
  C.J.Date
  Editorial Addison-Wesley iberoamericana 1986

* El modelo Entidad-Relación
  Richard Barker
  Editorial Addison-Wesley.

* LAN Applications-Client/Server Databases
  J Krochmal, L Morris,
  New Riders Publishing, 1993.

* Organización de las Bases de Datos
  James Martín
  Prentice Hall
  México 1988

* Técnica de Bases de Datos
  Shakuntala Atre
  Editorial Trillas, 1988

* Base Nacional de Datos de Sismos Fuertes. Catálogo de Estaciones acelerográficas 1960 - 1992
Idel, UNAM (R Quaas, S. Medina), FICA (JA Otero, H Aguilar), CIRES (JM Espinosa), CFE (M González),
Sociedad Mexicana de Ingeniería Sísmica, A.C., 1993.

* "Documentación de la Base Nacional de Sismos Fuertes",
Salvador Medina,
marzo 1996.

* Análisis y diseño de sistemas
Kenneth E Kendall
Julve E Kendall
Prentice Hall Hispanoamericana.

Apéndice

Manual del Usuario.

A-1. Instalación del Sistema para el control de información de estaciones y equipos de la red acelerográfica del I. de I.

Contenido:

- Requerimientos del sistema
- Instalación del sistema en Microsoft Windows 3.11 para trabajo en grupo y Microsoft Windows 95
- Elementos instados

REQUERIMIENTOS DEL SISTEMA

Antes de la instalación del sistema es necesario que se cumplan como mínimo los siguientes requisitos de:

Software:

- ✓ Microsoft Windows 3.11 para trabajo en grupo o Microsoft Windows 95.
- ✓ ODBC

Hardware:

- ✓ Procesador 486 o superior
- ✓ 4 Mb de memoria en RAM como mínimo, se recomienda 8 MB o más.
- ✓ Espacio disponible en disco duro de 10 MB como mínimo.
- ✓ Unidad de disco de 3.5"
- ✓ Monitor VGA o superior
- ✓ Ratón
- ✓ Conexión a red.

INSTALACIÓN DEL SISTEMA

Para instalar asegurarse de cubrir los requisitos de software y hardware, y de contar con el disco de instalación.

Las tareas de instalación son:

1. Crear un directorio llamado: SISBD en el disco C.

2. Copiar y descomprimir todos los archivos en el directorio especificado.

Para Windows 3.11:

3. Generar un grupo de trabajo y su icono de acceso directo al sistema, dentro del administrador de programas.

Para Windows 95:

4. En mi PC o en el explorador de Windows, seleccionar la opción de "Nuevo" en el menú de archivo y hacer click en carpeta para el nombre del directorio.

Procedimiento para instalación.

1. Inserte el disco del sistema dentro de la unidad de disco de 3.5"

2. Dentro del grupo principal, abrir el administrador de archivos.

3. Es muy importante crear dentro de la unidad lógica C un nuevo directorio con el nombre de: SISBD

4. Según sea el caso, seleccionar la unidad de disco "a:" o "b:", a continuación seleccionar todos los archivos del disco de 3.5" y copiarlos al directorio seleccionado.

5. En el administrador de programas se debe crear un grupo de programa con su icono de acceso directo al sistema.

ELEMENTOS INSTALADOS

Cuando se instala el sistema se crea el subdirectorio "C:\SISBD" dentro del cual se encuentra almacenado el programa ejecutable "SISBD.EXE".
Asimismo los archivos de extensión DLL (Librerías de conexión dinámica) y los archivos con extensión VBX (controles especiales de Visual Basic) son instalados en C:\windows\system

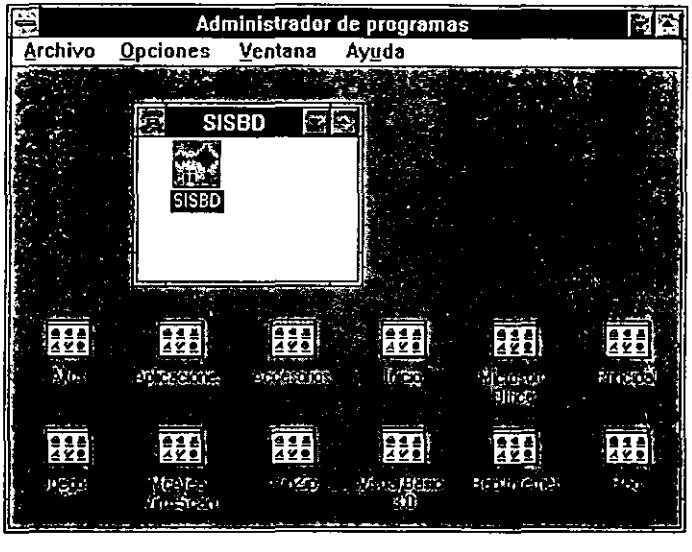
A-2. Trabajando con el sistema para el control de estaciones y equipos de la red acelerográfica del I. de I.

Contenido:

- Iniciando el sistema
- Pantallas de inicio y menú principal.
- Elementos básicos que aparecen en las pantallas del sistema.
- Captura e impresión de la información.

INICIANDO EL SISTEMA.

Cuando el sistema ha sido instalado, en la ventana del Administrador de programas de Windows se encuentra un grupo de programas llamado "SISBD" que incluye el icono del sistema como se ve en la siguiente pantalla:

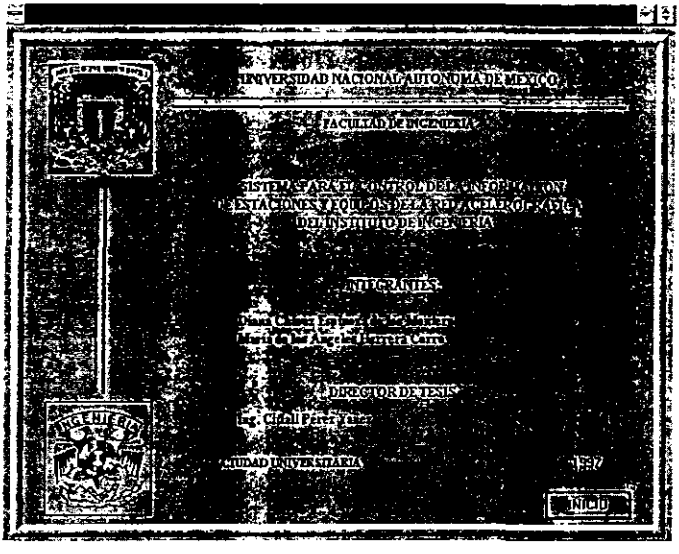


Para inicializar el programa se hace doble click sobre el icono del sistema del grupo de programas (Para Windows 95 se crea un acceso directo que muestre el icono de SISBD y se procede de manera similar).

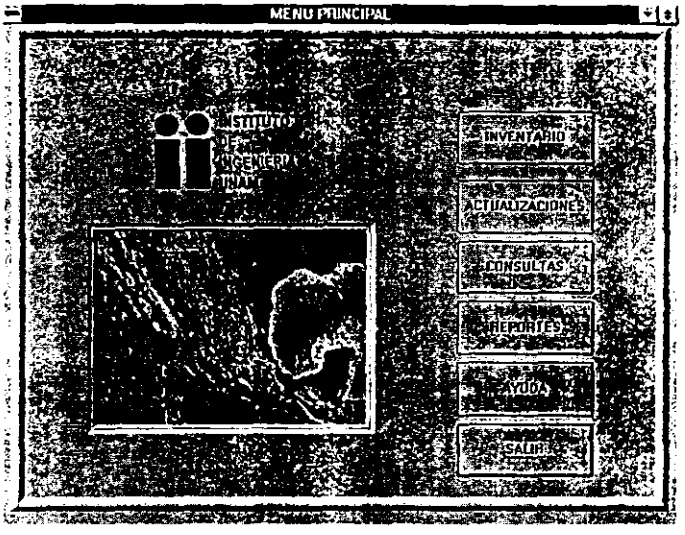
PANTALLAS DE INICIO Y MENÚ PRINCIPAL

El sistema permite para mayor comodidad que el usuario pueda utilizar el teclado (tabuladores, flechas de cursor) y el ratón.

La pantalla que aparecerá cuando se inicia una sesión del sistema será la de presentación, donde para continuar se debe presionar el botón de comando de Inicio, como se ve en la siguiente figura.



Si al oprimir inicio aparece un mensaje de error indica que el servidor tienen problemas, en este caso terminar la aplicación y comunicárselo al encargado del sistema o al administrador del servidor. Si no se generaron errores aparecerá la siguiente pantalla.

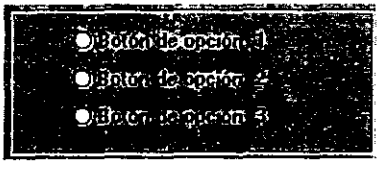


Como se observa, el menú principal tiene seis opciones que permiten trabajar con el sistema, de las cuales se puede seleccionar solo una a la vez.

ELEMENTOS BÁSICOS QUE APARECEN EN LAS PANTALLAS DEL SISTEMA.

- Botones de opción

Los botones sirven para seleccionar una opción de las que se encuentran agrupadas en un marco como el que se muestra a continuación



Para saber cual de los botones es seleccionado se establece lo siguiente:

◉ Botón activado ○ Botón desactivado

- Cuadros de opción múltiple.

Estos cuadros al igual que los botones de opción también se encuentran agrupados en un marco, permitiendo tener uno o más cuadros de opción activados al mismo tiempo.

Para saber cual de los cuadros de opción ha sido seleccionado se establece lo siguiente:

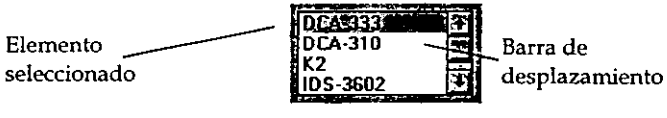
☒ Cuadro activado ☐ Cuadro desactivado

- Cuadros de lista

El cuadro tiene una lista de elementos de los cuales se puede elegir uno a la vez. Los elementos se pueden ver presionando el botón de desplazamiento.
El elemento aparece como en la ilustración.

- Listas

Este tipo de listas despliega los elementos que contiene, pudiéndose seleccionar mas de un elemento al mismo tiempo. Cuando el número de elementos que la integran es mayor al tamaño de la caja, aparece una barra vertical de desplazamiento que permite ver los datos faltantes como se muestra a continuación:



- Caja de texto

El lugar designado para recibir la información es el cuadro de texto, mismo que se muestra a continuación.

- Botones de comando

El botón de comando es utilizado para varias aplicaciones, entre ellas el iniciar, cancelar o terminar un procedimiento determinado. Cuando un botón esta activado sus letras aparecen en negro y una vez que se desactiva, el color cambia a gris.
Para trabajar con este comando, debe situarse sobre él y hacer click con el ratón o presionar: Enter

Las pantallas del sistema tienen los siguientes botones:

El botón "ACEPTAR" es seleccionado si se desea obtener el resultado de alguna operación y al término de ésta el resultado será desplegado en otra pantalla.

Por motivos de seguridad, el sistema no permitirá continuar cuando las variables necesarias no estén completas, en este caso se indicará con algún mensaje que señale la falta de datos.

El siguiente marco contiene tres botones básicos que aparecen en la parte inferior de las pantallas del sistema.

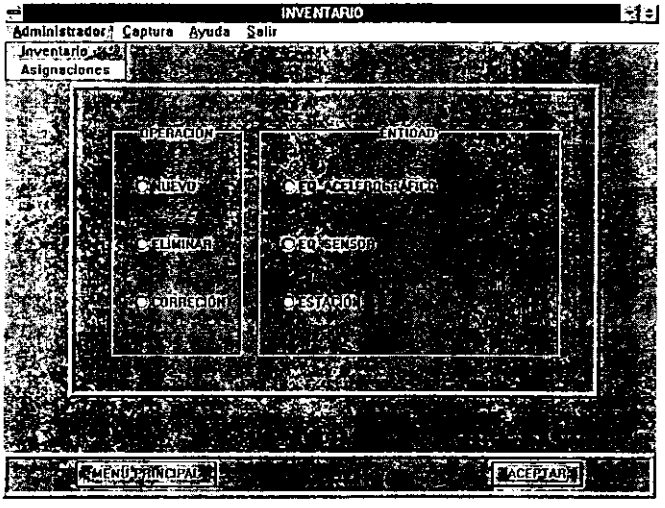
Los primeros dos botones, como su nombre lo indica, permiten regresar al menú principal o al menú anterior a la pantalla en que se encuentre.

El botón de cancelar sirve para interrumpir cualquier operación que no se desee continuar, activando como consecuencia los dos primeros.

- **Opciones del sistema.**

⇒ Inventario:

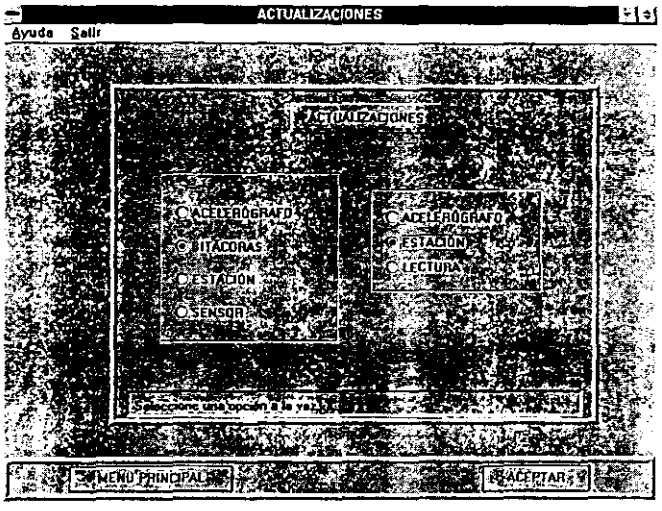
Cuando se selecciona esta opción en el menú principal se despliega la siguiente pantalla:



Esta nos lleva a la captura de información para la base de datos y a realizar nuevas asignaciones.

⇒ Actualizaciones

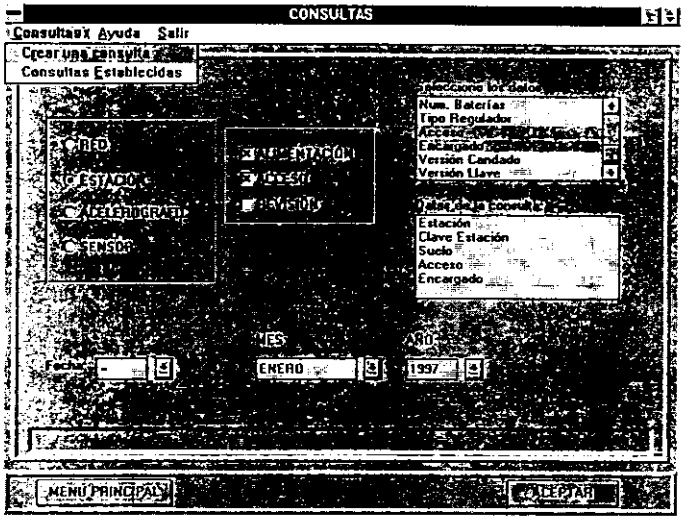
Cuando se selecciona esta opción en el menú principal se despliega la siguiente pantalla:



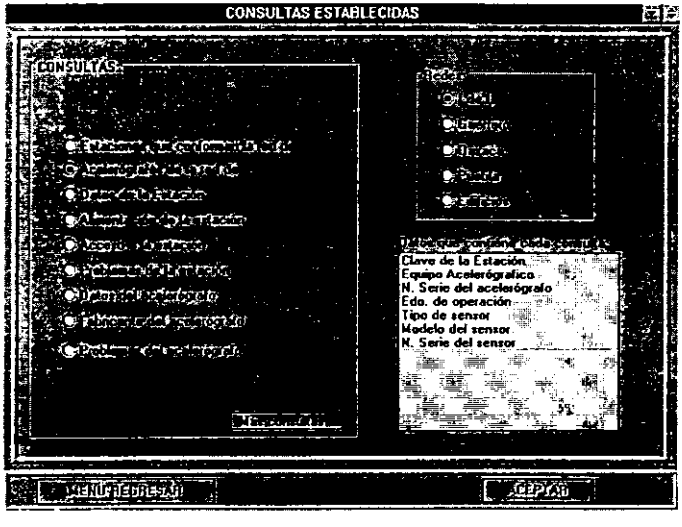
En ésta encontramos las opciones por las cuales se pueden actualizar los datos de las revisiones efectuadas a las estaciones, cualquiera de las combinaciones anteriores, nos llevará a otra pantalla que recibe la información.

⇒ Consultas:

Cuando se selecciona esta opción en el menú principal se despliega la siguiente pantalla:



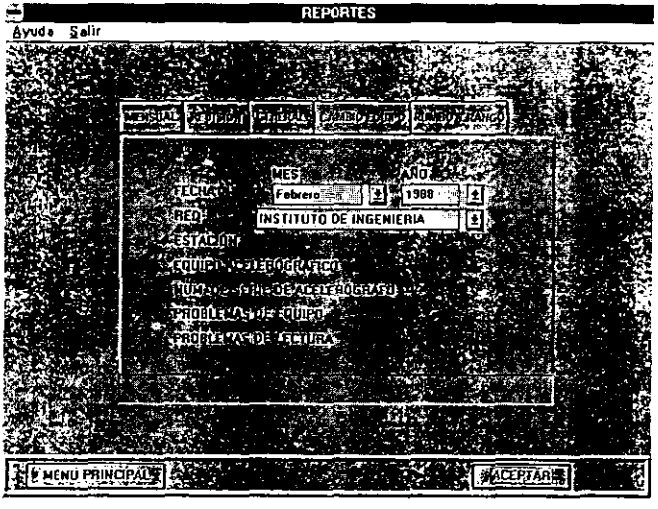
Esta presenta la opción de realizar una consulta con los datos que se soliciten pero si se desea escoger una de las consultas ya establecidas la pantalla que se desplegará será la siguiente:



Las dos pantallas anteriores generarán otra pantalla de resultados.

⇒ Reportes.

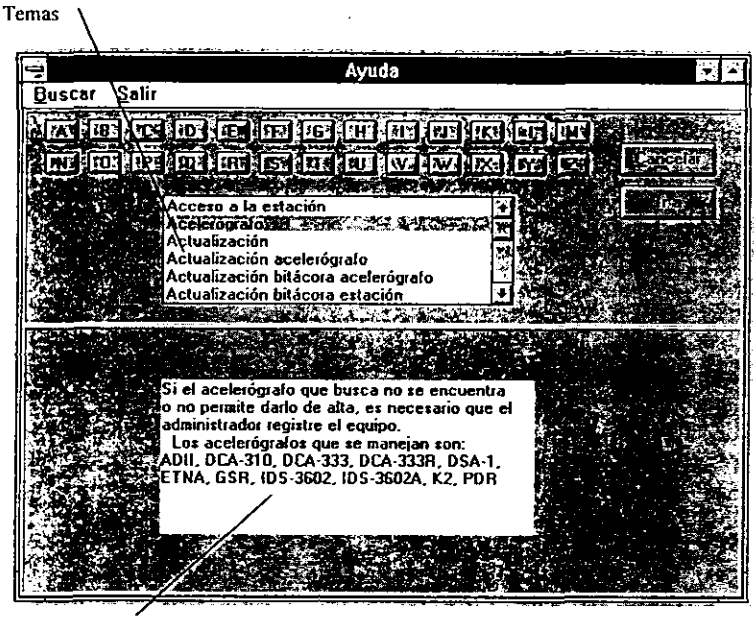
Cuando se selecciona ésta opción en el menú principal se despliega la siguiente pantalla:



En ésta se presentan cinco reportes, de los cuales solo puede ser seleccionado uno a la vez, cuando se selecciona un reporte se deben incluir las variables que se solicite, de lo contrario no habrá resultados. Cada reporte genera otra pantalla de resultados.

⇒ Ayuda

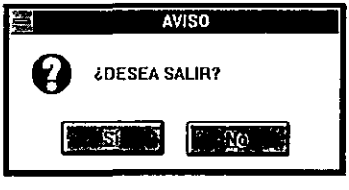
Cuando se selecciona esta opción en el menú principal o en el menú de cualquier pantalla del sistema se despliega la siguiente pantalla:



El resultado de cualquier tema seleccionado será desplegado en la parte inferior de la pantalla, como se puede observar.

⇒ Salir

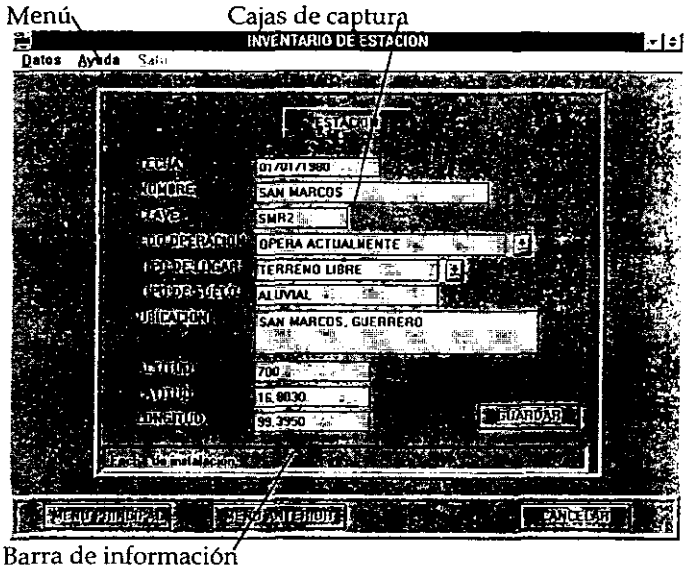
Cuando se selecciona esta opción en el menú principal o en el menú de algunas pantallas del sistema se despliega el siguiente mensaje:



Si se desea terminar con la aplicación, se debe presionar el botón con la opción de SI, terminando de esta manera el trabajo.

CAPTURA E IMPRESIÓN DE LA INFORMACIÓN.

Las pantallas de captura en este sistema, tienen una estructura como la siguiente:

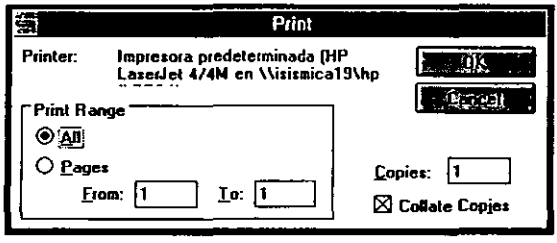


Esta se encuentra integrada de la siguiente manera:

Un menú con opciones para utilizar el sistema, cuadros que reciben la información, notas que aclaran lo que se debe introducir como información y una barra con botones de comando para continuar o interrumpir el procedimiento.

La impresión de información puede ser utilizada al realizar una consulta siguiendo los siguientes pasos:

Dentro de la pantalla de resultados de la consulta o reporte realizado, en la barra del menú se debe seleccionar la opción de imprimir. Al momento de seleccionar esta opción aparecerá la siguiente ventana en la que puede seleccionar qué y cuántas páginas desea imprimir.



Anexo:

Programa fuente (SQL)

PROGRAMA FUENTE (SQL).

```
create table T_ACCESO_ESTACION                    ' Creación de la tabla de acceso a la estación
( acc_est_cl_est char(4) not null,
  acc_est_acceso text not null,
  acc_est_encargado  varchar(30) null,
  acc_est_vers_cand  varchar(20) not null,
  acc_est_vers_llave  char(2) not null,
  primary key clustered(acc_est_cl_est ))

create table T_ALIMENTACION                       ' Creación de la tabla de alimentación de la estación
(alim_cl_est  char(4)  not null,
 alim_tipo_alim varchar(15) not null,
 alim_torre  char(2) null,
 alim_altura  float null,
 alim_num_celd  smallint null,
 alim_cap_celd text null,
 alim_tipo_cable varchar(15) null,
 alim_calibre smallint null,
 alim_num_bat smallint null,
 alim_tipo_regu  varchar(20) null,
 primary key nonclustered (alim_cl_est, alim_tipo_alim))

create table T_BITACORA                           ' Creación de la tabla de bitácora de estación
(bitac_cl_est char(4) not null,
 bitac_fech_rev_est datetime not null,
 bitac_cl_umbral char(12) not null,
 bitac_prob_est text not null,
 bitac_persona_rev char(25) not null,
 primary key nonclustered(bitac_cl_est,bitac_fech_rev_est))

create table T_BITACORA_LECTURA                   ' Creación de la tabla de bitácora de lectura
(bitac_lec_ns_eq_acel smallint not null,
 bitac_lec_cl_mode_eq_acel char(3) not null,
 bitac_lec_fech_rev_lec  datetime not null,
 bitac_lec_descrip  text not null,
 bitac_lec_persona char(25) not null,
 primary key nonclustered (bitac_lec_ns_eq_acel,
bitac_lec_cl_mode_eq_acel, bitac_lec_fech_rev_lec))

create table T_EDO_OPERA_EQ_ACELEROGRAFICO        ' Creación de la tabla del estado de operación
( ope_equi_acel_cl_mode char(3) not null,         ' del equipo acelerográfico
  ope_equi_acel_ns smallint not null,
  ope_equi_acel_fech_rev datetime not null,
  ope_equi_acel_descrip text not null,
  primary key nonclustered(ope_equi_acel_cl_mode,
   ope_equi_acel_ns, ope_equi_acel_fech_rev))

create table T_EDO_OPERACION_ESTACION             ' Creación de la tabla del estado de operación
(ope_est_cl_oper char(1) not null,                ' de la estación
 ope_est_descrip varchar(35) not null,
 primary key clustered (ope_est_cl_oper))
```

```
create table T_EQUIPO_ACELEROGRAFICO                ' Creación de la tabla del equipo acelerográfico
(equi_acel_cl_mode char(3) not null,
 equi_acel_ns smallint not null,
 equi_acel_cl_propie char(2) not null,
 equi_acel_cl_gana varchar(16) not null,
 equi_acel_fech_rev datetime not null,
 equi_acel_mues smallint not null,
 equi_acel_refe_tiem varchar(25) not null,
 equi_acel_prev float not null,
 equi_acel_posev float not null,
 primary key nonclustered (equi_acel_cl_mode,
 equi_acel_ns,equi_acel_fech_rev))

create table T_ERROR                                ' Creación de la tabla de errores, para la interfaz
(cl_error int not null,
 descripcion_error text not null,
 primary key nonclustered(cl_error))

create table T_ESTACION                             ' Creación de la tabla de estación
( est_cl_red char(2) not null,
  est_cl_oper char(1) not null,
  est_cl_est char(4) not null,
  est_nom_est varchar(35) not null,
  est_cl_tipo_lugar smallint not null,
  est_fech_inst datetime not null,
  est_fech_reti datetime null,
  est_tipo_suelo varchar(100) not null,
  est_ubicacion text not null,
  est_altitud smallint not null,
  est_latitud float not null,
  est_longitud float not null,
  primary key clustered (est_cl_est) )

create table T_FABRICANTE                           ' Creación de la tabla de fabricante
(fab_cl_mode_eq_acel char(3) not null,              ' de los equipos acelerográficos
 fab_cl_tipo_eq_acel char(1) not null,
 fab_mode_eq_acel varchar(11) not null,
 fab_fabricante varchar(35) not null,
 fab_pais varchar(10) not null,
 primary key clustered (fab_cl_mode_eq_acel))

create table T_GANANCIA                             ' Creación de la tabla de ganancia
(gana_cl_gana varchar(16) not null,                 ' de los equipos acelerográficos
 gana_canal char(2) not null,
 gana_ganancia smallint not null,
 primary key nonclustered(gana_cl_gana,gana_canal))

create table T_PROPIETARIO                          ' Creación de la tabla de propietarios
(prop_cl_propie char(2) not null,                   ' de los equipos acelerográficos
 prop_prop_eq_acel varchar(75) not null,
 primary key clustered(prop_cl_propie))
```

```
create table T_RED                                    ' Creación de la tabla de redes acelerográficas
( red_cl_red char(2) not null,
  red_nom varchar(25) not null,
  primary key clustered(red_cl_red))

create table T_RESULT                                 ' Creación de la tabla de resultados
(result int not null,
 primary key nonclustered(result))

create table T_SENSOR                                 ' Creación de la tabla de sensores
( sens_mode_sensor varchar(10) not null,
  sens_ns_sensor int not null,
  sens_fech_camb_rango datetime not null,
  sens_canal_sensor char(2) not null,
  sens_rango_sensor float not null,
  sens_frec_sensor float null,
  sens_amort_sensor float not null,
  sens_observaciones text null,
  sens_persona_rev char(25) not null,
  primary key nonclustered (sens_mode_sensor,
  sens_ns_sensor, sens_fech_camb_rango))

create table T_TIPO_EQ_ACELEROGRAFICO                 ' Creación de la tabla de tipo de equipo
(tip_acel_cl_tipo_eq_acel char(1) not null,           ' acelerográfico
 tip_acel_descrip varchar(10) not null,
primary key clustered(tip_acel_cl_tipo_eq_acel))

create table T_TIPO_EQ_SENSOR                         ' Creación de la tabla de tipo de equipo sensor.
(tip_sens_cl_tipo_eq_sens char(1) not null,
 tip_sens_descrip char(7) not null,
primary key clustered(tip_sens_cl_tipo_eq_sens))

create table T_TIPO_LUGAR                             ' Creación de la tabla de tipo de lugar
(tip_lug_cl_tipo_lugar smallint not null,             ' en el que se encuentra la estación
 tip_lug_descrip_lugar varchar(40) not null,
primary key clustered(tip_lug_cl_tipo_lugar))
create table T_UMBRAL
(umb_cl_umbral char(12) not null,
 umb_canal char(2) not null,
 umb_umbral smallint not null,
 primary key nonclustered(umb_cl_umbral,umb_canal))

create table TA_EQUIPO_SENSOR                         ' Creación de la tabla de equipo sensor
(equi_sens_mode_eq_sensor varchar(10) not null,
 equi_sens_ns_eq_sensor smallint not null,
 equi_sens_fech_inst_sensor datetime not null,
 equi_sens_fech_reti_sensor datetime null,
 equi_sens_mode_sensor varchar(10) not null,
 equi_sens_ns_sensor int not null,
 primary key nonclustered(equi_sens_mode_eq_sensor,
  equi_sens_ns_eq_sensor, equi_sens_fech_inst_sensor,
  equi_sens_mode_sensor ,equi_sens_ns_sensor))
```

```
create table TA_EQ_ACELEROGRAFICO_EQ_SENSOR     ' Creación de la tabla que relaciona al
(equi_sens_acel_cl_mode_eq_acel char(3) not null,  ' equipo acelerográfico con el equipo sensor
 equi_sens_acel_ns_eq_acel smallint not null,
 equi_sens_acel_fech_inst_eq_s datetime not null,
 equi_sens_acel_fech_reti_eq_s datetime null,
 equi_sens_acel_cl_tipo_eq_sens char(1) not null,
 equi_sens_acel_num_canal smallint not null,
 equi_sens_acel_mode_eq_sensor  varchar(10) not null,
 equi_sens_acel_ns_eq_sensor smallint not null,
 primary key nonclustered
(equi_sens_acel_cl_mode_eq_acel,equi_sens_acel_ns_eq_acel,
 equi_sens_acel_fech_inst_eq_s, equi_sens_acel_mode_eq_sensor,
 equi_sens_acel_ns_eq_sensor))


create table TA_ESTACION_EQ_ACELEROGRAFICO      ' Creación de la tabla que une a la estación
(est_equi_acel_cl_est char(4) not null,                ' con el equipo acelerográfico
 est_equi_acel_cl_mode_eq_acel char(3) not null,
 est_equi_acel_ns_eq_acel smallint not null,
 est_equi_acel_rumbo_orient varchar(98) not null,
 est_equi_acel_fech_inst_eq_ac datetime not null,
 est_equi_acel_fech_reti_eq_ac datetime null,
 primary key nonclustered(est_equi_acel_cl_est,
  est_equi_acel_fech_inst_eq_ac))


create default d_nulo as "00"                   ' Creación que llena por default con '00' algunos
                                                ' datos de la base

create rule r_tipo_lugar as @tipo_lugar like "[1-9]"   ' Regla para el tipo de lugar
create rule r_cl_est as @cl_est like "[A-Z][A-Z][A-Z]%"  ' Regla que restringe la clave de la estación
create rule r_cl_red as @cl_red like "[A-Z][A-Z]"  ' Regla que restringe la clave de la red acelerográfica
create rule r_cl_oper as @cl_oper like "[A-Z]"  ' Regla que restringe la clave de operación
create rule r_vers_llav as @vers_llav like "[1-9]"  ' Regla que restringe la versión de la llave


create trigger tr_acceso_estacion               ' Este trigger no permite introducir en la tabla de
on T_ACCESO_ESTACION                            ' acceso_estación una clave de estación no dada
for insert, update                              ' de alta en la tabla de estación
as                                              ' Se activa al hacer una operación de inserción o
if @@rowcount=0                                 ' cambio de datos en la tabla de acceso_estacion
   return
if(select count(*) from T_ESTACION,inserted,T_ACCESO_ESTACION
where
T_ESTACION.est_cl_est =inserted.acc_est_cl_est)=@@rowcount
begin
  print "Tratando de insertar una clave de estacion en la tabla
   acceso a estacion no valida en la tabla de estacion"
  rollback transaction
  return
end
commit transaction
return
```

```
create trigger tr_alimentacion                    ' Este trigger no permite introducir en la tabla de
on T_ALIMENTACION                                 ' alimentación una clave de estación no dada de
for insert, update                                ' alta en la tabla de estación
as                                                ' Se activa al realizar una operación de cambio o
if @@rowcount=0                                   ' de inserción de la clave de estación
   return
if(select count(*) from T_ALIMENTACION,inserted,T_ESTACION
where T_ESTACION.est_cl_est =inserted.alim_cl_est)=@@rowcount
begin
  print "Tratando de insertar una clave de estacion en la
 tabla alimentacion no valida en la tabla de estacion"
  rollback transaction
  return
end
commit transaction
return
```

```
create trigger tr_bitacora                        ' El trigger no permite introducir en la tabla de
on T_BITACORA                                     ' bitácora una clave de estación no dada de alta
for insert, update                                ' en la tabla de estación
as                                                ' Se activa al realizar una operación de cambio o
if @@rowcount=0                                   ' inserción de la clave de estación
   return
if(select count(*) from
T_ESTACION,inserted,T_BITACORA
where T_ESTACION.est_cl_est =inserted.bitac_cl_est)=@@rowcount
begin
  print "Tratando de insertar una clave de estacion en
  la tabla bitacora no valida en la tabla de estacion "
  rollback transaction
  return
end
commit transaction
return
```

```
create trigger tr_bitacora_lectura                ' El trigger no permite introducir en la tabla de
on T_BITACORA_LECTURA                             ' bitacora_lectura una clave de equipo y número de
for insert, update as                             ' serie de equipo acelerográfico que no hayan sido
if @@rowcount=0                                   ' dados de alta en la tabla de equipo acelerográfico.
   return                                         ' Se activa al realizar una operación de cambio o
if(select count(*) from T_EQUIPO_ACELEROGRAFICO,  ' inserción de los datos mencionados
inserted,T_BITACORA_LECTURA
where T_EQUIPO_ACELEROGRAFICO.equi_acel_ns
 =inserted.bitac_lec_ns_eq_acel) =@@rowcount
Begin
  print "Tratando de insertar en la tabla de bitacora lectura
 un numero de serie de equipo acelerografico no valido en
 la tabla de equipo acelerografico"
  rollback transaction
  return
end
if (select count(*) from T_EQUIPO_ACELEROGRAFICO,
inserted,T_BITACORA_LECTURA
where
T_EQUIPO_ACELEROGRAFICO.equi_acel_cl_mode=
inserted.bitac_lec_cl_mode_eq_acel) =@@rowcount
begin
  print "Tratando de insertar en la tabla de bitacora lectura una clave
```

```
 de modelo de equipo acelerografico no valido en la tabla de equipo acelerografico"
  rollback transaction
  return
end
commit transaction
return

create trigger tr_edo_opera_eq_acelerografico          ' El trigger no permite introducir en la tabla de
on T_EDO_OPERA_EQ_ACELEROGRAFICO                       '  estado de operación una clave y número de
serie
for insert, update                                     ' de equipo acelerográfico que no hayan sido dados
as                                                     ' de alta en la tabla de equipo acelerográfico
if @@rowcount=0                                        ' Se activa al realizar la operación de cambio o
  return                                               ' inserción
if(select count(*) from T_EDO_OPERA_EQ_ACELEROGRAFICO,
inserted,T_EQUIPO_ACELEROGRAFICO  where
T_EQUIPO_ACELEROGRAFICO.equi_acel_cl_mode=
inserted.ope_equi_acel_cl_mode) =@@rowcount
begin
  print "Tratando de insertar en la tabla de estado de operacion
   de equipo acelerografico un equipo con una clave de modelo
   no valido en la tabla de equipo acelerografico"
  rollback transaction
  return
end
if(select count(*) from
T_EDO_OPERA_EQ_ACELEROGRAFICO,inserted,
T_EQUIPO_ACELEROGRAFICO where
T_EQUIPO_ACELEROGRAFICO.equi_acel_ns =
 inserted.ope_equi_acel_ns)=@@rowcount
begin
  print "Tratando de insertar en la tabla de estado de operacion
   de equipo acelerografico un equipo con numero de serie no valido
   en la tabla de equipo acelerografico"
  rollback transaction
  return
end
commit transaction
return


create trigger tr_eq_acelerografico_eq_sensor          ' El trigger no permite introducir en la tabla
on TA_EQ_ACELEROGRAFICO_EQ_SENSOR                      ' de equipo acelerográfico una clave y número de
for insert, update                                     ' serie de equipo acelerográfico, ni insertar en la
as                                                     ' tabla de equipo_acelerografico_equipo_sensor
if @@rowcount=0                                        ' una clave del tipo de equipo sensor no valido en
  return                                               ' la tabla de tipo de equipo sensor
if(select count(*) from
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, inserted,
 TA_EQUIPO_SENSOR
where TA_EQUIPO_SENSOR.equi_sens_mode_eq_sensor=
inserted.equi_sens_acel_mode_eq_sensor) =@@rowcount
Begin
 print "Tratando de insertar en la tabla auxiliar de equipo
  acelerografico equipo sensor un modelo de equipo sensor
  no valido en la tabla de equipo sensor"
rollback transaction
return
end
if(select count(*) from
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, inserted,
```

```
TA_EQUIPO_SENSOR
where TA_EQUIPO_SENSOR.equi_sens_ns_eq_sensor=
inserted.equi_sens_acel_ns_eq_sensor) =@@rowcount
Begin
  print "Tratando de insertar en la tabla auxiliar de equipo
acelerografico equipo sensor un numero de serie de equipo
 sensor no valido en la tabla de equipo sensor"
rollback transaction
return
end
 if(select count(*) from
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, inserted,
T_EQUIPO_ACELEROGRAFICO
where T_EQUIPO_ACELEROGRAFICO.equi_acel_cl_mode
=inserted.equi_sens_acel_cl_mode_eq_acel) =@@rowcount
begin
  print " Tratando de insertar en la tabla auxiliar de equipo
    acelerografico una clave de modelo de equipo acelerografico
     no valido en la tabla de equipo acelerografico"
rollback transaction
return
end
if (select count(*) from
TA_EQ_ACELEROGRAFICO_EQ_SENSOR,inserted,
T_EQUIPO_ACELEROGRAFICO
where
T_EQUIPO_ACELEROGRAFICO.equi_acel_ns=
inserted.equi_sens_acel_ns_eq_acel) =@@rowcount
begin
  print "Tratando de insertar un numero de serie en la tabla
 auxiliar de equipo acelerografico equipo sensor no valido
en la tabla equipo_acelerografico "
  rollback transaction
  return
end
if(select count(*) from
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, inserted,
T_TIPO_EQ_SENSOR
whereT_TIPO_EQ_SENSOR.tip_sens_cl_tipo_eq_sens=
inserted.equi_sens_acel_cl_tipo_eq_sens) =@@rowcount
begin
  print "Tratando de insertar una clave de tipo de equipo
    sensor en la tabla auxiliar de equipo acelerografico
    equipo sensor no valido en la tabla de tipo de equipo sensor "
 rollback transaction
 return
end
 commit transaction
return
```

```
create trigger tr_equipo_acelerografico
on T_EQUIPO_ACELEROGRAFICO
for insert, update
as
if @@rowcount=0
   return
if(select count(*) from
T_EQUIPO_ACELEROGRAFICO,inserted,T_FABRICANTE
```

```
' El trigger no permite introducir en la tabla de
        ' equipo_acelerografico una clave de equipo
' acelerográfico que no haya sido dado de alta
' en la tabla de fabricante o insertar  en la tabla de
' equipo_acelerografico una clave de propietario
' no dado de alta en la tabla de propietario
```

```
 where
T_FABRICANTE.fab_cl_mode_eq_acel=
inserted.equi_acel_cl_mode)=@@rowcount
begin
  print "Tratando de insertar una clave de equipo acelerografico
    en la tabla equipo acelerografico no valida en la tabla de fabricante "
  rollback transaction
  return
end
if(select count(*) from T_EQUIPO_ACELEROGRAFICO,
 inserted, T_PROPIETARIO where
T_PROPIETARIO.prop_cl_propie=inserted.equi_acel_cl_propie) =@@rowcount
begin
   print "Tratando de insertar una clave de propietario en la
   tabla de equipo acelerografico no valida en la tabla de propietario"
   rollback transaction
  return
end
commit transaction
return


create trigger tr_equipo_sensor                    ' valida que exista el. sensor  que va a  ser a
on TA_EQUIPO_SENSOR                                ' asignado a un equipo sensor y asimismo que dicho
for insert, update                                 ' equipo sensor exista
as                                                 ' Forma de procedencia: man_eq_sens
if @@rowcount=0                                    ' Código de regreso: mensaje en caso de error
   return
if(select count(*) from
TA_EQUIPO_SENSOR,inserted,T_SENSOR
 Where T_SENSOR.sens_ns_sensor =
inserted.equi_sens_ns_sensor ) =@@rowcount
Begin
  print " Tratando de insertar en la tabla auxiliar de equipo
  sensor un numero de serie de sensor no valido en la tabla de sensor"
  rollback transaction
  return
end
if(select count (*) from TA_EQUIPO_SENSOR, inserted, T_SENSOR
where T_SENSOR.sens_mode_sensor=
inserted.equi_sens_mode_sensor)= @@rowcount
Begin
  print " Tratando de insertar en la tabla auxiliar de equipo
   sensor un modelo de sensor no valido en la tabla de sensor"
  rollback transaction
  return
end
if(select count(*) from TA_EQUIPO_SENSOR, inserted,
TA_EQ_ACELEROGRAFICO_EQ_SENSOR where
TA_EQ_ACELEROGRAFICO_EQ_SENSOR.equi_sens_acel_mode_eq_sensor=
inserted.equi_sens_mode_eq_sensor ) =@@rowcount
Begin
  print "Tratando de insertar en la tabla auxiliar de equipo sensor
  un modelo de equipo sensor no valido en la tabla auxiliar de
   equipo acelerografico equipo sensor"
  rollback transaction
  return
end
commit transaction
return
```

```
create trigger tr_estacion                                    ' El trigger no permite introducir en la tabla de
on T_ESTACION                                                 ' estación una clave de red no valida en la tabla de
 for insert, update                                                   ' red o insertar en la tabla de red una clave de
as                                                            ' operación no valida en la tabla de estado de
if @@rowcount=0                                               ' operación de estación o insertar en la tabla de
   return                                                     ' estación una clave de tipo de lugar no valida en la
if(select count(*) from T_ESTACION,inserted,T_RED             ' tabla tipo_lugar, se activa al realizar las
where T_RED.red_cl_red =inserted.est_cl_red )=@@rowcount      ' operaciones de cambiar o insertar datos
begin
  print "Tratando de insertar una clave de red en la tabla
   de estacion no valida en la tabla de red"
  rollback transaction
  return
end
if(select count (*) from T_ESTACION, inserted,T_EDO_OPERACION_ESTACION
where
T_EDO_OPERACION_ESTACION.ope_est_cl_oper=
inserted.est_cl_oper) =@@rowcount
begin
  print "Tratando de insertar en la tabla de red una clave de
   operacion no valida en la tabla de estado de operacion de estacion"
  rollback transaction
  return
end
if(select count (*) from T_ESTACION, inserted,T_TIPO_LUGAR where
T_TIPO_LUGAR.tip_lug_cl_tipo_lugar= inserted.est_cl_tipo_lugar) =@@rowcount
begin
    print "Tratando de insertar una clave de tipo lugar en la
    tabla de estacion, no valida en la tabla tipo lugar"
    rollback transaction
    return
end
commit transaction
return


                                                              ' El trigger no permite introducir en la tabla
create trigger tr_estacion_eq_acelerografico                          ' estación_equipo_acelerografico una clave de
on TA_ESTACION_EQ_ACELEROGRAFICO                                      ' estación no válida en la tabla de estación o
insertar
for insert,update                                             ' una clave de modelo y número de serie de equipo
as                                                            ' acelerográfico no válidos en la tabla de
if @@rowcount = 0                                             ' equipo_acelerografico
return                                                        ' Se activa al realizar operaciones de cambio o de
if (select count (*) from                                     ' inserción de datos
 TA_ESTACION_EQ_ACELEROGRAFICO,
inserted, T_ESTACION  where
T_ESTACION.est_cl_est = inserted.est_equi_acel_cl_est)= @@rowcount
Begin
  print " Tratando de insertar en la tabla auxiliar de estacion
   equipo acerografico una clave de estacion no valida en la tabla de estacion"
rollback transaction
return
end
if(select count (*) from TA_ESTACION_EQ_ACELEROGRAFICO,
inserted, T_EQUIPO_ACELEROGRAFICO where
T_EQUIPO_ACELEROGRAFICO.equi_acel_cl_mode =
inserted.est_equi_acel_cl_mode_eq_acel ) =@@rowcount
begin
  print " Tratando de insertar en la tabla auxiliar de
  estacion equipo acelerografico una clave de modelo
```

```
   de equipo acelerografico no valido en la
tabla de equipo acelerografico"
 rollback transaction
return
end
if(select count(*) from TA_ESTACION_EQ_ACELEROGRAFICO,
inserted, T_EQUIPO_ACELEROGRAFICO where
 T_EQUIPO_ACELEROGRAFICO.equi_acel_ns =
inserted.est_equi_acel_ns_eq_acel) =@@rowcount
begin
  print "Tratando de insertar tabla auxiliar de estacion equipo acelerografico  un
  numero de serie de equipo acelerografico  no valido en tabla de equipo acelerografico"
 rollback transaction
return
end
commit transaction
return


create trigger tr_fabricante                    ' El trigger no permite introducir en la tabla
on T_FABRICANTE                                 ' fabricante una clave del tipo de equipo
for insert, update                              ' acelerográfico no válida en la tabla de tipo de
as                                              ' equipo acelerográfico
if @@rowcount=0                                 ' Se activa al realizar las operaciones de cambio o
return                                          ' inserción de datos en la tabla fabricante

if(select count(*) from
T_FABRICANTE,inserted,T_TIPO_EQ_ACELEROGRAFICO
where T_TIPO_EQ_ACELEROGRAFICO.tip_acel_cl_tipo_eq_acel
=inserted.fab_cl_tipo_eq_acel)= @@rowcount
begin
  print "Tratando de insertar una clave de tipo de equipo acelerografico
   en la tabla fabricante no valida en la tabla tipo de equipo acelerografico"
  rollback transaction
  return
end
commit transaction
return


create proc acceso(@clave char(4)) as           ' Realiza la consulta  del acceso a una estación
begin                                           ' especifica, recibiendo como parámetro
select                                          ' la clave de la estación
acc_est_acceso, acc_est_encargado,                      ' Forma de procedencia: man_estación
acc_est_vers_cand,acc_est_vers_llave                    ' Código de regreso: datos de consulta
 from T_ACCESO_ESTACION, T_ESTACION
 where
acc_est_cl_est = est_cl_est and
acc_est_cl_est = @clave
if @@rowcount= 0
  insert into T_RESULT values(2)
end
return


create proc adm_fabricante                      ' El procedimiento realiza la inserción de un
nuevo
(@fabricante varchar(35),@modelo varchar(11)            ' fabricante y tipo de equipo acelerográfico,
,@cl_modelo char(3),  @tipo varchar(10),        ' validando que no haya sido dado de alta
@cl_tipo_eq char(1), @pais varchar(10))         ' Forma de procedencia: adm_fabricante
as                                              ' Código de regreso: mensaje en caso de error
```

```
declare @cl_tipo char(1)
begin
create table #t(c char(1))
  if exists(select * from T_TIPO_EQ_ACELEROGRAFICO
        where tip_acel_cl_tipo_eq_acel=@cl_tipo_eq)
   begin
    if not exists(select * from T_TIPO_EQ_ACELEROGRAFICO
          where tip_acel_descrip = @tipo)
     begin
      select * from #t
      insert into T_RESULT values(10)
     end
    else
     begin
      if exists(select * from T_TIPO_EQ_ACELEROGRAFICO where
        tip_acel_descrip = @tipo and tip_acel_cl_tipo_eq_acel=@cl_tipo_eq )
       begin
       select @cl_tipo=(select tip_acel_cl_tipo_eq_acel from
              T_TIPO_EQ_ACELEROGRAFICO where
              tip_acel_descrip=@tipo)
       if exists(select * from T_FABRICANTE where
         fab_cl_mode_eq_acel=@cl_modelo or fab_mode_eq_acel=@modelo)
       begin
         select * from #t
         insert into T_RESULT values(5)
       end
       else
        begin transaction
         insert T_FABRICANTE values
          (@cl_modelo,@cl_tipo,@modelo,@fabricante,@pais)
        commit transaction
       end
      if not exists(select * from T_TIPO_EQ_ACELEROGRAFICO where
        tip_acel_descrip = @tipo and tip_acel_cl_tipo_eq_acel=@cl_tipo_eq )
       begin
        select * from #t
        insert into T_RESULT values(28)
       end
     end
  end
 else
  begin
   if not exists(select * from T_TIPO_EQ_ACELEROGRAFICO
          where tip_acel_cl_tipo_eq_acel=@cl_tipo_eq)
    begin
     if exists(select * from T_TIPO_EQ_ACELEROGRAFICO
          where tip_acel_descrip=@tipo)
      begin
       select * from #t
       insert into T_RESULT values(11)
      end
     if not exists(select * from T_TIPO_EQ_ACELEROGRAFICO
            where tip_acel_descrip=@tipo)
      if not exists(select * from T_TIPO_EQ_ACELEROGRAFICO
          where tip_acel_cl_tipo_eq_acel=@cl_tipo_eq)
      begin
       if not exists(select * from T_TIPO_EQ_ACELEROGRAFICO
          where tip_acel_descrip=@tipo and
          tip_acel_cl_tipo_eq_acel=@cl_tipo_eq)
       if exists(select * from T_FABRICANTE where
        fab_cl_mode_eq_acel=@cl_modelo or fab_mode_eq_acel=@modelo)
```

```
        begin
          select * from #t
          insert into T_RESULT values(5)
        end
        else
        begin transaction
          insert T_TIPO_EQ_ACELEROGRAFICO values(@cl_tipo_eq,@tipo)
          insert T_FABRICANTE values
              (@cl_modelo,@cl_tipo_eq,@modelo,@fabricante,@pais)
        commit transaction
       end
      end
   end
end
return


create proc adm_lugarestacion(@lugar varchar(40),          ' Válida si la información de un nuevo
registro
@clave smallint)                                           ' es correcta o ya existe
as                                                      ' Forma de procedencia: adm_lugarestacion
begin                                                   ' Código de regreso: mensaje en caso de error
create table #t(c char(1))
  if exists(select * from T_TIPO_LUGAR
 where tip_lug_cl_tipo_lugar = @clave)
   begin
     if not exists(select * from T_TIPO_LUGAR where tip_lug_descrip_lugar= @lugar)
      begin
        select * from #t
        insert into T_RESULT values(21)
      end
     else
      begin
        if exists(select * from T_TIPO_LUGAR where tip_lug_cl_tipo_lugar=@clave
              and tip_lug_descrip_lugar=@lugar)
        begin
         select * from #t
         insert into T_RESULT values(5)
        end
        if not exists(select * from T_TIPO_LUGAR where tip_lug_cl_tipo_lugar=@clave
              and tip_lug_descrip_lugar=@lugar)
        begin
         select * from #t
         insert into T_RESULT values(28)
        end
      end
   end
 if exists(select * from T_TIPO_LUGAR where tip_lug_descrip_lugar=@lugar)
  begin
    if not exists(select * from T_TIPO_LUGAR where tip_lug_cl_tipo_lugar=@clave)
     begin
       select * from #t
       insert T_RESULT values(22)
     end
  end
 if not exists(select * from T_TIPO_LUGAR where tip_lug_cl_tipo_lugar=@clave)
 if not exists(select * from T_TIPO_LUGAR where tip_lug_descrip_lugar= @lugar)
 begin
  if not exists(select * from T_TIPO_LUGAR where tip_lug_cl_tipo_lugar=@clave
          and tip_lug_descrip_lugar=@lugar)
   begin transaction
```

```
      insert T_TIPO_LUGAR values(@clave,@lugar)
    commit transaction
   end
end
return


create proc adm_operacion_estacion
(@operacion varchar(35),@clave char(1)) as
begin                                          ' Procedimiento utilizado para insertar un nuevo
create table #t(c char(1))                     ' registro del catálogo del estado de operación de
  if exists(select * from T_EDO_OPERACION_ESTACION  ' una estación
where                                          ' Forma de procedencia: adm_operacionestacion
        ope_est_cl_oper = @clave)              ' Código de regreso: mensaje en caso de error
   begin
     if not exists(select * from T_EDO_OPERACION_ESTACION
 where
         ope_est_descrip= @operacion)
       begin
         select * from #t
         insert into T_RESULT values(23)
       end
      else
       begin
         if exists(select * from T_EDO_OPERACION_ESTACION where
              ope_est_cl_oper=@clave and ope_est_descrip=@operacion)
         begin
          select * from #t  insert into T_RESULT values(5)
         end
         if not exists(select * from T_EDO_OPERACION_ESTACION where
              ope_est_cl_oper=@clave and ope_est_descrip=@operacion)
         begin
          select * from #t
          insert into T_RESULT values(28)
         end
       end
   end
  if not exists(select * from T_EDO_OPERACION_ESTACION where
           ope_est_cl_oper=@clave)
   begin
     if exists(select * from T_EDO_OPERACION_ESTACION where
           ope_est_descrip=@operacion)
      begin
        select * from #t
        insert T_RESULT values(24)
      end
   end
  if not exists(select * from T_EDO_OPERACION_ESTACION where
           ope_est_cl_oper=@clave)
   if not exists(select * from T_EDO_OPERACION_ESTACION where
           ope_est_descrip= @operacion)
   begin
    if not exists(select * from T_EDO_OPERACION_ESTACION where
         ope_est_cl_oper=@clave and ope_est_descrip=@operacion)
    begin transaction
     insert T_EDO_OPERACION_ESTACION values(@clave,@operacion)
    commit transaction
   end
end
return
```

```
create proc adm_propietario(@propietario varchar(75),
@clave char(2)) as
begin
create table #t(c char(1))
  if exists(select * from T_PROPIETARIO where
        prop_cl_propie = @clave)
   begin
     if not exists(select * from T_PROPIETARIO where
        prop_prop_eq_acel= @propietario)
      begin
       select * from #t  insert into T_RESULT values(26)
      end
     else
      begin
       if exists(select * from T_PROPIETARIO where
         prop_prop_eq_acel=@propietario and prop_cl_propie=@clave)
        begin
        select * from #t  insert into T_RESULT values(5)
        end
       if not exists(select * from T_PROPIETARIO where
         prop_prop_eq_acel=@propietario and prop_cl_propie=@clave)
        begin
         select * from #t  insert into T_RESULT values(28)
        end
      end
   end
  if not exists(select * from T_PROPIETARIO where
          prop_cl_propie=@clave)
   begin
     if exists(select * from T_PROPIETARIO where
          prop_prop_eq_acel=@propietario)
      begin
       select * from #t  insert T_RESULT values(27)
      end
   end
  if not exists(select * from T_PROPIETARIO where
          prop_cl_propie=@clave)
   if not exists(select * from T_PROPIETARIO where
          prop_prop_eq_acel= @propietario)
   begin
     if not exists(select * from T_PROPIETARIO where
        prop_prop_eq_acel=@propietario and prop_cl_propie=@clave)
      begin transaction
       insert T_PROPIETARIO values(@clave,@propietario)
     commit transaction
   end
end
return
```

```
' Procedimiento para insertar nueva información del
' propietario de un equipo acelerográfico, validando
' la información
' Forma de procedencia: adm_propietario
' Código de regreso: mensaje en caso de error
```

```
create proc adm_red(@red varchar(25),@clave char(2))
as
begin
create table #t(c char(1))
  if exists(select * from T_RED where red_cl_red = @clave)
   begin
     if not exists(select * from T_RED where red_nom= @red)
      begin
       select * from #t  insert into T_RESULT values(18)
      end
```

```
' Realiza la inserción de un nuevo registro para el
' catálogo de redes, valida si la información es
' correcta o ya existe
' Forma de procedencia: adm_red
' Código de regreso: mensaje en caso de error
```

```
    else
     begin
      if exists(select * from T_RED where red_nom= @red
            and red_cl_red= @clave)
      begin
       select * from #t  insert into T_RESULT values(5)
      end
      if not exists(select * from T_RED where red_nom=@red
            and red_cl_red=@clave)
      begin
       select * from #t  insert into T_RESULT values(28)
      end
     end
   end
  if exists(select * from T_RED where red_nom=@red)
   begin
    if not exists(select * from T_RED where red_cl_red=@clave)
     begin
      select * from #t
      insert T_RESULT values(16)
     end
   end
  if not exists(select * from T_RED where red_cl_red=@clave)
   if not exists(select * from T_RED where red_nom= @red)
   begin
    if not exists(select * from T_RED where red_cl_red=@clave
            and red_nom=@red)
     begin transaction
      insert T_RED values(@clave,@red)
     commit transaction
    end
end
return


create proc adm_tipsens(@tipo char(7),@clave char(1))      ' Procedimiento para insertar un nuevo registro
as                                                         ' del catálogo de tipo de sensores, validando la
begin                                                      ' información que es insertada
create table #t(c char(1))                                 ' Forma de procedencia: adm_tipsens
  if exists(select * from T_TIPO_EQ_SENSOR where           ' Código de regreso: mensaje en caso de error
        tip_sens_cl_tipo_eq_sens = @clave)
   begin
    if not exists(select * from T_TIPO_EQ_SENSOR where
            tip_sens_descrip= @tipo)
     begin
      select * from #t
      insert into T_RESULT values(19)
     end
    else
     begin
      if exists(select * from T_TIPO_EQ_SENSOR where
       tip_sens_descrip=@tipo and tip_sens_cl_tipo_eq_sens=@clave)
       begin
        select * from #t  insert into T_RESULT values(5)
       end
      if not exists(select * from T_TIPO_EQ_SENSOR where
       tip_sens_descrip=@tipo and tip_sens_cl_tipo_eq_sens=@clave)
       begin
        select * from #t  insert into T_RESULT values(28)
       end
     end
```

```
  end
 if exists(select * from T_TIPO_EQ_SENSOR where tip_sens_descrip=@tipo)
  begin
    if not exists(select * from T_TIPO_EQ_SENSOR where
            tip_sens_cl_tipo_eq_sens=@clave)
     begin
       select * from #t
       insert T_RESULT values(20)
     end
  end
 if not exists(select * from T_TIPO_EQ_SENSOR where
          tip_sens_cl_tipo_eq_sens=@clave)
  if not exists(select * from T_TIPO_EQ_SENSOR where
           tip_sens_descrip= @tipo)
  begin
   if not exists(select * from T_TIPO_EQ_SENSOR where
       tip_sens_cl_tipo_eq_sens=@clave and tip_sens_descrip=@tipo)
    begin transaction
     insert T_TIPO_EQ_SENSOR values(@clave,@tipo)
   commit transaction
  end
end
return


create proc alimentacion(@clave char(4),
@tipo varchar(15)) as                    ' El procedimiento se encarga de mostrar las
begin                                    ' características de alimentación de la estación,
select                                   '  recibiendo como parámetros la clave de la
alim_tipo_alim,alim_torre,alim_altura,alim_num_celd,        '  estación y el tipo de alimentación
 alim_cap_celd, alim_tipo_cable, alim_calibre,              ' Forma de procedencia: alimentación
alim_num_bat, alim_tipo_regu             ' Código de regreso: consulta de datos
 from T_ALIMENTACION, T_ESTACION
 where
alim_cl_est=est_cl_est and
alim_cl_est = @clave
if @@rowcount= 0
  insert into T_RESULT values(2)
end
return


create proc eq_sens_int(@modelo varchar(10), @nserie int) as     ' Realiza la consulta de un equipo sensor
interno
begin                                    ' específico
select distinct                          ' Forma de procedencia: man_eq_sensor
sens_mode_sensor, sens_ns_sensor,                ' Código de regreso: consulta de datos
sens_fech_camb_rango,sens_rango_sensor,
sens_frec_sensor,sens_amort_sensor
from T_SENSOR
where sens_mode_sensor= @modelo and
sens_ns_sensor= @nserie
group by sens_ns_sensor
having sens_fech_camb_rango= min(sens_fech_camb_rango)
if @@rowcount= 0
  insert into T_RESULT values(25)
 end
return
```

```
create proc equipo(@modelo varchar(11), @nserie smallint) as      ' Utilizado para consultar la información sobre
begin                                                              ' un equipo acelerográfico existente
select distinct                                                    ' Forma de procedencia: man_eq_acelerografico
prop_prop_eq_acel,equi_acel_fech_rev,equi_acel_mues,               ' Código de regreso: consulta de datos
equi_acel_refe_tiem, equi_acel_prev,
equi_acel_posev,equi_acel_gana_c1,equi_acel_gana_c2,
equi_acel_gana_c3
from T_EQUIPO_ACELEROGRAFICO,
T_FABRICANTE, T_PROPIETARIO
where
fab_mode_eq_acel= @modelo and
equi_acel_ns= @nserie and
equi_acel_cl_mode=fab_cl_mode_eq_acel and
prop_cl_propie=equi_acel_cl_propie
group
by prop_prop_eq_acel
having equi_acel_fech_rev= min(equi_acel_fech_rev)
if @@rowcount= 0
  insert into T_RESULT values(1)
end
return


create proc estacion_ec(@clave char(4)) as                         ' Realiza la consulta de información sobre una
begin                                                              ' estación específica.
select                                                             ' Forma de procedencia: man_estacion
est_fech_inst, est_nom_est, est_cl_est, red_nom,ope_est_descrip,   ' Código de regreso: Consulta de datos
tip_lug_descrip_lugar,
est_tipo_suelo,est_ubicacion,est_altitud,est_latitud,est_longitud
from T_ESTACION, T_EDO_OPERACION_ESTACION,
T_RED, T_TIPO_LUGAR
where
est_cl_est= @clave and
est_cl_red= red_cl_red and
est_cl_tipo_lugar = tip_lug_cl_tipo_lugar  and
est_cl_oper = ope_est_cl_oper
if @@rowcount= 0
  insert into T_RESULT values(2)
end
return


create proc fabricante(@modelo varchar(11)) as                     ' El procedimiento selecciona los datos de fabricante
begin                                                              ' de un equipo acelerográfico específico
select                                                             ' Forma de procedencia: adm_fabricante
 fab_cl_mode_eq_acel, fab_cl_tipo_eq_acel,                         ' Código de regreso: consulta de datos
fab_mode_eq_acel,  fab_fabricante, fab_pais,
 tip_acel_descrip
 from T_FABRICANTE, T_TIPO_EQ_ACELEROGRAFICO
where
fab_mode_eq_acel = @modelo and
fab_cl_tipo_eq_acel = tip_acel_cl_tipo_eq_acel
if @@rowcount= 0
  insert into T_RESULT values(17)
end
return
```

```
create proc ganancia(@clgana varchar(16),                    ' Se utiliza para asignar  la clave de ganancia.
@g1 varchar(100), @numcanal smallint) as                     ' Procedimiento de procedencia: man_eq_acel
begin
 declare @coma char(1)
 declare @x char(1)
 declare @canal char(2)
 declare @c smallint
 declare @c1 smallint
 declare @gan varchar(5)
  select @c=1
  select @c1=0
  select @coma=","
  while @c1<@numcanal
  begin
    select @x=substring(@g1,@c,1)
    while @x<>@coma
    begin
        select @x=substring(@g1,@c,1)
        select @gan=@gan+@x
        select @c=@c+1
        select @x=substring(@g1,@c,1)
    end
    select @canal=convert(char(2),@c1+1)
    select @c=@c+1
    select @c1=@c1+1
    begin transaction
      insert T_GANANCIA values  (@clgana,@canal,convert(smallint,@gan))
    commit transaction
    select @gan=''
    select @canal=''
  end
end
return
```

```
create proc man_eq_sens(@fecha datetime,@modeloeq            ' Realiza la inserción de información sobre un
varchar(10),@nserieeq smallint, @modelo varchar(10),                ' nuevo equipo sensor
@nserie smallint, @canal char(2),@rango float,               ' Forma de procedencia: man_eq_sens
@frecuencia float,@amort float, @persona char(25)) as        ' Código de regreso: mensaje en caso de error
begin
declare @x varchar(25)
select @x='Sin observaciones'
create table #t(x char(1))
 if exists(select * from T_SENSOR, TA_EQUIPO_SENSOR where
       sens_mode_sensor = @modelo and
       sens_ns_sensor= @nserie and
       sens_fech_camb_rango=@fecha and
       sens_mode_sensor=equi_sens_mode_sensor and
       sens_ns_sensor=equi_sens_ns_sensor and
       equi_sens_mode_eq_sensor=@modeloeq and
       equi_sens_ns_eq_sensor=@nserieeq )
 begin
  select * from #t
  insert into T_RESULT values(5)
 end
else
begin
 if exists(select * from T_SENSOR where
       sens_mode_sensor = @modelo and
       sens_ns_sensor= @nserie and
```

```
        sens_fech_camb_rango=@fecha)
    begin
      select * from #t
      insert into T_RESULT values(29)
    end
    else
     begin
      if exists(select * from TA_EQUIPO_SENSOR where
         equi_sens_mode_eq_sensor=@modeloeq and
         equi_sens_ns_eq_sensor=@nserieeq  and
         equi_sens_fech_inst_sensor=@fecha and
         equi_sens_mode_sensor=@modelo and
         equi_sens_ns_sensor =@nserie)
      begin
        select * from #t
        insert into T_RESULT values(30)
      end
      else
       begin
        begin transaction
         insert T_SENSOR values
         (@modelo,@nserie,@fecha,@canal,@rango,@frecuencia,@amort,@x,@persona)
         insert TA_EQUIPO_SENSOR values
         (@modeloeq,@nserieeq,@fecha,'',@modelo,@nserie)
        commit transaction
       end
end
end
end
return



create proc man_estacion_red(@red varchar(25),
@cl_est char(4)) as
begin                                          ' El procedimiento asocia una nueva estación con
 declare @cl_red char(2)                       ' la red a la que pertenece, en caso de no existir
 create table #t(c char(1) not null)           ' asociación le asigna NP que significa que no
  if exists(select * from T_RED where                  ' pertenece a ninguna red
        red_nom= @red)                         ' Forma de procedencia: man_estación
  begin                                        ' Código de regreso: mensaje en caso de error
   select @cl_red=(select red_cl_red from
            T_RED where red_nom = @red)
      begin transaction
       update  T_ESTACION set est_cl_red=@cl_red where
       est_cl_est= @cl_est
      commit transaction
  end
   else
    begin
     select * from #t
     insert into T_RESULT values(6)
    end
end
return
```

```
create proc man_estacion(@fecha datetime,
@nombre varchar(35),@cl_est                    ' Inserta información de una nueva estación,
char(4),@operacion varchar(35), @lugar varchar(40),   ' valida que los datos insertados sean correctos
@suelo varchar(100), @ubicacion varchar(255),         ' o que no existan
@altitud smallint, @latitud float,                    ' Forma de procedencia: man_estacion
@longitud float) as                                   ' Código de regreso: mensaje en caso de error
begin
 declare @cl_oper char(1)
 declare @cl_tipo_lugar smallint
 create table #t(x char(1))
  if exists(select * from T_EDO_OPERACION_ESTACION where
        ope_est_descrip= @operacion)
  begin
    select @cl_oper=(select ope_est_cl_oper from
              T_EDO_OPERACION_ESTACION
where ope_est_descrip=@operacion)
if exists(select * from T_TIPO_LUGAR where
          tip_lug_descrip_lugar= @lugar)
    begin
      select @cl_tipo_lugar = (select tip_lug_cl_tipo_lugar
                        from T_TIPO_LUGAR
                        where tip_lug_descrip_lugar= @lugar)
      if exists(select * from T_ESTACION where
            est_cl_est= @cl_est)
       begin
         select * from #t
         insert into T_RESULT values( 5)
       end
      else
begin
       begin transaction
        insert T_ESTACION
        (est_cl_oper,est_cl_est,est_nom_est,est_cl_tipo_lugar,
         est_fech_inst,est_tipo_suelo,est_ubicacion,est_altitud,
         est_latitud,est_longitud) values
        (@cl_oper,@cl_est,@nombre,@cl_tipo_lugar,@fecha,
         @suelo,@ubicacion,@altitud,@latitud,@longitud)
       commit transaction
      end
    end
   else
    begin
     select * from #t
     insert into T_RESULT values(8)
    end
  end
else
  begin
    select * from #t
    insert into T_RESULT values(7)
    if not exists(select * from T_TIPO_LUGAR where
          tip_lug_descrip_lugar= @lugar)
      insert into T_RESULT values(8)
 end
end
return
```

```
create proc man_estacion_acc                          ' El procedimiento da de alta los datos de acceso
(@cl_est char(4), @acceso varchar(255),               ' a la estación, verifica que exista la estación y
@encargado varchar(30),                               ' que no estén duplicados los datos
@candado varchar(20), @llave char(2))                 ' Forma de procedencia: man_estacion_acc
as                                                    ' Código de regreso:  mensaje en caso de error
begin
  create table #t(x char(1))
 if exists(select * from T_ESTACION where est_cl_est= @cl_est)
  begin
    begin transaction
     insert T_ACCESO_ESTACION values
     (@cl_est,@acceso,@encargado,@candado,@llave)
    commit transaction
  end
  else
  begin
   select * from #t
   insert into T_RESULT values(2)
  end
end
return


create proc man_estacion_alim                         ' El procedimiento  verifica la existencia de una
(@cl_est char(4), @tipo_alim varchar(15), @torre char(2),  ' estación para dar de alta los datos referentes a la
@altura float, @num_celd smallint, @cap_celd varchar(255),  ' alimentación y que no se dupliquen
@tipo_cable varchar(15), @calibre smallint,@num_bat smallint,
@alim_tipo_regu varchar(20))
as
begin
  create table #t(x char(1))
  if exists(select * from T_ESTACION where est_cl_est= @cl_est)
  begin
    begin transaction
     insert T_ALIMENTACION values
     (@cl_est,@tipo_alim,@torre,@altura,@num_celd,@cap_celd,
      @tipo_cable,@calibre,@num_bat, @alim_tipo_regu)
    commit transaction
  end
  else
  begin
   select * from #t
   insert into T_RESULT values(2)
  end
end
return


create proc p_dat_cambioeq(@red varchar(25))          ' El procedimiento selecciona  los datos
as                                                    '  de los equipos que han cambiado
begin                                                 ' Forma de procedencia: reporte
create table dat_cambioeq(                            ' Código de regreso: datos de consulta
  est_cl_est char(4) not null,
  fab_mode_eq_acel varchar(11) not null,
  est_equi_acel_ns_eq_acel  smallint not null,
  est_equi_acel_fech_inst_eq_ac datetime not null,
  est_equi_acel_fech_reti_eq_ac datetime not null)
insert dat_cambioeq
select
 est_cl_est, fab_mode_eq_acel,
```

```
 est_equi_acel_ns_eq_acel, est_equi_acel_fech_inst_eq_ac,
 est_equi_acel_fech_reti_eq_ac
from
 T_ESTACION, T_FABRICANTE,
 TA_ESTACION_EQ_ACELEROGRAFICO,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR,
 T_RED
where
 est_cl_est = est_equi_acel_cl_est and
 est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
 est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
 est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
 red_cl_red = est_cl_red and
 red_nom = @red
end
return


create proc p_dat_rangos(@equipo varchar(11))          ' Selecciona datos que incluyen los cambios de
as                                                     ' rango de los equipos sensores
begin                                                  ' Forma de procedencia: reporte
create table dat_rangos(                               ' Código de regreso: datos de consulta
 fab_mode_eq_acel varchar(11) not null,
 equi_sens_acel_ns_eq_acel smallint not null,
 equi_sens_acel_mode_eq_sensor varchar(10) not null,
 equi_sens_acel_ns_eq_sensor smallint not null,
 sens_canal_sensor char(2) not null,
 equi_sens_ns_sensor int not null,
 sens_frec_sensor float not null,
 sens_amort_sensor float not null,
 equi_sens_acel_fech_reti_eq_s datetime not null,
 equi_sens_fech_reti_sensor datetime not null,
 sens_fech_camb_rango datetime not null,
 sens_rango_sensor float,
 sens_observaciones text not null)
insert dat_rangos
select
 fab_mode_eq_acel, equi_sens_acel_ns_eq_acel,
 equi_sens_acel_mode_eq_sensor, equi_sens_acel_ns_eq_sensor,
 sens_canal_sensor, equi_sens_ns_sensor,
 sens_frec_sensor, sens_amort_sensor,
 equi_sens_acel_fech_reti_eq_s,
 equi_sens_fech_reti_sensor,
 sens_fech_camb_rango, sens_rango_sensor,
 sens_observaciones
from
 TA_EQUIPO_SENSOR,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR,
 T_FABRICANTE, T_SENSOR
where
 fab_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
 equi_sens_acel_mode_eq_sensor = equi_sens_mode_eq_sensor and
 equi_sens_acel_ns_eq_sensor = equi_sens_ns_eq_sensor and
 equi_sens_mode_sensor = sens_mode_sensor and
 equi_sens_ns_sensor = sens_ns_sensor and
 fab_mode_eq_acel = @equipo
end
return
```

```
create proc p_dat_reporacelsens(@red varchar(25))        ' Selecciona los datos de los equipos acelerográficos
as                                                       ' existentes
begin                                                    ' Forma de procedencia: reporte
create table dat_reporacelsens(                          ' Código de regreso: datos de consulta
  est_cl_est char(4) not null,
  fab_mode_eq_acel varchar(11) not null,
  est_equi_acel_ns_eq_acel  smallint not null,
  est_equi_acel_fech_inst_eq_ac datetime not null,
  est_equi_acel_fech_reti_eq_ac datetime not null,
  est_equi_acel_rumbo_orient varchar(84) not null,
  sens_mode_sensor varchar(10) not null,
  sens_ns_sensor int not null,
  sens_rango_sensor float not null)
insert dat_reporacelsens
select
 est_cl_est, fab_mode_eq_acel,
 est_equi_acel_ns_eq_acel, est_equi_acel_fech_inst_eq_ac,
 est_equi_acel_fech_reti_eq_ac, est_equi_acel_rumbo_orient,
 sens_mode_sensor, sens_ns_sensor, sens_rango_sensor
from
 T_ESTACION, T_FABRICANTE,
 TA_ESTACION_EQ_ACELEROGRAFICO,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR,
 TA_EQUIPO_SENSOR, T_SENSOR, T_RED
where
 est_cl_est = est_equi_acel_cl_est and
 est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
 est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
 est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
 equi_sens_acel_mode_eq_sensor = equi_sens_mode_eq_sensor and
 equi_sens_acel_ns_eq_sensor = equi_sens_ns_eq_sensor and
 equi_sens_mode_sensor = sens_mode_sensor and
 equi_sens_ns_sensor = sens_ns_sensor and
 est_cl_red = red_cl_red and
 red_nom = @red
end
return


create proc p_reporm(@fecha datetime, @red varchar(25))  ' Selecciona datos de revisiones realizadas a
as                                                       ' estaciones, por fecha y red
begin                                                    ' Forma de procedencia: reporte
 create table reporm(                                    ' Código de regreso: datos de consulta
 red_nom varchar(25) not null,
 est_cl_est char(4) not null,
 fab_mode_eq_acel varchar(11) not null ,
est_equi_acel_ns_eq_acel smallint not null,
ope_equi_acel_fech_rev datetime not null,
ope_equi_acel_descrip text not null,
 bitac_lec_descrip text not null)
insert reporm
select
red_nom, est_cl_est, fab_mode_eq_acel, est_equi_acel_ns_eq_acel,
ope_equi_acel_fech_rev, ope_equi_acel_descrip, bitac_lec_descrip
from
T_RED, T_ESTACION, TA_ESTACION_EQ_ACELEROGRAFICO,
T_FABRICANTE, T_EDO_OPERA_EQ_ACELEROGRAFICO,
T_BITACORA_LECTURA
 where
red_cl_red = est_cl_red and
```

```
est_cl_est = est_equi_acel_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = ope_equi_acel_cl_mode and
est_equi_acel_ns_eq_acel = ope_equi_acel_ns and
est_equi_acel_cl_mode_eq_acel = bitac_lec_cl_mode_eq_acel and
est_equi_acel_ns_eq_acel = bitac_lec_ns_eq_acel and
bitac_lec_fech_rev_lec = ope_equi_acel_fech_rev and
bitac_lec_cl_mode_eq_acel = ope_equi_acel_cl_mode and
bitac_lec_ns_eq_acel = ope_equi_acel_ns and
red_nom = @red and
bitac_lec_fech_rev_lec>@fecha
end
return


create proc p_reporr(@fecha datetime, @red varchar(25)) as      ' Selecciona datos de revisión a las estaciones
begin                                                          ' Forma de procedencia: reporte
create table reporr(                                           ' Código de regreso: datos de consulta
red_nom varchar(25) not null,
est_cl_est char(4) not null,
fab_mode_eq_acel varchar(11) not null,
est_equi_acel_ns_eq_acel smallint not null,
bitac_prob_est text not null,
ope_equi_acel_descrip text not null,
bitac_persona_rev char(25) not null)
insert reporr
 select
red_nom, est_cl_est, fab_mode_eq_acel, est_equi_acel_ns_eq_acel,
bitac_prob_est, ope_equi_acel_descrip, bitac_persona_rev
 from
T_RED, T_ESTACION, TA_ESTACION_EQ_ACELEROGRAFICO,
T_FABRICANTE, T_EDO_OPERA_EQ_ACELEROGRAFICO,
T_BITACORA
 where
est_cl_red = red_cl_red and
est_cl_est = est_equi_acel_cl_est and
est_cl_est = bitac_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = ope_equi_acel_cl_mode and
est_equi_acel_ns_eq_acel = ope_equi_acel_ns and
ope_equi_acel_fech_rev = bitac_fech_rev_est and
ope_equi_acel_fech_rev >@fecha and
red_nom = @red
end
return

create view est_red1 as                                        ' La vista muestra los datos de la  red del instituto
select red_nom, est_nom_est, est_cl_est                        ' de ingeniería
from  T_RED, T_ESTACION                                        ' Forma de procedencia: num_consulta
where                                                          ' Código de regreso: datos de consulta
red_cl_red = est_cl_red and
red_cl_red = 'II'


create view est_red1edificios as                               ' La vista muestra los datos
select red_nom, est_nom_est, est_cl_est                        ' de la red de edificios
from  T_RED, T_ESTACION                                        ' Forma de procedencia: num_consulta
where                                                          ' Código de regreso: datos de consulta
red_cl_red = est_cl_red and
red_cl_red = 'ED'
```

```
create view est_red1guerrero as                 ' La vista muestra los datos
select red_nom, est_nom_est, est_cl_est         ' de la red de Guerrero
from T_RED, T_ESTACION                          ' Forma de procedencia: num_consulta
where                                           ' Código de regreso: datos de consulta
red_cl_red = est_cl_red and
red_cl_red = 'GR'

create view est_red1oaxaca as                   ' La vista muestra los datos
select red_nom, est_nom_est, est_cl_est         ' de la red de Oaxaca
from T_RED, T_ESTACION                          ' Forma de procedencia: num_consulta
where                                           ' Código de regreso: datos de consulta
red_cl_red = est_cl_red and
red_cl_red = 'OA'

create view est_red1puebla as                   ' La vista muestra los datos
select red_nom, est_nom_est, est_cl_est         ' de la red de Puebla
from T_RED, T_ESTACION                          ' Forma de procedencia: num_consulta
where                                           ' Código de regreso: datos de consulta
red_cl_red = est_cl_red and
red_cl_red = 'PU'

create view dat_acel2idei as                    ' La vista muestra la clave de estación, el modelo y
select est_cl_est, fab_mode_eq_acel,            ' número de serie del equipo acelerográfico, estado
est_equi_acel_ns_eq_acel, ope_est_descrip,      ' de operación, tipo de sensor, modelo y número de
tip_sens_descrip, equi_sens_acel_mode_eq_sensor, ' serie del sensor de la red del instituto de ingeniería
equi_sens_acel_ns_eq_sensor                     ' Forma de procedencia: num_consulta
from                                            ' Código de error: datos de consulta
 T_RED, T_ESTACION, T_EDO_OPERACION_ESTACION,
 TA_ESTACION_EQ_ACELEROGRAFICO, T_FABRICANTE,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR, T_TIPO_EQ_SENSOR
Where red_cl_red = est_cl_red and
est_cl_oper = ope_est_cl_oper and est_cl_est = est_equi_acel_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
red_cl_red = 'II'

create view dat_acel2edificios as               ' La vista muestra la clave de estación, el modelo y
select est_cl_est, fab_mode_eq_acel,            ' número de serie del equipo acelerográfico, estado
est_equi_acel_ns_eq_acel, ope_est_descrip,      ' de operación, tipo de sensor, modelo y número del
tip_sens_descrip, equi_sens_acel_mode_eq_sensor, ' equipo sensor
equi_sens_acel_ns_eq_sensor                     ' Forma de procedencia: num_consulta
from                                            ' Código de regreso: datos de consulta
T_RED, T_ESTACION, T_EDO_OPERACION_ESTACION,
TA_ESTACION_EQ_ACELEROGRAFICO, T_FABRICANTE,
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, T_TIPO_EQ_SENSOR
Where red_cl_red = est_cl_red and
est_cl_oper = ope_est_cl_oper and
est_cl_est = est_equi_acel_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
red_cl_red = 'ED'
```

```
create view dat_acel2guerrero as                    ' La vista muestra la clave de estación, el modelo y
select est_cl_est, fab_mode_eq_acel,                ' número de serie del equipo acelerográfico, estado
est_equi_acel_ns_eq_acel, ope_est_descrip,          ' de operación, tipo de sensor, modelo y número de
tip_sens_descrip, equi_sens_acel_mode_eq_sensor,    ' serie del sensor de la red de Guerrero
equi_sens_acel_ns_eq_sensor                         ' Forma de procedencia: num_consulta
from                                                ' Código de regreso: consulta de datos
T_RED, T_ESTACION, T_EDO_OPERACION_ESTACION,
TA_ESTACION_EQ_ACELEROGRAFICO, T_FABRICANTE,
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, T_TIPO_EQ_SENSOR
where
red_cl_red = est_cl_red and
est_cl_oper = ope_est_cl_oper and
est_cl_est = est_equi_acel_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
red_cl_red = 'GR'
```

```
create view dat_acel2oaxaca as                      ' La vista muestra la clave de estación, el modelo y
select est_cl_est, fab_mode_eq_acel,                ' número de serie del equipo acelerográfico, estado
est_equi_acel_ns_eq_acel, ope_est_descrip,          ' de operación, tipo de sensor, modelo y número de
tip_sens_descrip, equi_sens_acel_mode_eq_sensor,    ' serie del sensor de la red de Oaxaca
equi_sens_acel_ns_eq_sensor                         ' Forma de procedencia: num_consulta
from                                                ' Código de regreso: consulta de datos
T_RED, T_ESTACION, T_EDO_OPERACION_ESTACION,
TA_ESTACION_EQ_ACELEROGRAFICO, T_FABRICANTE,
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, T_TIPO_EQ_SENSOR
where
red_cl_red = est_cl_red and
est_cl_oper = ope_est_cl_oper and
est_cl_est = est_equi_acel_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
red_cl_red = 'OA'
```

```
create view dat_acel2puebla as                      ' La vista muestra la clave de estación, el modelo y
select est_cl_est, fab_mode_eq_acel,                  número de serie del equipo acelerográfico, estado
est_equi_acel_ns_eq_acel, ope_est_descrip,          ' de operación, tipo de sensor, modelo y número de
tip_sens_descrip, equi_sens_acel_mode_eq_sensor,    ' serie del sensor de la red de Puebla
equi_sens_acel_ns_eq_sensor                         ' Forma de procedencia: num_consulta
from                                                ' Código de regreso: consulta de datos
T_RED, T_ESTACION, T_EDO_OPERACION_ESTACION,
TA_ESTACION_EQ_ACELEROGRAFICO, T_FABRICANTE,
TA_EQ_ACELEROGRAFICO_EQ_SENSOR, T_TIPO_EQ_SENSOR
where
red_cl_red = est_cl_red and
est_cl_oper = ope_est_cl_oper and
est_cl_est = est_equi_acel_cl_est and
est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
red_cl_red = 'PU'
```

```
create view dat_estacion3 as
select est_nom_est, est_cl_est, est_fech_inst,
 est_tipo_suelo, est_ubicacion, tip_lug_descrip_lugar,
 est_altitud, est_latitud, est_longitud
from T_ESTACION, T_TIPO_LUGAR
 where
 est_cl_tipo_lugar = tip_lug_cl_tipo_lugar
```

```
' La vista muestra datos de la estación la estación
' como la clave de la fecha de instalación,
' el tipo de suelo, la ubicación, tipo de lugar,
' altitud, latitud y longitud
' Forma de procedencia: num_consulta
' Código de regreso: consulta de datos
```

```
create view alim_estacion4 as
select est_cl_est, alim_tipo_alim,alim_torre,
 alim_altura, alim_num_celd, alim_cap_celd, alim_tipo_cable,
 alim_calibre,alim_num_bat, alim_tipo_regu
from T_ESTACION, T_ALIMENTACION
 where
 est_cl_est = alim_cl_est
```

```
' La vista muestra datos de la alimentación de
' la estación como la clave de estación, tipo
' de alimentación, torre, altura, número y capacidad
' de las celdas solares, baterías y tipo de regulador
' Forma de procedencia: num_consulta
' Código de regreso: consulta de datos
```

```
create view acceso_estacion5 as
select est_nom_est, est_cl_est, acc_est_acceso,
 acc_est_encargado, acc_est_vers_cand, acc_est_vers_llave
from T_ESTACION, T_ACCESO_ESTACION
 where
 est_cl_est=acc_est_cl_est
```

```
' La vista muestra datos de acceso a la estación como
' clave de estación, acceso, personal encargado,
' versión del candado y de la llave
' Forma de procedencia: num_consulta
' Código de regreso: consulta de datos
```

```
create view prob_estacion6 as
select est_nom_est, est_cl_est,
 bitac_fech_rev_est, bitac_persona_rev,
 bitac_prob_est, ope_est_descrip
 from
 T_ESTACION, T_BITACORA,
 T_EDO_OPERACION_ESTACION
 where
 est_cl_est = bitac_cl_est and
 est_cl_oper = ope_est_cl_oper and
 bitac_fech_rev_est > '01/01/1997'
```

```
' La vista muestra comentarios de la estación
' como nombre y clave de la estación, fecha
' de revisión, estado de operación, personal de
' revisión y problemas de la estación
' Forma de procedencia: num_consulta
' Código de regreso: consulta de datos
```

```
create view dat_acelerografo7 as
select est_cl_est, fab_mode_eq_acel, equi_acel_ns,
 gana_cl_gana, gana_canal, gana_ganancia,
 equi_acel_fech_rev, equi_acel_mues,
 equi_acel_refe_tiem,equi_acel_prev,
 equi_acel_posev, prop_prop_eq_acel
from
 T_ESTACION, T_EQUIPO_ACELEROGRAFICO,
 T_FABRICANTE, T_PROPIETARIO,
 TA_ESTACION_EQ_ACELEROGRAFICO, T_GANANCIA
where
 est_cl_est = est_equi_acel_cl_est and
 equi_acel_cl_gana=gana_cl_gana and
 est_equi_acel_cl_mode_eq_acel = equi_acel_cl_mode and
 est_equi_acel_ns_eq_acel = equi_acel_ns and
 equi_acel_cl_propie = prop_cl_propie and
 equi_acel_cl_mode = fab_cl_mode_eq_acel and
 equi_acel_fech_rev > '01/01/1997'
```

```
' La vista muestra datos del acelerográfo como
' clave de estación, modelo y número de serie del
' acelerográfo, ganancia por canal, fecha de revisión,
' velocidad de muestreo, referencia de tiempo y
' propietario
' Forma de procedencia: num_consulta
' Código de regreso: consulta de datos
```

```
create view fab_acelerografo8 as                    ' Selecciona datos del fabricante del acelerógrafo
select distinct                                     ' Forma de procedencia:  num_consulta
 fab_mode_eq_acel, equi_acel_ns,                    ' Código de regreso: consulta de datos
 fab_fabricante, tip_acel_descrip, fab_pais
from
 T_EQUIPO_ACELEROGRAFICO,
 T_FABRICANTE, T_TIPO_EQ_ACELEROGRAFICO
Where  equi_acel_cl_mode = fab_cl_mode_eq_acel and
 fab_cl_tipo_eq_acel = tip_acel_cl_tipo_eq_acel


create view prob_acelerografo9 as                   ' Selecciona datos de comentarios de la
select                                              ' revisión a los equipos acelerográficos
 est_equi_acel_cl_est, fab_mode_eq_acel, equi_acel_ns,  ' Forma de procedencia:  num_consulta
ope_equi_acel_fech_rev, ope_equi_acel_descrip       ' Código de regreso: consulta de datos
from
 T_EQUIPO_ACELEROGRAFICO,
 TA_ESTACION_EQ_ACELEROGRAFICO,
 T_FABRICANTE, T_EDO_OPERA_EQ_ACELEROGRAFICO
Where  est_equi_acel_cl_mode_eq_acel = equi_acel_cl_mode and
 est_equi_acel_ns_eq_acel = equi_acel_ns and
 equi_acel_cl_mode = fab_cl_mode_eq_acel and
 equi_acel_fech_rev = ope_equi_acel_fech_rev and
 equi_acel_cl_mode = ope_equi_acel_cl_mode and
 equi_acel_ns = ope_equi_acel_ns and
 equi_acel_fech_rev > '01/01/1997'


create view prob_lectura10 as                       ' Selecciona datos de comentarios de la lectura
 select                                             ' de registros
 est_equi_acel_cl_est, fab_mode_eq_acel, equi_acel_ns,  ' Forma de procedencia: num_consulta
 bitac_lec_fech_rev_lec, bitac_lec_descrip, bitac_lec_persona  ' Código de regreso: consulta de datos
from
 T_EQUIPO_ACELEROGRAFICO,
 TA_ESTACION_EQ_ACELEROGRAFICO,
 T_FABRICANTE, T_BITACORA_LECTURA
where
 est_equi_acel_cl_mode_eq_acel = equi_acel_cl_mode and
 est_equi_acel_ns_eq_acel = equi_acel_ns and
 equi_acel_cl_mode = fab_cl_mode_eq_acel and
 equi_acel_cl_mode = bitac_lec_cl_mode_eq_acel and
 equi_acel_ns = bitac_lec_ns_eq_acel and
 equi_acel_fech_rev = bitac_lec_fech_rev_lec and
 est_equi_acel_cl_mode_eq_acel = bitac_lec_cl_mode_eq_acel and
 est_equi_acel_ns_eq_acel =bitac_lec_ns_eq_acel and
 equi_acel_fech_rev >'01/01/1997'


create view dat_sensor11 as                         ' Selecciona datos del equipo sensor
select                                              ' Forma de procedencia: num_consulta
 equi_sens_mode_eq_sensor, equi_sens_ns_eq_sensor,  'Código de regreso: consulta de datos
 equi_sens_fech_inst_sensor, equi_sens_mode_sensor,
 equi_sens_ns_sensor, sens_fech_camb_rango,
 sens_canal_sensor, sens_rango_sensor, sens_frec_sensor,
 sens_amort_sensor, sens_observaciones, sens_persona_rev
from
 TA_EQUIPO_SENSOR, T_SENSOR
where
 equi_sens_mode_sensor = sens_mode_sensor and
 equi_sens_ns_sensor = sens_ns_sensor
```

```
create view dat_redestacion12 as                    ' Selecciona datos de la estación de una red especifica
select                                              ' Forma de procedencia: num_consulta
 red_nom, est_cl_est, ope_est_descrip,              ' Código de regreso: consulta de datos
 est_ubicacion, est_altitud, est_latitud, est_longitud,
 alim_tipo, alim, acc_est_acceso, bitac_prob_est, bitac_fech_rev_est
from
 T_RED, T_ESTACION, T_ALIMENTACION,
 T_BITACORA, T_EDO_OPERACION_ESTACION,
T_ACCESO_ESTACION
where
 red_cl_red = est_cl_red and
 est_cl_est = bitac_cl_est and
 est_cl_est = alim_cl_est and
 est_cl_est = acc_est_cl_est and
 est_cl_oper = ope_est_cl_oper and
 bitac_fech_rev_est > '01/01/1997'
```

```
create view dat_estacionaceler13 as                 ' Selecciona datos de la estación de una red especifica
select                                              ' Forma de procedencia: num_consulta
 est_cl_est, ope_est_descrip,                       ' Código de regreso: consulta de datos
 alim_tipo_alim, acc_est_acceso,
 fab_mode_eq_acel, equi_acel_ns,
 equi_acel_refe_tiem, equi_acel_prev, equi_acel_posev
from
 T_ESTACION, T_ALIMENTACION, T_ACCESO_ESTACION,
 T_FABRICANTE, T_EQUIPO_ACELEROGRAFICO,
 T_EDO_OPERACION_ESTACION,
TA_ESTACION_EQ_ACELEROGRAFICO
where
 est_cl_est = alim_cl_est and
 est_cl_est = acc_est_cl_est and
 est_cl_est = est_equi_acel_cl_est and
 est_cl_oper = ope_est_cl_oper and
 est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
 est_equi_acel_cl_mode_eq_acel = equi_acel_cl_mode and
 est_equi_acel_ns_eq_acel = equi_acel_ns
```

```
create view dat_acelsensor14 as                     ' Selecciona datos del acelerógrafo y su sensor
select                                              ' Forma de procedencia: num_consulta
 fab_mode_eq_acel, equi_acel_ns,                    ' Código de regreso: consulta de datos
 equi_acel_refe_tiem, equi_acel_prev, equi_acel_posev,
 tip_sens_descrip, equi_sens_acel_cl_mode_eq_acel,
 equi_sens_acel_ns_eq_acel, equi_sens_acel_num_canal,
 equi_acel_fech_rev
from
 T_FABRICANTE, T_EQUIPO_ACELEROGRAFICO,
 T_TIPO_EQ_SENSOR,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR
where
 equi_acel_cl_mode = fab_cl_mode_eq_acel and
 equi_acel_cl_mode = equi_sens_acel_cl_mode_eq_acel and
 equi_acel_ns = equi_sens_acel_ns_eq_acel and
 equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
 fab_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
 equi_acel_fech_rev > '01/01/1997'
```

```
create view dat_estacelsens15 as                    ' Selecciona datos de estación, acelerógrafo y sensor
select                                              ' Forma de procedencia: num_consulta
 est_cl_est, alim_tipo_alim, acc_est_acceso,        ' Código de regreso: consulta de datos
 fab_mode_eq_acel, equi_acel_ns, equi_acel_refe_tiem,
 equi_acel_fech_rev, equi_acel_prev, equi_acel_posev,
 tip_sens_descrip, equi_sens_acel_cl_mode_eq_acel,
 equi_sens_acel_ns_eq_acel, sens_canal_sensor,
 equi_sens_mode_eq_sensor, equi_sens_ns_eq_sensor,
 sens_mode_sensor, sens_ns_sensor, sens_rango_sensor,
 sens_frec_sensor, sens_amort_sensor
from
 T_ESTACION, T_ALIMENTACION, T_ACCESO_ESTACION,
 TA_ESTACION_EQ_ACELEROGRAFICO,
 T_EQUIPO_ACELEROGRAFICO, T_FABRICANTE,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR,
 T_TIPO_EQ_SENSOR, TA_EQUIPO_SENSOR, T_SENSOR
where
 est_cl_est = alim_cl_est and
 est_cl_est = acc_est_cl_est and
 est_cl_est = est_equi_acel_cl_est and
 est_equi_acel_ns_eq_acel = equi_acel_ns and
 est_equi_acel_cl_mode_eq_acel = equi_acel_cl_mode and
 est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
 equi_acel_cl_mode = equi_sens_acel_cl_mode_eq_acel and
 equi_acel_ns = equi_sens_acel_ns_eq_acel and
 equi_sens_acel_cl_tipo_eq_sens = tip_sens_cl_tipo_eq_sens and
 equi_sens_acel_mode_eq_sensor = equi_sens_mode_eq_sensor and
 equi_sens_acel_ns_eq_sensor = equi_sens_ns_eq_sensor and
 equi_sens_mode_sensor = sens_mode_sensor and
 equi_sens_ns_sensor = sens_ns_sensor


create view dat_rangoscarpeta as
select
 fab_mode_eq_acel, equi_sens_acel_ns_eq_acel,
 equi_sens_acel_mode_eq_sensor, equi_sens_acel_ns_eq_sensor,
 sens_canal_sensor, sens_ns_sensor, sens_frec_sensor,
 sens_amort_sensor, equi_sens_acel_fech_reti_eq_s,
 equi_sens_fech_reti_sensor, sens_fech_camb_rango,
 sens_rango_sensor, sens_observaciones
from
 TA_ESTACION_EQ_ACELEROGRAFICO, T_FABRICANTE,
 TA_EQ_ACELEROGRAFICO_EQ_SENSOR,
 TA_EQUIPO_SENSOR, T_SENSOR
where
 est_equi_acel_ns_eq_acel = equi_sens_acel_ns_eq_acel and
 est_equi_acel_cl_mode_eq_acel = equi_sens_acel_cl_mode_eq_acel and
 est_equi_acel_cl_mode_eq_acel = fab_cl_mode_eq_acel and
 equi_sens_acel_mode_eq_sensor = equi_sens_mode_eq_sensor and
 equi_sens_acel_ns_eq_sensor = equi_sens_ns_eq_sensor and
 equi_sens_mode_sensor = sens_mode_sensor and
 equi_sens_ns_sensor = sens_ns_sensor
 group by sens_rango_sensor
```