22

UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

CAMPUS ARAGON

INGENIERIA EN COMPUTACION

TITULO DE LA TESIS

SISTEMA DE INFORMACIÓN PERIÓDICA

Y

PROGRAMA ANUAL DE CONTROL DE AUDITORIAS.

095913

Elaborada por el C. Jorge Flores Rodríguez para obtener el grado de Ingeniero en Computación.

Asesor de Tesis:
Ing. Juan Gastaldi Pérez.

Revisores:
Ing. José González Bedolla.
Lic. Israel Juárez Ortega.
Ing. Blanca Estela Cruz Luevano.
Lic. Ma. Guadalupe Cruz Luevano.

Septiembre-2001

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN

DIRECCIÓN

JORGE FLORES RODRÍGUEZ
P R E S E N T E.

En contestación a la solicitud de fecha 28 de septiembre del año en curso, relativa a la autorización que se le debe conceder para que el señor profesor, Ing. JUAN GASTALDI PÉREZ pueda dirigirle el trabajo de tesis denominado, "SISTEMA DE INFORMACIÓN PERIÓDICA Y PROGRAMA ANUAL DE CONTROL DE AUDITORIAS" con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

A t e n t a m e n t e
"POR MI RAZA HABLARÁ EL ESPÍRITU"
San Juan de Aragón, México, 10 de noviembre del 2000
EL DIRECTOR

M en R.I. CARLOS EDUARDO LEVY VÁZQUEZ

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON UNAM
DIRECCION

C p Secretaría Académica.
C p Jefatura de la Carrera de Ingeniería en Computación.
C p Asesor de Tesis.

CELV/AIR/RCV'lla.

VNIVERIDAD NACIONAL
AVENMA DE
MEXICO

ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES ARAGÓN

JEFATURA DE CARRERA DE INGENIERÍA
EN COMPUTACIÓN

OFICIO: ENAR/JACO/0420/2001.

ASUNTO: Asignación de Jurado.

LIC. ALBERTO IBARRA ROSAS
SECRETARIO ACADÉMICO
P r e s e n t e .

Por este conducto me permito presentar a usted. nombre de los profesores que sugiero integren el Sínodo del Examen Profesional del alumno **JORGE FLORES RODRÍGUEZ**, que presenta el tema de tesis : **"SISTEMA DE INFORMACIÓN PERIÓDICA Y PROGRAMA ANUAL DE CONTROL DE AUDITORÍAS".**

PRESIDENTE:	**ING. JOSE GONZÁLEZ BEDOLLA**
VOCAL:	**ING. JUAN GASTALDI PÉREZ**
SECRETARIO:	**LIC. ISRAEL JUÁREZ ORTEGA**
SUPLENTE:	**IING. BLANCA ESTELA CRUZ LUÉVANO**
SUPLENTE:	**LIC. MA. GUADALUPE CRUZ LUÉVANO**

Quiero subrayar que el director de tesis es el **Ing. Juan Gastaldi Pérez** el cual está incluido con base en lo que reza el reglamento de Exámenes Profesionales de ésta Escuela.

Sin otro en particular, me es grato enviarle un cordial saludo.

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPÍRITU"
San Juan de Aragón, Edo. de México, junio 29del 2001.
EL JEFE DE CARRERA

M. EN C. JESÚS DÍAZ BARRIGA ARCEO

c.c.p. Lic. Ma. Teresa Luna Sánchez.-Jefa del Departamento de Servicios Escolares.
Ing. Juan Gastaldi Pérez .-Asesor de tesis.

JDA/vjd

INDICE

INTRODUCCION.

Con el objeto de llevar un control en el desarrollo de auditorias dentro de la administración pública, la Secretaria de Contraloría y Desarrollo Administrativo, la cual llamaremos de aquí en adelante como SECODAM, en su Unidad de Seguimiento y Evaluación de la Gestión Público (USEGP) y más específicamente la Dirección General Adjunta de Control y Seguimiento (DGACS), que entre sus funciones viene recibiendo la información generada por cada una de las contralorías internas de las dependencias gubernamentales, los órganos desconcentrados y las entidades paraestatales tiene a bien el desarrollo de un sistema automatizado para la recepción, y el análisis de la información generada. Para lo anterior, y a fin de contar con los elementos técnicos necesarios para el desarrollo exitoso de su objetivo, se apoyaron en la Dirección de Administración de Bases de Datos (DABD) cuyas funciones entre otras son la evaluación, planeación y desarrollo de sistemas de software manejados por la misma Unidad.

Para fin de entender más la estructura de la información manejada en la DGACS. y el flujo que lleva desde su generación, empezaremos por dar algunos conceptos generales de lo que son:

PROGRAMA ANUAL DE CONTROL DE AUDITORIAS (PACA).

Cada dependencia gubernamental, entidad paraestatal y en su caso ciertos órganos desconcentrados, tienen un organismo de evaluación y control, que se nombra su Contraloría Interna. Estas contralorías anualmente deben programar las auditorias que se van a aplicar a la institución en el ejercicio posterior. esta programación se debe poner a consideración de la SECODAM, por lo que debe de existir una vía para enviar este PACA a la USEGP.

Las auditorias programadas las podemos definir según su naturaleza.

Auditorías Públicas: Generalmente llamamos auditoría al proceso de revisión de los principales aspectos referidos fundamentalmente a la organización de las áreas, a la correcta planeación de los recursos y actividades, a la estricta observancia de las medidas de austeridad, racionalidad y disciplina presupuestal. a la existencia de adecuados controles internos y contables, a la oportunidad y confiabilidad con que son registradas las operaciones y al cumplimiento de disposiciones legales, de metas y objetivos.

Algunas características de las auditorías son:

- La descripción de los objetivos y lineamientos se encauza principalmente al reforzamiento de los controles y a la verificación de las operaciones.
- Verificar la productividad de ciertas actividades, a fin de propiciar la promoción de la eficiencia.
- Cada auditoría debe cubrir los aspectos financieros. operacionales, de cumplimiento de leyes, etc.

  Su contenido permite a los niveles directivos de las dependencias y entidades. así como a las instancias normativas y al propio Ejecutivo Federal, conocer el resultado del manejo de los programas y recursos, con objeto de que en caso de desviaciones y/o deficiencias puedan ponerse en marcha oportunamente las medidas necesarias para evitar su recurrencia.

Cada auditoría debe pertenecer a uno de los grupos existentes que las dividen según su naturaleza.

Auditorias integrales.- La auditoría integral es la evaluación del grado y forma de cumplimiento de las metas, objetivos,

registros contables y presupuestales de una dependencia, órgano desconcentrado, entidad o Procuraduría General de la República.

Auditorías de desempeño.- Se refieren a la verificación de la estructura organizacional y de los sistemas de planeación, programación, operación e información, a fin de corroborar que las diferentes etapas del proceso administrativo se estén llevando a cabo con eficiencia, eficacia, economía y transparencia, el cumplimiento de las metas previstas para el ejercicio presupuestal correspondiente, así como los indicadores de gestión y los de desempeño y su impacto socioeconómico considerando la temporalidad de los mismos, la capacidad operativa de las áreas administrativas y evaluando los resultados de éstas en función de sus actividades y objetivo, considerando las condiciones y circunstancias que prevalecieron, y las acciones implementadas en materia de modernización y desarrollo administrativo, proponiendo las recomendaciones pertinentes que permitan subsanar las deficiencias detectadas.

Auditorías específicas.- Estas auditorías pueden ser administrativas, operativas, financieras, de legalidad o sustantivas, entre otras, y deben orientarse a reforzar los objetivos de las auditorías integrales a través de revisiones con alcances, enfoques y objetivos particulares bien definidos.

Auditorías de evaluación de programas.- Se orientan a la revisión de los programas prioritarios y/o estratégicos que el Gobierno Federal asigna a una dependencia, órgano desconcentrado, entidad o Procuraduría General de la República, así como a los especiales, por ejemplo: Procampo, Créditos del Banco Mundial, Programa de Modernización de la Administración Pública 1995-2000 o Programas Sectoriales, todos ellos por realizarse a través de las áreas sustantivas de la institución.

Auditorías de seguimiento.-Se refieren a la verificación que el órgano interno de control debe efectuar para asegurar que las áreas ya auditadas estén atendiendo, en los términos y plazos acordados y establecidos, las recomendaciones preventivas y correctivas para abatir la problemática detectada así como aquéllas referentes a la modernización administrativa planteadas por cualquier instancia fiscalizadora.

SISTEMA DE INFORMACIÓN PERIÓDICA (SIP).

Una vez programado un PACA, cada año es necesario informar sobre los asuntos que se observan en la elaboración del mismo, estos seguimientos se deben informar periódicamente a la USEGP a través del sistema denominado Sistema de Información Periódica (SIP). El SIP por lo tanto, además de trabajar el mismo concepto de revisión, integra una serie de procesos que envuelven nuevos conceptos:

Observaciones: Después de haber concluido cada una de las revisiones, se deberán identificar las principales deficiencias, desviaciones y los aspectos susceptibles de mejorar encontrados en las intervenciones, caracterizándose las mismas por su relevancia, claridad, concisión y objetividad, es decir las observaciones que surjan de la auditoría, esta información debe ser descriptiva y tener a su vez un texto de recomendaciones hechas por el mismo órgano interno de control.

Seguimientos: Siendo el propósito fundamental de la auditoría pública el coadyuvar a la mejora en general de la

administración de las dependencias y entidades, la verificación de que las acciones correctivas y de mejora se lleven a la práctica constituye una de las principales responsabilidades de la función.
Con motivo de estas acciones, el titular del órgano interno de control deberá comunicar, a través de este medio, el estado que guarda la implantación de las medidas correctivas propuestas como resultado de las observaciones detectadas en las auditorías, así como si dichas medidas están siendo atendidas oportuna y adecuadamente.

Costo y Recuperaciones: es el resumen de los costos y ahorros que se manifiestan en la elaboración de las auditoria, así como en el seguimiento de las acciones implantadas.

La anterior información se maneja y se envía a la SECODAM desde las contralorías internas de cada entidad, dependencia u órgano desconcentrado; con tal efecto que en la USEGP se recibe periódicamente esta información.
Como antecedentes técnicos del sistema automatizado requerido, podemos mencionar que en 1981 se contaba con una serie de programas hechos en lenguaje cobol sobre un equipo onyx con unix 3.0 cuyo objetivo principal era la generación de un reporte con las auditorias realizadas anualmente, y prácticamente se trataba de la recaptura de toda la información recibida en diferentes formatos y paquetes para su estandarización. cabe mencionar que en ese entonces no se pedía a las contralorías la cantidad de información de ahora se les pide. En 1990 las SECODAM, que en ese entonces se denominaba SECOGEF, aumenta sus requerimientos de información a la contralorías, lo que ocasionó que el sistema comenzara a crecer desmesuradamente y sin planeación; el área de trabajo de sistemas llego a tener hasta 2,000 archivos entre procesos y archivos de datos, esto ocasionó que el uso de los procesos realizados se hiciera lento e incomodo y por si fuera poco dependiente necesariamente del encargado que era el único que conocía del flujo de la información y de la mayoría de los proceso, teniendo que supervisarlos personalmente.

El sistema funcionaba a través de archivos planos que se clasificaban en aquellos que contenían la información correspondiente a las revisiones, estos archivos se dividían a su vez por el año de captura de las auditorías; existían también los archivos de observaciones que se clasificaban por el estatus de conclusión que guardaba cada observación y por último teníamos los archivos de seguimiento clasificados por ejercicio. A fin de año se realizaba un respaldo de la información para poder realizar una depuración de archivos, donde se seguían los siguientes procesos:

- **Archivos de revisiones**. estos archivos eran separados por año, cabe mencionar que cuando se comenzó el análisis del sistema actual se contaba con cuatro archivos el primero de ellos contenía la información de los periodos de 1981 a 1993 y los restantes a los ejercicios 1994, 1995 y 1996.

- **Archivo de observaciones**. Dentro del depuramiento anual de este archivo se eliminaban las observaciones concluidas en el año, dejándolas en un nuevo archivo y generando otro para las pendientes, que seria el archivo en uso. Como resultado al inicio del análisis se contaba con 14 archivos separados de igual manera que en el caso de los archivos de revisión, solo que duplicados en archivos con observaciones concluidas y archivos con observaciones pendientes.

- **Seguimientos**. Estos archivos seguían las mismas reglas que los archivos de observaciones, y de igual manera como producto se obtuvo 12 archivos de seguimientos clasificados como antes se mencionó.

El mismo procedimiento de copia de archivos se utilizaba para los programas, procesos y archivos auxiliares, es decir que se renombra toda el área de trabajo cambiando solo los dígitos del año en los nombres. El cambio de año ocasionaba la perdida de programas de uso no frecuentes y complicaba el funcionamiento del sistema al querer trabajar con archivos de años anteriores.

En 1994 la persona encargada deja de trabajar en esta dependencia, por lo que el sistema se queda sin supervisión y sin documentación real, es decir prácticamente inútil.

En el mismo año ingresa un nuevo encargado de la información con la misión de desarrollar y planear el rediseño del Sistema de Información Periódica (SIP), este debería contemplar que la información que se recibiera ya no fuera recapturada, y además su vía de transmisión sería por medios magnéticos (diskettes).

El proyecto se desarrolló, se liberó y distribuyó a las dependencias y entidades hasta 1995, no dando los resultados deseados ya que la información al llegar a SECODAM no se podía exportar al sistema central residente, esto se debido a que se pedía en hoja de calculo (lotus o excel), provocando inflexibilidad en el manejo del texto y del resto de la información.

Cabe mencionar que ni en el anterior sistema y ni en este rediseño se tomó en cuenta a la Dirección de Administración de Base de Datos.

En 1996, la persona responsable nuevamente deja de trabajar en esta dependencia, no dejando al igual que la ocasión anterior ningún tipo de documentación de su sistema , tampoco capacita a una persona para la operación del mismo, dejando nuevamente inútil todos los archivos y programas de proceso.

A partir de este momento se le hace el requerimiento del sistema actual a la Dirección de Administración de Bases de Datos (DABD), y hasta la fecha es esta dirección la responsable del manejo y soporte del Sistema de Información Periódica y el Programa Anual de Control y Auditorias.

El presente documento es una síntesis del trabajo realizado en la elaboración del sistema descrito en los párrafos anteriores. Se comenzará dentro del capítulo uno a describir la metodología empleada por la DABD para el diseño del mismo.. En los capítulos dos y tres se hace un esbozo de dicha metodología aplicada a los dos módulos del sistema. El trabajo de investigación puramente teórico de las herramientas utilizadas en el desarrollo del sistema se presenta en los capítulos cuatro y cinco, el primero encargado de la teoría relativa a una de las características mas importantes del sistema en cuestión que es la seguridad de la información y el segundo a las formas de transmisión de los datos, estos capítulos suenan interesantes ya que representan junto con el capítulo número uno la base teórica que nos permitió el desarrollo del presente sistema. Por último, el capítulo cinco nos permite darnos una idea de cómo evolucionaría el sistema hacia una de las técnicas mas comunes en la actualidad la estructura cliente / servidor, toda vez que se vayan dando las circunstancias propicias dentro de la misma dependencia.

CAPITULO I. METODOLOGÍA DEL DISEÑO DEL SISTEMA

La metodología empleada en el desarrollo del presente sistema esta clasificada dentro de las metodologías orientadas a objetos que constan de actividades administrativas como:

Definición de conceptos y diagramas.
Etapas y definición de entregas en cada una de ella.
Definición de actividades y recomendaciones.

Dentro de las metodologías orientadas a objetos podemos mencionar algunas de las más conocidas:
The Object Oriented Design , de Grady Booch, cuyas fases se resumen en : análisis de requerimientos, análisis de dominio y diseño.
The Objectory, de Ivar Jacobson con las fases de: análisis de requerimientos, análisis de robustez, diseño, implementación y pruebas.
The Object Modeling Technique, de James Rumbagh que consta de: análisis, diseño del sistema, diseño de objetos e implementación.

Las antes mencionadas metodologías, tiene similitudes entre si, y realmente se pueden fusionar en una sola cuyo uso se ha hecho común e incluso política dentro de la Dirección de Administración de Base de Datos (DABD) The Objetory Process.

Toda metodología Orientada a Objetos normalmente se apoya en el UML (Unified Modelling Language) que se podría traducir como lenguaje Unificado de Modelado, que es un conjunto de diagramas y herramientas lógicas para modelar cualquier sistema, con una sintaxis comprensiva e independiente del lenguaje de codificación o programación que se pretenda ocupar. Los diagramas que utilizaremos en nuestra metodología básicamente son: diagramas de caso de uso, de clases, de estado, de secuencia, de componentes y de actividades que más adelante definiremos dentro de cada fase en la construcción del sistema.

Las fases para la construcción de un sistema dentro de una metodología orientada a objetos siguen siendo las mismas de antaño :

Análisis : dentro de todas las técnicas estudiadas para el desarrollo de sistemas, no existe alguna que no tenga esta etapa inicial de análisis, en ella se conceptualiza los objetivos del sistema, sus alcances, las políticas que pudiera tener la elaboración, o que se deben de considerar para el producto final.

Elaboración: aquí la actividad principal es el diseño de la arquitectura del nuevo sistema, para tal diseño es indispensable establecer el dominio en que trabajará, desarrollar un buen plan de trabajo eliminando cualquier riesgo perceptible, para esta fase se utilizan varias técnicas dentro del Objetory Process que explicaremos más adelante como son el caso de uso y el

modelo de clases. Dentro de esta fase se identifican también todos los factores y actores que jugarán un rol dentro del sistema y obviamente todos los requerimientos del sistema.
Realmente en la vida práctica no es muy palpable la división entre las fases antes descritas y desde mi punto de vista esta división solo es necesaria para el estudio del mismo diseño de sistemas.

Construcción: el producto o meta principal de esta fase es el software completo, es decir listo para ponerse a prueba para la transición a los usuarios finales y con ésto también la entrega de todo tipo de manuales de usuario y el técnicos.

Transición: una vez que no existen más requerimientos a cubrir de las peticiones iniciales, el producto está listo para su liberación y distribución, esta fase podría en dado caso complementarse con el soporte (mantenimiento) que se pudiera vender al usuario.

The Objetory Process.

Realmente hasta ahora no se ha descrito más que las fases del diseño y construcción de un sistema en una metodología orientada a objetos; lo que explicaremos a continuación son los componentes del mismo Objetory Process..
En realidad este proceso puede definirse como una buena técnica para el desarrollo y construcción de sistemas de software realmente de calidad, describiendo la asignación de tareas y actividades en el desarrollo del mismo dentro de un plan de trabajo ya definido.
Llamamos componentes al agrupamiento o clasificación lógica de actividades bien definidas, y nuestra metodología consta de cuatro componentes: análisis de requerimientos, diseño, implementación y pruebas. Si viéramos en forma de una matriz bidimensional la metodología descrita, por un lado tendríamos las fases de desarrollo de un sistema y por otro los componentes de nuestro proceso, como lo indica el siguiente esquema (figura 1.1).

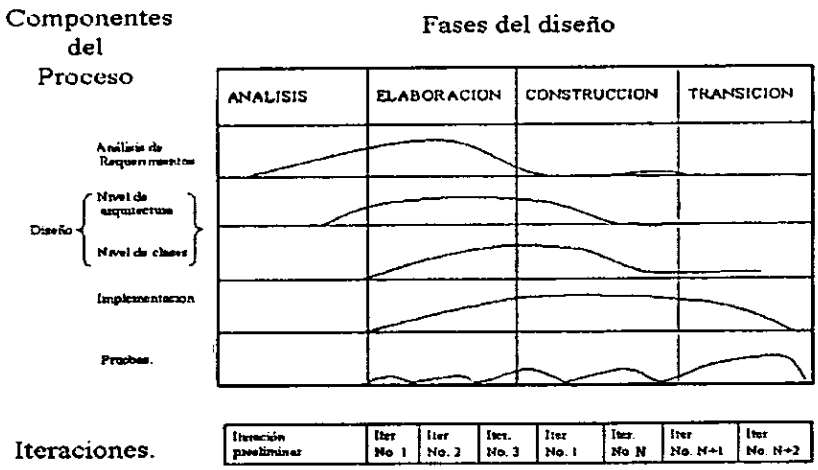


Figura 1.1.

El proceso como se puede apreciar es una repetición de la aplicación de sus cuatro componentes a través de cada fase de desarrollo, obviamente la forma o énfasis que cada componente toma en las diferentes etapas es la diferencia entre las iteracciones .
Explicado de diferente forma, nuestro proceso iteractivo debe cubrir los cuatro componentes dentro de cada repetición y al final de esta existen una serie de elementos y documentos que deben entregarse. Esto lleva a detallar las actividades que se deben realizar en cada componente.

Análisis de Requerimientos.

Lo que se busca dentro de estas actividades es lograr el entendimiento de lo que el usuario o cliente desea y la forma más eficaz de presentarle la solución a esto. Los elementos que intervienen son:
Los actores: que pueden ser tanto usuarios directos como indirectos que van hacer uso del producto.
Los casos: que son las operaciones o procedimientos que el sistema debe realizar.
Las relaciones: definir de que forma se relaciona cada actor con cada caso.
Modelo del mundo: esto resulta de la identificación de los tres elementos anteriormente mencionados.

Las actividades técnicas que se realizan en el análisis de requerimientos involucran a cada uno de los cuatro elementos antes mencionados, y dan como resultados documentos que se deben entregar al termino de este componente, como son el o los diagramas de caso de uso, la descripción de cada diagrama, descripciones textuales de la o las posibles soluciones que se le daría a cada caso de uso, el modelo del mundo inicial y por último los diagramas de secuencia.

A continuación resumiremos las actividades que se deben de realizar en esta etapa.

Identificación de actores y casos.- Es decir clasificar los usuarios primarios o secundarios del nuevo sistema, y los papeles que van a desenvolver dentro de este, quienes se pudieran beneficiar y en dado caso a quien pudiera perjudicar, los actores primarios son los que trabajarán directamente con el sistema, los secundarios son los que reciben información, supervisan procesos o dan mantenimiento al mismo; además se deben identificar otros posibles sistemas que van a relacionarse con el nuestro.

Identificar los casos de uso.- Descripción de los procedimientos u operaciones que realizará el sistema, clarificar cada una de las tareas que desempeñan los actores, de que forma van a consultar o modificar la información, que información manejará cada actor y de que forma la incorporará al sistema, que otra información requiere el sistema del exterior.

Definir las relaciones que existen entre: actores y casos, actores y actores, y entre caso y caso; aquí debemos comparar los casos de uso similares estableciendo las operaciones que los hacen diferentes, o si algún caso es usado como transacción de otro.

En la definición de las relaciones es necesario la descripción textual del objetivo y las operaciones de cada caso, explicando además las posibles variantes para cada uno, y describiendo con absoluta precisión la fórmula de su solución. Esta actividad

puede reforzarse con el comienzo de la interfaz de usuario, aunque no es necesario, detallando la descripción de su funcionamiento.

Construir un modelo inicial del mundo.- Primero es necesario definir el concepto de clase como un elemento físico o lógico con propiedades y procedimientos bien definidos.

En la construcción de un diagrama de clases o modelo conceptual del mundo, es necesario tomar la técnica Top-Down, es decir empezar por definir la clase más general, definiendo sus componentes hasta llegar a clases básicas o elementales. Como inicio en el desarrollo de un diagrama de clases podemos identificar los sustantivos del objetivo u objetivos del sistema y determinar si se podrían considerar clases dentro de nuestro diagrama.

Hemos de hacer una clasificación de clases HACE y clases ES, las primeras distinguidas por ser estados en que se encuentra el sistema, que eventos determinan estos estados, y como van evolucionando a través de su periodo de vida; las segundas se definen desde el punto de vista de la información, es decir son elementos objetivos como dispositivos, unidades organizacionales y cualquier otro objeto que interviene en el sistema, lo que hay que definir en estas clases son los eventos y procesos que debe guardar y o manipular, el papel que juega dentro del universo del sistema es decir sus funciones, sus mensajes y sus atributos.

Si hemos de organizar las actividades para la elaboración de un modelo inicial del mundo o diagrama de clases, como primer punto tendremos la identificación de actores, casos y clases, seguiríamos con la identificación de sus atributos y asociaciones, dentro de este punto determinar las características que determinan al objeto en el dominio del sistema, como se relacionan los objetos, las funciones y la información que necesita para funcionar. Con esto último podemos decir que hay que tener claro las entradas y las salidas (mensajes) de cada clase u objeto, dentro de los mensajes clasificarlos como aquellos que se requieren para manipular la información que contiene, aquellos requeridos para manejar las relaciones que se tiene con otros objetos y los mensajes que hacen que el objeto cambie de un estado a otro. Dentro de la definición de atributos debemos pensar en los valores que pueden tener y aquellos que no son posibles.

Otro de los factores que influyen en la clasificación de clases es la consideración de la herencia , es decir no ponemos estar definiendo clases que tienen como base los mismos atributos y asociaciones que otra, lejos de esto lo que debemos definir son clases padre y clases hijo, donde las clases hijo se basan esencialmente en las características del padre con algunos procedimientos, eventos o asociaciones diferentes o propias de ésta.

La agrupación lógica de clases trae el concepto de paquete que no es otra cosa que un subconjunto de clases con características similares o relacionadas funcionalmente.

Después de haber creado nuestro modelado del mundo, nos disponemos a validarlo, esta actividad es realmente un repaso de nuestro modelo, mismo que se pone a prueba con casos reales e incluso casos que no son validos en el mundo real para poder prever el comportamiento de nuestro sistema, también podemos ver con claridad que tan necesarios son los mensajes de cada clase y si son o no indispensables, o si cada clase puede obtener la información necesaria para su funcionamiento.

Dentro de la validación se pueden incluso involucrar a un usuario representativo de cada actor, para tomar en cuenta sus opiniones sobre todo de la interfaz de usuario.

Con todo lo anterior descrito, se puede decir que el análisis de requerimientos trae como productos finales los diagramas de caso de uso, las descripciones de cada caso, los diagramas de estado, el diagrama de clase (modelado del mundo inicial) y los diagramas de secuencia, estos últimos salen de la validación del modelado.

Diagramas de Caso de Uso.- Estos diagramas muestran la relación entre los usuarios y los casos (proceso o actividad) dentro del sistema desarrollado, o también entre caso y caso, este tipo de diagramas son muy útiles en la validación del modelo creado.

Dentro del diagrama un caso de uso se representa por medio de un ovalo con una descripción del proceso adjunto a él; un actor se representa por la figura de un hombre y una relación es una línea o flecha que une a un actor y un caso o a dos casos, este tipo relación se clasifican como sigue:
La línea entre un actor y un caso da a entender que el usuario de alguna manera participa en el proceso (caso) como se ve el la figura 1.2.

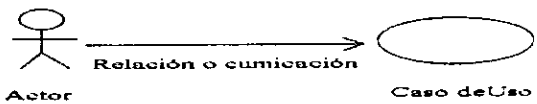


Figura 1.2.

La relación entre caso y caso pude ser de cuatro formas diferentes: una es de comunicación es decir los casos intercambian información como en el diagrama anterior, excepto que el diagrama representa la relación entre un caso y un actor.
La segunda en forma inclusiva, es decir se interpreta como que un caso es parte o utiliza al otro, esta relación se define como de uso (Figura 1.3).

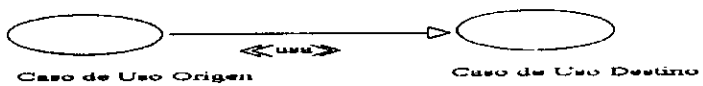


Figura 1.3.

Una tercera forma normalmente se refiere a una condicionante, es decir una alternativa de uno a otro caso de acuerdo a ciertas situaciones, además de que el comportamiento del primer caso delimita al comportamiento del segundo (Figura 1.4).

Figura 1.4.

La última relación es de herencia, es decir que el comportamiento del primer caso lo hereda el segundo caso. figura 1.5.

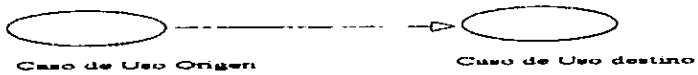


Figura 1.5.

El siguiente esquema representa un diagrama de casos de uso típico.

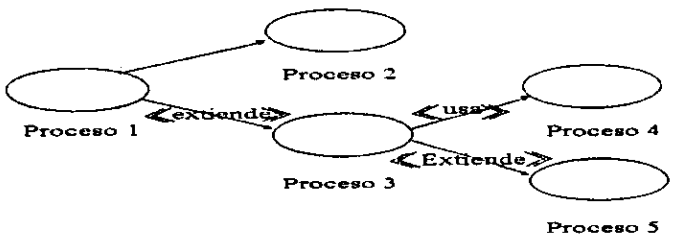


Figura 1.6.

En el segundo capítulo de esta tesis mostraremos prácticamente el uso de este y de los demás diagramas.

Diagrama de clases.- es una manera esquemática de mostrar todo el conjunto de clases que conforman el sistema, junto con las relaciones existentes entre ellas, cada clase se representa por un cuadro que contiene una lista de los atributos que la constituyen y más abajo la lista de sus eventos o métodos, encabezando esta lista por el nombre de la clase.

A continuación se muestra un típico diagrama de clases (figura 1.7):

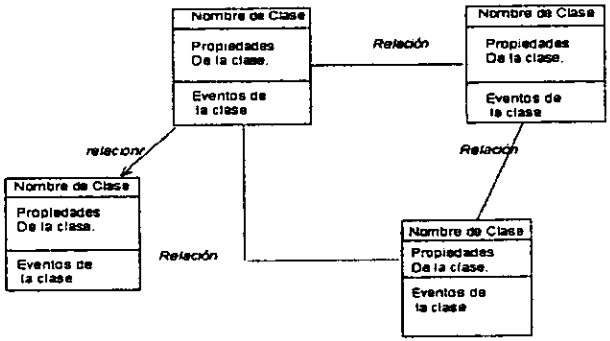


Figura 1.7.

En realidad el modelado del sistema se compone de varios diagramas de clases, clasificados lógicamente dependiendo los módulos del sistema.

Dentro de la lista de atributos, cada uno de ellos tiene la siguiente sintaxis:

[dominio] [nombre] : [tipo] = [valor inicial] { [propiedades]}

donde:

Dominio.- puede tomar los siguientes simbolos:

+ publico.- el atributo es visible incluso fuera de la clase misma.

protegido.- el atributo solo es visto por la clase y por sus descendientes, cualquier otra clase no tiene acceso a ellos.

- privado.- el atributo solo lo pueden acceder sobre la misma clase, por ninguna otra clase.

Nombre.- es el nombre o identificador de la clase.

Tipo.- es un identificador predeterminado, que agrupa una serie de propiedades esenciales para el atributo.

Valor Inicial.- dependiendo el tipo, este valor es el que toma el atributo desde la creación de una instancia de la clase.

Propiedades.- posibles cualidades no tan especificas del tipo de atributo.

Los eventos o métodos de la clase definen su funcionalidad y su forma de comportarse ante cualquier situación, y su sintaxis es la siguiente:

[dominio][nombre] ([lista de parámetros]) : [tipo de expresión que devuelve] {[propiedades]}

Donde lo explicado anteriormente para los atributos se aplica para los métodos. faltando agregar solamente que la lista de parámetros se denota de igual forma que un atributo.

Relación.- representada por una línea que une a dos o más clases, dependiendo las características de esta línea podemos saber de que tipo de relación se trata, pudiendo existir las siguientes:

Asociación Binaria.-la línea se denota sólida entre dos clases, la relación existente aquí no es de dependencia existencial para ninguna de las clases que une, es decir que si bien están relacionadas no implica que una deba existir para que la otra de igual forma exista.

Asociación N-aria.- se describiría como una asociación binaria, excepto que las clases involucradas suman más de dos, para unir varias clases se emplea un diamante del cual salen varias líneas hacia las clases unidas.

Composición.- se caracteriza por un rombo relleno del lado de la clase que contiene a la otra, esta relación representa una unión fuerte entre una y otra clase, al grado de que si no existiera una no existiera la otra. y dependiendo la cardinalidad de esta relación, incluso al tiempo que se crea o se destruye una la otra también lo hace (figura 1.8).

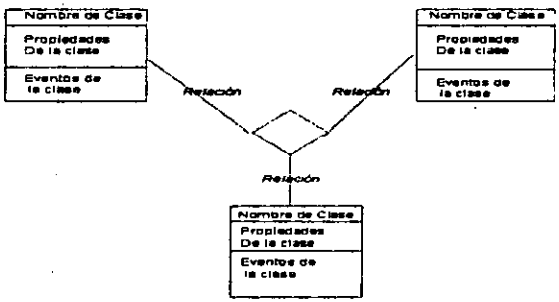


Figura 1.8.

Generalización.- esto hace notar una de las características más importantes del concepto de clases, que es la herencia, representado con un triangulo sin relleno del lado de la clase que heredará sus características a la otra.

Dependencia.- la línea es punteada y se dirige de la clase dependiente a la clase de la que depende, esta dependencia indica que si se llegará cambiar algún tipo de elemento (atributo o método) de una clase, esto traería cambios en la otra.
Paquete.- en un sistema altamente complejo, es impráctico un solo diagrama de clases, lo más propio es realizar diagramas que envuelven grupos lógicos de clases, uniéndose a través de relaciones de dependencia y generalización. Por cada grupo se deberá construir su propio diagrama de clases.

Pueden existir además de clases y relaciones, notas dentro del diagrama para aclarar cualquier circunstancia en el mismo. Estas notas se representan con líneas punteadas que involucran a uno o varios elementos (clases y/o relaciones), con el testo entre corchetes, y del lado superior derecho del rectángulo punteado.

Estereotipo.-cualquier elemento dentro de nuestro diagrama puede clasificarse dentro de ciertas características o criterios que los hacen agruparse lógicamente, esta clasificación se representa sobre el nombre de la clases o relación y entre signos de "menor y mayor que".

Interfaz.-se representa por una línea terminada en un circulo, y un nombre representativo adjunto, esto nos sugiere un protocolo o reglas que la clase debe seguir obligatoriamente.

Asociación "Or" Exclusiva.- existen casos en donde una clase debe relacionarse ya sea con una u otra diferente, esto en un diagrama de clases se representa uniendo las relaciones entre la clase y las otras dos con una línea punteada, y entre llaves la condición que hace la diferencia (figura 1.9).

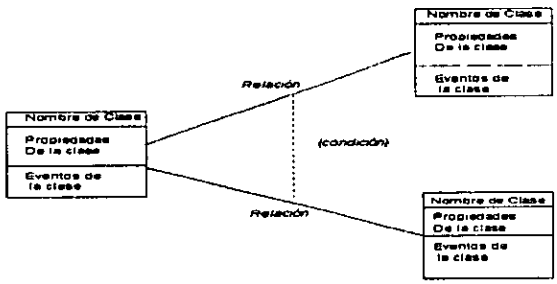


Figura 1.9.

Clase Asociación.- esta clase tiene su naturaleza dentro de una relación de otras dos clases existentes, es decir surge de detallar la relación entre dos clases, su representación es la normal de una clase unida a una relación con una línea punteada (figura 1.10.).

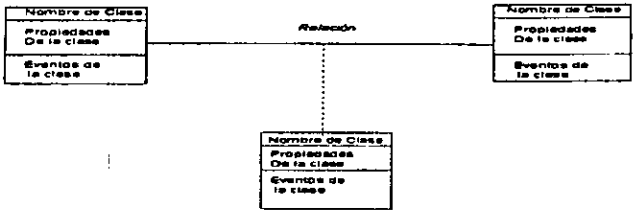


Figura 1.10.

Otra característica importante de las clases es la multiplicidad entre clases que establece restricciones de existencia para los objetos de las clases asociadas. La especificación de multiplicidad se denota utilizando los siguientes símbolos al lado de la relación entre clases:

1	uno y solo uno.
0 .. 1	cero o uno.
M .. N	de M a N (enteros).
*	de cero a varios.
0 .. *	de cero a varios.
1 .. *	de uno a varios.

Diagramas de Secuencia.- es una representación gráfica de la vida de uno o varios objetos dentro de un módulo determinado. Su estructura es la siguiente:

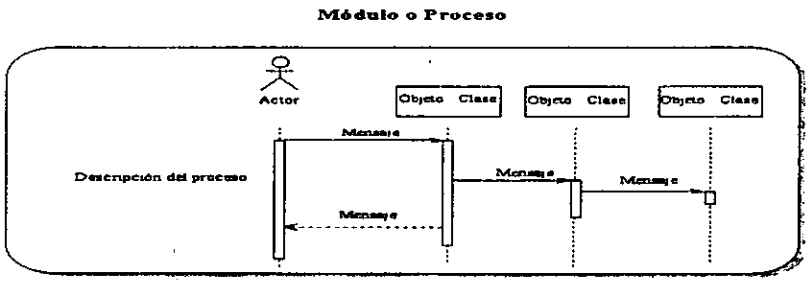


Figura 1.11.

El diagrama lo encabeza el titulo del proceso o módulo representado, le sigue una lista horizontal de los objetos o actores que intervienen, los objetos se representan en forma de rectángulos dentro de los cuales se encuentra el nombre del mismo y su clase; cada uno de esos objetos cuenta con una línea punteada vertical que representa su tiempo de vida. A través de esa

línea se pueden representar los métodos utilizados por el objeto con un rectángulo. De una cierta distancia del método (rectángulo), puede salir un mensaje dirigido a otro objeto, este mensaje se representa por una línea sólida, y un enunciado de descripción.

Además de los elementos básicos del diagrama de secuencia mostrados en la anterior gráfica, existen otros que pueden o no aparecer dependiendo el grado de complicidad de nuestro módulo.

Los tiempos de transición se presentan cuando el proceso analizado tiene tiempos críticos de respuesta, y se representarían con variables dispuestas a lo largo del rectángulo representativo de nuestro método.

Existen además procesos en donde hay alternativas en su ejecución, representadas por líneas alternas a nuestra línea de vida principal.

Cuando un método tiene como fin la destrucción de su propio objeto, al final del rectángulo correspondiente se agregará una cruz indicando el termino del objeto mismo.

Los métodos recursivos se identifican con líneas saliendo del método y entrando nuevamente a él, normalmente dentro de un sub-método que se indica con un pequeño rectángulo incrustado al costado del rectángulo principal.

La figura 1.12 representa cada uno de los conceptos mencionados en el párrafo de arriba.

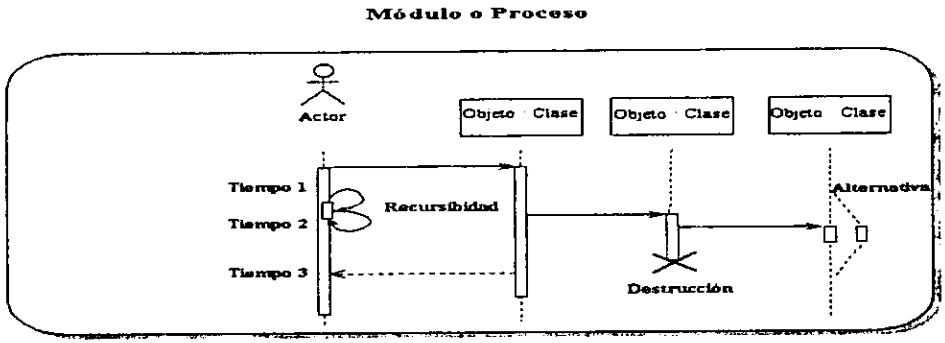


Figura 1.12

Diseño

El diseño produce los elementos que establecen cómo el sistema cumplirá los requerimientos identificados durante el análisis de los mismos. El primer paso en el diseño es identificar los informes y las salidas que el sistema producirá, obviamente preestableciendo la estructura de nuestro sistema, posteriormente se señalarán los datos específicos de cada uno de los componentes del sistema, incluyendo su localización exacta sobre el papel, utilizando técnicas de (UML) como el diagrama de componentes, el diagrama de ejecución y el diagrama de estados. El diseño también describe los datos calculados o almacenados que se introducirán. Los datos y los procedimientos de cálculo se describen con detalle. Se seleccionan las estructuras de los archivos y los dispositivos de almacenamiento , como son discos o cintas magnéticas o papel. Los procedimientos deben de mostrar cómo se van a procesar los datos y cuales van a ser la salidas, todo lo anterior

implica el detalle de la implementación del modelo del mundo hecho en el análisis de requerimientos. Se desarrollan los modelos de control y de comunicaciones entre nuestro sistema, sus actores y posibles sistemas ya existentes que interactúen con el nuestro. En esta etapa una vez concluidas las características generales de nuestro sistema se practica una evaluación tanto técnica como económica de la propuesta de solución, ya que muchas veces el sistema propuesto no va de acuerdo con el presupuesto de la empresa o incluso de su estructura funcional, o por otro lado no posee la tecnología apropiada para el nuevo sistema.

El último paso del diseño es pasar la información al grupo de programación para el desarrollo de un prototipo del nuevo sistema.

A continuación se describirán los diagramas ocupados en esta etapa de diseño.

Diagrama de Componentes. Muestra las dependencias lógicas entre componentes definidos de software del tipo que sean, ya sea binarios, fuentes o ejecutables, en este tipo de diagramas no se consideran instancias del software sino solo tipos y resaltan el uso que se les da ya sea en tiempo de ejecución de compilación o enlace.

Un componente es un grupo de clases que mantienen una estrecha relación funcional. Como ejemplo pondremos el siguiente:

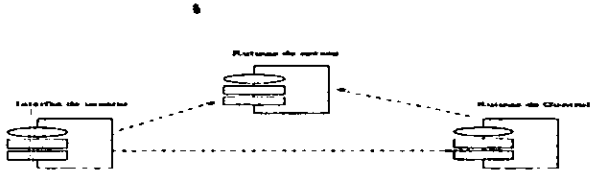


Figura 1.13.

Como se ve en el diagrama, un componente se representa como un pequeño diagramita de flujo básico encerrado sobre un cuadro, sobre del cual se encuentra el nombre descriptivo de su funcionalidad. La relación existente entre ellos se muestra a través de líneas punteadas que los unen.

Diagrama de Ejecución. También conocido como diagrama de distribución, ya que representa la distribución de los componentes de software descritos en su respectivo diagrama, en conjunción con los procesos y objetos con quienes tienen relación en tiempo de ejecución. Este diagrama se representa con una serie de nodos conectados por asociaciones de comunicación, cada nodo representado por un cubo dentro del cual se observan componentes, instancias de objetos y procesos, además de contener el nombre propio del nodo, incluso un nodo puede ser a su vez la instancia de una clase es decir un caso particular de objeto, un típico diagrama de ejecución se presenta en la figura 1.14.

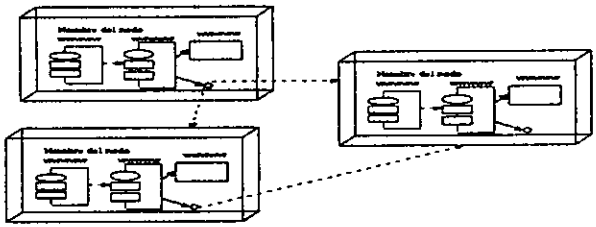


Figura 1.14.

Un nodo también se define como un objeto físico en tiempo de ejecución que representa un recurso computacional con memoria y capacidad de procesamiento.

Diagrama de Estados. Este diagrama representa el comportamiento a través del tiempo de cada objeto esencial del sistema. Un estado representa un periodo de tiempo donde el objeto desarrolla cierta actividad característica; en el diagrama un estado se muestra como un rectángulo con los bordes redondeados, en este se observan tres divisiones: en la primera se encuentra el nombre , en la siguiente el valor característico del estado y en la última las actividades que el objeto desarrolla en su permanencia en ese estado. Los estados de inicio y final se representan el primero por un circulo relleno totalmente y el segundo con un circulo relleno circunscrito a otra circunferencia. Además el diagrama se puede acompañar de la representación del objeto como en su diagrama de clases (figura 1.15)

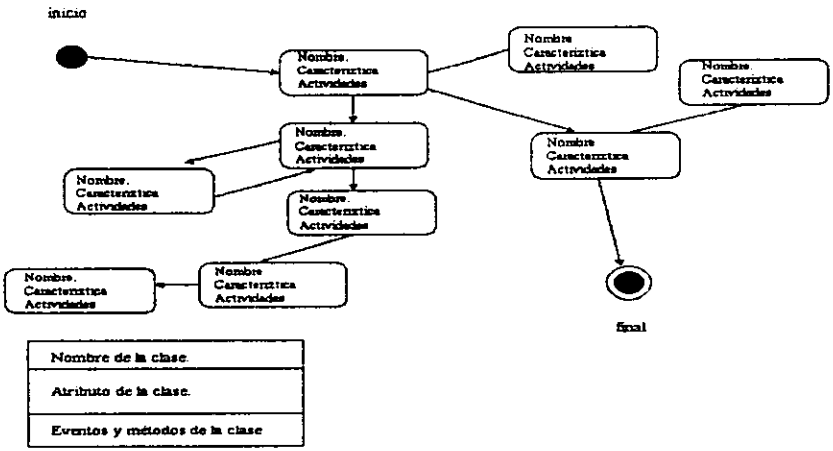


Figura 1.15.

En un diagrama complejo es posible reunir una serie se estados relacionados en una sola actividad como un solo estado con el fin de no crear un diagrama demasiado extenso y así parcializarlo en varios diagramas. Un subestado es un elemento de la descomposición de un estado generalizado, con conexiones a nivel inferior que el diagrama principal.

Se llama transición al paso de uno a otro estado, representada por una línea de conexión entre esos estados; así como existen transiciones simples que solo involucran a dos estados. existen transiciones complejas que relacionan a más de dos estados, y generalmente dependen de una condición o pueden simplemente dividir el proceso en dos líneas paralelas de estados; en el grafo se representa mediante una línea vertical de la cual salen dos mas líneas dirigidas a un respectivo estado, como lo muestra la siguiente figura.

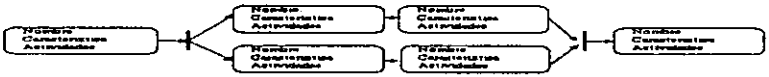


Figura 1.16.

Prototipo.- Un sistema prototipo es un sistema piloto o prueba. En algunas ocasiones se seleccionan como prototipo situaciones únicas, situaciones de alto costo y alto riesgo. Se espera que el prototipo se modifique después de varios intentos; el diseño evolucionará conforme se vaya obteniendo mayor información adicional del diseño a través de su uso. El prototipo es un sistema de trabajo diseñado para modificarse con facilidad. La información obtenida con su uso se aplicará a un diseño modificado, que a su vez, puede utilizarse como sistema prototipo de nuevo para obtener más información. El proceso puede repetirse tantas veces sea necesario para así obtener los requerimientos esenciales del diseño.

Implantación y Pruebas.

En esta etapa el sistema es utilizado en forma experimental para asegurar que el software no falle, es decir, que trabaje de acuerdo a las especificaciones y de la manera en la que los usuarios esperan que lo haga. En la prueba del sistema se examinan los datos de entrada de procesamiento y los resultados para localizar algunos problemas inesperados. Es preferible detectar cualquier falla o anomalía antes de que la empresa ponga en marcha el nuevo sistema. La prueba debe ser realizada por personas diferentes a aquellas que desarrollaron el sistema (programadores), ya que de esta manera se asegura una mayor y más completa prueba, ya que es imparcial , lo que origina un software más confiable y de más calidad.

Después de implementar las pruebas correspondientes y ajustar el software de acuerdo a sus resultados, el personal de sistemas verifica y pone en uso el equipo nuevo, se instala la nueva aplicación, se entrena al personal que manejará el sistema y construyen los archivos de datos que se necesiten. Cuando estas actividades terminan, entonces se dice que el sistema está puesto en marcha. Los desarrolladores del sistema pueden escoger una parte (un área o departamento) de la empresa para probar el nuevo sistema con sólo una o dos personas; a su vez este puede estar trabajando en forma paralela con sistema anterior para comparar resultados (beneficios). Una vez instalado, la aplicación se utilizará por muchos años,

sin embargo, las empresas, el personal y el medio ambiente cambiarán a través del tiempo. Por lo tanto, la aplicación necesitará mantenimiento; es decir, se harán cambios y modificaciones al software, a los archivos y procedimientos para así cubrir los nuevos requerimientos de la empresa. La puesta en marcha es un proceso continuo.

Código Fuente. La entrega de la codificación de los programas es producto de la etapa de implementación. El código debe cumplir con los estandares preestablecidos, además de programas de soporte como son los procesos de instalación, la creación de los directorios o el ambiente idóneo para el buen funcionamiento del sistema entre otros.

Pruebas Unitarias. Al inicio de la etapa de implantación se deben establecer las pruebas unitarias que se aplicarán a cada módulo del programa con el fin de documentarlas debidamente, desde el proceso de instalación hasta los llamados casos de prueba.

Documentación del Código. Se refiere a la explicación de cada rutina o por lo menos de las más importantes o esenciales. La documentación del código se lleva a cabo desde la codificación misma, e inclusive puede encontrarse sobre este, sin embargo nunca sale sobrando un documento aparte donde se clasifiquen los programas y sus descripciones de la manera más pertinente, normalmente siguiendo un orden funcional.

El desarrollo del sistema como se ha visto hasta aquí consta de una serie de actividades referidas al análisis, elaboración, construcción y transición del mismo, sin embargo paralelamente se desarrollan actividades propias de la administración y control del proyecto, actividades que comienzan incluso desde la presentación del proyecto a los usuarios (clientes); a manera de resumen podemos mencionar algunas de ellas:

Como inicio, en el área de recursos humanos se debe de especificar el personal que laborará en el desarrollo del sistema, esta actividad la realiza normalmente un líder de proyecto que en nuestro caso particular será nuestro jefe superior inmediato; éste debe también designar un tiempo estimado para cada fase del proyecto. Como resultado de juntas de trabajo con los usuarios involucrados, se debe presentar minutas donde se especifiquen los acuerdos ahí tomados, teniendo en consideración responsables, fechas de compromiso y las reglas de normatividad que se deben emplear.
En pleno desarrollo se deben registrar por escrito todo tipo de modificaciones requeridas por los usuarios. Una de las actividades principales de un líder de proyecto es llevar un control de actividades para cada miembro del grupo de trabajo evadiendo con ello saturar de trabajo a algunos y evitando tiempos de holgura en otros, este control normalmente se lleva a cabo mediante un reporte de actividades semanal de cada integrante.
Los documentos que se deben emplear normalmente son:

Un plan de trabajo.- donde se describen las actividades que se van a realizar indicando fechas y responsables.

Herramientas Utilizadas.- es una descripción de todas las herramientas técnicas empleadas en el desarrollo del sistema, con su debida licencia si así lo amerita.

Requerimientos.- aquí dividimos en la presentación de los requerimientos por parte de los usuarios debidamente documentados y avalados, y los requerimientos que la misma dirección de sistemas hace a los usuarios para facilitar el

desarrollo del sistema.

Especificación de Reglas de Negocio.- explicación detallada de la normatividad (reglas del negocio) que rige tanto la información manipulada en el proyecto como el desarrollo del mismo.

Matriz de Programas-Archivo y Programas-Entidad.- el primero es una tabla con el nombre del programa como primer columna, archivo o tablas empleadas en la segunda columna, rutinas importantes en la tercera, el segundo es una relación de los programas relacionados en la actividad de una unidad funcional especifica.

Diccionario de datos.-es un glosario de los conceptos empleados en el sistema, consta de una tabla con el nombre del dato, su descripción, unidades de medida u otro atributo de definición, llaves primarias de orden y / o llaves secundarias. Normalmente un diccionario de datos corresponde a la descripción de los campos de una tabla dentro de una base de datos, pero aplicado al universo de los datos del sistema.

Modelo entidad-relación.- una tabla descriptiva de los entes (entidades), que se trabajan en el sistema y la relación entre ellas mismas; comúnmente es un diagrama de las tablas diseñadas dentro de nuestra base de datos y sus respectivas relaciones.

Diagramas elaborados a través de la evolución del proyecto .- diagramas de caso de uso, diagramas de clase, diagramas de secuencia, diagramas de componentes, diagramas de ejecución y diagramas de estados.

Una vez descrita a groso modo la metodología empleada en la DABD para la creación de sistemas de software, pasaremos a describir la forma en que fue implementada sobre el sistema el cual es objeto de análisis del presente trabajo.

Adelantando y resumiendo parte del análisis inicial llevado a cabo, mencionaremos que debido al requerimiento de contar con un módulo emisor y un módulo receptor para manejar la información, esto a causa de que cada contraloría interna tiene un lugar de trabajo extraño al de las instalaciones de la misma SECODAM, incluso en el interior de la Republica Mexicana; el análisis se presentará dividido en dos partes, el módulo del informante donde se lleva a cabo la captura de la información y el módulo central que recibe la información, la analiza y saca conclusiones.

Claro esta que esta división obedece a un análisis previo de las herramientas con que se contaban al momento de iniciar el desarrollo del sistema. Sobre todo la vía de comunicación posible entre uno y otro módulo, que prácticamente nos obligó a generar módulos funcionalmente independientes, obviamente teniendo entre esas funciones el intercambio de información con el otro módulo.

Así pues, en el siguiente capitulo analizaremos el módulo del informante o emisor.

CAPITULO II. DISEÑO DE LA BASE DE DATOS PARA EL MODULO INFORMANTE

Partiendo del objetivo principal que es la creación de un sistema automatizado para el control de auditorias y sus seguimientos practicadas dentro de la administración pública federal, y una vez descrita la metodología empleada para llevar a cabo esta tarea, en este capítulo describiremos las actividades realizadas para uno de los módulos del sistema al que llamaremos módulo del informante.

Análisis de Requerimientos.

Objetivo Principal :

Crear un módulo de captura y envío automático de la información generada por las contralorías internas de cada entidad hacia la Unidad de Seguimiento y Evaluación de la Gestión Pública.

Objetivos específicos:

Estandarizar la forma de presentar la información para todas las entidades, siguiendo una estructura bien definida de la misma.
Automatizar los reportes requeridos por la normatividad de la USEGP dentro del mismo sistema.
Usar métodos de comunicación más rápidos y eficientes para acortar los tiempos de envío-recepción.
Establecer normas de seguridad en los paquetes enviados a / y desde la USEGP.

Identificación de actores:

Dentro del proceso de captura, envío y recepción de información, podemos fácilmente identificar los siguientes actores:

Contraloría Interna.- nos referimos a contraloría interna como el personal que labora en las distintas entidades dentro del órgano de control interno, encargado de la instalación, captura, recepción y envío de la información, es decir los responsables del módulo en la entidad. Dentro de este conjunto de actores, también podemos hacer una clasificación en personal de apoyo técnico, analistas y encargados de la captura de la información.

Dirección General Adjunta de Control y Seguimiento (DGACS).- encargada de la recepción, análisis y retroalimentación de la información dentro de la Unidad de Seguimiento y Evaluación de la Gestión Pública, ubicada dentro de la SECODAM.

Dirección de Administración de Bases de Datos.- encargada de dar soporte técnico tanto a las Contralorías Internas como a la DGACS, desprendemos de aquí que es la responsable del funcionamiento del sistema, de sus posibles actualizaciones y de la generación de manuales de usuario y asesorías técnicas.

Podemos mencionar que el actor más beneficiado con el manejo del nuevo sistema es la DGACS, ya que los grandes volúmenes de información y sus diferentes formatos se cambian por paquetes bien definidos y con una estructura clara y sólida, permitiendo también que la incorporación de estos a su base de datos es más eficiente y de forma automática.

El beneficio dentro de las contralorías internas es menos palpable, ya que no obstante que el manejo del nuevo sistema es muy claro y preciso, la información estandarizada y validada en éste no siempre abarca el universo de la información que estas trabajan; sin embargo esto representaba precisamente uno de los más graves problema dentro DGACS, ya que el hecho de que cada contraloría mandara información que de hecho era irrelevante para esta dirección, provocaba la tediosa tarea de clasificarla y en su caso eliminarla; con el nuevo sistema la DGACS recibe únicamente lo que tiene que analizar.
Por otro lado, las contralorías internas posiblemente tendrán que trabajar la información sobre un sistema interno que abarca el universo de su información y el nuevo módulo de la SECODAM que solo requiere los datos que requiere la DGACS.
De lo mencionado anteriormente podemos definir dentro de los requerimientos del sistema. que éste deberá tener la suficiente claridad en la estructura de su base de datos, así como flexibilidad para lograr una posible convivencia con sistemas alternos desarrollados por las diferentes contralorías internas, es decir que el personal informático de las contraloría pueda fácilmente construir una interfase para la comunicación y el llenado de la información de su sistema al nuestro.
Por lo anterior, los procesos de validación dentro del sistema deben cerrar cualquier alteración no permitida. previendo con esto una buena integridad de la información enviada a la DGACS, no obstante la misma contraloría utilice vías alternas para la captura de la información (Interfaces).

Casos de Uso.

Las principales operaciones que podemos ver dentro del sistema es la captura de la información por parte de las Contralorías Internas, esta captura se clasifica de acuerdo a un calendario y al tipo de información requerida, anualmente capturan el Programa Anual de Control de Auditorías, mensualmente capturan lo concerniente al Sistema de Información Periódica que no es otra cosa que el registro de observaciones que nacen de una auditoría y sus seguimientos así como posibles auditorías no programadas y por último trimestralmente capturan el módulo de recuperaciones y costos.
De los tres módulos de información. solo el segundo el Sistema de Información Periódica mantiene una retroalimentación mensual por parte de la DGACS, los otros dos actúan sin recibir prácticamente ningún dato extra de esta dirección.
Para el envío de la información, se consideró necesario crear procesos estandares para generar la información deseada a fin de crear un solo archivo descriptivo de información y el periodo enviado. la generación de dicho archivo debe implicar la validación de la información que se pretende enviar.
Una vez generado el archivo, nuestra siguiente operación es el envío de este a la DGACS, ocupando vías de transmisión más rápidas y previendo la integridad y seguridad de nuestro paquete, dicha transmisión como cualquier otra debe establecer un acuse de recibido.
Como se mencionó en párrafos anteriores una de las tres formas de información requiere de una constante retroalimentación por parte de la DGACS, por lo que otra de las operaciones de nuestro sistema corresponde a la recepción periódica de esa información.
La generación de reportes y con ello la respectiva validación de la información en papel es uno de los procesos más empleados por las contralorías internas.

En el esquema siguiente se da una explicación de cómo se relacionan las operaciones mencionadas con los actores previamente identificados.

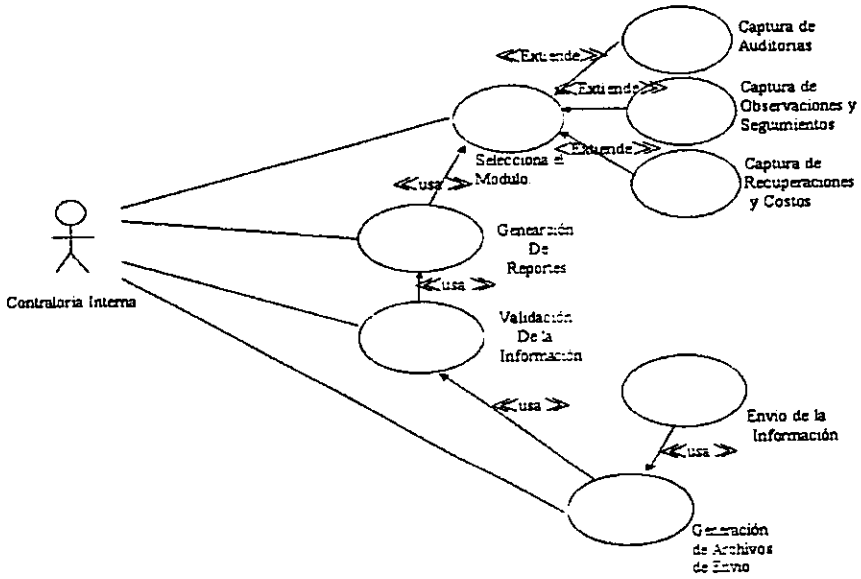


Figura 2.1.

Lo primero que vemos es la relación directa de la Contraloria Interna con los procesos de captura, generación de reportes, validación de información y por último la generación y envío de la misma a la DGACS.

Dentro de las relaciones que notamos entre proceso y proceso está la relación de uso entre el envío del paquete generado y su propia generación, entre la generación del archivo de envio y la validación previa de la información que se va a enviar, estas relaciones son mucho muy estrechas ya que sin uno no puede existir el posterior proceso.

Después tenemos la relación de la validación de la información y la generación de reportes, y entre estos reportes y la previa captura de la información, estas relaciones no son tan estrechas ya que por ejemplo la validación de la información puede ocupar los reportes impresos pero no son imprescindibles.

El proceso general de captura lo podemos extender hacia tres procesos independientes, que viene siendo la clasificación de la información dentro de los tres módulos SIP, PACA y Recuperaciones y Costos.

En realidad el diagrama presentado engloba todas las actividades esenciales del sistema, sin embargo existen casos particulares de rutinas previstas como la generación de respaldos y su proceso contrario la restauración de respaldos; y otro más particular la restauración de mes para modificar información ya enviada, estas actividades las presentamos en el siguiente esquema.

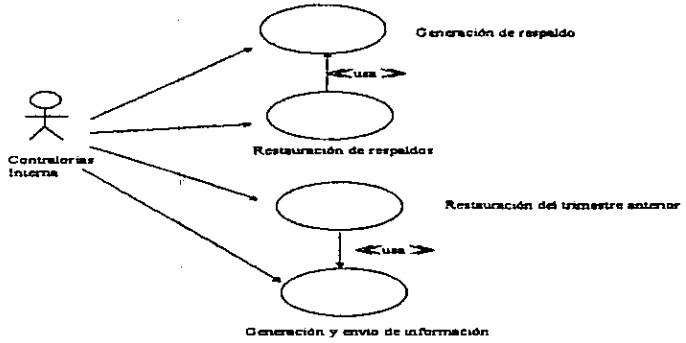


Figura 2.2.

Las relaciones existentes entre los procesos de generación de respaldo y restauración de respaldo es muy estrecha, ya que sin uno no es posible la existencia del otro; al igual que la restauración de la información del trimestre pasado, con la generación y envío de cada trimestre.

El siguiente diagrama de caso de uso representa la segunda parte del proceso dentro del sistema, que es la recepción y análisis de la información enviada por las contralorías internas.

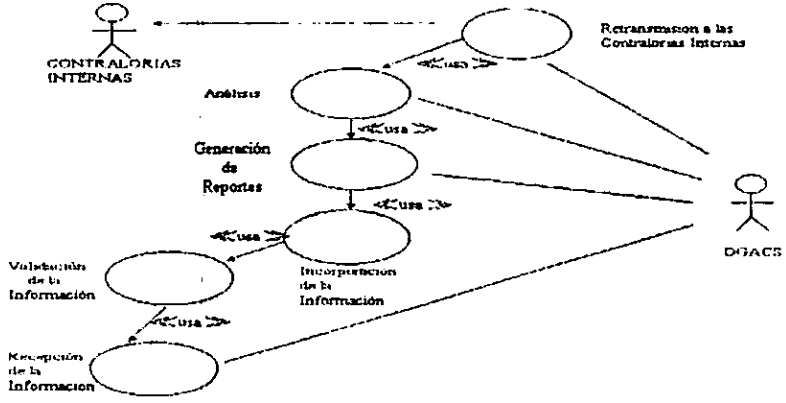


Figura 2.3.

El análisis de este diagrama se profundizará en el siguiente capítulo, donde se habla del desarrollo del módulo central del sistema, por ahora solo en necesario observar que dentro de este caso vemos que existe una posible respuesta por parte de la DGACS hacia las contralorías internas, esto nos lleva a la realización del siguiente caso de uso donde podemos ver la recepción e incorporación de la retroalimentación de datos.

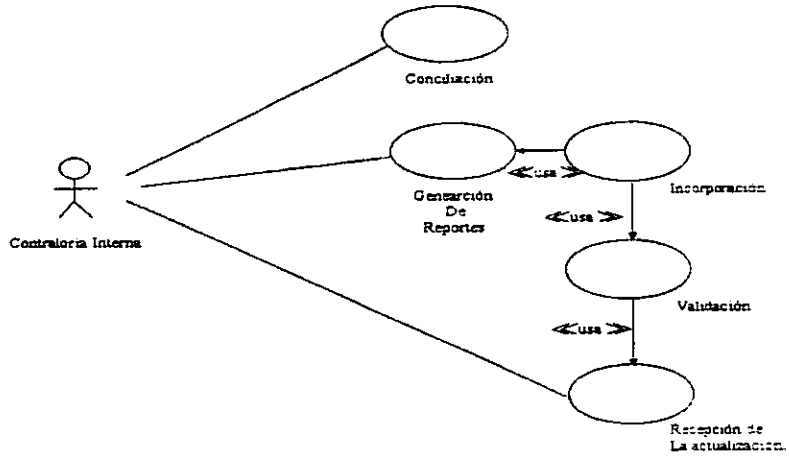


Figura 2.4.

El único proceso que en realidad no se puede automatizar en este módulo es la conciliación entre Contralorías Internas y la DGACS, ya que para ello es necesario la comunicación personal entre uno y otro actor, a través incluso de juntas de trabajo. Por lo demás todos los procesos son fácilmente de automatizar.

De hecho lo único que explica el diagrama anterior es la recepción de la actualización que realiza periódicamente la DGACS, esta recepción lleva consigo la validación de la información recibida y su incorporación en la estructura de los datos del módulo, los procesos de generación de reportes para checar la información recibida y la conciliación son posteriores a dicho proceso.

En los diagramas anteriores no hemos visto a la totalidad de los actores que se identificaron con anterioridad, de hecho la Dirección de Administración de Bases de Datos no actúa como actor primario por lo que actúa solo como asesor técnico para los otros dos usuarios, y esta relación no forma parte de los procesos automatizados.

Diagrama de Clases.

Es necesario comentar que debido al universo extenso de todas las clases empleadas en este sistema, únicamente pondremos las clases mas representativas del mismo, dejando fuera las clases que representan componentes de programación especiales para el lenguaje utilizado, esto es con el objeto de no perdernos en la multitud de clases empleadas. Aclarado lo anterior, el siguiente diagrama de clases presenta las elementos funcionales del sistema.

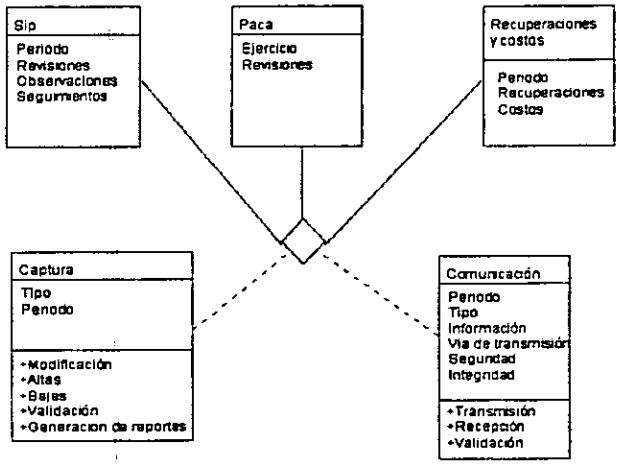


Figura 2.5.

En realidad las tres clases generales del sistema clasifican la información según su estructura y contenido, para lo que es el Sistema de Información Periódica (SIP), se tiene que la información manipulada son tanto revisiones, observaciones y seguimientos de un cierto periodo; para el caso del Programa Anual de Control de Auditorias (PACA) solo manipulamos las auditorias de cierto año o periodo, y para el caso del módulo de recuperaciones y costos, se manipulan como su nombre lo indica las recuperaciones (ingresos) y los costos (egresos) dentro de un periodo de auditorias.

Las tres clases mencionadas se relacionan por medio de dos clases funcionales de asociación que son el manejo de la información y su comunicación con el módulo central en la SECODAM. La primer clase de asociación consta de funciones básicas de manipulación de información como son la inserción, el borrado y la validación de registros, así como la generación de reportes requeridos para cada clase. En la segunda nombrada como la comunicación, se presentan atributos que identifican el tipo de información de envío, el periodo que se esta transmitiendo o recibiendo, la clasificación de la información que se debe de enviar para cada tipo de la misma es decir, si es total o parcial, la integridad de la información y la seguridad de la comunicación.

Una vez establecido nuestro diagrama de clases funcionales, podemos empezar a clarificar las clases reales en que podemos dividir la estructura de la información que se va a trabajar dentro de este esquema.

Dentro de la estructura de nuestros datos podemos fácilmente identificar las siguientes clases:

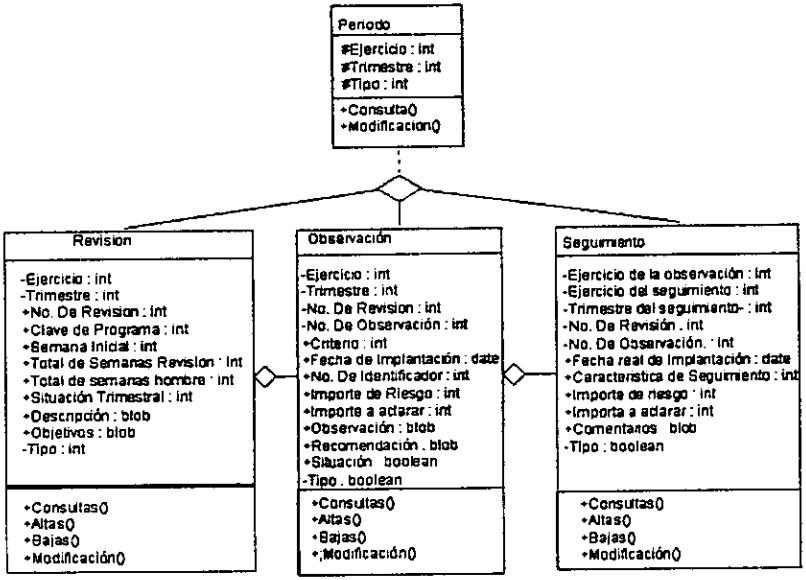


Figura 2.6.

Lo que a simple vista vemos es la relación de composición que existe entre una revisión y sus observaciones, y entre una observación y sus seguimientos. Esto nos dice que no puede haber observaciones sin una liga correspondiente a una revisión existente y a su vez no existen seguimientos sin una observación.

Existe también una clase de asociación que es el periodo que involucra a cada una de las tres clases principales mencionadas.
Comenzaremos a explicar cada clase y sus atributos brevemente.

Clase periodo.- Identifica ya sea el ejercicio y el trimestre dentro de los módulos del SIP y Recuperaciones, o simplemente el año en el caso del PACA; este periodo puede ser de captura, de envío o de consulta según el proceso en el que nos encontremos. Dentro de sus procesos principales esta su consulta, que es necesaria para crear un marco de referencia cronológico, y su modificación para cuando se transmitan la información requerida; este último proceso debe ser automático dentro del proceso de transmisión sin que el usuario, en este caso, las contralorías lo modifiquen directamente.

Clase de Revisión.- guarda los atributos propios de la información que maneja. Las auditorías tiene un número de identificación, una clave de programa, una semana inicial, el total de las semanas que llevará su realización, el total de las semanas por hombre empleado por trimestre, la situación que guarda en cada trimestre, la descripción detallada de la misma auditoria , sus objetivos y además el tipo de auditoria que se trata de acuerdo a la clasificación que se dio en la introducción de este documento, auditorias integrales, específicas, de evaluación de programas, de seguimiento o de desempeño; cuentan también con un ejercicio y un trimestre es decir un periodo que las identifica cronológicamente.

Los métodos empleados de esta clase son la consultas, modificaciones, altas y bajas; la mayoría de estas actividades dependen del periodo de captura en que se encuentren los dos módulos donde se manipulan las auditorias, así por ejemplo en el módulo del PACA solo se podrán consultar y en dado caso modificar, dar de alta o borrar las revisiones pertenecientes al periodo de programación que siempre esta desfasado un año arriba del periodo de captura del módulo del SIP.

Para las auditorias pertenecientes al año en curso o las pendientes de años anteriores se debe recurrir al módulo del SIP, además la información manejable en este caso se reduce a la captura de la situación trimestral que guardan estas auditorías, así como sus semanas hombre por trimestre. Sin embargo también en este módulo es posible dar de alta una revisión con todos sus atributos, agregando una marca que la distinga como adicional al programa inicial.

Clase de Observación.- Una observación como ya se vio anteriormente, representa una anomalía en las actividades normales de la entidad auditada, que se registra en la realización de una auditoria en especial. Por lo anterior el primer atributo visible es el número de revisión al que pertenecen, su año y su trimestre, el número de identificación dentro de la misma revisión, la fecha de implantación o registro, un número de identificador de programa y subprograma, una situación trimestral, un importe de riesgo y uno de aclaración que manifiestan montos involucrados en la observación, la descripción de lo que se observó dentro de la auditoria, las recomendaciones que da el órgano interno de control para su solvencia, y un tipo que nos indica si la observación es histórica o se esta dando de alta en el trimestre actual.
Esta clase solo es manejable dentro del módulo del SIP, donde se llevan acabo sus diferentes métodos de consulta, modificación, altas o bajas, estos tres últimos dependen de un periodo actual de captura.

Clase de Seguimiento.- a partir de que se registra una observación, incluyendo ese mismo trimestre se le puede dar seguimiento, que no es otra cosa que registrar la forma en que se implementan las medidas necesarias para aclarar o solventar la observación. Un seguimiento por lo tanto debe contener el número de revisión al que pertenece la observación , el identificador de esta misma incluyendo su periodo, debe tener un año y trimestre que identifique el periodo en que se esta dando de alta dicho seguimiento, de tener un importe de riesgo o uno de aclaración que visualizan como se viene manejando los importes de riego y aclaración registrados cuando se dio de alta la observación, la característica del seguimiento que indican como evoluciona la situación de la observación, contiene una fecha de implantación real que indica la fecha en que se registra el seguimiento, y además una descripción de cómo se vienen realizando las actividades correctivas.

Al igual que las observaciones, los seguimientos solo son manipulables dentro del periodo actual de captura, cualquier otro periodo solo es de consulta.

En la figura 2.6 se representa solo los datos que se manejaran en dos de los tres módulos clasificados en el diagrama de clases funcionales. El tercer módulo que identifica las recuperaciones y costos se trata en un diagrama diferente, debido a la independencia que guarda la información que se maneja en éste con respecto al los módulos anteriores.

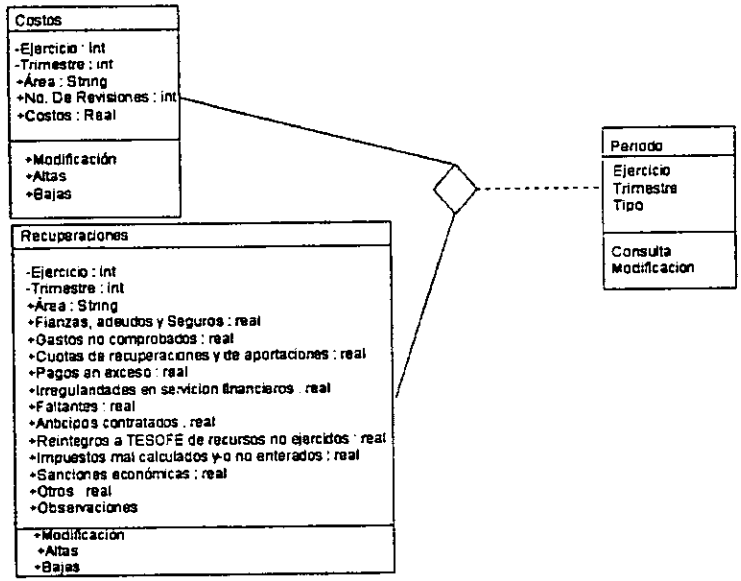


Figura 2.7.

Como vemos aquí las dos clases principales en este módulo se asocian mediante una tercer clase identificada y descrita en la figura 2.7 como la clase de periodo. Las dos clases no tienen ninguna otra relación que no sea el periodo de su captura.

Clase de costos.- identifica el monto, el número de revisiones y el área administrativa de la entidad de los egresos (costos) en el desarrollo de las auditorías, tiene además un periodo de identificación. Sus métodos son simples de consulta, modificación, altas y bajas; y dependen al igual de que en los módulos anteriores de un periodo actual de captura.

Clase de recuperaciones. Las recuperaciones hablan de montos de ingreso en las diferentes actividades clasificadas de acuerdo a su giro, esta clasificación representa cada uno de los atributos de esta clase. una recuperación también puede tener una observación descriptiva o comentario respecto a dichos montos. Sus métodos son similares al de las clases descritas anteriormente, consultas, bajas, altas y modificaciones.

Existe una clase independiente para el manejo de la información donde podemos colocar atributos necesarios para la identificación de cada dependencia, esta clase contendrá los datos necesarios que prevengan la generación de un paquete único de envio para cada entidad y periodo.

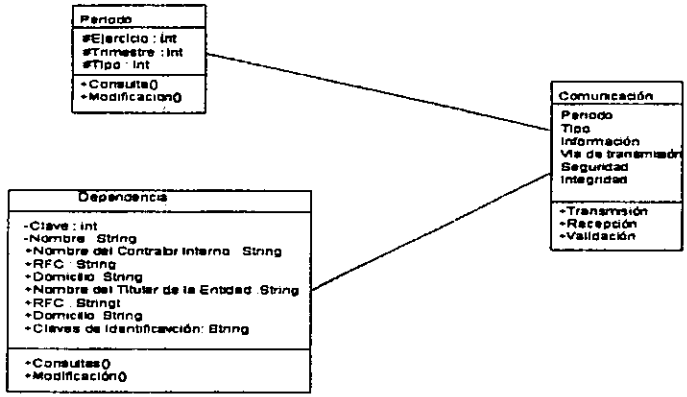


Figura 2.8.

La relación presentada aquí es entre la clase de dependencia y la clase de periodo con la clase funcional de la comunicación, ya que de las primeras dos se obtienen los datos necesarios para la creación de un paquete perfectamente identificado por cada dependencia y por cada periodo de información.

Diagramas de Secuencia.- Una vez elaborados los diagramas representativos de las clases que identificamos en el desarrollo del módulo del informante, nos disponemos al desarrollo de los diagramas de secuencia para explicar cada uno de los procesos que debe llevar a cabo cada uno de los objetos.

Para ello presento el siguiente esquema.

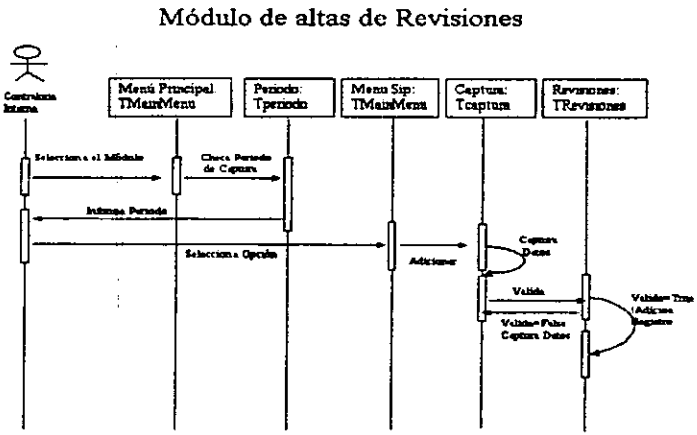


Figura 2.9.

El diagrama pertenece a la forma en que se da de alta una revisión.

- Primero el usuario debe seleccionar el módulo en el que va a dar del alta esta información mandando un mensaje al objeto "Menú Principal" que es una instancia de la clase TmainMenu que guarda todas las propiedades y eventos de un menú de opciones horizontal ligado a una pantalla o forma principal.
- Al elegir el módulo, este objeto checa de una instancia de "periodo" el ejercicio y el trimestre actual de captura para el módulo seleccionado.
- Una vez conocido el periodo, el usuario debe seleccionar de un segundo menú la opción de insertar revisión, este segundo menú manda un mensaje a una clase funcional de captura para adicionar un registro.
- Los procesos que se llevan a cabo son la captura de los datos y el envió de un mensaje de validación a una instancia de "Revisión", esta instancia se encarga de validar los datos y elegir entre dar de alta un registro o volver a la clase de captura para capturar de nueva cuenta la información.

De igual forma los diagramas de secuencia para cada inserción ya sea de una observación, o seguimiento arrastran el proceso antes mencionado. Por lo que me resulta inecesario, no obstante se hayan elaborado en la documentación del proyecto, repetir el mismo diagrama con sus respectivas particularidades.

El diagrama de secuencia para la generación y transmisión de la información.

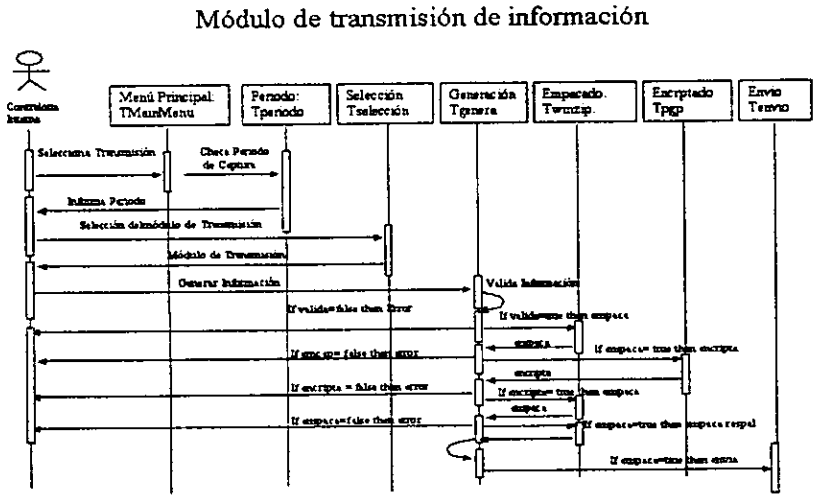


Figura 2.10.

El proceso representado en el diagrama tiene la siguiente secuencia:

- Primero el usuario manda un mensaje al menú inicial seleccionando los procesos de transmisión de datos, antes de presentar dichos procesos se checa el periodo actual de transmisión; posteriormente y ya teniendo el periodo de

transmisión, el usuario selecciona el módulo de transmisión, pudiendo elegir entre las tres opciones posibles SIP, PACA o Recuperaciones y Costos.

- Independientemente del módulo seleccionado, el usuario debe dar la orden para generar la información de envío. El módulo de generación de envío valida la información y según el resultado manda la información validada para su empacado o bien manda un mensaje de error al usuario, que para entonces lleva a cabo un proceso de monitoreo.
- El objeto encargado de empacar la información da como resultado un archivo compactado que contiene la información generada o en su caso un mensaje de error al objeto de generación, mismo que manda el error al usuario o manda un mensaje al objeto de encriptado para que realize su función sobre la información ya empacada.
- El encriptado pasa como respuesta un error o un archivo cifrado nuevamente a la instancia generadora, que al igual que antes manda el error al usuario o manda un mensaje nuevamente al empacador.

El mismo proceso anteriormente descrito se lleva a cabo otra vez pero como resultado de la nueva compactación.

- El generador envía un mensaje para un nuevo empacado pero esta vez se empacará un respaldo de la información, que una vez terminado manda un mensaje al generador para que este comience un proceso de depuración de la base de datos, y por último mandará un mensaje al objeto encargado del envío de la información doblemente empacada y encriptada.

La doble compactación obedece a la prevención del posible multivolumen del archivo generado.

La instancia encargada del envío tiene rutinas de conexión a través de sockets a un posible servidor de correo electrónico, o bien el traslado del archivo generado hacia un disco físico.

La depuración de las bases mencionada entre las actividades del proceso de transmisión, tiene como objetivo el eliminar la información que quedo solventada en el actual periodo, y bien podemos mostrar su diagrama de secuencia.

Módulo Depuración de bases

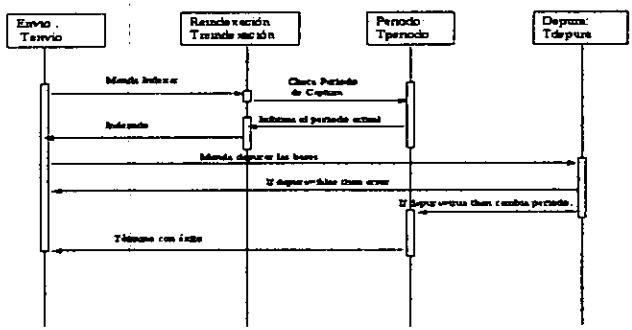


Figura 2.11.

- El módulo de envío manda un mensaje hacia un módulo de reindexación.
- Este módulo antes de realizar su función debe saber cual es el periodo actual, por lo que lo obtiene de una instancia de periodo. Una vez obtenido el periodo realiza su función de eliminar cualquier información no deseada o fuera de orden; esto principalmente se realiza para evitar errores no tanto en la captura directa en el sistema, sino para errores en las posibles interfaces de usuarios externos.
- El módulo de reindexación manda un error o un mensaje de éxito al envío.
- El módulo de envio, da la orden de depuración al módulo respectivo; este realiza su función, que es el de eliminar primero todos los seguimientos marcados como finalizados, posteriormente realizar la eliminación de las observaciones de las cuales se desprenden los susodichos seguimientos y por último checa si la eliminación de esas observaciones da como resultado el fin de su revisión y en este caso la elimina. Una vez realizado el proceso manda ya sea un mensaje de error al módulo de envío, o una orden de cambiar el periodo actual a la instancia del periodo
- Por último, la instancia de periodo manda un mensaje de éxito al modulo de envío, que realizará su siguiente tarea explicada en el diagrama anterior.

Para objeto de estudio de cómo se realizó el diseño de este módulo que llamamos del informante, los diagramas presentados hasta aquí son los más representativos. Evitamos mostrar todos los realizados junto con sus descripciones primero para evitar conflictos con la política de la dirección de trabajo, segundo no hacer muy extensa la explicación de nuestro trabajo.

Diseño.

Diagrama de Componentes.

Como se explicó anteriormente, en esta etapa se desarrollan diagramas de componentes y de distribución, además se detallan los diagramas de clases con información obtenida de la implementación de un prototipo.

En el módulo del informante se pueden localizar los siguientes componentes.

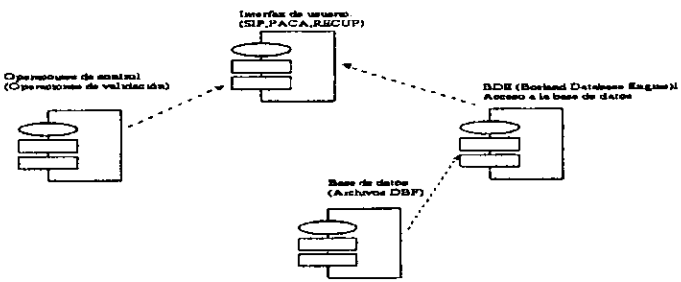


Figura 2.12.

Dentro de los componentes mostrados tenemos primero la interfaz de usuario, que en este caso dividiríamos en tres partes, las pantallas de captura del módulo de SIP, las de recuperaciones y costos y la del PACA todas las reunimos en un solo componente.
Reunimos también cada una de las operaciones aritméticas y de control en otro componente que llamamos de control, aquí es necesario notar que esta división en puramente simbólica ya que en la programación estos dos componentes están estrechamente unidos.

La forma más común de enlazar una interfaz de usuario con la información en el lenguaje utilizado (Delphi 5), es a través del DBE (Borland Database Engine), que no es otra cosa que un intermediario entre los componentes de accesos a la bases de datos propios del lenguaje y la misma base de datos, éste provee de todas las rutinas de control y acceso a la información dentro una estructura de datos que posiblemente sea variable. Es decir que el BDE nos da una independencia entre la programación de acceso a los datos y la estructura de los mismos.
El último componente que vemos es el que encierra la estructura de los datos. en este caso estamos hablando de la utilización de archivos DBF, el por que de utilizarlos primero nos remonta a la historia del sistema en cuestión, donde existe un antecedente de este tipo de estructura para el manejo de la información, y por esto mismo existe el requerimiento de seguir con ellos para evitar un cambio drástico en posibles sistemas alternos fuera de nuestro conocimiento. La flexibilidad que dan estos tipos de archivos nos da la ventaja de que cualquier tipo de mantenimiento u operaciones de corrección se puedan llevar a cabo con la menor complicación, teniendo en cuenta que este módulo va a trabajar fuera de nuestras instalaciones, y peor aún sin soporte directo de nuestra parte, esta ventaja es de gran utilidad. Por otra parte. la utilización de estos archivos nos dan la tarea de poner más énfasis a las operaciones de control y validación dentro del sistema, para evitar que cualquier tipo de manipulación no permitida a la tablas de la base de datos sea detectada y en su caso corregida por el sistema, o por lo menos que este tenga la capacidad de indicar el fallo al usuario para que el tome las medidas necesarias.

A continuación explicaré un segundo diagrama de componentes que envuelven otras operaciones importantes de este módulo.

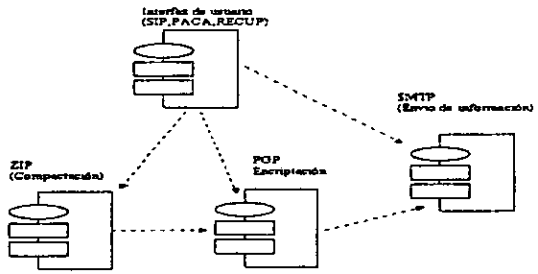


Figura 2.13.

En el diagrama (figura 2.13) definimos tres componentes más, uno encargado del proceso de compactación y descomprensión de la información, un segundo encargado de envolver algoritmos de encriptación de desencriptación y por último el componente encargado de realizar la transmisión de la información que se requiere. Estos tres componentes se ligan a una interfaz de usuario que se encarga de mandar todas estas operaciones.

Diagramas de Ejecución.

Una vez realizado los anteriores diagramas de componentes podemos fácilmente agruparlo de la siguiente forma.

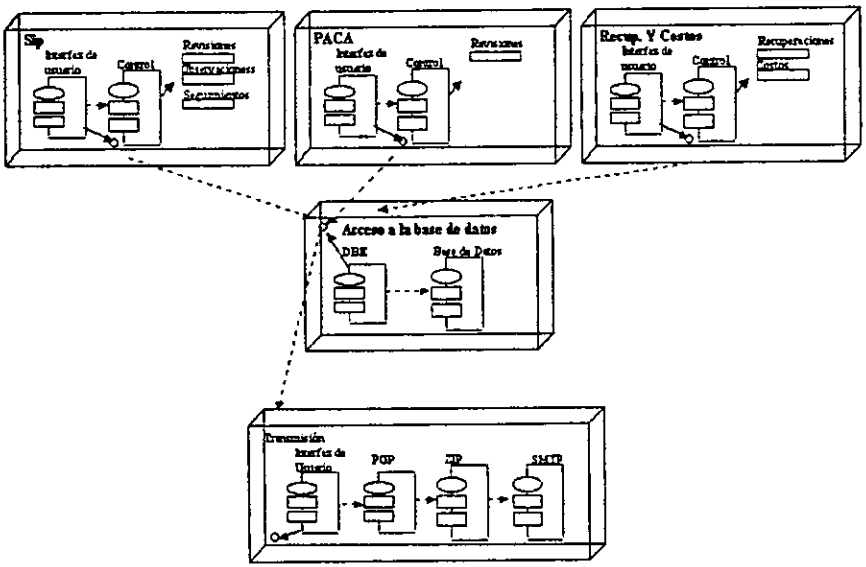


Figura 2.14.

Lo que tratamos de explicar en este diagrama es la agrupación de componentes por módulo de información , podemos pues ver que los tres tipos de información tienen su respectiva interfaz de usuario, y sus operaciones de control, también podemos ver en cada nodo las clases de información manipulada en él. En el caso del SIP vemos las clases de Revisión, Observación y Seguimiento; el PACA únicamente la de revisión y en el de Recuperaciones y costos sus dos datos trabajados (Costos y recuperaciones).

Estos tres paquetes se enlazan con un tercero que envuelve los componentes del BDE y de la base de datos, y a su vez estos en su debido momento con un tercer paquete que nombramos de transmisión, cuyos componentes son el PGP, ZIP y SMTP además de su interfaz de usuario.

Diagramas de Estado.

Una vez identificados y explicados los componentes del módulo del informante, procederemos a la elaboración de los diagramas de estados para las clases más representativas del sistema.

El primero que mostraré es el de la clase de revisión, que prácticamente es el alma del sistema, procederemos con el de una observación y el de un seguimiento.

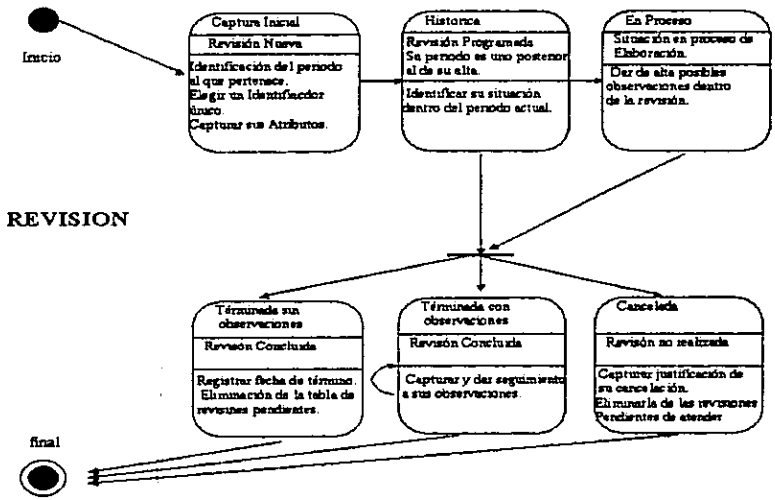


Figura 2.15.

Como vemos en el diagrama de estados una instancia de revisión se crea en el momento que el usuario la da de alta en el sistema, este proceso como hemos visto puede realizarse en el módulo del PACA o en el módulo del SIP, de tal forma lo único que las diferencia es el periodo de alta; en el primer módulo siempre será un año posterior al de captura y en el segundó el periodo será igual al actual.

Una vez transmitido el periodo (PACA o SIP), la revisión pasa a un estado histórico, donde tiene registrados todos sus atributos excepto su situación y su fecha de término.
Toda vez que se capturen estos dos datos, la revisión puede pasar a uno de los cuatro estados siguientes.

Situación en proceso.- esto quiere decir que la auditoria se sigue realizando en el periodo actual, en este estado el usuario puede capturar posibles observaciones que surjan en su realización, he incluso puede capturar una fecha aproximada de terminación.

Terminada sin Observaciones.- La auditoria se terminó y en su desarrollo no se encontraron anomalías en la actividad auditada, en este caso solo se captura la fecha de término y se marca para su borrado en el siguiente proceso de transmisión.

Terminada con Observaciones.- La auditoria se terminó y en su transcurso se registraron "n" observaciones que requieren de un seguimiento posterior a la realización de la revisión; se anota la fecha de término pero no se marca para su borrado. Este estado se repite durante "n" periodos hasta que la revisión no tiene ya ninguna observación pendiente.

Cancelada.- en este estado se indica que la revisión no se llevo a cabo y es necesario explicar el porque. Aquí la revisión si se marca para su borrado.

Otro detalle que podemos observar en el diagrama es que del estado de "Histórico" se puede llegar a cualquiera de los estados que terminan con la revisión directamente o primero pasar a un estado intermedio de "Proceso". Esto permite que las revisiones se elaboren no solo en un periodo sino dentro de dos o más de ellos.

Diagrama de estados de una observación.

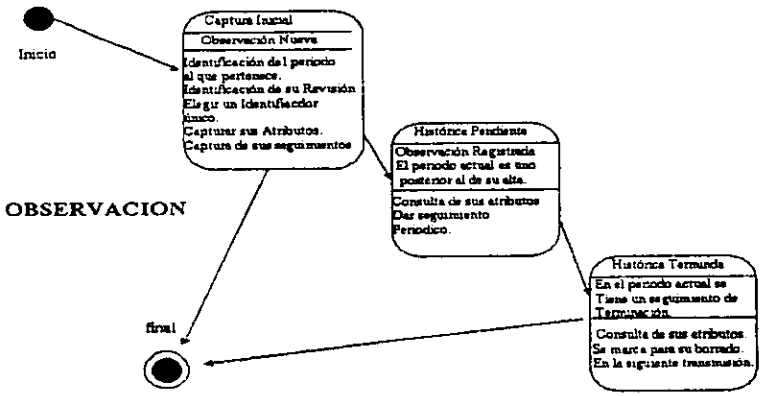


Figura 2.16.

Para una observación se tiene únicamente tres estados posibles dentro de su periodo de vida.

El primero de ellos es cuando se da de alta en el sistema, para ello es necesario identificar el periodo de captura y la revisión a la que pertenecerá, así como captura sus otros atributos. En este estado se puede dar de alta un seguimiento a la observación, e incluso este seguimiento puede darla por terminada sin pasar a otro estado.

El segundo estado es cuando dicha observación ya se transmitió a la SECODAM, por lo que ahora es una observación histórica y además pendiente de atender, en este estado sus atributos son únicamente de consulta para el usuario, y se le pueden dar seguimientos periódicos hasta que pase al siguiente estado.

Por último el estado de término se caracteriza porque en el periodo actual de captura se da de alta un seguimiento con característica de terminación. Los atributos de la observación siguen siendo de consulta, pero indirectamente se modifica el atributo que marca a la observación para su borrado posterior en el proceso de transmisión del periodo.

Diagrama de estados de un seguimiento.

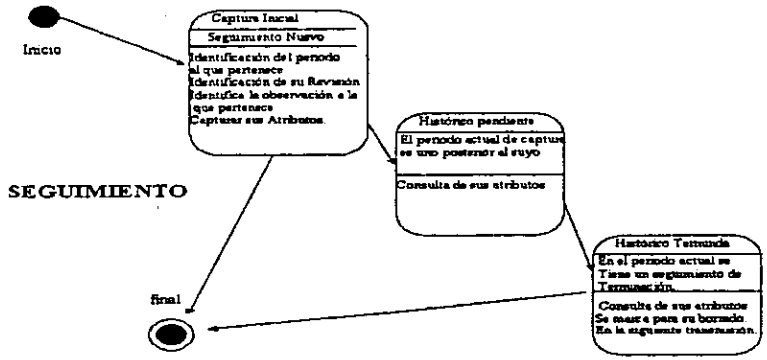


Figura 2.17.

El diagrama es similar al de una observación y sigue las mismas reglas para el paso de uno a otro estado. No obstante existe una gran diferencia entre uno y otro concepto sobre todo con respecto a su cronología, mientras una observación pertenece a un solo periodo, un seguimiento trabaja dos instancias de periodo en sus atributos, uno que identifica al periodo de su observación y otro que identifica el periodo de captura del mismo seguimiento.

Sin embargo, para llegar a su estado de terminación al igual que en las observaciones depende de una instancia de seguimiento diferente a el, con un valor de terminación en su de atributo de "característica de seguimiento".

Los anteriores tres diagramas de estados son representativos de la información central del sistema, y realmente nos dan una idea fija de cómo se trabaja la información dentro del sistema. Con esto justifico el no incluir en el presente documento la implementación de los diagramas para las demás clases identificadas.

Implementación y Pruebas.

La implementación de este módulo tuvo varias etapas, primero se definió el lenguaje de programación que no fue otro que el Delphi 5 que es el lenguaje utilizado por la SECODAM. Se tomó la decisión de utilizar el BDE para el manejo de base de datos, por lo que el instalador debería llevarlo consigo.

En esta etapa se entregaron varios esquemas utilizados en la documentación del sistema, como son los diagramas de la relación de las tablas, la relación de columnas que viene siendo la descripción de los atributos de las clases antes mencionadas, referencias cruzadas programa-tabla, y por supuesto el código de programación.
Los anteriores esquemas son los siguientes.

Relación de Tablas. – Nombre y descripción de cada tabla dentro de la bases de datos.

ACUS	ESTADISTICAS DE RECEPCION Y/O ENVIO
AYUCLV	RELACION DE CLAVES Y NOMBRES PARA PROCESOS DE REVISION
COSTO	DATOS DE LOS COSTOS DE LA ENTIDAD Y/O DEPENDENCIA
EJERCIO	AÑO DEL EJERCICIO ACTUAL
INFBAS	DATOS DE LA ENTIDAD Y/O DEPENDENCIA
MONTOS	GUARDA LOS MONTOS DE INTEGRACION DE FUERZA DE TRABAJO
OBSERVA	OBSERVACIONES DE LOS SEGUIMIENTOS ENVIADOS
OBTRIM	PARA REALIZAR ENVIOS A LA D.G.A.C.S.
PACA	DATOS DEL PACA CAPTURADOS POR LA ENTIDAD
PACA_PROG	ESTADO DE LA PROGRAMACION DEL PACA
PERIODO	MES ACTUAL PARA EL SIP
PROGRAMA	CLAVES DE PROGRAMA ASI COMO SU NOMBRE
REC_PER	TRIMESTRE ACTUAL PARA RECUPERACIONES Y COSTOS
RECUP	DATOS DE RECUPERACIONES ENTIDAD Y/O DEPENDENCIA
RESPON	DATOS DEL RESPONSABLE SOBRE LOS COSTOS
REVISION	DATOS HISTORICOS DE LAS REVISIONES
REVTRIM	REVISIONES MENSUALES QUE ENVIA LA ENTIDAD
SEGTRIM	SEGUIMIENTOS MENSUALES QUE ENVIA LA ENTIDAD
SEGUI	DATOS HISTORICOS DE LOS SEGUIMIENTOS

Diagrama de Relación.- Relaciones entre tablas.

REVISION	
EJERC	N(04)
TRIM	N(01)
N_REV	N(04)
CLV_P	N(04)
S_INI	N(02)
TOT_SEM_R	N(02)
TOT_SEM_H	N(04)
SIT_TRI	N(02)
FEC_TER	F(08)
SEM_H_T	N(05)
MARCA	N(01)
DESC_REV	M(10)
OBJETIVOS	M(10)

OBSERVA	
EJER	N(04)
TRIM	N(01)
NO_REV	N(04)
NO_BÓXER	N(04)
CRITERIOS	C(04)
OBSER	M(10)
IMP_RIES	N(13)
ACLARAR	N(13)
RECOM	M(10)
FEC_IMP	F(08)
NO_IDEN	N(02)
MARCA	C(01)

REVTRIM	
EJER	N(04)
TRIM	N(01)
N_REV	N(04)
CLV_P	N(04)
S_INI	N(02)
TOT_SEM_R	N(02)
TOT_SEM_H	N(04)
SIT_TRI	N(02)
FEC_TER	F(08)
SEM_H_T	N(05)
MARCA	N(01)
DESC_REV	M(10)
OBJETIVOS	M(10)

OBTRIM	
EJER	N(04)
TRIM	N(01)
NO_REV	N(04)
NO_OBSER	N(04)
CRITERIOS	C(04)
OBSER	M(10)
IMP_RIES	N(13)
ACLARAR	N(13)
RECOM	M(10)
FEC_IMP	F(08)
NO_IDEN	N(02)
MARCA	C(01)

SEGTRIM	
EJER_A	N(04)
SEGUI	N(05)
NO_REV	N(04)
NO_OBSER	N(04)
FEC_REAL	F(08)
CARAC_SEG	N(02)
COMENTARIO	M(10)
IMP_RIESGO	N(13)
IMP_ACLARA	N(13)

SEGUI	
EJER_A	N(04)
SEGUI	N(05)
NO_REV	N(04)
NO_OBSER	N(04)
FEC_REAL	F(08)
CARAC_SEG	N(02)
COMENTARIO	M(10)
IMP_RIESGO	N(13)
IMP_ACLARA	N(13)

PACA	
EJERC	N(04)
TRIM	N(01)
N_REV	N(04)
CLV_G	N(04)
S_INI	N(02)
TOT_SEM_R	N(02)
TOT_SEM_H	N(04)
DESC_REV	M(10)
OBJETIVOS	M(10)
CLAVE	N(02)
CLV_P	N(04)

MONTOS	
FUNCIONES	N(10)
CAPACITA	N(10)
VACACIONES	N(10)
ACT_NO_REU	N(10)
ACT_ADMON	N(10)
TOT_FUN	N(10)
TOT_CAPA	N(10)
TOT_VACA	N(10)
TOT_REU	N(10)
TOT_ADMON	N(10)

RECUP	
EJER	N(04)
TRIM	N(01)
TOTAL	N(13,2)
FIANZ	N(13,2)
GASTO	N(13,2)
CUOTAS	N(13,2)
PAGOS	N(13,2)
IRREG	N(13,2)
FALTA	N(13,2)
ANTIC	N(13,2)
REINT	N(13,2)
IMPUE	N(13,2)
SANCI	N(13,2)
OTROS	N(13,2)
AREAS	C(40)
NOTA1	M(10)

COSTO	
EJER	N(04)
TRIM	N(01)
AREAS	C(40)
NO_REVS	N(04)
COSTO	N(13)
MARCA	C(01)

REC_PER	
PEJER	N(04)
PTRIM	N(01)

PERIODO	
PEJER	N(04)
PTRIM	N(01)

AYUCLV	
NOMBRE	C(01)
CLAVE	N(02)
DESC	C(30)

EJERCIO	
EJERCIO	N(04)

INFBAS	
CLV_ENT	N(05)
DESC1	C(50)
DESC2	C(50)
DNOMBRE	C(45)
DPUESTO	C(45)
DRFC	C(13)
DCALLE	C(45)
DCOLONIA	C(25)
DCIUDAD	C(20)
DCP	N(05)
DEDO	C(20)
DLADA1	N(03)
DTEL1	N(09)
DEXT1	N(05)
DLADA2	N(03)
DTEL2	N(09)
DEXT2	N(05)
CNOMBRE	C(45)
CPUESTO	C(45)
CRFC	C(13)
CCALLE	C(45)
CCOLONIA	C(25)
CCIUDAD	C(20)
CCP	N(05)
CEDO	C(20)
CLADA1	N(03)
CTEL1	N(09)
CEXT1	N(05)
CLADA2	N(03)
CTEL2	N(09)

ACUS	
TIPO	N(01)
ACUSE	C(21)
EJER	N(04)
TRIM	N(01)
FEC_GEN	F(08)
FEC_REC	F(08)
NO_REV	N(10)
NO_OBS	N(10)
NO_SEG	N(10)
NO_REC	N(10)
NO_COST	N(10)

RESPON	
EJER	N(04)
N_REV	N(04)
N_OBS	N(04)
NOMBRE	C(70)
RFC	C(14)

PACA_PROG	
CLV_P	N(04)
DES_P	C(40)

PROGRAMA	
CLV_P	N(04)
DES_P	C(40)

CORREO	C(50)
CEXT2	N(05)
DIREC	C(60)
CLAVES	C(60)

Relación de Columnas.- Descripción de cada campo dentro de cada tabla de la base de datos.

TABLA: ACUS		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
TIPO DE ENVIO	TIPO	N(01)
CLAVE PARA ACUSE	ACUSE	C(2)
EJERCICIO QUE SE ENVIA	EJER	N(04)
TRIMESTRE QUE SE ENVIA	TRIMES	N(01)
FECHA DE GENERACION DEL ENVIO	FEC_GEN	F(08)
FECHA DE RECEPCION	FEC_REC	F(08)
NUMERO TOTAL DE REVISIONES QUE SE ENVIAN	NO_REV	N(10)
NUMERO TOTAL DE OBSERVACIONES ENVIADAS	NO_OBS	N(10)
NUMERO TOTAL DE SEGUIMIENTOS ENVIADOS	NO_SEG	N(10)
NUMERO TOTAL DE RECUPERACIONES ENVIADAS	NO_REC	N(10)
NUMERO TOTAL DE COSTOS QUE SE ENVIAN	NO_COST	N(10)

TABLA: AYUCLV		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
CLAVE DE EMISOR DEL PROGRAMA	NOMBRE	C(01)
NUMERO DE CLAVE DE PROGRAMA	CLAVE	N(02)
DESCRIPCION DE CLAVE DEL PROGRAMA	DESC	C(30)

TABLA: COSTO		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
AÑO DE EJERCICIO DEL COSTO	EJER	N(04)
TRIMESTRE DEL COSTO	TRIMES	N(02)
NOMBRE DEL AREA PARA EL COSTO	AREAS	C(40)
TOTAL DE REVISIONES	NO_REVS	N(04)
VALOR DEL COSTO	COSTO	N(13)
BANDERA PARA EL ENVIO DEL COSTO	MARCA	C(01)

TABLA: EJERCIO		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
AÑO DEL EJERCICIO	EJERCIO	N(04)

TABLA: INFBAS		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
NUMERO DE LA ENTIDAD	CLV_ENT	N(05)
NOMBRE DE LA ENTIDAD PARTE UNO	DESC1	C(50)
NOMBRE DE LA ENTIDAD PARTE DOS	DESC2	C(50)
NOMBRE DEL TITULAR DE LA ENTIDAD	DNOMBRE	C(45)
PUESTO DEL TITULAR DE LA ENTIDAD	DPUESTO	C(45)
RFC DEL TITULAR DE LA ENTIDAD	DRFC	C(13)

CALLE DEL TITULAR DE LA ENTIDAD	DCALLE	C(45)
COLONIA DEL TITULAR DE LA ENTIDAD	DCOLONIA	C(25)
CIUDAD DEL TITULAR DE LA ENTIDAD	DCIUDAD	C(20)
CODIGO POSTAL DEL TITULAR DE LA ENTIDAD	DCP	N(05)
ESTADO DEL TITULAR DE LA ENTIDAD	DEDO	C(20)
TELEFONO LADA DEL TITULAR DE LA ENTIDAD	DLADA1	N(03)
TELEFONO DEL TITULAR DE LA ENTIDAD	DTEL1	N(09)
EXTENSION DEL TITULAR DE LA ENTIDAD	DEXT1	N(05)
CLAVE LADA DEL TITULAR DE LA ENTIDAD	DLADA2	N(03)
TELEFONO DEL TITULAR DE LA ENTIDAD	DTEL2	N(09)
EXTENSION DEL TITULAR DE LA ENTIDAD	DEXT2	N(05)
NOMBRE DEL TITULAR DEL OIC	CNOMBRE	C(45)
PUESTO DEL TITULAR DEL OIC	CPUESTO	C(45)
RFC DEL TITULAR DEL OIC	CRFC	C(13)
CALLE DEL TITULAR DEL OIC	CCALLE	C(45)
COLONIA DEL TITULAR DEL OIC	CCOLONIA	C(25)
CIUDAD DEL TITULAR DEL OIC	CCIUDAD	C(20)
CODIGO POSTAL DEL TITULAR DEL OIC	CCP	N(05)
ESTADO DEL TITULAR DEL OIC	CEDO	C(20)
CLAVE LADA DEL TITULAR DEL OIC	CLADA1	N(03)
TELEFONO DEL TITULAR DEL OIC	CTEL1	N(09)
EXTENSION DEL TITULAR DEL OIC	CEXT1	N(05)
CLAVE LADA DEL TITULAR DEL OIC	CLADA2	N(03)
TELEFONO DEL TITULAR DEL OIC	CTEL2	N(09)
EXTENSION DEL TITULAR DEL OIC	CEXT2	N(05)
CLAVE DE ENCRIPTAMIENTO DEL PGP	DIREC	C(60)
SIN USO	CLAVES	C(60)
DIRECCION DE CORREO ELECTRONICO DEL OIC	CORREO	C(50)

TABLA: MONTOS		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
VALOR DE MONTO DE FUNCIONES Y QUEJAS	FUNCIONES	N(10)
VALOR DE MONTO PARA CAPACITACION	CAPACITA	N(10)
VALOR DE MONTO PARA VACACIONES, VACANTES, INCAPACIDADES	VACACIONES	N(10)
VALOR DE MONTO DE ACTIVIDADES QUE NO CAUSAN REMUNERACION	ACT_NO_REU	N(10)
VALOR DE MONTO DE ACTIVIDADES ADMINISTRATIVAS.	ACT_ADMON	N(10)
TOTAL DE MONTO PARA FUNCIONES	TOT_FUN	N(10)
TOTAL DEL MONTO PARA CAPACITACIONES	TOT_CAPA	N(10)
TOTAL DEL MONTO PARA VACACIONES	TOT_VACA	N(10)
TOTAL DEL MONTO PARA REQUISITOS DE REVISION	TOT_REU	N(10)
TOTAL DEL MONTO PARA ACTIVIDADES ADMINISTRATIVAS	TOT_ADMON	N(10)

TABLA: OBSERVA		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
AÑO DE EJERCICIO DE LA OBSERVACION	EJER	N(04)
MES DE LA OBSERVACION	TRIMES	N(02)
NUMERO DE LA REVISION	NO_REV	N(04)
NUMERO DE OBSERVACION	NO_OBSER	N(04)
CLAVE DE CRITERIO	CRITERIOS	C(04)
DESCRIPCION DE LA OBSERVACION	OBSER	M(10)
MONTO DEL IMPORTE DE RIESGO	IMP_RIES	N(13)
CANTIDAD DE IMPORTE A ACLARAR	ACLARAR	N(13)
DESCRIPCION DE LA RECOMENDACION	RECOM	M(10)
FECHA DE IMPLEMENTACION PROGRAMADA	FEC_IMP	F(08)
NUMERO DE IDENTIFICADOR	NO_IDEN	N(02)
BANDERA PARA ENVIO DE INFORMACION	MARCA	C(01)

TABLA: OBTRIM		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
AÑO DE EJERCICIO DE LA OBSERVACION DEL MES	EJER	N(04)
MES DE LA OBSERVACION DEL MES	TRIMES	N(02)
NUMERO DE LA REVISION DEL MES	NO_REV	N(04)
NUMERO DE OBSERVACION DEL MES	NO_OBSER	N(04)
CLAVE DE CRITERIO DEL MES	CRITERIOS	C(04)
DESCRIPCION DE LA OBSERVACION DEL MES	OBSER	M(10)
MONTO DEL IMPORTE DE RIESGO DEL MES	IMP_RIES	N(13)
CANTIDAD DE IMPORTE A ACLARAR DEL MES	ACLARAR	N(13)
DESCRIPCION DE LA RECOMENDACION DEL MES	RECOM	M(10)
FECHA DE IMPLEMENTACION PROGRAMADA DEL MES	FEC_IMP	F(08)
NUMERO DE IDENTIFICADOR DEL MES	NO_IDEN	N(02)
BANDERA PARA ENVIO DE INFORMACION DEL MES	MARCA	C(01)

TABLA: PACA		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
AÑO DE EJERCICIO	EJERC	N(04)
TRIMESTRE DE LA REVISION	TRIMES	N(01)
NUMERO DE LA REVISION	N_REV	N(04)
NUMERO DE CLAVE GENERICA	CLV_G	N(04)
SEMANA INICIAL	S_INI	N(02)
TOTAL DE SEMANAS DE LA REVISION	TOT_SEM_R	N(02)
TOTAL DE SEMANAS HOMBRE	TOT_SEM_H	N(04)
DESCRIPCION DE LA REVICION	DESC_REV	M(10)
DESCRIPCION DE LOS OBJETIVOS	OBJETIVOS	M(10)
DESCRIPCION DE LA CLAVE ESPECIFICA	CLAVE	N(02)
NUMERO DE CLAVE ESPECIFICA	CLV_P	N(04)

TABLA: PACA_PROG		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
NUMERO DE CLAVE DE PROGRAMACION	CLV_P	N(04)
DESCRIPCION DE LA CLAVE	DES_P	C(40)

TABLA: PERIODO		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
AÑO DE EJERCICIO DEL SIP PARA CAPTURA	PEJER	N(04)
MES DEL SIP PARA CAPTURA	PTRIM	N(02)

TABLA: PROGRAMA		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
NUMERO DE LA CLAVE DEL PROGRAMA	CLV_P	N(04)
DESCRIPCION DE LA CLAVE DEL PROGRAMA	DES_P	C(40)

TABLA: REC_PER		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
AÑO DEL EJERCICIO PARA RECUPERACIONES Y COSTOS PARA CAPTURA	PEJER	N(04)
TRIMESTRE PARA RECUPERACIONES Y COSTOS PARA CAPTURA	PTRIM	N(02)

TABLA: RECUP		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
AÑO DE EJERCICIO PARA RECUPERACIONES	EJER	N(04)
TRIMESTRE PARA RECUPERACIONES	TRIMES	N(02)
CANTIDAD TOTAL PARA RECUPERACIONES	TOTAL	N(13.2)
MONTO DE FIANZA PARA RECUPERACIONES	FIANZ	N(13.2)
MONTO DEL GASTO PARA RECUPERACIONES	GASTO	N(13.2)
MONTO DE CUOTAS PARA RECUPERACIONES	CUOTAS	N(13.2)
MONTO DE PAGOS PARA RECUPERACIONES	PAGOS	N(13.2)
MONTO DE IRREGULARIDADES PARA RECUPERACIONES	IRREG	N(13.2)
MONTO FALTANTE PARA RECUPERACIONES	FALTA	N(13.2)
MONTO ANTICIPADO PARA RECUPERACIONES	ANTIC	N(13.2)
MONTO DE RECURSOS NO RECUPERADOS	REINT	N(13.2)
MONTO DE IMPUESTOS PARA RECUPERACIONES	IMPUE	N(13.2)
MONTO DE SANCIONES	SANCI	N(13.2)
MONTO DE OTROS PARA RECUPERACIONES	OTROS	N(13.2)
NOMBRE DEL AREA PARA RECUPERACIONES	AREAS	C(40)
DESCRIPCION DE LAS OBSERVACIONES	NOTAI	M(10)

TABLA: RESPON		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
EJERCICIO CON QUE SE RESPONSABILIZA LA OBSERVACION	EJER	N(04)
NUMERO DE LA REVISION DE LA OBSERVACION	N_REV	N(04)
NUMERO DE LA OBSERVACION	N_OBS	N(04)
NOMBRE DEL RESPONSABLE DE LA OBSERVACION	NOMBRE	C(70)
RFC DEL RESPONSABLE DE LA OBSERVACION	RFC	C(14)

TABLA: REVISION		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
EJERCICIO DE LA REVISION (RECEPCION)	EJERC	N(04)
MES DE LA REVISION (RECEPCION)	TRIMES	N(02)
NUMERO DE LA REVISION (RECEPCION)	N_REV	N(04)
NUMERO DE LA CLAVE PROGRAMADA (RECEPCION)	CLV_P	N(04)
NUMERO DE LA SEMANA INICIAL (RECEPCION)	S_INI	N(02)
TOTAL DE SEMANAS DE LA REVISION (RECEPCION)	TOT_SEM_R	N(02)
TOTAL DE SEMANAS HOMBRE (RECEPCION)	TOT_SEM_H	N(04)
SITUACION TRIMESTRAL (RECEPCION)	SIT_TRI	N(02)
FECHA DE TERMINO DE LA REVISION (RECEPCION)	FEC_TER	F(08)
SEMANAS HOMBRE TOTALES (RECEPCION)	SEM_H_T	N(05)
MARCA DE ENVIO (RECEPCION)	MARCA	N(01)
DESCRIPCION DE LA REVISION (RECEPCION)	DESC_REV	M(10)
DESCRIPCION DE LOS OBJETIVOS (RECEPCION)	OBJETIVOS	M(10)

TABLA: REVTRIM		
DESCRIPCION DE LA COLUMNA	**NOMBRE CAMPO**	**TIPO DE DATO**
EJERCICIO DE LA REVISION (ENVIO)	EJER	N(04)
MES DE LA REVISION (ENVIO)	TRIMES	N(02)
NUMERO DE LA REVISION (ENVIO)	N_REV	N(04)
NUMERO DE LA CLAVE PROGRAMADA (ENVIO)	CLV_P	N(04)
NUMERO DE LA SEMANA INICIAL (ENVIO)	S_INI	N(02)
TOTAL DE SEMANAS DE LA REVISION (ENVIO)	TOT_SEM_R	N(02)
TOTAL DE SEMANAS HOMBRE (ENVIO)	TOT_SEM_H	N(04)
SITUACION TRIMESTRAL (ENVIO)	SIT_TRI	N(02)
FECHA DE TERMINO DE LA REVISION (ENVIO)	FEC_TER	F(08)
SEMANAS HOMBRE TOTALES (ENVIO)	SEM_H_T	N(05)
MARCA DE ENVIO (ENVIO)	MARCA	N(01)
DESCRIPCION DE LA REVISION (ENVIO)	DESC_REV	M(10)
DESCRIPCION DE LOS OBJETIVOS (ENVIO)	OBJETIVOS	M(10)

TABLA: SEGTRIM		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
EJERCICIO DEL SEGUIMIENTO (ENVIO)	EJER_A	N(04)
NUMERO DE L SEGUIMIENTO (ENVIO)	SEGUI	N(05)
NUMERO DE LA REVISION (ENVIO)	NO_REV	N(04)
NUMERO DE LA OBSERVACION (ENVIO)	NO_OBSER	N(04)
FECHA REAL DEL SEGUIMIENTO (ENVIO)	FEC_REAL	F(08)
NUMERO DE LA CARACTERISTICA DEL SEGUIMIENTO (ENVIO)	CARAC_SEG	N(02)
DESCRIPCION DE LOS COMENTARIOS (ENVIO)	COMENTARIO	M(10)
IMPORTE DE RIESGO DEL SEGUIMIENTO (ENVIO)	IMP_RIESGO	N(13)
IMPORTE A ACLARAR DEL SEGUIMIENTO (ENVIO)	IMP_ACLARA	N(13)

TABLA: SEGUI		
DESCRIPCION DE LA COLUMNA	NOMBRE CAMPO	TIPO DE DATO
EJERCICIO DEL SEGUIMIENTO (RECEPCION)	EJER_A	N(04)
NUMERO DE L SEGUIMIENTO (RECEPCION)	SEGUI	N(06)
NUMERO DE LA REVISION (RECEPCION)	NO_REV	N(04)
NUMERO DE LA OBSERVACION (RECEPCION)	NO_OBSER	N(04)
FECHA REAL DEL SEGUIMIENTO (RECEPCION)	FEC_REAL	F(08)
NUMERO DE LA CARACTERISTICA DEL SEGUIMIENTO (RECEPCIÓN)	CARAC_SEG	N(02)
DESCRIPCION DE LOS COMENTARIOS (RECEPCION)	COMENTARIO	M(10)
IMPORTE DE RIESGO DEL SEGUIMIENTO (RECEPCION)	IMP_RIESGO	N(13)
IMPORTE A ACLARAR DEL SEGUIMIENTO (RECEPCION)	IMP_ACLARA	N(13)

Relación de Programas.- Rutinas de la programación.

PROGRAMA	DESCRIPCION
ACERCA	NOMBRE, NUMERO DE LA VERSION Y FECHA. (VERSION DEL SISTEMA)
ADVERTEN	GENERA UNA VENTANA DE ADVERTENCIA DE RECUPERACION DE RESPALDO
CAPREPACI	MENU DE REPORTES DEL SIP (AUDITORIAS: INTEGRALES, ESPECIFICAS, DE PROGRAMA, DE SEGUIMIENTO, DE DESEMPEÑO Y TOTALES)
DATENT	CAPTURA DE INFORMACION DE DATOS DEL TITULAR DE LA ENTIDAD, ASI COMO DEL TITULAR DEL OIC
ENVCOR	GENERA CONTROL DE ENVIO DE CORREOS DEL OIC A UESGP
EXPORTA	GENERA PANTALLA DE GENERACION DE ARCHIVOS DE ENVIO DEL PACA, INFORMACION PERIODICA, Y RECUPERACIONES Y COSTOS
G9	GENERA REPORTE DE OBSERVACIONES RELEVANTES PENDIENTES DE ATENDER NUEVO REPORTE
GLOB	AYUDA DEL SISTEMA
IMPRIME	GENERA PANTALLA DE OPCION DE IMPRIMIR (VISTA PREVIA O A IMPRESORA)

INTRO	GENERA PANTALLA PRINCIPAL CONTENIENDO LAS OPCIONES PRINCIPALES DEL MENU (SALIR, PACA, SIP, RECUPERACIONES, DATOS DE ENTIDAD, CORREO, UTILERIAS, VERSION)
BOXER	GENERA REPORTE DE OBSERVACIONES CON SU INFORMACION RELEVANTE
PACA	PANTALLA PRINCIPAL DE CAPTURA DEL PACA
PACADET	GENERA REPORTE DE PROGRAMA ANUAL DE AUDITORIA DETALLADO, CONTENIENDO LA DESCRIPCION DE LA OBSERVACION Y LOS OBJETIVOS ESTABLECIDOS
PACARE	GENERA EL REPORTE ANUAL DE AUDITORIA DETALLADO
PARENV	GENERA PANTALLA DE CONTROL DE ENVIO DE BASES A EQUIPO PARCIAL (SIN USO)
PASSWD1	GENERA PANTALLA PARA INTRODUCIR LA CLAVE DE USUARIO (SIN USO)
PRSIP	PANTALLA PRINCIPAL DE CAPTURA DEL SISTEMA DE INFORMACION PERIODICA
RECEPCION	GENERA PANTALLA DE CONTROL DE RECEPCION DE PAQUETES DE INFORMACION, DESENCRIPTANDO LA INFORMACION A LAS BASES CORRESPONDIENTES DE CADA RECEPCION
RECREP	GENERA REPORTE DE INFORMACION ESTADISTICA DE RECUPERACIONES DEL ORGANO INTERNO DE CONTROL
RECUPE	GENERA PANTALLA PRINCIPAL DE CAPTURA DE LAS RECUPERACIONES Y COSTOS DEL ORGANO INTERNO DE CONTROL
REINDEX	MODULO PARA REINDEXAR LAS BASES DE DATOS DEL SISTEMA
REPCOSTO	GENERA REPORTE DE INFORMACION ESTADISTICA DE LOS COSTOS DEL ORGANO INTERNO DE CONTROL
REPNOTAS	GENERA REPORTE DE LAS NOTAS ACLARATORIAS (USE-R-01) CAPTURADAS DE LAS RECUPERACIONES DEL ORGANO INTERNO DE CONTROL
REPSIP	GENERA PANTALLA PARA SELECCIONAR EJERCICIO, REVISION Y OBSERVACION DE LA INFORMACION PERIODICA
UNIT1	GENERA EL REPORTE DE MEDIDAS CORRECTIVAS DE LA INFORMACION PERIODICA

Referencias Cruzadas Programa – Tablas .- Diagrama que representa las tablas que se consultan en cada rutina del sistema.

	TABLAS																		
PROGRAMAS	AYUCLV	ACUS	COSTO	EJERCIO	INBAS	MONTOS	OBTRIM	OBSERVA	PACA	PERIODO	PROGRAMA	PACA_PROG	REVTRIM	REVISION	REC_PER	RESPON	RECUP	SEGUI	SEGTRIM
ACERCA																			
ADVERTEN																			
CAPREPACI																			
DATENT					*														
ENVCOR					*														
EXPORTA	*		*		*		*	*	*	*			*	*	*		*	*	*
G9								*										*	
GLOB																			
IMPRIME																			
INTRO																			
OBSER							*	*					*	*					
PACA					*	*			*										
PACADET									*				*						
PACARE						*					*								
PARENV	*				*														
PASSWD1																			
PRSIP	*				*		*	*			*		*	*		*		*	*
RECEPCION							*		*				*	*		*	*		*
RECREP					*												*		
RECUPE			*		*												*		
REINDEX			*			*	*	*	*				*	*		*	*	*	*
REPCOSTO			*		*														
REPNOTAS																			
REPSIP																			
UNIT1							*	*					*	*				*	*

Hasta aquí llegaremos con la explicación del desarrollo del módulo del informante o emisor, no sin antes recalcar que la explicación detallada de los esquemas realizados dentro de este diseño así como la programación e incluso parte de la definición del software empleado no se muestran por políticas del área de trabajo.

A continuación describiré el otro módulo del sistema que llega a cerrar la aplicación completa, el módulo central encargado de la recepción, consulta, análisis y evaluación de la información enviada por las contralorías internas.

CAPITULO III. DISEÑO DE LA BASE DE DATOS PARA EL MODULO CENTRAL.

Con el presente capítulo, cerraremos la descripción del diseño de nuestro sistema completo. el módulo central se encargará de operaciones típicas de un concentrador de información, funciones como la recepción. análisis y evaluación de la información deben ser fundamentales en el diseño de este módulo.

Análisis de Requerimientos.

Objetivos Principal :

Crear un módulo que reciba y ordene la información capturada por las contralorías internas.
Generar informes estadísticos y evaluaciones periódicas sobre la actuación de las contralorías internas.

Objetivos específicos:

- Obtener la información actualizada por trimestre.
- Elaboración de reportes precisos y detallados.
- Detección de desviaciones en el aprovechamiento de recursos de los órganos internos de control.
- Detección de irregularidades en las dependencias y entidades a través de los informes que presentan los órganos internos.
- Continuar con los procesos de seguridad empleados en el módulo del informante.

Identificación de actores:

La identificación de los actores se vio en el capítulo anterior; sobre este módulo lo único que cambiamos es tipo de usuario que se le da a cada uno de ellos.

Contraloría Interna.- que en el anterior módulo era un actor principal, en este módulo pasa a ser un actor secundario debido a que solo interviene en el envió de la información sin involucrarse en los procesos dentro de la USEGP.

Dirección General Adjunta de Control y Seguimiento (DGACS).- Actor principal dentro de este módulo encargado de la recepción, generación de reportes, del análisis y la evaluación de las contralorías.

Dirección de Administración de Bases de Datos.- encargada de dar soporte técnico. el contacto con el usuario principal (DGACS) es más directo previéndole de asesorías y modificaciones al módulo con mayor rapidez.

Casos de Uso.

Las operaciones de este módulo pueden resumirse de la siguiente forma:

Primero recibe la información generada por el informante, esta recepción no es automática, es decir no se tiene un proceso permanente para recibir e incorporar la información, el porque de esto es muy sencillo, por que solo el analista de la información sabe cuando esta en posibilidades de introducir la información de un trimestre al sistema, una vez que ya tenga lista la evaluación del anterior periodo.

Una vez incorporada la información, se procede a analizar los nuevos datos del sistema, para ello este módulo cuenta con la generación de reportes especificos que muestran al usuario la nueva información.

Con la visión tanto global como detallada de la información se procede a su análisis, y con ello a la modificación de los datos de acuerdo a la consideración del analista, en esta etapa es muy común la conciliación con la contraloria interna.
Si el personal lo considera necesario, se le indica a la contraloria que integre nuevamente la información de forma más convincente, y para ello se le proporciona una clave que le ayude al módulo del informante a regresarse un periodo atrás con el fin de realizar dicha modificación.

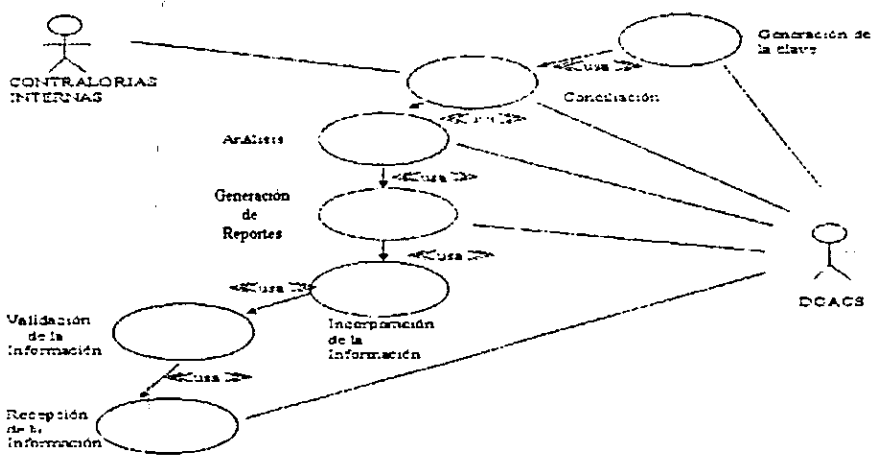


Figura 3.1.

Una vez terminado el análisis, la DGACS se dedica a generar evaluaciones periódicas, reportes diseñados previamente, donde se visualiza el comportamiento que se sigue en la contraloria sobre los asuntos pendientes.

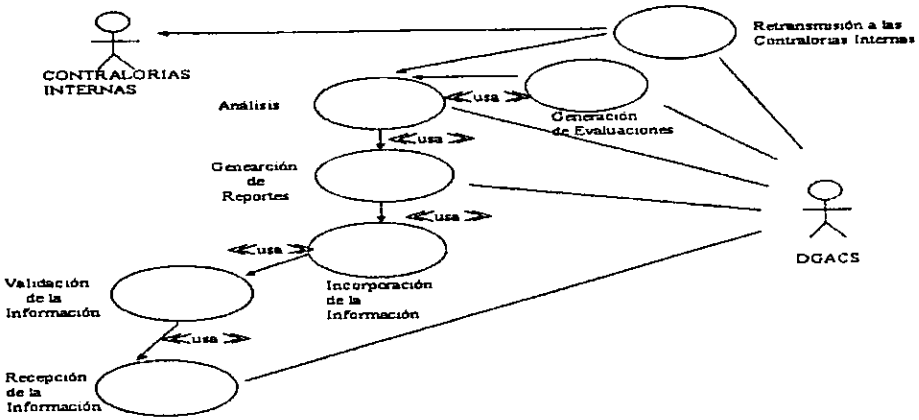


Figura 3.2.

Por último y si es necesario, la DGACS retroalimenta a las contralorías internas de la información evaluada y en su caso modificada para continuar con el siguiente periodo teniendo exactamente la misma información tanto en el módulo central como en el del informante.

Realmente estos dos últimos esquemas de caso de uso son los más representativos en el módulo central, e indican de forma precisa el flujo de los datos a través de un proceso definido.

Diagrama de Clases.

Tanto el módulo del informante como en el central trabajan las mismas clases dentro de la información, con la gran diferencia de que en el central se debe identificar un atributo que clasifica la información dentro de un universo de entidades, tal y como lo muestra la siguiente figura.

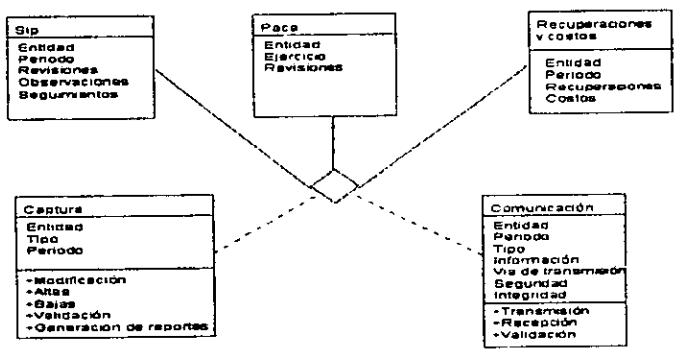


Figura 3.3.

Dentro de la estructura de nuestros datos también podriamos incluir el atributo de entidad (figura 3.4):

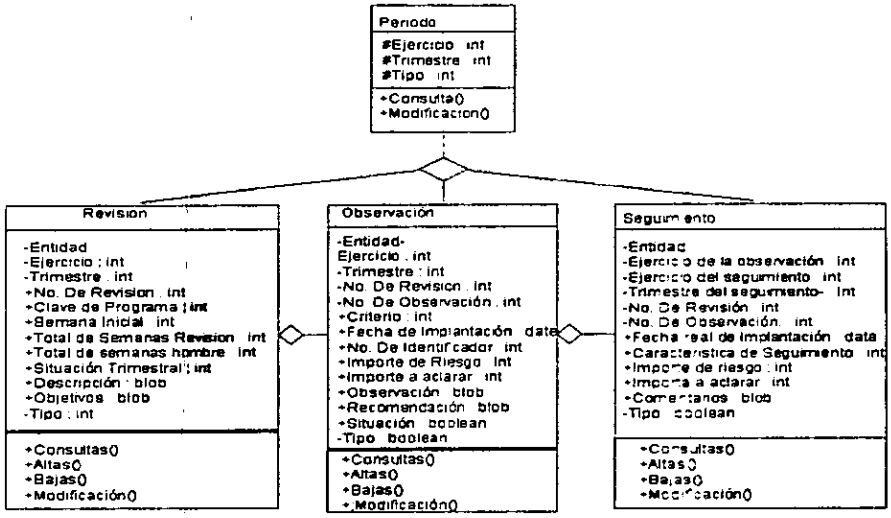


Figura 3.4.

Podríamos repetir en este apartado todos los diagramas de clase incluidos en el capitulo anterior con sus respectivas descripciones, únicamente agregando a cada una de las clases el atributo de entidad. sin embargo no lo haremos ya que este documento trata de ser representativo de la metodologia empleada en el desarrollo del sistema y no una documentación completa del mismo.

Diagrama de Secuencias.

Los diagrama de secuencia son similares a los del informante con algunas variantes:

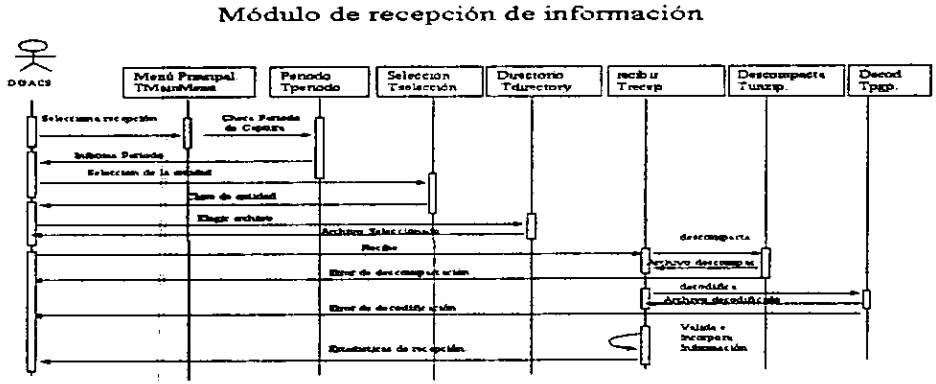


Figura 3.3.

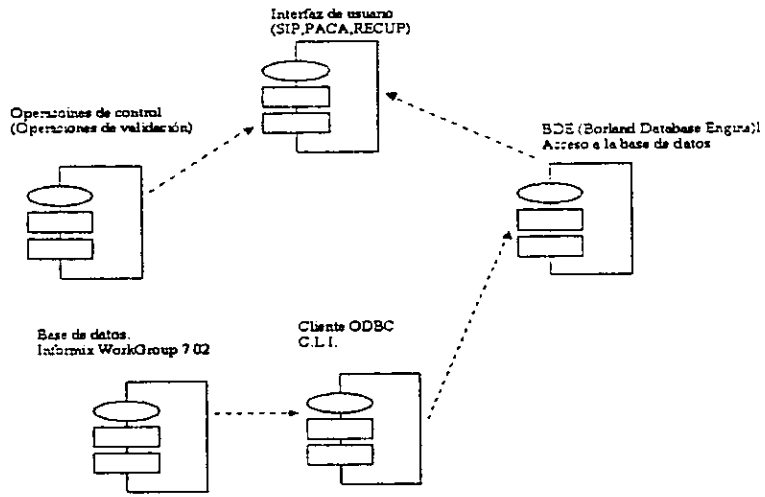
Realmente este es el diagrama de secuencia más importante dentro del módulo central. Aunque muy sencillo en su aspecto, encierra procesos y rutinas de programación muy importantes para la integridad de la información, y sobre todo para que éste pueda cumplir con sus objetivos primarios. Es decir que dado que este módulo debe ser una herramienta de análisis de la información recibida que sirve incluso para realizar una evaluación del trabajo de las contralorías, es mucho muy importante que los datos recibidos coincidan con los que la contraloría envía, y a su vez que los datos no tengan ningún tipo de error lógico dentro de los estados posibles de la información.

Los demás diagramas de secuencia los podemos obtener variando los ya explicados en el capitulo anterior, únicamente incorporando una previa selección de la entidad para las altas, modificaciones o bajas tanto de las revisiones, observaciones y seguimientos.

Cabe mencionar que en este módulo no se incorporó ninguna opción con rutinas de mantenimiento para las tablas que contienen la información. Esto es por que la administración de las mismas esta a cargo de la Dirección de Administración de Base de Datos, y todas las rutinas de respaldo o corrección de tablas están dentro de las actividades de la misma.

Diseño.

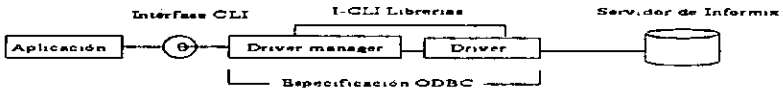
El diagrama de componentes.- para éste módulo es:



De este diagrama nos enfocaremos a los tres últimos componentes que son el DBE de Borland que es el intermediario entre la interfase de usuario y nuestros datos , el informix-CLI que es el software cliente para nuestra base de datos y el mismo servidor de base de datos Informix Workgroup.

De los anteriores DBE (Borland Database Engine), ya se contempló dentro de la explicación del diagrama de componentes del informante en el capitulo anterior, y en este módulo solo agregaremos que la conexión con el motor de base de datos ya no es directo sino a través del ODBC que es una interfase que permite a las aplicaciones el acceso a base de datos que utilizan el Structured Query Languaje (SQL).

El Informix-CLI no es otra cosa que una interfase basada en ODBC de Microsoft para acceder información en un motor de informix. Su estructura es la siguiente.



Aún que todo esto es transparente para nosotros. el driver manager y el driver se ubican en una aplicación como una unidad que contiene llamadas a funciones de acceso a la base de datos. Obviamente el motor de informix corresponde a la fuente de los datos (DBMS). El DBE vuelve esto mucho más transparente. ya que encierra las llamadas a esas funciones dentro de su estructura y manejo de Alias.

El motor de base de datos es el informix Online Workgroup Server Ver 7.31 para Windows NT que es una base de datos relacional. Un servidor de base de datos es un paquete que maneja los accesos a una o más bases de datos para una o más aplicaciones clientes. En especial. el Informix Online es un RDBMS es decir es relacional donde los datos se organizan en tablas que contiene renglones y columnas.

Las características principales de nuestro servidor informix son:

Una arquitectura cliente / servidor.- el servidor procesa requerimientos de datos de aplicaciones cliente. y envía resultados. para ello maneja actividades como la coordinación de requerimientos concurrentes de multiples clientes. el performance de las operaciones de escritura y lectura. y operaciones de consistencia tanto física como lógica de la información.

Escalabilidad DSA (Dynamic Scalable Architecture).-. permite adicionar tanto memoria compartida como hilos de procesos aún cuando el servidor este en linea.

Alto performance. El performance lo caracteriza los siguientes puntos:

1.- Un manejador dinámico de memoria compartida.- todas las aplicaciones usan una instancia de la información compartida dentro de un espacio de memoria en el servidor . Después de que la aplicación lee los datos de la tabla. otras aplicaciones pueden acceder los datos que se encuentran cargados en memoria. Con esto los accesos al disco y con ello la

degradación del performance no ocurre. Además de los datos requeridos en memoria, también se guarda toda la información de control.

2.- Hilos dinámicos y paralelismo.- El servidor utiliza lo que se llama procesos virtuales, que son hilos de Windows NT para servir a múltiples clientes, y en su caso, para un solo cliente se pueden utilizar múltiples hilos de informix, cada uno para un determinado Query.

Tolerancia a fallas.-para proteger la integridad de la información y su consistencia, el servidor provee de las siguientes características:

1.- Respaldos de los Dbspace y logical-log de los registros de transacciones.-el servidor te da la habilidad de respaldar los datos manipulados asi como los cambios del servidor de la base de datos en archivos llamados logical-log ; esta habilidad conlleva a la incrementabilidad en los respaldos generados. Es decir que los respaldos de logical-log permiten, en dado caso realizar restauraciones de información tan solo desde el último movimiento realizado .

2.- Restauración rápida.-esta caracteristica se nota al inicializar el servidor después de una falla. Lo que realiza en servidor es un chequeo de los archivos fisicos y lógicos del sistema, y con ello la recuperación del mismo.

3.- Espejos.- se le llama espejo a un disco en paralelo con la misma información que se trabaja en el servidor en producción. En este caso, el servidor es el encargado de tener actualizado este disco. La utilidad del espejo es obvia, elimina la posibilidad de la pérdida total de la información dentro de nuestra base de datos.

4.- Replica de datos.-además de un disco espejo donde se contiene la misma información que en el de producción, si el sistema en cuestión requiere de un alto grado de disponibilidad, es decir que por ninguna circunstancia debe dejar de trabajar, entonces se tiene la posibilidad de trabajar réplicas del servidor. En las réplicas además de la información de las bases de datos, se tienen trabajando el servidor de bases de datos. Cuando uno de nuestros servidores falla inmediatamente las aplicaciones son redireccionadas a una réplica sin afectar a los usuarios.

Soporta multimedia.- el servidor puede tener dos tipos de campos blob (Binary large Object), el TEXT y el BYTE, que prácticamente no limitan el tamaño del objeto guardado. El servidor ocupa los llamados Blobspace para almacenar este tipo de datos, que son espacios designados del disco o los discos duros, que incluso pueden irse incrementando de acuerdo a las necesidades de la base de datos.

Query's sobre datos distribuidos.-la habilidad de seleccionar o modificar query's sobre diferentes servidores de base de datos, es decir la capacidad de conectarse a múltiples servidores a través de un protocolo de dos fases (Two-Phase Commit)

Seguridad como servidor de base de datos. La primer caracteristica de seguridad en el servidor se da a nivel de privilegios en las tablas o sobre la base de datos con los comandos GRANT y REVOKE del SQL. En adición existen procesos básicos de auditoria de los eventos de una base de datos, meramente para mencionar un comando común el ONSTAT con sus distintos parámetros.

Dadas las caracteristicas explicadas del servidor de informix, solo mencionaremos que en nuestro sistema solo se realizan o se ocupan algunas de estas, dejando fuera las que por falta de presupuesto no son aplicables como las replicas de

servidor. los discos espejos, los querys distribuidos entre otros. Sin embargo se mantienen políticas de respaldos incrementales y a nivel cero periódicamente. así como el monitoreo constante de los dbspace y los blobspace.

Dado el caso de que los objetos de este módulo coinciden con los del módulo del informante, evitamos colocar los mismos diagramas de estado para esta sección, solo indicamos que cada objeto analizado en el anterior capitulo es idéntico en sus estados que en este módulo.

Implementación y Pruebas.

En general la documentación realizada para el informante sigue en mismo formato que para el módulo central. a continuación se presentan los diagramas realizados.

Relación de Tablas.

TABLA	DESCRIPCION
costo	COSTOS DEL O.I.C.
datenti	DATOS DE LA ENTIDAD
entcosto	COSTOS DEL O.I.C. SUBDIVIDIDOS POR AREA (ORIGINALES)
entidad	CATALOGO DE ENTIDADES
entrecup	RECUPERACIONES SUBDIVIDIDAS POR AREA (ORIGINALES)
evaltex	TEXTOS DE LA EVALUACION TRIMESTRAL
observacion	OBSERVACIONES RELEVANTES
paca	PROGRAMA ANUAL DE CONTROL Y AUDITORIA
passwd	CONTROL DE ACCESO AL MODULO
programa97	CLAVES DE PROGRAMA
recup	RECUPERACIONES
registro	OBSERVACIONES QUE EJEMPLIFICARAN EN LA EVALUACION
revision	REVISIONES PROGRAMADAS Y ADICIONALES
rfc	RFC DE LAS OBSERVACIONES
sector	CATALOGO DE SECTORES
seguimiento	SEGUIMIENTO A LAS OBSERVACIONES

Relación de columnas.

TABLA : ENTIDAD

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
CLAVE DE LA ENTIDAD	clv_ent	integer
DESCRIPCION 1 DE LA ENTIDAD	desc1	char(70)
DESCRIPCION 2 DE LA ENTIDAD	desc2	char(70)
DESCRIPCION 3 DE LA ENTIDAD	desc3	char(70)
CLAVE DE SECTOR	sector	integer
NOMBRE CORTO	nom_cor	char(15)

DIRECCION RESPONSABLE	dir_res	integer
SUBDIRECCION RESPONSABLE	sub_res	integer
JEFE DE DEPARTAMENTO RESPONSABLE	jef_res	Integer
CLAVE DEL COMISARIO	clave	integer
SI PERTENECE LA ENTIDAD AL SIP (S/N)	sip	char(1)
SI PERTENECE LA ENTIDAD AL INTRAGUB (S/N)	intragub	char(1)
ENTIDAD DE CUENTA PUBLICA	cp	char(1)
TIPO DE ENTIDAD	tp	char(1)

TABLA : RFC

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO	ejer	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
NUMERO DE REVISION	n_rev	integer
NUMERO DE OBSERVACION	n_obs	integer
REGISTRO FEDERAL DE CAUSANTES	rfc	char(13)
NOMBRES	nombres	char(20)
APELLIDO PATERNO	apate	char(20)
APELLIDO MATERNO	amate	char(20)

TABLA : ENTIDAD

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
CLAVE DE LA ENTIDAD	clv_ent	integer
DESCRIPCION 1 DE LA ENTIDAD	desc1	char(50)
DESCRIPCION 2 DE LA ENTIDAD	desc2	char(50)
NOMBRE DEL DIRECTOR	dnombre	char(45)
RFC DEL DIRECTOR	drfc	char(13)
PUESTO DEL DIRECTOR	dpuesto	char(45)
CALLE	dcalle	char(45)
COLONIA	dcolonia	char(25)
CIUDAD	dciudad	char(20)
CODIGO POSTAL	dcp	integer
ESTADO	dedo	char(20)
LADA 1	dlada1	integer
TELEFONO 1	dtel1	integer
ESXTENSION 1	dext1	integer
LADA 2	dlada2	integer
TELEFONO 2	dtel2	integer
EXTENSION 2	dext2	integer
COMBRE DEL CONTRALOR INTERNO	cnombre	char(45)
RFC DEL CONTRALOR INTERNO	crfc	char(13)
PUESTO DEL CONTRALOR INTERNO	cpuesto	char(45)
CALLE	ccalle	char(45)
COLONIA	ccolonia	char(25)
CIUDAD	cciudad	char(20)
CODIGO POSTAL	ccp	integer

ESTADO	cedo	char(20)
LADA 1	clada1	integer
TELEFONO 1	ctel1	integer
EXTENSION 1	cext1	integer
LADA 2	clada2	integer
TELEFONO 2	ctel2	integer
EXTENSION 2	cext2	integer
CLAVE PGP	clavepgp	char(20)
CORREO ELECTRONICO	d_email	char(50)

TABLA : REGISTRO

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
NUMERO CONSECUTIVO	numero	integer
CLAVE DE LA ENTIDAD	entidad	integer
EJERCICIO DE LA EVALUACION	ejer	integer
TRIMESTRE DE LA EVALUACION	trimestre	integer
EJERCICIO DE LA OBSERVACION	ejercicio	integer
NUMERO DE REVISION	revision	integer
NUMERO DE OBSERVACION	observacion	integer

TABLA : SECTOR

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
CLAVE DEL SECTOR	clv_sec	integer
DESCRIPCION DEL SECTOR	des_sec	char(50)

TABLA : PACA

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO	ejer	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
FECHA DE RECEPCION	fec_rec	date
FECHA DE ANALISIS	fec_ana	date
FECHA DE APROBACION	fec_apr	date
FECHA DE CAPTURA	fec_cap	date
REVISIONES PROGRAMADAS	rev_pro	integer
REVISIONES ADICIONALES	rev_adi	integer
FUERZA DE TRABAJO CONTROL Y AUDITORIA	fzt_cya	integer
TOTAL DE LA FUERZA DE TRABAJO	fzt_tot	integer
TIPO DE ENTIDAD	tip_ent	integer
INDICADOR DE PACA	ind_paca	integer
SUBDIRECCION RESPONSABLE	sub_res	integer
SITUACION DE LA ENTIDAD	sit_ent	integer
SITUACION RELEVANTE	sit_rel	integer
CONTROL Y AUDITORIA	control	integer

RESECTORIZADO	sec_nue	integer
FUNCIONES DE QUEJAS Y DENUNCIAS	fun	integer
CAPACITACION, JUNTAS CONSEJOS COMITES	cap	integer
VACACIONES	vac	integer
ACTIVIDADES QUE NO REUNEN REQUISITOS DE REVISION	act_sin	integer
ACTIVIDADES ADMINISTRATIVAS	act	integer

TABLA : PROGRAMA97

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
CLAVE DEL PROGRAMA	clv_pro	integer
DESCRIPCION 1 DEL PROGRAMA	desc1_pro	char(50)
DESCRIPCION 2 DEL PROGRAMA	desc2_pro	char(50)
ENFOQUE 1	enfoque1	integer
ENFOQUE 2	enfoque2	integer

TABLA : EVALTEX

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO DE LA EVALUACION	ejer	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
TRIMESTRE DE LA EVALUACION	trimes	integer
TEXTO DEL CUMPLIMIENTO	cumplimiento	text
TEXTO DE LA PROBLEMÁTICA	problematica	text
TEXTO DE LAS OBSERVACIONES	observaciones	text
TEXTO DEL CUADRO	cuadro	text

TABLA : REVISION

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO	ejer	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
NUMERO DE REVISION	n_rev	integer
CLAVE DEL PROGRAMA	clv_pro	integer
ENFOQUE	enfoque	integer
TIPO	tipo	char(1)
NUMERO DE SEMANAS DE LA REVISION	semanas	integer
TRIMESTRE PROGRAMADO	trim	integer
TOTAL DE SEMANAS	t_sem	integer
SEMANAS UTILIZADAS EN EL MES 1	sh1	integer
SEMANAS UTILIZADAS EN EL MES 2	sh2	integer
SEMANAS UTILIZADAS EN EL MES 3	sh3	integer
SEMANAS UTILIZADAS EN EL MES 4	sh4	integer
SITUACION EN EL MES 1	st1	integer
SITUACION EN EL MES 2	st2	integer

SITUACION EN EL MES 3	st3	integer
SITUACION EN EL MES 4	st4	integer
SEMANA INICIAL	s_ini	integer
FECHA DE TERMINACION DE LA REVISION	fec_ter	date
DESCRIPCION DE LA REVISIÓN	descripcion	text
OBJETIVO DE LA REVISIÓN	objetivo	text
SEMANAS UTILIZADAS EN EL MES 5	Sh5	integer
SEMANAS UTILIZADAS EN EL MES 6	Sh6	integer
SEMANAS UTILIZADAS EN EL MES 7	Sh7	integer
SEMANAS UTILIZADAS EN EL MES 8	Sh8	integer
SEMANAS UTILIZADAS EN EL MES 9	Sh9	integer
SEMANAS UTILIZADAS EN EL MES 10	Sh10	integer
SEMANAS UTILIZADAS EN EL MES 11	Sh11	integer
SEMANAS UTILIZADAS EN EL MES 12	Sh12	integer
SITUACION EN EL MES 5	St5	integer
SITUACION EN EL MES 6	St6	integer
SITUACION EL EL MES 7	St7	integer
SITUACION EN EL MES 8	St8	integer
SITUACION EN EL MES 9	St9	integer
SITUACION EN EL MES 10	St10	integer
SITUACION EL EL MES 11	St11	integer
SITUACION EN EL MES 12	St12	integer

TABLA : COSTO

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO	ejer	integer
TRIMESTRE	trimes	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
NUMERO DE REVISION	no_rev	integer
COSTO	costo	float
TEXTO DE LAS NOTAS	notas	text

TABLA : PASSWD

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
LOGIN DEL USUARIO	login	char(12)
CLAVE DEL USUARIO	passwd	char(8)
TIPO DE USUARIO	tipo	integer
DIRECTOR RESPONSABLE	dir_res	integer
SUBDIRECTOR RESPONSABLE	sub_res	integer
JEFE DE DEPARTAMENTO RESPONSABLE	jef_res	integer
NOBMBRE DEL USUARIO	asignada	char(40)
NUMERO DE USUARIO	num_usu	integer
CUENTA DE CORREO ELECTRONICO	correo	char(40)
TELEFONO O EXTENSION	telext	char(20)

TABLA : RECUP

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO DE CAPTURA	ejer	integer
TRIMESTRE DE CAPTURA	trimes	integer
CLAVE DE LA ENTIDAD.	clv_ent	integer
FIANZAS,ADEUDOS Y SEGUROS	fianz	float
GASTOS NO COMPROBADOS	gasto	float
CUOTAS DE RECUPERACION Y DE APORTACIONES	cuotas	float
PAGOS EN EXCESO	pagos	float
IRREGULARIDADES EN SERVICIOS FINANCIEROS	irreg	float
FALTANTES	falta	float
ANTICIPOS DE CONTRATOS RESCINDIDOS OBRA PAGADA NO EJECUTADA.	antic	float
REINTEGROS A TESOFE DE RECURSOS NO EJERCIDOS O RECUP.	reint	float
IMPUESTOS MAL CALCULADOS Y/O NO ENTERADOS	impue	float
SANCIONES ECONOMICAS, MULTAS Y PLIEGOS DE RESPONSABILIDADES	sanci	float
OTROS	otros	float
RESPONSABLE DE LA CAPTURA DEL FORMATO	responsable	char(100)
TELEFONO DEL RESPONSABLE	telefono	char(20)
NOTAS ACLARATORIAS	nota	text

TABLA : ENTCOSTO

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO DE CAPTURA	ejer	integer
TRIMESTRE DE CAPTURA	trimes	integer
CLAVE DE LA ENTIDAD.	clv_ent	integer
AREAS DONDE SE LOCALIZÓ EL COSTO	areas	char(40)
NUMERO DE REVISIONES	no_revs	integer
COSTO POR AREA	costo	float
MARCA DE MODIFICACIÓN .	marca	char(1)

TABLA : ENTRECUP

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
Ejercicio de captura	ejer	integer
Trimestre de captura	trimes	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
AREAS DE RECUPERACIÓN	areas	char(100)
TOTAL POR AREA	total	float
FIANZAS,ADEUDOS Y SEGUROS	fianz	float
GASTOS NO COMPROBADOS	gasto	float

CUOTAS DE RECUPERACION Y DE APORTACIONES	cuotas	float
PAGOS EN EXCESO	pagos	float
IRREGULARIDADES EN SERVICIOS FINANCIEROS	irreg	float
FALTANTES	falta	float
ANTICIPOS DE CONTRATOS RESCINDIDOS OBRA PAGADA NO EJECUTADA.	antic	float
REINTEGROS A TESOFE DE RECURSOS NO EJERCIDOS O RECUP.	reint	float
IMPUESTOS MAL CALCULADOS Y/O NO ENTERADOS	impue	float
SANCIONES ECONÓMICAS, MULTAS Y PLIEGOS DE RESPONSABILIDADES	sanci	float
OTROS	otros	float
NOTAS ACLARATORIAS	nota1	text

TABLA : OBSERVACION

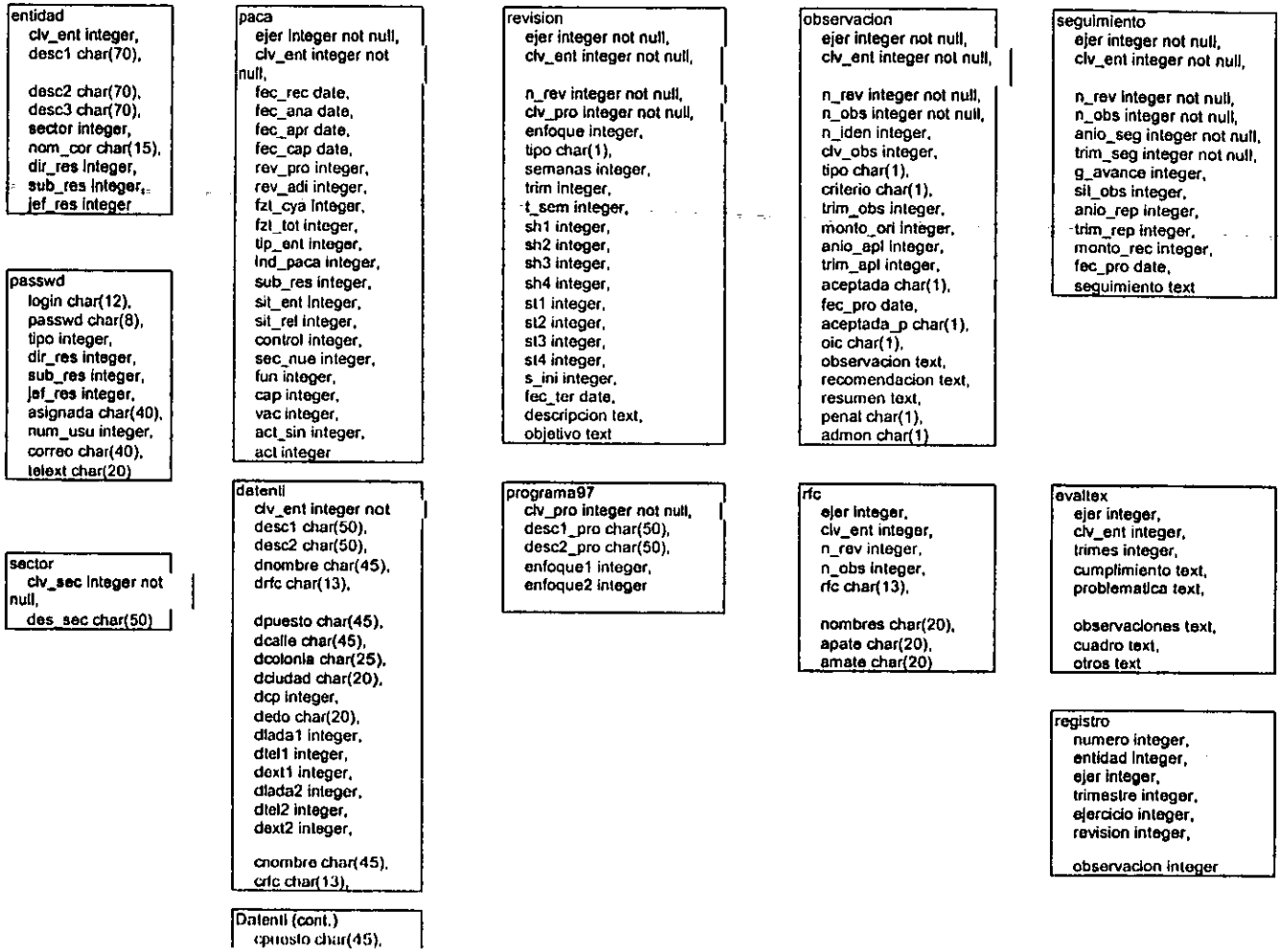
DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO	ejer	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
NUMERO DE REVISION	n_rev	integer
NUMERO DE OBSERVACION	n_obs	integer
CLAVE DE INSTANCIA QUE GENERO LA OBSERVACION	n_iden	integer
CLAVE DE LA OBSERVACION	clv_obs	integer
TIPO	tipo	char(1)
CRITERIO	criterio	char(1)
TRIMESTRE DE LA OBSERVACION	trim_obs	integer
MONTO ORIGINAL	monto_ori	integer
AÑO DE APLICACIÓN	anio_apl	integer
TRIMESTRE DE APLICACIÓN	trim_apl	integer
ACEPTADA POR CONTADURIA MAYOR DE HACIENDA	aceptada	char(1)
FECHA PROPUESTA	fec_pro	date
ACEPTADA CON PRESUNTA RESPONSABILIDAD	aceptada_p	char(1)
OBSERVACION AL OIC	oic	char(1)
TEXTO DE LA OBSERVACION	observacion	text
TEXTO DE LA RECOMENDACIÓN	recomendacion	text
TEXTO DEL RESUMEN DE LA OBSERVACION	resumen	text
TIPO DE RESPONSABILIDAD PENAL	penal	char(1)
TIPO DE RESPONSABILIDAD ADMINISTRATIVA	admon	char(1)

TABLA : SEGUIMIENTO

DESCRIPCION DE LA COLUMNA	NOMBRE DE LA COLUMNA	TIPO DE DATO DE LA COLUMNA
EJERCICIO	ejer	integer
CLAVE DE LA ENTIDAD	clv_ent	integer
NUMERO DE REVISION	n_rev	integer
NUMERO DE OBSERVACION	n_obs	integer

AÑO DEL SEGUIMIENTO	anio_seg	integer
TRIMESTRE DEL SEGUIMIENTO	trim_seg	integer
GRADO DE AVANCE	g_avance	integer
CLAVE DE LA SITUACION DE LA OBSERVACION	sit_obs	integer
AÑO DE REPROGRAMACION	anio_rep	integer
TRIMESTRE DE REPROGRAMACION	trim_rep	integer
MONTO RECUPERADO	monto_rec	integer
FECHA	fec_pro	date
TEXTO DEL SEGUIMIENTO	seguimiento	text

El diagrama siguiente representa la relación de tablas dentro de la base de datos, donde se nota que dentro de nuestras llaves principales siempre existe un campo destinado a la entidad.


entidad
clv_ent integer,
desc1 char(70),
desc2 char(70),
desc3 char(70),
sector integer,
nom_cor char(15),
dir_res integer,
sub_res integer,
jef_res integer
passwd
login char(12),
passwd char(8),
tipo integer,
dir_res integer,
sub_res integer,
jef_res integer,
asignada char(40),
num_usu integer,
correo char(40),
telext char(20)
sector
clv_sec integer not null,
des_sec char(50)
paca
ejer integer not null,
clv_ent integer not null,
fec_rec date,
fec_ana date,
fec_apr date,
fec_cap date,
rev_pro integer,
rev_adi integer,
fzt_cya integer,
fzt_tot integer,
tip_ent integer,
ind_paca integer,
sub_res integer,
sit_ent integer,
sit_rel integer,
control integer,
sec_nue integer,
fun integer,
cap integer,
vac integer,
act_sin integer,
act integer
datenti
clv_ent integer not
desc1 char(50),
desc2 char(50),
dnombre char(45),
drfc char(13),
dpuesto char(45),
dcalle char(45),
dcolonia char(25),
dciudad char(20),
dcp integer,
dedo char(20),
dlada1 integer,
dtel1 integer,
dext1 integer,
dlada2 integer,
dtel2 integer,
dext2 integer,
cnombre char(45),
crfc char(13),
Datenti (cont.)
cpuesto char(45),
revision
ejer integer not null,
clv_ent integer not null,
n_rev integer not null,
clv_pro integer not null,
enfoque integer,
tipo char(1),
semanas integer,
trim integer,
t_sem integer,
sh1 integer,
sh2 integer,
sh3 integer,
sh4 integer,
st1 integer,
st2 integer,
st3 integer,
st4 integer,
s_ini integer,
fec_ter date,
descripcion text,
objetivo text
programa97
clv_pro integer not null,
desc1_pro char(50),
desc2_pro char(50),
enfoque1 integer,
enfoque2 integer
observacion
ejer integer not null,
clv_ent integer not null,
n_rev integer not null,
n_obs integer not null,
n_iden integer,
clv_obs integer,
tipo char(1),
criterio char(1),
trim_obs integer,
monto_ori integer,
anio_apl integer,
trim_apl integer,
aceptada char(1),
fec_pro date,
aceptada_p char(1),
oic char(1),
observacion text,
recomendacion text,
resumen text,
penal char(1),
admon char(1)
rfc
ejer integer,
clv_ent integer,
n_rev integer,
n_obs integer,
rfc char(13),
nombres char(20),
apate char(20),
amate char(20)
seguimiento
ejer integer not null,
clv_ent integer not null,
n_rev integer not null,
n_obs integer not null,
anio_seg integer not null,
trim_seg integer not null,
g_avance integer,
sit_obs integer,
anio_rep integer,
trim_rep integer,
monto_rec integer,
fec_pro date,
seguimiento text
avaltex
ejer integer,
clv_ent integer,
trimes integer,
cumplimiento text,
problematica text,
observaciones text,
cuadro text,
otros text
registro
numero integer,
entidad integer,
ejer integer,
trimestre integer,
ejercicio integer,
revision integer,
observacion integer

```
ccalle char(45),
ccolonia char(25),
cciudad char(20),
ccp integer,
cedo char(20),
clada1 integer,
ctel1 integer,
cext1 integer,
clada2 integer,
ctel2 integer,
cext2 integer,
clavepgp char(20),
d_email char(50)
```

```
costo
    ejer integer not null,
    trimes integer not null,
    clv_ent integer not null,
    no_rev integer not null,
    costo float
    notas text
```

```
entcosto
    ejer integer not null,
    trimes integer not null,
    clv_ent integer not null,
    areas char(40) not null,
    no_revs integer,
    costo float,
    marca char(1)
```

```
recup
    ejer integer not null,
    trimes integer not null,
    clv_ent integer not null,
    fianz float,
    gasto float,
    cuotas float,
    pagos float,
    irreg float,
    falta float,
    antic float,
    reint float,
    impue float,
    sanci float,
    otros float,
    responsable char(100),
    telefono char(20),
    nota text
```

```
entrecup
    ejer integer not null,
    trimes integer not null,
    clv_ent integer not null,
    areas char(100) not null,
    total float,
    fianz float,
    gasto float,
    cuotas float,
    pagos float,
    irreg float,
    falta float,
    antic float,
    reint float,
    impue float,
    sanci float,
    otros float,
    nota1 text
```

Relación de Programas.

PROGRAMA	DESCRIPCION
Acerca	VERSION DEL MODULO SIP/PACA
bom1	REPORTE ESPECIAL DE REVISIONES PROGRAMADAS AL TRIMESTRE
Cuadro2	GENERACION DE LA EVALUACION TRIMESTRAL
Datent	DATOS BASICOS DE LA ENTIDAD
Entidad	CATALOGO DE ENTIDADES
envcor	ENVIA CORREO VIA INTERNET
esp1	CUMPLIMIENTO DEL PROGRAMA ANUAL DE CONTROL Y AUDITORIA
esp2	NUMERO DE REVISIONES REALIZADAS AL TRIMESTRE
esp3	SITUACION DE LAS REVISIONES PROGRAMADAS AL TRIMESTRE
esp4	OBSERVACIONES DETERMINADAS POR INSTANCIA FISCALIZADORA
esp5	OBSERVACIONES REPORTADAS POR RUBRO ACUMULADAS AL TRIMESTRE
esp6	OBSERVACIONES REPORTADAS, SOLUCIONADAS Y PENDIENTES DE ATENDER AL TRIMESTRE
esp7	OBSERVACIONES PENDIENTES DE ATENDER POR INSTANCIA FISCALIZADORA
esp8	TOTAL DE OBSERVACIONES PENDIENTES DE ATENDER POR ANTIGUEDAD Y CON PRESUNTA RESPONSABILIDAD
esp9	OBSERVACIONES PENDIENTES DE ATENDER POR RUBRO
evaluacion	DATOS BASICOS DE LA EVALUACION TRIMESTRAL
exporta	EXPORTACION DE TRIMESTRAL DE LA BASES
fcpassword	CAMBIO DE PASSWD POR PARTE DEL USUARIO
fgsip	MOVIMIENTOS DE REVISIONES, OBSERVACIONES Y SEGUIMIENTOS
FmxUtils	FORMA AUXILIAR DE DELPHI
g2p	(G2) Captura de la Situacion de Revisiones
g3	(G3) Cumplimiento P.A.C.A. No. de Revisiones Del Trimestre
g4	(G4/1) Revisiones Programadas (Numero de Semanas-Hombre)
g6	(G6) Textos de Observaciones Para Validar
g7	(G7) Observaciones Pendientes de Atender para Indicar Seguimiento
g7cmh	(G7cmh) Observaciones de CMH Sin Certificación
g7pre	(G7pre) Observaciones de Presunta Sin Certificación
g8	(G8/1) Problematica Reportadas,Solucionadas y Pendientes
g8_2	(G8/2) Observaciones Con Presunta Responsabilidad
g8_3	(G8/3) Observaciones Relevantes Pendientes
g9	(G9) Textos de Observaciones Relevantes Pendientes
genrep	GENERACION DE REPORTES DE LA ENTIDAD
genrepoic	GENERACION DE REPORTES DEL OIC
ht	GENERACION DE LA EVALUACION PARA INTRANET/INTERNET
infor	GENERACION DE INFORMES ESPECIALES
Intro	PANTALLA PRINCIPAL DEL MODULO
mensajes	MUESTRA MENSAJES IMPORTANTES DEL SISTEMA
Paca	PROGRAMA ANUAL DE CONTROL Y AUDITORIA
passwd	ACCESO AL MODULO (LOGIN Y PASSWD)
presun	OBSERVACIONES TRASFERIDAS AL SISTEMA AUXILIAR DE CONTROL QUE PRESENTAN TRAMITE DE RESPONSABILIDAD ADMINISTRATIVA
programa	CATALOGO DE PROGRAMAS
recepcion	RECEPCION VIA DISCO O INTERNETDEL SIP-PACA-RECUPERACIONES
RECREP	INFORMACION ESTADISTICA DE RECUPERACIONES
RECUPE	MODULO DE CAPTURA DE RECUPERACIONES

repcap	SELECCIÓN DE REPORTES DE RECUPERACIONES
repcoss	INFORMACION ESTADISTICA DEL MODULO DE COSTO
REPCOSTO	REPORTE DE COSTOS
rg3	(G3) Cumplimiento P.A.C.A. No. de Revisiones Al Trimestre
secrep	RECUPERACIONES DEL ORGANO INTERNO DE CONTROL
usuarios	CAPTURA DE USUARIOS QUE ACCESAN AL MODULO

Referencias Cruzadas Programa-Tabla.

PROGRAMAS	TABLAS															
	costo	Datenti	enticosto	Entidad	entrecup	evaltex	observacion	paca	passwd	programa g7	recup	registro	revision	rfc	sector	seguimiento
Acerca																
bom1				X									X			
Comobj																
cuadro2						X	X					X	X			X
datent		X														
entidad				X											X	
envcor																
esp1				X									X		X	X
esp2				X									X		X	X
esp3				X									X		X	X
esp4				X			X								X	X
esp5				X			X								X	X
esp6				X			X								X	X
esp7				X			X								X	X
esp8				X			X								X	X
esp9				X			X								X	X
evaluacion						X		X				X			X	
exporta		X	X		X		X	X					X			X
fcpassword									X							
fgsip							X	X		X			X	X		X
FmxUtils																
g2p				X									X			
g3				X				X		X			X			
g4				X				X					X			
g6				X			X									
g7				X			X	X								X
g7cmh				X			X	X								X
g7pre				X			X	X								X
g8				X			X			X						X
g8_2				X			X									X
g8_3				X			X			X						X
g9				X			X									X
genrep								X								

genrepoic								X								
globales													X			X
ht						X	X								X	
infor																
Intro																
mensajes								X					X			
paca									X							
passwd				X												
presun							X			X					X	
programa											X		X			X
recepcion	X		X		X		X				X					
RECREP											X					
RECUPE	X			X							X					
repcap																
repcoss	X															
REPCOSTO	X												X			
rg3				X				X								
secrep				X							X					
tcorreo																
usuarios									X							

Con reservas de poner toda la documentación del sistema, hasta aquí terminaré la explicación con respecto al diseño del Sistema de Información Periódica y el Programa de Control de Auditorías.

Repito aquí lo que he venido haciendo notar en cada capítulo, el trabajo pretende presentar la aplicación de una metodología orientada a objetos, dentro de un sistema que actualmente se maneja en las contralorías internas de cada dependencia gubernamental; y aunque presenta parte de la documentación del mismo, no pretende ser solo esto. Las características más particulares y las situaciones concretas no se explican aquí, debido a políticas de seguridad de la Dirección de Administración de Bases de Datos, y de la misma información que se maneja dentro de este sistema.

Los dos capítulos posteriores manejan la teoría involucrada en procesos clave para el sistema; y la finalidad de integrarlos a este documento es de empapar al lector de el trabajo de investigación que se realizó en la etapa de análisis del sistema.

Al fin de cada uno de ellos se podrá notar el por que de la elección de cada servicio que ocupa el sistema, dentro de una gama bastante amplia de estos.

En el capítulo cuatro se habla de las formas existentes de encriptar la información, comenzando con una pequeña historia de criptología y su evolución, hasta llegar a la explicación del programa utilizado en el SIP para el envió de la información.

El capítulo cinco sigue la misma metodología con los diferentes tipos de transmisión de datos y sus protocolos, explicando como trabaja el servicio empleado en el sistema.

CAPITULO IV. ENCRIPTACIÓN DE LA INFORMACIÓN.

La inclusión de este capitulo dentro de la presente tesis obedece a la necesidad de respaldar el uso de la encriptación de datos en los procesos de envío y recepción; ya que estos procesos utilizan un canal de comunicación común y accesible para un universo enorme de usuarios o personas que no deben ni tienen por que acceder a la información de nuestro sistema como es el correo electrónico.
Describiremos la opción más viable que se eligió, y aunque realmente no se programó ningún algoritmo de encriptación si se estudió su funcionamiento y sus bases.
Empiezo pues con una pequeña introducción a los conceptos de encriptación, para posteriormente mencionar y repasar algunos algoritmos anteriormente usados, y por último la descripción y el manejo del programa "PGP" que se usó en el sistema para cifrar la información.

La Criptografía podría definirse como la ciencia que estudia la manera de cifrar y descifrar datos para que resulte si no imposible por lo menos extremadamente difícil conocer el contenido de los mismos a personas que no dispongan de un algoritmo o en su caso de una clave específica. Etimológicamente criptografía significaría escritura extraña (Criptos: extraño, Graphos: escritura).

Los códigos de encriptación no son otra cosa que un conjunto de símbolos que permiten pasar los datos de un lenguaje claro para el total de nuestro universo de usuarios a otro que solo entenderán un limitado número de éstos; con esto queremos decir que solo aquellos que posean las herramientas para interpretar estos códigos podrán comunicarse entre si evadiendo al resto de usuarios.

De lo anterior podemos definir codificación o cifrado como el proceso de aplicar un código a la información para transformarlo ilegible para la mayoría de nuestros usuarios, y decodificación o descifrado como el proceso de trasformar la información cifrada a su lenguaje natural quitando el código utilizado.

Las claves de encriptación o las llaves de encriptación son variables introducidas en la codificación de los datos para hacer más complejo el descifrado de la información para gente extraña a la misma.

Como mero repaso mencionaré los tipos de cifrados conocidos dividiéndolos en cifrados clásicos y cifrados actuales.

Cifrados Clásicos.

Aquellos utilizados en la antigüedad y cuyo algoritmo de encriptación son muy simples, dentro de ellos tenemos:

Cifrado por sustitución.- consiste en cambiar un símbolo o un conjunto de símbolos del lenguaje común por otro símbolo que podía también pertenecer al mismo lenguaje.

Cifrador de alfabetos desplazados circularmente.- como una variación del cifrado por sustitución, este tipo de cifrado consiste en utilizar el mismo lenguaje para la codificación, únicamente desplazando cada símbolo del mensaje en "n" posiciones dentro del orden natural del alfabeto, donde "n" es la clave de codificación.

Cifrador monoalfabético.- básicamente es el mismo que el anterior, pero la sustitución de cada símbolo por otro no guarda una relación de desplazamiento dentro del alfabeto, esto lo remplaza una tabla de asignación con un total de asignaciones igual al número de símbolos dentro de nuestro alfabeto. La persona que quisiera descifrar un mensaje aparte de saber la forma de cifrado debe conocer la llave que en este caso seria nuestra tabla de asignación, este tipo de cifradores no son tan seguros, sobre todo con la nuevas técnicas de análisis lingüístico, el estudio de las propiedades estadísticas de los lenguajes naturales que analizan las repeticiones de diferentes combinaciones de los símbolos etc, esto con una buena herramienta de calculo como lo son los ordenadores actuales permite el fácil acceso a los datos cifrados mediante esta forma.

Cifrador polialfabético .- el número de alfabetos para el cifrado se multiplica, es decir para cada símbolo del texto a cifrar, se ocupa un alfabeto de encriptación diferente, siguiendo una rotación de alfabetos secuencial, y comúnmente de acuerdo a una clave. Para complicar más este tipo de cifrados, se pensó en cifrar no un símbolo sino una agrupación de estos, es decir se comenzaron a cifrar diagramas y triagramas.
Conforme creció la unidad de encriptación, se creo el concepto de código, que se diferencia de cifrado por que en lugar de cifrar un símbolo del alfabeto, cifra una unidad lingüística como lo son las palabras o incluso frases, para poder codificar o decodificar un texto es necesario la existencia de un libro de código, que hace las veces de una clave de cifrado.
La combinación de códigos con cifrados da lugar a los supercifrados, en estos primero pasamos el texto sobre un libro de código, el resultado obtenido lo ciframos símbolo a símbolo utilizando un cifrador polialfabetico.

Cifradores de transposición.- La diferencia de los cifradores de transposición frente a los códigos y los cifradores de sustitución, es que los segundos preservan el orden de los símbolos del texto original, pero los enmascaran, en cambio los primeros no enmascaran los símbolos, pero si alteran su orden.

Cifrados Actuales.

Algoritmo de DES.- Con la aparición de los ordenadores, la encriptación se llevo hasta el hardware, tenemos por ejemplo los cifradores de circuitería, que son circuitos electrónicos que realizan transposiciones y sustituciones de los bits de información. Uno de los algoritmos más conocidos en este tipo de cifradores es el algoritmo de DES.

En el algoritmo de DES el cifrado del texto original se realiza en bloques de 64 bits (que dan como resultado bloques de 64 bits cifrados). El algoritmo tiene como parámetro una clave de 56 bits y consta de 19 etapas diferentes:

La primera etapa es una transposición (que no está parametrizada por la clave).
Las otras 16 etapas siguientes realizan un intercambio de bits con parte de la clave siguiendo una secuencia de la

parte que le corresponde de dicha clave.
La penúltima etapa realiza un intercambio de bits. Intercambia los 32 bits de la parte izquierda con los 32 bits de la parte derecha.
La última etapa realiza el proceso inverso de la transposición realizada en la primer etapa.

Para entender mejor este algoritmo explicaré el funcionamiento de una de las 16 etapas intermedias donde se realiza un intercambio de bits entre el texto cifrado y parte de la clave.
Cada una de estas etapas toma dos entradas de 32 bits y produce dos salidas de 32 bits. Las transformaciones que realiza a estas entradas son:
La salida de la izquierda es una copia de la entrada de la derecha.
La salida de la derecha es un OR EXCLUSIVO bit a bit de la entrada de la izquierda, y una función de la entrada de la derecha y la clave de la etapa (Ki). La complejidad de cada etapa reside en esta función.
Esta función tiene cuatro pasos ejecutados secuencialmente:
Se construye un numero N de 48 bits mediante la expansión de los 32 bits derechos de entrada a la etapa mediante una regla fija de transposición y duplicación.
Se realiza un OR EXCLUSIVO de N y Ki .
La salida anterior se divide en 8 grupos de 6 bits, cada uno de los cuales alimenta una caja-S diferente. Cada una de las cajas-S produce salidas de 4, en lugar de 6 bits. Cada una de estas 64 posibles entradas a una caja-S se corresponde con salidas de 4 bits.
Estos 32 bits pasan por una caja-P.
Cada una de estas etapas utiliza una clave diferente (extraída de la clave de 56 bits). Además antes de todo se aplica una transposición de 56 bits a la clave y antes de cada iteración se divide la clave en dos unidades de 28 bits, cada una de las cuales es rotada a la izquierda según el número de bits que depende del número de iteración. El valor de Ki se obtiene de esta clave rotada, haciendo una transposición de 56 bits sobre ella.
Aunque pueda parecer muy complejo, el algoritmo DES es un cifrador de substitución monoalfabética que usa un carácter de 64 bits.

Además en el mes de julio de 1998 se pudo romper la seguridad del mismo descifrando un mensaje cifrado con ese algoritmo en menos de tres días obviamente con la ayuda de un ordenador. El problema con el algoritmo de DES no es su diseño sino el empleo de una llave bastante corta (56 bits) que provoca que un ataque por la fuerza bruta (llamamos ataque por fuerza bruta a la prueba llave por llave para descifrar algoritmos) pueda llegar a descifrar un mensaje. No obstante que el algoritmo ya ha sido violado, se siguen usando una serie de variaciones del mismo en la actualidad.

Algoritmo IDEA.- Como alternativa del algoritmo de DES surgió el algoritmo IDEA que trabaja con bloques de 64 bits y emplea una llave de 128 bits, la longitud de su llave impide el ataque por fuerza bruta como no ocurrió con el de DES.
A continuación explicaré brevemente la funcionalidad del algoritmo IDEA.
Lo primero es la división del bloque de 64 bits en cuatro partes de 16, a estas cuatro partes se les aplican operaciones

junto con 52 subclaves de 16 bits; estas 52 subclaves se obtienen de la clave de 128 bits empleada en este algoritmo. Las operaciones realizadas se repiten en ocho rondas, cada ronda consta de las siguientes operaciones:
Llamaremos X1, X2, X3 y X4 a las cuatro partes en las que se divide el bloque a cifrar, y Zi a cada una de las 52 subclaves obtenidas de la clave de encriptación.

1.- Se multiplica X1 por Z1.
2.- Se suma X2 mas Z2.
3.- Se suma X3 con Z3.
4.- Se multiplica X4 por Z4.
5.- Se aplica XOR a los resultado de los pasos 1 y 3.
6.- Se aplica XOR a los resultado de los pasos 2 y 4.
7.- Se multiplica el resultado 5 por Z5.
8.- Se suma los resultados 6 y 7.
9.- Se multiplica el resultado 8 por Z6.
10.- se suman los resultados 7 y 9.
11.- Se aplica XOR entre los resultado 1 y 9.
12.- Se aplica XOR entre los resultado 3 y 9.
13.- Se aplica XOR entre los resultado 2 y 10.
14.- Se aplica XOR entre los resultado 4 y 10.

Al terminar cada iteración de las anteriores operaciones los resultados obtenidos son de los pasos 11,12,13 y 14 que son cuatro bloques de 16 bits, para la siguiente iteración se toman como entradas (X1,X2,X3 y X4), junto con las siguientes seis subclaves (Z7,Z8,Z9 y Z10). Al termino de las ocho iteraciones ya se han ocupado 48 subclaves, posteriormente se realizan las siguientes operaciones:

1.- Multiplicar X1 por Z49.
2.- Sumar X2 con Z50.
3.- Sumar X3 con Z51.
4.- Multiplicar X4 por X52.
Obteniendo con ello el bloque codificado.

Para sacar las 52 claves de 16 bits cada una, de la clave global de 128 bits, se aplica la siguiente regla, las primeras 8 claves se obtienen dividiendo la clave en bloques de 16 bits; para las restantes primero se rota la clave de 128, 25 bits a la izquierda y posteriormente se divide en ocho subclaves; esta operación se repite hasta lograr el total de claves requeridas.

Algoritmos de Clave Pública.- Los anteriores métodos de encriptación, son clasificados como simétricos ya que ocupan la misma clave tanto para cifrar como para descifrar el mensaje, esto traía como consecuencia el problema de transmisión de la llave de encriptamiento. En la actualizada se utilizan mecanismos más elaborados de encriptación;

es muy común el uso de algoritmos asimétricos que tiene la característica de poseer dos claves una para encriptar y otra para desencriptar, la deducción de una mediante la otra es extremadamente complicada, y de esto depende la seguridad de estos método de encriptación.

El método antes planteado se conoce como algoritmo de clave pública del MIT, y toma su nombre debido a que una clave de encriptación es conocida públicamente, y la otra clave que nos sirve para desencriptar es conocida como privada, y solo la posee el individuo al que va dirigido el mensaje.
El problema a este método es encontrar unos algoritmos que cumplan los requisitos especificados.

Algoritmo RSA.- En 1978, investigadores del MIT encabezados por Rivest descubren unos de los mejores algoritmos para cifrado de clave pública conocidos. A grandes rasgos el método consiste en lo siguiente:

1.-Se eligen dos números primos grandes mayores que 10^100, Primo1 y Primo2.
2.-Se calcula un tercer numero mediante la operación N1 = Primo1 * Primo2
3.-Se calcula un cuarto número con N2 = (Primo1-1) * (Primo2 - 1)
4.-Se elige otro número de tal forma que sea un primo con el número N2, es decir que estos dos números no tengan factores en común, a este quinto número lo llamaremos N3.
5.-Se busca un sexto número que multiplicado por el anterior, sea igual a 1 mod N2. donde mod es la operación módulo (x mod y = x-(x div y)*y). Quedando N4 * N3 = 1 mod N2.

La clave pública obtenida estaría constituida por el par (N3,N1) y la clave secreta por (N4,N1).
Teniendo las dos claves, para cifrar o descifrar un mensaje se haría de la siguiente forma.

Dividimos el texto no cifrado en bloques, el valor de cada bloque debe de ser menor al número antes calculado N1. Generalmente estos bloques se agrupan en n bits, de tal forma que 2^ n sea menor que N1 con el mayor valor posible.

Dado el mensaje M, para encriptarlo (C), hallaríamos C=M^N4 (mod N1).

Para desencriptarlo (D), sería D = M = C^N3 (mod N1).

La razón de que este método pueda considerarse seguro se basa en que es matemáticamente muy complicado la factorización de números grandes. De lo contrario sería posible factorizar N1 que se conoce, pues está como clave pública, y extraer primo1 y primo2 a partir de estos factores, con lo que podría ser determinado el valor de N2. Disponiendo del valor de N1 y N2, podría deducirse fácilmente el valor de N3 (Euclides) con lo que obtendríamos ya la clave secreta.

Al anterior método de encriptación se le conoce con el nombre de algoritmo RSA.

Métodos de Autentificación.

Existe un grave problema dentro de los procesos de envío, recepción de información por vía electrónica . y es el de autentificar el transmisor y el receptor, es decir que uno y otro puedan estar seguros de la identidad del otro.
En un inicio se emplearon todo tipo de métodos para autentificar al emisor de un mensaje, se autentificaba a través de claves de usuario cuando este inicia la comunicación; esto hacía necesario poner en manos de una determinada entidad un listado de contraseñas lo cual era un gran problema de seguridad y donde el usuario se ve obligado a poner en manos de la entidad la confidencialidad de sus propios movimientos.

Dentro del método de cifrado de clave pública las primeras prácticas de autentificación consistían en métodos como el explicado a continuación. Cuando una de las partes involucradas en la comunicación deseaba hacer una petición confidencial a la otra, seleccionaba un número al azar y lo enviaba al transmisor cifrado con la clave pública de éste. El transmisor a su vez reenvía el mensaje cifrado mediante el número aleatorio que aquella seleccionó. Además de el número aleatorio, se podían adicionar claves secretas, fechas de envío y demás conceptos para obtener mayor seguridad en la autentificación del emisor.
Posteriormente se empleo el concepto de funciones resúmenes para autentificación. Esto es, suponiendo que A recibe un mensaje de B, y desea saber si realmente B es quien lo envió, lo primero es que B genera una función sobre el mensaje M, dicha función la encripta con su clave privada y la manda a A; con la clave pública de B que A tiene puede decencriptar esta función y verificar con ella a B. En realidad lo que llamamos funciones resúmenes no son otra cosa que una firma digital.

Firmas Digitales.

Las firmas digitales hoy en día forman parte de los procesos de encriptación, y resuelven el problema de autentificación de uno y otro lado (recepción-transmisión). Como se vio anteriormente el método de encriptamiento por llave pública debe cumplir con la formula:

$$D(E(M))=M.$$

Donde D es el algoritmo de descifrado, E es el de cifrado y M el mensaje.

El uso de una forma digital además involucra la siguiente formula:

$$E(D(M))=M.$$

Es decir consiste en claves duales, donde la clave de encriptación tiene su respectiva llave de descifrado. y cuando esta se emplea para descifrar la otra se emplea para cifrar.

Para explicar las firmas digitales vamos a suponer que una persona A transmite un mensaje cifrado con firma digital

mediante criptografía de clave pública utilizando el algoritmo que hace uso de su clave privada D para encriptar el mensaje, y a continuación lo encripta de nuevo con el algoritmo E' que usa la clave pública de la parte receptora, obteniendo el mensaje Z como lo indica la siguiente formula.

$Z = E'(D(M))$

Cuando el receptor, que llamaremos B, lo recibe lo transforma mediante:

$M = E(D'(Z))$

Donde E es el algoritmo con clave pública de A y D' el algoritmo con clave privada de B. Además D'(Z) coincidiría con D(M). Imaginemos que A negara haber transmitido el mensaje M a la entidad B. En ese caso un juez podría comprobar a partir del mensaje cifrado D(M) que B si conoce, que aplicándole el algoritmo con clave pública E de A, se obtendría el mensaje originalmente enviado, M.

$E(D(M)) = M.$

Como B no puede conocer la clave secreta de A, la única forma que tuvo B de obtener el mensaje cifrado D(M) es que A efectivamente lo hubiera transmitido.

Realmente la firma digital no se obtiene de cifrar todo el mensaje sino solo una función resumen, para esto las funciones resúmenes deben cumplir con las siguientes características.

1.- La longitud de la función resumen r(m), debe ser independiente de la longitud del mensaje.

2.-Teniendo el mensaje m, debe ser fácil sacar r(m)

3.-Teniendo la función r(m), debe ser prácticamente imposible obtener m.

4.-Teniendo el mensaje m, debe ser imposible obtener otro mensaje m' , tal que r(m)=r(m');

Existen varios algoritmos para la generación de firmas digitales entre ellos el MD5 y el DSA, el primero es el empleado dentro del programa PGP, en la versión que nuestro sistema ocupa.

PGP (pretty good privacy).

El pgp es una aplicación que encierra algoritmos de encriptación simétricos y asimétricos, totalmente gratuita y distribuible, características que para nuestro fin son de importancia. Actualmente es tan común el uso de este programa que se ha convertido en un estandar internacional. Se encuentra comúnmente en el envío y recepción de mensajes vía correo electrónico, pero incluso se puede hallar en la encriptación de particiones de discos duros (pgpdisk). Los algoritmos clave para este programa son el algoritmo IDEA y RSA.

La versión empleada en nuestro sistema es la versión 2.6.3ia, que es una de las primeras versiones internacionales del PGP, actualmente se están utilizando versiones gratuitas 6.0.2i y comerciales 6.5.1. Nuestro sistema no ha actualizado de pgp debido que la simplicidad para el manejo de la vieja versión incorpora una vía más flexible para todo equipo de computo, evitando con ello problemas para configurar equipos totalmente ajenos a nuestra administración.

Funcionamiento del PGP.- lo primero que hace el pgp es el cifrado simétrico de nuestro mensaje con una llave generada aleatóriamente, para posteriormente codificar la llave con la clave pública del destinatario. En el proceso inverso, para desencriptar, busca en las cabeceras del mensaje la llave pública y pide una contraseña para buscar la llave privada que permita descifrar el mensaje.

Como podemos prever del texto anterior, el pgp tiene implícito los conceptos de anillos de claves publicas y privadas, que no son otra cosa que los recipientes de las llaves que tenemos a nuestra disposición, estos archivos generalmente contienen un identificador de las claves publicas y una contraseña para la obtención de las mismas, por ejemplo a la hora de querer descifrar un mensaje el pgp localiza la llave pública de las cabeceras del mensaje, y después pide la contraseña que permite sacar del anillo de llaves privadas la que le permita continuar con su proceso.

Por lo anterior existen tres archivos indispensables para el funcionamiento del pgp, archivos que deben cuidarse con atención, el secring.pgp que es el anillo de llaves privadas y generalmente contiene solo la clave privada única de la entidad , el pubring.pgp que es el anillo de llaves publicas y contiene todas las llaves posibles de destinatario (llaves que incluso pueden publicarse en internet). El tercer archivo es randseed.bin que contiene la semilla de generación de números aleatorios, este archivo se considera sensible debido a que si alguien puede descifrar la secuencia de generación de claves aleatorias de esta semilla, entonces también puede descifrar cualquier mensaje cifrado con cualquiera de las llaves generadas por este.
Generalmente la forma de encriptamiento de mensajes en el sistema expuesto es:

Pgp –es archivo destinatario –u origen.

Donde :

pgp .- es el programa.
-es .- son parámetros que indican que se va a encriptar y a firmar el mensaje.
Archivo .- indica el objeto que se va a encriptar.
Destinatario .- es el destino del mensaje.
-u .- es un parámetro que antecede a la identificación del usuario que va a firmar el mensaje.
Origen .- es el identificador del usuario que firma el mensaje, normalmente el origen.

Esta instrucción normalmente debe de seguir a una serie de comandos que tiene por objetivo colocar variables de ambiente que permiten localizar las claves que se van a utilizar dentro de los anillos de encriptación.
set tz=pst8pdt .- pone la variable tz que indica la obtención de fecha y hora.

set pgppath .-Coloca la trayectoria donde se localiza el programa pgp.

set pgppass .- Indica una contraseña por la cual se va ha buscar las claves de encriptación.

Por lo expuesto anteriormente, podemos establecer claramente que el PGP debido a los algoritmos de encriptación que ocupa, sigue siendo una manera muy segura de enviar información, siempre y cuando se tenga cuidado en el manejo de los anillos de encriptación utilizados. Por tal razón y en virtud de que es un programa bastante accesible y fácil de utilizar, en el análisis del presente sistema se opto por el manejo de éste, y como se explico en párrafos anteriores en su versión 2.6.3.ia para MSDos.

El proceso de cifrado persigue cubrir los requerimientos de seguridad al enviar la información de las Contralorías Internas hacia la DGACS y viceversa, pero no toca para nada la vía de comunicación entre uno y otro. El capitulo siguiente planteará la teoría que conlleva a la selección de un medio de comunicación apto y disponible para nuestro sistema.

CAPITULO V.- TRANSMISIÓN DE DATOS

Antes de la existencia de nuestro sistema, los formatos de la información se mandaban en hojas de EXCEL o WORD estandarizadas; como se dijo en un inicio, esto variaba la forma en que se presentaban los datos dependiendo de la interpretación que diera cada una de las contralorías a la formas patrón. Además normalmente ocupaban un disco magnético acompañado de las respectivas impresiones de cada formato así como sus oficios pertinentes para la entrega a la DGACS. Con el diseño del sistema automatizado se pretende unificar la forma de envió sin permitir interpretaciones erróneas, además de utilizar una forma de envió mucho más flexible y rápida.

Dentro del estudio que se hizo se plantearon varios tipos de proceso por los que las contralorías pudieran hacer llegar la información a la DGACS, estos indicaban la existencia o en su caso la implementación de varios tipos de servicios dentro de un servidor de la SECODAM, obviamente al final se tuvo que optar por uno que estuviera al alcance de esta dirección, sin embargo en este capitulo describiremos cada una de las opciones que de inicio se propusieron.

Servidores FTP. (File Transfer Protocol) .

Como su nombre lo indica este protocolo sirve para transferir archivos de una máquina a otra en ambas direcciones. Para ello es necesario que el cliente se firme a través de un "login" y "password" dentro de un servidor; dentro de este punto el encargado de los procesos de autentificación y acceso es el servidor.
El FTP en realidad ocupa el TCP (Transmisión Control Protocol) como su protocolo de transferencia, y mantiene una estructura de cliente –servidor.
En ambas partes cliente –servidor se pueden observar los mismos componentes en una conexión con FTP.

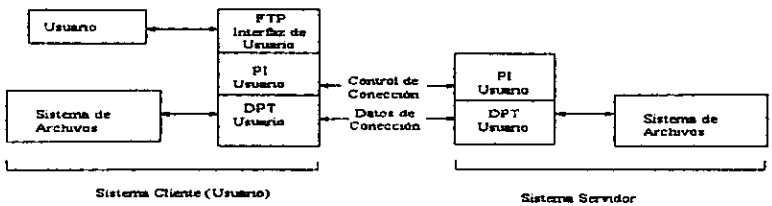


Figura 5.1.

Dentro del sistema de usuario podemos ver una interfaz que tiene comunicación directa con un interprete de protocolo (PI) que se encarga del control de la conexión con el servidor que también posee un interprete de protocolo

set pgppath .-Coloca la trayectoria donde se localiza el programa pgp.

set pgppass .- Indica una contraseña por la cual se va ha buscar las claves de encriptación.

Por lo expuesto anteriormente, podemos establecer claramente que el PGP debido a los algoritmos de encriptación que ocupa, sigue siendo una manera muy segura de enviar información, siempre y cuando se tenga cuidado en el manejo de los anillos de encriptación utilizados. Por tal razón y en virtud de que es un programa bastante accesible y fácil de utilizar, en el análisis del presente sistema se opto por el manejo de éste, y como se explico en párrafos anteriores en su versión 2.6.3.ia para MSDos.

El proceso de cifrado persigue cubrir los requerimientos de seguridad al enviar la información de las Contralorías Internas hacia la DGACS y viceversa, pero no toca para nada la vía de comunicación entre uno y otro. El capitulo siguiente planteará la teoría que conlleva a la selección de un medio de comunicación apto y disponible para nuestro sistema.

encargado en este caso de responder la petición del usuario. Existe un tercer elemento en la aplicación cliente y a su vez también en el servidor un proceso de transferencia de datos (DTP), este es el encargado de trabajar directamente con el sistema de archivos de la máquina y del mantenimiento de los datos durante la transferencia de los mismos de uno a otro punto. Una vez realizada la comunicación el PI del servidor debe cerrar la conexión.

A continuación mencionaremos los pasos mas comunes en una conexión con FTP.

1.- Conectarse al Host remoto.

2.- Seleccionar Directorios.

3.- Listar los archivos disponibles para transportarlos.

4.- Definir el modo de transmisión.

5.- Copiar los archivos desde o hacia el host remoto.

6.- Desconectar la conexión.

Normalmente estas acciones se llevan a cabo con comandos específicos. Así por ejemplo:

Para conectarse a un host remoto es necesario un login (ID) y un password de entrada como parte de la seguridad del protocolo, el comando para establecer una conexión es: OPEN seguido del identificador de host ya sea su nombre o su IP.

El comando USER identifica el ID del usuario que se intenta firmar al servidor.

PASS indica el password útil para firmarse (autentificarse).

Para seleccionar directorio tenemos el comando CD , es obvio que dentro del servidor un ID debe tener un directorio de trabajo que limita un espació de memoria donde el dueño del ID puede trabajar sin restricciones, fuera de este un usuario no puede realizar ninguna operación.

Para listar los archivos dentro del directorio actual esta el comando LS o DIR según el sistema operativo de la máquina.

Para transferir archivos de una plataforma a otra es común que los datos deban sufrir un proceso de transformación, por lo que el usuario debe decidir sobre dos aspectos: la forma en que los bits deben ser movidos de un lugar a otro. y las diferentes representaciones de los datos en la arquitectura del sistema.

Con respecto al primero, el comando MODE indica la estructura en que los datos pasan de un lugar a otro, teniendo dos formas:

BLOCK el registro lógico de el archivo es preservado.

STREAM el archivo es tratado como una cadena de bytes, este es el mode por default.

Desde el punto de vista del segundo aspecto el comando TYPE indica la tabla de caracteres (character set) que se utilizará para representar los datos, pudiendo ser :

ASCII, EBCDIC ambos códigos de caracteres conocidos, o de tipo IMAGE o BINARIO donde los datos son tratados como paquetes continuos de 8 bits (Byte).

Los comandos más conocidos para transportar archivos son:

Get, Mget.-encargados de copiar un archivo en el caso del primer comando o múltiples archivos en el caso del segundo comando del host remoto al host local.

Put, Mput.- encargados de copiar un archivo en el caso del primer comando o múltiples archivos en el caso del segundo comando del host local al host remoto.

Los comandos para abandonar una conexión FTP son:

Close.- desconecta del host remoto y queda sobre el programa FTP del cliente.
Quit.- Desconecta del host remoto y a la vez abandona también el FTP del cliente. Este comando en muchas ocasiones es cambiado por el comando BYE.

Cuando se mantiene una conexión cliente-servidor en el protocolo FTP, ambos mantienen un dialogo permanente usando la convención del TELNET. Cuando el cliente usa algún comando, el servidor le responde con un código de replica (Reply Codes), que además puede contener comentarios que le pueden ayudar al usuario, aunque el programa FTP del cliente solo utilice los códigos..
Los códigos de replica tiene la longitud de tres dígitos, y el digito de la izquierda representa cierta clasificación de respuesta del servidor como a continuación describimos:

1xx	es una respuesta preeliminar positiva del host remoto.
2xx	es una replica completa positiva del host remoto al cliente.
3xx	es una replica intermedia positiva del host.
4xx	es una replica negativa del host.
5xx	es una respuesta negativa permanente del host.

Así por ejemplo, en una sesión se puede tener lo siguiente:

FTP nombre del host.
220 servicio listo.
USERNAME login del cliente.
331 nombre del usuario correcto.
PASSWORD clave del usuario
230 usuario conectado.
TYPE Image
200 comando correcto
. . .

Donde podemos ver los códigos de replica que el host remoto nos rebota para cada comando que utilizamos. Los dos dígitos de la derecha de cada código nos proporcionan más detalles acerca de la respuesta. dependiendo el servidor de FTP.
Algunos servidores implementan lo que conocemos como usuario anónimo (anonymous), que es un login y pasword convencional para permitir el accesos a ciertos directorio dentro del servidor, el login quedaría como "anonymous" y el pasword como "guest".

Servidores Web.

El World Wide Web es un sistema global de manipulación de hipertexto desarrollado inicialmente en 1989 por Tim

Berners Lee, que facilitaba en un inicio un camino fácil para compartir y buscar un documento en una cantidad considerable de equipos dispersos geográficamente.
Hoy en día los servidores de Web y los browsers se encuentran disponibles en todas las computadoras y para todo tipo de plataformas.

Browser.- Un browser es una aplicación que provee acceso a un servidor de Web, normalmente se componen de un interprete de HTML y un cliente de HTTP el cual se usa para acceder a las paginas HTML, existen browsers que también implementan soporte para FTP, NNTP o E-MAIL . Esquemáticamente un browser se representa como lo indica la figura 5.2.

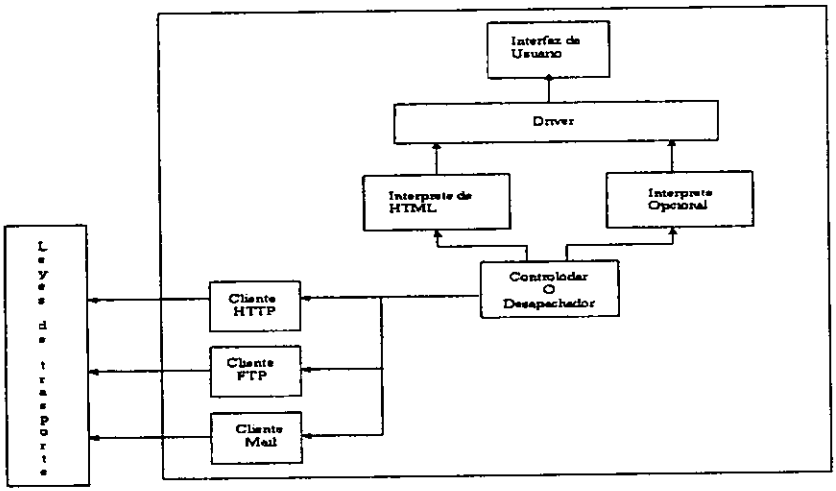


Figura 5.2.

Como la mayoría de las aplicaciones de Internet, el Web utiliza un modelo de procesamiento cliente/servidor. Donde el browser es el componente cliente encargado de formatear y desplegar la información, interactuando con el usuario e invocando a funciones externas tales como el TELNET o programas externos que el mismo browser no soporta directamente.

Servidores Web.- Los servidores de Web son los encargados de recibir el servicio de información requerido por los browsers. La información puede ser un archivo del disco local del servidor o pudiera ser generada por un programa llamado por el servidor.
El servidor de Web puede entonces tener dos tipos de contenido el estático normalmente representado por las paginas HTML, o el dinámico generado por programas que invoca el mismo servidor. Las paginas dinámicas generalmente se pueden clasificar según las siguientes formas:

CGIs (Common Gateway Interface).- Una de las primeras formas de realizar paginas dinámicas en un

servidor de Web son la utilidad de programas CGI, que se encuentran en el servidor y generan una salida en la consola del mismo (normalmente un archivo) basada en los parámetros o entradas del cliente. La mayoría de los servidores de Web en la actualidad soportan CGI's, y con ellos una gran variedad de lenguajes de programación de los mismos (llámese PERL, lenguaje "C" entre otros), sin embargo los programas CGI's no son tan fácilmente portables a través de las diferentes plataformas.

Server-Specific APIs.- Algunos servidores de Web permiten el desarrollo de programas que pueden utilizar llamadas a funciones especiales del sistema operativo, obviamente estos no tienen portabilidad a las diferentes plataformas existentes. Uno de los más comunes es el Netscape Server API (NSAPI) de Microsoft.

Servlets.- esta tecnología permite la invocación de programas escritos en el lenguaje de Java dentro de la memoria del servidor. La ventaja más importante es que es fácilmente transportable entre plataformas, e incluso existen los procedimientos sobrecargados, cuya principal característica es su múltiple definición.

Servlet-Side Includes (SSI) .- esta tecnología se utiliza para convertir una sección de una página HTML dentro de una porción dinámica alternativa, cada vez que el documento se envía al browser cliente. Esta porción dinámica invoca a un servlet y pasa a él los parámetros que necesita. Las páginas que utilizan esta tecnología tienen extensión ".shtml".

Java Server Pages (JSP).- esta es una solución fácil de usar para generar paginas HTML con un contenido dinámico. Un archivo JSP contiene combinaciones de instrucciones HTML, NCSA , <SERVLET> y sintaxis de Java. Una de las múltiples ventajas de esto es que permite a los programadores separar el código HTML de las reglas del negocio dentro de las páginas Web.

Hypertext Transfer Protocol (HTTP).

Para seguir describiendo los servidores de WEB, se tine que mencionar su protocolo estandar, el http que se utiliza para transferir documentos en lenguaje hipertexto HTML (HyperText Markup Languaje). Los documentos HTML incluyen ligas a otros documentos que contienen información adicional acerca de términos o sujetos. Estos documentos pueden contener otros elementos aparte de texto tales como gráficas, audio, video o programas Java Applets.

El http se basa en una actividad pregunta-respuesta. Un cliente corre su aplicación browser, establece una conexión con un servidor y envía su requerimiento a través de un método implementado en su aplicación. El servidor responde con una línea de estatus que incluye la versión del protocolo del mensaje y un código de operación exitosa o de error según sea el caso, siguiendo un mensaje que contiene la información del servidor.
Una transacción en http se divide en cuatro pasos (referirse a la figura 5.3):

1.- El browser abre una conexión.

2.- El browser envía un requerimiento al servidor.

3.- El servidor envía una respuesta al browser.

4.-La conexión se cierra.

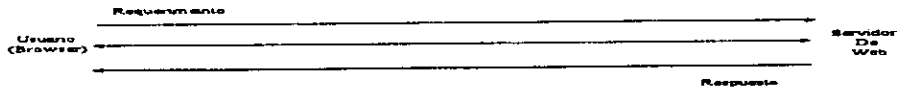


Figura 5.3.

En la mayoría de los casos, la comunicación es directa entre el browser cliente y el servidor de Web como se ve en la anterior gráfica (figura 5.3), sin embargo existen casos también muy comunes en los que los requerimientos y las respuestas de uno y otro pasan por intermediarios tales como proxys, gateway o túneles. En estos casos la información es evaluada según las políticas de seguridad de cada elemento.

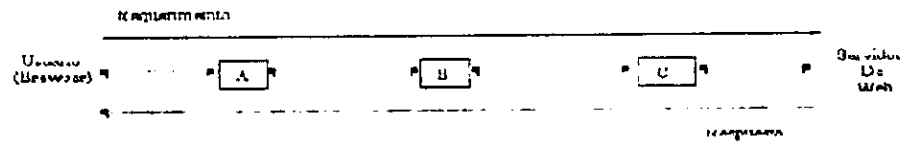


Figura 5.4.

El la figura 5.4 se describe la situación anterior, donde el intercambio entre browser y servidor pasa por un aserie de intermediarios.

La diferencia entre un proxy, un gateway y un túnel es que el primero cacha el contenido del dato y modifica parte o la totalidad de éste de acuerdo a sus políticas internas, y posteriormente lo envía al siguiente destinatario; un gateway recibe el mensaje y lo envía a la gama de protocolos internos con un formato apropiado y por último un Túnel lo que hace es reenviar el mensaje sin tocar ni su contenido ni su formato.

Los proxys y los gateway pueden cachar los mensajes del http, lo que no puede realizar los túneles ya que estos últimos no pueden entender el contenido del mensaje, el emplear mensajes cachados reduce el tiempo de respuesta y

el tráfico en la red, esto lo explica la figura 5.5.

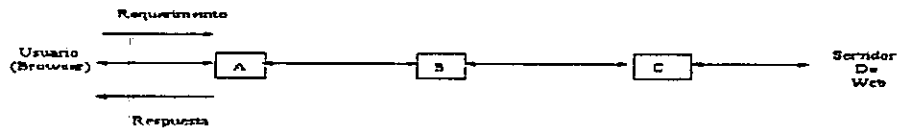


Figura 5.5.

Aquí el intermediario "A" cuenta con una copia de una respuesta sencilla del servidor de Web, y puede responder directamente al requerimiento del cliente sin pasar el mensaje a los demás intermediarios.
No obstante no todas las respuestas del servidor son susceptibles de cachar, y de hecho las respuestas del servidor pueden estar marcadas como no-cachables, publicas o privadas.

Parámetros del protocolo HTTP.

Algunos de los parámetros más comunes del HTTP son:

Versión.- se utilizan dos números separados por un punto decimal para identificar la versión del protocolo, el número de la izquierda se incrementa cuando el protocolo sufre una modificación significativa tal como el cambio del formato del mensaje. El número de la derecha del punto se incrementa con cambios de poca importancia que no afectan en la estructura del protocolo.
La versión del protocolo se envía en el campo HTTP-Versión dentro de la primera línea del mensaje .

HTTP-Version = "HTTP" "/" 1-DIGIT "." 1*DIGIT

Uniform Resource Identifiers (URI).- generalmente se refiere a una dirección WWW y una combinación de Uniform Resource Locator (URL) y Uniform Resource Name (URN); en realidad son cadenas que tienen la localización y el nombre de la fuente dentro de un servidor. El URL permite localizar la red fuente via el protocolo HTTP y su sintaxis es la siguiente:

HTTP_URL = "http" "//" host [":" port] [abs_path]

Donde el Puerto es opcional, y por default es el número 80.

Mensajes HTTP. Los siguientes son los campos en que podemos dividir un mensaje en el protocolo:

1.-Tipo de Mensaje.- puede ser ya sea un requerimiento del cliente o un respuesta del servidor:

HTTP-message = Request | Response

2.-Cabecera del Mensaje.- puede ser de los tipo:

Encabezado General.- se aplican tanto a los mensajes de requerimientos como a las respuestas, y pueden caer en:

Cache-Control.
Conection
Date
Pragma
Transfer-Encoding.
Upgrade
Via.

Encabezado de requerimiento.- Normalmente incluye un método que se aplica al recurso, un identificador de la fuente y una versión del protocolo.

Request = Request-Line
*(encabezados-generales | encabezados de requerimiento | encabezados de entidad)
CRLF
[cuerpo del mensaje]

Encabezado de Respuesta.- Después de evaluar el requerimiento, el servidor de WEB debe responder con el siguiente formato:

Response = Response-Line
*(encabezados-generales | encabezados de respuesta | encabezados de entidad)
CRLF
[cuerpo del mensaje]

Cuerpo del Mensaje.- Si un código de transferencia no ha sido aplicado, es el cuerpo de la entidad ya sea un requerimiento o una respuesta.
Longitud del mensaje.- indica la longitud del cuerpo del mensaje.

En un inicio, las versiones de http tenían un conexión separada de TCP para cada URL, y la aplicación cliente tenia

que realizar múltiples requerimientos para imágenes y objetos asociados sobre el mismo URL, esto ocasionaba congestión y un rendimiento bajo en el performance de la red. A partir de la versión 1.1 del http, este mantiene una conexión persistente por default.

Definición de Métodos.

Dentro del HTTP, existe diferentes formas de manipulación de la información, estos métodos son:

1.-OPTIONS.- permite al cliente determinar la opción o requerimiento asociado con una fuente o capacidad del servidor.

2.-GET.- Permite al cliente recuperar el dato que determinó el requerimiento URI a través de la variable de ambiente QUERY_STRING.

3.-HEAD.-Permite al cliente recuperar la metainformación sobre la entidad, que no es requerida para transferir el cuerpo del mensaje.

4.-POST.- Permite al cliente recuperar el dato que determinó el requerimiento URI, en este caso el dato se pasa a través de la salida estandar STDOUT.

5.-PUT.- este método es similar al POST exceptuando que en este último se identifica el recurso dentro de los encabezados.

6.-DELETE.- requiere que el servidor borre la fuente determinada por el requerimiento URI.

7.-TRACE.- permite ver al cliente la forma en que el mensaje fue requerido en el otro sitio para propósitos de examinar y diagnosticar la transacción.

Autentificación

El HTTP define dos mecanismos para permitir el acceso de clientes a cada uno de sus recursos.

Esquema básico de Autentificación.- se basa en login y password de usuarios, es decir el servidor debe verificar dentro de su tabla de usuarios si existe o no un cliente y validar sus claves de acceso.

Esquema de compendio.-aplica la encriptación a través de claves de usuario, obviamente el servidor debe contar con un directorio de claves para permitir el acceso a sus recursos. Este mecanismo es el más seguro, pero lamentablemente algunos browsers no lo soportan.

Caching.- una de las características más importantes del HTTP es su capacidad de guardar temporalmente una respuesta a un requerimiento por una cantidad de tiempo razonable, esto con el fin de responder a un futuro requerimiento igual sin necesidad de realizar una petición más al servidor; trayendo como consecuencia un tiempo de respuesta menor además de reducir el requerimiento del ancho de banda de la red. Existe por tanto un tiempo estimado en el que la respuesta guardada tiene validez para un requerimiento y este es calculado por el servidor. Además también existe un mecanismo que indica si el dato guardado ha cambiado o no.

Mecanismo de expiración.-indica si el dato guardado se debe actualizar con una petición al servidor o no. En la mayoría de los casos el mismo servidor adjunta en su respuesta el tiempo estimado en que ésta es válida antes de necesitar actualización. Si el servidor no definiera el tiempo de expiración, siempre hay otros métodos para estimarlo o calcularlo, tomando en cuanta a veces la última modificación de la respuesta por ejemplo.

Mecanismo de Validación.- cuando el tiempo de expiración calculado es demasiado grande, puede ocasionar que la respuesta no sea totalmente valida. Con el fin de estar seguro de la validación de la respuesta, el cache debe checar con el servidor de origen o posiblemente con otro cache intermediario con una respuesta más actual. A partir de la versión 1.1 de HTTP estos métodos se implementaron.
Cuando el servidor manda una respuesta total, en esta se adjunta una pequeña validación dentro del mensaje. Esta validación debe guardarse sobre un cache que utilizara el cliente en sus próximos requerimientos condicionales. El servidor analiza el requerimiento y responde con un código especial (304) y no con un cuerpo de entidad (mensaje completo).

Lenguaje HTML (Hypertext Markup Language).- Se compone de tags entendibles tanto por los browsers como para el servidor de Web, los tags describen elementos básicos del documentos de Web, tales como encabezados, párrafos, estilos de texto y listas, así como elementos más sofisticados como las formas, scripts o Java Applets.
Como los tags son elementos independientes, una pagina HTML puede enviarse a cualquier tipo de plataforma; además de que cada tag puede contener ligas a otros documentos localizados ya sea en el mismo equipo o en otro incluso dentro de otra red.

A través de la evolución de la Internet, se desarrollan lenguajes de programación con un enfoque distributivo, es decir que se trata de generar código portable en cada plataforma existente. El lenguaje Java es el más común dentro de las aplicaciones en Internet, su gran ventaja es precisamente su portabilidad, que radica en lo que se denomina la máquina virtual Java, que no es otra cosa que un software montado arriba del hardware del equipo y además arriba del sistema operativo. Esta máquina virtual es capaz de correr o en su caso de interpretar los programas compilados de Java. Esto provoca una arquitectura neutral y una independencia de la plataforma en que se corre los programas.

Correo Electrónico.

El correo electrónico es quizá una de las aplicaciones del TCP/IP más comunes en el mundo de las redes; entre los protocolos más conocidos de correo electrónico mencionamos el SMTP (Simple Mail Transfer Protocol) que provee del intercambio de mensajes entre diferentes hosts con TCP/IP.

Protocolo SMTP.

El SMTP trabaja con una conexión entre el cliente y el servidor directamente, una vez establecida la conexión el cliente guarda el correo hasta que éste es copiado completamente en el recipiente de SMTP, no como otro tipo de sistemas de correo donde el correo pasa de un host a otro hasta encontrar el destinatario. Hablando de una red mucho

muy grande, resulta impractico que una máquina intente establecer contacto directo con otra, en estos casos se implementaron aplicaciones llamadas "mail gateways" o "mail bridges" que no son sino sistemas de correo SMTP.

A continuación presento un esquema que representa el modelo de comunicación con SMTP.

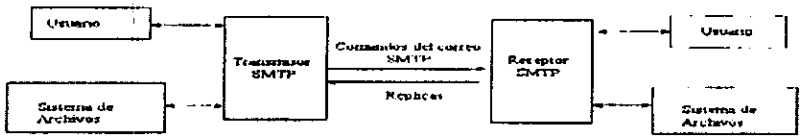


Figura 5.6.

1.- El transmisor establece contacto con el destinatario que puede ser el host de destino final o un intermediario (mail gateway) a través de TCP, y espera la respuesta del servidor que puede ser un código 220 de servicio de mensajes listo o un 421 servicio no disponible cuando el destino no contesta.

2.- El transmisor manda un mensaje de HELLO (hola) con el objetivo que el receptor lo identifique y mande como respuesta el nombre del dominio, mismo que el transmisor ocupará para verificar si en verdad se trata del destino requerido. Si el transmisor soporta Service Extensions , que no es otra cosa que servicios extendidos del SMTP, el comando de HELO es sustituido por EHLO, si es que su receptor también lo soporta contestará con un código de 250 de aceptación, de lo contrario contestará con 500, código de error de sintaxis, por lo cual el transmisor deberá intentar con el comando normal de HELO, o en dado caso con un comando de cierre de sesión QUIT.

3.-El transmisor comienza una transacción del correo enviando un comando MAIL, este comando contiene una dirección de retorno para posibles errores, acompañada de las rutas explicitas. Si el receptor acepta este comando contesta con el código 250.

4.- Posteriormente, se da a conocer el recipiente destino al servidor de SNTP, que pueden ser más de uno, esto se hace mediante el comando RCPT TO <direcciones>, cada uno de estos comandos debe tener una replica 250 si se conoce el recipiente o 550 si el usuario no existe.

5.-Cuando se termina de enviar los comandos anteriores para otorgar los recipientes al servidor, entonces el transmisor manda una orden de DATA que le dice al receptor que a partir de entonces comienza el contenido del mensaje. El servidor envía un mensaje de 354 para indicar la recepción del comando DATA.

6.- El cliente envía linea por linea los datos, terminando estos con un secuencia de 5 caracteres <CRLF>. Esta última secuencia es respondida con 250 por el servidor o con un código de error si pasa algo inesperado.

7.- De aquí hay varias situaciones posibles:

Si el transmisor o cliente no tiene más mensajes que enviar, debe terminar la sesión con un comando QUIT, y el servidor contestará con un código de 221 que indica que se esta cerrando la conexión.

Si el transmisor o cliente no tiene más mensajes que enviar pero esta listo para recibir mensajes, para ello manda un comando TURN que indica que los dos programas SMTP deben cambiar de roll el emisor se convierte en receptor y viceversa.

Si el emisor tiene otro correo que enviar, entonces se repite el paso número tres.

La tabla siguiente muestra lo explicado en el párrafo anterior.

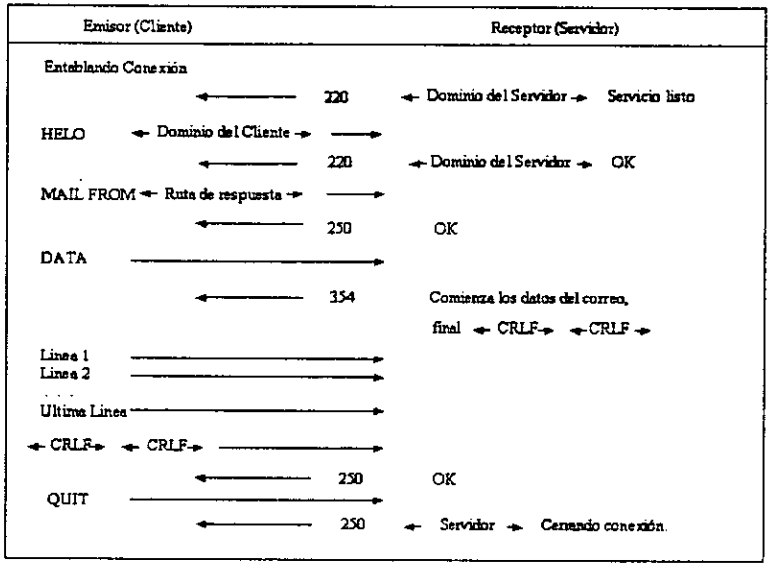


Figura 5.7.

La forma general para las direcciones de correo electrónico es:

<Parte local>@<nombre del dominio>.

Y puede tener varias variantes:

1.- .- se utiliza para una dirección de destino sobre el misma red TCP/IP.

2.-Usuario% host gateway .- para un usuario sobre un host remoto sin conexión SMTP, con una ruta a través de gateway.

3.-@host b:usuario@host c .-esta dirección trae explicita la trayectoria a través de los host. Es decir el mensaje pasa primero por el host "a", posteriormente al "b", hasta llegar al "c".

Protocolo MIME (Multipurpose Internet Mail Extensions).

Debido a las limitantes del protocolo anteriormente descrito SMTP, sobre todo que:

1.-El texto es limitado a 1000 caracteres de longitud y tiene un formato de translación de 7 bits ASCII:

2.-Que no puede transmitir directamente objetos binarios o executables, para realizarlo se vale de métodos de encapsulamiento por ejemplo el codificar el objeto como puro hexadecimal, o incluso dentro del sistema operativo UNIX existe utilerias Uuencode y Uudecode para encapsular los objetos para darle la vuelta a la limitación de 7 bits de transportación del SMTP.

3.- Hay caracteres del lenguaje que no pueden transmitir sin codificarlos.

4.- Los servidores de SMTP o gateways que trasladan la información del ASCII a EBCDIC o viceversa quizá no tiene una consistencia en sus paginas de codificación, ocasionando problemas en la transacción.

5.- Inclusive algunas versiones del SMTP no respetan los estándares del mismo.

Todo lo anterior dio origen al protocolo MIME. Las ventajas de dicho protocolo son su compatibilidad con los estándares existentes; a través de su uso se ha hecho un protocolo muy robusto, y además es fácil de extender, es decir dentro de su estructura se tiene siete estándares de tipos de contenido, cada uno de los cuales tiene subtipos, pero además define mecanismos para registrar nuevos tipos.

Un mensaje con el protocolo MIME debe contener un campo de encabezado con la forma:

MIME-Version: 1.0

Como en el caso anterior, el nombre del encabezado no tiene relevancia, pero el valor del mismo si lo tiene dependiendo el tipo de encabezado.

Así tenemos los siguientes campos de encabezado:

1.-MIME-VERSION . Indica la versión del protocolo.

2.-CONTENT-TYPE. Describe como es interpretado el objeto dentro del cuerpo del mensaje; su forma es:

Content-Type : tipo/subtipo; parámetro=valor:parámetro=valor.

Claro esta que si un tipo o subtipo no tiene parámetros, la segunda parte de la sentencia no tiene sentido. Existen siete tipos estandares:

A.-Text. Con un subtipo definido PLAIN, texto sin formato para este tipo se tiene que especificar la tabla de caracteres utilizada, en este caso a través de un parámetro "charset" cuyos valores pueden ser:

us-ascii que define la tabla de caracteres ASCII en el rango de 0 a 127.
iso-8859-x donde x esta en el rango de 1 a 9 para las diferentes partes del ISO-8859.

Otros tipos de subtipos pueden ser adicionados para describir otros formatos de texto como el texto enriquecido utilizado por WORD. El valor por defauld es text/plain; charset=us-ascii, que indica que se trata de un texto plano en código ASCII.

B.-Multipart .El cuerpo del mensaje contiene diferentes objetos con diferente tipo de datos; cada objeto es dividido por una línea específica definida en el mismo encabezado. Por ejemplo:

Content-Type: multipart/mixed; boundary="1995021309105517"

Donde la palabra "boundary" indica la línea de separación. Este tipo tiene 4 subtipos:

Mixed .- los objetos aunque son diferentes, deben ser transmitidos juntos, y en recipiente de recepción deben aparecer en el mismo orden que aparecen en el correo.

Parallel .- difiere del anterior en que el orden de los objetos esta adscrito a la parte, y en dado caso el programa receptor puede incluso presentarlos en paralelo.

Alternative .- Las diferentes partes son versiones alternativas de la misma información, el encargado de entregar la mejor versión es el sistema de correo del recipiente.

Digest .- es una variante del multipart/mixed donde el default de tipo y subtipo es message/rfc822.

C.- Message. El cuerpo en si es un mensaje encapsulado. Se definen tres subtipos:

rfc822.- como su nombre lo indica sigue un estandar RFC 822 . El RFC 822 (Request for Comments) es el formato de mensajes de texto desarrollado por ARPANET.

Partial.- Se utiliza para permitir la fragmentación de un mensaje demasiado extenso. Esto debido al limitante en el tamaño de en un agente SMTP. Para el recipiente destino esta fragmentación debe ser transparente ya que el mismo agente receptor se encarga de ensamblar nuevamente las porciones del mensaje.
Existen tres parámetros para este tipo de subtipo:

Id.- Un identificador único y común para todas las partes del mensaje.

Numer.- Identifica cada parte del mensaje con un consecutivo, la primera de las partes tendra el numero uno.

Total.-indica el total de partes en que se fragmento el mensaje, este parámetro es opcional para todas las partes, exceptuando la última que siempre debe tener este parámetro identificando con ello que se trata efectivamente del final del correo.

External-body.- Este tipo contiene un puntero a un objeto externo que existe por si solo , es decir no esta dentro del mensaje mismo del correo. Para ello este tipo tiene un parámetro que identifica el tipo de acceso con que el lector del correo tratará de sincronizarse con el objeto externo; teniendo cinco tipos de acceso:

ftp : El recipiente debe tener la configuración necesaria (login y password) para tener acceso al objeto al que apunta el mensaje. Por razones de seguridad estos datos no se transmiten en el mensaje mismo.

Tftp.- Es un puntero a un acceso a través del protocolo Trivial Transfer Protocol, que es una versión mejorada del ftp estandar.

Anon-ftp acceso ftp con el usuario anónimo.

Local-file.- el dato esta contenido en un archivo de acceso directo vía el sistema local de archivos.

Mail-server.-el dato es accesible vía un servidor de correo electrónico. A diferencia de los demás, este acceso es necesariamente asincrónico.

D.- Image. El cuerpo contiene datos de una imagen, para recibirlo se debe contar con un programa que pueda desplegar o en su caso imprimir imágenes. Los subtipos definidos aquí son:

jpeg.- la imagen tiene el formato JPEG, codificado JFIF.

Gif.- formato GIF.

E.- Video. El cuerpo contiene datos de imágenes en movimiento y posiblemente sincronizadas con datos de audio. Para su recepción se necesita de un reproductor de multimedia o una terminal inteligente. El subtipo básico es:

mpeg .- que identifica el formato MPEG.

F.- Audio. El cuerpo contiene datos de audio utilizando bocinas y una tarjeta de sonido para recibirlos.

G.- Application. Este tipo identifica para aquellos correos o cuerpos de correo que no caen dentro de las demás categorías; y particularmente para datos que deben ser procesados por otra aplicación antes de ser transferidas al usuario. Dos tipos de subtipos son definidos en esta categoría:

Postscript Adobe Systems PostScript.- es un formato de datos para impresora, un lenguaje de programación generalmente para impresora y el uso de un interprete para este formato se ocuparían para este subtipo.

Octet-stream.- indica generalmente datos binarios de 8 bits (un byte).

3.-CONTENT-TRANSFER-ENCODING.-indica como fue codificado el objeto dentro del cuerpo del mensaje. Como se vio en el apartado anterior, existen varias formas de objetos que se pueden transmitir a través del MIME, pero existen dos formas de codificarlos dentro del mensaje de correo. El Ontent-Transfer-Encoding puede tener cinco valores, dos de los cuales indican el tipo de codificación que lleva el cuerpo del mensaje:

A.- 7 bit Encoding. Es el valor por default, indica que no existe codificación en el cuerpo del mensaje, y que este consiste de líneas de texto ASCII con longitud no mayor a 1000 caracteres. Con este tipo de transmisión no se garantiza el paso integro del correo sobre todo sobre gatways que se basen en otro tipo de codificación (EBCDIC).

B.- 8 bit Encoding. Implica que tampoco existe una codificación en el cuerpo del mensaje, y que las líneas de éste son lo suficientemente pequeñas para permitir el paso a través del SMTP, sin embargo no deben ser precisamente texto ASCII, toda vez que son octetos de bits.

C.- Binary Encoding. EL cuerpo del mensaje no es texto ASCII y es mucho más largo que la longitud permitida por SMTP, por lo que este de transmisión se debe utilizar con otro tipo de mecanismo de transporte, o con servicios extendidos de SMTP que soporten una longitud más alta. Normalmente este tipo de transporte se utiliza con el protocolo TCP/IP para transmisiones vía internet o en redes basadas en TCP/IP.

D.- Quoted-Printable Encoding. Este es el primero de los dos tipos de codificación reales de los que se hablaron al inicio de este apartado. Intenta dejar archivos de texto legibles en su forma codificada. En estos archivos los caractares no permitidos en el correo por la representación hexadecimal de su código ASCII, además introduce retornos de línea para conservar la longitud de las mismas a 76 caracteres o menos..

El Quoted-Printable Enconding usa el signo de igual como un carácter para indicar ambos casos anteriormente mencionados. Además debe seguir las siguientes reglas:

1.-Cualquier carácter, exceptuando uno que es parte de la secuencia de una nueva línea (X'0d0A) puede ser representado por: =XX, donde XX es un digito hexadecimal. Si ninguna de las otras reglas se aplica el carácter debe representarse de esa forma.

2.- cualquier carácter en el rango X'21' al X'7E' excepto X'#D' que es el signo "=", puede representarse con su código ASCII.

3.- El TAB ASCII X'09' y el APACE X'20', pueden representarse como su código ASCII excepto cuando este es el último carácter de la línea.

4.- Un salto de línea debe representar se por un <CRLF> con secuencia X'0D0A'. Cuando el dato binario codificado X'0D0A no es un rompimiento de línea y debe ser codificado, esto se debe llevar a cabo con forme la regla número uno, es decir =0D=0A.

5.-Las líneas codificadas no deben ser más grandes que 76 caracteres, excluyendo el <CRLF>. Si una línea es mayor que esa longitud, un salto de línea debe ser insertada en la columna 75.

E.- Base64 Encoding. este tipo de codificación se implementa para datos que no consisten principalmente de caracteres de texto.El Quoted-Printable Enconding reemplaza cada carácter no imprimible con una secuencia de tres bytes , lo cual es inificiente para datos binarios. La codificación en Base64 trabaja tratando la entrada de datos como una secuencia de bits, reagrupándolos dentro de bytes ordenados y trasladándolos a caracteres conocidos y permitidos por el correo electrónico. Como solo existen 73 caracteres permitidos dentro del correo, la máxima longitud usada es de 6 bits que representan 64 únicos caracteres

La siguiente figura representa este traslado:

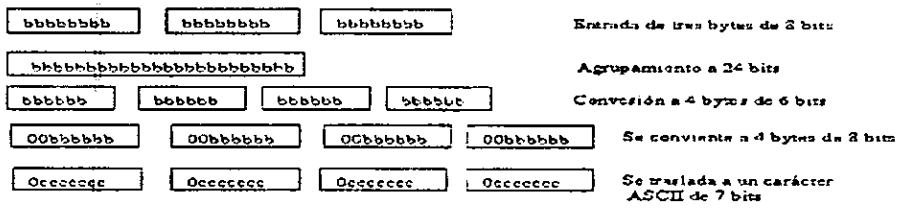


Figura 5.8.

4.-CONTENT-DESCRIPTION .- es la descripción en texto plano del objeto dentro del mensaje, es muy útil cuando el objeto no es transformable por ejemplo datos de audio.

5.-CONTENT-ID .- una palabra única para especificar el contenido de una parte del mensaje.

Selección del Servidor.

Toda la anterior teoría se repasó dentro del estudio de factibilidad para implementar uno de los servidores mencionados, ya que inicialmente se planeaba subir un servidor a un sector de red con salida a INTERNET.

El Servidor de FTP que realmente era la opción mas viable para nuestro objetivo de transferencia de archivos de uno a otro lugar, se tuvo que desechar a consecuencia de las restricciones que en ese entonces tenia la propia Dirección de Administración de Bases de Datos. La planeación sugería un servidor FTP con seguridad a través de login y Pasword de entrada, incluso también restringir el acceso a nivel del IP de la máquina del usuario.

Las aplicaciones sobre el servidor de WEB seguían en la lista de posibilidades, se diseñaron aplicaciones sobre la INTRANET de la institución, pero no se logró la autorización para dejar un servidor nuestro en INTERNET.

Lo anterior nos llevó a la selección de utilizar el servidor de correo electrónico existente como medio de comunicación entre las contralorías y la DGACS. La administración es este caso solo se da en las cuentas asignadas por el administrador del mismo servidor, dejando fuera del sistema todo lo relativo al paso de la información de uno a otro lado y diseñando una política de acuses de recibido dentro del mismo sistema. Este tipo de comunicación es la que hasta la fecha se tiene dentro del sistema, sin embargo la misma estructura del sistema propone una evolución hacia formas mas actuales, incluso se puede llegar a establecer una aplicación cliente servidor donde los clientes serían las Contralorías Internas y el servidor estaría dentro de la DABD. Creando aplicaciones que corran en los browser de cada entidad que realicen conexiones al servidor de WEB y este a su vez realice peticiones al servidor de base de datos, logrando con ello eliminar las funciones de transmisión y recepción que se tienen en los dos módulos del sistema y teniendo por lo tanto actualizaciones en línea.

La teoría correspondiente a los sistemas Cliente / Servidor se presenta en el siguiente capitulo, donde también mencionó algunas características propias del lenguaje de programación utilizado en el desarrollo de este sistema, el Delphi 5 Interprise.

CAPITULO VI.- APLICACIÓN CLIENTE SERVIDOR.

Típicamente, nos referimos a una aplicación cliente-servidor como aquel sistema que guarda en su estructura dos módulos lógicos funcionales claramente divididos.
Cliente.- dedicado a la interfaz del usuario final, teóricamente no interviene de manera directa con las rutinas de almacenamiento y mantenimiento de la información dentro de nuestra base de datos.
Servidor.-contiene las rutinas de manejo de requerimientos y actualizaciones de los clientes, esta aplicación mantiene contacto directo con nuestro servidor de base de datos; de hecho este módulo se encarga de las conexiones con la base de datos y de las políticas de seguridad de acceso.

Los eventos elementales que sigue una aplicación cliente servidor se describen a continuación.

1.- El usuario final corre su aplicación cliente, ésta trata de establecer una comunicación con la aplicación servidor especificada ya sea en tiempo de diseño o en tiempo de ejecución. Si el servidor esta corriendo manda un identificador de conexión al cliente, si no el cliente debe poder inicializar al servidor y realizar la conexión.

2.- El cliente requiere información del servidor. Para ello debe de requerir todos los datos necesarios o posibles.

3.-El servidor recibe el requerimiento, empaqueta la información y devuelve este paquete al cliente, obviamente el paquete contiene información extra por ejemplo los constraints impuestos por la base de datos.

4.- El cliente decodifica la información y la muestra al usuario dentro de su interfaz diseñada.

5.- La información manipulada por el cliente, se guarda y se va actualizando sobre archivos logs dentro de la aplicación cliente, este tipo de almacenamiento, puede incluso llevarse a cabo dentro de una pequeña base de datos intermedia.

6.- Eventualmente, y en respuesta a una acción u orden del usuario estas modificaciones se actualizan en la aplicación servidor; para ello el cliente empaqueta los archivos logs y los envía a la aplicación servidor.

7.-El servidor decodifica el paquete y trata de actualizarlos en la base de datos en una transacción, si esta no es posible debido a algún tipo de error, entonces puede salvar los registros no actualizados y darlos a conocer al cliente.

8.- El cliente vuelve a recibir los registros no actualizados y los manipula para que en la siguiente comunicación con el servidor no exista ningún problema para actualizarlos en la base de datos.

9.- El cliente vuelve a refrescar los datos con una petición al servidor.

En Delphi 5 la tecnología más empleada en la creación de aplicaciones cliente-servidor, es la Multi-tier Distributed

Application Services (MIDAS), que proporciona los mecanismos necesarios para la creación tanto de aplicaciones cliente como aplicaciones servidor.

La estructura de una aplicación cliente, sigue las reglas de lo que es una aplicación de base de datos standalone, donde las actualizaciones se llevan a cabo directamente en las tablas (en este caso archivos logs o tablas intermedias), y cuentan con componentes de conexión a una aplicación servidor a través de diferentes protocolos de comunicación.

Componentes	Protocolo Utilizado
TDCOMConnection	DCOM
TsocketConnection	Windows Sockets (TCP/IP)
TwebConnection	HTTP
TOLEnterpriseConnection	OLEnterprise (RPCs)
TcorbaConnectión	CORBA(IIOP)

La estructura de una aplicación servidor incluye un módulo de acceso a datos remotos, que provee a la aplicación de un interfase de enlace con las aplicaciones cliente (IappServer). En Delphi existen tres tipos de módulos para acceder a datos remotos.

1.- TRemoteDataModule.- este componente soporta conexiones de aplicaciones cliente con protocolos DCOM, HTTP, sockets o OLEnterprise.
2.- TMTSDataModule.- soporta los mismos protocolos del anterior componente, con la variación de que este tipo de componentes genera una aplicación servidor como una librería dinámica (.DLL), en realidad este componente soporta el ambiente de Microsoft Transaction Server , que maneja diferentes servicios en las transacciones como el manejo de los recursos del sistema, implementación de reglas de conexión, etc.
3.- TcorbaDataModule.- Este componente es propio de clientes con protocolo CORBA.

En una aplicación servidor además existen componentes para cada conjunto de datos (dataset provider) que están disponibles al cliente. Un componente de este estilo tiene las siguientes funciones.

1.- Recibe los requerimientos del cliente. obtiene la información del servidor de base de datos. la empaqueta y la envía al cliente.
2.- Recibe las actualizaciones del cliente, trata de realizarlas en el servidor y el resultado lo envía nuevamente al cliente.
Para poder establecer que tipo de protocolo utilizar en una aplicación cliente servidor, primero debemos ver que tipo de beneficios trae uno u otro, y hacer un análisis con respecto al tipo de aplicación.

Conexiones DCOM (Distributed Component Object Model).-el COM y DCOM son tecnologías desarrolladas por

Microsoft para enlazar objetos dentro de una estructura binaria, el primero para correr en una sola máquina y el segundo a través de redes heterogéneas. Para conexiones cliente-servidor, dentro del servidor no necesitaríamos ninguna aplicación extra para el enlace entre uno y otro. Además como es totalmente compatible con el ambiente Microsoft Transaction Server (MTS), se puede hacer uso de las ventajas de éste.

Conexiones Sockets.- el DCOM y el MTS son productos directos de Microsoft, y no sabemos si realmente una "X" maquina lo soporte, caso contrario TCP/IP son protocolos que generalmente se incluyen dentro de cualquier equipo de computación, lo anterior nos lleva a concluir que los sockets de conexión TCP/IP son más generalizados.
A diferencia de una conexión con DCOM, en una conexión con sockets la aplicación servidor debe correr un programa extra ScktSrvr.exe, que se encarga de las llamadas del cliente e inicializa el modulo de acceso de datos remotos usando la tecnología COM, es decir tenemos por un lado que la comunicación se lleva a cabo por medio de TCP/IP, y una ves enlazada la comunicación, la comunicación interna dentro del servidor se realiza con COM. Otra desventaja de este tipo de conexiones es que no existe protección implementada en el servidor para fallas en los sistemas cliente, como existe en las conexiones DCOM.

Conexiones WEB.-El protocolo para este tipo de conexiones es el http, que permite crear aplicaciones cliente que pueden comunicarse con servidores protegidos por un "firewall". En este tipo de conexiones la aplicación servidor corre un servicio de WEB, normalmente httpsrvr.dll quien es el encargado de recibir y manipular los requerimientos de las aplicaciones cliente.

Conexiones CORBA.- La gran ventaja del CORBA es que es multiplataformas, y pueden encontrarse productos con esta tecnología corriendo sobre diferentes sistemas operativos.

En anterior texto tiene como objetivo dejar en claro el ambiente de una aplicación cliente servidor dentro del lenguaje utilizado por nosotros (Delphi 5 interprise); obviamente para obtener más información con respecto a aplicaciones cliente servidor es necesario consultar la bibliografía que al final de este documento se presenta.

Dentro del lenguaje de programación, en la Dirección de administración de base de datos se ha tenido poca experiencia en la creación de aplicaciones cliente servidor, incluso puedo mencionar que se han levantado dos sistemas con estas características utilizando conexiones DCOM. Estas aplicaciones son de carácter interno, es decir solo trabajan dentro de la Red Institucional de la institución.

El sistema que nos compete es un poco más complicado debido a la distribución de un módulo a oficinas fuera de nuestro control. Esta característica nos obliga a tener la necesidad de controlar el acceso por lo menos de un servidor que esté dentro de Internet; este control realmente no lo poseemos, y por lo consiguiente la aplicación cliente servidor solo se planteó teóricamente, dejando únicamente la tendencia a un desarrollo futuro.

En realidad el módulo del informante como lo hemos venido llamando, es fácil de transportar a una aplicación cliente, cambiando algunos componentes y desarrollando rutinas de conexión a un servidor remoto. En cuanto a la

aplicación servidor, realmente se necesitaría montar sobre un servidor dentro de una zona desmilitarizada, dejando acceso únicamente a usuarios autorizados y de acuerdo a las políticas de seguridad de nuestra institución. Sin embargo este tipo de sistemas aún no son tan posibles dentro de nuestra dirección, debido precisamente a la falta de control de la RED.

Otra idea como se mencionó en los últimos párrafos del capitulo anterior es desechar completamente la programación existente en el módulo del informante y desarrollar aplicaciones que corran sobre los browser's existentes, sin embargo en esto también se necesita tener el servidor dentro de INTERNET, meta que hasta la fecha no se ha logrado obtener.

Todo lo anterior nos limita a publicar paginas estáticas en la WEB donde podemos colocar algunos archivos de instalación o actualización de un sistema. Con respecto a la comunicación entre el informante y el módulo central nos limitamos a hacer uso de correos electrónicos para evitar demoras en la entrega de información, teniendo cuidado de encriptar la información antes de enviarla por correo.

Con este capítulo se concluye el presente trabajo, la documentación completa del Sistema de Información Periódica y Programa Anual de Control de Auditorías, se encuentra en los archivos de las DABD dentro de la Secretaría de la Contraloría y Desarrollo Administrativo. Las conclusiones obtenidas de él se mencionan a continuación.

CONCLUSIONES:

El desarrollo de un sistema automatizado para la captura, envió y recepción de información de auditorias facilitó el manejo de esta información y la estandarizó, redujo los tiempos de demoras en los procesos de análisis, liberó a los analistas de la tarea de recapturar información automatizando los procesos de recepción, y por ende redujo el tiempo entre la captura de la información y la generación de la evaluación periódica que se les hace a cada entidad.

Tener estructurada la información en el módulo central permitió la generación de reportes estadísticos más confiables y más rápidos, mismos que sirven de referencia para hacer todo tipo de indicaciones a las contralorías internas.

Con respecto al desarrollo del sistema se puede concluir que mientras se siga una metodología bien definida y sobre todo orientada a la tecnología empleada como lo es el Objectory Process, se pueden obtener resultados óptimos y eficientes.

Consecuentemente se llega a la conclusión de que el desarrollo del Sistema de Información Periódica y Programa Anual de Control de Auditorías se llevó con éxito y cumplió todos los requerimientos iniciales, dando ventajas que no se tenían anteriormente en el manejo de la información, y permitiendo reducir tiempos y procesos inecesarios.

Por último podemos mencionar que no siempre es factible implementar la tecnología más reciente, cuando no se tiene la infraestructura adecuada o por lo menos el control de ésta; y sin embargo si se puede desarrollar un buen sistema que satisfaga todos los requerimientos de los usuarios.

Todas las conclusiones anteriormente mencionadas nos dan pie a decir que se cumplió con el objetivo principal dentro del planteamiento del sistema que fue la creación de un sistema automatizado para el control de auditorías y sus seguimientos dentro de la administración pública.

BIBLIOGRAFÍA.

'CP/IP Tutorial and Technical Overview
Martín W. Murhammer, Orcun Atakan, Stefan Bretz,
Larry R. Pugh, Kazunari Suzuki, David H. Wood
BM. Oct. 1998.

Criptografía y Seguridad en Computadores.
Segunda Edición.
Manuel José Lucena López
Departamento de Informática,
Escuela Politécnica Superior
Universidad de Jaén. Septiembre 1999.

PGP(tm) User's Guide
Volume II: Special Topics
Philip Zimmermann
Revised 11 Oct 94.

Curso: Desarrollo de Sistemas
Paradigma Java
Dr. Gabriel Guerrero
SECODAM. Oct. 1999

Developer's Guide
Borland Delphi 5
Fro Windows 98, Windows 95, & Windows NT
Interprise Corporation.
Copyright 1983-1999

Informix-CLI, Programmer's Manual
Informix Press.
Ver 2.5 Ago. 1996

Informix-Online, Workgroup Server
Administrator's Guide, Volume 1
For Windows NT.
Informix Press. ,April 1996

Análisis y Diseño Orientado a Objetos
Usando Notación UML
Patricio Letelier Torres, Pedro Sánchez Palma
Universidad Politécnica de Valencia. Jun 2000.

Análisis y Diseño de Sistemas de Información.
James A. Senn
Mc. Graw Hill. Ene 88.