

38
2y.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

SISTEMA DE AYUDA NACIONAL

T E S I S

Que para obtener el título de:

INGENIERO EN COMPUTACION

p r e s e n t a

CARLOS JIMENEZ GONZALEZ



Director:

Ing. Fernando Ramírez Godoy

Ciudad Universitaria 1998

TESIS CON
FALLA DE ORIGEN

263294



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE AYUDA NACIONAL

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

PRESENTA:

CARLOS JIMENEZ GONZÁLEZ

DIRECTOR:

ING. FERNANDO RAMÍREZ GODOY



CIUDAD UNIVERSITARIA 1998

Agradecimientos

A mis padres *Abel* y *Guadalupe* que con amor y paciencia, me apoyaron y supieron guiarme hacia un camino de madurez . Mi reconocimiento profundo a mi *Madre* por permanecer con entrega y devoción en cada uno de los pasos de mi existencia.

A mi esposa *Edith* quien ha compartido con amor verdadero todas las vicisitudes de mi destino como persona. La solidez de su compañía ha sido manifiesta en el logro de mis metas, obteniendo de ella consejo y apoyo.

A mi hermana *Laura* por su cariño y generosidad, por apoyar mis decisiones y dedicar tiempo a escuchar mis proyectos.

A mi hija *Jessica* que no solo ha sido parte importante de inspiración en este trabajo, sacrificando el valioso tiempo de sus juegos en preguntar el sentido y el porque de la tesis.

A *todas las personas* que de forma directa o indirecta han asistido al logro de mis objetivos, a todos ellos mis agradecimientos.

Dedicatoria

A la *Universidad Nacional Autónoma de México*, recinto de trabajo y formación, donde el crecer y hacer, el saber y empeñarse, el pretender y alcanzar alberga lo mejor de nuestra juventud.

A mis *maestros* cuya amistad, dedicación y apoyo instaura en cada estudiante el imperativo de aprender respondiendo cotidianamente y a ultranza de hacernos profesionales. A ellos gratitud permanente.

La presente tesis no hubiese llegado a feliz término sin el apoyo desinteresado del Ing. Fernando Ramírez Godoy, quien con su asesoría orientó consistentemente el desarrollo, proporcionando observaciones y consejos respecto a diversos problemas técnicos y conceptuales que se afrontaron en el desarrollo de la tesis. Así también es importante anotar aquí la sustancial colaboración de los maestros Ing. Adolfo Millán Nagera, M. en I. Lauro Santiago Cruz, Ing. Sergio Castro y Dra. Ana María Vázquez, quienes fungieron como síndicos durante el examen profesional.

Sistema de Ayuda Nacional.

| | |
|--|-----|
| Introducción | iii |
| Capítulo 1. La Dependencia | 1 |
| 1.1 Objetivos y funciones de la dependencia | 1 |
| 1.2 Problemática actual | 10 |
| 1.3 Necesidades de un sistema de ayuda | 14 |
| Capítulo 2. Infraestructura de Software para el Desarrollo | 20 |
| 2.1 Panorama general | 20 |
| 2.2 Herramientas de desarrollo | 24 |
| 2.3 Herramientas de conectividad | 32 |
| 2.4 El manejador de Base de Datos | 34 |
| 2.5 Selección de la herramienta a emplear | 38 |
| Capítulo 3. Análisis | 39 |
| 3.1 Modelado de datos | 40 |
| 3.2 Modelado de procesos | 45 |
| Capítulo 4. Diseño | 55 |
| 4.1 Estructura de la Base de Datos | 55 |
| 4.2 Diseño del ambiente de la aplicación | 57 |
| 4.3 Diseño del plan de pruebas e implantación | 65 |

| | |
|--------------------------------------|-----|
| Capítulo 5. Construcción | 68 |
| 5.1 Construcción de la Base de Datos | 68 |
| 5.2 Construcción de la aplicación | 72 |
| 5.3 Pruebas de la aplicación | 100 |
| Capítulo 6. Conclusiones | 101 |
| Anexo A manual de usuario | 103 |
| Bibliografía | 115 |

Introducción

El objetivo principal de este trabajo es desarrollar un sistema que apoye las actividades y objetivos del **Centro Nacional de Atención a Usuarios**, para la correcta identificación, atención y manejo de las incidencias o problemáticas operacionales de software y hardware, que se presenten en los centros de cómputo de la Secretaría de Hacienda y Crédito Público (S.H.C.P.), particularmente en la Subsecretaría de Ingresos.

La Secretaría de Hacienda y Crédito Público con la finalidad de cumplir con su actividad sustancial, que es la de recaudar los impuestos interiores y exteriores en la República Mexicana, derivados de los contribuyentes registrados por la misma institución, mantiene una infraestructura en toda la república de centros de cómputo, la cual se agrupa en dos grandes rubros: Recaudación (referente a impuestos interiores) y Aduanas (impuestos gravables de comercio exterior). Los centros de cómputo fungen como responsables de mantener actualizados los datos fiscales de cada contribuyente obteniendo datos de la recaudación mensual, contribuyentes deudores, estimaciones de la Balanza Comercial (Importaciones vs. Exportaciones), etc., como es de suponerse los datos almacenados en los centros de cómputo contienen la información necesaria para un seguimiento detallado de cada contribuyente.

En un entorno local cada centro de cómputo almacena y explota sus datos en él mismo, sin embargo existe la necesidad de realizar explotaciones de información centralizada para realizar cruces de información y constatar la buena fe del contribuyente, otra de las causas de requerir una información centralizada es estimar el crecimiento del país relacionado con el PIB y el nivel de exportación de nuestro país, para cubrir esta necesidad se cuenta con un enlace lógico en topología estrella respecto a un Centro de Procesamiento Nacional (CPN), es decir los centros distribuidos en la República Mexicana están conectados a un CPN el cual funge como almacén central y ruleo de datos entre los centros locales.

En tamaño e importancia la S.H.C.P. ha desarrollado dos aplicaciones para sus centros locales, el SIR (Sistema Integral de Recaudación) y el SAAI (Sistema Automatizado Aduanero e Integral) para la captación de impuestos interiores y exteriores respectivamente, en promedio los centros locales trabajan entre 8 y 10 horas de lunes a domingo, sin embargo por la explotación centralizada de información y otros procesos que intervienen en la operación del centro (respaldos de Base de

Datos, actualización a la programación de las aplicaciones como son adiciones de criterios fiscales y otros), es necesario mantener encendido el equipo todos los días del año las 24 horas del día.

La S.H.C.P. estableció como plataforma tecnológica para basar su estructura computacional en Unix como Sistema Operativo, Informix como Base de Datos (englobando sus herramientas de desarrollo en 4gl de Informix y C de HP-UX), TCP/IP como protocolo de comunicaciones (montando el protocolo X.25 en lugares remotos o donde la infraestructura satelital o RDI es nula) y equipos Unisys y HP. De aquí surge la necesidad de soportar todos los centros en cuanto a su hardware, software y comunicaciones con la finalidad de no interrumpir los procesos en línea o en batch que ofrece cada centro de cómputo.

Por otro lado la administración de problemas es la forma de manejar todo tipo de fallas (incidencias). Su propósito no es sólo el minimizar el impacto de las fallas cuando éstas ocurren sino también, utilizar la información detectada durante la resolución del incidente para la corrección de las causas raíz de la falla y reducir los tiempos que un centro de cómputo se encuentra inoperante, así mismo la información estadística para identificar a los productos de hardware o software con fallas constantes y en su momento tomar decisiones de cambio de plataforma de trabajo.

Para solventar cualquier problema en el centro de cómputo, las medidas que se han tomado actualmente son llamar directamente al área que se considera la involucrada sin embargo es probable que la problemática este relacionada con otras áreas o se deba a otros factores, por lo anterior se complica gravemente la solución de cualquier problema por simple que parezca, para ayudar a la identificación y finalmente a solucionar cualquier problemática, el presente trabajo abarca cuatro aspectos principales:

- Control de incidencias: Es el restaurar el servicio normal cuando algo lo alteró.
- Control de problemas: Obtención de la causa raíz de los incidentes.
- Control de errores: Corrección de los problemas.
- Administración de la Información generada por los conceptos anteriores.

La Administración de Problemas, descrita anteriormente puede clasificarse como un servicio de nivel más especializado que detecta la existencia de un problema mayor de acuerdo a los incidentes que tienen síntomas comunes.

A continuación se hace una breve descripción del contenido de la tesis por capítulo:

En el capítulo uno, se explica el objetivo de la dependencia, cual es su función principal y como surge la necesidad de realizar el Sistema de Ayuda Nacional, así mismo se enumeran las áreas beneficiadas y que tipo de beneficios obtendrán. Finalmente se especifica el alcance del proyecto así como la funcionalidad que deberá tener el sistema enmarcándolo en un Centro de Atención a Usuarios.

El capítulo dos, presenta las diferentes herramientas que pueden ser empleadas en la construcción del proyecto, realizando un estudio de factibilidad para cada herramienta, en este estudio se presentan capacidades y ventajas de cada una, al final de este capítulo se realiza una evaluación y se determina que herramienta será empleada.

En el capítulo tres, se presenta el análisis detallado el cual tiene como propósito establecer y documentar los requerimientos del sistema, identificar entidades, datos y procesos, así como sus relaciones y flujos de información. En esta etapa se establecen los elementos esenciales que permiten elaborar un modelo que representa adecuadamente el funcionamiento de la aplicación y asegura un buen diseño y construcción de la misma. Productos que se obtienen de la fase; modelo entidad-relación, diagramas de contexto y diagrama de flujo de datos.

En el capítulo cuatro, se plantea el diseño, esta fase define las características específicas del sistema de acuerdo a los requerimientos identificados en la fase anterior. Productos que se obtienen de la fase; esquema de la base de datos a emplear, diagrama funcional de la aplicación y miniespecificaciones de los programas.

En el capítulo cinco, se plantean los detalles finales en la construcción de la base de datos así como las pruebas que se realizaron y el proceso de implantación en el centro de cómputo, tomando en cuenta el diseño previo, como productos de esta fase se tienen la construcción de la aplicación, la construcción de la base de datos, la elaboración de las ayudas del usuario y la ejecución de las pruebas a efecto de corroborar el buen funcionamiento del sistema.

El capítulo seis, contiene las conclusiones del proyecto en cuanto a su funcionalidad y los posibles cambios que se puedan realizar para mejorarlo.

El anexo A contiene el manual de usuario del sistema.

I. La Dependencia

1.1 Objetivos y funciones de la dependencia

Con los avances de la electrónica, el procesamiento de datos ha sido utilizado para referirse al área de la informática que trabaja con aplicaciones comerciales tales como el control de un inventario, nóminas, etc., sin embargo en la actualidad con el auge tecnológico en las comunicaciones es posible interconectar en tiempo real centros de cómputo de una manera rápida y eficiente por lo que es posible explotar información contenida en diferentes Bases de Datos, en este entorno la Secretaría de Hacienda y Crédito Público (S.H.C.P.) ha ubicado de manera estratégica centros de cómputo en toda la República Mexicana para llevar a cabo la recaudación de impuestos, como es sabido una de las funciones primordiales de esta Secretaría, es la de captar los impuestos generados por personas físicas y morales. Estos impuestos se agrupan en Interiores y Exteriores, con el fin de tener un panorama de la importancia que representa para la S.H.C.P. el control y captación de impuestos en los siguientes párrafos se esquematiza brevemente los rubros de impuestos exteriores.

En relación a los impuestos exteriores podemos mencionar que las funciones administrativas relativas a la entrada de mercancías al territorio nacional o a la salida del mismo, se realizarán por las autoridades aduaneras y apegándose a los regímenes aduaneros para tal efecto, obligándose a pagar los impuestos originados del uso de los recursos de la Aduana (derechos) y de los aranceles y cuotas compensatorias que se apliquen según sea el caso, por lo que una mercancía puede destinarse a los siguientes regímenes:

Regímenes Aduaneros

Importación

- Importación Definitiva mediante el procedimiento en revisión en origen
- Regularización de Mercancías Internadas en forma Irregular (Importación Definitiva)
- Importación Temporal para retornar en el mismo estado
- Importación Temporal para someterse a un proceso de elaboración, transformación o reparación

Exportación

- Exportación Definitiva
- Exportación Temporal para retornar en el mismo estado
- Exportación Temporal para someterse a un proceso de elaboración, transformación o reparación

Depósito Fiscal

Tránsito de Mercancías

- Tránsito Interno de Mercancías
- Tránsito Interfronterizo de Mercancías
- Tránsito Internacional de Mercancías por Territorio Nacional
- Tránsito Internacional por Territorio Extranjero
- Tránsito para el Transbordo

Elaboración, Reparación o Transformación en Recinto Fiscalizado

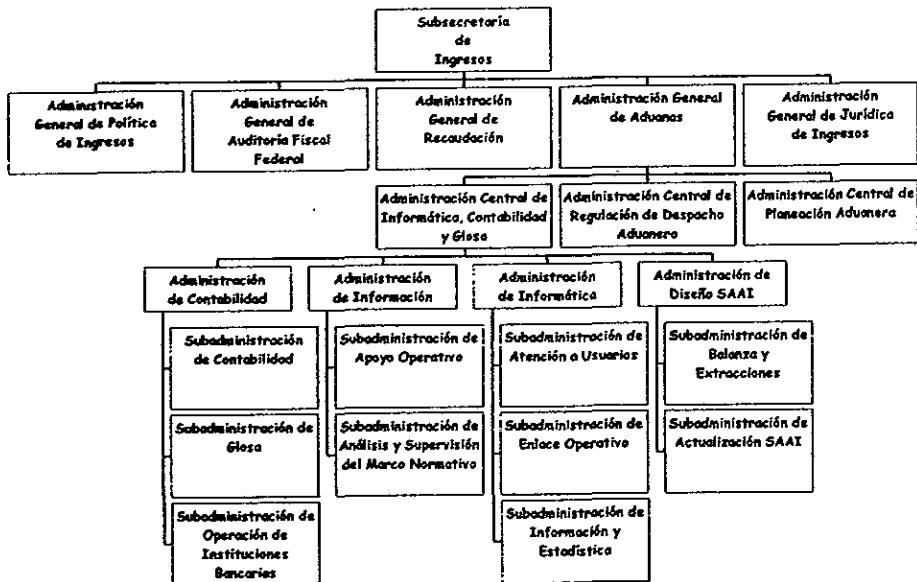
Impuestos que son Pagados:

- ISAN (Impuesto Sobre Automóviles Nuevos)
- IEPS (Impuesto Especial sobre Productos y Servicios)
- IVA (Impuesto al Valor Agregado)
- DTA (Derecho de Trámite Aduanero).
- DTI (Derecho de Tránsito Internacional)
- TIGI (Tarifa de Impuestos Generales de Importación).
- TIGE (Tarifa de Impuestos Generales de Exportación)
- Cuotas Compensatorias

De esta forma una persona (o empresa) que importe calzado deportivo a nuestro país, deberá pagar 20 % del valor en aduana de la mercancía como impuesto TIGI + el DTA + el IVA + sus cuotas compensatorias (si es que las hubiera dependiendo de acuerdos entre países, derivado del origen de las mercancías TLC, TLCAN, etc.).

Con la finalidad de llevar un control de las actividades y tareas en la recaudación de impuestos interiores y exteriores dentro de la S.H.C.P., la Subsecretaría de Ingresos tiene la siguiente estructura organizacional (se detalla únicamente el área de Administración de Informática Contabilidad y Glosa, sin embargo todas las demás áreas mantienen una estructura similar):

ORGANIGRAMA DE LA SUBSECRETARIA DE INGRESOS



Para llevar a cabo la recaudación de los impuestos desde el punto de vista de centros de cómputo la S.H.C.P. ha generado la siguiente infraestructura:

Globalmente se subdivide en:

- Centros de Recaudación
- Centros de Aduanas
- Centros de Administraciones Jurídicas y de Fiscalización.
- Un Centro de Procesamiento Nacional (CPN).

A continuación se enuncian las funciones principales de cada centro de cómputo.

CPN (Centro Procesamiento Nacional)

Por políticas internas la S.H.C.P. centraliza toda su información en el CPN, en este centro se almacenan los movimientos de sus contribuyentes, así como un histórico de cinco años. Por esta centralización solo en el CPN se procesa y valida la información que se envía de todos los demás centros de cómputo una vez procesada es reenviada a su centro de origen para que finalmente llegue al contribuyente.

El CPN Cuenta con los Siguietes Equipos:

| | Aplicación | Descripción |
|---------------------------|---------------------------|---|
| HP 9000 serie 845 y 890 | Desarrollo | Desarrollo y prueba de nuevas versiones o subsistemas. |
| HP 9000 serie 890 | Producción Recaudación | Consulta y procesamiento (batch) de información de Recaudación |
| HP 9000 serie 847 | Producción Aduanas | Consulta y procesamiento de información de Aduanas |
| HP 9000 serie T500 | Producción | Histórico de 5 años información de Recaudación. |
| HP 9000 serie 827 | Respaldo | Máquina de respaldo del equipo 890. |
| Unisys 6000 serie 60 y 65 | Pruebas | Equipos de pruebas en la liberación de aplicaciones para centros locales de Recaudación |

Centros de Recaudación.

Los centros de recaudación tienen la función de recabar los datos fiscales de personas físicas y morales en todas sus modalidades. Para tal efecto están divididos en dos tipos de centros:

Centros Regionales de Recaudación

Se cuenta con ocho Centros Regionales, como enlace y depuración de la información recibida por los centros locales, las regiones fueron definidas de acuerdo a las características y tipo de contribuyentes. Los centros regionales funcionan con equipos HP/9000 serie 835.

Centros Locales de Recaudación

Sesenta y siete centros locales ALR (Administraciones Locales de Recaudación), distribuidos estratégicamente en toda la república, la ALR es la encargada de capturar la información fiscal del contribuyente. Cabe mencionar que no todos los centros locales se encuentran enlazados por red ya sea con su centro regional o el CPN. Equipo que manejan las ALR's U.6000 (UNISYS) series 60 y 65.

Centros de Aduanas.

Estos centros de cómputo son los encargados de almacenar las operaciones de importaciones, exportaciones y tránsito de mercancías en todas sus modalidades, su estructura es la siguiente:

Centros Regionales de Aduanas

Nueve centros regionales de aduanas, conectados al CPN, además de funcionar como una aduana normal deben cumplir con las siguientes tareas.

- Mantener un equipo de respaldo ante contingencias de sus locales.
- Laboratorio de pruebas para instalación del nuevo software y hardware.
- Mantener un estándar en la configuración de comunicaciones respecto a sus locales.

Los centros regionales de aduanas están agrupados de acuerdo a la zona fronteriza y generalmente cuentan con dos tipos de equipo. Un equipo de producción HP 9000 serie G30 o F10 (dependiendo del tamaño de la aduana) y un equipo de pruebas y respaldo HP 9000 serie 807.

Centros Locales de Aduanas

Treinta y cinco aduanas locales con equipo HP/9000 serie 817, 827 y F10.

Los centros aduanales deben operar entre 16 y 20 horas diarias los 365 días del año, en función de las operaciones que son registrados en este lugar.

Es siguiente mapa contiene los Centros regionales del país (aduanas y recaudación)





Recaudación:

| Metropolitano | Noreste |
|-----------------------------------|-----------------------------------|
| | |
| CPM -> Distrito Federal | CD. Obregón -> Sonora |
| Centro -> Distrito Federal | Culiacán -> Sinaloa |
| Norte -> Distrito Federal | Ensenada -> Baja California Norte |
| Sur -> Distrito Federal | Hermosillo -> Sonora |
| Oriente -> Distrito Federal | La Paz -> Baja California Sur |
| Tlalnepantla -> Edo. Mex. | Los Mochis -> Sinaloa |
| Toluca -> Edo. Mex. | Mazatlán -> Sinaloa |
| Mexicali -> Baja California Norte | Nogales -> Sonora |
| Tijuana -> Baja California Norte | |

| Norte Centro | Occidente |
|----------------------------|----------------------------------|
| | |
| Cd. Juárez -> Chihuahua | Guadalajara -> Jalisco |
| Chihuahua -> Chihuahua | Aguascalientes -> Aguascalientes |
| Torreón -> Coahuila | Cd. Guzmán -> Aguascalientes |
| Durango -> Durango | Colima -> Colima |
| Piedras Negras -> Coahuila | Pto. Vallarta -> Jalisco |
| Saltillo -> Coahuila | Tepic -> Nayarit |
| Zacatecas -> Zacatecas | Tlaquepaque -> Jalisco |
| | Zapopan -> Jalisco |

| Centro | Noroeste |
|------------------------------------|-------------------------------------|
| Celaya -> Guanajuato | Monterrey -> Nvo. León |
| Nvo. León -> Guanajuato | Cd. Victoria -> Tamaulipas |
| Incafi -> Querétaro | Guadalupe -> Nvo. León |
| Irapuato -> Guanajuato | Matamoros -> Matamoros |
| Morelia -> Michoacán | Nvo. Laredo -> Tamaulipas |
| Pachuca -> Hidalgo | Reynosa -> Tamaulipas |
| Querétaro -> Querétaro | Sn. Pedro Garza García -> Nvo. León |
| Sn. Luis Potosí -> San Luis Potosí | Tampico -> Tamaulipas |
| Uruapan -> Michoacán | Tuxpan -> Veracruz |

| Golfo Pacífico | Sur |
|---------------------------|-----------------------------|
| Tlaxcala -> Tlaxcala | Oaxaca -> Oaxaca |
| Puebla -> Puebla | Campeche -> Campeche |
| Jalapa -> Veracruz | Cancún -> Quintana Roo |
| Coatzacoalcos -> Veracruz | Chetumal -> Quintana Roo |
| Acapulco -> Guerrero | Mérida -> Yucatán |
| Cuernavaca -> Morelos | Tapachula -> Chiapas |
| Córdoba -> Veracruz | Tuxtla Gutiérrez -> Chiapas |
| Iguala -> Guerrero | Villahermosa -> Tabasco |



Aduanas

| Guadalajara | Matamoros |
|----------------------------------|---------------------------------|
| Guadalajara -> Jalisco | Matamoros -> Tamaulipas |
| La Paz -> B.C. | Cd. Miguel Alemán -> Tamaulipas |
| Mazatlán -> Sinaloa | Reynosa -> Tamaulipas |
| Aguascalientes -> Aguascalientes | Querétaro -> Querétaro |

| Cd. Juárez | Tijuana |
|---------------------------|---------------------------------|
| Cd. Juárez -> Chihuahua | Tijuana -> B.C. |
| Chihuahua -> Chihuahua | Tecate -> B.C. |
| Ojinaga -> Chihuahua | Ensenada -> B.C. |
| Torreón -> Coahuila | Mexicali -> B.C. |
| Pto. Palomas -> Chihuahua | San Luis Río Colorado -> Sonora |
| Sonoyta -> Sonora | |

| Nvo. Laredo | Nogales |
|-----------------------------------|----------------------------|
| Nvo. Laredo -> Tamaulipas | Nogales -> Sonora |
| Colombia -> Nvo. León | Aguaprieta -> Sonora |
| Monterrey -> Nvo. León | Guaymas -> Sonora |
| Cd. Acuña -> Coahuila | Naco -> Sonora |
| Piedras Negras -> Coahuila | |
| Aeropuerto | Veracruz |
| Progreso -> Yucatán | Veracruz -> Veracruz |
| Aeropuerto -> Distrito Federal | Coatzacoalcos -> Veracruz |
| Cancún -> Quintana Roo | Tuxpan -> Veracruz |
| Acapulco -> Guerrero | Altamira -> Tamaulipas |
| Cd. Hidalgo -> Chiapas | Tampico -> Tamaulipas |
| Subteniente López -> Quintana Roo | Cd. del Carmen -> Campeche |
| Lázaro Cárdenas -> Michoacán | |
| Pantaco -> Distrito Federal | |
| Salina Cruz -> Oaxaca | |
| Toluca -> Edo. México | |

CENTROS DE ADMINISTRACIONES JURÍDICAS Y FISCALIZACIÓN

Los centros de Administraciones Jurídicas (ALJ), tienen como objetivo, normar y auditar los casos de contribuyentes que por sus actividades resultan relevantes en cada estado de la república. Equipo que maneja UNISYS serie 6000 /60. El D.F. funge como centro coordinador.

Las estructuras anteriores arrojan aproximadamente 150 centros conectados al CPN los cuales deben ser administrados, con la política de brindar un servicio eficiente al contribuyente.

1.2 Problemática Actual

Operación en el Centro de Cómputo.

La operación normal de cada centro de cómputo depende de la actividad recaudadora o de seguimiento que se presente en la zona, teniendo la siguiente clasificación:

| <i>Tipo de Centro</i> | <i>Horas promedio de operación</i> | <i>Horas críticas (promedio) de operación diarias</i> | <i>Días de operación por semana</i> |
|--|------------------------------------|---|-------------------------------------|
| Área Central CPN | 24 | 16 | 7 |
| Centros Regionales Aduanas | 18 | 10 | 6 |
| Centros Locales Aduanas | 12 | 6 | 6 |
| Centros Regionales Recaudación | 18 | 7 | 6 |
| Centros Locales Recaudación | 9 | 5 | 5 |
| Centros de Jurídica y Fiscalización | 10 | 5 | 5 |

De la tabla anterior se concluye que todos los centros deben operar como mínimo 10 horas diarias, durante cinco días a la semana, esto significa que los problemas derivados por software o hardware se deben minimizar en lo posible, es importante hacer notar que los centros de cómputo de

aduanas como Nuevo Laredo, Tijuana, Mexicali, Cd. Juárez, Aeropuerto de la Cd. de México, Veracruz y Cd. Hidalgo por la importancia en el despacho de mercancías de procedencia extranjera que ingresan o salen del país (percederos, aparatos electrónicos animales vivos o despojos, etc.), es de suma importancia que en horario normal (de 9 a 20 hrs) la operación de la aduana no se afecte por problemas en el equipo de cómputo el cual en concreto debe registrar la operación de despacho y controlar los procesos de revisión que se llevan a cabo en la misma aduana, para dar un ejemplo de las pérdidas que esto puede llegar a representar para el país; así como empresas extranjeras y nacionales, en 1995 en la Aduana de Nuevo Laredo se registro una contingencia en el manejador de Base de Datos (*datos corruptos*), deteniendo la operación del sistema por 12 días, iniciando en operación el sistema alterno del despacho de mercancías, según datos de la S.H.C.P. se dejaron de recaudar impuestos por 1 millón de dólares y las pérdidas reportadas en percederos y mercancías que por su naturaleza pierden su características sustantivas después de un determinado periodo ascendieron a más de 2.5 millones de dólares, así mismo se tuvo sospechas de la entrada de mercancía ilegal (armamento o electrónicos registrados en otra clasificación arancelaria comúnmente llamado contrabando) en virtud de no contar con el seguimiento detallado en la revisión de las mercancías. Ejemplos como el anterior tienen impactos desastrosos en la economía de un país, siempre que su economía se basó en la recaudación de impuestos (como es el caso de nuestro país), si la problemática solo se concentrara en la Base de Datos, se puede establecer un camino alterno de contingencia para evitar cualquier problema, desafortunadamente en un centro de cómputo se presentan infinidad de problemas con igual variedad en su origen, a lo cual se debe contar con un esquema que proporcione un servicio de soporte eficiente.

La Base de Datos

En términos de Base de Datos la S.H.C.P. decidió adquirir como manejador institucional para el almacenamiento de su información a Informix Inc., respecto de los centros de cómputo podemos decir que el CPN en sus equipos mantiene la estructura única que será manejada por todos los demás centros computacionales, de tal forma que las aplicaciones son realizadas en el CPN y posteriormente liberadas a todos los demás centros de cómputo. El CPN cuenta con una aplicación de consulta a los datos de contribuyentes a nivel nacional (centros enlazados). Para dar un panorama de la importancia a continuación se enumeran las bases de datos mas importantes almacenadas en CPN:

| <i>Nombre de la Base de Datos</i> | <i>Espacio en Disco</i> | <i>Equipo Residente</i> | <i>Arrea General</i> |
|-----------------------------------|-------------------------|-------------------------|----------------------|
| Hacienda | 120 GB (aprox.) | HP 9000/890 | Recaudación |
| Miga | 10 GB " " | HP 9000/857 | Jurídicas |
| Padrón | 5 GB " " | HP 9000/890 | Padrón Vehicular |
| Aduanas | 140 GB " " | HP 9000/847 | Aduanas |

A nivel local o regional cada centro de cómputo cuenta con su parte de datos relacionados con él mismo, así también cuentan con una licencia de Informix para el almacenamiento de la información.

Esquema actual en la resolución de problemas:

En el momento en que se suscita un problema en cualquier centro de cómputo (sea local o regional), referente a hardware o software este intenta localizar quien puede ayudarle a solventarlo, independientemente de si es el canal correcto, esto origina las siguientes situaciones:

- El problema se reporta a muchas áreas al mismo tiempo (supuesto de solución rápida), originando que el área involucrada en la solución pierda el enfoque y el tiempo de solución sea muy grande.
- Del punto anterior se deriva que un área tome las funciones de soporte que no le corresponden (ejemplo "áreas de desarrollo de aplicaciones se convierten en soporte de aplicaciones")
- El problema puede tener un origen distinto del cual se esta presentando (ejemplo "Impresora no funciona correctamente" se reporta a mantenimiento de hardware y realmente no esta configurada correctamente en sistema operativo).
- Los centros regionales pierden la función de soporte a sus centros locales.
- Siempre que existen problemas graves, todas las áreas involucradas tratan de deslindar su responsabilidad.
- El control de problemas se realiza a través de oficios con copia a todos los posibles involucrados generando ineficiencia en el control de los reportes originados en un centro de cómputo.

- Cada área dedicada al soporte técnico debe contar con los recursos humanos necesarios para poder atender las llamadas telefónicas (se pierde mucho tiempo al contestar el teléfono e identificar si es un problema que debe resolver).
- Existe un desconocimiento de los volúmenes de problemas que son reportadas por cada centro y a nivel nacional.
- Cuando un área solicita se le apoye en la solución de algún problema no se tiene definida la responsabilidad de las áreas de soporte.

Ventajas que se tienen al reportar a las áreas centrales (o relacionadas directamente con el problema) :

- El reporte es solventado rápidamente, siempre que el área a la cual se le reporta sea la directamente involucrada (no existen triangulaciones en la solución de los reportes).
- Las áreas llevan un control local de sus reportes.

El siguiente cuadro describe las áreas que más intervienen en la solución de los reportes:

| <i>Problemas Tipo</i> | <i>Area que recibe el reporte</i> |
|--|---|
| Comunicaciones | Gerencia de Comunicaciones |
| Sistema Operativo | Gerencia de Apoyo Técnico |
| Manejador de Base de Datos | Gerencia de Apoyo Técnico |
| Productos de Automatización de Oficinas | Gerencia de Tecnología |
| Hardware | Gerencia de Infraestructura |
| Aplicaciones de Aduanas | Gerencia de Desarrollo para Aduanas |
| Aplicaciones de Recaudación | Gerencia de Desarrollo para Recaudación |
| Otras Aplicaciones | Gerencia de Sistemas Institucionales |

1.3 Necesidades de un sistema de ayuda

La asistencia telefónica es el medio básico y más importante de contacto entre un centro de cómputo y las áreas de soporte técnico, este servicio permite resolver dudas relacionadas con el funcionamiento de aplicaciones y hardware, aclarar situaciones confusas encontradas dentro de la documentación de usuario y obtener soluciones a problemas de mal funcionamiento que afecten la operación del centro de cómputo. Por lo anterior es necesario contar con un sistema de apoyo automatizado que permita registrar y dar seguimiento a todas las solicitudes de atención recibidas por los diferentes centros de cómputo, con políticas y procedimientos de atención bien definidos, asegurando que los reportes serán resueltos con la prontitud que se requiere. Para llevar un control de los requerimientos de soporte técnico solicitados por los diferentes centros de cómputo, éstos deberán ser registrados en un sistema computarizado. Cada requerimiento origina la creación de una Solicitud de Asistencia (SA) a la cual se le asigna un número de identificación. Se deberá establecer un formato único que ayude en el levantamiento de los reportes. Por otra parte el sistema debe permitir agilizar el registro de los reportes por medio de los datos asociados a la clave de usuario que reporta. Los datos asociados al contacto técnico de cada centro de cómputo deberán ser nombre, dirección, número de teléfono, número de red satelital, número de fax y tipo de contacto (Soporte Técnico, Coordinador de Area, Gerente Regional, etc.). El horario de atención en la recepción de solicitudes de asistencia deberá ser las 24 horas de todos los días de la semana, incluso días de descanso establecidos por la Ley de Federal del Trabajo.

Políticas de Atención de Solicitudes de Asistencia:

Información que se deberá requerir para levantar una Solicitud de Asistencia

Antes de que cualquier área de Soporte Técnico pueda comenzar a trabajar en cualquier problema, se debe conocer la naturaleza y localización del mismo. Cuando se levante una solicitud de asistencia se deberán proporcionar la siguiente información:

- Número de identificación de usuario.
- Centro de cómputo del cual se llama.
- Número telefónico, RDI y satelital.
- Sistema operativo y versión en la cual se encuentra el problema.
- Producto y componentes con su versión.
- Número de error, si apareció alguno.

- Descripción breve y detallada del problema.
- Severidad del problema.

Contactos Técnicos.

El sistema debe contemplar la alta de usuarios por centro de cómputo autorizados para levantar solicitudes de asistencia técnica, de tal forma que sean los Contactos Técnicos, estos deben ser los únicos responsables de canalizar las solicitudes de asistencia técnica. Los Contactos Técnicos deben ser personas con amplios conocimientos de la operativa diaria del centro de cómputo, de la administración de la base de datos y con fundamentos sólidos en el sistema operativo donde se opere, lo anterior con la finalidad de agilizar la identificación del problema y en su caso la aplicación de una solución idónea. El contacto técnico deberá comunicarse telefónicamente a una Mesa de Ayuda la cual contará con 3 líneas telefónicas (Lada, RDI y Satelital en total 9 canales de comunicación y una línea lada 800, este servicio será exclusivamente para tratar asuntos relacionados con la atención de Solicitudes de Asistencia).

Niveles de Severidad

De común acuerdo entre el personal que levanta el reporte y el personal de la Mesa de Ayuda, se asignará la severidad al reporte Mesa de Ayuda deberá clasificar los problemas de acuerdo a la manera en que impactan al centro de cómputo. La tabla que aparece a continuación lista los Niveles de Severidad y las metas en tiempo de resolución a los problemas. Si la Solicitud de Asistencia requiere que algún producto de software sea modificado, la resolución final queda determinada por el tiempo que le tome a la gente de desarrollo en distribuir el "parche" que solucionará el problema. La respuesta de las áreas de Soporte Técnico deberán comenzar cuando se envíe toda la información necesaria para identificar el problema.

SEVERIDAD**IMPACTO****META DE
RESOLUCIÓN**

| | | |
|--------------------|---|---------|
| Severidad 1 | Impacto crítico. El sistema de producción de el usuario se encuentra caído o ha experimentado pérdida de datos. Se trabaja ininterrumpidamente con el usuario | 12 hrs. |
| Severidad 2 | Impacto importante. Características importantes del sistema de el usuario no están operables, no existe una solución alterna razonable. El sistema se encuentra operando, pero de manera restringida. | 3 días |
| Severidad 3 | Impacto pequeño. Preguntas referentes a características del producto y funcionalidad; consultas sobre documentación no muy clara; características importantes inoperables, pero con una solución alterna o características no tan importantes inoperables sin solución alterna razonable. | 5 días |

La solución a una SA (Solicitud de Asistencia) puede incluir una respuesta por escrito, cinta de parche, documentación suplementaria, una solución alterna al problema mientras se espera un nuevo nivel de mantenimiento u otro tipo de ayuda.

En caso de que la solución proporcionada por las áreas de soporte técnico no cumplan estrictamente con la solución del reporte, el usuario tendrá derecho a escalar su problema de acuerdo a los lineamientos que se detallan posteriormente.

Escalamiento de Problemas.

Se han definido una lista de metas relativas a la resolución de una Solicitud de Asistencia (SA). La resolución de una SA queda establecida en el momento de cerrarla, cuando ésta no está relacionada con otro SA o hace referencia a un error conocido del cual se estima sea resuelto en un tiempo relativamente corto.

Las severidades y metas de resolución de una SA quedan descritas en la sección de niveles de severidad, de tal forma que el usuario puede utilizar estos procedimientos de escalamiento cuando no se cumplan los tiempos de resolución establecidos como metas.

Los tres niveles de escalamiento son:

| | |
|--|-------------------|
| Escalamiento al Coordinador de Soporte Técnico de las Áreas Involucradas | Atención ESPECIAL |
| Escalamiento al Gerente Responsable del Área | Atención SERIA |
| Escalamiento al Director de Área. | Atención CRÍTICA |
| Escalamiento al Coordinador de Soporte Técnico | |

El usuario deberá iniciar un escalamiento al Coordinador de Soporte Técnico (atención especial) cuando considere que su SA no ha sido resuelta de acuerdo a los lineamientos de resolución.

El usuario deberá llamar y pedir hablar con el gerente de Mesa de Ayuda para iniciar el escalamiento. El Coordinador de Soporte Técnico de el área que tenga asignado el reporte deberá permanecer como responsable del problema hasta que se resuelva y notificará al cliente de su progreso diariamente.

Escalamiento al Gerente.

El usuario puede iniciar un escalamiento al gerente (atención seria) cuando se presente alguna de las siguientes situaciones:

- No se han cumplido los lineamientos establecidos para la SA de prioridad uno.
- Se tienen varias SA que están afectando seriamente la operación del centro de cómputo.
- No se obtuvo una respuesta satisfactoria con el escalamiento al Coordinador de Soporte Técnico.

El usuario deberá contactar al gerente de Mesa de Ayuda, el director de área se informará de la situación de la problemática. Todas las acciones desde el momento en que se solicita un escalamiento, hasta su solución y cierre, se registran en la SA

Escalamiento al Director

El escalamiento al director ocurre si el usuario percibe que el problema deteriora en gran medida la operación del centro de cómputo y los tiempos estimados de solución no se han cumplido, otra causa deberá ser que los niveles previos de escalamiento han fallado para remediar la situación.

Los procedimientos anteriores existen para situaciones especiales que requieren de una intervención directa e inmediata de los niveles directivos de la empresa y se reservan especialmente para esa ocasión

Cierre de una SA.

1. Una SA puede ser cerrada cuando se ha proporcionado la solución completa y el usuario no esta en condiciones de probarla en ese momento. Después de probar la solución el usuario puede reabrir la SA.
2. Una SA se cierra definitivamente cuando el usuario probó la solución dada y está de acuerdo en cerrarla.
3. Una SA se cierra condicionalmente cuando el usuario no ha podido ser localizado en un término de tres días, a partir de que se abrió la SA, o de la última acción registrada. El usuario puede reabrir la SA en caso de que lo considere necesario.

Procedimiento de atención de Solicitudes de Asistencia

Inmediatamente después de que sus datos son registrados, automáticamente, el sistema le asigna un número de SA a su requerimiento de soporte. Este número deberá conservarlo, ya que le será pedido cada vez que llame para conocer el avance de su solicitud. Además del número de SA se le informará el nombre del ingeniero de Soporte Técnico que lo atenderá. Sin embargo para evitar conflictos en la comunicación respecto de cada reporte levantado, únicamente lo que se encuentre capturado en la Solicitud de Asistencia será la respuesta oficial a su reporte, de tal forma que la Mesa de Ayuda será el único canal de comunicación entre el centro de cómputo (usuario que reporta) y el ingeniero de Soporte Técnico que atiende el reporte, de esta forma no se permite tener diferentes versiones del reporte y por otro lado cada área se aboca a realizar una actividad a la vez.

Asistencia Técnica Remota Vía la RED de la SSI (Subsecretaría de Ingresos)

En la atención de un problema complejo donde se requiere hacer una investigación más a fondo o reproducir el problema la conexión vía red al centro de cómputo, es de suma importancia para llegar a la solución lo antes posible. La conexión por red deberá permitir también la transferencia de archivos de trace y vaciados de memoria para diagnosticar la falla. Es importante aclarar que el acceso a los diferentes centros de cómputo solo podrá ser realizada por la Gerencia de Apoyo Técnico la cual es responsable del enlace y las tareas que llegaran a realizarse en el centro de cómputo, garantizando el menor impacto en el mismo.

Asistencia Técnica Telefónica 7x24

Este servicio será proporcionado por la Mesa de Ayuda las 24 horas del día los 7 días de la semana. Para poder aprovechar este servicio es necesario que se dote de localizadores a las personas asignadas al primer nivel de soporte.

A continuación se enuncian los alcances y tareas principales que deberá cubrir el sistema.

Deberá almacenar todos los reportes generados en los diferentes centros de cómputo e identificarlos como únicos en la Base de Datos respecto a otros centros de cómputo u otros reportes, para lo cual deberá identificar los datos de la persona que reporta, los datos del centro de cómputo, una descripción general del problema y una descripción detallada del mismo.

El reporte será asignado lo antes posible a una sola área para su solventación de acuerdo a sus características, sin embargo un mismo reporte se reasignará a otra área que pudiera proporcionar la solución al problema, de tal forma que exista un histórico de las actividades que ha desempeñado cada área en la solución del reporte, teniendo identificada una sola área como responsable del reporte.

En cualquier momento se debe conocer el estado del reporte, por lo que un reporte se encuentra abierto, cancelado, pendiente o cerrado, el sistema debe tener la capacidad de emitir reportes y estadísticos diarios, semanales y mensuales de los siguientes rubros:

- Reportes asignados.
- Reportes pendientes.
- Reportes con estado de terminado
- Reportes reasignados
- Comportamiento de un área o conjunto de ellas
- Comportamiento de una aplicación o software de uso general
- Comportamiento de un hardware particular.

II. Infraestructura de Software para el Desarrollo

2.1 Panorama General

Actualmente para el desarrollo de un sistema, se tiene una variedad amplia de herramientas de software, por lo que es necesario realizar un breve estudio de las posibles herramientas a emplear, inicialmente se identifican cuatro rubros:

- El Manejador de Base de Datos.
- Herramientas de desarrollo.
- Herramientas de presentación
- Herramientas de conectividad.

Es importante tomar en cuenta que actualmente la Secretaría de Hacienda y Crédito Público, tiene como plataforma institucional en la Subsecretaría de Ingresos el manejador Informix – OnLine y como herramienta de desarrollo Informix – 4GL y C embebido, sin embargo este esquema no excluye la posibilidad de implantar una solución Cliente Servidor Multicapa. Respecto de la conectividad, el enfoque que tendrá este análisis es dar una alternativa de interconexión entre herramienta de desarrollo, herramienta de presentación y manejador de Base de Datos.

Las siguientes secciones de este apartado, harán referencia de los conceptos y características estándar de cada herramienta.

Bases de Datos.

Una base de datos como definición estándar es una colección de información organizada que cumple con un fin común. La anterior definición es extraída de un texto como definición general, sin embargo serán enunciadas las reglas de Codd que se consideran parte fundamental con lo que debe contar un manejador de Base de Datos Relacional:

| <i>Regla</i> | <i>Descripción</i> |
|--------------|---|
| 1 | Concepto formal de una Base de Datos relacional "es la conjunción de información visible al usuario, organizada estrictamente como tablas de valores, y en donde todas las operaciones respecto de la información se da sobre tablas. |
| 2 | Refuerza la importancia de las claves primarias para localizar datos en la Base de Datos. |
| 3 | Recomienda la existencia de valores aun cuando la información no se encuentre presente (valores null). |
| 4 | Una Base de Datos debe ser autodescriptiva, es decir se debe contar con un sistema de tablas que describa la propia estructura de la Base de Datos. |
| 5 | Ordena la utilización de un lenguaje de Base de Datos relacional, capaz de soportar las funciones básicas del DBMS (creación, consulta, actualización e inserción de datos). |
| 6 | Trata del manejo de vistas (tablas virtuales). |
| 7 | Refuerza la naturaleza orientada a conjuntos de una Base de Datos relacional, requiere que las filas sean tratadas como conjuntos en operaciones de inserción y supresión. |
| 10 | Dice que el lenguaje de Base de Datos debe ser capaz de soportar restricciones de integridad que restringen la inserción de datos. |
| 11 | Dice que el lenguaje de Base de Datos debe ser capaz de manipular datos distribuidos. |
| 12 | Impide que la actualización de la estructura relacional y su integridad, por un lenguaje de bajo nivel. |

De forma genérica podemos decir que un manejador de Base de Datos esta compuesto de los siguientes módulos:

| |
|---|
| Lenguaje de manipulación de datos. |
| Lenguaje de control de datos. |
| Lenguaje de definición de la base de datos. |
| Lenguajes sin procedimientos para realizar consultas |
| Rutinas para organización y recuperación de la base de datos. |
| Rutinas internas de manejo de datos. |

Tipos de Bases de Datos.

En la actualidad, existen en el mercado diferentes tipos de Sistemas Administradores de Bases de Datos que se han desarrollado para medios magnéticos que van desde las grandes computadoras hasta los equipos personales de cómputo y que de acuerdo a la estructura de los datos y al mecanismo de acceso tienen los siguientes enfoques:

| | |
|---------------------|--|
| Gestión de Archivos | Basicamente no tiene un modelo de datos, por lo que el conocimiento del contenido de un archivo estaba incorporado a los programas de aplicación, de esta forma una aplicación de nómina contenía sus propios archivos, distintos de los archivos de contabilidad, etc. las aplicaciones eran verdaderos laberintos de directorios y archivos con la desventaja de que si algún archivo se dañaba el proveedor de la aplicación era la única persona que podía brindar la ayuda. |
| Jerárquico | En este modelo cada registro de la Base de Datos representa una pieza específica, los registros tenían relación padre/hijo que ligaba cada pieza a su subpieza y así sucesivamente. La recuperación de datos se daba por la navegación de padre a hijo y moviéndose hacia los lados, registro por registro. Este modelo fue creado en un esquema de línea de producción (o ensamblado de productos), en donde cada parte contenía una subparte, sin embargo esto se complica cuando se tiene la necesidad de que un mismo nodo interactue mas de una vez en el mismo instante con otro nodo. |
| Red | Con la idea de solventar las carencias del modelo jerárquico se creo el modelo de red en el cual un registro participará en múltiples relaciones padre/hijo, en este modelo las carencias se derivaron de la falta de flexibilidad, rendimiento y normalización. |
| Relacional | Las desventajas de los modelos jerárquicos y de red condujeron a un intenso interés en el nuevo modelo de datos relacional, cuando fue escrito por primera vez por el Dr. Codd. El modelo relacional era un intento por simplificar la estructura de la Base de Datos, eliminaba las estructuras explícitas padre/hijo y en su lugar representaba los datos en forma de fila/columna de valores de datos, (se describió con anterioridad las características que debe tener un manejador de Base de Datos Relacional). |

A continuación se enuncian dos manejadores de Base de Datos relacionales, considerados como los más importantes en el mercado:

INFORMIX-OnLine es un manejador de Base de Datos que soporta no solo el almacenamiento de datos. Permite agregar, modificar, consultar y borrar información de una tabla en una forma iterativa. Utiliza un SQL (Structured Query Lenguaje) y QBE (Query By Example) como lenguaje de programación. Ejecuta consultas desde una pantalla o en forma iterativa. Portabilidad de programas en diferentes ambientes.

ORACLE se centra en el compromiso de ofrecer innovaciones técnicas comprometer la calidad de 195 productos. Como resultado, Oracle ofrece las mejores herramientas en el mercado para la administración de negocios. Conforme las empresas van evolucionando, pueden seguir contando con las nuevas necesidades que van surgiendo. La tecnología de Oracle está diseñada en su totalidad para adaptarse rápidamente dentro de los ambientes heterogéneos de la computación. Oracle ofrece soluciones basadas en multiprocesamiento simétrico, sistemas abiertos, plataformas masivamente paralelas. Por ser abierta y portable la tecnología de Oracle se mantiene al ritmo con

la evolución de la computación. Oracle 7 es el primer sistema de administración de bases de datos que maneja múltiples fuentes de datos de manera transparente, sin requerir de programación adicional Oracle 7 aumenta la productividad de los programadores, soporta bases de datos distribuidas y ofrece escalabilidad en sus aplicaciones.

Todos los productos Oracle 7 ofrecen portabilidad total, corriendo en todas las plataformas principales, desde microcomputadores personales, mainframes y sistemas masivamente paralelos. Oracle 7 permite adecuar el tamaño de los sistemas sin comprometer las aplicaciones o la integridad de los datos y gracias a que Oracle 7 es abierto, las aplicaciones pueden acceder a datos distribuidos que no sean de Oracle, tales como datos almacenados en DB2 y/o Rdb, de igual manera que si provinieran de una sola fuente. Oracle 7 garantiza la integridad de los datos ya que posee características tales como commit de dos fases (two-phase commit), integridad referencial y procedimientos almacenados.

El Cooperative Development Environment (CDE), o ambiente de desarrollo cooperativo de Oracle, es el principal ambiente de desarrollo para la creación de aplicaciones cliente servidor de misión crítica. Oracle Applications posee la misma portabilidad a través de los sistemas de cómputo, las aplicaciones pueden cambiar de un ambiente a otro de forma fácil, entre ambientes de Pc's, estaciones de trabajo, minicomputadoras y mainframes, tanto en sistemas propietarios o de cliente-servidor, Oracle Applications hereda cualquier adaptación a la medida de los sistemas existentes y garantiza la capacidad de hacer modificaciones futuras. Conforme se va incrementado el potencial en del hardware, Oracle applications permite adaptar las aplicaciones a las nuevas necesidades sin requerir de programación especial. Oracle Applications está integrado y es distribuido para lograr una administración más eficiente de los datos y manejar la carga de trabajo. Por ejemplo, Oracle General Ledger puede correr en un mainframe, mientras Oracle Accounts Payable corre en un minicomputador y Oracle Accounts Receivable corre en una red de minicomputadoras, con la ventaja de compartir los mismos datos.

Las siguientes secciones explican en detalle las ventajas y desventajas que ofrecen los productos de Informix y Oracle, serán explicadas iniciando con Informix.

2.2. Herramientas de Desarrollo

Herramientas Informix.

INFORMIX-SQL es una herramienta para el desarrollo de aplicaciones con características de rapidez, poder y seguridad además de ser un producto versátil y puede ser usado sin un conocimiento profundo de Base de Datos. INFORMIX-SQL contiene cinco aplicaciones que son; un generador de formas, un generador de reportes, un editor interactivo SQL, un constructor de tablas y un constructor de menús y de esta forma realizar aplicaciones completas.

INFORMIX-SQL es completamente portable en el ambiente de hardware y software de aproximadamente 450 plataformas incluyendo (UNIX, XENIX, DOS y OS/2).

Menú Principal:

```

INFORMIX-SQL: Form Report Query-Language User-menu Database Table Exit
Run, Modify, Create, or Drop a form.

----- Press CTRL-W for Help -----
    
```

Especificaciones de Informix-SQL

Especificaciones de Memoria

| | |
|--|--------------------------|
| Memoria en modo multiusuario (1-10 usuarios) | 1MB-5MB |
| Máximo por usuario | 5MB+ |
| Ambiente de Desarrollo(ISQL) | 520K+ espacio del engine |
| Engine INFORMIX-SE | 700K |
| Engine INFORMIX-OnLine | 1000K |
| Espacio en Disco | 5.1MB |
| Versión Runtime | 3.3MB |

Especificaciones de Reportes

| | |
|----------------------------|---|
| (especificaciones máximas) | |
| Longitud máxima de línea | sin limite lógico* |
| Columnas en un reporte | sin limite lógico* |
| Variables de usuario | 100 |
| Columnas de ORDER BY | Arriba de 120 bytes con INFORMIX-SE |
| Columnas de ORDER BY | Arriba de 255 bytes con INFORMIX-OnLine |
| Niveles de break | Arriba de 8 |

Especificaciones para Formas

| | |
|--------------------------------------|-------------------------|
| Número de Formas | sin límite* |
| Número máximo de pantallas por forma | 20 |
| Número de columnas por forma | Dependiente del equipo. |

* Depende del direccionamiento de la memoria que emplee el equipo.

VENTAJAS DE I-SQL

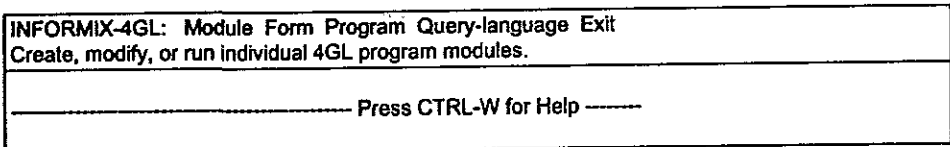
- Desarrollo de prototipos, como producto agregado Informix permite la facilidad de tomar el prototipo como base en la programación que será definitiva.
- Empleo de un editor SQL (interactivo) para administración de la Base de datos.
- Actualización directa a la base de datos con instrucciones SQL.
- Permite la actualización de bases de datos remotas.

Lenguajes 4GL

Como lenguaje para el desarrollo de aplicaciones, INFORMIX proporciona un 4GL interactivo, con este lenguaje se pueden desarrollar sistemas a la medida de las necesidades del usuario, para facilitar la tarea de programación INFORMIX proporciona:

- 4GL /RDS (4GL Rapid Development System) con la finalidad de minimizar los tiempos de compilación Informix proporciona un lenguaje de cuarta generación Interprete de código, siendo fácil la verificación y depuración de programas.
- 4GL/Compilador, como su nombre lo indica liga las librerías necesarias para dejar un código ejecutable, es posible migrar este código a diferentes plataformas UNIX, con la salvedad de recompilar la aplicación. Cabe mencionar que existen algunas diferencias entre RDS y Compilador por lo que requiere una revisión el código que sea migrado en ambos sentidos (de RDS a Compilador y viceversa)
- I-4GL/Debugger en combinación con 4GL/RDS, puede depurar o actualizar aplicaciones complejas, rastreando y visualizando los resultados de cada línea de código.

La siguiente figura muestra el menú general para 4GL



Especificaciones generales para 4GL

Requisitos de Hardware

| | |
|---------------------------------|-------------------------|
| Ambiente de programación (r4gl) | 320K+ Manejador de B.D. |
| Memoria para fglo(runner) | 350K+Manejador de B.D. |

Manejador de Base de Datos

| | |
|------------------|-------|
| INFORMIX-SE | 700K |
| INFORMIX-OnLine | 1000K |
| Espacio en Disco | 8.6MB |
| Versión Runtime | 4.4MB |

Entorno de Reportes

(especificaciones máximas)

| | |
|--------------------------------|--|
| Longitud de la línea de salida | Sin límite + |
| Columnas en el reporte | Sin límite |
| Variables | Dependiente del equipo + |
| Ordenamiento de columnas | Arriba de 120 bytes en INFORMIX-SE Arriba de 255 bytes en INFORMIX-OnLine |
| Numero de break | Arriba de 8 |

Entorno de Ventanas y Formas

(especificaciones máximas)

| | |
|------------------------------|--------------------------|
| Numero de formas | Sin límite + |
| Número de ventanas por forma | 1 |
| Número de columnas por forma | Dependiente del equipo + |

Entorno de Colores y Atributos

Colores

Blanco, Amarillo, Magenta, Rojo, Cyan, Verde, Azul y Negro

Atributos

Normal, Underline, Bold, Dim, Reverse, Blink, Invisible

(El color y atributos son dependientes del tipo de terminar y termcap que sea empleado).

Nota 4GL Compilador requiere del compilador de C para el sistema operativo en que sea instalado.

+ Dependiente del direccionamiento en memoria

VENTAJAS DEL EMPLEO DE UN LENGUAJE DE CUARTA GENERACIÓN

- Empleo de cursores dinámicos, sobre la Base de Datos.
- Empleo de ventanas y formas para el desplegado o captura de datos
- Programación de ayudas en línea.
- Actualizaciones y consultas directas a la base de datos
- Manipulación de variables *date-time*, *Blob's*, *char's* (dinámicos), interval (intervalos de tiempo), etc.
- Generación de código "C"
- Facilidad en el llamado de funciones "C".
- Actualización de Bases de Datos Distribuidas a través de sentencias SQL abriendo de forma remota una Base de Datos a la vez (con la condición de que el ambiente y productos se encuentren instalados).

I-ESQL/C y Cobol (SQL Incrustado)

En virtud de que muchas aplicaciones existentes fueron desarrolladas en otras plataformas, INFORMIX proporciona interfaces con C, Cobol, Fortran, Pascal y Ada para ser migrados paulatinamente a Informix o agregar módulos que deben compartir información en diferentes plataformas. Un ejemplo de diferentes plataformas, se tiene en la S.H.C.P., las aplicaciones de la Contaduría Mayor de Hacienda están bajo ambiente Oracle, por otra parte el manejador institucional en los centros Regionales y Locales de la Subsecretaría de Ingresos es Informix, por lo que para compartir información entre estos dos manejadores de Base de Datos se emplea "C".

Especificaciones para INFORMIX-ESQL 6.0 UNIX



| | |
|------------------|--|
| RAM | arriba de 650 kilobytes por aplicación |
| Espacio en Disco | 1 megabyte |



| | |
|------------|---|
| ESQL/C | UNIX C (con soporte para formatos binarios) |
| ESQL/COBOL | Micro Focus o Ryan-McFarland (Liant) |



| | | |
|------------|-----------|-------|
| Byte* | Character | Date |
| Datetime | Decimal | Float |
| Integer | Interval | Money |
| Smallfloat | Smallint | Text* |
| Varchar* | | |

* Tipos exclusivos para Informix On_Line

Ventajas del empleo de SQL incrustado

- Accesos a la base de datos por instrucciones directas (proporcionadas por ESQL/X)
- Facilidad en el uso de las instrucciones o variables de la herramienta.
- Adecuaciones modulares a sistemas que se pretendan migrar a Informix y los cuales se encuentren en producción.
- Facilidad en el manejo de cursores dinámicos de información.
- Facilidad en la generación de bibliotecas para la actualización o consultas a la Base de Datos.

INFORMIX-NewEra

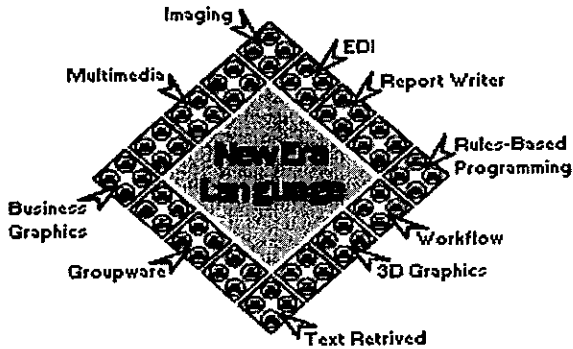
Informix - NewEra es la segunda generación de front-end's orientado a objetos desarrollado específicamente para cubrir aplicaciones complejas, es una herramienta flexible y es capaz de desarrollar prototipos gráficos de forma rápida además tiene la facilidad de crear aplicaciones particionadas, separando la operación y procesamiento tanto del cliente como del servidor.

Una de las principales ventajas que puede proporcionar NewEra es la flexibilidad para agregar o modificar librerías por lo que se pueden manejar proyectos de forma estándar.

Ventajas de NewEra

- Estandarización de librerías
- Incluye un debugger gráfico
- Generación de prototipos gráficos
- On-Line Help (help sensitivo en el área de interés).
- Compatibilidad con ODBC (proporciona una DLL estándar)

NewEra



Requerimientos de Software

Requerimientos para Desarrollo

| | |
|---|--|
| Lenguaje | Microsoft Visual C++™ Professional Edition |
| Espacio requerido para librerías (ejem. VCL y CCLs) | 11 MB. |
| Espacio del producto | 48 MB |
| INFORMIX-SE para Windows (producto adicional) | 10 MB |

Requerimientos Mínimos de Memoria

| | |
|----------------------------|-------|
| Deployment class libraries | 8 MB |
| Development | 16 MB |

Manejo de Datos (Informix)
INFORMIX-OnLine 5.x para UNIX®
INFORMIX-SE 5.x para Windows NT
Sybase, Oracle vía ODBC

Herramientas Oracle

PL/SQL es una herramienta práctica que nos proporciona una explotación de la Base de Datos a través de sentencias SQL, el poder de PL/SQL radica en soportar el estándar de SQL y adicionar funciones en el manejo de datos, un ejemplo de esto es la asignación de un rol (o perfil) de cada usuario, de esta forma se definen los permisos a tablas, columnas y en casos muy sencillos, permisos sobre aplicaciones, por lo anterior resulta sencillo en Oracle llevar un control de los accesos que debe tener cada usuario respecto de la Base de Datos.

Todas las herramientas que proporciona Oracle, requieren de establecer una conexión a la Base de Datos en tiempo de ejecución, lo que significa que introduzca el usuario y clave que se asignaron en tablas del sistema de la Base de Datos y con ello tome el perfil que le corresponde. Las herramientas de desarrollo que se describen a continuación son parte de la segunda generación de herramientas Cliente/Servidor.

SQL FORMS

Teniendo como base una poderosa Base de Datos y complementada con un potente lenguaje para la creación de informes (SQL *PLUS) era natural que Oracle se complementara con un generador de pantalla ya que es la principal forma que tienen los usuarios de tratar con la Base de Datos. Dicha herramienta es Oracle Forms. Se efectúa en un entorno de interfaces gráficas (GUI Graphical User Interface), con el aspecto y el funcionamiento de MicroSoft Windows.

Los desarrolladores de aplicaciones, diseñan las pantallas de introducción de datos y de consulta con Oracle Forms; los usuarios finales pueden utilizar dichas pantallas para manejar los datos en la Base de Datos; la interface con el usuario está controlada por eventos o teclas de función. La combinación de un conjunto gráfico con la versatilidad de la Base de Datos forma parte del arte de trabajar con Oracle Forms.

SQL REPORT

Como su nombre lo indica, está herramienta nos permite generar reportes complejos de manera rápida, tiene la facilidad de proporcionar un seguimiento del esquema de la base de datos y de esta forma integrar las columnas de las tablas a plantillas de reportes, cuenta con una amplia variedad de funciones como son promedio, valor máximo/mínimo, desviación estándar, etc.

Es importante mencionar que cada extracción de información puede ser configurada como una conexión diferente a la base de datos, lo anterior para definir con mayor precisión que usuarios deben realizar que tipo de extracción, a continuación se enuncian los informes más comunes que pueden ser realizados por Oracle Reports:

Maestro Detalle

Matriz

Circulares

PRO C

Un camino alternativo para emplear SQL en una aplicación es a través de la técnica llamada *embedded SQL*. Cuando se emplea un lenguaje como C es posible adicionar sentencias SQL, Oracle permite este tipo de herramienta proporcionando la facilidad de manipular los datos almacenados en una base de datos, esta herramienta envía el código a convertir a funciones ya establecidas e integra las librerías y bibliotecas necesarias en el programa, siendo el compilador el encargado de ligar dichas librerías e integrarlas al ejecutable.

Developer 2000 y Designer 2000

Developer 2000 (CDE o Cooperative Development Environment) y Designer 2000 ofrecen una solución completa cuando los clientes de Oracle necesitan diseñar, programar, implementar y mantener sistemas. Ofrecen un rápido desarrollo de aplicaciones en un entorno Windows Cliente/Servidor. Sus funciones avanzadas incorporan BPR (Business Process Reengineering) y mecanismos para aprovechar el procesamiento del servidor que puede realizarse mediante el motor de Base de Datos de Oracle; ambos productos tienen un alto grado de integración con el Oracle Server. El procesamiento se divide entre el servidor y el cliente utilizando procedimientos almacenados y disparadores (triggers), en developer 2000 se reúnen los siguientes productos Oracle:

Oracle Forms

Oracle Reports

Oracle Graphics

Oracle Book

Oracle Browser

Oracle Procedure Builder

Oracle Open Client Adapter para ODBC
SQL*PLUS
SQL*NET.

Una funcionalidad del GUI común, incorporada en la última generación de herramientas de desarrollo de informes y formularios de Oracle, proporciona la interfaz entre el analista y developer 2000/designer 2000. El desarrollo puede realizarse en pequeños grupos de trabajo y las aplicaciones se distribuyen con facilidad a cientos de usuarios. Developer 2000 ofrece generación automática de código a demás de distribución automatizada del software. PL/SQL, se ha incorporado en todos sus productos de desarrollo, le ofrece a designer 2000 y ha developer 2000 el mismo entorno de desarrollo de programas en el cliente que en el servidor. Designer 2000 admite una amplia variedad de funciones del modelo de negocios, que les permite a las empresas construir sistemas desde el más sencillo hasta el mas complejo, siendo designer 2000 la siguiente generación del conjunto de productos CASE de Oracle.

Ventajas de emplear herramientas de desarrollo ORACLE:

- Proporciona un ambiente gráfico
- Las funciones adicionales que proporcionan son mayores en número y calidad
- Es sencillo definir perfiles de usuarios a nivel Base de Datos.

2.3 Herramientas de Conectividad

Las herramientas de conectividad como su nombre lo indican, son las encargadas de enviar/recibir datos de una aplicación a otra aplicación residente en otro equipo de cómputo, el protocolo que más comúnmente se emplea es TCP/IP. A continuación se detallan las dos herramientas de conectividad que proporciona Informix.

Herramienta INFORMIX

I-NET

Los productos de informix fueron desarrollados en dos arquitecturas diferentes las cuales son:

- 1.- Aplicaciones que controlan la interfaz con el usuario a través de la manipulación de datos en el servidor, es decir por cada proceso cliente se levanta un proceso servidor en el equipo servidor.
- 2.- En el sistema estándar las comunicaciones entre las herramientas de informix y el servidor de la red, informix NET es usado para establecer un circuito virtual, de tal forma que solo son enviadas las sentencias SQL del cliente al servidor.

Las dos formas presentan ventajas respecto a la otra y la mayor diferencia es el lugar en que se ejecutan los procesos la información viaja completa en ambos sentidos y los procesos se ejecutan en el servidor, para el segundo punto I-NET solo funge como un traductor de sentencias SQL para el servidor. Una cualidad en las herramientas cliente / servidor es la transparencia en la localización de la Base de Datos, es decir el acceso a diferentes Bases de Datos es sencillo porque se pueden emplear variables de ambiente y sinónimos para los nombres y de forma automática I-NET realiza una búsqueda en el cliente si no es encontrada la B.D. toma el valor de las variables de medio ambiente para buscar en los lugares especificados y finalmente si no es encontrada la B.D. establece un circuito virtual entre el cliente y el servidor para hacer la búsqueda siempre que se tengan los privilegios adecuados.

INFORMIX-NET ofrece integridad y seguridad en la transmisión de datos por la red, esto se implementa cuando se establece el circuito virtual, el servidor siempre mantiene el proceso "sqlxecd" el cual esta en espera de alguna petición.

Especificaciones para INFORMIX-NET Versión 7.0x

Requerimientos de Espacio

| | |
|------------------|---------------|
| Espacio en Disco | 1.9 megabytes |
|------------------|---------------|

Informix/StarGROUP

TCP/IP
StarGROUP

TLI o Berkeley
versión 3.1, 3.2, 3.3
o 3.4 TLI
TLI

SPX/IPX

I-Star

Herramienta de conectividad residente en el servidor de la Base de Datos, para versiones anteriores a la 6.0, I-Star debe instalarse con un I-On_Line.

Las herramientas anteriores hacen posible la implementación de esquemas de Base de Datos Distribuidas, así como Base de Datos Cliente/Servidor.

| | |
|-------------------------------------|---|
| Capacidades de Acceso | Máximo número de Bases de Datos distribuidas 26 |
| Requerimientos de Memoria | Memoria compartida para INFORMIX-STAR (sqlxecd) 246 kilobytes |
| Espacio en disco para INFORMIX-STAR | 2 megabytes |

Herramienta ORACLE

Por la estructura que maneja Oracle, no requiere de una herramienta de conectividad.

2.4 El Manejador de Base de Datos

INFORMIX proporciona dos tipos de servidores de Base de Datos, en función de las necesidades del usuario.

I-SE

INFORMIX Standard Engine, es un servidor de Base de Datos pensado en aplicaciones pequeñas, o ligadas con el front-end, su costo es bajo en comparación con On_Line u otros manejadores comerciales.

I-On_Line

On_Line es un servidor de Base de Datos *OLTP*, Con la capacidad de procesos de recuperación ante fallas. Como se puede observar este *RDBMS* esta pensado en el manejo de grandes volúmenes de información, para sistemas abiertos.

A continuación se presentan tres tablas, con características y parámetros de ambos servidores de Base de Datos.

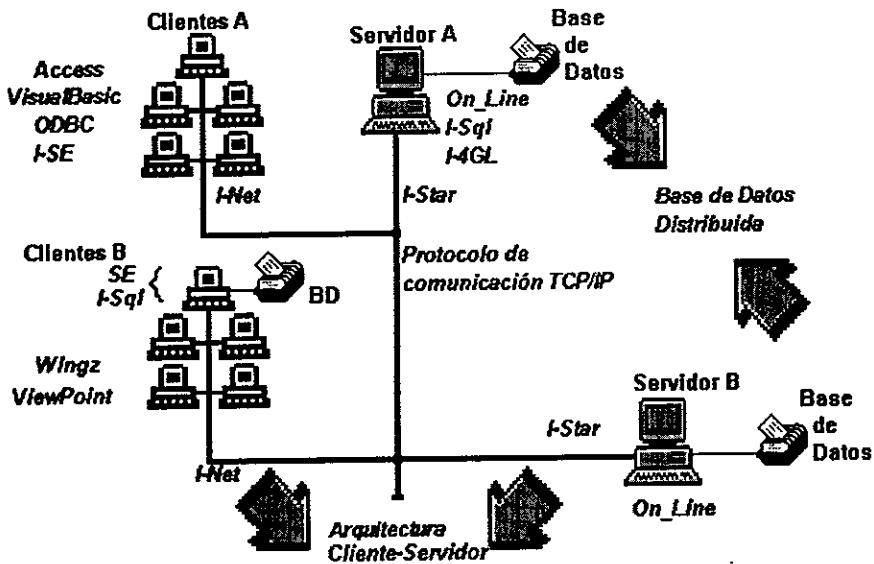
| PARAMETROS DE ESPACIO | SE | On_Line |
|--------------------------------------|----------------------------|-----------------------|
| Número de registros por tabla | 2,146 millones | 4,278 millones |
| Número de datos por tabla | 8,350 millones | 16,775 millones |
| Máxima longitud de registro | 32,767 bytes | 32,767 bytes |
| Máximo número índices por tabla | 70 | 77 |
| Máximo número de columnas por tabla | 32,767 | 32,767 |
| Máximo número de columnas por índice | 16 | 16 |
| Longitud máxima de llave primaria | 255 | 255 |
| Longitud máxima de una tabla | longitud máxima de archivo | 33,822 millones Bytes |

| Capacidad de Acceso | SE | On_Line |
|---|--------------------|-------------|
| Número máximo de BD's definidas | Dependiente del SO | 21 millones |
| Máximo número de tablas abiertas | Dependiente del SO | 32,000 |
| Máximo número de candados tablas/usuarios | Dependiente del SO | 32 |
| Número de cursores por usuario | 32 | 32 |
| Máximo número de niveles B-tree | Dependiente del SO | 20 |
| Máxima longitud de TEXT o Byte | No aplica | 2 Gbytes |
| Máxima cuota de string | 256 | 256 |
| Máxima longitud en un campo | 32,767 | 32,767 |

| Funcionalidad | SE | On_Line |
|--------------------|---------------------|-------------------------|
| Manejo de Disco | FileSystem | FileSystem y Área Cruda |
| Memoria Compartida | No (Depende del SO) | Asignada por el Admin |

| Mecanismos de Respaldo | Por SO y en Ascii | Tabla, Base de Datos, Total |
|-------------------------------|-------------------|-----------------------------|
| OLTP | No | Si |
| Herramienta de Administración | Ninguna | Herramientas TB* |
| Método de Recuperación | Ninguno | Fast-Recovery |
| Manejo de datos tipo Blob | No | Si |

La siguiente figura ilustra la ubicación de cada herramienta, en un contexto general.



VENTAJAS DE SERVICIOS INTEGRADOS DE INFORMIX - OnLine.

- Se tienen estadísticos acerca del espacio en disco utilizado por las aplicaciones.
- Cuenta con mecanismos integrados para el respaldo y recuperación de información.
- Mejora el uso y rendimiento de los recursos críticos, tales como la memoria y el espacio en disco, ya que son considerados para todas las bases de datos.
- Proporciona a los administradores una herramienta con interfaces amigable para administrar y monitorear el comportamiento de los recursos.
- Permite el uso de tipos de datos Varchar, Text y Byte.
- Facilita el acceso a sistemas distribuidos mediante el uso de otras herramientas de Informix.

RDBMS ORACLE

Oracle Server es un moderno entorno de gestión de información, es un almacén para grandes cantidades de datos y le proporciona al usuario un rápido acceso a dichos datos. Oracle Server ofrece la compartición de datos entre las aplicaciones; la información esta almacenada en un lugar y se utiliza en muchos sistemas; Oracle Server puede ejecutarse en docenas de computadores distintos y admite las siguientes configuraciones:

Basado en Anfitrión: los usuarios se conectan directamente al mismo computador en el que se encuentra la Base de Datos.

Cliente/Servidor: Los usuarios acceden a la Base de Datos desde su computador independiente (servidor)

Procesamiento distribuido: los usuarios acceden a una Base de Datos que reside en más de un computador encontrándose repartida en más de una máquina, y los usuarios no son conscientes de la ubicación física de los datos con los que trabajan.

Los sofisticados mecanismos de seguridad de Oracle permiten controlar el acceso a datos delicados mediante una gran variedad de privilegios. A los usuarios se les conceden derechos para examinar, modificar y crear datos basándose en el nombre que han utilizado para conectarse a la base de datos. El usuario utiliza estos mecanismo para asegurarse de que cada usuario en específico pueda tener acceso a datos que posiblemente se encuentren prohibidos para el resto de los usuarios.

Por otra parte Oracle proporciona un mecanismo de copia de seguridad y recuperación, la copia de seguridad crea una secundaria de los datos de Oracle, mientras que la recuperación restablece una copia de los datos a partir de dicha copia de seguridad. Con la estrategia de copia de seguridad y recuperación de Oracle se minimiza la pérdida de datos y el tiempo de indisponibilidad en caso de que se produzcan problemas. Siendo el acceso ininterrumpido a la base datos las 24 horas del día, proporcionando conectividad hacia y desde el software de otros vendedores, mediante módulos añadidos a la Base de Datos de esta forma es posible trabajar con información distribuida, el Oracle Server, denominado habitualmente el motor de la Base de Datos admite una amplia gama de herramientas de desarrollo, las cuales fueron enunciadas con anterioridad.

Componentes de Oracle Server

El producto básico ofrece todas las funciones necesarias para la mayoría de los clientes de Oracle, cuando los clientes necesitan las opciones procedimental (procedural), distribuida (distributed), consultas en paralelo (parallel query) o servidor en paralelo para cumplir con los requisitos de su negocio, son fácil de ser integradas, a continuación se enuncian las características más importantes.

Opción procedimental

Con la versión 7.0 esta opción debe comprarse por separado del Server; con la versión 7.1 forma parte del mismo. La base de esta opción se encuentra en el lenguaje de programación de Oracle denominado PL/SQL.

Procedimientos Almacenados

Son programas o segmentos de código que se almacenan en la Base de Datos Oracle y llevan a cabo las funciones centrales para la instalación, por ejemplo en una aplicación de facturación de televisión por cable, puede utilizarse un procedimiento almacenado para crear una carta recordatoria para enviar a los clientes morosos, la ejecución de dicho procedimiento activaría la ejecución de una sentencia mensual del cliente cuando existan recibos impagados durante mas de 60 días.

Opción servidor en paralelo

Algunos fabricantes crean computadores agrupados: cada máquina del grupo (CLUSTER) tiene su propia memoria aunque todos ellos comparten servicios de almacenamiento en discos. La opción del servidor en paralelo permite que Oracle funcione con esta configuración cada máquina se denomina nodo del grupo, y se utiliza el término vagamente acoplador (loosely coupled) para hacer referencia a los nodos, la opción de consulta en paralelo permite que los clientes aprovechen las ventajas de un grupo de procesadores con más de una unidad central de proceso.

Cuando se utiliza la opción de consulta en paralelo en máquinas con varias CPU's, Oracle despacha varios procesos de consulta que funcionan de forma simultánea, repartiéndose el procesamiento de la consulta; los resultados se combinan y se presentan al usuario cuando están preparados.

Ventajas de Oracle Server:

- Al producto Oracle Server se le pueden añadir cuatro opciones que proporcionan mayor funcionalidad y las características adicionales que necesitan una parte de los clientes de Oracle.
- La opción procedimental incluye PL/SQL, además de procedimientos almacenados, disparadores, funciones y paquetes. Estos segmentos de código se encuentran en la base de datos y las aplicaciones pueden compartirlos.

- La opción distribuida permite que los usuarios trabajen con datos que se encuentran en una base de datos remota como si fueran locales.
- La opción de servidor en paralelo permite que un grupo de computadores compartan un espacio en disco común pero manteniendo su propia memoria.

2.5 Selección de la Herramienta a Emplear

Como resultado de las características y cualidades de las herramientas de desarrollo y del manejador de Base de Datos, se concluye que la plataforma que debe ser empleada para implantar y desarrollar el sistema es Informix puesto que proporciona las siguientes ventajas globales sobre Oracle:

- El requerimiento de Espacio y Memoria RAM son menores.
- Con cualquier herramienta de desarrollo es posible armar una aplicación que contemple reportes, captura de datos, consultas y actualización a los mismos (una aplicación completa).
- El rendimiento del manejador Informix es más óptimo (mayor velocidad de acceso a datos).
- Considerando que actualmente la plataforma oficial es Informix, la posibilidad de exportar datos entre diferentes equipos no requiere de ningún software adicional.

III. Análisis

El análisis estructurado como todos los demás métodos de análisis de requisitos, es una actividad de construcción de modelos, mediante una notación que es única del método empleado, de tal forma que se crean modelos que reflejan el flujo y contenido de la información (datos y control); se particiona el sistema en sus funciones básicas y según los distintos comportamientos, se establece la esencia de lo que se debe construir.

La tarea del análisis de sistemas conlleva más allá de realizar análisis de requisitos, una de las principales labores del analista es descubrir detalles y documentar las reglas del negocio que pueden encontrarse implícitas, "transmitidas de generación en generación", por los usuarios u operadores de sistemas anteriores, las cuales generalmente no se encuentran documentadas formalmente. El analista debe distinguir entre síntomas, problemas del usuario y causas. Con sus conocimientos tecnológicos y de software, el analista debe ayudar al usuario a explorar aplicaciones novedosas y más útiles de éstos, así como nuevas formas de cumplir con sus reglas de negocio. Aunque muchos de los sistemas antiguos sólo se limitaban a perpetuar el negocio original del usuario, sin embargo con las actualidades tecnológicas hoy en día, los analistas se enfrentan al desafío de ayudar al usuario a encontrar productos y mercados radicalmente innovadores con la ayuda de la computadora.

3.1 Modelado de Datos

Como es sabido las dos grandes ramas en el desarrollo de un sistema son *datos* y *procesos*, en el rubro de datos contamos con los siguientes elementos que nos facilitan el proceso de desarrollo del sistema :

| <i>Etapa</i> | <i>Herramienta</i> |
|---------------------|---------------------------|
| <i>Análisis</i> | Entidad – Relación |
| <i>Diseño</i> | Tablas Relacionales |
| <i>Construcción</i> | Bases de Datos (Archivos) |

El modelo entidad relación (E-R) fue propuesto por Chen en el año de 1976 y ha sido estudiado por varios autores, el E-R se ve principalmente como una herramienta al análisis. También puede ser utilizada como un lenguaje de alto nivel, para generar esquemas los cuales pueden ser convertidos en entradas de otros diagramas. Teniendo como elementos básicos del E-R a la entidad y la relación.

Componentes de un E-R.

Entidad

Considerando un número de conjuntos cada uno orientado a un tipo particular de objetos, los conjuntos están relacionados con dominios y atributos. Si se tiene la relación dada por el producto cartesiano de estos dominios, una interpretación que se le da a cada una de estas tuplas es que cada una corresponde a una entidad particular. En términos de abstracción, un "tipo de entidad" corresponde a la agregación de atributos, así las entidades son una correspondencia de un tipo de entidad.

Interrelación

Es la ocurrencia (existencia) entre dos entidades. Ejemplo, la relación "dueño de" puede ser interpretada como una interrelación entre dos tipos de entidades *persona* y *auto*.

Atributo

Los elementos "tipo de entidad" y "tipo de interrelación" son descritos por una relación de atributos, estos atributos se pueden dividir en diferentes categorías:

- A. Descriptivos
- B. Clave Descriptiva
- C. Clave Asociativa

Restricciones

Cuando se consideran las veces que una entidad puede relacionarse con otra, se hace referencia a una restricción del esquema propuesto. En algunas implementaciones del modelo E-R estas restricciones se representan utilizando el concepto de cardinalidad en la cual se indican el máximo y el mínimo para la participación de una entidad con otra en la interrelación.

Cómo modelar en E-R

Para modelar en E-R se debe:

- A. Identificar los tipos de entidades.
- B. Identificar los tipos de interrelaciones.
- C. Encontrar las cardinalidades.
- D. Identificar los atributos de cada tipo de entidad.
- E. Identificar las claves de cada tipo de entidad.

La regla básica es distinguir tipos de entidades e interrelaciones de atributos. Así, los atributos deben ser atómicos y característicos del tipo de entidad o interrelación que describan, también los atributos deben pertenecer al tipo de entidad o interrelación que describen y no a otro tipo. Otra diferencia entre tipo de entidad y atributo es la relación natural que se da entre entidades, claro ejemplo de esto es la interrelación entre departamento y empleado en donde la interrelación "trabaja_en" relaciona la entidad empleado con la de departamento en ese sentido.

Otras características que debe cumplir un E-R

Complejidad

Un esquema es completo cuando representa todas las características pertinentes al dominio de la aplicación. Esto se puede comprobar mirando en detalle todos los requerimientos del dominio de la aplicación y verificando que cada uno de ellos esté representado en algún lugar del esquema (el esquema es completo respecto a los requerimientos) y también es posible revisar el esquema para verificar que cada concepto este mencionado en los requerimientos (los requerimientos están completos respecto al esquema).

Corrección

Un esquema es correcto cuando usa con propiedad los conceptos del modelo (MER en este caso), un esquema es sintácticamente correcto cuando los conceptos se definen con propiedad en el esquema; por ejemplo, los subconjuntos y las generalizaciones se definen entre entidades pero no entre interrelaciones. Un esquema es semánticamente correcto cuando los conceptos (entidades, interrelaciones, etc.) se usan de acuerdo con sus definiciones, por ejemplo es un error semántico usar un atributo para representar los productos de una empresa manufacturera cuando se necesita representar varias propiedades de los productos (código del producto, precio, partes que lo componen, etc.) en virtud de que un atributo es una propiedad elemental.

De acuerdo a las definiciones anteriores se dará inicio con el diagrama entidad relación, este diagrama debe cubrir las necesidades de almacenamiento y las posibles relaciones que se identifiquen entre las entidades. Derivado de los requerimientos planteados en el capítulo uno, se identificaron como entidades fundamentales a las siguientes (entidades que existen por sí solas):

CENTRO DE CÓMPUTO

GERENCIA

USUARIO

Se identificaron como entidades débiles o dependientes de otras:

ÁREA

NIVEL

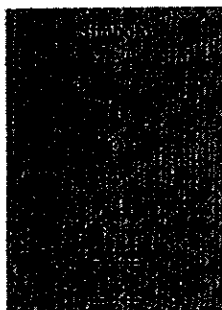
EMPLEADO

y como entidad asociativa:

REPORTE

A continuación se muestra el diagrama con sus relaciones y campos por entidad

Simbología empleada:



Significado

Llave primaria

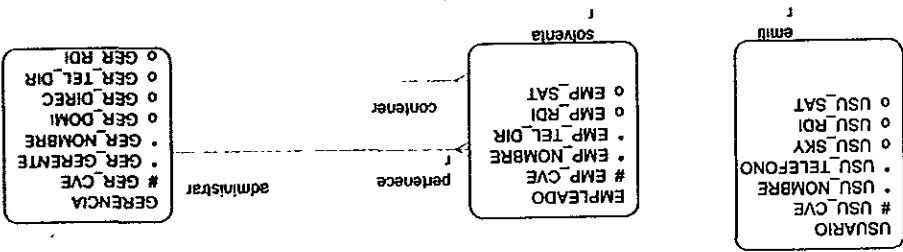
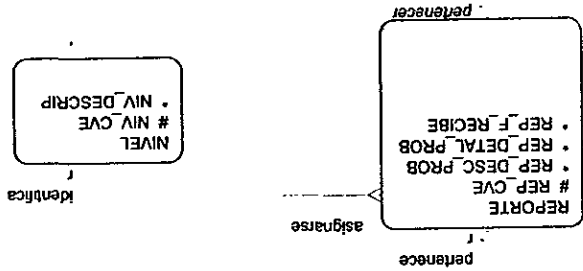
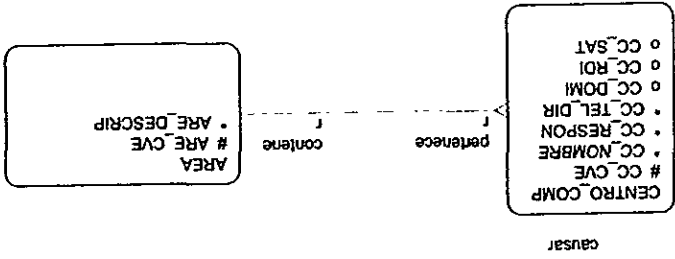
Campo no requerido

Campo requerido (no nulo)

Relación que se lee "Puede + verbo "

Relación que se lee "Debe + verbo "

Relación de uno a muchos



¿Qué almacena cada entidad?

| | |
|--------------------|---|
| USUARIO | <i>Almacena los usuarios permitidos para levantar Solicitudes de Asistencia</i> |
| GERENCIA | Almacena los nombres del responsable y el nombre del área encargada de brindar el soporte técnico (sistemas operativos, comunicaciones, base de datos, áreas de desarrollo, etc.) a los centros de cómputo. |
| EMPLEADO | Almacena los nombres de responsables por gerencia de soporte técnico a los diferentes centros de cómputo |
| NIVEL | Almacena los niveles de escalamiento permitidos en el sistema |
| ÁREA | Almacena el nombre y descripción del área a la cual pertenece el centro de cómputo |
| CENTRO_COMP | Deberá contener los datos del centro de cómputo, así como el nombre del responsable del mismo. |
| REPORTE | Deberá almacenar datos relacionados con el problema y su solución |

Como se observa del diagrama anterior la entidad "reporte" es la más importante porque en ella estarán contenidos los datos del problema así como su solución, en esta misma entidad se almacenará el estado que guarda cada reporte.

3.2 Modelado de Procesos

Cuando se considera una solución modular para cualquier problema, pueden formularse muchos niveles de abstracción. En el nivel superior de abstracción, se establece una solución en términos amplios, usando el lenguaje del entorno del problema. En los niveles inferiores de abstracción se toma una orientación más procedimental. La terminología orientada al problema se acompaña con una terminología orientada a la implantación, en un esfuerzo para establecer una solución. Por último en el nivel más bajo de abstracción, se establece la solución de forma que pueda implementarse directamente.

Conceptos Aplicables al Modelado de Procesos

Cohesión.

Grado en el cual los componentes de un módulo (típicamente las instrucciones individuales que lo conforman) son necesarios y suficientes para llevar a cabo una sola función bien definida. En la práctica, esto significa que el diseñador debe asegurarse de no fragmentar los procesos esenciales en módulos, y también debe asegurarse de no juntar procesos no relacionados en módulos sin sentido. Los mejores módulos son aquellos que son funcionalmente cohesivos (es decir, los

módulos en los cuales cada instrucción es necesaria para poder llevar a cabo una tarea bien definida). Los peores módulos son aquellos que se caracterizan por ser coincidentalmente cohesivos (es decir, donde sus instrucciones no tienen una relación significativa entre uno y otro).

Los grados de cohesión, de menor a mayor son:

- A) **Cohesión coincidental.** No existe una relación significativa entre los elementos del módulo.
- B) **Cohesión lógica.** La relación entre los elementos del módulo esta basada en obtener ventajas en el procesamiento, por ejemplo, todos manipulan el mismo dato. Normalmente esto implica tener un código truculento o compartido que degrada los propósitos de un buen diseño.
- C) **Cohesión temporal.** Los elementos del módulo constituyen un conjunto que se ejecuta secuencialmente en un punto fijo en el tiempo. Aunque tiende a veces a confundirse con la cohesión lógica, la diferencia está en que este tipo de módulo es mas simple y se ejecuta sin la intervención de otras aplicaciones.
- D) **Cohesión comunicacional.** Los elementos del módulo hacen referencia al mismo conjunto de datos. Aquí se presenta un grado "aceptable" de cohesión.
- E) **Cohesión secuencial.** Implica que la salida de un elemento es la entrada para el próximo.
- F) **Cohesión funcional.** Aquí todos los elementos del módulo están orientados a la realización de una función única.

Acoplamiento.

Grado en el cual los módulos se interconectan o se relacionan entre ellos. Entre mas fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo, pues entonces se necesitará un estudio cuidadoso para la modificación de algún módulo o módulos, en la práctica esto significa que cada módulo debe tener interfaces sencillas y limpias con otros, además se debe compartir un número mínimo de datos entre módulos. También significa que un módulo dado no debe modificar la lógica interna o los datos de algún otro módulo lo que se conoce como una conexión patológica.

Diagrama de Flujo de Datos

Los diagramas de flujo de datos también llamados Carta de Burbujas, DFD's, modelo de procesos, diagrama de trabajo o modelo de función en la literatura computacional. A medida que la información se mueve a través del software es modificada por una serie de transformaciones, el DFD es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

Componentes de un DFD

El Proceso

Sinónimos comunes son burbuja, función o transformación.

El proceso muestra una parte del sistema que transforma entradas en salidas; es decir, muestra como es que una o más entradas se transforman en salidas. El proceso se representa gráficamente como un ovalo o un rectángulo con esquinas redondeadas. Estas referencias son solo de forma y se debe optar por alguna de ellas y utilizarla en forma consistente, las representaciones utilizadas para procesos, se cuenta con Gane y Sarson, o la de Ward y Mellor, así como por Yourdon y De Marco, en todos los casos el proceso se nombra con una palabra o frase que intenta dar una primera aproximación de lo que hace, por ejemplo VALIDAR ENTRADA, CONTROL TEMPERATURA, etc.

El Flujo

Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso. El flujo se usa para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra, por ello los flujos representan datos en movimiento, mientras que los almacenes representan datos en reposo.

En la mayoría de los sistemas que se modelan, los flujos realmente representarán datos, es decir, bits, caracteres mensajes, números de punto flotante y los diversos tipos de información con los que se suele tratar en sistemas computarizados, de lo anterior se dice que los flujos de datos pueden converger o divergir en un DFD.

El Almacén (Data Store)

El almacén se utiliza para modelar un conjunto de paquetes de datos en reposo. Se denota por dos líneas paralelas, de modo característico el nombre que se utiliza para un almacén es el plural del que se usa para los paquetes que entran y salen del almacén por medio del flujo.

A menudo los almacenes de datos se implementan como archivos o base de datos. También pueden ser implementados en sistemas manuales como archivadores, carpetas, etc.

La Entidad Externa (o Terminador)

Un terminador gráficamente se representa como un círculo, los terminadores representan entidades externas con las cuales el sistema se comunica, comúnmente un terminador es una persona o un grupo, por ejemplo una organización externa o una agencia gubernamental, o un grupo o departamento que este dentro de la misma compañía u organización pero fuera del control del

sistema que se está modelando. En algunos casos el terminador puede ser otro sistema, suele ser muy fácil identificar los terminadores en el sistema que se está modelando, a veces el terminador es el usuario que nos dice "pienso entregar los datos A, B y C al sistema y espero que éste me entregue los datos X, Y y Z". En otros casos el usuario se considera parte del sistema y ayudará a identificar los terminadores relevantes.

Puntos Importantes en la Construcción de un DFD

- a. Escoger nombres con significados para los procesos, flujos, almacenes y terminadores.
- b. Numerar los procesos.
- c. Redibujar el DFD tantas veces como sea necesario estéticamente.
- d. Editar los DFD's excesivamente complejos.
- e. Asegurarse de que el DFD sea internamente consistente y que también lo sea con cualesquiera DFD relacionado con él (evitar procesos con solo entradas o salidas, así como flujos y procesos no etiquetados).

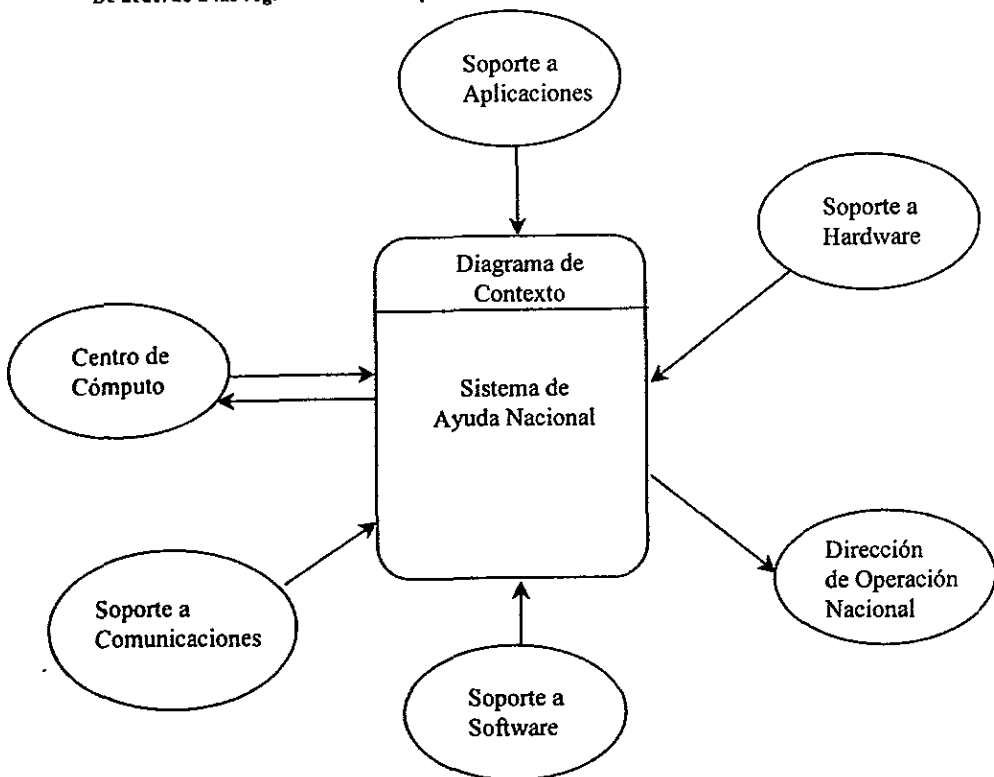
DFD por Niveles

Se organiza el DFD global en una serie de niveles de modo que cada uno proporcione sucesivamente más detalles sobre una porción del nivel anterior. Esto es análogo a la organización de mapas en un atlas.

El DFD de primer nivel consta sólo de una burbuja, que representa el sistema completo; los flujos de datos muestran las interfaces entre el sistema y los terminadores externos (junto con los almacenes externos que pudiera haber). Este DFD especial se conoce como diagrama de contexto.

El DFD que se muestra se conoce como la figura cero, representando la vista de más alto nivel de las principales funciones del sistema, al igual que sus principales interfaces.

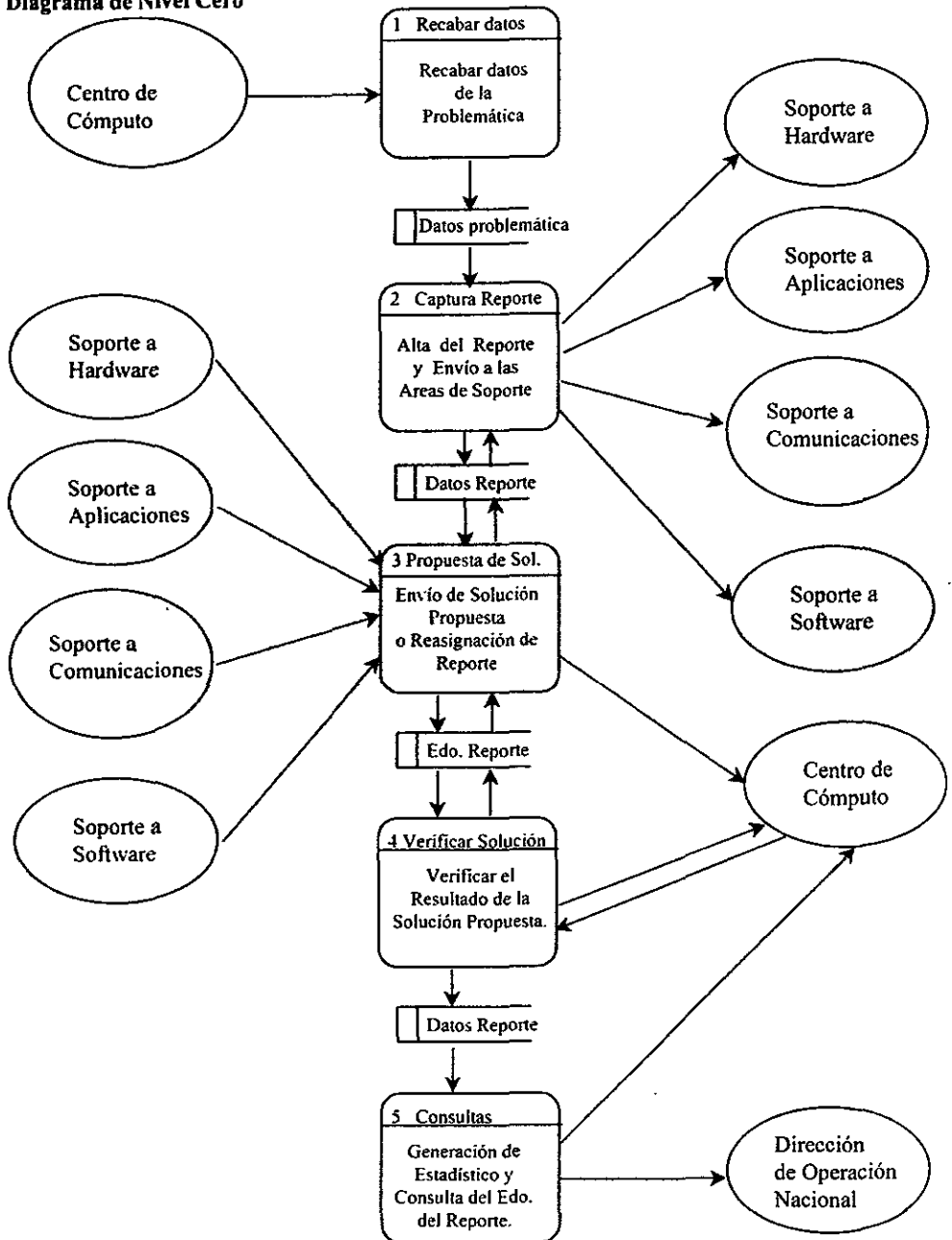
De acuerdo a las reglas anteriores se presenta el DFD de contexto para el sistema de Ayuda Nacional.



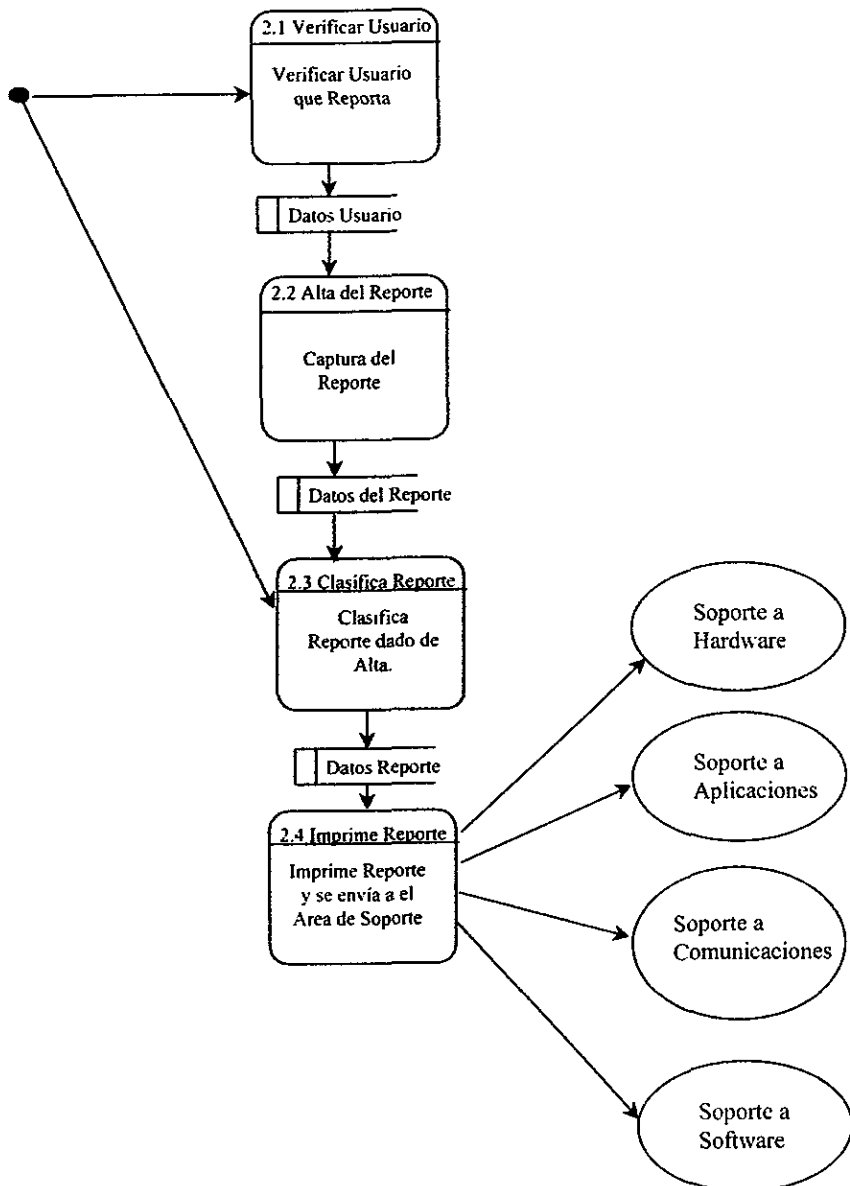
Del diagrama anterior podemos observar que los agentes externos con los cuales el sistema deberá interactuar son; el centro de cómputo (envío y recepción de problemas), y las áreas de soporte (hardware, software, desarrollo de aplicaciones) y comunicaciones.

El diagrama siguiente divide el proceso de control y seguimiento de los reportes en cada centro de cómputo en cinco procesos de los cuales el primero (Recabar Datos), es un proceso manual, todos los demás procesos se deberán automatizar, los siguientes diagramas son la descomposición de las burbujas del diagrama cero:

Diagrama de Nivel Cero

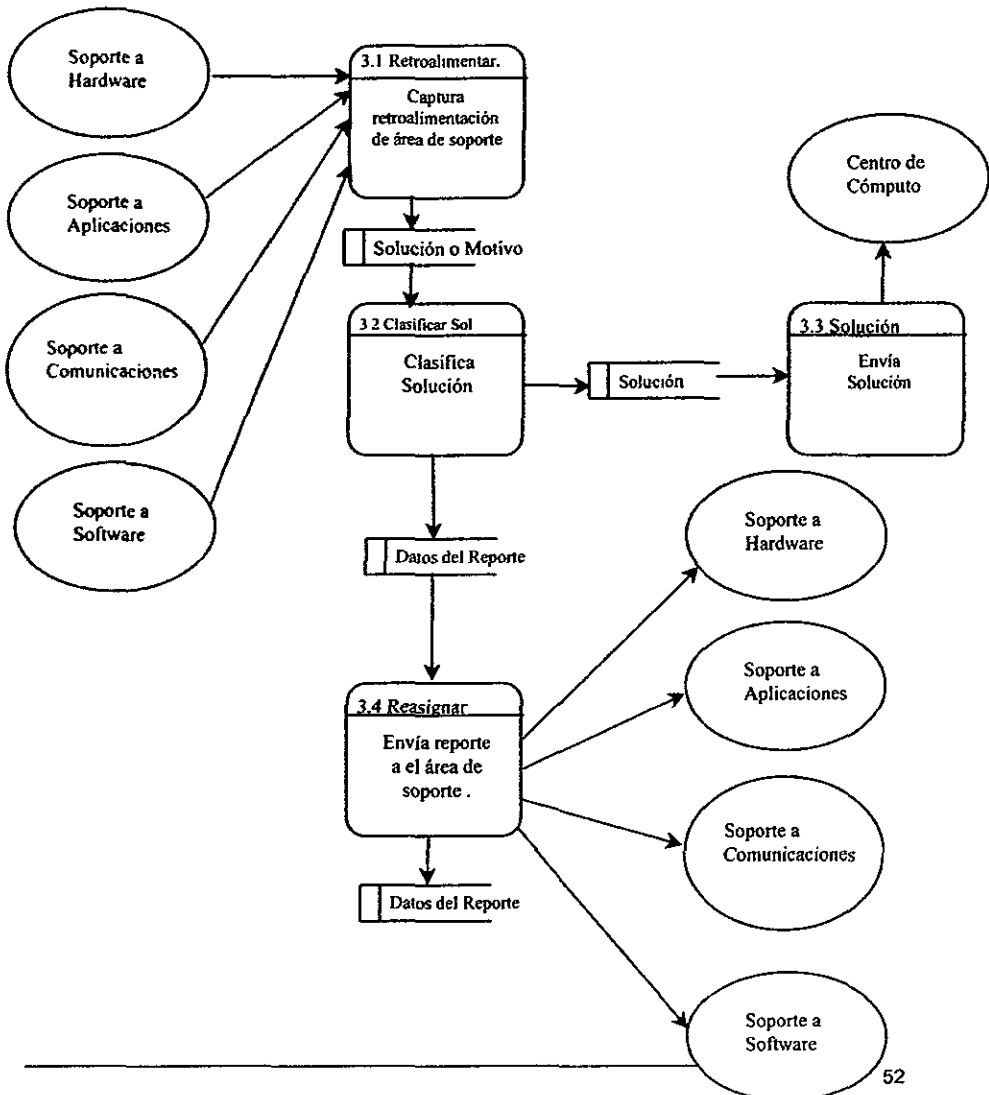


Diagramas de Captura del Reporte



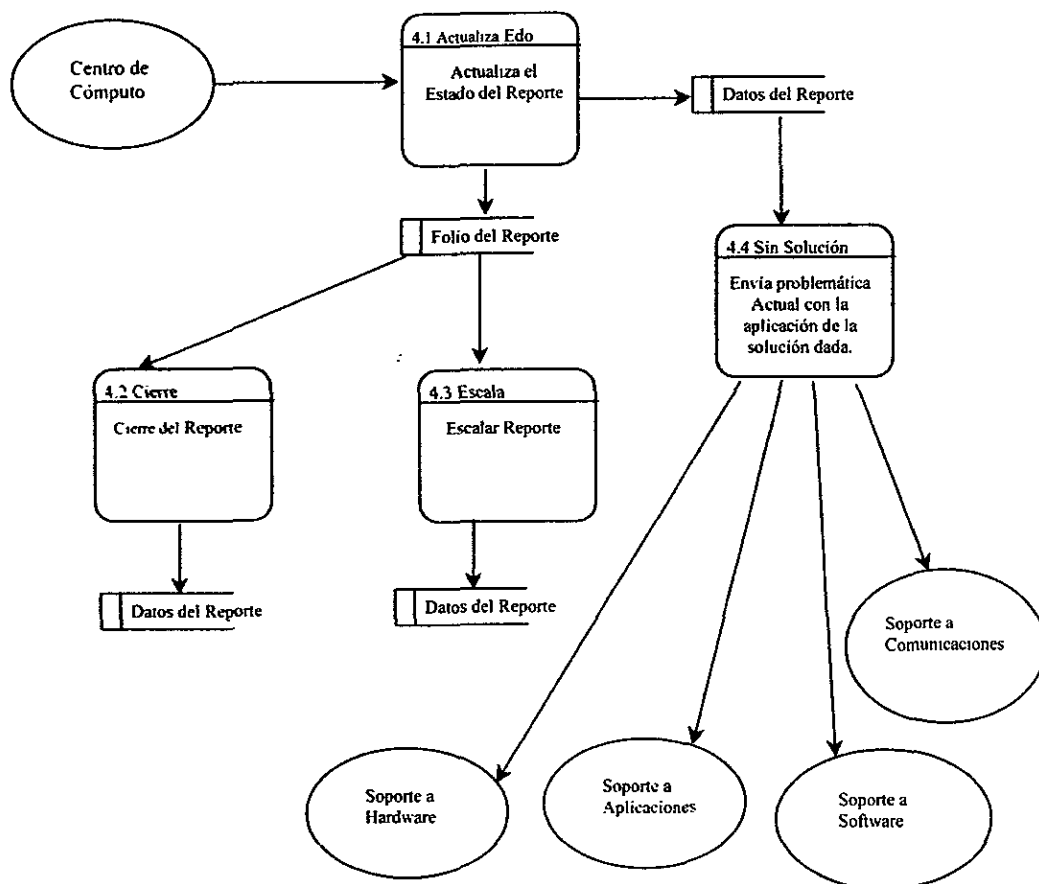
Como se observa del diagrama anterior inicialmente se debe verificar el usuario que levanta el reporte, posteriormente se captura en detalle los datos del problema empleando un campo tipo texto para la problemática y otro para la solución a esta problemática, el siguiente proceso es el encargado de clasificar el nivel de seguridad del reporte conjuntamente con el usuario que levanta el reporte, por último se imprime el reporte y es enviado al área que debe atenderlo. Todos los niveles posteriores a estos procesos son minispec, por lo que se mostrarán en la etapa de diseño.

Descomposición de los diagramas de “Envío de Solución propuesta o Resignación del Reporte”



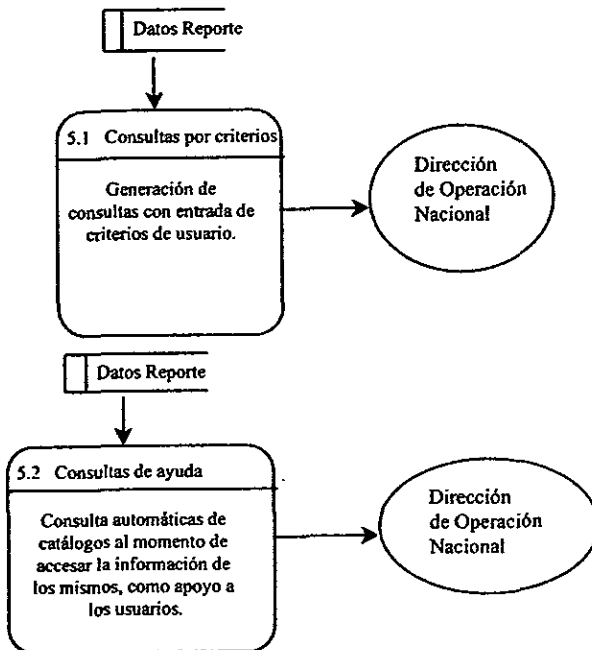
Como se observa del diagrama anterior una vez emitida una solución por el área de soporte a la cual fue asignado el reporte, tiene dos posibilidades, una es que se proporcione al centro de cómputo como una alternativa para solventar su problemática y la segunda es que el reporte después de ser analizado se determine que pertenece a un área diferente, aquí es importante tomar en cuenta los tiempos en los cuales se deben emitir ambas respuestas, debido a que el usuario tiene la facultad de escalar el problema si no recibe una atención rápida (acorde con la severidad) a su reporte.

Explotación de los procesos "Verificar Solución"



De acuerdo con las burbujas del diagrama anterior existen tres posibilidades sobre el reporte que son; cerrar el reporte una vez que el centro de cómputo este de acuerdo con la solución proporcionada por el área de soporte, la segunda es escalar el reporte (cada causa de escalamiento será explicada posteriormente) y la tercera es solicitar a las áreas de soporte una nueva solución. Es importante hacer notar que en cada burbuja el siguiente nivel de explotación debe ser una minispec, así mismo el datastore "Datos del Reporte" representa una actualización en la información de la tabla "reporte".

Diagrama de Consultas



Del diagrama anterior podemos concluir que se deberán realizar dos tipos de consultas, las que nos apoyen en la localización de datos referentes a reportes y en la cual es posible ingresar criterios como son rangos de fechas, el número asignado de reporte, etc., por otra parte también se tendrán las consultas que apoyen principalmente la captura del reporte en la cual es necesario contar con un acceso automático a los catálogos, ejemplo de esto se presentará al digitar un número de centro de cómputo que no exista, por lo que se debe desplegar una pantalla con los centros de cómputo existentes y una posibilidad de dar de alta un nuevo registro.

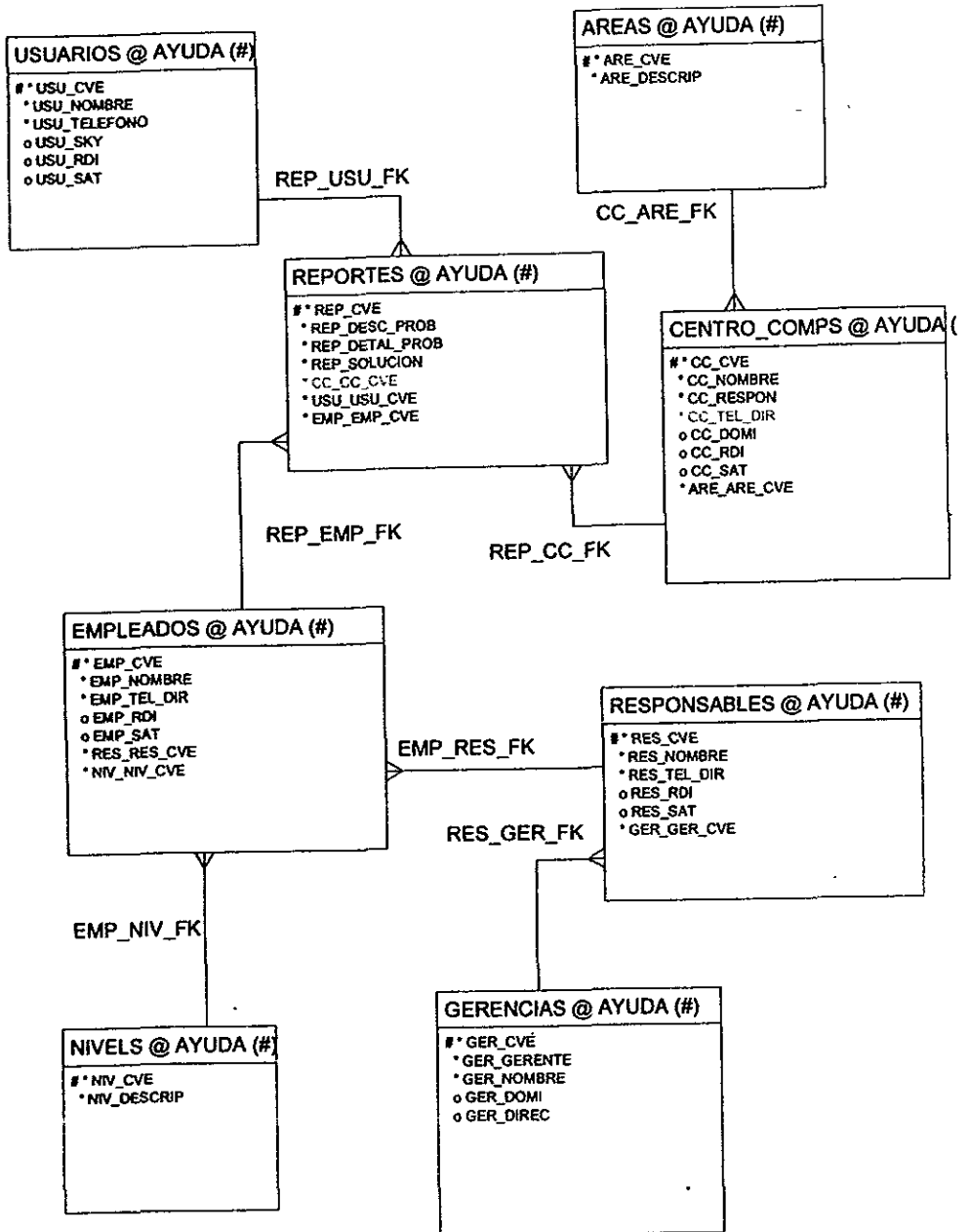
IV. Diseño

El diseño de software es un proceso mediante el que se traducen los requisitos en una representación del software. Inicialmente, la representación describe una visión logística del software, posteriores refinamientos conducen a una representación de diseño que se acerca mucho al código fuente. En el diseño se realizan dos pasos. El diseño preliminar se centra en la transformación de los requisitos en los datos y arquitectura del software. El diseño detallado se ocupa del refinamiento de la representación arquitectónica que lleva a una estructura de datos detallada y a las representaciones algorítmicas del software.

Dentro del contexto de los diseños preliminar y detallado, se llevan a cabo varias actividades de diseño diferentes. Además del diseño de datos y del diseño procedimental, muchas aplicaciones requieren de un diseño de la interfaz. El diseño de la interfaz establece la disposición y los mecanismos para la interacción hombre máquina (no cubierto por las herramientas del diseño estructurado).

4.1 Estructura de la Base de Datos

A continuación se presenta el diagrama de tablas como resultado del diagrama entidad relación, es importante notar que en este diagrama se dibujan los campos llave y los campos que deberán migrar como foráneos a las diferentes entidades, se respeta la misma simbología que en el diagrama entidad relación.



4.2 Diseño del Ambiente de la Aplicación

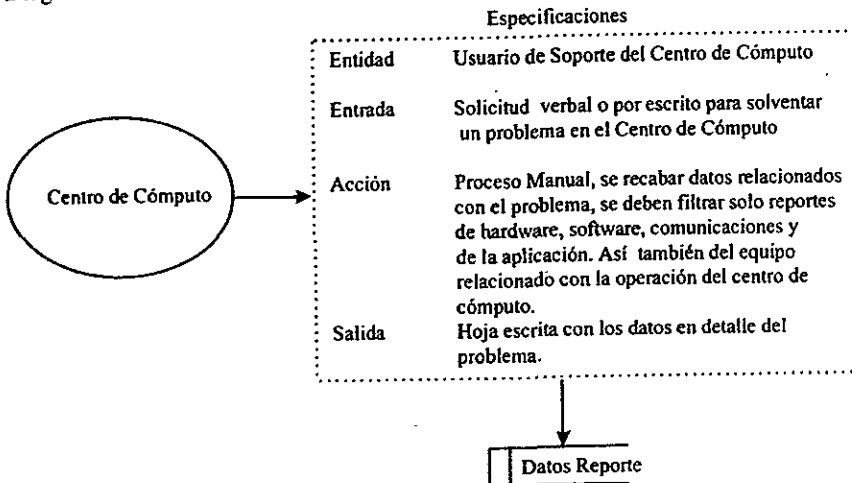
La especificación del proceso es la descripción de qué es lo que sucede en cada burbuja primitiva en el nivel más bajo en un DFD. También es llamado Minispec o miniespecificación. Su propósito es definir lo que debe hacerse para transformar entradas en salidas.

La forma más utilizada para realizar las especificaciones de procesos es el lenguaje estructurado, pero se puede utilizar cualquier método, el proceso debe especificarse en una forma que pueda ser comunicada efectivamente al público amplio que esté involucrado. A pesar de que el analista es típicamente quien escribe la especificación del proceso, habitualmente será un público bastante diverso de usuarios, administradores, auditores, personal de control de calidad y otros, el que leerá la especificación del proceso.

A continuación se presentan las Minispec, resultado de los diagramas de flujo de datos.

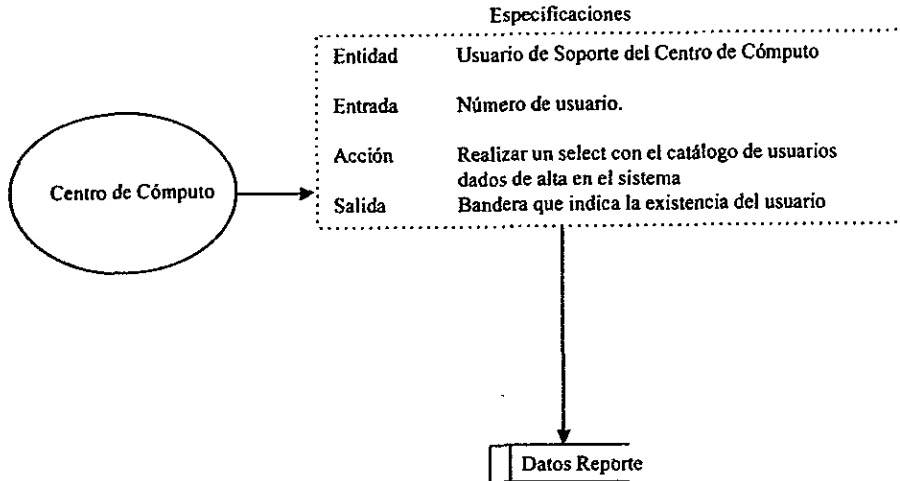
Recabar datos.

Diagrama 1.1

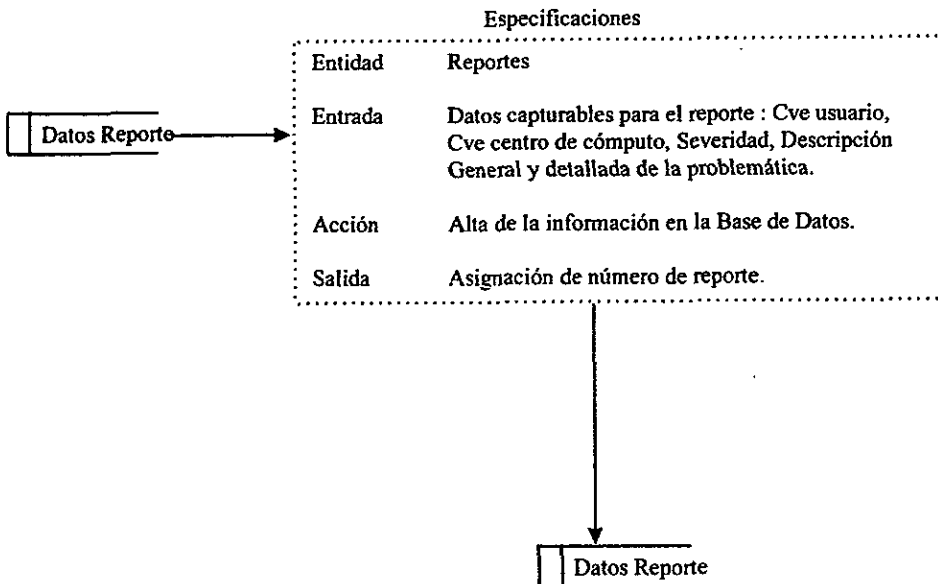


Verificar Usuario

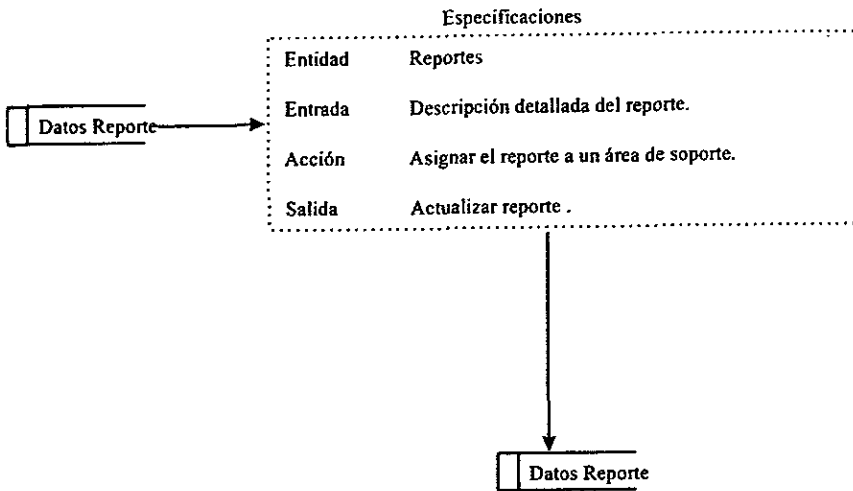
Diagrama 2.1.1



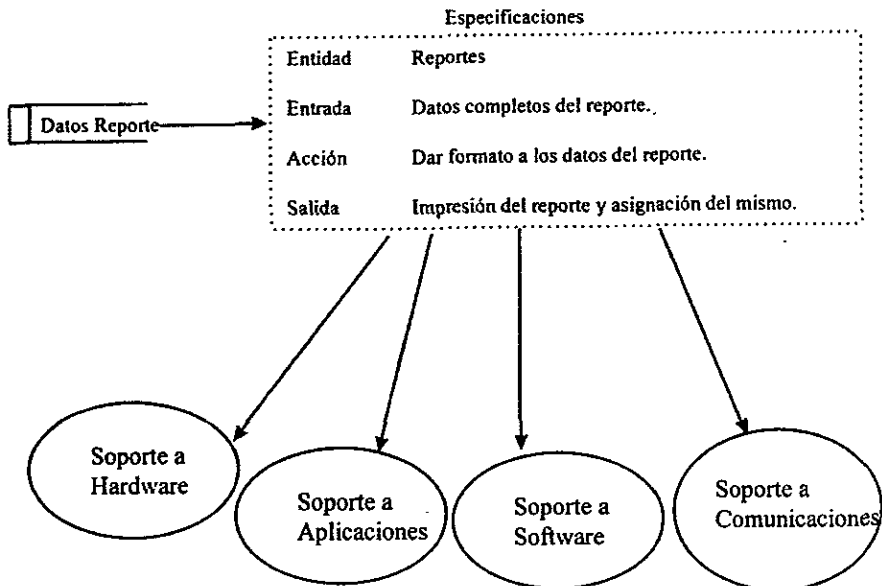
Captura del Reporte Diagrama 2.2.1



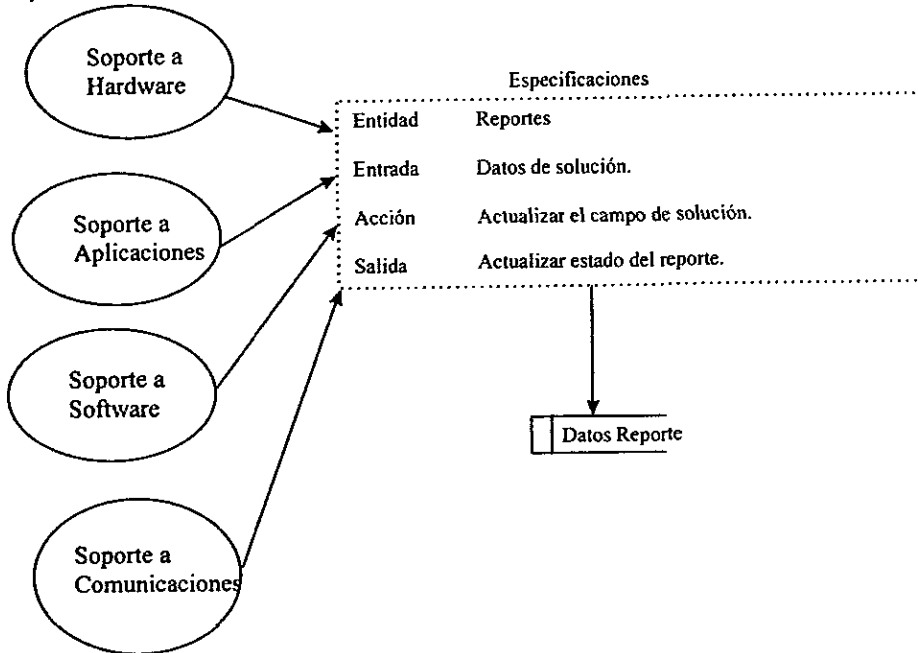
Clasificación del reporte Diagrama 2.3.1



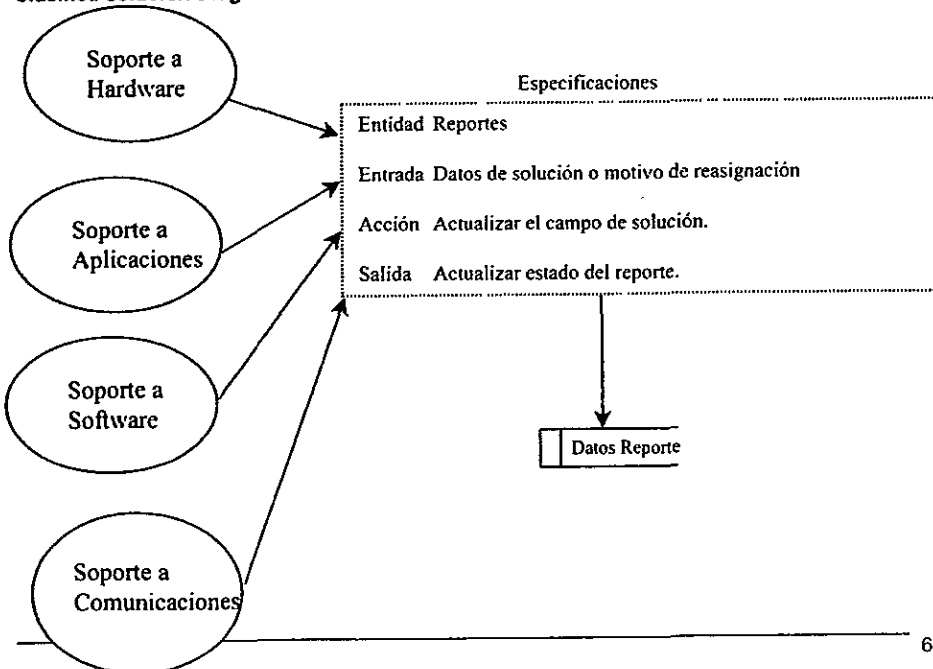
Impresión del Reporte Diagrama 2.4.1

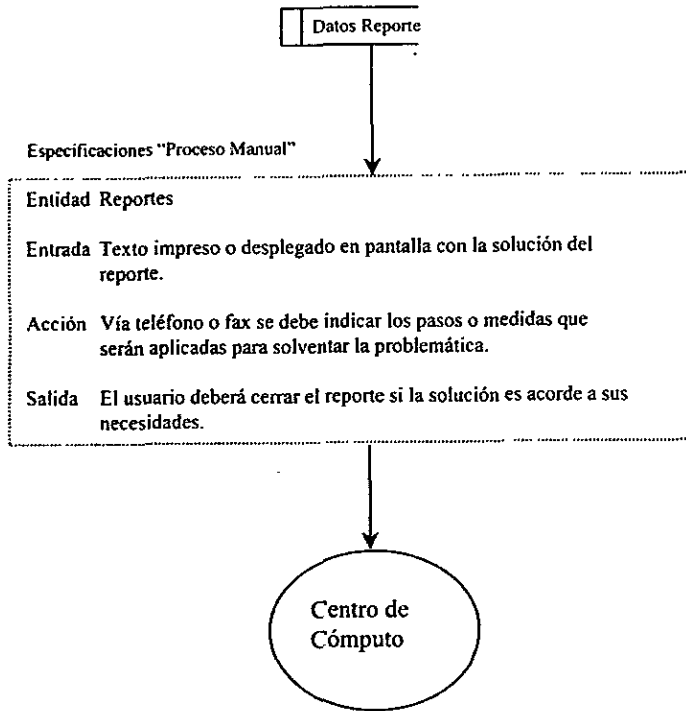


Captura de la solución o motivo de reasignación Diagrama 3.1.1

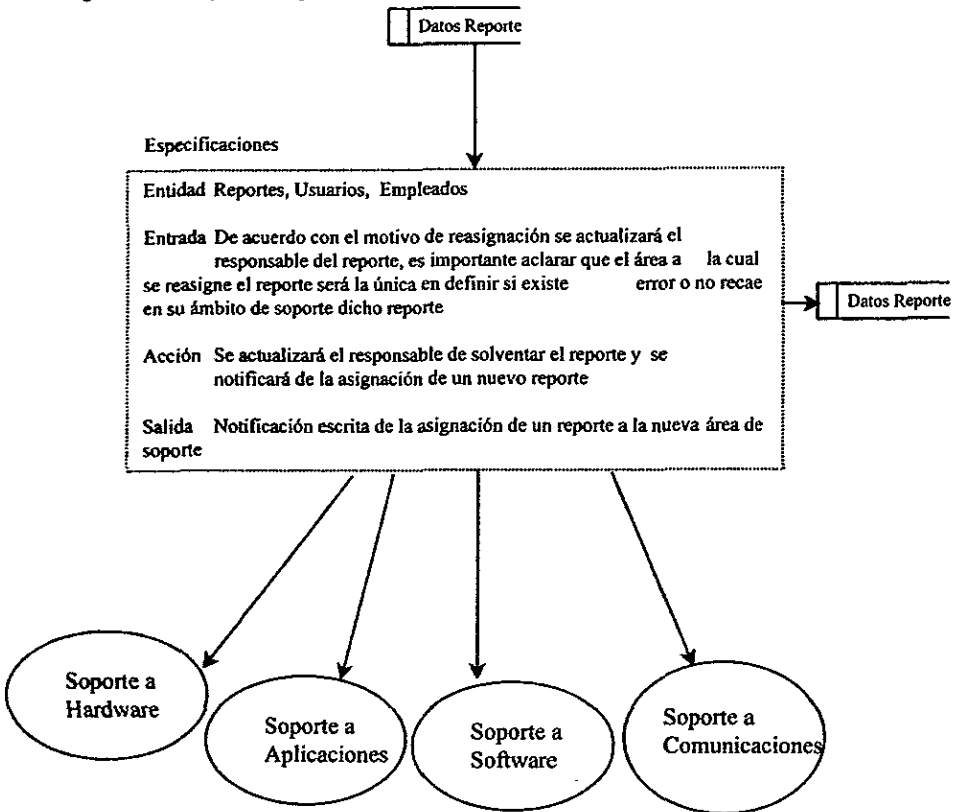


Clasifica solución Diagrama 3.2.1

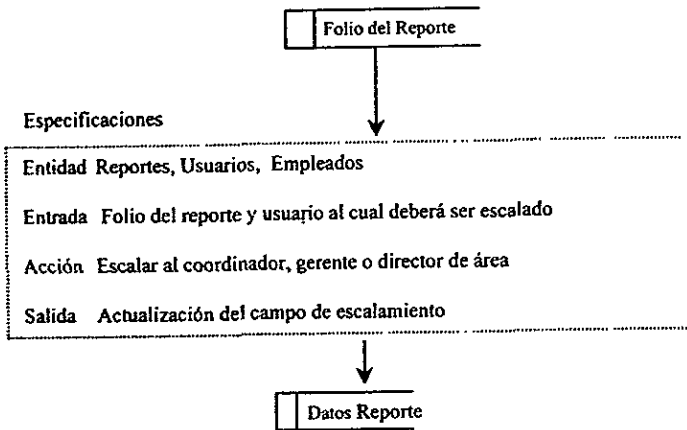


Envío de solución Diagrama 3.3.1

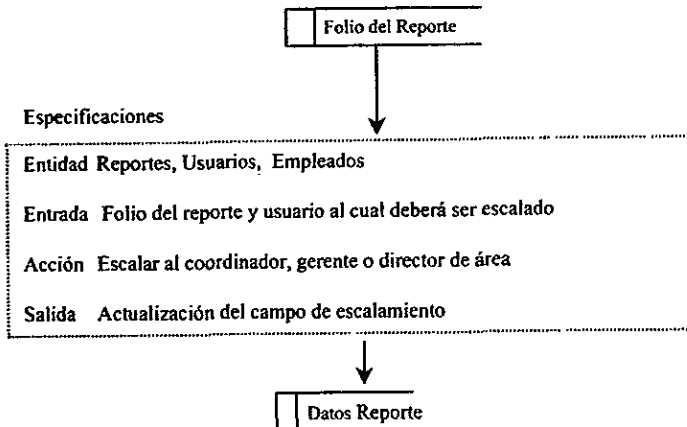
Reasignación de reporte Diagrama 3.4.1



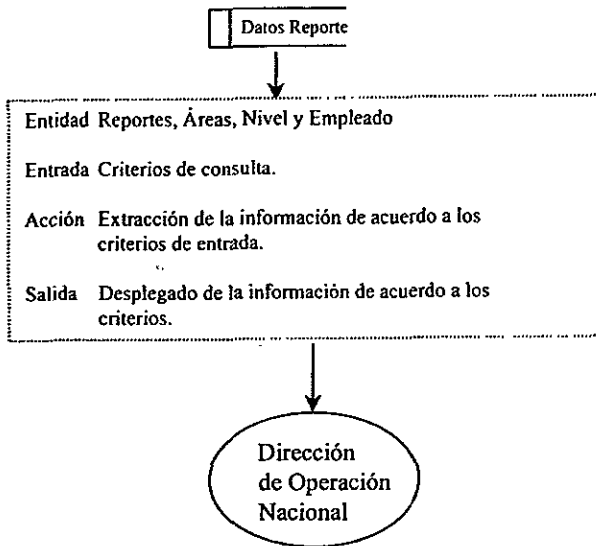
Cierre del reporte Diagrama 4.2.1



Escalamiento Diagrama 4.3.1



Consultas Diagrama 5.1.1



4.3 Diseño del Plan de Pruebas e Implantación.

En este punto se deben plantear las políticas y el nivel de calidad que debe tener el sistema de tal forma que se cumpla con las especificaciones del mismo. La etapa de pruebas es la confirmación de la etapa de construcción, por lo que se deben llevar a cabo una minuciosa revisión de los entregables del sistema y por otra parte constatar las ventajas que ofrece el mismo.

Plan de Pruebas

Inicialmente se debe considerar el nivel de funcionalidad con que cuenta el sistema, respecto de los objetivos planteados para en capítulos anteriores, por lo que debe cubrir los siguientes rubros:

1. Para dar seguimiento a un reporte o problema, el sistema deberá proporcionar un folio o número único que identifique al reporte y a través de este medio conocer de forma global y en detalle el estado del reporte.
2. Los datos de problemática y solución no deben tener un límite en su almacenamiento, por lo que puede ser tan amplio como se presente la ocasión.
3. El sistema deberá proporcionar los reportes necesarios, para cumplimentar las políticas de respuesta implantadas a las áreas de Soporte Técnico.
4. Cuando se presente un reporte que detenga la operación de un centro de cómputo y sea de consecuencias graves, deberá proporcionar un mecanismo que alerte de forma rápida de ser necesario a niveles directivos, con la finalidad de que sea solventado con prontitud.

Por otra parte el sistema debe ser fácil de operar, y presentar una interface amigable al usuario, dando como beneficio un incremento en la productividad de las personas que operen el sistema, de manera puntual, el sistema debe cubrir lo siguiente:

1. Debe presentar ayuda en línea de los catálogos que emplee.
2. Debe presentar ventanas tipo pull-down y teclas rápidas en su operación.
3. *Debe ser un sistema multiusuario.*
4. En todo tipo de consulta masiva que sea realizada, debe facilitar la forma de consultar, teniendo la facilidad de emplear wilcards (por ejemplo *, ?, etc..).
5. El sistema deberá estar programado de forma modular, por lo que su mantenimiento y adiciones sean sencillas

- 6. El sistema debe contar con un manual de usuario, documentación técnica y de la programación del mismo.

Diseño de la Implantación

La infraestructura empleada para el sistema es la siguiente:

Servidor :

| | |
|--------------------------------------|--------------------------------|
| Equipo | HP Apollo 715 con 32 MB en RAM |
| Espacio en disco para datos | 100 MB |
| Manejador de Base de Datos | Informix On-Line 5.0 |
| | Runtime de I-4GL |
| Sistema Operativo | HP-UX 9.04 serie 700 |
| Red | Tarjeta de red 3COM |
| Protocolo de Comunicación | TCP/IP |
| Multiplexor de Terminales (opcional) | HP |

De emplearse un equipo PC los requerimientos de la misma son los mínimos para correr windows 3.11 o posterior, además deberá contar con red y TCP/IP como protocolo de comunicaciones, obviamente debe encontrarse en la misma red o contar con la posibilidad de acceder el servidor vía telnet.

El sistema tendrá la facilidad de manejar tres perfiles de usuarios, los usuarios que levantan reportes y actualizan el estado en todo momento de reporte, como es la captura de la solución, la retroalimentación del usuario del centro de cómputo, etc., otro usuario que realizará los escalamientos de reportes y por último el usuario de consultas. Cada usuario deberá acceder solo a las opciones que tiene derecho, aun cuando se presenten en el menú los demás accesos, En la base de datos también se deben agregar estos perfiles, almacenando en el system de los manejadores los permisos necesarios para cada usuario (el esquema de la Base de Datos debe contener los permisos).

Respecto a las líneas telefónicas se asignarán 3 líneas, con una extensión, de estas tres líneas una deberá contar con salida RDI, otra con salida satelital y la última con salida de largas distancias nacionales. Por otra parte se deberá tener el servicio de lada 800.

Para certificar que el sistema se encuentra en condiciones de operar a nivel nacional, se realizará una prueba piloto con tres centros de cómputo, una aduana (Nuevo Laredo), un centro local de recaudación (D.F. Norte) y un centro regional de fiscalización, el tiempo estimado de prueba será de tres meses.

V Construcción e Implantación

5.1 Construcción de la Base de Datos.

A continuación se presenta el esquema de Base de Datos empleado:

```

***** ORDEN DE EJECUCIÓN DE ARCHIVOS *****

# Versión 5 5.3 2.0 on 04-APR-97
#
# For application PRU_HD versión 1 database AYUDA
#
SET SCAN OFF
SPOOL AYUDA.txt
# DATABASE CREATION
start AYUDA.db

# TABLE CREATION
start AYUDA.tbl

# INDEX CREATION
start AYUDA.ind
# CONSTRAINT CREATION
start AYUDA.con
#
# End of command file
#
SPOOL OFF

***** GENERACIÓN DE TABLAS *****

# Versión 5 5.3 2.0 on 04-APR-97
#
# For application PRU_HD versión 1 database AYUDA
#
# TABLE
# AREAS
# CENTRO_COMPS
# EMPLEADOS
# GERENCIAS
# NIVELS
# REPORTES
# RESPONSABLES
# USUARIOS

PROMPT
PROMPT Creating Table AREAS
CREATE TABLE areas(
  are_cve          NUMBER(4,0)          NOT NULL,
  are_descrip     VARCHAR2(60)         NOT NULL
);

COMMENT ON COLUMN areas.are_cve   IS 'Identificador de las areas';
COMMENT ON COLUMN areas.are_descrip IS 'Descripción de las diferentes areas';
#
PROMPT
PROMPT Creating Table CENTRO_COMPS
CREATE TABLE centro_comps(
  cc_cve          NUMBER(4,0)          NOT NULL,
  cc_nombre       VARCHAR2(60)         NOT NULL,
  cc_respon       VARCHAR2(60)         NOT NULL,
  cc_tel_dir      VARCHAR2(40)         NOT NULL,
  cc_doma         VARCHAR2(60)         NULL,
  cc_rdi          VARCHAR2(40)         NULL,
  cc_sam          VARCHAR2(40)         NULL,
  are_are_cve     NUMBER(4,0)          NOT NULL
);

COMMENT ON COLUMN centro_comps.cc_cve   IS 'Clave para el centro de computo';
COMMENT ON COLUMN centro_comps.cc_nombre IS 'Nombre del centro de computo';
COMMENT ON COLUMN centro_comps.cc_respon IS 'Responsable del centro de computo';
COMMENT ON COLUMN centro_comps.cc_tel_dir IS 'Telefono directo del centro de computo (Lada)';

```

```

COMMENT ON COLUMN centro_comps cc_domi IS 'Dominio del centro de computo.';
COMMENT ON COLUMN centro_comps cc_rfd IS 'Telefono de RDI del centro de computo.';
COMMENT ON COLUMN centro_comps cc_sat IS 'Telefono de red satelital del centro de computo.';
COMMENT ON COLUMN centro_comps are_are_cve IS 'Identificador de las areas';
#

```

```

PROMPT
PROMPT Creating Table EMPLEADOS
CREATE TABLE empleados
emp_cve          NUMBER(4,0)      NOT NULL,
emp_nombre      VARCHAR2(60)   NOT NULL,
emp_tel_dir     VARCHAR2(40)   NOT NULL,
emp_rfd        VARCHAR2(40)    NULL,
emp_sat        VARCHAR2(40)    NULL,
res_res_cve    NUMBER(4,0)    NOT NULL,
niv_niv_cve    NUMBER(1,0)    NOT NULL
#

```

```

COMMENT ON COLUMN empleados.emp_cve IS 'Clave par el empleado.';
COMMENT ON COLUMN empleados.emp_nombre IS 'Nombre para el empleado.';
COMMENT ON COLUMN empleados.emp_tel_dir IS 'Numero directo de telefono (Lada)';
COMMENT ON COLUMN empleados.emp_rfd IS 'Numero de rdi del empleado.';
COMMENT ON COLUMN empleados.emp_sat IS 'Numero de red satelital del empleado.';
COMMENT ON COLUMN empleados.res_res_cve IS 'Identificador para el responsable.';
COMMENT ON COLUMN empleados.niv_niv_cve IS 'Identificador para el nivel de usuarios';
#

```

```

PROMPT
PROMPT Creating Table GERENCIAS
CREATE TABLE gerencias
ger_cve          NUMBER(4,0)      NOT NULL,
ger_gerente     VARCHAR2(60)   NOT NULL,
ger_nombre      VARCHAR2(60)   NOT NULL,
ger_doma       VARCHAR2(60)    NULL,
ger_dircc      VARCHAR2(60)    NULL
#

```

```

COMMENT ON COLUMN gerencias.ger_cve IS 'Clave que identifica a cada gerencia.';
COMMENT ON COLUMN gerencias.ger_gerente IS 'Nombre del gerente asignado a la misma.';
COMMENT ON COLUMN gerencias.ger_nombre IS 'Nombre de la gerencia.';
COMMENT ON COLUMN gerencias.ger_doma IS 'Ubicacion de la gerencia.';
COMMENT ON COLUMN gerencias.ger_dircc IS 'Direccion a la cual pertenece.';
#

```

```

PROMPT
PROMPT Creating Table NIVELES
CREATE TABLE niveles
niv_cve          NUMBER(1,0)      NOT NULL,
niv_descrip     VARCHAR2(60)    NOT NULL
#

```

```

COMMENT ON COLUMN niveles.niv_cve IS 'Identificador para el nivel de usuarios.';
COMMENT ON COLUMN niveles.niv_descrip IS 'Descripcion del nivel.';
#

```

```

PROMPT
PROMPT Creating Table REPORTES
CREATE TABLE reportes
rep_cve          NUMBER(7,0)      NOT NULL,
rep_desc_prob   VARCHAR2(60)    NOT NULL,
rep_detal_prob  LONG             NOT NULL,
rep_solucion    LONG             NOT NULL,
cc_cc_cve      NUMBER(4,0)      NOT NULL,
usu_usu_cve    NUMBER(4,0)      NOT NULL,
emp_emp_cve    NUMBER(4,0)      NOT NULL
#

```

```

COMMENT ON COLUMN reportes.rep_cve IS 'Identificador del reporte.';
COMMENT ON COLUMN reportes.rep_desc_prob IS 'Descripcion general del reporte.';
COMMENT ON COLUMN reportes.rep_detal_prob IS 'Descripcion detallada del problema.';
COMMENT ON COLUMN reportes.rep_solucion IS 'Descripcion de la solucion.';
COMMENT ON COLUMN reportes.cc_cc_cve IS 'Clave para el centro de computo.';
COMMENT ON COLUMN reportes.usu_usu_cve IS 'Clave asignada para un usuario.';
COMMENT ON COLUMN reportes.emp_emp_cve IS 'Clave par el empleado.';
#

```

```

PROMPT
PROMPT Creating Table RESPONSABLES
CREATE TABLE responsables
res_cve          NUMBER(4,0)      NOT NULL,
res_nombre      VARCHAR2(60)   NOT NULL,
res_tel_dir     VARCHAR2(40)   NOT NULL,
res_rfd        VARCHAR2(40)    NULL,
res_sat        VARCHAR2(40)    NULL,
res_ger_cve    NUMBER(4,0)    NOT NULL
#

```

```

COMMENT ON COLUMN responsables.res_cve IS 'Identificador para el responsable.';
COMMENT ON COLUMN responsables.res_nombre IS 'Nombre del responsable.';
COMMENT ON COLUMN responsables.res_tel_dir IS 'Telefono directo del responsable (Lada)';
COMMENT ON COLUMN responsables.res_rfd IS 'Telefono rdi del responsable.';
#

```


COMMENT ON COLUMN responsables.res_sat IS 'telefono satelital del responsable';
 COMMENT ON COLUMN responsables.per_ger_cve IS 'Clave que identifica a cada gerencia';

```

*
PROMPT
PROMPT Creating Table USUARIOS
CREATE TABLE usuarios(
usu_cve          NUMBER(4,0)      NOT NULL,
usu_nombre      VARCHAR2(60)    NOT NULL,
usu_telefono    VARCHAR2(40)    NOT NULL,
usu_sky         VARCHAR2(40)    NULL,
usu_rdi        VARCHAR2(40)    NULL,
usu_sat        VARCHAR2(40)    NULL
)
    
```

COMMENT ON COLUMN usuarios.usu_cve IS 'Clave asignada para un usuario';
 COMMENT ON COLUMN usuarios.usu_nombre IS 'Nombre del usuario';
 COMMENT ON COLUMN usuarios.usu_telefono IS 'Telefono directo (lida) del usuario';
 COMMENT ON COLUMN usuarios.usu_sky IS 'SKyTel del usuario';
 COMMENT ON COLUMN usuarios.usu_rdi IS 'Telefono de RDI del usuario';
 COMMENT ON COLUMN usuarios.usu_sat IS 'Telefono satelital del usuario';

***** GENERACIÓN DE ÍNDICES *****

```

*
* For application PRU_HD versión 1 database AYUDA
*
# INDEX
# CC_ARE_FK_1
# EMP_NIV_FK_1
# EMP_RES_FK_1
# REP_CC_FK_1
# REP_EMP_FK_1
# REP_USU_FK_1
# RES_GER_FK_1
PROMPT
PROMPT Creating Index CC_ARE_FK_1 on Table CENTRO_COMPS
CREATE INDEX CC_ARE_FK_1 ON CENTRO_COMPS
(
    are_ger_cve )
PCTFREE 40,
*
PROMPT
PROMPT Creating Index EMP_NIV_FK_1 on Table EMPLEADOS
CREATE INDEX EMP_NIV_FK_1 ON EMPLEADOS
(
    emp_ger_cve )
PCTFREE 40,
*
PROMPT
PROMPT Creating Index EMP_RES_FK_1 on Table EMPLEADOS
CREATE INDEX EMP_RES_FK_1 ON EMPLEADOS
(
    emp_res_cve )
PCTFREE 40,
*
PROMPT
PROMPT Creating Index REP_CC_FK_1 on Table REPORTES
CREATE INDEX REP_CC_FK_1 ON REPORTES
(
    cc_cc_cve )
PCTFREE 40,
*
PROMPT
PROMPT Creating Index REP_EMP_FK_1 on Table REPORTES
CREATE INDEX REP_EMP_FK_1 ON REPORTES
(
    emp_emp_cve )
PCTFREE 40,
*
PROMPT
PROMPT Creating Index REP_USU_FK_1 on Table REPORTES
CREATE INDEX REP_USU_FK_1 ON REPORTES
(
    usu_usu_cve )
PCTFREE 40,
*
PROMPT
PROMPT Creating Index RES_GER_FK_1 on Table RESPONSABLES
CREATE INDEX RES_GER_FK_1 ON RESPONSABLES
(
    ger_ger_cve )
PCTFREE 40,
    
```

***** GENERACIÓN DE CONSTRAINTS *****

```

# Versión 5.5.8.2.0 on 04-APR-97
#
# For application PRU_HD versión 1 database AYLDA
#
# CONSTRAINT

PROMPT Adding PRIMARY Constraint To AREAS Table

ALTER TABLE AREAS ADD (
  CONSTRAINT ARE_PK
  PRIMARY KEY (ARE_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To CENTRO_COMPS Table

ALTER TABLE CENTRO_COMPS ADD (
  CONSTRAINT CC_PK
  PRIMARY KEY (CC_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To EMPLEADOS Table

ALTER TABLE EMPLEADOS ADD (
  CONSTRAINT EMP_PK
  PRIMARY KEY (EMP_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To GERENCIAS Table

ALTER TABLE GERENCIAS ADD (
  CONSTRAINT GER_PK
  PRIMARY KEY (GER_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To NIVELS Table

ALTER TABLE NIVELS ADD (
  CONSTRAINT NIV_PK
  PRIMARY KEY (NIV_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To REPORTES Table

ALTER TABLE REPORTES ADD (
  CONSTRAINT REP_PK
  PRIMARY KEY (REP_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To RESPONSABLES Table

ALTER TABLE RESPONSABLES ADD (
  CONSTRAINT RES_PK
  PRIMARY KEY (RES_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding PRIMARY Constraint To USUARIOS Table

ALTER TABLE USUARIOS ADD (
  CONSTRAINT USU_PK
  PRIMARY KEY (USU_CVE)
  USING INDEX
  PCTFREE 10)

PROMPT Adding FOREIGN Constraint To CENTRO_COMPS Table

ALTER TABLE CENTRO_COMPS ADD (
  CONSTRAINT CC_ARE_FK
  FOREIGN KEY (ARE_ARE_CVE)
  REFERENCES AREAS (
    ARE_CVE))

PROMPT Adding FOREIGN Constraint To EMPLEADOS Table

ALTER TABLE EMPLEADOS ADD (
  CONSTRAINT EMP_RES_FK
  FOREIGN KEY (RES_RES_CVE)
  REFERENCES RESPONSABLES (
    RES_CVE))

PROMPT Adding FOREIGN Constraint To EMPLEADOS Table

```

```
ALTER TABLE EMPLEADOS ADD (
  CONSTRAINT EMP_NIV_FK
  FOREIGN KEY (NIV_NIV_CVE)
  REFERENCES NIVELS (
    NIV_CVE))

PROMPT Adding FOREIGN Constraint To REPORTES Table

ALTER TABLE REPORTES ADD (
  CONSTRAINT REP_CC_FK
  FOREIGN KEY (CC_CC_CVE)
  REFERENCES CENTRO_COMPS (
    CC_CVE))

PROMPT Adding FOREIGN Constraint To REPORTES Table

ALTER TABLE REPORTES ADD (
  CONSTRAINT REP_USU_FK
  FOREIGN KEY (USU_USU_CVE)
  REFERENCES USUARIOS (
    USU_CVE))

PROMPT Adding FOREIGN Constraint To REPORTES Table

ALTER TABLE REPORTES ADD (
  CONSTRAINT REP_EMP_FK
  FOREIGN KEY (EMP_EMP_CVE)
  REFERENCES EMPLEADOS (
    EMP_CVE)
)
/

PROMPT Adding FOREIGN Constraint To RESPONSABLES Table

ALTER TABLE RESPONSABLES ADD (
  CONSTRAINT RES_GER_FK
  FOREIGN KEY (GER_GER_CVE)
  REFERENCES GERENCIAS (
    GER_CVE))
```

5.2 Construcción de la Aplicación

A continuación se presentan los programas de la aplicación :

```
Programa de Variables globales :
-----
*
*
* Carlos Jiménez González --- Sistema de Reportes          *
*
* --- PROGRAMA GLOBALS 4GL ---                               *
*
* Programa de variables globales                             *
*
*
-----
globals

define des array[15] of char(50). # opcion en el menu vertical contiene
                                # el texto de cada opcion

opc array[15] of char(1). # contiene la letra asociada con el
                          # numero de opcion del menu vertical

i_query char(250). # empleada en todos los construct
let1 char(250). # VARIABLE PARA REPORTES
asigna char(1) # variable para el responsable

end globals

-----
*
*
* Carlos Jiménez González --- Sistema de Soporte          *
*
* --- PROGRAMA SOP000 4GL ---                               *
*
* Programa de utilerías generales contiene las siguientes funciones :
*
* menuopt( ) , mon_ver(x) , display_sopt( ) , display_usot( )
*
*
-----
```

```

global
"global: $g"

# -----#
#
# Función MENSAJE es empleada para desplegar un string en una ventana #
# es empleada por las pantallas de trabajo #
# -----#

funcion mensaje(cade)
define
cade:char(40) = longitud del string a desplegar

open window k at 8,12 with 3 rows,45 columns attribute(message line 2;border)
message cade
sleep ?
close window k
end funcion

funcion cuenta(cade)
define
cade:char(40) = longitud del string a desplegar

open window k at 8,12 with 3 rows,45 columns attribute(message line 2;border)
for i=1 to cade
display "NUMERO DE REGISTROS : "i at 2,5
end for
sleep 3
close window k
end funcion
# -----#
#
# Función OPCION es empleada para la pregunta de eliminación de registros #
# -----#

funcion opcion()
define
op:char(1) = longitud del string

let op=""
open window lk at 8,12 with 3 rows,40 columns attribute(border,reverse,prompt line 2)
while (op<>"S" and op<>"s" and op<>"n" and op<>"N")
prompt " DESEA LA IMPRESION <S/N> "
for char op attribute (bold)
end while
close window lk
return op
end funcion

funcion confirm()
define
op:char(1) = longitud del string

let op=""
open window lk at 8,12 with 3 rows,40 columns attribute(border,prompt line 2)
while (op<>"S" and op<>"s" and op<>"n" and op<>"N")
prompt " ESTA SEGURO <S/N> "
for char op attribute (bold)
end while
close window lk
return op
end funcion

# -----#
#
# Función CUANTOS es empleada para la pregunta de eliminación de registros #
# -----#

funcion cuantos()
define
op:char(1)

let op=""
open window lk at 8,12 with 3 rows,50 columns attribute(border,prompt line 2)
while (op<>"T" and op<>"Y" and op<>"A" and op<>"n" and op<>"N" and op<>"n")
prompt "IMPRIMIR: T-> Todos A-> Actual N-> Ninguno "
for char op
end while

```

```

close window lo
return op
end function

function muestra()
define i smallint
for i=1 to 15
let des(i)=""
end for
end function

```

```

-----
#
# Función MEN_VER() construye un menú vertical requiere de los arreglos #
# globales des y opc #
# a través de un input by name sobre una forma con 3 campos de tipo char() #
# desplega lo que contenga el arreglo des y regresa la posición del #
# arreglo al oprimir ESC o alguna de las letras contenidas en el arr opc #
#
#

```

```

funcion men_ver(op,col,ren,titulo,subtitulo)
DEFINE nop SMALLINT, # numero de opciones en el menu
col SMALLINT, # Columna de despliegue
ren SMALLINT, # Rowlen de despliegue para el menu
titulo CHAR(48), # Titulo para el menu
subtitulo CHAR(48), # Sub-Titulo para el menu no implementado
i SMALLINT, # indice
j SMALLINT, # Var auxiliar
opac SMALLINT, # Opcion actual para referenciar el arr.
opac SMALLINT, # Opcion anterior para referenciar el arr.
op1 CHAR(1), # Referencia con la forma campo a
op2 CHAR(1), # Referencia con la forma campo b
op3 CHAR(1), # Referencia con la forma campo c
ancho SMALLINT, # Ancho total del menu con el recuadro
cadena SMALLINT, # Ancho total del menu con el recuadro
auxiliar SMALLINT # Almacena opciones extras como Ctrl-p

```

```

OPTIONS ACCEPT KEY control-y

```

```

INITIALIZE opac TO NULL
let ancho = 0
let cadena = 0

```

```

FOR i = 1 TO nop
let j = ren - i
let cadena = LENGTH(des[i])
display " ",des[i] CLIPPED AT j,col

if (cadena > ancho) then
let ancho = cadena
end if

```

```

end FOR

```

```

CALL _fj_draw_bot(nop - 2,ancho - 3,ren,col - 1)
CALL _fj_draw_bot(nop - 2,ancho - 6,ren,col - 1)

```

```

let auxiliar = 0

```

```

INPUT BY NAME op1,op2,op3

```

```

BEFORE FIELD op1
CLEAR op1
if (opac IS NULL) then
let opac = 1
let opan = 1
let j = ren = opac
display " ",des[1] clipped at j,col aux:bot(reverse)
NEXT FIELD op2
do
let opac = opac
let opac = opac - 1
if (opac < 1) then
let opac = nop
end if
let j = ren = opac
display " ",des[opac] clipped at j,col
let j = ren = opac
display " ",des[opac] clipped at j,col aux:bot(reverse)
NEXT FIELD op2
end if

```

```

AFTER FIELD op1

```

```

CLEAR op1

AFTER FIELD op2
CLEAR op2
FOR i = 1 to nop STEP 1
if (op2 = opci1) then
let opam = opac
let opac = i
EXIT INPUT
end if
end FOR

if (op2 <> " ") then
let op2 = ""
ERROR ""
NEXT FIELD op2
end if

BEFORE FIELD op3
CLEAR op3
let opam = opac
let opac = opac + 1
if (opac=acp) then
let opac = 1
end if

let j = res + opam
display "" .des[opam] clipped at j.col
let j = res + opac
display "" .des[opac] clipped at j.col attribute(reverse)
NEXT FIELD op2

AFTER FIELD op3
CLEAR op3

ON KEY (control-p)
let auxiliar=200
exit input
end (INPUT)

if (set_flag = true) then
let set_flag = FALSE
return (100)
end if

let j = res + opam
display "" .des[opam] clipped at j.col
let j = res + opac
display "" .des[opac] clipped at j.col attribute(reverse)

if (auxiliar >0) then
return (auxiliar)
else
return (opac)
end if

end function

```

```

*
*
* Función AYL_CENTRO  despliega en la forma de soporte sop213 el nombre
* de la persona que atiende el problema
*
*

```

```

function ayla_centro()
define reg array[600] of
record ch_ centro integer,
nom_ centro char(50)
end record.
; unalint

options accept key esc

open window ayla_centro at 5.5
with form "for00" attribute (border)

declare c_ayla cursor for
select ch_ centro,nom_ centro
from centro
order by ch_ centro

let i=1
foreach c_ayla into reg(i),*
let i=i+1
if i>600 then
exit foreach
end if
end foreach
call set_cursor(i)

```

```

display array reg to w_reg *

let v=arr_curr()
close window eye_con
return reg[i] chv_centro

end function

function display_con(clave1,x,y)

define
  x,y,clave1 integer,
  nombre char(60)

  select nom_centro into nombre
  from centro
  where chv_centro = clave1
  display nombre at x,y

end function

function eye_con(centro)
define reg array(600) of
  record chv_ano integer,
  nom_ano char(50)
  end record,
  centro integer,
  i smallest

options accept key esc

open window eye_con at 5,5
  with form "for00a" attribute (border)

display " Centro -> " ,centro at 4,35
declare c_ayal cursor for
  select chv_ano,nom_ano
  from centro
  where chv_centro=centro or chv_centro=9999
  order by chv_ano

let i=1
foreach c_ayal into reg[i] *
  let i=i+1
  if i=600 then
    c_x foreach
    end if
  end foreach
  call set_cursor(i-1)
  display array reg to w_reg *

let v=arr_curr()
close window eye_con
return reg[i] chv_ano

end function

function display_ano(clave1,x,y)

define
  x,y,clave1 integer,
  nombre char(60)
  select nom_ano into nombre
  from centro
  where chv_ano = clave1
  display nombre at x,y

end function

function eye_apli()
define reg array(300) of
  record chv_apli integer,
  nom_apli char(50)
  end record,
  i smallest

options accept key esc

open window eye_apli at 5,3
  with form "for00b" attribute (border)

declare c_ayal2 cursor for
  select chv_apli,nom_apli
  from aplicacion
  order by chv_apli

let i=1

```

```

foreach c_ayn2 into reg1)*
  let i=i-1
  if i<300 then
    exit foreach
  end if
end foreach
call set_cursor(i-1)
display array reg to w_req.*

let i=nr_curr()
close window aya_aph
return reg1.chv_aph

end function

function display_aph(clave1,x,y)
define
  x,y,clave1 integer,
  nombre char(60)
  select nom_aph into nombre
  from aplicacion
  where chv_aph = clave1
  display nombre at x,y
end function

function aya_ego()
define reg array(300) of
  record chv_equipo integer,
  nom_ego char(50)
  end record,
  i unsigned
options accept key esc

open window aya_ego at 5,5
with form "for00c" attribute (border)

declare c_ayn3 cursor for
  select chv_equipo,nom_ego
  from equipo
  order by chv_equipo

let i=1
foreach c_ayn3 into reg1)*
  let i=i-1
  if i<300 then
    exit foreach
  end if
end foreach
call set_cursor(i-1)
display array reg to w_req.*

let i=nr_curr()
close window aya_ego
return reg1.chv_equipo

end function

function display_ego(clave1,x,y)
define
  x,y,clave1 integer,
  nombre char(60)
  select nom_ego into nombre
  from equipo
  where chv_equipo = clave1
  display nombre at x,y
end function

function aya_resp()
define reg array(300) of
  record chv_resp integer,
  nom_resp char(50),
  aya char(50)
  end record,
  i unsigned
options accept key esc

open window aya_resp at 3,3
with form "for00d" attribute (border)

declare c_ayn4 cursor for

```



```

select ch_resp.nombre_area
from responsable
order by ch_resp

let r=1
foreach c_aym4 into reg(1) *
let r=r+1
if r>300 then
exit foreach
end if
end foreach
call set_count(+1)
display array reg to sr_reg.*

let r=arr_cursor()
close window aya_resp
return reg(1).ch_resp

end function

function aya_crec()
define reg array(300) of
record ch_resp integer,
nom_resp char(50),
area char(50)
end record,
i smaltint
options accept key esc

open window aya_resp at 3,3
with form "for004" attributes (border)

declare c_aym5 cursor for
select ch_resp.nombre_area
from responsable
where ch_resp=499 and ch_resp<9999
order by ch_resp

let r=1
foreach c_aym5 into reg(1) *
let r=r+1
if r>300 then
exit foreach
end if
end foreach
call set_count(+1)
display array reg to sr_reg.*

let r=arr_cursor()
close window aya_resp
return reg(1).ch_resp

end function

function display_resp(chrv1,x,y)
define
x,y,tlevel integer,
nombre char(60)
select responsable.nombre into nombre
from responsable
where ch_resp = chrv1
display nombre at x,y
end function

function valida()
define nombre char(200)

options
prompt line 3

open window aya at 15,15
with form "for12a" attributes(border)
menu "SALIDA."
command "Archivo" "Salida a un archivo"
prompt "NOMBRE DEL ARCHIVO > "
for nombre
on key (control-c)
let nombre=null
end prompt
if nombre is not null then
let nombre="cp /tmp/tali "nombre
run (nombre)
call mesajec(" REPORTE ENVIADO ")

```

```

end if
command "Impresora" "Reporte enviado a impresora"
run ("lp -mp:ah")
call mensaje(" REPORTE ENVIADO A IMPRESORA")
command "Pantalla" "Reporte enviado a pantalla"
run ("clear:pg:mp:ah")
command "Terminar"
exit menu
end menu
close window menu
end function

function fin_pap()
clear form
display " " at 7,27
display " " at 9,27
display " " at 11,27
display " " at 13,27
display " " at 15,27
end function

funcion l_r(x,y)
define x,y integer
display " " at x,y
end function

database help_desk

globals
"globals.4gl"

funcion al_cen()
define
op integer,
r record like cenro.*

OPEN window al_cen at 4,4
WITH FORM "form" attributes (border) # ventana para alta de centros de computo

let opr=1
realize r " to null
select max(ch_cenro) into r.ch_cenro
from cenro
where ch_cenro<-9999
let r.ch_cenro=r.ch_cenro+1

input by name r.* without defaults

after field ch_cenro
select ch_cenro into r.ch_cenro
from cenro
where ch_cenro=r.ch_cenro

if r.ch_cenro is null or status="NOTFOUND" then
call mensaje(" VALOR INCORRECTO VERIFIQUE")
next field ch_cenro
end if

after field nom_cenro
if r.nom_cenro is null then
call mensaje(" SE REQUIERE TECLEAR EL NOMBRE")
next field nom_cenro
end if

on key (control-p)
let opr=0
exit input

after field satisfal
next field ch_cenro

end input

close window al_cen
if not flag=TRUE or opr=0 then
let int_flag=false
call mensaje(" ALTA CANCELADA")
return 9999
else
insert into cenro values(r.*)
call mensaje(" REGISTRO INSERTADO")
return r.ch_cenro
end if
end function

```

```

funcion al_usa(ccarro)

define
  ccarro integer,
  op integer,
  r record like usuario.*

OPEN window al_usa at 4,4
WITH FORM "for01a" attribute (border)

let op=1
initialize r * to null
select max(chv_usa) into r.chv_usa
from usuario
where chv_usa<>9999
let r.chv_usa=r.chv_usa+1
let r.chv_ccarro=ccarro

input by name r.* without defaults

after field chv_usa
select chv_usa into r.chv_usa
from usuario
where chv_usa=r.chv_usa

if r.chv_usa is null or status='NOTFOUND' then
  call message(" VALOR INCORRECTO VERIFIQUE")
next field chv_usa
end if

after field nom_usa
if r.nom_usa is null then
  call message(" SE REQUIERE TECLEAR EL NOMBRE")
next field nom_usa
end if

on key (control-p)
let op=0
exit input

end input

close window al_usa
if not flag=exit or op=0 then
  let not_flag=false
  call message(" ALTA CANCELADA")
  return 9999
the
  insert into usuario values(r.*)
  call message(" REGISTRO INSERTADO")
  return r.chv_usa
end if
end function

```

```

funcion al_apli()

define
  op integer,
  r record like aplicacion.*

OPEN window al_apli at 4,4
WITH FORM "for01b" attribute (border)

let op=1
initialize r * to null
select max(chv_apli) into r.chv_apli
from aplicacion
where chv_apli<>9999
let r.chv_apli=r.chv_apli+1
input by name r.* without defaults

after field chv_apli
select chv_apli into r.chv_apli
from aplicacion
where chv_apli=r.chv_apli

if r.chv_apli is null or status='NOTFOUND' then
  call message(" VALOR INCORRECTO VERIFIQUE")
next field chv_apli
end if

after field nom_apli
if r.nom_apli is null then
  call message(" SE REQUIERE TECLEAR EL NOMBRE")
next field nom_apli
end if

after field desc_apli
next field chv_apli

```

```

on key (control-p)
  let op= 0
  exit input

end input

close window al_apli
if not flag=true or op= 0 then
  let not_flag=false
  call message(" ALTA CANCELADA")
  return 9999
else
  insert into aplicacion values(r *)
  call message(" REGISTRO INSERTADO")
  return r.ch_apli
end if
end function

```

```

funcion al_ogpo()
define
  op integer,
  r record like equipo.*
OPEN window al_ogpo in 4,4
WITH FORM "for01c" attribute (border)

let op=1
initialize r.* to null
select max(ch_equipo) into r.ch_equipo
from equipo
where ch_equipo<>9999
let r.ch_equipo=r.ch_equipo-1
input by name r.* without defaults

after field ch_equipo
select ch_equipo into r.ch_equipo
from equipo
where ch_equipo=r.ch_equipo

if r.ch_equipo is null or status="NOTFOUND then
  call message(" VALOR INCORRECTO VERIFIQUE")
  next field ch_equipo
end if
after field nom_ogpo
if r.nom_ogpo is null then
  call message(" SE REQUIERE TECLEAR EL NOMBRE")
  next field nom_ogpo
end if

after field des_equipo
next field ch_equipo

on key (control-p)
  let op= 0
  exit input

```

```

end input

close window al_ogpo
if not flag=true or op= 0 then
  let not_flag=false
  call message(" ALTA CANCELADA")
  return 9999
else
  insert into equipo values(r.*)
  call message(" REGISTRO INSERTADO")
  return r.ch_equipo
end if
end funcion

```

```

funcion al_resp()
define
  op integer,
  r record like responsable.*
OPEN window al_resp in 4,4
WITH FORM "for01d" attribute (border)

let op=1
initialize r.* to null
select max(ch_resp) into r.ch_resp

```

```

from responsible
where chv_resp=500
let r.chv_resp=r.chv_resp+1
input by name r, * without defaults

after field chv_resp
select chv_resp into r.chv_resp
from responsible
where chv_resp=r.chv_resp

if r.chv_resp is null or status='NOTFOUND' then
call message(" VALOR INCORRECTO VERIFIQUE")
next field chv_resp
end if

after field nombre
if r.nombre is null then
call message(" SE REQUIERE TECLER EL NOMBRE")
next field nombre
end if

after field estadist
next field chv_resp

on key (control-p)
let op= 0
exit input

end input

close window al_resp
if not _flag=run or op= 0 then
let int_flag=false
call message(" ALTA CANCELADA")
return 9999
else
insert into responsible values(r,*)
call message(" REGISTRO INSERTADO")
return r.chv_resp
end if
end function

function escalar()
defmc
rpt record like reporte,*,
hs record like historico,*,
act,con menor,
op char(1),
condicion char(150),
a,b char(50),
c distance year to minute,
t,hand smaller

openon accept key ESC
locate rpt.desc_det in memory # localizacion del texto en memoria
locate rpt.desc_sol in memory # localizacion del texto en memoria

current window is prin # Hace current la forma mas grande
OPEN window w_modif at 2,2
WITH FORM "for11b" attribute (border)
let band=0
initialize s_query to null
initialize condicion to null
display * *** ESCALAMIENTO DE REPORTES *** at 3,2

declare cur18 scroll cursor for select no_rpt
from reporte
where estado matches "A*"
open cur18
fetch cur18 into act;

if status=workfound then
call message(" NO EXISTEN REPORTES PARA ESCALAMIENTO")
close cur18
else
select count(*) into con from reporte
where estado matches "A*"
call cuenta(con)

let band=1
select * into rpt_no_rpt,rpt_chv_centro,rpt_chv_usu,rpt_chv_apli,rpt_chv_equipo,rpt_no_serie,rpt_fecha_in,rpt_desc_brev,rpt.desc_det,rpt.desc_sol,rpt_estado,rpt_fecha_fin
from reporte
where no_reportes
display by name rpt,*
call display_con(rpt_chv_centro,7,27)
call display_usu(rpt_chv_usu,9,27)
call display_apli(rpt_chv_apli,11,27)

```

```

call display_eqpo(rep.chv.equipo,13,27)
menu "ESCALAR"
command "Seguiente" "Consultar el Seguiente Registro"

fecha next cur 18 into aux

if status="notfound" then
call message(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")

else
select * into rep.no_rep,rep.chv_cesoro,rep.chv_osa,rep.chv_apli,rep.chv_equipo,rep.no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
from repore
where no_rep=aux
display by name rep *
call display_ces(rep.chv_cesoro,7,27)
call display_osa(rep.chv_osa,9,27)
call display_apli(rep.chv_apli,11,27)
call display_eqpo(rep.chv_equipo,13,27)

end if

command "Previo" "Consultar el Registro Previo"

fecha prior cur 18 into aux

if status="notfound" then
call message(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")

else
select * into rep.no_rep,rep.chv_cesoro,rep.chv_osa,rep.chv_apli,rep.chv_equipo,rep.no_serie,rep.fecha_in,rep.desc_brev,rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
from repore
where no_rep=aux
display by name rep *
call display_ces(rep.chv_cesoro,7,27)
call display_osa(rep.chv_osa,9,27)
call display_apli(rep.chv_apli,11,27)
call display_eqpo(rep.chv_equipo,13,27)

end if

command "Estado" "Estado del Reporte en Pantalla"

open window eq at 2,2 with 20 rows,70 columns attribute(prompt line last;border)
if rep.estado[1,1]="A" then
let a="ABIERTO"
end if
if rep.estado[1,1]="C" then
let a="CERRADO"
end if
if rep.estado[1,1]="X" then
let a="CERRADO"
end if
if rep.estado[2,7]="A" then
let b="SOFTWARE"
else
let b="HARDWARE"
end if
display "ESTADO DEL REPORTE " a at 2,4
display "PROBLEMA DE " b at 3,4
declare c_rp cursor for
select responsible,nombre,historico,motivo,historico.fecha_in
from responsable,historico
where historico.chv_resp=responsible.chv_resp and
historico.no_rep=rep.no_rep
open c_rp
let x=6
foreach c_rp into a,b,c
display a,c at x,4
let x=x+1
display b at x,4
let x=x+2
end foreach
prompt "Oprime cualquier tecla para continuar ... ?" for char op
close window eq

command key ("C") "oCalar"
call sys_exit() returning his.chv_resp

open window rtk at 8,12 with 3 rows,50 columns attribute(border,prompt line 2)
prompt "Motivo : "
for his.motivo attribute (bold)
close window rtk
call confirm() returning op
if (op="S" or op="s")then
insert into historico values(rep.no_rep,his.chv_resp,current,NULL,his.motivo)
call message(" REPORTE ESCALADO ")
call option() returning op
if (op="S" or op="s")then
start report sal1 to printer
output to report sal1(rep *)
finish report sal1
call message(" REPORTE ENVIADO A IMPRESORA")

```

```

end if
end if

command "Terminar"
exit menu
end menu
end if

if use_flag then
let use_flag = false
end if

close window w_ajustif
current window es portada
current window es principal
current window es sub_rep
end function

-----#
#                                     #
# Carlos Jiménez González — Sistema de Soporte #
#                                     #
#      — PROGRAMA SOP010.AGL —      #
#                                     #
# Programa PRINCIPAL contiene las siguientes funciones: #
#                                     #
# main()                               #
#                                     #
#                                     #
#-----#

database help desk

globals
"globals.tgl"

main

define
main_op      mainList

define interrupt

OPEN window inicio at 5.10
WITH FORM "for10c" attributes (border)
sleep 3
close window inicio

OPEN window prin at 2.2
WITH FORM "for10" * ventanas con letras y abarros toda la pant.

OPEN window portada at 2.10
WITH FORM "for10a" attributes (border)

OPEN window principal at 7.3
WITH FORM "for10a" * atributos (border) * Ventana para el menu principal

while true
call limpia()

let des(1) = "A) HOJA DE SOPORTE"
let des(2) = "B) REPORTES Y ESTADISTICOS ."
let des(3) = "C) CATALOGOS"
let des(4) = ""
let des(5) = "D) SALIDA"
let opc(1) = "A"
let opc(2) = "B"
let opc(3) = "C"
let opc(4) = ""
let opc(5) = "D"

CALL main_ver(5.2.2,"") returning main_op

case
when main_op = 1
call sop01(1)

when main_op = 2
call sop01(2)

when main_op = 3
call sop01(3)

when main_op = 4
run "des.ksh"

when main_op = 5 or main_op=100
close window principal
exit while

```

```

end case
end while

close window portals
close window prin

end main

-----#
#                                     #
# Carlos Jiménez González -- Sistema de Sopone #
#                                     #
#      PROGRAMAS SOP011.4GL      #
#                                     #
# Programa que contiene completa la opción A #
#                                     #
#                                     #
-----#

database hclp_dcat

globals
"globals 4gl"

funcion sop011()

define
num_op      smallint

OPEN window sub_rep at 10,6
WITH FORM "for11"

while true
call hclp4()

let des(1) = "A) ALTA DE UN REPORTE"
let des(2) = "B) CONSULTAS DE REPORTE"
let des(3) = "C) ESCALAMIENTO DE REPORTE"
let des(4) = "D) MODIFICACIONES A REPORTE"
let des(5) = "E) CERRAR-CANCELAR REPORTE"
let des(6) = ""
let des(7) = "F) TERMINAR"

let opc(1) = "A"
let opc(2) = "B"
let opc(3) = "C"
let opc(4) = "D"
let opc(5) = "E"
let opc(6) = "F"
let opc(7) = "F"

CALL men_1 or(7,2,2,"") returning num_op

case
when num_op = 1
call alta_rep()

when num_op = 2
call consu_rep()

when num_op = 3
call escala()
when num_op = 4
call mod_rep() @ sop001.4gl

when num_op = 5
call cer_rep()

when num_op = 7 or num_op=100 or num_op=200
close window sub_rep
exit while

end case
end while

end funcion

-----#
#                                     #
# Función ALTA  despiega pantalla de trabajo para las actualizaciones #
# al almacenar emplea s_query como global. #
#                                     #
#                                     #
-----#

funcion alta_rep()

define
rep record like repone.*,
response integer,
op      char(1),

```



```

head          waitfirst

openos
accept key ESC

locate rep_desc_det in memory # localizacion del texto en memoria
locate rep_desc_sof in memory # localizacion del texto en memoria

current window is pma # Hace current la ventana mas grande
OPEN window w_alta at 2,2
WITH FORM "for11g" attribute (border)
display " *** ALTA DE UN REPORTE ***" at 3,2

let head=1
select max(no_rep) into rep_no_rep
from reporte
let rep_no_rep=rep_no_rep-1
display rep_no_rep to reporte_no_rep
let rep_fecha_m = current
display rep_fecha_m to fecha_in

wait
rep.chv_centro.rep.chv_usa.rep.chv_apli.rep.chv_equipo.responso.rep.no_serie.rep.desc_brev.e.rep.desc_det
chv_centro.chv_usa.chv_apli.chv_equipo.chv_resp.no_serie.desc_breve.desc_det

after field chv_centro

select chv_centro into rep.chv_centro
from centro
where chv_centro = rep.chv_centro and chv_centro <> 9999

if rep.chv_centro is null or status=NOTFOUND then
call sys_center() returning rep.chv_centro
if rep.chv_centro = 9999 then
call al_center() returning rep.chv_centro
if rep.chv_centro = 9999 then
next field reporte.chv_centro
end if
end if

call display_center(rep.chv_centro,7,27)
display rep.chv_centro to reporte.chv_centro

after field chv_usa

select chv_usa into rep.chv_usa
from usuario
where chv_usa = rep.chv_usa and chv_usa <> 9999
and chv_centro=rep.chv_centro

if rep.chv_usa is null or status=NOTFOUND then
call sys_usuario(rep.chv_centro) returning rep.chv_usa
if rep.chv_usa = 9999 then
call al_usuario(rep.chv_centro) returning rep.chv_usa
if rep.chv_usa = 9999 then
next field chv_usa
end if
end if

call display_usuario(rep.chv_usa,9,27)
display rep.chv_usa to chv_usa

after field chv_apli

select chv_apli into rep.chv_apli
from aplicacion
where chv_apli = rep.chv_apli and chv_apli <> 9999

if rep.chv_apli is null or status=NOTFOUND then
call sys_apli() returning rep.chv_apli
if rep.chv_apli = 9999 then
call al_apli() returning rep.chv_apli
if rep.chv_apli = 9999 then
next field chv_apli
end if
end if

call display_apli(rep.chv_apli,11,27)
display rep.chv_apli to chv_apli

after field chv_equipo

select chv_equipo into rep.chv_equipo
from equipo
where chv_equipo = rep.chv_equipo and chv_equipo <> 9999

if rep.chv_equipo is null or status=NOTFOUND then
call sys_equipo() returning rep.chv_equipo
if rep.chv_equipo = 9999 then
call al_equipo() returning rep.chv_equipo
if rep.chv_equipo = 9999 then

```

```

    next field chr_equipo
  end if
end if
end if
call display_opco(rep.chr_equipo,13,27)
display rep.chr_equipo to chr_equipo

after field chr_resp

select chr_resp into respuesta
from responsable
where chr_resp=respuesta

if respuesta is null or status=NOTFOUND then
call ayn_resp() returning respuesta
if respuesta =9999 then
call al_resp() returning respuesta
if respuesta =9999 then
next field chr_resp
end if
end if
end if
call display_resp(respuesta,15,27)
display respuesta to chr_resp

after field desc_breve
if rep.desc_breve is null then
call message("  TECLÉE UNA DESCRIPCION BREVE")
next field desc_breve
end if

after field desc_det
next field chr_cesaro

on key (control-p)
let band=0
cut input

end input

if not flag=one or band =0 then
let not_flag=false
call message("  ALTA CANCELADA")
else

let asigna=""
open window rt at 8,10 with 3 rows,60 columns attribute(border,prompt,line 2)
while (asigna<>"A" and asigna<>"a" and asigna<>"e" and asigna<>"E")
prompt " EL REPORTE SE ASIGNA A-> APLICACION E->EQUIPO "
for char asigna
end while
close window rt
if asigna="a" or asigna="A" then
let rep.estado="AA"
else
let rep.estado="AE"
end if
insert into reporte values(0, rep.chr_cesaro,rep.chr_usu,rep.chr_apli,rep.chr_equipo,rep.no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,NULL,rep.estado,NULL)
let rep.no_rep=SQLCA.SQLERRD(2)
insert into historico values(rep.no_rep,rep.respuesta,rep.fecha_in,NULL,"Asignacion nivel cero")
call message("  ALTA CONCLUIDA")
let rep.fecha_fin=NULL
let asigna=""

call opcional() returning op
if op="S" or op="s" then
start report alta to printer
output to report alta(rep.*)"
finish report alta
start report alta to printer
output to report alta(rep.*)"
finish report alta
call message("  REPORTE ENVIVADO")
end if
end if

close window w_alta
current window is portada
current window is principal
current window is sub_rep

end funcion
report alta(idm)

define
idm record like reporte.
cesaro char(60),
usu char(60),
apli char(60),
det char(60).

```

run (char 60)

output

left margin 3
right margin 78
top margin 1
bottom margin 1
page length 65

format

```
first page header
print "ISOSA                      Gerencia de Sistemas "
print "                      HELP DESK          Corporativos "
print "                      _____"
* skip to top of page
```

```
page trailer
print "          *** "pagos USING "esr"," ***"
```

```
on every row
skip 1 line
print "No Reporte "adm.no_rep USING "#####"
skip 2 line
print "          ESTADO DEL REPORTE          "
skip 1 line
if adm.estado(1,1)="A" then
  let centro="ABERTO"
end if
if adm.estado(1,1)="C" then
  let centro="CERRADO"
end if
if adm.estado(1,1)="X" then
  let centro="CANCELADO"
end if
print column 2,"Estado " ,centro
print column 2,"Fecha inicio "adm.fecha_in
print column 2,"Fecha Cierre "adm.fecha_fin
skip 2 line
print "          DATOS GENERALES          "
skip 1 line

select nom_centro,direccion,telefono,sucursal into centro.ano,dos,tres
from centro where ch_centro=adm.ch_centro
print column 2,"CENTRO "adm.ch_centro USING "####"," ,centro
print column 16,ano clipped
print column 16,dos clipped

skip 1 line
select nom_usu,puesto,telefono into centro.ano,dos
from usuario where ch_usuario=adm.ch_usuario
print column 2,"USUARIO "adm.ch_usuario USING "####"," ,centro
print column 16,ano clipped
print column 16,dos clipped

select nom_apli,desc_apli into centro,dos
from aplicacion where ch_apli=adm.ch_apli
skip 1 line
print column 2,"APLICACION "adm.ch_apli USING "####"," ,centro
print column 16,dos
skip 1 line
select nom_equipo into centro
from equipo where ch_equipo=adm.ch_equipo
print column 2,"EQUIPO "adm.ch_equipo USING "####"," ,centro
print column 16,adm.no_serie

skip 2 line
print "          DATOS DEL RESPONSABLE          "
skip 1 line

select nombre,puesto,area,telefono into centro.ano,dos,tres
from responsable where ch_resp in
(select unique ch_resp from historico where adm.no_rep=no_rep)
if adm.estado(2,2)="A" then
  print column 2,"PROBLEMA DE SOFTWARE "
else
  print column 2,"PROBLEMA DE HARDWARE "
end if
skip 1 line
print column 2,"ASIGNACION " ,centro
print column 16,ano
print column 16,dos
print column 16,tres
skip 3 line
print "          DESCRIPCION DEL REPORTE          "
skip 1 line
print column 2,"DESCRIPCION BREVE :?"
skip 1 line
print column 2,adm.desc_breve
```

```

skip 1 line
print column 2,"DESCRIPCION DETALLADA :*"
skip 1 line
print column 2,adm.desc_det wordwrap right margin 70

end report

funcion consu_rep()

define
rep record like reporte.*,
rep1 record like reporte.*,
aaa integer,
op char(1),
prs char(250),
con smallest,
condicion char(250),
a,b char(50),
c distance year to month,
s,head smallest

options accept key ESC
otherwise error continue

locate rep1.desc_det in memory # localización del texto en memoria
locate rep1.desc_sol in memory # localización del texto en memoria
locate rep.desc_det in memory # localización del texto en memoria
locate rep.desc_sol in memory # localización del texto en memoria

current window is prin # Hace current la forma aaa grande
OPEN window w_modif at 2,2
WITH FORM "fill" attributes (border)
is hand=0
simulate s_query to null
simulate condicion to null
display * *** CONSULTA DE REPORTES *** at 3,2

menu "CONSULTAS"
command "Consulta"
call sim_pant()
construct
reporte.ao_rep,reporte.ch_cestro,reporte.ch_usa,reporte.ch_apli,reporte.ch_equipo,reporte.ao_serie,reporte.fecha_in,reporte.desc_breve,reporte.estado,reporte.fecha_fin
ao_rep,reporte.ch_cestro,reporte.ch_usa,reporte.ch_apli,reporte.ch_equipo,ao_serie,fecha_in,desc_breve,estado,fecha_fin
on key (control-y)
on state
end construct

if aa_flag=true then
let aa_flag=false
call message(" CONSULTA CANCELADA")
continue menu
end if

let s_query="select no_rep from reporte where ",condicion clipped
prepare s11 from s_query
declare cur11 scroll cursor for s11
open cur11
fetch cur11 into aa1

if state=notfound then
call message(" NO EXISTE INFORMACION CON ESA CONDICION")
close cur11
continue menu
end if

let prs="select count(*) from reporte where ",condicion clipped
prepare r_count from prs
declare scur scroll cursor for r_count
open r_count
fetch scur into con
call counta(con)
close scur

let hand=1
select * into rep.ao_rep,rep.ch_cestro,rep.ch_usa,rep.ch_apli,rep.ch_equipo,rep.ao_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
from reporte
where no_rep=aaa
display by name rep.*
call display_cons(rep.ch_cestro,7,27)
call display_usa(rep.ch_usa,9,27)
call display_apli(rep.ch_apli,11,27)
call display_equipo(rep.ch_equipo,13,27)

command "Signific" "Consultar el Significativo Registro"

if hand=0 then
call message("PRIMERO DEBE REALIZAR UNA CONSULTA")
continue menu
end if

```

```

fetch next cur11 into aux
if status=notfound then
call mensaje(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")

else
select * into rep_no_rep,rep.chv_control,rep.chv_usa,rep.chv_apli,rep.chv_equipo,rep.no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
from reporte
where no_rep=aux
display by name rep.*
call display_com(rep.chv_control,7,27)
call display_usa(rep.chv_usa,9,27)
call display_apli(rep.chv_apli,11,27)
call display_eqpo(rep.chv_equipo,13,27)
end if

command "Previo" "Consultar el Registro Previo"
if found=0 then
call mensaje(" PRIMERO DEBE REALIZAR UNA CONSULTA")
continue menu
end if

fetch prior cur11 into aux
if status=notfound then
call mensaje(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")
else
select * into rep_no_rep,rep.chv_control,rep.chv_usa,rep.chv_apli,rep.chv_equipo,rep.no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
from reporte
where no_rep=aux
display by name rep.*
call display_com(rep.chv_control,7,27)
call display_usa(rep.chv_usa,9,27)
call display_apli(rep.chv_apli,11,27)
call display_eqpo(rep.chv_equipo,13,27)
end if

command "Estado" "Estado del Reporte en Pantalla"
if found=0 then
call mensaje("PRIMERO DEBE REALIZAR UNA CONSULTA")
continue menu
else
open window q at 2,3 with 20 rows,70 columns attribute( prompt line last,border)
if rep.estado(1,1)="A" then
let a="ABIERTO"
end if
if rep.estado(1,1)="C" then
let a="CERRADO"
end if
if rep.estado(1,1)="X" then
let a="CERRADO"
end if
if rep.estado(2,2)="A" then
let b="SOFTWARE"
else
let b="HARDWARE"
end if
display "ESTADO DEL REPORTE "a at 2,4
display "PROBLEMA DE "b at 3,4
declare c _rop cursor for
select responsable.nombre,historico.motivo,historico.fecha_in
from responsable,historico
where historico.chv_resp=responsible.chv_resp and
historico.no_rep=rep.no_rep

open c _rop
let x=6
foreach c _rop into a,b,c
display a,c at x,4
let a=a+1
display b at x,4
let x=x+2
end foreach
prompt "Oprima cualquier tecla para continuar ...?" for char op
close window q
end if

command key("X") "eXo" "Consulta del texto V Esta Opcion no Realiza Actualizaciones al Campo"
if found=0 then
call mensaje(" PRIMERO DEBE REALIZAR UNA CONSULTA")
continue menu
end if

input rep.desc_det,rep.desc_sol without defaults from desc_det,desc_sol
on key (control-p)
exit menu
end aspt

```

```

if not _flag=true then
  let int_flag=false
end if

command "Impresora"
  if base=0 then
    call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  command menu
  end if
  call command() returning op
  if (op="T" or op="Y")then
    let s_query="Select * from reporte where "condicion clipped
    prepare sp from s_query
    declare c_s cursor for sp
    open c_s
    start report sal1 to printer

    foreach c_s into rep1.*
      output to report sal1(rep1.*)
    end foreach
    finish report sal1
    call message(" REPORTE ENVIADO A IMPRESORA")
  end if

  if (op="A" or op="A")then
    start report sal1 to printer
    output to report sal1(rep.*)
    finish report sal1
    call message(" REPORTE ENVIADO A IMPRESORA")
  end if

command "Terminar"
  exit menu
end menu

if not _flag then
  let int_flag=false
end if

close window w_modif
current window is portada
current window is principal
current window is sub_rep
end function

report sal1(admin)

define
  adm record like reporte,
  cover char(60),
  uno char(60),
  dos char(60),
  tres char(60),
  max integer,
  tiempo datetime year to minute

output
  left margin 2
  right margin 78
  top margin 2
  bottom margin 2
  page length 66

format

page header
  print "ISOSA"
  print "HELP DESK"
  print "Gerencia de Sistemas"
  print "Cooperativas"

page trailer
  print "*** pagano USING ***"

on every row
  skip 1 line
  print "No Reporte ",adm.no_rep USING "####"

  skip 2 line
  print "ESTADO DEL REPORTE"
  skip 1 line
  if adm.estado(1,1)="A" then
    let control="ABIERTO"
  end if
  if adm.estado(1,1)="C" then
    let control="CERRADO"
  end if
  if adm.estado(1,1)="X" then
    let control="CANCELADO"
  end if

```

```

end if
print "Estado " ,column 16,centro
print "Fecha Inicio " ,column 16,ads.fecha_in
print "Fecha Cierre " ,column 16,ads.fecha_fin

skip 2 line
print "          DATOS GENERALES          "
skip 1 line
select nom_centro,direccion,telefono,sitio into centro,mo,dos,tres
from centro where chv_centro=ads.ch_centro
print column 2,"CENTRO " ,ads.chv_centro USING "####",",centro
print column 16,mo clipped
print column 16,dos clipped,tres clipped

skip 1 line
select nom_usu,puesto,telefono into centro,mo,dos
from usuario where chv_usu=ads.chv_usu
print column 2,"USUARIO " ,ads.chv_usu USING "####",",centro
print column 16,mo clipped
print column 16,dos clipped

select nom_apli into centro
from aplicacion where chv_apli=ads.chv_apli
skip 1 line
print column 2,"APLICACION " ,ads.chv_apli USING "####",",centro
skip 1 line
select nom_eqpo into centro
from equipo where chv_eqpo=ads.chv_eqpo
print column 2,"EQUIPO " ,ads.chv_eqpo USING "####",",centro

skip 2 line
print "          DATOS DEL RESPONSABLE          "
skip 1 line

if ads.estado(2,2)="A" then
    print column 2,"PROBLEMA DE SOFTWARE "
else
    print column 2,"PROBLEMA DE HARDWARE "
end if

skip 1 line
print column 2,"ASIGNACION : "
skip 1 line
declare c_rep cursor for
select responsable,asobire,responsable,area,historico,motivo,historico,fecha_in
from historico where responsable
where historico.no_rep=ads.no_rep
and historico.chv_resp=responsable.chv_resp

open c_rep
foreach c_rep into centro,mo,dos,siempo
    print column 2,centro clipped," ",mo
    print column 2,dos clipped," ",siempo

    skip 1 line
end foreach

skip 2 line
print "          DESCRIPCIÓN DEL REPORTE          "
skip 1 line
print column 2,"DESCRIPCIÓN BREVE : "
skip 1 line
print column 2,ads.desc_breve
skip 1 line
print column 2,"DESCRIPCIÓN DETALLADA : "
skip 1 line
print column 2,ads.desc_det wordwrap right margin 70
skip to top of page

skip 2 line
print "          DESCRIPCIÓN DE SOLUCION          "
skip 2 line
print column 2,ads.desc_sol wordwrap right margin 60
skip to top of page

end report

function mod_rep()

define
    rep record like reporte ,
    mo smallint,
    x_cdo char(1),
    ed char(2),
    band smallint,
    responsa,back integer

options
    accept key ESC

let band=0
locate rep desc_det in memory # localización del texto en memoria

```

```

locate rep_desc_sol in memory # localizacion del texto en memoria

current window in print # Hace current la forma mas grande
OPEN window w_mod at 2,2
WITH FORM "for114" attribute (border)
display " *** ACTUALIZACIONES A REPORTES ***" at 3,2

menu "ACTUALIZAR"
  command "Consulta"
  call fun_pant()
  construct
    s_query
    repno:ao_rep:repno:chv_centro:repno:chv_usu:repno:chv_apli:repno:chv_equipo:responsable:chv_resp:repno:ao_serie:repno.fecha_in:repno.desc_breve:repno.estado from
    no_rep:chv_centro:chv_usu:chv_apli:chv_equipo:chv_resp:ao_serie.fecha_in:desc_breve:estado
    on key (control-p)
    exit menu
  end construct

  if not flag=rep then
    let not_flag=false
    call message(" CONSULTA CANCELADA")
    continue menu
  end if

  let s_query="select no_rep from repno where " & s_query clipped
  prepare z11 from s_query
  declare c1 scroll cursor for z11
  open c1
  fetch c1 into mx1

  if status="eofeof" then
    call message(" NO EXISTE INFORMACION CON ESA CONDICION")
    close c1
    continue menu
  end if

  let band=1
  select
    no_rep:chv_centro:chv_usu:chv_apli:chv_equipo:ao_serie:fecha_in:desc_breve:desc_det:estado into
  repno:rep:chv_centro:rep:chv_usu:rep:chv_apli:rep:chv_equipo:repno:serie:rep:fecha_in:rep:desc_breve:rep:desc_det:rep:estado
  from repno
  display
  no_rep:chv_centro:chv_usu:chv_apli:chv_equipo:ao_serie:fecha_in:rep:desc_breve:rep:desc_det:rep:estado to

  select unique chv_resp into respuesta
  from historico where rep:ao_rep=no_rep
  display respuesta to chv_resp
  call display_cen(rep:chv_centro,7,27)
  call display_usu(rep:chv_usu,9,27)
  call display_apli(rep:chv_apli,11,27)
  call display_eqpo(rep:chv_equipo,13,27)
  call display_resp(respuesta,15,27)

  command "Siguenste" "Consultar el Siguenste Registro"
  if band=0 then
    call message("PRIMERO DEBE REALIZAR UNA CONSULTA")
    continue menu
  end if

  fetch next c1 into mx1

  if status="eofeof" then
    call message(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")
  else
    select
      no_rep:chv_centro:chv_usu:chv_apli:chv_equipo:ao_serie:fecha_in:desc_breve:desc_det:estado into
    rep:ao_rep:rep:chv_centro:rep:chv_usu:rep:chv_apli:rep:chv_equipo:rep:ao_serie:rep:fecha_in:rep:desc_breve:rep:desc_det:rep:estado
    from repno
    where no_rep=mx1

    display
    no_rep:chv_centro:chv_usu:chv_apli:chv_equipo:ao_serie:fecha_in:desc_breve:desc_det:estado to

    declare c_smp scroll cursor for
    select chv_resp
    from historico where rep:ao_rep=no_rep

    open c_smp
    fetch first c_smp into respuesta
    display respuesta to chv_resp
    close c_smp
    call display_cen(rep:chv_centro,7,27)
    call display_usu(rep:chv_usu,9,27)
    call display_apli(rep:chv_apli,11,27)
    call display_eqpo(rep:chv_equipo,13,27)
    call display_resp(respuesta,15,27)

    end if

  command "Previo" "Consultar el Registro Previo"

```



```

if band=0 then
  call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if

fetch prior c1 into ans

if status=undefined then
  call message(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")
else
  select
    no_rep.chv_estado,chv_usa,chv_apli,chv_equipo,no_serie,fecha_in,desc_breve,det_estado
  rep.no_rep,rep.chv_estado,rep.chv_usa,rep.chv_apli,rep.chv_equipo,rep.no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep_estado
  from reporte
  where no_rep=ans
  display
    rep.no_rep,rep.chv_estado,rep.chv_usa,rep.chv_apli,rep.chv_equipo,rep.no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep_estado
  no_rep.chv_estado,chv_usa,chv_apli,chv_equipo,no_serie,fecha_in,desc_breve,det_estado
  10

declare c_esp scroll cursor for
select chv_resp
from historico where rep.no_rep=no_rep

open c_esp
fetch first c_esp into respuesta
display respuesta to chv_resp
close c_esp
call display_con(rep.chv_estado,7,27)
call display_usa(rep.chv_usa,9,27)
call display_apli(rep.chv_apli,11,27)
call display_equipo(rep.chv_equipo,13,27)
call display_resp(respuesta,15,27)

end if

command key("X") "cXto" "Consultas del texto" " Esta Opcion no Realiza Actualizaciones al Campo"

if band=0 then
  call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if
input rep.desc_det without defaults from desc_det
on key (control-p)
exit menu
end input

if int_flag=menu then
  let int_flag=false
end if

command "Modificar"

let back= respuesta
let edo= rep_estado(1,1)
if band=0 then
  call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if
input
  rep.chv_estado,rep.chv_usa,rep.chv_apli,rep.chv_equipo,respuesta,rep.no_serie,rep.desc_breve,rep.desc_det,rep_estado
  chv_estado,chv_usa,chv_apli,chv_equipo,chv_resp,no_serie,dese_breve,dese_det,estado
  without defaults from

  after field chv_estado

select chv_estado into rep.chv_estado
from c_estado
where chv_estado = rep.chv_estado and chv_estado<>9999

if rep.chv_estado is null or status=NOTFOUND then
  call sys_control() returning rep.chv_estado
if rep.chv_estado =9999 then
  call al_con() returning rep.chv_estado
if rep.chv_estado =9999 then
  next field reporte.chv_estado
end if
end if

call display_con(rep.chv_estado,7,27)
display rep.chv_estado so reporte.chv_estado

after field chv_usa

select chv_usa into rep.chv_usa
from usuario
where chv_usa = rep.chv_usa and chv_usa<>9999
and chv_estado=rep.chv_estado

if rep.chv_usa is null or status=NOTFOUND then
  call sys_usa(rep.chv_estado) returning rep.chv_usa
if rep.chv_usa =9999 then
  call al_usa(rep.chv_estado) returning rep.chv_usa
if rep.chv_usa =9999 then

```

```

    next field chr_usa
end if
end if
end if
call display_usa(rep.chv_usa,9,27)
display rep.chv_usa to chr_usa

after field chr_apli

select chr_apli into rep.chv_apli
from aplicacion
where chr_apli = rep.chv_apli and chr_apli<>9999

if rep.chv_apli is null or status=NOTFOUND then
call sys_apli() returning rep.chv_apli
if rep.chv_apli =9999 then
call al_apli() returning rep.chv_apli
if rep.chv_apli =9999 then
next field chr_apli
end if
end if
end if
call display_apli(rep.chv_apli,11,27)
display rep.chv_apli to chr_apli

after field chr_equipo

select chr_equipo into rep.chv_equipo
from equipo
where chr_equipo = rep.chv_equipo and chr_equipo<>9999

if rep.chv_equipo is null or status=NOTFOUND then
call sys_equipo() returning rep.chv_equipo
if rep.chv_equipo =9999 then
call al_equipo() returning rep.chv_equipo
if rep.chv_equipo =9999 then
next field chr_equipo
end if
end if
end if
call display_equipo(rep.chv_equipo,13,27)
display rep.chv_equipo to chr_equipo

after field chr_resp

select chr_resp into respuesta
from respuesta
where chr_resp=respuesta

if respuesta is null or status=NOTFOUND then
call sys_resp() returning respuesta
if respuesta =9999 then
call al_resp() returning respuesta
if respuesta =9999 then
next field chr_resp
end if
end if
end if
call display_resp(respuesta,15,27)
display respuesta to chr_resp

after field desc_breve
if desc_breve is null then
call mensaje(" TECLEA UNA DESCRIPCION BREVE")
next field desc_breve
end if

before field estado
if rep.estado(1,1)~"X" or rep.estado(1,1)~"C" then
next field chr_estado
end if

after field estado
let ed=edo_rep.estado(2,2)
let rep.estado=ed
display ed to estado
next field chr_estado

on key (control-p)
let band= 0
exit input

end input
if int_flag=true or band =0 then
let int_flag=false
call mensaje(" ACTUALIZACION CANCELADA")
else

```

```

update
reporte
(ch_cestro, ch_usu, ch_apli, ch_equipo, no_serie, desc_det, estado) = (rep, ch_cestro, rep, ch_usu, rep, ch_apli, rep, ch_equipo, no_serie, rep, desc_det, rep, estado)
a)
where rep.no_rep=no_rep

update historico set ch_resp=response
where no_rep=no_rep and ch_resp=back

call mensaje(" ACTUALIZACION CONCLUIDA")
call fin_pant()
let band=0
end if
command "Terminate"
end menu
end menu
class window w_mod
current window is portada
current window is principal
current window is sub_rep
end function

function cor_rep()

define
rep record like reporte *,
rep1 record like reporte *,
usu integer,
op char(1),
condicion char(250),
a,b char(50),
c = descomponer year to minute,
a,band modifine

options accept key ESC
Whenever error continue
locate rep1.desc_det in memory # localizacion del texto en memoria
locate rep1.desc_sol in memory # localizacion del texto en memoria
locate rep.desc_det in memory # localizacion del texto en memoria
locate rep.desc_sol in memory # localizacion del texto en memoria

current window is prin # Hace current la forma mas grande
OPEN window w_modif at 2,2
WITH FORM "for11b" attribute (border)
let band=0
initialize s_query to null
initialize condicion to null
display " *** CONSULTA DE REPORTES ***" at 3,2

# command "Consulta"
call fin_pant()
reporte.no_rep, reporte.ch_cestro, reporte.ch_usu, reporte.ch_apli, reporte.ch_equipo, reporte.no_serie, reporte.fecha_in, reporte.desc_det, reporte.estado, reporte.fecha_fin
no_rep, ch_cestro, ch_usu, ch_apli, ch_equipo, no_serie, fecha_in, desc_det, estado, fecha_fin
# on key (control-p)
# exit menu
# end construct
#
# if int_flag=true then
# let int_flag=false
# call mensaje(" CONSULTA CANCELADA")
# continue menu
# end if

# let s_query="select no_rep from reporte where estado matches ('A') and "condicion clipped
# prepare x19 from s_query
# declare cur19 scroll cursor for x19
# declare cur19 scroll cursor for select no_rep from reporte where estado matches "A"
open cur19
fetch cur19 into aux

if status=notfound then
call mensaje(" NO EXISTE REPORTES CON ESTADO DE ABIERTO")
close cur19
continue menu
# end if
# else

let band=1
select * into rep.no_rep, rep.ch_cestro, rep.ch_usu, rep.ch_apli, rep.ch_equipo, rep.no_serie, rep.fecha_in, rep.desc_det, rep.desc_sol, rep.estado, rep.fecha_fin
from reporte
where no_rep=aux
display by name rep.*
call display_cas(rep.ch_cestro, 7, 27)
call display_usu(rep.ch_usu, 9, 27)
call display_apli(rep.ch_apli, 11, 27)
call display_eqpo(rep.ch_equipo, 13, 27)

menu "REPORTES"

```

```

command "Seguiente" "Consultar el Siguiete Registro"

if band=0 then
  call message("PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if

fetch next cur19 into aux

if status=notfound then
  call message(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")
else
  select * into rep_no_rep,rep_civ_centro,rep_civ_usa,rep_civ_apli,rep_civ_equipo,rep_no_serie,rep.fecha_in,rep.desc_breve,c.rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
  from reporte
  where no_rep=aux
  display by name rep.*
  call display_con(rep.civ_centro,7,27)
  call display_usa(rep.civ_usa,9,27)
  call display_apli(rep.civ_apli,11,27)
  call display_equo(rep.civ_equipo,13,27)
end if

```

```

command "Previo" "Consultar el Registro Previo"

if band=0 then
  call message("PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if

fetch prior cur19 into aux

if status=notfound then
  call message(" NO EXISTE MAS INFORMACION EN ESA DIRECCION")
else
  select * into rep_no_rep,rep_civ_centro,rep_civ_usa,rep_civ_apli,rep_civ_equipo,rep_no_serie,rep.fecha_in,rep.desc_breve,rep.desc_det,rep.desc_sol,rep.estado,rep.fecha_fin
  from reporte
  where no_rep=aux
  display by name rep.*
  call display_con(rep.civ_centro,7,27)
  call display_usa(rep.civ_usa,9,27)
  call display_apli(rep.civ_apli,11,27)
  call display_equo(rep.civ_equipo,13,27)
end if

```

```

command "Estado" "Estado del Reporte en Pantalla"

if band=0 then
  call message("PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
else
  open window eq at 2,2 width 20 rows,70 columns attribute(prompt line last, border)
  if rep.estado(1,1)="" then
    let a="ABIERTO"
  end if
  if rep.estado(1,1)="-C" then
    let a="-CERRADO"
  end if
  if rep.estado(1,1)="-X" then
    let a="-CERRADO"
  end if
  if rep.estado(2,3)="" then
    let b="SOFTWARE"
  else
    let b="HARDWARE"
  end if
  display "ESTADO DEL REPORTE " a at 2,4
  display "PROBLEMA DE " b at 3,4
  declare c_rj cursor for
  select responsable,nombre,historico,motivo,historico fecha_in
  from responsable,historico
  where historico.civ_resp =responsable.civ_resp and
  historico.no_rep=rep.no_rep

  open c_rj
  let x=6
  foreach c_rj into a,b,c
    display a,c in x,4
    let x=x+1
  display b at x,4
  let x=x+2
  end foreach
  prompt "Oprime cualquier tecla para continuar ...?" for char op
  close window eq
  end if

```

```

command key("A") "cAnular" "Cancelacion de reportes"
if band=0 then
  call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu

```

```

end if

let rep fecha_fin=current
if rep estado[2,3]="A" then
  let rep estado="XA"
else
  let rep estado="XE"
end if
display rep fecha_fin to fecha_fin
display rep estado to estado
input rep desc_sol from desc_sol
on key (control-p)
  exit menu
after field desc_sol
if rep desc_sol is null then
  call message(" DEBE TECLLEAR EL MOTIVO DE LA CANCELACION")
  menu field desc_sol
end if
end input

if set_flag=true then
  let set_flag=false
  call message(" ACTUALIZACION CANCELADA ")
else
  update reporte set (desc_sol,estado,fecha_fin)=(rep.desc_sol,rep.estado,rep.fecha_fin)
  where no_rep=rep.no_rep
  call message(" REPORTE CANCELADO ")
  call fun_pant()
  let band=0
end if

command key ("R") "ctrl+r"
if (band=0) then
  call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if

if rep estado="AA" then
  let rep estado="CA"
else
  let rep estado="CE"
end if
let rep fecha_fin=current
display rep estado to estado
display rep fecha_fin to fecha_fin
input rep desc_sol from desc_sol
after field rep desc_sol
if rep desc_sol is null then
  call message(" DEBE TECLLEAR LA SOLUCION")
  menu field desc_sol
end if
end input
if set_flag then
  let set_flag=false
  call message(" ACTUALIZACION CANCELADA")
  continue menu
end if
update reporte set (desc_sol,estado,fecha_fin)=(rep.desc_sol,rep.estado,rep.fecha_fin) where no_rep=rep.no_rep
call message(" REGISTRO ACTUALIZADO")
call fun_pant()
let band=0

command "keyprnt"
if band=0 then
  call message(" PRIMERO DEBE REALIZAR UNA CONSULTA")
  continue menu
end if
call cursor() returning op
if (op="T" or op="Y") then
  let s_query="Select * from reporte where ",condicion clipped
  prepare s0 from s_query
  declare s_s cursor for s0
  open s_s
  start report sal to printer

  foreach s_s into rep1 *
    output to report sal(rep1.*)
  end foreach
  finish report sal
  call message(" REPORTE ENVIADO A IMPRESORA")
end if

if (op="a" or op="A") then
  start report sal to printer
  output to report sal(rep.*)
  finish report sal
  call message(" REPORTE ENVIADO A IMPRESORA")
end if

```

```
command "Terminal"  
  class main  
  end main  
  
end if  
if ml_flag then  
  is ml_flag = false  
end if  
  
close window e_modif  
current window is portada  
current window is principal  
current window is sub_rep  
end function
```

5.3 Pruebas de la aplicación.

A continuación se describen las pruebas que se realizaron al sistema:

1. Se dieron de alta 100 reportes, verificándose la unicidad en el folio del reporte, así mismo se probaron todos los módulos del sistema, no encontrándose errores.
2. El sistema cumple con las siguientes características:
 - Presenta ayuda en línea de los catálogos que emplea.
 - Presenta ventanas tipo pull-down y teclas rápidas en su operación.
 - El sistema está programado de forma modular, por lo que su mantenimiento es sencillo
3. Por último se probó la logística del sistema, en cuanto a la forma en que deberá fluir la información del mismo.

VI Conclusiones

El desarrollo del sistema finalmente cumple con las necesidades planteadas en la unidad uno, entre los requerimientos más importantes a mencionar tenemos; el seguimiento y control de las problemáticas presentadas en un centro de cómputo, consecuentemente se tiene la solventación eficiente de estas problemáticas, y en un punto más estadístico se conoce si un producto de software como el sistema operativo, la base de datos o software de comunicaciones (software de aplicación), presenta errores frecuentes para en su caso tomar las medidas necesarias con los proveedores y que estos proporcionen la solución, así mismo, se tiene una forma de medir la calidad de las áreas de desarrollo que proporcionan sistemas a la medida para la Secretaría de Hacienda y Crédito Público, como se ha mencionado con anterioridad, las aplicaciones con mayor importancia son los sistema que da seguimiento a los impuestos interiores (SIR Sistema Integral de Recaudación), y el sistema que controla las mercancías que entran, salen o transitan por el país (SAAI Sistema Automatizado Aduanero Integral), por otra parte se puede identificar de forma rápida la causa por la cual un centro de cómputo se encuentra detenido, lo anterior con la finalidad de hacer estimaciones en el tiempo que puede tardar solventar el problema y reestablecer la operativa del centro de cómputo en su totalidad, es decir, el "Sistema de Ayuda" es una herramienta de apoyo en la toma de decisiones, más aun cuándo los problemas presentados en un centro de cómputo son críticos y se debe decidir si optar por una operación en contingencia (sistemas alternos en el centro de cómputo, que en ocasiones no existen y la operación es manual), o una espera en la solución de la problemática, el "Sistema de Ayuda" cuenta con la información necesaria para tomar una alternativa correcta.

Como se observa de capítulos anteriores el esquema de operación para el "Sistema de Ayuda Nacional", es a través de terminales y en un ambiente texto con una Base de Datos centralizada, esto es un área de oportunidad para una futura versión, así mismo se puede incrementar la funcionalidad del Sistema contemplando los aspectos siguientes:

- Envío automático de Reportes a las diferentes áreas de soporte (así también la retroalimentación de soluciones o reasignaciones), el esquema debe plantear el manejo de una firma electrónica que permita definir cuando un usuario del sistema ha recibido y leído un mensaje.

- **Contratación de radiolocalizadores y asignación de los mismos a personal estratégico en la solución de problemas, siendo factible ligar la alta de un reporte con carácter de urgente a un envío de mensajes al radiolocalizador.**
- **Con una base de datos histórica es posible establecer patrones de fallas en hardware y comunicaciones.**
- **Siendo un punto crucial el medio de comunicaciones es posible establecer un monitor de red y el servidor que contiene las altas de reportes, para que en el momento de detectar una falla en la comunicación con los centros de cómputo conectados, automáticamente iniciar un reporte.**

Manual de Usuario

En esta sección se presentan las pantallas y principales acciones de operación, como se observa la pantalla de presentación es la siguiente:

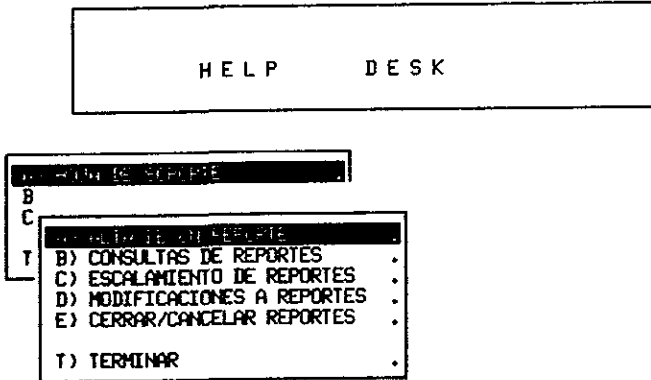


La pantalla principal es la que a continuación se presenta, en términos generales la opción A) relaciona las acciones con alta, reasignaciones, actualizaciones y cierre de Solicitudes de Ayuda, la opción B) contiene las estadísticas y reportes que el sistema proporciona y por último la opción C) contiene las opciones de mantenimiento a catálogos.



<< Ctrl-c >> Cancelar << ESC >> Aceptar (o letra inicial en el menu)
<< Ctrl-p >> Pantalla anterior

A continuación se presentan las opciones de altas de reportes las cuales se explicarán mas adelante en detalle. Como observamos de las últimas líneas de esta pantalla se tienen definidas teclas (Ctrl-p, Ctrl-c y ESC) las cuales están activas en todo momento en el sistema.



<< Ctrl-c >> Cancelar << ESC >> Aceptar (o letra inicial en el menú)
<< Ctrl-p >> Pantalla anterior

Opción A.A), la siguiente ventana "ALTA DE REPORTE", se destacan los datos que deben ser capturados en la alta de una solicitud de ayuda (SA). Los datos número de reporte y fecha de alta son asignados automáticamente por el sistema.

```

*** ALTA DE UN REPORTE ***
Número de Reporte [ 20 ] Fecha [ 1998-03-13 21:00 ]
Clave del Centro [ ]
Clave del Usuario [ ]
Clave Aplicación [ ]
Clave del Equipo [ ]
Responsable [ ]
No serie Equipo [ ]
Desc. Breve [ ]
Desc. Detallada [ ]
Clave para el Centro de Cómputo [ ]

```

Para los casos en los cuales datos como la clave de usuarios, número de centro de cómputo, o cualquier otro tipo de clave, el sistema proporciona una ayuda en línea, la cual es desplegada con una ventana que se sobrepone a la ventana de origen, para los campos que sea indispensable digitar, si el dato no es correcto o no se tecléa nada, automáticamente se despliega esta ventana de ayuda, la siguiente pantalla ejemplifica una ayuda del catálogo de usuarios.

```

Nú [ 00 ]
Cl Catálogo de USUARIOS Centro --> 1
Cl Clave Nombre
Cl [ 1 ] ELIC, SERGIO GARCIA ]
Cl [ 2 ] EHECTOR GUADARRAMA ]
Cl [ 3 ] [ARMANDO VALENCIA BELTRAN ]
Cl [ 4 ] [BENJAMIN VILLASANA ]
Re [ 5 ] [LETICIA JIMENEZ ]
No [ 6 ] [RAUL DOMINGUEZ ]
De Presiona la tela ESC para seleccionar un registro ]
Desc. Detallada [ ]
Clave para el USUARIO [ ]

```

Con la finalidad de no restringir la descripción de la problemática, el sistema fue diseñado para capturar esta información en texto, el cual puede ser tan amplio como un libro, para acceder este campo estando en él se debe digitar "!", en este momento se desplegará el editor default de unix que (en esta caso es vi), una vez concluida la captura de la problemática y proporcionando las opciones de guardar texto, el control del proceso regresa al sistema, como se muestra en la siguiente figura:

```

*** ALTA DE UN REPORTE ***
Número de Reporte [ ] Fecha [1992-03-13 11:00]
Clave del Centro [ 1]CENTRO DE PROCESAMIENTO NACIONAL
Clave del Usuario [ 2]DIRECTOR GUADARRAMA
Clave Aplicación [ 2]MS-OFFICE
Clave del Equipo [ 1]HP 9000/ SERIE 800
Responsable [ 3]SILVESTRE HDEZ.
No serie Equipo [45F657BWE ]
Desc. Breve [PROCESOS AUTOMÁTICOS NO SON EJECUTADOS ]
Desc. Detallada [ ]
Descripción Detallada del Problema oprime ! para acceder el campo ]

```

Posterior a que se han capturado los datos necesarios, se pregunta si el reporte es de hardware o software, para lo cual se asigna a un equipo de soporte diferente.

```

*** ALTA DE UN REPORTE ***
Número de Reporte [ 00000000 ] Fecha [ 1998-05-15 21:00 ]
Clave d
Clave d EL REPORTE SE ASIGNA A-> APLICACION E->EQUIPO ■
Clave Aplicación [ 2]MS-OFFICE
Clave del Equipo [ 1]HP 9000/ SERIE 800
Responsable [ 3]SILVESTRE HDEZ.
No serie Equipo [45F6578HME ]
Desc. Breve [PROCESOS AUTOMÁTICOS NO SON EJECUTADOS . ]
Desc. Detallada [Los procesos automáticos que se ejecutan en este equipo]

```

Por último se pregunta al usuario si desea en ese momento realizar la impresión y con esto concluye la alta del reporte.

Opción A.B) consulta de reportes en estas pantallas se cuenta con la facilidad de digitar los criterios que son necesarios para realizar la consulta, una vez que el manejador de Base de Datos ha extraído los datos, es posible navegar en la información, por otra parte la opción de estado nos proporciona el historial de este reporte, así la opción de texto nos permite editar en modo consulta la descripción detallada del reporte y finalmente la opción de imprimir, envía el reporte a la impresora predeterminada para el sistema operativo, la siguiente pantalla muestra las diferentes opciones:

```

CONSULTAS: [Consulta] Siguiente Previo Estado texto Imprimir ...
          *** CONSULTA DE REPORTES ***
No de Reporte [ ] Fecha de inicio [ ]
Clave del [ ]
Clave del [ ]
Clave Aplicación [ ]
Clave del Equipo [ ]
No_serie Equipo [ ]
Desc. Breve [ ]
Desc. Detallada [ ]
Desc. de Solución [ ]
Estado [ ] Fecha Término [ ]

```

La opción A.C) Escalamientos como su nombre lo indica, tiene las opciones necesarias para realizar un escalamiento del reporte, en este punto es importante destacar que es necesaria la captura de un motivo en la descripción detallada, el reporte toma el estado AA y siempre que sea accesada esta opción, automáticamente el sistema despliega los reportes que se encuentren en estado de pendientes.

```

ESCALAR: [Escalamiento] Previo Estado esCalar Terminar
Consultar el Siguiente Registro
          *** ESCALAMIENTO DE REPORTES ***
No de Reporte [ ] Fecha de inicio [1995-02-09 17:36]
Clave del Centro [ 1] CENTRO DE PROCESAMIENTO NACIONAL
Clave del Usuario [ 8] ALEJANDRO CARREON
Clave Aplicación [ 3] LAN MANAGER
Clave del Equipo [ 4] APOLLO SERIE 700
No_serie Equipo [ ]
Desc. Breve [ ]
Desc. Detallada [ ]
Desc. de Solución [ ]
Estado [ ] Fecha Término [ ]

```

Una vez que se encuentra posicionado en el reporte a escalar se accesa a esta opción tecleando la letra "C", se desplegará una lista con las personas a las cuales es posible asignar este reporte, en este punto se debe elegir uno y con la tecla ESC abandonar dicha ventana, posteriormente se desplegará un recuadro en el cual se debe capturar el motivo del escalamiento como se muestra en la siguiente ventana:

```

ESCALAR:  Siguinte  Previo  Estado  Escalar  Terminar
          ***  ESCALAMIENTO DE REPORTES  ***
No de Reporte [ 1 ]      Fecha de inicio [ 1995-02-03 17:56 ]
Clave del
Clave del Motivo :
Clave Aplicación [ 3 ] LAN MANAGER
Clave del Equipo [ 4 ] APOLLO SERIE 700
No_serie Equipo [ ]
Desc. Breve [ ]
Desc. Detallada [ ]
Desc. de Solución [ ]
Estado [ ]      Fecha Término [ ]
  
```

Opción A.D) Modificar los datos de un reporte, se refiere a la posibilidad de cambiar algunos datos del mismo que en su momento fueron capturados de forma incorrecta, como se observa en la ventana siguiente, se debe iniciar con una consulta posteriormente accesar la opción de Modificar, en este punto es posible cambiar datos del centro de cómputo, usuario, aplicación descripción breve y detallada, inhabilitando los campos de responsable, número de reporte y fecha de inicio, de tal forma que si un reporte fue asignado de forma incorrecta este debe ser cancelado y nuevamente se debe levantar uno nuevo.

| | | | | | | |
|------------------------------------|----------|-------------------------|----------------------------------|-------|------------------|-------------------|
| ACTUALIZAR: | Consulta | Siguiente | Previo | texto | Modificar | Terminar |
| *** ACTUALIZACIONES A REPORTES *** | | | | | | |
| Número de Reporte | [| |] | Fecha | [| 1995-01-30 12:56] |
| Clave del Centro | [| 1] | CENTRO DE PROCESAMIENTO NACIONAL | | | |
| Clave del Usuario | [| 2] | HECTOR GUADARRAMA | | | |
| Clave Aplicación | [| 2] | MS-OFFICE | | | |
| Clave del Equipo | [| 3] | LAPTOP IBM | | | |
| Responsable | [| 3] | SILVESTRE HDEZ. | | | |
| No serie Equipo | [| |] | | | |
| Desc. Breve | [| INSTALACION DE SOFTWARE | | |] | |
| Desc. Detallada | [| | | | |] |
| Estado | [| CA] | | | | |
| Clave para el Centro de Cómputo | | | | | | |

Opción A.E) Cierre de reportes, esta opción inicia con la consulta de los reportes que presentan un estado de pendiente, se debe navegar hasta el reporte que se desea cerrar se accesa la opción de Cerrar el cursor se posiciona en la captura de la solución, se debe digitar "!" para capturar desde el editor el texto que se propone como solución, es importante aclarar que si no se cuenta con datos en este campo no es posible que el reporte sea cerrado, lo anterior debido a que el se proporciona como restricción para el sistema que los reportes deben ser cerrados siempre que tengan una solución, ventana siguiente:

```

REPORTES:  Siguiete  Previo  Estado  cancelar  Definir  Imprimir  ...
          ***  CONSULTA DE REPORTES  ***
No de Reporte  [ 1 ]      Fecha de inicio  [ 1998-02-13 17:58 ]
Clave del Centro [ 1 ] CENTRO DE PROCESAMIENTO NACIONAL
Clave del Usuario [ 8 ] ALEJANDRO CARREON
Clave Aplicación [ 3 ] LAN MANAGER
Clave del Equipo [ 4 ] APOLLO SERIE 700
No_serie Equipo [
Desc. Breve [ ]
Desc. Detallada [ ]
Desc. de Solución [ ]
Estado [ ]      Fecha Término [1998-03-13 21:33]
Descripcion detallada de la solución al problema

```

Opción 8 Reportes y Estadísticos, en esta parte del menú es posible acceder con reportes que proporciona el sistema, en todas las opciones se presenta la siguiente ventana que indica si el resultado del reporte será por pantalla, a un archivo o a la impresora predeterminada:

```

HELP  DESK

```

```

A) HOJA DE SOPORTE
C
T A) FECHA SOLUCION >2 DIAS .
C) REP SALIDA: [ ] Impresora Pantalla Terminar
D) REP Salida a un archivo
E) REP
T) SAL

```

<< Ctrl-c >> Cancelar << ESC >> Aceptar (o letra inicial en el menú)
 << Ctrl-p >> Pantalla anterior

La siguiente ventana muestra la salida de los datos a pantalla, como se observa los datos de cada reporte son separados por "—", en todos los reportes se emplea la misma forma de despliegue lo que varía es el contenido de la información que es:

- Reportes con solución mayor a dos días
- Reportes escalados
- Reportes abiertos
- Reportes cerrados y
- Reportes cancelados.

En todos los casos los datos son los *mínimos necesarios* para identificar el volumen de los mismos, el sistema esta capacitado para realizar impresiones masivas desde la consulta opción A.B.

```

ISOSA                               Gerencia de Sistemas
-----                               Corporativos
-----                               -----
REPORTES CON MAS DE DOS
DIAS DE RESPUESTA

No Reporte  11                      Fecha Inicio 1995-02-09 17:36
                                                Fecha Cierre 1998-03-13 21:33

-----                               -----
DATOS  GENERALES
-----                               -----
CENTRO      1 CENTRO DE PROCESAMIENTO NACIONAL
USUARIO     8 ALEJANDRO CARREDON

-----                               -----
DATOS DEL RESPONSABLE
PROBLEMA DE SOFTWARE

ASIGNACION :
LIC. SERGIO GARCIA GERENCIA DE TECNOLOGIA

```

■

Opción C actualización a catálogos, en esta parte del sistema se puede proporcionar mantenimiento a los catálogos desde menú, es importante hacer notar que una vez instalado el

sistema, los catálogos se encuentran cargados con los datos mínimos para que inicie su funcionalidad, y en la medida que se demanden mas datos, el sistema tiene la capacidad de actualizar los mismos. A continuación se presenta la pantalla correspondiente:

```

      H E L P      D E S K
  
```

```

A) HOJA DE SOPORTE      .
B) REPORTES Y ESTADISTICOS .
  
```

```

T
  B) CATÁLOGO DE EQUIPOS      .
  C) CATÁLOGO DE APLICACIONES .
  D) CATÁLOGO DE USUARIOS     .
  E) CATÁLOGO DE RESPONSABLES .
  T) TERMINAR
  
```

```

<< Ctrl-c >> Cancelar      << ESC >> Aceptar ( o letra inicial en el menú)
<< Ctrl-p >> Pantalla anterior
  
```

De igual forma que en los reportes las pantallas para los cambios a los catálogos presentan las mismas opciones y el mismo método de operar por lo cual se explicará solo la pantalla que actualiza el catálogo de centros de cómputo. Al seleccionar las opciones C.A. se desplegará una pantalla con las opciones de consulta, siguiente, previa las cuales nos son útiles para navegar en la información.

Así también se despliegan las opciones "Alta" que como se indica nos es útil para agregar un registro al catálogo, la opción "Modifica" que nos facilita la actualización de datos en específico de un registro del catálogo (no es posible actualizar los datos de clave) y finalmente tenemos Bajas, esta opción solo se encuentra disponible siempre que los datos de este registro no se encuentren relacionados con datos de reportes, de lo contrario no es factible borrar dicho registro, lo anterior debido a que la información puede llegar perder la integridad.

| CENTRO: | Consulta | Siguiente | Previo | Alta | Modifica | Baja | Terminar |
|--|----------|-----------|--------|------|----------|------|----------|
| *** CATÁLOGO DE CENTROS DE CÓMPUTO *** | | | | | | | |
| Clave Centro | [| |] | | | | |
| Nombre del Centro | [| | | | | |] |
| Dirección | [| | | | |] | |
| C. P. | [| |] | | | | |
| Teléfono | [| | | | | |] |
| Fax | [| | | |] | | |
| Red Satelital | [| | |] | | | |

<< Ctrl-c >> Cancelar << ESC >> Aceptar (o letra inicial en el menú)
<< Ctrl-p >> Pantalla anterior

Los catálogos que es factible actualizar son:

- Centros de cómputo
- Equipos
- Aplicaciones
- Usuarios
- Responsables.

Bibliografía

- Informix On_Line Dynamic Server System Administration
Informix SQL Reference
Informix 4GL User Guide
Informix SQL Tutorial
Informix Inc 1997
- Oracle Developer's Guide
David McClanahan
McGraw-Hill
- Ley Aduanera y Reglamento
ISEF 1996
- Análisis Estructurado Moderno
Edward Yourdon
Prentice Hall 1989
- Introducción al Análisis de Sistemas
Enid Squire
Presentaciones y Servicios de Ingeniería de México

-
- **Structured Analysis Methods For Computer Information Systems**

Lavette C. Teague

McMillan Publishing New York

- **Aplique SQL**

James R. Groff

Mc Graw Hill 1992

- ***Oracle Guía para el Principiante***

Michael Abbey

Mc Graw Hill 1996