

64

25gen



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistema Automatizado de Averiguaciones Previas

TESIS PROFESIONAL

Para obtener el Título de Ingeniero en Computación

presentan

Mauricio Orozco Martínez
Rolando Tamayo Valenzuela

Director de la Tesis
Ing. Héctor Campos Estrada

Codirector de la Tesis
Ing. Domingo Palao Muñoz



México D. F.

1994

FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A nuestros padres

Agradecimientos

Quisiéramos agradecer al Dr. Alvaro Bunster, al Lic. Bernardo Alcántara y al Dr. Rolando Tamayo quienes nos brindaron su valiosa ayuda durante el desarrollo del presente trabajo. Agradecemos también el apoyo brindado por el Ing. Sergio Matute, la Lic. Leticia Briasco y el Ing. Héctor Campos, que permitieron la realización de este proyecto.

Descripción de Capítulos

Capítulo I Introducción

En este capítulo se plantean los objetivos y metas de la tesis, así como las necesidades y razones que determinan la creación de un sistema de Averiguaciones Previas.

Capítulo II Metodología

Se plantea la metodología a utilizar a lo largo del sistema, tanto en el Análisis, Diseño e Implementación, así como las razones de dicha elección, observando sus ventajas en relación a otras metodologías. Se plantearán a su vez los diferentes esquemas y modelos utilizados en todas las etapas de su ciclo de vida. Además se explican todas las herramientas y modelos gráficos que se utilizan en capítulos posteriores.

Capítulo III Análisis del Sistema

Es el comienzo formal del Análisis del sistema, se empleará la Metodología del Análisis, explicada en el capítulo II, enfocada propiamente al sistema, se utilizarán la herramientas gráficas antes explicadas y se obtiene un Modelo Esencial del sistema que debe satisfacer las necesidades de los usuarios. Se describe también la función correspondiente a la averiguación previa dentro del aparato jurídico mexicano.

Capítulo IV

Diseño del Sistema

Posterior a la etapa de análisis del sistema se debe continuar con el diseño del mismo, de igual forma que en el análisis, se seguirá la metodología de diseño que se explica en el capítulo II. Durante el diseño del sistema se obtendrá una arquitectura adecuada para la implementación del mismo y se diseñará también la base de datos que este requiere.

Capítulo V

Implementación

En este capítulo se desarrolla lo que corresponde propiamente a la implementación del sistema; la codificación de los módulos definidos durante la etapa de diseño, además de la descripción de diferentes pruebas a que se someten, y se detalla la capacitación necesaria para su correcta utilización.

Capítulo VI

Conclusiones

En este capítulo se darán las conclusiones a las que el presente trabajo nos ha conducido, así como recomendaciones para una mejor proyección del sistema.

Capítulo I

Introducción	1
---------------------------	----------

Planteamiento del problema	1
Objetivo de la tesis.....	2
Metas de la tesis.....	3

Capítulo II

Metodología.....	4
-------------------------	----------

Introducción.....	4
--------------------------	----------

Herramientas para el Análisis.....	7
---	----------

Diagramas de Flujo de Datos.....	7
---	----------

Diagrama Entidad-Relación.....	11
---------------------------------------	-----------

Diagramas de transición de Estados	15
---	-----------

Diccionario de Datos.....	17
----------------------------------	-----------

Especificación de Procesos	18
---	-----------

Balanceo del modelo	19
----------------------------------	-----------

El Análisis	21
--------------------------	-----------

Modelo Esencial.....	22
-----------------------------	-----------

Modelo Ambiental	22
-------------------------------	-----------

Modelo Conductual.....	25
-------------------------------	-----------

Características Operativas	27
---	-----------

Metodología del Diseño	27
-------------------------------------	-----------

Ideas Básicas del Diseño Estructurado	28
--	-----------

Lo que el Diseño Estructurado no es.....	28
---	-----------

Herramientas del Diseño Estructurado.....	29
La Carta de Estructura.....	29
Características de un Buen Diseño	33
Acoplamiento	33
Cohesión.....	34
Consejos Adicionales	36
Creación de las Cartas de Estructura.....	38
Análisis de Transacciones.....	39
Transformación	39
Integración.....	47
Diseño de Bases de Datos Relacionales.....	48
Modelo Relacional	48
Normalización.....	55
Consideraciones Adicionales	60
Implementación	60
Programación.....	60
Pruebas	63
Instalación.....	65
Capacitación	65

Capítulo III

Análisis del Sistema.....	66
----------------------------------	-----------

Antecedentes	66
La averiguación previa.....	66
Revisión de otros sistemas	77
Objetivos del S. A. A. P.....	79

Modelo Contextual	80
Diagrama Contextual	80
Lista de Eventos	81
Modelo Conductual	82
Diagrama de Flujo de Datos	82
Especificación de procesos	92
Diagrama de transición de Estados	99
Diccionario de datos	100
Diagrama Entidad-Relación	104
Características Operativas del Sistema	104

Capítulo IV

Diseño del Sistema	106
---------------------------------	-----

Cartas de Estructura	106
Miniespecificación de los Módulos	110
Diseño de la Base de Datos	115

Capítulo V

Implementación	126
-----------------------------	-----

Programación	126
Pruebas	128
Instalación	130
Capacitación	130

Capítulo VI

Conclusiones..... 133

Bibliografía..... 136

Capítulo I

Introducción

Planteamiento del problema

Una de las principales medidas a tomar para el mejoramiento en la procuración de justicia de nuestro país es el de agilizar el desarrollo de la Averiguación Previa, el cual presenta un serio rezago, produciendo día con día una gran acumulación en el volumen de trabajo. La necesidad de un Sistema Automatizado de Averiguaciones Previas se hace evidente tanto para agilizar dicho proceso como para manejar el inmenso volumen de información con el fin de dar a la Procuraduría General de la República una herramienta útil y dinámica para el mejor desempeño de sus actividades sustantivas.

En la Fig. 1.1 se muestran los porcentajes correspondientes a la eficiencia, productividad, y Averiguaciones Previas en trámite.

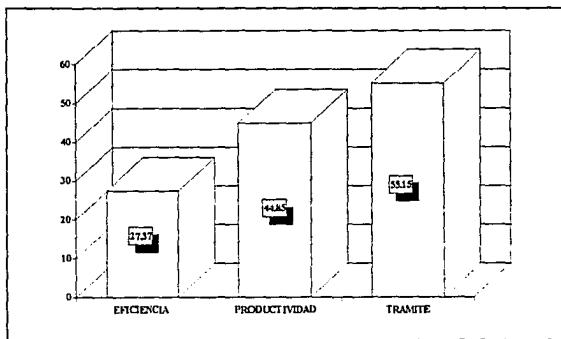


Fig. 1.1 Eficiencia, Productividad y Averiguaciones Previas en trámite.
Agosto de 1993, Procuraduría General de la República.

En la Fig. 1.2 se observa un comparativo de la eficiencia en Averiguaciones Previas entre diferentes estados para el mes de agosto de 1993, entendiéndose por eficiencia el número de Averiguaciones Previas concluidas entre el de iniciadas. Como se puede observar, existe una eficiencia promedio del 27% al mes y el 73% restante se acumula, produciendo un enorme rezago que constituye un serio obstáculo para la procuración de justicia.

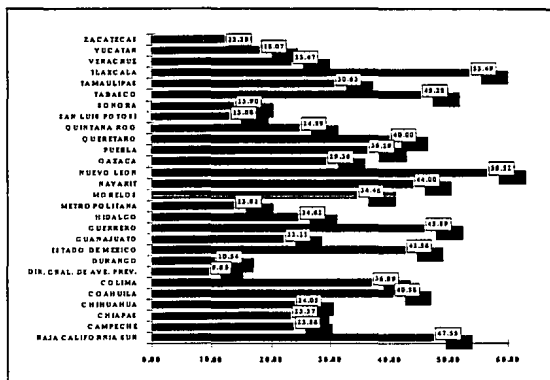


Fig. 1.2 Comparativo del porcentaje de eficiencia por delegación estatal .
Agosto de 1993, Procuraduría General de la República.

Objetivo de la Tesis

Desarrollar un sistema automatizado de Averiguaciones Previas que agilice los tramites del proceso, que facilite el manejo de información y permita la explotación estadística intensiva de los datos en caso necesario.

Metas de la Tesis

Las metas principales de esta tesis son las siguientes:

- Analizar y modelar el proceso de una A.P.
- Poner en práctica la metodología aprendida en la Universidad
- Diseñar e implementar un sistema de cómputo estable capaz de auxiliar al Ministerio Público en la generación de Averiguaciones Previas.

Capítulo II

Metodología

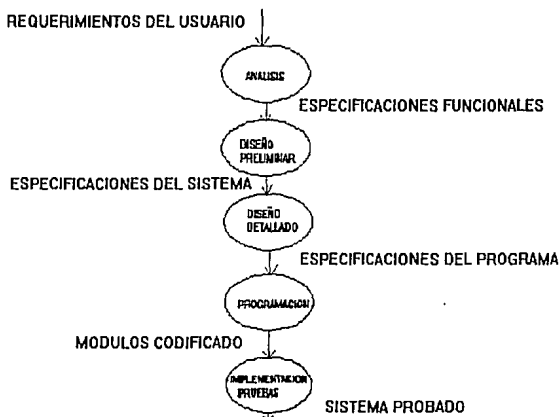
Introducción

Hasta hace poco tiempo el desarrollo de sistemas de cómputo se había manejado de manera improvisada —aunque esta situación en algunas organizaciones no ha desaparecido por completo—, sin un método capaz de controlar la gran cantidad de actividades involucradas en la realización de un proyecto de esta naturaleza, trayendo como consecuencia una crisis generalizada dentro de las organizaciones responsables del manejo de software, ya que los sistemas que se producían rebasaban por mucho el tiempo planeado para su realización y el presupuesto asignado al proyecto. Por si fuera poco, el producto terminado dejaba mucho que desear debido a fallas en su funcionamiento, a la dificultad de su mantenimiento y a que éste no cubría por completo con los requerimientos que el cliente o la organización realmente deseaban. Ante esta situación se hizo patente la necesidad de una metodología capaz de controlar todas las actividades necesarias para la creación de sistemas de cómputo eficientes y realizados de acuerdo al calendario y presupuesto estimados. El desarrollo de esta tesis será guiado por una metodología estructurada que es ampliamente conocida en el ámbito de la ingeniería de software.

En todo sistema de información (independientemente de cual se trate) se atraviesa por diferentes fases, desde que es concebido como idea, generalmente a partir de una necesidad, hasta su completa realización. Este conjunto de fases es conocido como el Ciclo de Vida de un Sistema. La existencia de diferentes metodologías hace posible que existan distintos tipos de Ciclo de Vida, por tanto, podemos afirmar que el ciclo de vida de un sistema de información depende de la metodología que se emplee para desarrollarlo. Sin embargo, podemos observar que todos, independientemente de la metodología que se utilice atraviesa por tres grandes etapas: algún tipo de análisis, diseño e Implementación (aunque dichas etapas pudieran tener diferentes subdivisiones). En el análisis se trabajan los requerimientos funcionales que tienen que ver con la concepción del usuario respecto al sistema; sus entradas, sus salidas, restricciones, objetivos etc. La etapa de diseño se resume en una forma ordenada, estructurada de transformar los requerimientos funcionales del Análisis en módulos

programables, generando también un modelo estable de Bases de Datos para proceder con la implementación del sistema. La etapa de implementación consiste en transformar nuestro diseño a un modelo real, en esta etapa se encuentran integradas la codificación y las pruebas del sistema.

El Ciclo de Vida Clásico se observa en la siguiente figura:



Cabe mencionar que dicho Ciclo puede tener diferentes variaciones con respecto a la figura; sin embargo, existen dos características que lo distinguen: Una fuerte tendencia secuencial progresiva, que consiste en terminar una fase antes de continuar con la siguiente; y una implementación de tipo Inductiva o "Bottom Up", donde las diferentes pruebas a módulos del sistema son efectuadas antes que las pruebas de los diferentes subsistemas y estos a su vez antes que las del sistema completo. Esto asemeja a un rompecabezas en donde se comienza a partir de piezas básicas, las cuales van construyendo diferentes subsistemas y estos a su vez constituyen el sistema o la meta deseada. Estas dos características presentan importantes desventajas ya que por un lado la tendencia secuencial progresiva imposibilita continuar con una fase posterior sin tener terminadas las anteriores. Esto puede ocasionar grandes retrasos que se presentan

con un efecto de avalancha. Otra de las desventajas consiste en que la prueba del modelo completo del sistema se presenta en las últimas etapas (si hubiera un error sería muy difícil de corregir en este punto). Este tipo de consideraciones dan nacimiento al Ciclo de Vida Semiestructurado.

En el Ciclo de Vida conocido como Semiestructurado se sustituye la implementación de tipo Inductiva con la Deductiva "Top Down", es decir; que primero son probados los conceptos esenciales del sistema y a partir de ahí se prueban los diferentes subsistemas que pudieran existir y así sucesivamente hasta que se llegue a las primitivas propias del sistema; esto es semejante a un Atlas, primero se tiene un panorama general del país y después se concentran en otros mapas las diferentes zonas de dicho país que a su vez pudieran estar divididas en otros mapas y así sucesivamente. En dicho ciclo de vida, la fase de diseño es sustituida por un diseño de tipo estructurado, es decir, un diseño en el cual se apoya en herramientas gráficas como son diagramas de flujo de datos, diagramas de entidad relación, diccionario de datos y otros cuya función tradicional sea de transformar las narrativas del usuario, generados en el análisis, en un diagrama funcional, entendiéndose por éste, un diagrama que tenga una correspondencia directa con un sistema real. En el momento en que dichas herramientas gráficas se emplean en la fase de Análisis, surge el Análisis Estructurado dando lugar al Ciclo de Vida Estructurado. Una característica fundamental en esta ciclo de vida es que las etapas tanto de Análisis, Diseño e Implementación se traslapan entre sí, no existe una delimitación definida de lo que corresponde al inicio y fin de una fase. Esto da como resultado que ciertas fase se pueden llevar paralelamente sin que sea necesario la terminación estricta de una, también esto permite una retroalimentación entre las diferentes etapas como suele suceder en la vida real.

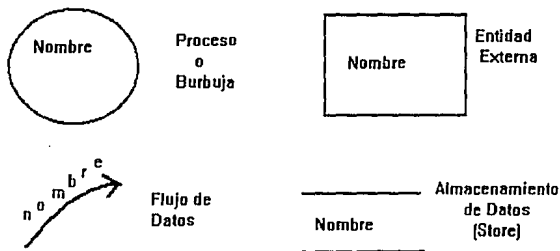
Una de las metodologías estructuradas más conocidas y aceptadas en el desarrollo de sistemas modernas es la propuesta por Edward Yourdon. Como toda metodología estructurada esta hace uso de herramientas gráficas de apoyo, tanto en su etapa de análisis como en la de diseño. Antes de explicar las diferentes etapas de esta metodología es conveniente familiarizarse con dichas herramientas ya que es muy importante que todo tipo de herramienta usada en el sistema sea explicada de manera detallada para evitar cualquier confusión que se pudiera presentar en su uso. De esta manera se asegura una consistencia en el uso de las herramientas en todo el sistema que con la ayuda de la metodología permite tener la concordancia deseada.

Herramientas para el Análisis

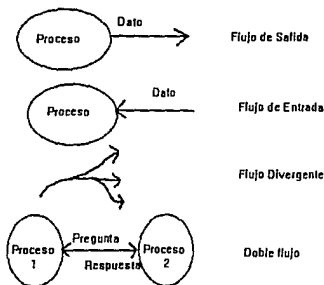
Diagrama de Flujo de Datos (DFD)

Un Diagrama de Flujo de Datos (DFD) es una de las herramientas gráficas más importantes que se utilizan en el desarrollo de sistemas. Un DFD ejemplifica las diferentes funciones desarrolladas en el sistema, muestra los diferentes procesos que se tienen, al igual que la interacción (en cuestión de flujo de datos) que existe entre dichos procesos, tanto entre sí, como con entidades externas al sistema (terminadores). Una de sus mayores ventajas es la facilidad que se tiene para comprender un DFD, ya que dicha representación es prácticamente transparente para cualquier persona. El DFD está constituido por procesos que se representan mediante burbujas, el flujo de los datos representados por flechas que van de un proceso a otro, entidades externas o terminadores representados por rectángulos y almacenes de datos temporales ("Stores") representados por rayas paralelas.

Un proceso o burbuja es la parte del sistema que transforma entradas en salidas, se representa generalmente con un círculo el cual debe tener su nombre (el nombre del proceso en la parte de adentro).

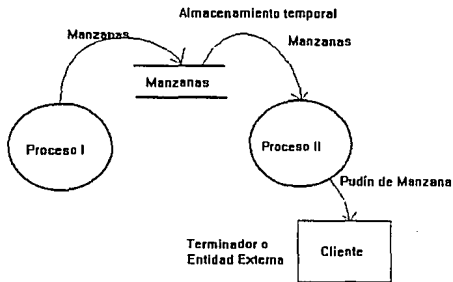


El flujo representa el camino de cierta información única para el sistema, también debe de tener el nombre especificado en su parte superior. Un flujo puede ser de salida cuando saca la información de un proceso, de entrada cuando llega el dato al proceso; puede ser divergente si se divide para diferentes caminos, de doble flujo (para simplificar dos flujos entre dos procesos)



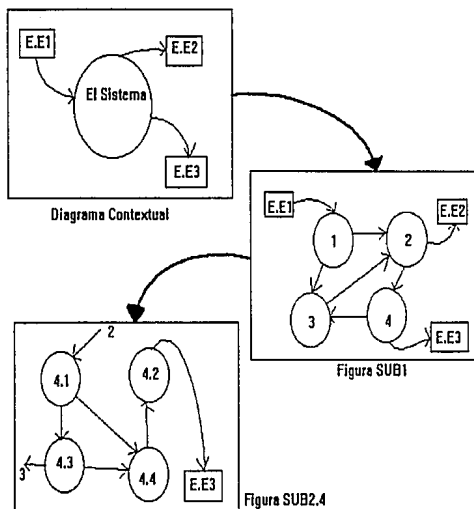
Las Entidades Externas (Terminadores) son entidades foráneas al sistema que de una u otra manera alimentan al sistema de información o bien utilizan la información procesada en el sistema. De esta manera se representa la interrelación que existe entre el sistema y el mundo exterior en cuanto al flujo de datos de información. Son representados por rectángulos y no pueden ser modificados ya que no forman parte del sistema en si.

Los Almacenes de Datos ("Stores") son representados con rayas paralelas y significan un almacén temporal de un tipo de datos, ya que no necesariamente los datos producidos por un proceso dado serán utilizados inmediatamente por otros, lo que hace necesario que existan almacenes para determinados tipos de datos mientras sean utilizados.



Es recomendable que todos los nombres que aparezcan en el DFD sean apropiados, es decir procurar que semánticamente corresponda el nombre de un proceso, de un dato, o terminador a lo que realmente corresponde de acuerdo a su función (integridad semántica), de esta manera diferentes personas podrán tener una mejor idea de lo que el sistema representa si se tienen los nombres adecuados del mismo sin tener que consultar al Diccionario de Datos del Sistema.

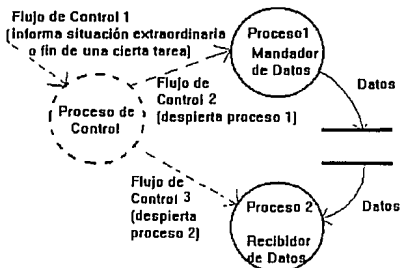
La mayoría de los sistemas en la actualidad cuentan con muchísimos procesos, y el tratar de representarlos todos en un solo DFD lo haría a este demasiado complicado para poder seguirlo. Una de las maneras de poder solucionar este problema es de englobar a varios procesos que se relacionen un uno sólo, y así sucesivamente con los procesos ya englobados hasta que el grado de complejidad sea satisfactorio. De tal manera que el DFD resultante puede dividirse en diferentes etapas y subetapas de una manera organizada como se muestra en la siguiente figura.



Es muy importante guardar la consistencia entre los diferentes DFD's de las diferentes etapas. Un DFD hijo es la representación de un sólo proceso en el DFD padre, por lo que las entradas y salidas de dicho proceso, hacia entidades foráneas del mismo (entidades externas, otros procesos o "Stores") serán las únicas representadas en el DFD hijo y ninguna otra más. El DFD hijo sin embargo, puede tener diferente interacciones entre los procesos propios de él. El número de subniveles se recomienda que no pase de diez, ya que de otra manera se haría fastidioso seguir tantos subniveles.

Existen burbujas y flujos de control representados por círculos y flechas punteadas respectivamente, un flujo de control se diferencia de los flujos normales ya que no transporta ningún tipo de dato sino que más bien da una señal binaria para despertar un proceso. Un proceso de control o burbuja de control funciona como un coordinador de las actividades de otros procesos. Los flujos de salida de estos procesos generalmente despiertan a otros procesos; los flujos de entrada informan al

proceso de control que cierta tarea a terminado o que una situación extraordinaria se ha presentado. Este tipo de burbuja se presenta en sistemas de tiempo real. Un ejemplo de este tipo de burbujas de control se observa en la siguiente figura.



Como se puede observar entonces, y pese a su nombre, el DFD no sólo ejemplifica el flujo de datos al sistema sino puede también especificar el etapas de control del mismo.

Todos los procesos en un DFD deben de tener conexión con otras entidades y es recomendable evitar procesos de sólo entradas y de sólo salidas, es decir, que no exista un proceso en si (es muy probable que si esto sucede haya un error). Además todos los nombres que aparezcan en el DFD tienen que estar referidos en el Diccionario de Datos del Sistema, de tal manera que se pueden evitar confusiones en cuanto a los datos referenciados.

Diagrama Entidad Relación

Otra de las herramientas usadas en el desarrollo de sistemas es el Diagrama Entidad Relación (DER) su principal objetivo es resaltar los tipos de relaciones que se tienen entre las diferentes entidades del sistema.. Un DER se compones de objetos y de relaciones.

Un objeto es representado por un rectángulo y representa un grupo característico que ocupa una importancia dentro del sistema. Cada objeto tiene un papel fundamental

para que el sistema funcione y generalmente tienen varios datos asociados a él (por ejemplo en un objeto llamado Cliente podría tener datos como nombre del cliente, dirección etc.).

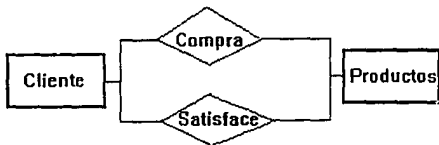


Ejemplo de objeto:

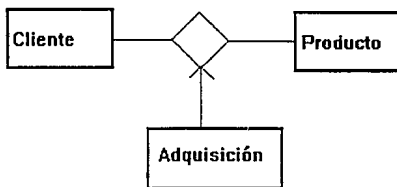
Los objetos se asocian entre si mediante relaciones, estas son representadas mediante rombos con un verbo descriptor en su interior. Cada representación gráfica de una relación significa la posibilidad de cero o más ocurrencias de esa relación en diferentes instancias. Un ejemplo de relación se observa abajo:



Dos objetos pueden estar unidos entre si por una o más relaciones



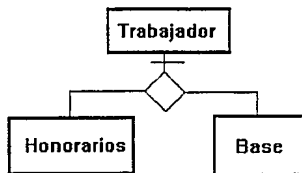
Existen objetos de tipo asociativo, estos tienen una doble función: sirven como relación de dos objetos pero a su vez también son objetos ellos mismos. Esto se usa si de la relación queremos guardar datos importantes como el momento en que se hizo etcétera.



Como se observa existe un objeto apuntando al diamante, es dicho objeto que definirá la relación de los otros dos objetos, cuando esto sucede dicho objeto almacena información referente a la relación misma, como pudiera ser hora de la relación, lugar del encuentro etcétera.

Un objeto puede tener subobjetos, entendiéndose por tales distintas clasificaciones hechas a los objetos mismos, creándose de tal manera relaciones conocidas como subtipo supertipo, donde los diferentes subtipos son nuevos objetos resultantes de la clasificación del supertipo; el supertipo será la unión de todas las características comunes de los subtipos. Un ejemplo de esto se muestra la figura:

Subtipo y Supertipo

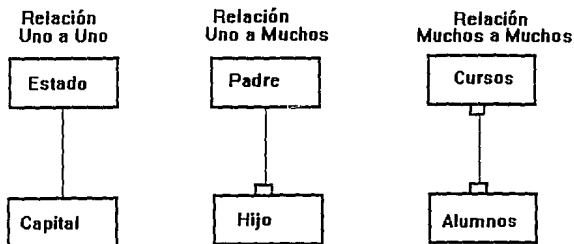


Esto sirve para resaltar que existen dos subtipos de objetos, de esta manera se deberá tener siempre en consideración que cuando tenemos una relación hacia el trabajador estamos teniendo una relación a todos los tipos de trabajadores que hay.

Una relación entre dos objetos puede ser de estos tipos:

- Una relación uno a uno
- Una relación uno a muchos
- Una relación muchos a muchos

Existen varias maneras de representar este tipo de relaciones, en el sistema se utilizo el diagrama de Banchman para ejemplificar estos caso. En Diagramas de Banchman los objetos siguen siendo rectángulos y están unidos a otros objetos mediante conexiones de líneas que simbolizan una relación. Para simbolizar una relación uno a uno basta simplemente unir los objetos con una sola línea, una relación uno a muchos se representa mediante una línea que sale del primer objeto hacia segundo, esta línea es recibida por un pequeño rectángulo adyacente al segundo objeto, de esta manera representa que el primer objeto tiene una relación de uno a muchos con el segundo. En una relación muchos a muchos ambos objetos tienen dicho rectángulo. Todo esto se ejemplifica en la figura.



Se pueden combinar ambas nomenclaturas para proporcionarnos gráficas como la siguiente, en donde se puede observar tanto el tipo de relación de acuerdo a la clasificación pasada tanto como el nombre de la relación específica como se observa a continuación.



En esta figura se puede observar que la gráfica da más información ya que podemos saber cual es la relación específica que existe de uno a muchos entre el cliente y el producto.

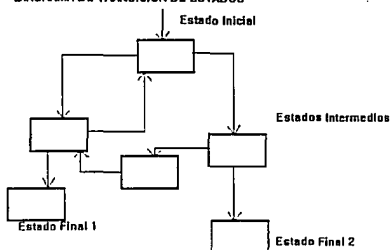
Existen relaciones que pueden hacer una entidad dependiente de otras, es decir que una entidad no puede ser concebida si no se tiene otra asociada. Un ejemplo típico de este tipo de relación es la de padre e hijos ya que no se puede concebir la idea de un hijo si no se tiene un padre asociado. La Integridad Relacional es precisamente cuidar este tipo de ligas y no tener una entidad totalmente dependiente a otra si esta otra no existe.

Diagrama de Transición de Estados

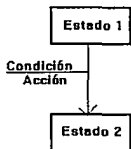
El Diagrama de Transición de Estados (DTE) muestra el sistema en función de los tiempos de ejecución de los diferentes procesos. Se resalta la secuela en tiempos de diferentes procesos. Se forman mediante diferentes estados o puntos característicos del sistema, que se manifiestan al cumplirse ciertas condiciones que pueden ser de tiempo o de secuencia (representados por rectángulos), y flechas que marcan el orden secuencial de las diferentes etapas. Cada rectángulo representa un estado posible del sistema en un tiempo dado.

Un estado inicial en el DTE representa el inicio de un proceso, él cual, conduce a otros estados del mismo proceso, hasta llegar a un estado final o un punto terminal del proceso. El estado inicial en el DTE es representado por una flecha sin conexión que desemboca a un estado, el estado final se representa mediante estados que no tienen flechas de salida. Un DTE tiene un estado inicial, uno a varios estados finales y puede o no tener diferentes estados intermedios.

DIAGRAMA DE TRANSICION DE ESTADOS



En algunos casos para que un sistema pueda pasar de un estado a otro se requiere cumplir cierta condición, para proseguir con una acción esto se representa de la siguiente manera:



Un ejemplo de condición sería que cierta variable fuera mayor que cero, la acción resultante pudiera ser la activación de cierto proceso.

Todos los nombres que se hacen referencia en un DTE tienen que estar contenidos en el Diccionario de Datos del Sistema

Diccionario de Datos

El diccionario de datos del sistema contiene la explicación de todos los datos pertinentes del sistema, absolutamente todas las variables que se utilizan en el sistema deben de estar referidas en el diccionario de datos.

Para este propósito se pueden emplear diferentes notaciones; en el sistema se usa la notación propuesta por Yourdon, la cual explicaremos a continuación:

=	se compone de
+	y
()	opcional
{ }	iteración
[]	selecciona una de varias opciones
**	comentario
@	identifica "Stores" o campos llave
	separa opciones en []
-	especifica un rango

Usando dicha notación podemos definir los siguientes ejemplos:

Ejemplo de iteraciones:

A= 1{B}	"A" está compuesto de una o más "B"
A={B}10	"A" está compuesto de 10 o menos "B"
A=1{B}10	"A" está compuesto de 1 a 10 "B"
A= {B}	"A" está compuesto de 0 a un número indeterminado de "B"

Ejemplo compuesto:

nombre = (titulo) + nombre de pila + apellido paterno + apellido materno
titulo [Lic. | Ing. | Mtro | Dr.]
nombre de pila = 1{caracteres}25 ** se quiere que sea entre 1 y 25 caracteres
apellido paterno 1{caracteres}25
apellido materno=1{caracteres}25
caracteres= [A-Z| a-z| 0-9| ' | á | é | í | ó | ú | ü | |]

En este ejemplo se define que la variable nombre este formada por un título, que puede ser opcional, y un nombre de pila (combinación de 1 a 25 caracteres) , apellido paterno (combinación de 1 a 25 caracteres) y apellido materno (combinación de 1 a 25 caracteres); igualmente define un carácter como la opción de las letras del alfabeto tanto en mayúsculas como en minúsculas o uno de los números existentes o las vocales acentuadas o el apóstrofe o una u con diéresis o el espacio en blanco.

De una manera similar a los ejemplos mencionados, todas las variables del sistema deberán de ser presentadas. De tal forma se puede asegurar que en la programación se respeten las diferentes definiciones y se reconozcan en su totalidad todas las variables y de igual forma facilitará la continuidad del mismo desarrollo en siguientes versiones.

Especificación de Procesos

Se refiere a la manera en que un proceso convierte sus entradas en salidas, en un DFD los procesos se representan con burbujas, en la especificación de procesos cada burbuja tiene que estar especificada de acuerdo a un lenguaje estructural (pseudocódigo). Para este propósito, se utiliza sentencias típicas de la programación estructurada como son :

EMPIEZA PROCEDIMIENTO
se realiza procedimiento
FINALIZA PROCEDIMIENTO

SI condición
sentencia
DE OTRA MANERA SI
sentencia
SI NO
sentencia
FIN

MIENTRAS condición
Sentencia
FIN MIENTRAS
etc.

Cabe distinguir que un lenguaje estructurado (pseudo código) es mucho más general que un lenguaje de programación ya que en el primero se hace una mención de las condiciones para realizar o no ciertas acciones (no especifica la forma que dichas acciones tienen que hacerse) mientras en el segundo se realizan únicamente sentencias válidas definidas previamente por dicho lenguaje de programación.

Un ejemplo del uso de una especificación de procesos se ve a continuación:

Supóngase que existe un proceso llamado Llenado de Cisterna

```
MIENTRAS cisterna prendida
  SI el agua esta menor que el nivel inferior
    verifica que bomba este prendida
    SI bomba está apagada
      prender bomba
    FIN SI
  DE OTRA MANERA Si el agua esta mayor que el nivel superior
    verifica que bomba esta prendida
    SI bomba está prendida
      apagar bomba
    FIN SI
  FIN SI
FIN MIENTRAS
```

De una manera similar todos los procesos mencionados en el Diagrama de Flujo de Datos tendrán que estar explicados. Esto con el fin de familiarizar a todas las personas interesadas con respecto a las diferentes funciones de cada proceso dentro del sistema.

Modelo Balanceado

Un Modelo Balanceado no es sino una verificación entre diferentes tipos de modelos que conforman un mismo sistema, si dichos modelos concuerdan se puede decir que dicha estructura esta balanceada. Se han analizado diferentes modelos que resaltan el sistema en diferentes maneras (el DFD resalta las funciones del sistema, el DTE que resalta los diferentes Estados del sistema con respecto al tiempo etc.), estos modelos

tienen que seguir una serie de reglas entre si para guardar una concordancia lógica. Si dicha concordancia lógica no existe se tiene un error conceptual que se tiene que considerar. A continuación se muestran las reglas de concordancia que tienen que tener ciertos modelos entre si.

Balaceo de un DFD con el Diccionario de Datos

- Todos los flujos y los almacenes temporales ("Stores") deben de aparecer en el Diccionario de Datos
- Todos los elementos de datos definidos en el Diccionario de Datos deben de aparecer en el DFD

Balaceo de un DFD con la Especificación de Procesos

- Cada burbuja del DFD debe de estar asociada con una burbuja de nivel inferior o un proceso específico pero no con dos.
- Cada especificación de proceso debe tener una burbuja de nivel inferior asociada a él.
- Las entradas y salidas en una especificación de proceso tienen que coincidir con las entradas y salidas de la burbuja de nivel inferior asociada a dicho proceso.

Balaceo de la Especificación de Proceso vs DFD y Diccionario de Datos

Cada dato referenciada en la especificación de procesos debe de cumplir con lo siguiente:

- Corresponde a un flujo o a un "Store" o
- Corresponde a un término local usado en la especificación de proceso ó
- Que una variable "X" y "Y" aparecen en la especificación de procesos y no aparezcan en el DFD pero en el Diccionario de Datos aparece una variable "Z" compuesta de "X" y "Y" que si aparece en el DFD entonces concluimos que si esta balanceado.

Balanceo del Diccionario de Datos contra el DFD y la especificación de procesos

Debe cumplir:

- Cada entidad en un Diccionario de Datos debe estar referenciado en la especificación de procesos o en el DFD o en otra entrada del Diccionario de Datos.

Balanceo del DER contra el DFD y la especificación de procesos

Debe cumplir:

- Cada "Store" de un DFD debe corresponder a un objeto de un DER, o a una relación o una combinación de relación y objeto (objeto de tipo asociativo)
- Nombres de objetos y de "Store" deben de ser los mismos
- El Diccionario de Datos debe satisfacer tanto al DER como el DFD.

Balanceo del DFD contra DTE

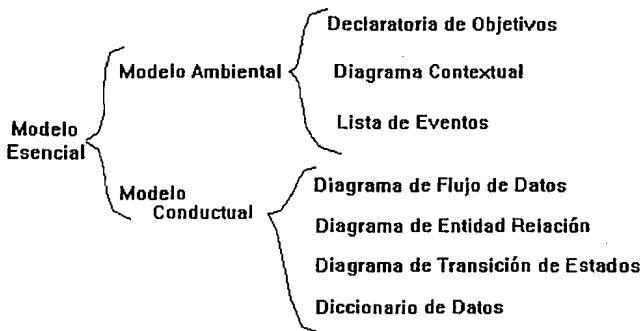
- Cada burbuja de control en un DFD tiene asociada a un DTE como especificación de proceso. De igual manera cada DTE debe estar relacionada con una Burbuja de Control
- Cada condición corresponde con un flujo de datos de entrada hacia la Burbuja de Control.
- Cada Acción corresponde a un flujo de salida de una burbuja de control.

El Análisis

De acuerdo con la metodología propuesta por Yourdon, el primer paso en el análisis de un sistema consiste en crear el Modelo Esencial del mismo. Dicho modelo marca los requerimientos esenciales para que el sistema satisfaga las necesidades del usuario. El Modelo Esencial a su vez se divide en dos grandes grupos: El Modelo Conductual ("Behavioral Model") y El Modelo Ambiental ("Environmental Model"). Ambos Modelos utilizan herramientas de apoyo para su explicación como son: Diagrama de Flujo de Datos, Diagrama Entidad Relación, Diagrama de Transición de Estados y Diccionario de Datos.

El Modelo Esencial

El modelo esencial es aquel que pretende cubrir los requerimientos primordiales del sistema desde el punto de vista del usuario. Dicho modelo es una parte crítica en el análisis del sistema y se compone de lo siguiente:



El Modelo Ambiental y el Modelo Conductual son las dos grandes divisiones del Modelo Esencial, el primero modela la parte externa del sistema, determina que entidades están fuera del sistema y cuales dentro. El Modelo Conductual describe la conducta necesaria dentro del sistema para que pueda interactuar con éxito con el medio exterior.

El Modelo Ambiental

El Modelo Ambiental define la interacción que existe entre el sistema y el medio ambiente, define el campo de actividades que conciernen al sistema dentro del medio ambiente



Muchas veces en esta etapa no existe una frontera clara entre los límites del sistema con respecto al ambiente; puede ser que no se haya decidido englobar un área específica o no. Esto produce un área gris como se ejemplifica a continuación:



En la gráfica del ejemplo se interpreta de la siguiente manera: En dicha tienda de Frutas están completamente seguros de vender manzanas y mangos; no están seguros de vender o no peras. Todas las demás frutas no las tienen contempladas en la venta. La zona gris por lo tanto corresponde a una holgura en que el sistema puede o no interactuar con el ambiente (en este caso el universo de frutas existentes).

Las herramientas usadas para la explicación del Modelo Ambiental son las siguientes:

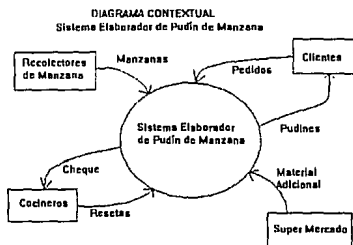
- Una Declaración de Objetivo
- Diagrama Contextual
- Lista de Eventos

La Declaración de Objetivos es una declaratoria por escrito (que no sea mayor de un párrafo) al respecto de los objetivos y razones por la que se desea hacer un sistema nuevo. Un ejemplo de dicha declaración podría ser por ejemplo la siguiente:

" En nuestra compañía queremos tener una galleta de tamaño medio que no tenga par alguno en nuestra época."

Una vez teniendo dicha declaratoria los objetivos y metas principales del sistema son resaltados.

El Diagrama Contextual es un DFD que contiene únicamente un proceso y entidades externas. Dicho proceso representa el sistema en su totalidad y las entidades externas son las que interactúan con dicho sistema. La principal función del Diagrama Contextual es resaltar la interacción que existe entre el sistema y el mundo exterior. Una representación de un diagrama de burbuja se observa en la figura.



Como se observa en la figura todo el sistema es representado con una sola burbuja. Los flujos muestran las diferentes salidas y entradas de datos que vienen del sistema hacia las entidades externas y viceversa. Visto de otra manera es el diagrama que muestra la interacción del sistema con el ambiente en cuanto a flujo de datos.

Una lista de eventos es una lista en donde se ubican los posibles estímulos que provocan una respuesta del sistema. Dicha lista posteriormente representará los procesos mismos que ejecutará el sistema. Un ejemplo se encuentra a continuación:

Cliente ordena un pedido

Cliente paga factura
Contabilidad requiere reporte diario de ventas
Almacén requiere inventario mensual

Una vez que se tenga enlistados todos los eventos que el sistema realizará se tiene terminada la Lista de Eventos

Modelo Conductual

El Modelo Conductual como se mencionó anteriormente rige la conducta interna que debe tener el sistema. En esta etapa se realizan los Diagramas de Flujos de datos, Diagramas de Entidad Relación, Diagramas de Transición de Estados, El Diccionario de Datos, y la Especificación de Procesos.

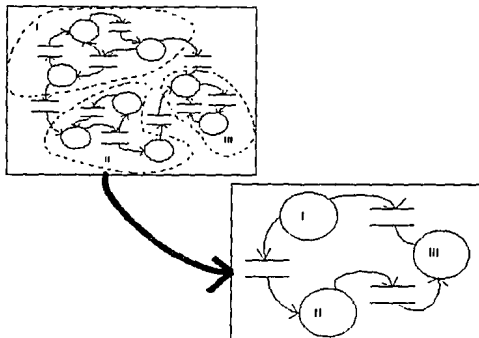
Construcción del DFD

La construcción del DFD empieza a partir de la Lista de Eventos tomando en cuenta lo siguiente:

- Por cada evento en la Lista de Evento corresponde una burbuja en el DFD
- El nombre de la burbuja corresponde a la respuesta que da el sistema a un evento
- Entradas y Salidas se dibujan como elementos de datos (flujos)

Se prosigue con la formación de las diferentes burbujas (con sus diferentes entidades externas etc.) donde cada una de ellas corresponde a un sólo proceso; esto es similar a un Diagrama Contextual diferenciándose con el hecho de que cada burbuja corresponde solamente a un evento de la lista de eventos y no a todo el sistema (todos estos procesos de nivel 0 deben de estar explicados en la especificación de procesos). Una vez habiendo construido los diferentes "Diagramas Contextuales" se ponen todos en un sólo diagrama conectándose a través de las entidades externas o "Stores" que pudieran tener en común diferentes procesos. Las burbujas no se conectan directamente entre si, sino a través de "Stores", esto es porque los procesos representan respuestas a los diferentes estímulos del ambiente, dichos estímulos son generalmente asíncronos y no podemos asegurar que dos eventos ocurran instantáneamente.

Una vez teniendo todas nuestras figuras en un diagrama de un solo nivel podemos englobar diferentes procesos en uno solo como se muestra en la figura



Ai englobar procesos tenemos que escoger aquellos que estén muy relacionados en cuestiones de datos y eventos. Es recomendable englobar aquellos que compartan un mismo "Store" por ejemplo, de esta manera en el siguiente nivel podemos prescindir de dicho "Store". La razón primordial para englobar diferentes procesos es que no se desea diagramas muy complicados, además que una persona que examina un diagrama generalmente se concentra sólo en una parte.

Prosiguiendo de la misma manera se pueden desarrollar varios diagramas de nivel superior hasta obtener el Diagrama Contextual del Sistema. En estos momentos ya se tiene el DFD deseado. A partir del DFD y de la Lista de Eventos se puede fabricar tanto el Diagrama Entidad relación como El Diagrama de Transición de Estados tomando en cuenta las reglas de balanceo que se mencionaron anteriormente. Recordaremos que la Especificación de Procesos englobara a todas las burbujas de nivel 0 correspondientes a cada evento (habrá el mismo número de especificaciones de proceso que de eventos). Todo las variables, nombres de procesos etc. estarán explicados en el Diccionario de Datos del Sistema.

Una vez que todos los modelos tanto como el Diccionario de Datos y la Especificación de procesos se encuentren balanceados obtenemos los requerimientos funcionales del sistema teniéndose por terminado el Análisis del Sistema.

Las decisiones respecto a las características de como se logran los objetivos se obtienen en el diseño del sistema.

Características Operativas del Sistema

Un paso necesario al final del análisis del sistema, es que se definan las características operativas del equipo o equipos y del sistema operativo en que será utilizado el sistema, de acuerdo a las necesidades planteadas por el mismo y a los recursos de que se dispone para implementarlo, para que la etapa de diseño se encargue de adaptar dicho sistema a las necesidades y restricciones del ambiente de operación. También será necesario elegir el software con el cual se programará el sistema.

Metodología para el Diseño del Sistema

Una vez que el análisis de un sistema ha alcanzado un progreso tal que sea posible avanzar a la siguiente etapa, se tiene un modelo esencial, en donde se plasman formalmente los requerimientos básicos del sistema que se está desarrollando; el siguiente paso es definir una arquitectura para el sistema, es decir; dividirlo en módulos funcionales y determinar la estructura que adoptará la base de datos de acuerdo con el diagrama Entidad-Relación y el diccionario de datos que arrojó la fase de análisis. Estas y muchas otras actividades se integran dentro de una etapa conocida como *diseño del sistema*.

Esta fase especifica la arquitectura que el sistema de computo tendrá, describiendo completa y formalmente los módulos que lo han de integrar, así como las interfaces que habrá entre estos, permitiendo en consecuencia el logro de una solución óptima con base en criterios de funcionalidad, costo y calidad deseada. Esto es posible gracias a que un diseño bien logrado nos conduce a la obtención un sistema compuesto de módulos bien definidos, relativamente pequeños y poco dependientes unos de otros, de tal forma que la depuración y mantenimiento del sistema en su totalidad y de los módulos por separado se vuelve más sencilla. El diseño por sí mismo puede adoptar

diversas metodologías para cumplir su objetivo, la que se adoptó en el presente trabajo es la correspondiente al Diseño Estructurado.

Ideas Básicas del Diseño Estructurado

El diseño de sistemas de cómputo es una actividad que transforma una definición de qué es lo que se necesita, en un plan para implementar esos requerimientos dentro de un sistema computarizado. El diseño estructurado es un enfoque disciplinado del diseño de sistemas de cómputo y considera los siguientes aspectos:

1. Utiliza la definición de un problema como guía para la obtención de la solución del mismo.
2. El diseño estructurado trata de manejar la complejidad de un problema muy grande mediante la partición de éste en varios subproblemas bien definidos, conocidos con el nombre de "cajas negras", cuya solución implica menos esfuerzo; además, pone también mucho cuidado en organizar dichas cajas negras de una manera jerárquica adecuada para la implementación del sistema.
3. El diseño estructurado se auxilia de varias herramientas, sobre todo gráficas, para obtener diseños fáciles de entender, analizar e implementar.
4. Ofrece un conjunto de estrategias para lograr el diseño a partir de una buena definición del problema a resolver.
5. El diseño estructurado proporciona los medios para evaluar la calidad del diseño con respecto al problema que se quiere resolver.

En suma, el diseño estructurado conduce a la producción de sistemas confiables cuya construcción es fácil de entender, sistemas flexibles, durables y eficientes, en otras palabras, sistemas económicos que funcionan adecuadamente.

Lo que el diseño estructurado no es

El diseño estructurado no es *programación estructurada* ya que ésta no es más que un conjunto de reglas que, aplicadas adecuadamente, ayudan a generar código fuente confiable y fácil de entender. Adicionalmente, proporciona algunas sentencias de

programación que ayudan a este propósito (repeat until, while, etc.). La programación estructurada, aplicada por sí misma, es útil en la construcción de pequeños programas, ya que utiliza un enfoque "top-down" en su implementación, sin embargo, es incapaz de actuar sobre sistemas muy grandes y complejos; es en este punto donde el diseño estructurado hace su aparición tomando lo que la programación estructurada no considera: un método para bosquejar la solución de sistemas grandes y medianos con un grado de complejidad considerable. De cualquier forma ambos esquemas no son contradictorios, más aun, son complementarios ya que una vez que se tiene el esquema completo que ha surgido a partir del diseño estructurado, resulta muy natural programarlo estructuradamente.

El diseño estructurado no es un medio para obtener una buena especificación del sistema, ya que se parte precisamente de este punto. Si no se tiene una buena especificación aun se puede hacer un excelente diseño pero, muy probablemente, del problema equivocado.

El diseño estructurado no es un diseño "top-down" ya que este último recomienda una fragmentación el problema en problemas más pequeños pero no pone atención en la jerarquía que debe existir entre estos.

Finalmente, el diseño estructurado no es una técnica a través de la cual el diseñador pueda evitar el pensar; es sólo un conjunto de reglas heurísticas que ayudan a concebir el mejor camino para la implementación de un sistema.

Herramientas del Diseño Estructurado

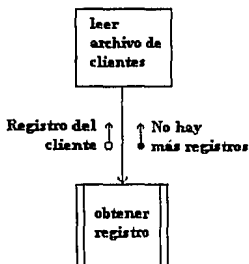
La Carta de estructura

La carta de estructura es la principal herramienta del diseño estructurado y muestra, precisamente, la estructura general del sistema: La partición del programa en módulos, la manera en que estos se organizan y la información que pasa de unos a otros. Puede observarse por tanto, que la carta de estructura se enfoca principalmente al aspecto de caja negra de cada uno de los módulos.

Considérese a un módulo simplemente como una subrutina que puede ser llamada por otra rutina, su representación se hace a través de un rectángulo con el nombre del módulo dentro de él (el nombre del módulo debe ser representativo de la función que realiza). Un módulo predefinido (que ya existe) será indicado mediante el mismo rectángulo con dos líneas paralelas en los extremos laterales.

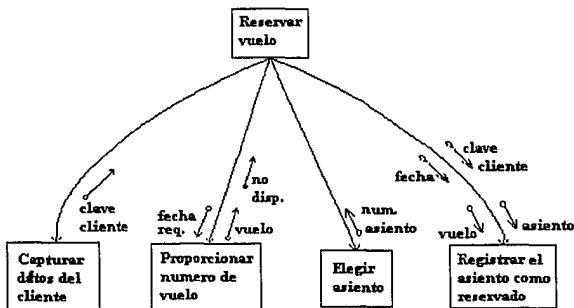


Los módulos están organizados jerárquicamente y se comunican unos con otros mediante flechas que parten de los módulos que invocan y llegan a los módulos que son invocados. Adicionalmente se muestran los datos que estos comparten, los cuales pueden ser datos del sistema y banderas de control entre módulos.



Acerca de la carta de estructura cabe mencionar que los módulos que son utilizados varias veces, sólo se dibujan una vez y tendrán tantas conexiones como módulos distintos los invoquen, de esta manera es fácil constatar si las interfaces entre módulos son consistentes. Por otro lado, el orden de izquierda a derecha en que

aparecen los módulos en el diagrama no necesariamente corresponde al orden en que estos se ejecutan en el sistema. Con respecto a los datos entre los módulos, por convención el nombre que se le da al dato es el nombre que le asigna el módulo que llamada a otra subrutina, recuérdese que el módulo invocado es capaz de manejar el mismo dato con un nombre distinto. Veamos un pequeño y simple ejemplo de una carta de estructura:



Aún cuando la carta de estructura proporciona una idea muy buena de la arquitectura del sistema, para un programador no es de mucha utilidad ya que ésta no incluye detalle procedural alguno. Esta información es proporcionada al programador a través de la especificación de cada módulo, la cual puede ser a partir de la interfase del módulo, o mediante pseudo código.

La especificación a partir de la interfase del módulo es tal vez el mejor medio para definirle al programador su trabajo, ya que contiene suficientes datos para saber la función que el módulo debe hacer, cuales son sus entradas y salidas, así como una descripción de la relación entre ellas. De esta manera no se inhibe la creatividad del programador y tampoco se permite que tome decisiones que no le corresponden. A continuación se presenta un ejemplo para mostrar claramente en qué consiste este tipo de especificación:

Consideremos un fragmento de la gráfica anterior como un módulo programable cuya especificación proporcionaremos a continuación.



Módulo: Proporcionar Número de Vuelo.

Propósito: Dada la fecha deseada para reservar un vuelo, el módulo deberá desplegar una tabla con los vuelos disponibles en dicha fecha y el usuario elegirá uno de ellos.

Var. Entrada: FechaReq

Retorna: Número_de_Vuelo
No_Disponible

Detalles Funcionales:

1. Analizar tabla_de_vuelos y mostrar vuelos disponibles para la fecha proporcionada.
2. Si no hay vuelos disponibles retornar No_Disponible
2. Preguntar vuelo deseado
3. Retornar Número_de_vuelo

Como ya se ha mencionado, otra forma de especificar un proceso es por medio de pseudo código el cual, si esta hecho correctamente, le otorgaría al programador el papel de algo cercano a un mecanógrafo que no requiere de gran ingenio para realizar su trabajo. Adicionalmente, no son muchos los diseñadores que quieren llegar a ese grado de detalle ni los programadores que deseen que otro haga el trabajo que a ellos les corresponde.

Características de un buen diseño

A continuación se analizarán los criterios para calificar el diseño de un sistema. Los principales son acoplamiento y cohesión, adicionalmente existen otros conceptos que son útiles para ese propósito, los cuales mencionaremos en su momento.

Acoplamiento

Acoplamiento es el grado de interdependencia que existe entre los módulos de un sistema. Es por tanto, una característica que debe ser reducida al mínimo. Las ventajas que proporciona son muy grandes: El sistema es más fácil de entender, más fácil de depurar, y se reduce al mínimo el riesgo de modificar un módulo y causar efectos negativos en los módulos relacionados.

Existen tres grandes categorías de acoplamiento y se listan a partir de la más deseable hasta la menos deseable.

1. **Acoplamiento normal;** se hace a través de parámetros entre módulos y se distinguen varios tipos:
 - i) *Acoplamiento de datos:* La comunicación entre módulos es a través de datos simples (un número, un "string").
 - ii) *Acoplamiento con etiquetas:* La interfase es una composición de datos como por ejemplo un registro o un arreglo..
 - iii) *Acoplamiento de control:* La interfase entre módulos se usa explícitamente para controlar a través de banderas, el comportamiento del módulo que es llamado. Ocurre frecuentemente en módulos que realizan más de una tarea, de tal forma que el módulo que llama debe conocer un poco sobre su

subordinado para indicarle qué es lo que debe hacer, en otros casos los papeles se invierten y le dicen al jefe qué debe hacer.

2. **Acoplamiento global o común;** el acceso a los datos es por medio de variables globales
3. **Acoplamiento patológico;** se hacen saltos repentinos en el código. Esta situación se evita con la programación estructurada.

Es importante señalar otras características indeseables dentro de un sistema:

- i) Que existan datos que tengan que viajar a través de varios módulos que no los necesitan para llegar al módulo destino.
- ii) Que existan estructuras de datos artificiales que agrupen (en un arreglo, por ejemplo) datos no relacionados entre sí, con el único objetivo de disminuir la cantidad de parámetros dentro de un módulo.
- iii) Asignar significados distintos a un mismo dato dependiendo del rango en que dicho dato caiga; por ejemplo, un número que del 0001 al 6999 se asigne a clientes normales, del 7000 al 8999 a clientes con 10% de descuento y del 9000 al 9999 a datos especiales de control de embarque. Esta situación es conocida como **acoplamiento híbrido** y debe evitarse a toda costa (más sobre esto se verá en lo referente a bases de datos).

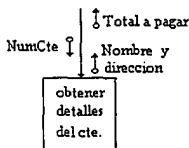
Cohesión

Otra medida de la calidad de la arquitectura de un sistema es la cohesión, y de manera general significa el agrupar funciones que estén altamente asociadas dentro de un solo módulo. Un diseñador tiene la obligación de crear módulos con un alto grado de cohesión. Logrando un alto grado de cohesión se asegura que el acoplamiento se reduzca al mínimo. En seguida se describen los principales tipos de cohesión comenzando por el más deseable hasta llegar a las clases de cohesión que deben evitarse.

Cohesión Funcional. Se presenta cuando la suma de los elementos o actividades de un módulo realizan una tarea bien definida sin que lo complicado que esta sea se tome en cuenta; por ejemplo, obtener el coseno de un ángulo, calcular el inverso de una matriz, calcular el salario neto mensual de un empleado, etcétera.

Cohesión Secuencial. Se manifiesta cuando una actividad dentro de un módulo sirve como entrada para otra actividad dentro del mismo módulo pero sin que la suma de estas realice una actividad bien definida. Por ejemplo, un módulo que extraiga una línea de un archivo y la mande a la impresora. Aunque no hay un problema grave con esta situación, sucede que en ocasiones desea utilizarse alguna de las funciones que engloba dicho módulo y no es posible hacerlo con una simple llamada a la subrutina.

Cohesión por comunicación. Un módulo con cohesión comunicacional es aquél cuyos elementos contribuyen a realizar actividades que utilizan los mismos datos de entrada y/o salida. Analicémos el siguiente ejemplo:



El único problema que aquí existe es que otro módulo pudiese necesitar el nombre y dirección del cliente y, si invocara este módulo obtendría un dato no deseado "Total a pagar" o bien, tendría que escribir un módulo que se encargara de esa tarea, lo cual sería redundante.

Cohesión procedural. Se asemeja un poco a la cohesión secuencial pero en este caso la relación entre las distintas actividades es más débil, ahora las salidas de una etapa pueden no requerirse en otra, lo que las caracteriza es que en dichas actividades el control fluye de una a la otra. En suma, están relacionadas por el orden de ejecución pero no porque tengan que ver con una sola función bien definida. Como ejemplo podemos citar el caso de un módulo que calcula el promedio de ventas y el promedio de gastos de una empresa.

Cohesión Temporal. Un módulo con esta característica es aquel que agrupa actividades cuya única relación es que se ejecutan en el mismo instante, esta situación produce un alto grado de acoplamiento entre módulos. Un ejemplo clásico es un módulo que inicializa todas las variables de un sistema, rebobinar

las cintas, etcétera. En caso de que se desee rebobinar la cinta¹ en otro momento, la opción de reinicializar todo el sistema de nuevo resultaría poco conveniente.

Cohesión Lógica. Un módulo lógicamente cohesivo es aquél que contiene cierto número de actividades de la misma clase y para usarlo, un parámetro indica qué tipo de actividad se requiere. Aunque las actividades sean distintas se ven forzadas a compartir la misma interfase y el significado de los datos depende del tipo de actividad que se le solicite. Veamos el pseudo código del siguiente módulo:

```
Módulo Envía/Borra Mensaje(acción, mensaje)
  Si acción == 1
    Despliega(mensaje)
  O_si acción == 3
    BorraMensaje( )
  Fin_si
Fin Módulo
```

Cohesión Coincidente. Un módulo de esta clase es aquel que enmarca actividades que no guardan relación alguna entre ellas, tal es el caso de un módulo de funciones misceláneas (que no cupieron en ningún otro lado). Este es el peor tipo de cohesión.

Consejos adicionales

Aún cuando son los principales, la cohesión y el acoplamiento no son los únicos medios para llegar a un diseño de alta calidad; adicionalmente existen otros criterios para lograr la meta deseada y a continuación se mencionan.

Factorización. La factorización es la creación de un módulo nuevo a partir de una función realizada como parte de otro módulo. De esta forma es posible crear unidades cada vez más simples y fáciles de entender, que realizan funciones mejor definidas y que es posible reutilizar posteriormente. otro criterio para "factorizar" un módulo es el separar la administración del trabajo, es decir, crear un módulo

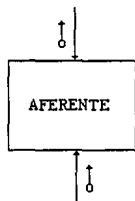
¹ Actualmente eso no es necesario, esas funciones las realiza siempre el sistema operativo, pero por esta ocasión supóngase que se tuviera que programar.

ejecutivo que coordine a varios módulos obreros. El único límite para la "factorización" de módulos es la complejidad que se crea entre las interfases del módulo.

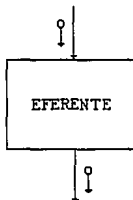
Decisiones repartidas ("Decision splitting"). Evite al máximo que un módulo tome una decisión dependiendo del valor obtenido por otro módulo que esté alejado del primero, ya que tendríamos en camino un *dato viajero*, es decir, un dato que pasa (sin ser procesado) por uno o varios módulos intermedios hasta llegar al módulo que lo requiere.

Tipos de módulos. Antes de continuar es pertinente mencionar que básicamente encontramos cuatro tipos de módulos diferentes:

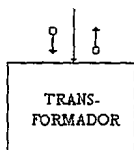
- i) **Aferentes.** Estos módulos reciben información de módulos inferiores, la procesan y la envían a módulos superiores.



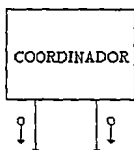
- ii) **Eferentes.** El módulo manda información que recibió de un módulo superior a su subordinado.



- iii) **Transformador.** El módulo obtiene información de un superior, la transforma y la envía de regreso.



- iv) **Coordinador.** El módulo coordina la comunicación de sus subordinados



"Fan-Out": Es el número de subordinados inmediatos que tiene un módulo. Se recomienda mantener el "fan-out" mayor que uno y menor que siete.

"Fan-In": Es el número de jefes inmediatos que tiene un módulo. Un "fan-in" elevado es muestra de una factorización adecuada y de la existencia de módulos altamente re utilizables.

Creación de las cartas de estructura

La fase de análisis entrega como producto final un diagrama de flujo de datos (DFD) que servirá como entrada para la etapa de diseño. En esta sección se describirá un método para derivar las tablas de estructura —que definirán la arquitectura del

sistema— a partir del DFD que arrojó el análisis; una vez hecho esto, es posible comenzar a programar el sistema.

El método se compone de dos técnicas (análisis de transacciones y transformación) para obtener una tabla de la estructura del sistema. Posteriormente habrá que hacer uso de los criterios, descritos anteriormente, para lograr un excelente diseño.

Análisis de las transacciones del sistema, partición del mismo

A través de este análisis se identifican los distintos tipos de transacciones del sistema y se utilizan como unidades separadas de diseño. Este proceso resulta muy sencillo debido a la forma en que el análisis estructurado obtiene el DFD del sistema.

Una transacción consta de cinco partes que a continuación se describen:

EVENTO. suceso que demanda respuesta del sistema.

ESTIMULO al sistema.

ACTIVIDAD realizada por el sistema.

RESPUESTA del sistema.

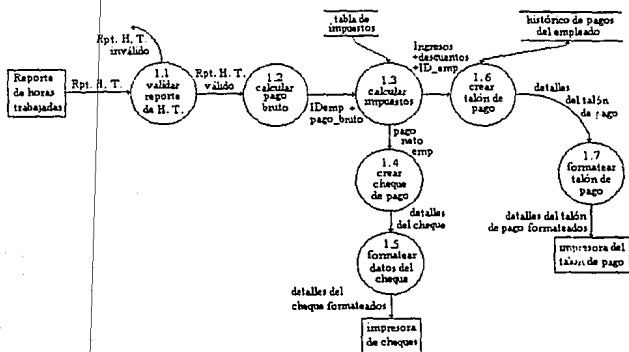
EFECTO de la actividad del sistema.

Por ejemplo, consideremos a un F1 y su piloto como un sistema; antes de entrar a una curva (evento) es necesario disminuir la velocidad del auto y el piloto acciona el freno (estímulo). Las partes mecánicas, hidráulicas y electrónicas del subsistema de frenos interactúan (actividad) para ejercer presión sobre los discos de frenado (respuesta) y así lograr disminuir la velocidad del auto (efecto). En este caso una transacción sería considerada como el conjunto de factores y elementos que intervienen en el frenado de un F1; un análisis similar podría hacerse para cada subsistema del auto en cuestión.

Transformación

Es un método para transformar en una carta de estructura cada pieza que fue aislada en el análisis de transacciones. Esta estrategia consta de varios pasos:

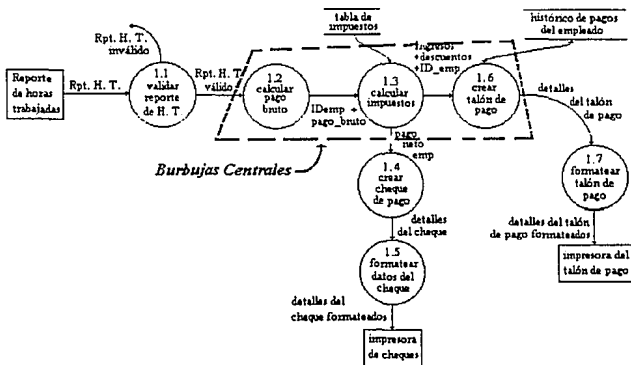
- a) **Identificar burbujas centrales.** Lo primero que hay que hacer es buscar los procesos más importantes del diagrama, una manera de hacerlo es eliminando las partes aferentes y eferentes del DFD en estudio: Empezando por el lado aferente por ejemplo, avanzamos hacia el centro del diagrama y nos detenemos cuando la rama transporte el dato en su forma más elemental, esto es, cuando ya fue probado, validado o formateado de algún modo pero aun no ha sido procesado. De manera similar se hace con el lado eferente, sólo que en esta ocasión marcamos la rama que produce el dato en su forma más elemental, es decir, aun no ha sido formateado. Al final quedará un pequeño grupo de burbujas que deberán ser encerradas dentro de una línea punteada y que representan las burbujas centrales del proceso. A continuación se presenta el DFD para calcular los ingresos e impuestos de un empleado dado, e imprimir el cheque y el talón de pago correspondiente.



El primer paso a seguir es entonces localizar las burbujas centrales del DFD; eliminando los extremos aferentes y eferentes, de tal manera que se localicen las burbujas que realizan la transformación central del diagrama, esto es, las

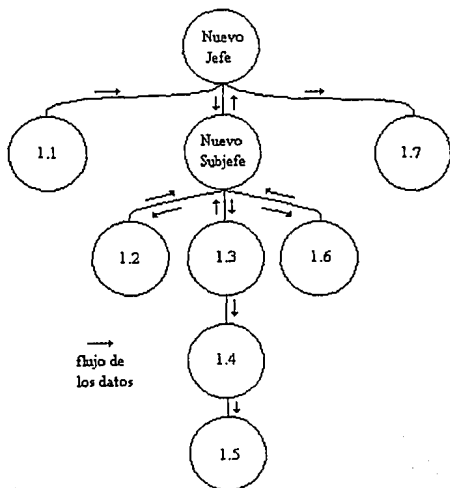
burbujas que realizan las operaciones más importantes de la transacción que se está analizando.

Recuérdese que los módulos aferentes son aquellos que reciben datos de módulos inferiores, los procesan y los envían a módulos superiores, por tanto, del DFD puede observarse que la burbuja 1.1 es un módulo aferente, ya que sólo valida el reporte de horas trabajadas, por tanto será descartada. La burbuja 1.2 procesa el dato Rpt.H.T., así que será considerada como una burbuja central. Considérese ahora el lado eferente; la burbuja 1.5 sólo formatea los datos que entrega la burbuja 1.4, la cual por cierto realiza una tarea de recolección más que de alguna transformación importante de datos los datos, razón por la cual también será excluida. La burbuja 1.7 tampoco formará parte del grupo central. El resultado se muestra en la siguiente gráfica.

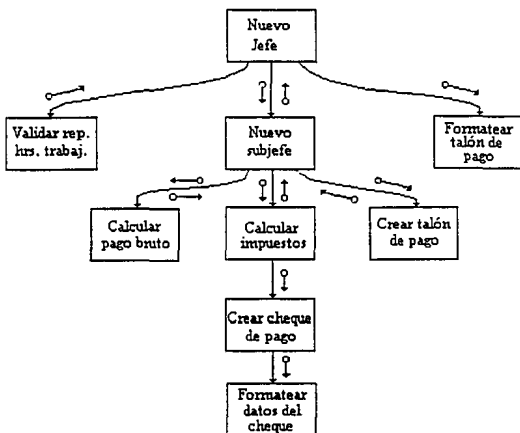


- b) **Generar una carta de estructura preliminar.** Con base en el DFD que fue analizado en el punto anterior se creará un diagrama de estructura preliminar, y para lograrlo se habrá de elegir un líder que coordine las tareas de las demás burbujas. Es posible elegir entre una de las burbujas del DFD para promoverlo a

jefe, o bien, "contratar" un jefe externo, es decir, crear un módulo extra para coordinar las acciones entre módulos. Una burbuja del grupo de burbujas centrales es una buena candidata para convertirse en jefa ya que se presume que realiza una gran actividad de coordinación —debido al análisis hecho en el inciso anterior—. Es muy probable que exista más de una candidata elegible, en este caso, pruebe una por una hasta encontrar la burbuja ideal, si esto no sucede, entonces se "contrata" un jefe externo para que realice la labor de coordinar a las otras burbujas, si la elección no fue la correcta, el error se hará evidente en revisiones posteriores. Poco a poco se irá ganando experiencia y será más sencillo encontrar al jefe idóneo para la parte del sistema que se está analizando. Una vez elegido, se imagina que las demás burbujas penden de él y se eliminan las puntas de las flechas del diagrama flujo de datos, ya que en la carta de estructura, estas indican la jerarquía que existe entre módulos y no el flujo de datos.

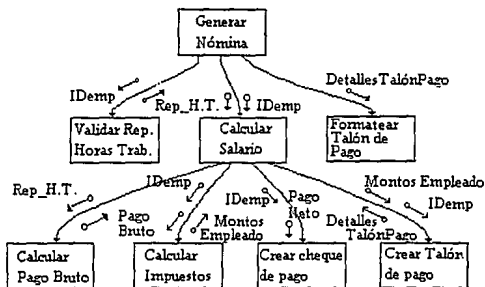


Ahora, se asume que cada burbuja es un módulo del sistema, por lo tanto habrá de dibujarse un rectángulo por cada burbuja, se asignan nombres adecuados a los módulos —no necesariamente los mismos del diagrama de flujo de datos— y se señalan las direcciones de las flechas de acuerdo a la jerarquía que impera en la tabla (quién llama a quién); se colocan también las flechas que indican el flujo de datos entre módulos.

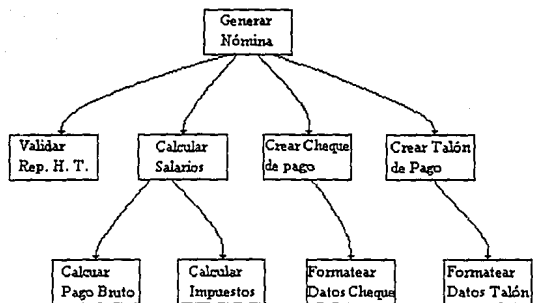


- c) **Revisar la carta de estructura preliminar.** En esta fase es necesario mejorar, mediante los principios básicos del diseño estructurado, la carta de estructura, que sólo es medianamente aceptable. La revisión de la carta de estructura se debe hacer de la siguiente forma:
- Agregar módulos de lectura y escritura para acceder a archivos, bases de datos y otras fuentes de información.
 - Factorización y reorganización de módulos aferentes y eferentes.
 - Factorización de las burbujas centrales.

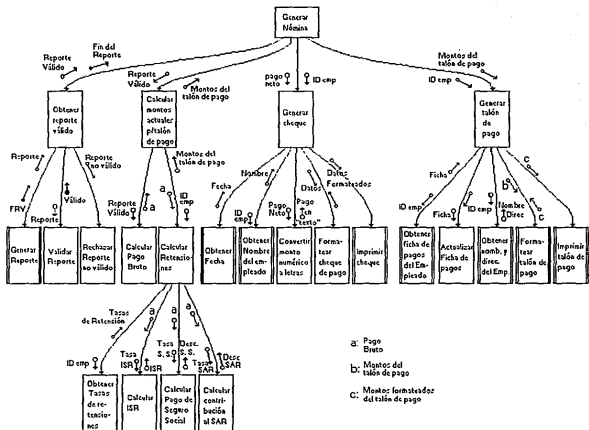
- Agregar módulos para el manejo de errores.
- Si es necesario, agregar detalles de inicialización y terminación.
- Asegurarse de que todos los módulos tienen nombres adecuados.
- Agregar banderas de control que no aparecían en el DFD (por ejemplo un EOF).
- Asegurarse de que la carta de estructura cumple con los criterios de un buen diseño (bajo acoplamiento, cohesión alta, etc.).



Crear_Cheque_de_Pago fue subordinado directamente a Calcular_Salario para mantener coherencia jerárquica en Calcular_Impuestos —Crear_Cheque_de_Pago no tiene mucho que ver con Calcular_Impuestos—, por la misma razón se subordinará directamente a Generar_Nómina y así tendrá más sentido la organización de la tabla. Algo semejante ocurre con Crear_Talón_de_Pago y Formatear_Talón_de_Pago, lo cual provoca un "dato viajero" entre Calcula_Salario y Generar_Nómina. A continuación se muestra un mejor diseño.



Aún falta completar el diagrama con módulos de E/S y manejo de errores.
El diseño final sugerido es el siguiente:

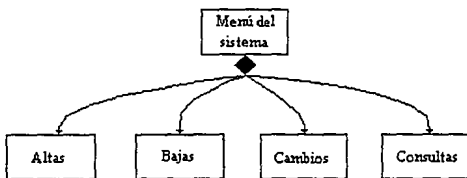


71

- d) **Asegurarse de que el diseño funciona.** Una forma de hacerlo es mostrando la carta de estructura al analista, para que confirme que el diseño implementa correctamente la especificación. Otra forma es imaginar a grandes rasgos el pseudo código de cada módulo, si éste parece demasiado oscuro, revise de nuevo su diseño.

Integración

Cuando se han creado todas las cartas de estructura del sistema, se tiene un conjunto de diseños separados, ahora es necesario reunirlos todo de nuevo en una sola estructura de tal modo que cada vez que aparezca el estímulo de una transacción, el módulo correspondiente sea activado. Una forma de hacerlo es a través de un menú de selección o de alguna estructura adecuada de opción múltiple ("case", por ejemplo). Una vez hecho esto podemos considerar que la etapa de diseño ha concluido, aún cuando el diseñador habrá de bosquejar una estrategia de implementación y colaborar estrechamente con los programadores para obtener un producto terminado de alta calidad.



El rombo indica que se trata de una función semejante a una sentencia "CASE" de un lenguaje de tercera generación.

Diseño de Bases de Datos Relacionales

El diseño estructurado como tal, no hace referencia a la forma en que deben organizarse las bases de datos que utilizará el sistema, adicionalmente es imperativo que se lleven a cabo tareas encaminadas a la configuración de una base de datos adecuada, que satisfaga los requerimientos de funcionalidad y desempeño del sistema. Estas tareas normalmente se realizan de manera simultánea, y debido a la fuerte relación que existe entre ellas, es necesario que los equipos responsables mantengan una estrecha comunicación para que la integración del sistema sea más eficaz.

En esta sección se revisarán los conceptos básicos del diseño de bases de datos relacionales junto con algunas recomendaciones que deberán ser tomadas en cuenta para lograr una estructura saludable en la base de datos que requiere el sistema.

El Modelo Relacional

Uno de los más importantes aliados en los sistemas de información es sin duda la independencia de los datos. Esta independencia permite tener una gran flexibilidad tanto en la explotación como en el mantenimiento de los mismos; además, es la característica fundamental que diferencia al modelo relacional del modelo jerárquico o el de red.

La explotación de datos en el modelo jerárquico es totalmente dependiente de los distintos árboles de conexión en la estructura de los datos; esto hace obligatorio caminos de acceso que a simple vista parecen sobrados. A medida que surgen diversos tipos de consulta comienza a crecer el tipo de conexiones aumentando considerablemente la complejidad de la estructura, haciendo evidente entonces la poca flexibilidad que un modelo de esta clase representa. Todo esto da como resultado una fuerte dependencia en las distintas conexiones y una difícil y compleja explotación de los datos.

El modelo relacional ofrece una solución viable a las carencias que tienen los modelos anteriores a éste. En principio se puede pensar que el modelo relacional es una estructura que nos ayuda a representar los datos desde el punto de vista del usuario; como concibe los datos, como los manipula y de que manera realiza operaciones sobre éstas. Esta representación se logra principalmente a través de relaciones entre

conjuntos de datos y no implica cuestiones físicas ni técnicas, logrando así una gran independencia entre la representación de los datos y la implementación del sistema de información que se está construyendo.

El modelo relacional proviene de la Teoría de Conjuntos, donde dado los conjuntos S_1, S_2, \dots, S_n una relación R se define como un subconjunto del producto cartesiano $S_1 \times S_2 \times S_j \times \dots \times S_n$, donde S_j se puede expresar como el Dominio j de la relación R (recordando que un producto cartesiano entre dos conjuntos $A \times B$ es la colección de todas las parejas ordenadas de $[a,b]$ donde el elemento "a" pertenece a A y "b" a B). Dicha relación debe observar las siguientes características:

- Cada elemento del producto Cartesiano o Renglón forma una Tupla n de la Relación
- El orden de los renglones es insignificante
- Todos los renglones son distintos
- El orden de las columnas corresponde al orden de los dominios en los que R fue definida (S_1, S_2, \dots, S_n) significado de las columnas se resuelve parcialmente etiquetando los dominios con diferentes nombres.

Al aplicar dicho modelo para Bases de Datos, la estructura de los datos se puede representar mediante relaciones o tablas bidimensionales (renglones y columnas) donde:

1. Cada columna representa un único valor por renglón
2. Los datos para una columna son del mismo tipo o dominio
3. Cada renglón dentro de una relación es único
4. La secuencia de los datos de las columnas es insignificante
- 5 La secuencia de los datos en los renglones es insignificante
6. Cada columna tiene un nombre único

Existe una diferencia entre la relación de los conjuntos y la de la relación enfocado a las bases de datos: En la relación en conjuntos el orden de las columnas no se debe alterar (esto porque el producto cartesiano no es conmutativo), mientras que para las relaciones de datos son consideradas como equivalentes todas las relaciones originadas de las diferentes permutaciones de columnas de la relación original.

En la figura se observa una relación de orden 4 llamada Pedido, donde sus dominios son No_Pedido, Artículo, Cantidad, Precio. Cada registro corresponde a una tupla de la relación y por lo tal son únicos. Dicha relación es equivalente a Pedido2. Dicho de otra manera Pedido2 es una permutación posible de los dominios de Pedido1, por lo tanto se consideran la misma relación.

Relación Pedido

No Pedido	Artículo	Cantidad	Precio
1	2	3	2
2	2	4	6
3	3	2	4
4	4	3	2

Relación Pedido de orden 4

Relación Pedido2

No Pedido	Cantidad	Artículo	Precio
1	3	2	2
2	4	2	6
3	2	3	4
4	3	4	2

Relación Equivalente Pedido2

Como las relaciones son conjuntos todas las operaciones aplicables a conjuntos también pueden ser aplicadas a estas. Este hecho en sí es muy significativo, ya que permite la realización de distintas operaciones entre diversas relaciones dejando atrás las complicadas rutas de acceso para relacionar diferentes tipos de datos. Sin embargo, no todas las operaciones aplicadas en conjuntos dan como resultado relaciones (la unión de una relación binaria con una ternaria no da una relación).

Los tipo de operaciones que se pueden efectuar en los datos de tipo relacional son de dos tipos: de ASIGNACIÓN donde se asigna relaciones a otras relaciones, y OPERACIONALES donde se usan ocho tipos de operadores, estos son:

SELECT, que selecciona un subconjunto de renglones de acuerdo a cierto criterio, por ejemplo Select de Pedido aquellos cuyo precio sea mayor que 2.

Relación Resultante

No Pedido	Artículo	Cantidad	Precio
2	2	4	6
3	3	2	4

Selecto de Pedido Precio mayor de 2

PROJECT, que selecciona un subconjunto de columnas de acuerdo a cierto criterio, por ejemplo Project de Pedido los dominios No_Pedido y Precio

Relación Resultante

No Pedido	Precio
1	2
2	6
3	4
4	2

Project de Pedido en No_Pedido y Precio

PRODUCT, que realiza un producto cartesiano de dos tablas. formando una tabla resultante, ejemplo tenemos dos tablas de relaciones producto de dichas relaciones

Relación NumParticipantes

Deporte	Participantes
tennis	2
boliche	1

Relacion DatParticipantes

Nombre	Edad
Pedro	16
Laura	23

Relación Resultante

Deporte	Participantes	Nombre	Edad
tenis	2	Pedro	16
boliche	1	Pedro	16
tenis	2	Laura	23
boliche	1	Laura	23

Product de Numparticipante con DatParticipante

JOIN, es un producto cartesiano pero para un cierto criterio en su resultado, por ejemplo el join de la relación Num participantes con Dat Participantes para Nombre igual a Laura.

Relación Resultante

Deporte	Participantes	Nombre	Edad
tenis	2	Laura	23
boliche	1	Laura	23

Join de Numparticipante con DatParticipante para Nombre igual a Laura

UNIÓN, combina a dos tablas en forma vertical eliminando duplicados, para que dos relaciones puedan tener una Unión deben de tener el mismo tipo de dominios, ejemplo:

Relación Vendedor Zona Norte

Nombre	Lugar
Antonio	Sonora
Jorge	Coahuila

Relación Zona Sur

Nombre	Lugar
Pedro	Yucatán
Laura	Tabasco

Relación Resultante

Nombre	Lugar
Antonio	Sonora
Jorge	Coahuila
Pedro	Yucatán
Laura	Tabasco

Unión de Vendedor Zona Norte y Zona Sur

INTERSECCIÓN, toma los renglones repetidos en dos tablas

Relación Aficionado Ajedrez

Nombre	Edad
Antonio	30
Jorge	21

Relación Aficionado Boliche

Nombre	Edad
Pedro	15
Antonio	30

Relación Resultante

Nombre	Edad
Antonio	30

Intersección entre ambas tablas

DIFERENCIA, el resultado corresponde a renglones que no aparezcan en una tabla pero en otra si. Por ejemplo la diferencia entre las Relaciones Aficionado de Ajedrez y Aficionados de boliche.

Relación Resultante

Nombre	Edad
Jorge	21

Diferencia entre Aficionados de Ajedrez y Aficionados de Boliche

DIVISIÓN, el resultado corresponde a un subconjunto de valores en una columna cuyas demás columnas corresponden a todos los renglones de otra tabla (divisora). Ejemplo de la Relación Pedido considerada como Dividendo el divisor será la tabla:

Relación Pedido (dividendo)

No Pedido	Artículo	Cantidad	Precio
1	2	3	2
2	2	3	6
3	3	2	4
4	4	3	2

Relación Pedido de orden 4

Relación Artículo (divisor)

Artículo	Cantidad
2	2

Diferencia entre Aficionados de Ajedrez y Aficionados de Boliche

Relación Resultante

No Pedido	Precio
1	2
2	6

Relación Pedido de orden 4

Existen muchas alternativas de como usar el modelo relacional en las bases de datos, sin embargo se recomienda el uso de la forma normalizada ya que esta permite conservar la integridad de los datos y evitar redundancias.

Antes de explicar la normalización se necesita introducir conceptos importantes como llave primaria, llave foránea, dominio activo, dominio compuesto.

Llave Primaria: Así se le conoce al dominio o combinación de dominios en el que se puede diferenciar una tupla de la relación con otras para todo el tiempo. En el caso en que más de un dominio o combinación de dominios puedan funcionar con ese fin, la llave primaria se seleccionará arbitrariamente entre las candidatas.

Llave Foránea: Se le conoce como llave foránea en una relación R al dominio o combinación de dominios que no es la llave primaria de dicha relación pero forman una llave primaria de otra relación

Dominio Activo: Se le conoce como el conjunto de todos los dominios que se presentan en el tiempo presente

Dominio Compuesto: se le conocen así a dominios que no son atómicos, es decir, a dominios que pueden representar más de un atributo en una misma relación.

La Integridad de Datos se refiere a que los datos que conciernen a nuestras relaciones deben de cumplir con ciertas reglas de integridad. Estas reglas de integridad son de dos tipos: Relacionales y de Entidades. La integridad relacional se fija en aspectos de validación lógica entre las diferentes relaciones heredadas (esto se lleva a cabo mediante llaves foráneas). Mientras que las reglas de integridad de entidades se fijan en el aspecto de consistencia de las tablas en donde no permite que el identificador de un renglón de una tabla (llave primaria) sea nulo.

Una base de datos relacional engloba aspectos del modelo relacional y de las bases de Datos en donde los usuarios perciben los datos como tablas que obedecen a las 6 propiedades de relación que son manipuladas mediante 8 operadores y protegidas mediante reglas de integridad.

Normalización

En el modelo Entidad-Relación contamos con atributos asociados intuitivamente a entidades, el proceso de normalización es una técnica formal para evaluar modelos de datos y modificarlos para obtener relaciones que posean propiedades más deseables como son: Consistencia en los datos, mínima redundancia, minimización de anomalías en la actualización y borrado, minimizar el uso de espacio en disco y simplicidad de estructura.

Primera Forma Normal

Se deben eliminar atributos repetitivos o no atómicos.

Consideremos el siguiente ejemplo:

PEDIDO	
(PK) No Pedido	
No Cliente	
Fecha Pedido	
Artículo 1	
Cantidad 1	
Precio 1	
Artículo 2	
Cantidad 2	
Precio 2	
Artículo 3	
Cantidad 3	
Precio 3	

Los atributos Artículo_n, Cantidad_n y Precio_n constituyen un grupo repetido. Para que se cumpla la primera forma normal (PFN) deberemos crear otra entidad que contenga todas las líneas del pedido:

PEDIDO

(PK) No Pedido
No Cliente
Fecha

LINEA

(PK) No Pedido
(PK) No Línea
No Artículo
Cantidad
Precio

La utilidad de la PFN es principalmente el evitar desperdicio de espacio y simplificar la estructura de la base de datos. Otra violación a la PFN es el incluir datos compuestos dentro de un atributo. Por ejemplo:

ALMACÉN

(K) No Artículo	Posición (Pasillo/Nivel/Cajón)	Existencia
-----------------	--------------------------------	------------

Transformando nuestra estructura a atributos atómicos tendremos:

ALMACÉN

(PK) No Artículo	Pasillo	Nivel	Cajón	Existencia
------------------	---------	-------	-------	------------

Segunda Forma Normal

Una entidad en PFN puede ser reducida de primera a segunda forma normal si todos los atributos de una entidad deben ser dependientes de la llave primaria completa, es decir, no deben ser identificables por una porción de la llave. Veamos el siguiente ejemplo:

PEDIDO

(PK) No Pedido	(PK) No Cliente	Fecha	Nombre Cliente
----------------	-----------------	-------	----------------

Observemos que Nombre_Cliente es dependiente de No_Cliente y no de la llave entera. Para cumplir con la SFN debemos eliminar Nombre_Cliente de PEDIDO ya que es redundante. El lugar indicado para Nombre_Cliente es una entidad como CLIENTE en donde se describan las características del cliente.

Tercera Forma Normal

Es posible reducir las entidades en SFN a tercera forma normal eliminando atributos dependientes o identificables por otros atributos que no sean llave primaria (ni llave candidata). Considérese el siguiente ejemplo:

LINEA

(PK) No Ped	(PK) No Lin	No Artículo	Descripción	Cantidad	Precio
-------------	-------------	-------------	-------------	----------	--------

Podemos observar que la descripción del artículo depende del atributo No_Artículo. Eliminando Descripción tenemos a la entidad LINEA en TFN.

Forma Normal Boyce/Codd

La Forma Normal Boyce/Codd asegura que al elegir una llave candidata se cumpla hasta la TFN de tal modo que para probarla es necesario utilizar las llaves candidatas como llaves primarias y asegurar que se cumpla hasta la TFN.

Cuarta Forma Normal

Reduce entidades en forma normal Boyce/Codd a la cuarta forma normal eliminando cualesquiera componentes independientes con múltiples valores, en dos entidades padre nuevas. Conserva la entidad original sólo si tiene atributos que no son llave primaria. En otras palabras, identifica y separa atributos independientes con valores múltiples que constituyan una llave primaria compuesta. Es muy probable que intuitivamente se llegue a cumplir con esta forma normal. Veamos el siguiente ejemplo:

Supóngase que se tiene una entidad ESCUDERÍA GRAN PREMIO_PILOTO que forma parte de un sistema para carreras de Fórmula 1

ESCUADERÍA GRANPREMIO PILOTO

Escudería	Gran Premio	Piloto
PK	PK	PK
Ferrari	Mónaco	Alesi
Ferrari	Mónaco	Berger
Ferrari	Tokio	Alesi
Ferrari	Tokio	Berger

Podemos observar que en esta estructura se repite el piloto para cada Gran Premio de cada escudería. Es fácil encontrar una mejor solución:

ESCUADERÍA GRAN PREMIO

Escudería	Gran Premio
PK	PK
Ferrari	Mónaco
Ferrari	Tokio

ESCUADERÍA PILOTO

Escudería	Piloto
PK	PK
Ferrari	Alesi
Ferrari	Berger

Quinta Forma Normal

Reduce entidades en CFN a quinta forma normal eliminando las dependencias cíclicas en pares, que aparecen dentro de llaves primarias compuestas por tres o más atributos, mediante la creación de tres o más entidades padre nuevas. Una dependencia cíclica es provocada por una regla impuesta dentro de la empresa. Tal vez sea más claro si analizamos el siguiente ejemplo:

Supóngase que existe una pequeña asociación de vendedores independientes de revistas para distribución por mayoreo a distintas zonas de la ciudad. Una política interna de la empresa les obliga a comprarle a un editor, además de otras revistas, todas las revistas relacionadas con la computación que el editor publique.

COMPRAS

Vendedor	Editor	Título	Cantidad
PK	PK	PK	
Filemón	Panamericana	BYTE	350
Filemón	Panamericana	RED	200
Filemón	Panamericana	Mecánica Popular	400
Dionisio	Panamericana	BYTE	460
Dionisio	Panamericana	RED	320
Dionisio	Panamericana	Audio y Video	100
Filemón	Editores S. A.	UNIXWORLD	95
Dionisio	Editores S. A.	UNIXWORLD	170

En esta entidad existe una dependencia cíclica generada por la condición que obliga a un vendedor a comprar todos los títulos relacionados con computación, de un editor. Si, por ejemplo, Editores S. A. publicase una revista de computación sería necesario actualizar todas las ocurrencias de vendedores que le compren a Editores S. A. Para cumplir con la quinta forma normal se tendrá que descomponer esta estructura en tres entidades padre cuyos atributos serán los pares dependientes:

VENDEDOR EDITOR		EDITOR TITULO		TITULO VENDEDOR	
Vendedor	Editor	Editor	Título	Título	Vendedor
PK	PK	PK	PK	PK	PK

VENTAS

Vendedor (PK)	Editor (PK)	Título (PK)	Cantidad
---------------	-------------	-------------	----------

La quinta forma normal es bastante teórica y compleja; sin embargo, es importante ya que es la última de las formas normales y está enfocada a eliminar redundancias en operaciones JOIN y PROJECT. Aún así, resulta poco práctica y difícil de entender, además si el modelo cumple hasta la tercera forma normal podemos considerar que éste es bastante robusto por lo que podemos considerar a la quinta forma normal como de menor prioridad.

Consideraciones Adicionales sobre la Normalización

1. No separar entidades completamente normalizadas en entidades menores
2. Revisa las entidades normalizadas y, si es necesario, agrega atributos redundantes para asegurar la integridad de datos crítico. También es posible incluir datos derivados (que pueden ser calculados a partir de otro u otros atributos) para mejorar el desempeño del sistema, pero es necesario señalarlos.

Aún cuando las reglas de normalización nos ayudan a detectar y controlar muchos tipos de redundancia, no solucionan todos los problemas de diseño de una base de datos, deberán hacerse consideraciones especiales acerca de la distribución física de los datos y los métodos de acceso a los mismos: creación de índices y de tablas temporales así como de reglas de integridad y seguridad de acceso a los datos, que influyen en el desempeño total, la confiabilidad y robustez del sistema.

Implementación

Una vez que las etapas de Análisis y Diseño tienen ya un buen grado de avance (no necesariamente terminadas) se cuenta ya, por lo menos con una parte significativa de las cartas de estructura y por tanto con los cimientos necesarios para llevar a cabo lo que se le conoce como implementación. En esta etapa se realiza la programación, las diferentes pruebas del sistema, instalación y capacitación. Es la parte más larga del Ciclo de Vida de un Sistema. Esta implementación como se mencionó anteriormente es de tipo "Top Down" o Deductivo.

Programación

La programación es la herramienta que permite implementar las especificaciones que se cubren en el análisis de acuerdo a la organización vista en el diseño en una serie de declaraciones de un lenguaje de programación.

El primer paso que se recomienda es empaquetar los diferentes módulos del Sistema, descritos en el Diagrama Estructural, en unidades de carga ("Load Units"); es

muy probable que en muchos sistemas las Unidades de Carga coincidan con los módulos pero también en muchas otras es necesario descomponer un módulo en diferentes Unidades de Carga o inclusive, el caso contrario, juntar diferentes módulos en una Unidad de Carga. La características de dichas Unidades de Cargas son las siguientes: tienen una alta Acoplación entre los diferentes procedimientos que las componen, pero en conjunto se tiene una altísima Cohesión. Entre los diferentes Unidades de Carga el Acoplamiento se reduce al mínimo, por lo que llamadas Dinámicas (llamadas de procedimientos que se encuentran en diferentes Unidades de Carga) se vuelven mínimas mientras las llamadas Estáticas (llamadas de procedimientos que se encuentran en la misma Unidad de Carga) se dan con mayor frecuencia. Una buena empaquetación aumenta la eficiencia del sistema además facilita el mantenimiento del mismo.

La programación también se recomienda de tipo "Top Down", los módulos de nivel superior se programan, en general, antes que los otros. Estos módulos de nivel superior generalmente llaman a otros, entonces se crean módulos fantasmas o "dummies" que serán sustituidos paulatinamente por los módulos reales cuando estos sean programados. Un módulo se considera de un nivel superior a otro si el comportamiento de uno esta en función al resultado de otro, o bien si un módulo llama a otro. Es gracias a este tipo de programación que permite llevar a cabo paralelamente etapas de análisis y diseño. Una ventaja más de este modelo de programación es que existe un avance palpable en poco tiempo, ya que a partir de la programación de los módulos superiores se obtiene un esqueleto del sistema; paulatinamente dicho esqueleto evoluciona a un nuevo esqueleto más sofisticado y así sucesivamente hasta formar el cuerpo completo del sistema. Cada esqueleto representa un cambio palpable tanto para el programador como para los usuarios, esto psicológicamente es una gran ventaja. El hecho de poder tener diferentes esqueletos del sistema hace posible tener versiones funcionales del mismo, aunque éste no este terminado en su totalidad.

En general, en la programación de tipo "Top Down", como se ha mencionado, los módulos superiores se programan antes que los otros; sin embargo esto no es riguroso, ya que en muchas ocasiones es necesario terminar con un módulo y sus diferentes submódulos para poder programar otros módulos. Es recomendable, cuando esto sea el caso, comenzar con los módulos ligadores de alto nivel y luego seguir con los módulos eferentes y aferentes, para después continuar con los transformados y coordinados. Esto se conoce como la implementación de tipo paraguas.

Lenguajes de Programación

Un lenguaje de programación es la herramienta indispensable para que un programa pueda ser codificado y ejecutado por la máquina. Es por lo tanto una herramienta vital para cualquier tipo de implementación en la que intervenga una computadora y no necesariamente se utiliza un sólo lenguaje para este propósito. Existen diferentes tipos de lenguajes de programación pero estos se pueden dividir en 4 grandes grupos de acuerdo a su evolución:

1ª GENERACIÓN: nace en los años 50s , los lenguajes de programación de este están muy enfocados en la lógica de máquina, para codificar sus instrucciones se tiene que hacer en con lógica binaria de unos y ceros.

2ª GENERACIÓN: son los sucesores del lenguaje de máquina binario, sigue estando muy enfocado en la lógica de máquina, sin embargo, la codificación binaria se sustituye por declaraciones en la cual cada declaración corresponde a una instrucción de máquina (por esta razón se consideran de Bajo Nivel). Lenguajes de este tipo se observan en los ensambladores

3ª GENERACIÓN: no están enfocados en la lógica de máquina ya que cada declaración representa un conjunto de declaraciones de bajo nivel (generalmente de 5 a 10 pero puede ser inclusive de cientos, es por esta razón que se consideran que son de Alto Nivel). Este tipo de lenguaje involucra directamente al programador en labores tediosas de validación de variables, formateo visual etcétera. Ejemplos de este tenemos Pascal, C, Basic y otros.

4ª GENERACIÓN: En los lenguajes de este tipo se reúnen en una sola declaración cientos de declaraciones de Bajo Nivel, muchas de las labores tediosas como validación de variables son transparentes para los usuarios , se facilita también la elaboración de reportes y de formas. Ejemplos de estos son SyBase, Oracle, Progress, Informix etc.

Independientemente del tipo de lenguaje que se seleccione existen aspectos en común que se deben tomar en consideración durante la programación, estos son: Productividad, Eficiencia, Correctividad, Portabilidad, Mantenimiento.

PRODUCTIVIDAD: La capacidad del programador para obtener el mayor número de módulos en el menor tiempo posible.

EFICIENCIA: El los recursos de la máquina como memoria, tiempo de CPU y de otros recursos como discos, impresoras etc. Muchas veces la eficiencia y la productividad son características encontradas, ya que a menudo se sacrifica unos de estos aspectos para beneficiar al otro.

CORRECTIVIDAD: Un sistema debe de trabajar correctamente, debe de cumplir adecuadamente con los requisitos de análisis que fueron planteados. Este es el punto más importante de todos, no importa que tan eficiente y productivo pueda ser un sistema si este no funciona.

PORTABILIDAD: La capacidad de que un sistema funcione en distintos ambientes, un usuario querrá trabajar sus sistemas en distintos tipos de computadora.

MANTENIMIENTO: La vida del sistema tomando en consideración modificaciones a futuro. ¿Que tan fácil es hacer dichas modificaciones ? Para cuidar este aspecto se recomienda una programación de tipo estructurada (si se selecciona un lenguaje de 3a o 4a generación), tener módulos pequeños y tener un estilo claro que facilite a un programador promedio entender la lógica del programa

Pruebas

Una vez que se tiene el sistema se encuentra terminado (o parte del sistema. recordando que no necesariamente este tiene que estar terminado por completo, ya que se puede comenzar a probar módulos superiores del sistema antes de que la totalidad de los módulos se hayan programado) se realizan las pruebas. Una prueba es una manera de ejercitar el sistema para asegurar que produce las salidas requeridas, observando un comportamiento adecuado para un rango grande de entradas, incluyendo entradas incorrectas.

Existen dos grandes estrategias en las pruebas: Pruebas de tipo Inductivo ("Bottom Up"), y Deductivo ("Top Down"). Como se mencionó anteriormente las

pruebas de tipo Inductivo comienzan por los módulos más bajos del sistema hasta llegar los módulos superiores del mismo. En el Deductivo primero se prueba el esqueleto externo del sistema, es decir, los niveles más altos de este, en donde los módulos más bajos existen únicamente como fantasmas o "Dummies". Las primeras pruebas que se realizan con una estrategia de tipo Deductivo resultan muy sencillas ya que la mayoría de las funciones detalladas del sistema aún no están implementadas y lo que prueban son las diferentes interfases entre los módulos de mayor jerarquía. Subsecuentemente las pruebas se vuelven más detalladas en la medida que se prueban módulos más profundos del sistema.

De acuerdo al tipo de pruebas existentes, podemos clasificarlas de la siguiente manera:

Pruebas de Funcionalidad: Donde se hace énfasis en que el sistema realice cabalmente las funciones que fueron programadas. Este tipo de pruebas son las más comunes.

Pruebas de Recuperamiento: Donde se producen intencionalmente una serie de errores y se verifica que el sistema se pueda recuperar de estos.

Pruebas de Desempeño: Donde se somete al sistema al procesamiento de volúmenes inmensos de información para conocer su capacidad de acuerdo al rango de datos que maneja.

En el plano hipotético lo más recomendable es hacer pruebas exhaustivas, es decir, presentarle al sistema todas las entradas posibles para todo tipo de condiciones. Esto sin embargo, es imposible, a menos que el sistema sea muy simple es ocioso pensar en poder proporcionar todas las entradas posibles. Entonces es recomendable tener un plan de pruebas que debe contemplar lo siguiente:

Objetivo de la Prueba. Debe responder las siguientes cuestiones: ¿Para qué se hace esta prueba?, ¿Qué enfoque se cubrirá?, horario de las pruebas, contar con un calendario pensado de antemano para las diferentes pruebas

Descripción de las Pruebas, tener una descripción de todo tipo de entradas, en qué condiciones se hicieron, todo esto para poderlas reproducir.

Procedimientos de las Pruebas cómo se van a registrar los diferentes resultados que se conozcan durante las pruebas, cómo será la captura de dichos resultados, cómo se prepararan los datos que serán sometidos a prueba, etcétera.

Instalación

Una vez que las pruebas se han llevado a cabo se tendrán que preparar aspectos concernientes a la instalación del producto; muchas veces ésta es demasiado simple, pero en sistemas muy grandes es un asunto que se debe contemplar con mucha anticipación. Cuestiones como el espacio físico que se requiere en disco duro, cantidad de memoria RAM, tipo de procesamiento (multiprocesamiento, o procesamiento paralelo) tienen que contemplarse con mucha anterioridad a la implementación del sistema. Otros aspectos importantes serán los requerimientos de la plataforma de software necesarios para que el sistema funcione adecuadamente.

Capacitación

El último aspecto en tomar en consideración en la implementación del sistema tiene que ver con la capacitación de los diferentes usuarios. Aspectos como generación de manuales y guías de referencia al igual que clases y seminarios se tendrán que tomar en cuenta. La selección del personal que haga la capacitación, al igual que los horarios en que ésta se realice se tendrán que hacer de acuerdo a una estrategia bien definida con mucha anticipación, atendiendo a las políticas de trabajo dentro de la empresa.

Capítulo III

Analisis del Sistema

Antecedentes

El proceso de Análisis observa una etapa de investigación, dicha etapa tiene como objetivo conocer el significado de lo que una Averiguación Previa representa, de tal manera que se pueda tener un punto de referencia, no sólo para empezar el desarrollo del sistema, sino para poder evaluar sistemas similares o anteriores—Aunque estos sistemas son fuente de información durante esta fase—. Para este propósito, además de estudiar bibliografía al respecto, se tuvo siempre la fortuna de contar con el respaldo de investigadores reconocidos en materia de derecho, mismos que ayudaron tanto en la evaluación de los sistemas anteriores como en el diseño del prototipo. Se sostuvieron mesas redondas donde el grupo de trabajo analizaba el proceso de la averiguación previa y se discutían los alcances que debería tener el sistema. En las reuniones se incluyó la participación de Ministerios Públicos Federales, quienes explicaban las principales necesidades que a éstos se les presentan y lo mucho que les beneficiaría un sistema de esta naturaleza.

De estas reuniones se obtuvo un conocimiento importante que nos familiarizó con diferentes términos y conceptos que se manejan a lo largo del análisis.

La Averiguación Previa

Todas las sociedades cuentan con procedimientos mediante los cuales cierto tipo de conductas, consideradas como reprochables, pueden ser corregidas y castigadas. Dichas conductas, que pudieran presentarse tanto por comisión u omisión, son llamadas delitos. En cuanto a la decisión de castigar o no a los sujetos responsables, le corresponde exclusivamente a autoridades judiciales (en México el Poder Judicial para el fuero común y federal, y tribunales militares para el fuero militar). En el ámbito de los delitos

penales (delitos tipificados de una manera especial en un Código Penal, diversas leyes federales y tratados internacionales) el conjunto de procedimientos en esta actividad judicial se le conoce con el nombre de Proceso Penal.

La preparación del ejercicio de la acción penal, se realiza en la Averiguación Previa, etapa procedimental en la que el agente del Ministerio Público, en ejercicio de sus facultades y apoyado por la Policía Judicial, practica las diligencias necesarias que le permitan estar en aptitud de ejercitar o no la acción penal. La denuncia de un hecho considerado como criminal ante el Ministerio Público (sea denunciado por el ofendido o por cualquier otra persona) es conocida como la noticia del delito (notitia criminis). En cuanto el Ministerio Público tiene conocimiento de un hecho que se pudiera considerar como criminal, sea cual sea el modo saberlo, está obligado a investigar los hechos. Es dicha investigación la que se conoce con el nombre de Averiguación Previa.

La averiguación previa es la primera etapa de la Administración de Justicia Penal Federal (Art. 1 CFPP).

Los principales pilares legales de la averiguación previa y del Ministerio Público Federal como encargado por las leyes para llevarla a cabo, son la Constitución Política de los Estados Unidos Mexicanos en sus artículos 21 y 102, así como la Ley Orgánica del Ministerio Público Federal, el Reglamento de ésta y el Código Federal de Procedimientos Penales.

La etapa de la averiguación previa comprende desde la denuncia o la querrela hasta el ejercicio de la acción penal, con la consignación; con el acuerdo de no ejercicio de la acción penal, con el acuerdo de archivo que da por concluida la averiguación, o con la determinación de reserva, que solamente suspende la averiguación.

El Ministerio Público, además de en los casos de denuncia o querrela, debe también iniciar la averiguación previa cuando éste tenga conocimiento de que se ha cometido un acto que puede ser constitutivo de un delito, y que dicho acto no requiera de un requisito previo para poder proceder, como es el caso de la querrela; entonces actuará de oficio. (art. 123 del CFPP)

La averiguación previa tiene por objeto que el Ministerio Público Federal practique todas las diligencias o actuaciones necesarias para acreditar el elementos del tipo penal del delito de que se trate y la probable responsabilidad del indiciado y estar en

condiciones de consignar los hechos materia de la averiguación a los tribunales y ejercer la acción penal. (arts. 1 y 2 del CFPP)

La acción penal, facultad exclusiva de la institución del Ministerio Público, es el acto jurídico ante los tribunales que da inicio al proceso penal a fin de que estos resuelvan sobre la responsabilidad o irresponsabilidad del indiciado y le sea o no aplicada una pena o una medida de seguridad.

El Ministerio Público Federal, en actuaciones relativas a la averiguación previa puede utilizar todos los medios de prueba que determina la ley. (art. 132 y 206 del CFPP)

Los intervinientes en la averiguación previa

Los participantes en este procedimiento son el Ministerio Público Federal, el indiciado o inculpado, los testigos, la víctima u ofendido y los auxiliares del Ministerio Público Federal.

El Ministerio Público Federal es la autoridad dependiente del Ejecutivo Federal, titular de la facultad investigadora de los delitos, de la procuración de justicia y del poder de ejercicio de la acción penal, que realiza las actuaciones, diligencias y determinaciones necesarias para desarrollar la averiguación previa en el ámbito federal.¹

1. Artículo 21 constitucional dispone: "...La persecución de los delitos incumbe al Ministerio Público y a la Policía Judicial, la cual estará bajo la autoridad y mando inmediato de aquél."

El artículo 102, apartado A, párrafo primero, de la Constitución Federal establece: "La ley organizará el Ministerio Público de la Federación, cuyos funcionarios serán nombrados y removidos por el Ejecutivo, de acuerdo con la ley respectiva, debiendo estar presididos por un Procurador General...; En el párrafo segundo indica: "Incumbe al Ministerio Público de la Federación, la persecución, ante los tribunales, de todos los delitos del orden federal; y, por lo mismo, a él le corresponderá solicitar las órdenes de aprehensión contra los inculcados; buscar y presentar las pruebas que acrediten la responsabilidad de éstos..."

Ley Orgánica de la Procuraduría General de la República en el artículo primero explica que la Procuraduría General de la República es la dependencia del Poder Ejecutivo Federal que integran la institución del Ministerio Público Federal y sus organismos auxiliares directos, para el despacho de los asuntos que a aquélla y a su titular, en su caso, atribuyen los artículos 21 y 102 de la Constitución Federal, la LOPGR y las demás disposiciones legales relativas.

En su artículo segundo la LOPGR dispone: "La institución del Ministerio Público Federal, presidida por el Procurador General de la República, y éste personalmente, en los términos del artículo 102 constitucional, tendrán las siguientes atribuciones que ejercerán conforme a lo establecido en el artículo 10 de esta ley...II. Promover la pronta, expedita y debida procuración e impartición de justicia...V. Perseguir los delitos del orden federal..."

El indiciado o inculpado, como sujeto que realizó una conducta que probablemente pueda ser constitutiva de un delito y que ese delito pudiera ser atribuido a él.

Los testigos, que son los que declaran algún hecho que pudieron percibir por los sentidos y cuyo testimonio apoyará las actuaciones de la autoridad en el sentido de comprobar la responsabilidad o la falta de ella del indiciado y los elementos del tipo penal respectivo.

El ofendido por el delito o la víctima.

La víctima es la persona la sufre un daño o perjuicio en algún bien jurídico de la que es titular y ofendido es la persona que indirectamente sufre tal daño o perjuicio.

No existe una clasificación legal o doctrinal precisa y de la que no haya discusión sobre la naturaleza del ofendido o la víctima, ya que muchas veces se usan indistintamente los términos, y no es considerado una "parte" del proceso penal mexicano sino un coadyuvante de la autoridad. (arts. 141 CFPP y 9o. del CPPDF y demás relativos en los diversos Estados de la Federación).

Un ejemplo de esto es el sujeto pasivo del delito de robo, ya que reúne las calidades de víctima u ofendido a la vez. En cambio en el delito de homicidio queda claro que la víctima es la persona que fue privada de la vida y los ofendidos los familiares de tal persona.

Sería conveniente determinar con precisión la naturaleza del ofendido y la víctima, puesto que del hecho de que una persona sea considerada como víctima u ofendido derivan una serie de derechos y obligaciones para ella y para el Ministerio Público Federal.

Los auxiliares del Ministerio Público Federal son la Policía Judicial Federal y los Servicios Periciales de la Procuraduría General de la República. A estos los clasifica la ley como auxiliares directos.

Como otros auxiliares tiene el Representante Social al Ministerio Público del Fuero Común, a las policías judicial y preventiva, en el Distrito Federal y las Entidades de la República, según los requisitos respectivos; los cónsules y vicecónsules de

nacionalidad mexicana que se encuentren en el extranjero; los capitanes, patrones o encargados de naves y aeronaves nacionales; y, los funcionarios de otras dependencias del Ejecutivo Federal con las condiciones que limita la ley. (art. 14 de la LOPGR)

Ámbito de competencia del Ministerio Público Federal

La Ley Orgánica del Poder Judicial de la Federación en su artículo 51, fracción primera, establece cuales son los delitos de carácter federal, que son los delitos que puede investigar el Representante Social Federal (arts. 21 y 102 de la CPEUM)

Los previstos en las leyes federales y en los Tratados internacionales; Los cometidos en el extranjero por los agentes diplomáticos, personal oficial de las legaciones de la República y Cónsules mexicanos; Los cometidos en las embajadas y delegaciones extranjeras; Aquellos en que la Federación sea sujeto pasivo; Los cometidos por un funcionario o empleado federal, en ejercicio de sus funciones o con motivo de ellas; Los cometidos en contra de un funcionario o empleado federal, en ejercicio de sus funciones o con motivo de ellas; Los perpetrados con motivo del funcionamiento de un servicio público federal, aunque dicho servicio esté descentralizado o concesionado; Los perpetrados en contra del funcionamiento de un servicio público federal o en menoscabo de los bienes afectados a la satisfacción de dicho servicio, aunque éste se encuentre descentralizado o concesionado; Todos aquellos que ataquen, dificulten o imposibiliten el ejercicio de alguna atribución o facultad reservada a la Federación; Los señalados en el artículo 389 del Código Penal, cuando se prometa o se proporcione un trabajo en dependencia, organismo descentralizado o empresa de participación estatal del Gobierno Federal..."

También es competente el Ministerio Público Federal para conocer de los delitos señalados en los artículos 2o. a 5o. del Código Penal de aplicación Federal. (art. 51, fracción I, inciso "b")

El art. 2o. prescribe que se aplicará este Código por los delitos que se inicien, preparen o cometan en el extranjero, cuando produzcan o se pretenda que tenga efectos en el territorio de la República, y por los cometidos en los consulados mexicanos o en contra de su personal, cuando no hubieren sido juzgados en el país en que se cometieron.

El art. 3o. ordena que los delitos continuos cometidos en el extranjero, que se sigan cometiendo en la República, se perseguirán con arreglo a las leyes de ésta, sean

mexicanos o extranjeros los delincuentes. Y más adelante dice que la misma regla se aplicará en el caso de delitos continuados.

El artículo 4o. establece que los delitos cometidos en territorio extranjero por un mexicano contra mexicanos o contra extranjeros, o por un extranjero contra mexicanos, serán penados en la República, con arreglo a las leyes federales, si concurren los requisitos que lista.

Por considerar como cometidos o ejecutados dentro del territorio de la República el artículo 5o. los delitos cometidos por mexicanos o por extranjeros en alta mar, a bordo de buques nacionales; los ejecutados a bordo de un buque de guerra nacional surto en puerto o en aguas territoriales de otra nación. Esto se extiende al caso en que el buque sea mercante, si el delincuente no ha sido juzgado en la nación a que pertenezca el puerto; los cometidos a bordo de un buque extranjero surto en puerto nacional o en aguas territoriales de la República, si se turbare la tranquilidad pública o si el delincuente o el ofendido no fueren de la tripulación. En caso contrario, se obrará conforme al derecho de reciprocidad; los cometidos a bordo de aeronaves nacionales o extranjeras que se encuentren en territorio o en atmósfera o aguas territoriales nacionales o extranjeras. en casos análogos a los que señalan para buques las fracciones anteriores, y los cometidos en las embajadas y delegaciones mexicanas.

Facultades y obligaciones del Ministerio Público durante la averiguación previa:

Dichas atribuciones están determinadas por lo que establecen el artículo 7o. de la Ley Orgánica de la Procuraduría General de la República en lo que a la persecución de los delitos se refiere en el procedimiento; al Código Federal de Procedimientos Penales, en su artículo 2o.; y el artículo 15 del Reglamento de la Ley Orgánica de la Procuraduría General de la República al establecer lo referente a la Dirección General de Averiguaciones Previas.

Esta regulación normativa establece que el MPF deberá:

- Recibir denuncias, acusaciones o querellas, ya sea en forma escrita u oral sobre hechos que puedan constituir delitos de carácter federal, guardando los requisitos y formalidades que establece el artículo 16 constitucional.

- Realizar actuaciones y recabar las pruebas que comprueben el cuerpo del delito [terminología que debe modificarse para estar acorde con la de "elementos del tipo penal" que usa la Carta Magna, pues sólo lo actualiza el CFPP] y la probable responsabilidad del indiciado, con el objeto de fundar el ejercicio de la acción penal.
- El RLOPGR ordena expresamente que integre la averiguación previa, con ayuda de la Policía Judicial Federal y los Servicios Periciales.
- Solicitar a la autoridad jurisdiccional las medidas precautorias de arraigo, aseguramiento o embargo que se requieran para la averiguación previa. (art. 133 bis del CFPP)
- En forma independiente el art. 2 del CFPP, fracción III, lo faculta expresamente para solicitar ordenes de cateo; y, el art. 15 del RLOPGR, también en su fracción III, de manera general, para "adoptar o solicitar a la autoridad judicial" las medidas precautorias que legalmente correspondan.
- Cuando el Ministerio Público Federal tenga conocimiento por sí o por medio de sus auxiliares de la probable comisión de un delito cuya persecución dependa de una querrela u otro acto similar que deba formular una autoridad, lo comunicará de inmediato a ésta para resuelva lo que según sus atribuciones corresponda y cuya determinación deberán avisar al Ministerio Público Federal por escrito. Las autoridades que deban resolver sobre si ejercitan o no la querrela o requisito similar tendrán cuarenta y ocho horas para resolver lo conducente en el caso de que el Representante Social tenga detenidos a su disposición. (Art. 7 de la LOPGR).
- Actos conducentes a la protección del ofendido por el delito en los términos de ley. (art. 7 de la LOPGR) Pero no determina exactamente si es en su persona, en sus derechos o en ambos.
- Acordar el no ejercicio de la acción penal y notificarlo al ofendido o víctima, resolviendo, en su caso, la inconformidad que formulen éstos. (art. 2 del CFPP, fracción VIII)
- Asegurar o restituir al ofendido en sus derechos en los términos del artículo 38 del CFPP. (art. 2 del CFPP, fracción VI)

- Dictar medidas y providencias para proporcionar auxilio y seguridad a las víctimas del delito. (Arts. 20, penúltimo y último párrafos, de la CPEUM y 123 del CFPP)
- Realizar actos para la comprobación de los elementos que pueden llevar a solicitar ante la autoridad jurisdiccional la reparación del daño (Art. 2 del CFPP, fracción II, *in fine*).
- Acordar la detención o retención de los indiciados cuando así proceda (art. 16 de la CPEUM, arts. 2 y 123 fracción IV del CFPP).
- Resolver casos de incompetencia, acumulación de averiguaciones y los demás procedan legalmente durante la averiguación previa (art. 15 del RLOPGR, fracción IV)
- Ejercitar la acción penal. (art. 2 del CFPP, art. 7 de la LOPGR y art. 15 del RLOPGR, fracción IV)
- Recibir elementos de prueba presentados por los indiciados o sus representantes para la integración de la averiguación previa (artículo 20, fracción V y penúltimo y último párrafos de la Constitución Federal y art. 15 del RLOPGR, fracción II)
- Promover la conciliación de las partes en los casos pertinentes (art. 2 del CFPP, fracción X)
- Conceder o revocar la libertad provisional del indiciado, en los casos que la ley lo permita (Artículo 20, fracción I y penúltimo y último párrafos de la Constitución Federal y art. 2 del RLOPGR, fracción IX).
- Turnar a la instancia competente los expedientes con el acuerdo, debidamente fundado y motivado, de los casos de archivo por el no ejercicio de la acción penal o de los casos de reserva (art. 15 del RLOPRG, fracción V). El CFPP faculta al Representante Social para "Determinar la reserva o el ejercicio de la acción penal" (art. 2, fracción VII), pero dentro de las atribuciones que enumera no está la de el archivo, que se encuentra ubicado dentro del mismo Código en otro artículo expreso.

- Obtener de las áreas de control de procesos, la información del resultado del ejercicio de la acción penal y de los procesos instruidos. (art. 15 del RLOPGR, fracción VI).
- Auxiliar por instrucciones superiores, o a petición del Director General de Asuntos Legales Internacionales, en el levantamiento de actas o constancias sobre hechos que legítimamente deban ser conocidos por autoridad investida de fe pública o en cumplimiento de pedimento expreso de autoridad extranjera, para la práctica de diligencias solicitadas con motivo de algún tratado de asistencia mutua, de las que podrán expedirse copias certificadas bajo los lineamientos establecidos para esos efectos, y en su oportunidad deberán ser archivados en la forma que proceda. (art. 15 del RLOPGR, fracción VII)
- Recibir las recomendaciones de la Comisión Nacional de Derechos Humanos y los casos de amigable composición que les sean enviados y turnarlos a la autoridad competente, dándole la atención, seguimiento y evaluación hasta su total cumplimiento. (art. 15, fracción VIII, del RLOPGR)
- Formular ante la autoridad jurisdiccional, al ejercitar la acción penal, los pedimentos que correspondan. (art. 7 de la LOPGR).
- El art. 2 del CFPP y el 15 del RLOPGR en la última de sus correspondientes fracciones le dan al Ministerio Público Federal la posibilidad de adicionar como fundamento de sus actuaciones en la averiguación previa todas aquellas facultades que les otorguen otras disposiciones legales.

Existen otras facultades, no menos importantes, y que se encuentran a través de las normas que hemos estado citando. Como muestra están las siguientes:

- Dentro de las actuaciones de la averiguación previa deberá impedir que se pierdan, destruyan o alteren las huellas o vestigios del hecho delictuoso, los instrumentos o cosas objeto o efectos del mismo; saber qué personas fueron testigos; evitar que el delito se siga cometiendo y, en general impedir que se dificulte la averiguación, procediendo a la detención de los que intervinieron en su comisión en los casos de delito flagrante. (art. 123 del CFPP)
- Nombrar un traductor para aquellas personas que no hablen o entiendan suficientemente el castellano. (art. 124 del CFPP)

- Al inicio de sus actuaciones levantará una acta que contenga los datos que ordena la ley para el caso. (art. 124 del CFPP)
- Requerir informes, documentos, opiniones y elementos de prueba, en general, a las dependencias y entidades de la administración pública federal, a las correspondientes al Distrito Federal, y a otras autoridades o personas que puedan suministrar elementos para el debido ejercicio de sus atribuciones. Esta facultad también la tienen sus auxiliares en los casos que marquen las leyes. (art. 11 de la LOPGR)
- Expedir constancias de actuaciones o registros que obren en su poder cuando exista mandamiento de autoridad competente que funde y motive su requerimiento, o cuando resulte indispensable la expedición de dichas constancias para el ejercicio de derechos o el cumplimiento de obligaciones previstos en la ley. La misma obligación se establece para la Policía Judicial Federal. (art. 32. LOPGR)
- Citar a declarar a las personas que por cualquier causa participen o tengan relación en los hechos o tengan conocimiento de ellos.
- En caso de la detención de extranjeros comunicará su detención inmediatamente a la representación diplomática o consular correspondiente. (art. 128, fracción IV)
- Hacer saber al detenido o presentado ante él los derechos y garantías que las leyes y la Constitución Federal le otorgan, sobre todo para el caso de la averiguación previa, y dejar constancia de ello en el acta. (art. 128 del CFPP)
- Comunicar a los responsables de las instituciones de salud donde se interne a alguna persona el carácter con el que ingresa ésta, pues de ser el caso sólo podrá salir con la autorización correspondiente del Ministerio Público Federal. (art. 129 del CFPP)
- Expedir órdenes para autopsia e inhumación de cadáveres y el levantamiento de las actas de defunción respectivas y fuere probable que la muerte de la persona fuere consecuencia de un delito o en caso contrario ordenar sólo el levantar las actas inhumar el cuerpo. (art. 130 del CFPP)

Los auxiliares del Ministerio Público Federal también tienen facultades para realizar algunas actuaciones de las arriba mencionadas en ausencia del éste. (arts. 3o, 123 y 126 del CFPP)

Formas de concluir la averiguación previa

El ejercicio de la acción penal

Si en la averiguación previa se han acreditado los elementos del tipo penal del delito y la probable responsabilidad del indiciado el Ministerio Público ejercitará la acción penal ante los tribunales. (art. 134 del CFPP)

En ejercicio de la acción penal, corresponde al Ministerio Público: Promover la incoación del proceso penal; Solicitar las órdenes de comparecencia para preparatoria y las de aprehensión, que sean procedentes; Pedir el aseguramiento precautorio de bienes para los efectos de la reparación del daño; Rendir las pruebas de la existencia de los delitos y de la responsabilidad de los inculpaos; Pedir la aplicación de las sanciones respectivas; y hacer todas las promociones que sean conducentes a la tramitación regular de los procesos. (art. 136 del CFPP)

El NO ejercicio de la acción penal.

El Ministerio Público no ejercitará la acción penal si la conducta o hechos no constituyen delitos conforme a lo descrito en la ley penal; si se acreditó que el inculpaado no tuvo participación en los hechos delictuosos; si aún siendo delictivos los hechos o actos no pueda probarse su existencia por un obstáculo insuperable; si está extinguida la responsabilidad penal según la ley; si se comprueba en las diligencias que el inculpaado se encontraba en algunas de las circunstancias que excluyen del delito. (art. 137 del CFPP).

En estos casos el Ministerio Público Federal solicitará el sobreseimiento del asunto y promoverá la libertad del inculpaado y la resolución que recaiga tendrá el efecto de "impedir definitivamente el ejercicio de la acción penal respecto de los hechos que las motiven" (arts. 138 y 139 del CFPP). El querellante, el denunciante o el ofendido podrá poner este procedimiento a consideración del Procurador General de la República el que, a través de sus Agentes Auxiliares, decidirá en definitiva si debe o no ejercitarse acción penal (arts. 133, 140, 294 y 295 del CFPP).

El acuerdo de reserva

Otra de las formas de concluir la averiguación previa consiste en la determinación de reserva, que se da cuando de las diligencias practicadas no resultan elementos bastantes para hacer la consignación a los tribunales y no aparece que se puedan practicar otras, pero con posterioridad pudieran allegarse datos para proseguir la averiguación; entonces se **reservará** el expediente hasta que aparezcan esos datos, y entretanto se ordenará a la policía que haga investigaciones tendientes a lograr el esclarecimiento de los hechos. (art. 131 del CFPP)

El acuerdo de archivo

Esta institución jurídica no suspende la averiguación previa, como la determinación de reserva, sino que termina definitivamente con ella y da una mayor seguridad jurídica a los inculcados de un delito. El procedimiento cesará y se mandará archivar el expediente en los casos de procedencia del sobreseimiento relativos, que son: Si el Procurador General de la República confirma o formula conclusiones no acusadoras; Cuando el Ministerio Público lo solicite en los casos que determina la ley; si aparece que la responsabilidad penal está extinguida o prescrita; cuando estando agotada la averiguación previa se compruebe que no existió el hecho delictuoso que la motivó.

Estas primeras cuatro razones de sobreseimiento podrán hacerse valer a petición de parte o de oficio y las siguientes dos sólo a petición de parte (art. 300 del CFPP): Cuando, habiéndose decretado la libertad por desvanecimiento de datos, esté agotada la averiguación y no existan elementos posteriores para dictar nueva orden de aprehensión, o se esté en el caso previsto por la parte final del artículo 426; y, cuando esté plenamente comprobado que en favor del inculcado existe alguna causa de exclusión del delito. (arts. 298 y 299 del CFPP)

Revisión de Otros Sistemas

Uno de los problemas que existe en el Sector Público es la falta de continuidad en diferentes proyectos que se presentan cuando la administración cambia. Es muy común

no tener la comunicación necesaria para conocer proyectos anteriores; y por lo tanto, no es extraño que muchas veces se pierdan esfuerzos valiosos que se pudieron haber aprovechado. Teniendo en cuenta este problema, se decidió investigar la existencia de sistemas de averiguaciones previas anteriores en la Procuraduría General de la República.

Se encontraron dos sistemas de Averiguaciones Previas, uno desarrollado por el Instituto Nacional de Ciencias Penales (I.N.A.C.I.P.E.) y otro desarrollado en la Subprocuraduría de Delegaciones. Se decidió evaluar dichos sistemas con la ayuda de expertos en materia de derecho.

Sistema Integral Automatizado para la Procuración de Justicia (INACIPE)

El sistema evaluado presenta varias ventajas. Entre ellas el uso exhaustivo de datos que el agente del M. P. debe tener presente en el momento de realizar una averiguación previa. Sin embargo, el sistema se revela muy poco atractivo; pero, sobre todo, poco útil. Los diferentes "pasos" del sistema no son sino campos de un formulario que podrían aparecer en un sólo despliegue de pantalla. Hace un listado exhaustivo de cuestiones tales como colonias, barrios, municipios, etcétera; mecanismo que resulta lento y repetitivo. Es mejor asentar directamente los datos y después, si es el caso, verificarlos cartográficamente.

El sistema es erróneo, ya que en ocasiones inventa o duplica pasos (Por ejemplo: comunicación del delito e inicio de la averiguación). De esta forma el esquema de las funciones del Ministerio Público en la fase de la averiguación, básicamente simple, se hace profuso y confuso.

Ciertamente es necesario recordarle al agente del Ministerio Público todas las cuestiones que hay que asentar en el expediente. Para esto basta el listado de descriptores que precedan campos que se vayan progresivamente llenando con información, si la hay.

Sistema realizado por la Subprocuraduría de Delegaciones

Este sistema tiene una mejor idea. la participación de los desarrolladores con agentes del Ministerios Públicos es evidente. Tiene una mejor división de las etapas que componen

la Averiguación Previa y es especialmente rescatable el módulo de aseguramientos que presenta. Sin embargo, existen muchas restricciones que hacen muy difícil la integración de una Averiguación Previa en el sistema, existen duplicidad de datos, un manejo muy complicado que de los escritos de la averiguación previa. En el plano práctico no ofrece al Ministerio Público ninguna ventaja, más bien complica su trabajo. En el plano técnico el sistema es deplorable, se presentan frecuentemente defectos en su programación tiene una estructura muy poco sólida que puede producir pérdidas de información.

En realidad las etapas de la averiguación previa son tres: La primera es el conocimiento de los hechos, la segunda las actuaciones del agente del Ministerio Público tendientes a esclarecer los hechos; de tal forma que habrá declaraciones de denunciantes, de ofendidos, de testigos y, por otro lado, acciones realizadas por el agente del Ministerio Público. La tercera fase es el resultado de la averiguación. Dentro de estas opciones debe crearse un espacio para que se capturen declaraciones, testimonios, citatorios, peritajes, medidas cautelarias, calificaciones de los hechos delictivos, etcétera.

Objetivos del Sistema Automatizado de Averiguaciones Previas

- Facilitar el trabajo del Ministerio Público Federal Investigador y personas que lo auxilian en la creación de una Averiguación Previa.
- Fuente de datos para alimentar las bases de datos de las delegaciones y del organismo central (PGR).
- Agilizar la generación de informes estadísticos y de esta forma no distraer de sus funciones prioritarias a las agencias del Ministerio Público Federal.
- Optimar el tiempo en que se ejecutan labores administrativas.
- Ser piedra angular de un sistema integral de apoyo al Ministerio Público Federal.

Diagrama Contextual

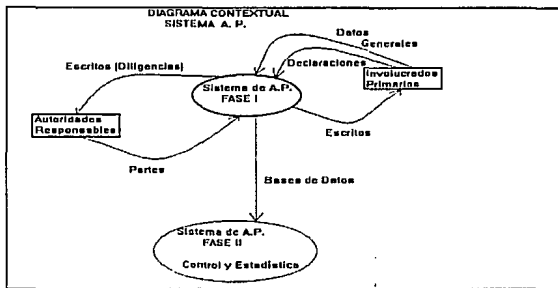


Fig. 2.1 Diagrama Contextual. Obsérvese la relación del sistema con entidades externas al mismo.

En la Fig. 2.1 se observa la interrelación que existe entre el sistema y otras entidades externas (la segunda fase del mismo sistema se considera como una entidad externa). Como se puede observar existen dos tipos de entidades externas los involucrados primarios (denunciantes, testigos e indiciados) y autoridades responsables que pueden ser la policía judicial, peritos etc. Se observa que el sistema genera escritos para dichas entidades externas y a su vez las entidades producen información al sistema como declaraciones, escritos y partes.

El sistema genera bases de datos en todos sus procesos (representados por burbujas) que serán utilizadas en el sistema de Control y Estadística. En las siguientes figuras será omitido el flujo de las bases de datos pero es de entenderse que todos los procesos generan dichas bases.

Lista de Eventos

1. El M. P. toma conocimiento de un hecho considerado como delictuoso.
2. El M. P. genera una averiguación previa.
3. El M. P. toma los datos generales y la declaración del denunciante o denunciantes.
4. El M. P. toma los datos generales y la declaración del testigo o los testigos.
5. El M. P. toma los datos generales y la declaración del indiciado o indiciados.
6. El M. P. asocia delitos al indiciado o indiciados.
7. El M. P. genera un *acuerdo* de la averiguación previa..
8. El M. P. genera *diligencias* a partir del acuerdo.
9. El M. P. registra las *providencias cautelarias* realizadas en la averiguación previa.
10. El M. P. envía a *consulta* la averiguación previa.
11. El M. P. recibe la respuesta a una o varias diligencias.
12. El M. P. genera la *determinación* de la averiguación previa.
13. El M. P. busca una averiguación previa a partir del nombre de indiciado.
14. El M. P. busca una averiguación previa a partir del nombre de denunciante.
15. El M. P. busca una averiguación previa a partir del nombre de testigo

Diagramas de Flujo de Datos

Sistema Automatizado de Averiguaciones Previas

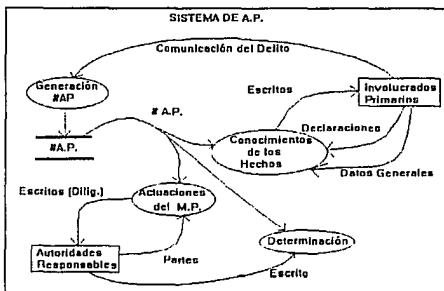


Fig. 2.2 DFD del Sistema Automatizado de Averiguaciones Previas.

El Sistema Automatizado de Averiguaciones Previas está formado básicamente por tres módulos que modelan el proceso que sigue una averiguación previa y un módulo complementario donde se genera el número de A. P., el cual liga los módulos principales de todo el sistema. El primero es el de **Conocimiento de los Hechos**, en el cual el agente del Ministerio Público toma conocimiento de la comisión de un delito; en esta etapa se tomarán, en su caso, declaraciones y datos generales de denunciantes, testigos e indiciados. El segundo es el de **Actuaciones del M. P.**, donde el agente del M. P. realiza ciertas acciones (diligencias) encaminadas a esclarecer los hechos; estas acciones implican, en la mayoría de los casos, la cooperación de otras autoridades, quienes a su vez deben dar respuesta oportuna a las peticiones del agente del M. P. a través de lo que se conoce como parte. La etapa final del proceso de A. P. es conocido como **determinación**, en donde el agente del M. P. llega a una conclusión basándose en la investigación que se ha realizado.

Conocimiento de los hechos

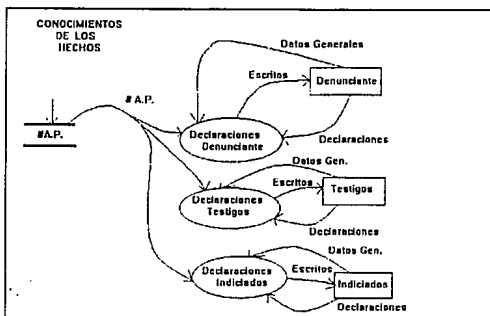


Fig. 2.3 DFD del módulo Conocimiento de los hechos.

La Fig. 2.3 representa el proceso denominado **Conocimiento de los hechos** al que se hace referencia en la Fig. 2.2. Como se observa, éste módulo se forma a su vez de tres módulos muy parecidos (declaraciones de denunciante, declaraciones de testigos, declaraciones de indiciados); en cada uno de estos procesos se generan las bases de datos formadas por las diferentes declaraciones y datos generales de los involucrados primarios. Todos estos procesos usan el número de averiguación previa generado por procesos anteriores. A su vez cada uno de estos procesos puede generar escritos que no son otra cosa que la impresión de dichas declaraciones.

Actuaciones del M. P.

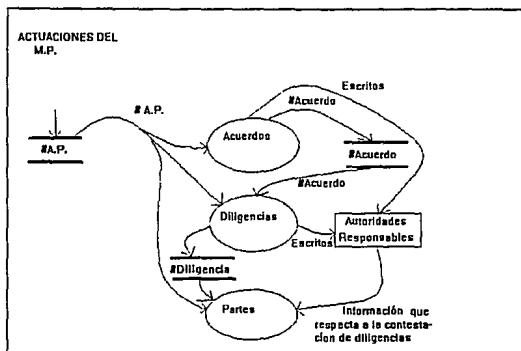


Fig. 2.4 DFD del módulo Actuaciones del M. P.

La Fig. 2.4 representa el proceso denominado como **Actuaciones del M.P.** al que se hace referencia en la Fig. 2.2. Como se observa dicho proceso se forma a su vez de tres procesos (Acuerdos, Diligencias y Partes), todos estos procesos usan el número de averiguación previa generado por procesos anteriores. El proceso de acuerdos genera tanto escritos para las autoridades correspondientes como un número particular por cada acuerdo. El proceso de diligencia genera los escritos para las autoridades correspondientes; usa el número de acuerdo y a su vez genera un número de diligencias. El proceso de partes toma información relevante como el número de diligencia también de las autoridades correspondientes toma la información que tiene que respecta a la contestación de diligencias.

Determinación

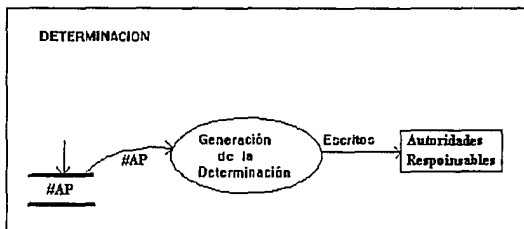


Fig. 2.5 DFD del módulo Determinación.

La Fig. 2.5 representa el proceso denominado como **Determinación** al que se hace referencia en la Fig. 2.2. Se observa que usa el número de averiguación previa y que genera escritos para las autoridades correspondientes.

Declaración del Denunciante

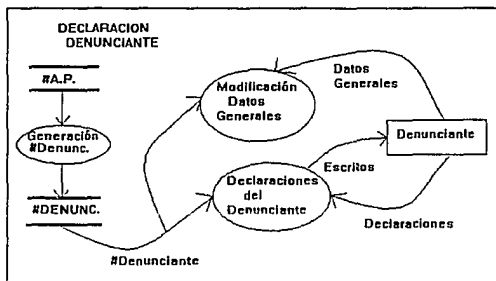


Fig. 2.6 DFD del módulo Declaración del denunciante.

En la Fig. 2.6 se representa el proceso denominado **Declaración del Denunciante** al que se hace referencia en la figura 2.3. Se observa que se forma de tres procesos (Generación de Número de Denunciante, Modificación de Datos

Generales, y Declaración del denunciante) y que también usa el número de averiguación previa generada en procesos pasados. La Generación del #Denunciante genera un número particular por cada denunciante que se usa en los procesos inmediatos, Modificación de Datos Generales genera o modifica los datos generales de un denunciante en particular. Declaración del Denunciante generan las bases de datos correspondientes a las diferentes declaraciones y puede generar los escritos para dichas declaraciones.

Declaración del Testigo

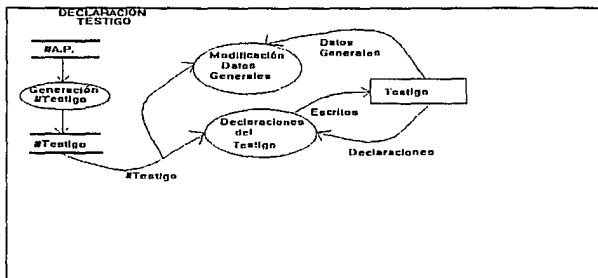


Fig. 2.7 DFD del módulo declaración del testigo.

La Fig. 2.7 es casi idéntica a la Fig. 2.6 con la única diferencia de que existe un proceso de generación de número de testigo en lugar de número de denunciante.

Declaración del Indiciado

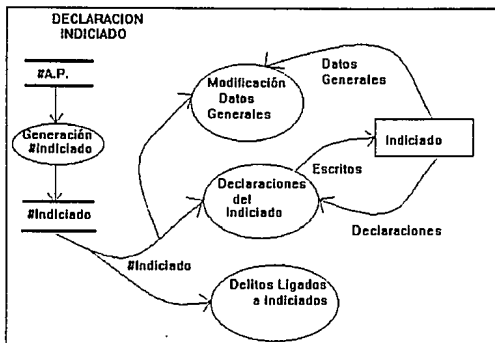


Fig. 2.8 DFD del módulo Declaración del indiciado.

La explicación de la Fig. 2.8 es también muy parecida a la explicación de la Fig. 2.6. Las diferencias son que existe un proceso de generación de número de testigo en lugar de número de denunciante y que existe un proceso más (Delitos ligados a Indiciados) donde se generan las bases de los delitos que se ligan a diferentes indiciados.

Acuerdos

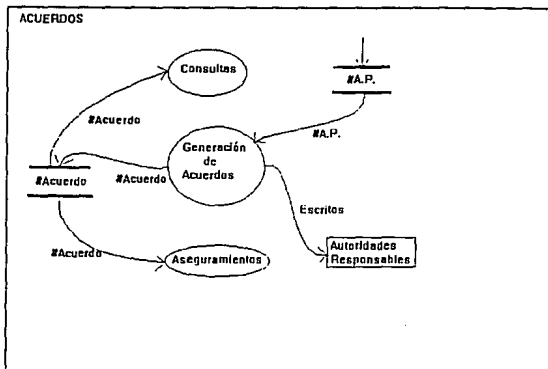


Fig. 2.9 DFD del módulo Acuerdos.

La Fig. 2.9 corresponde al proceso referido en la figura 2.4 (Acuerdos); como se observa está formado por tres procesos, el más importante de todos es la Generación de Acuerdos donde tomando un número de A.P. se puede generar un número de acuerdo. En este proceso se genera el número que corresponde al acuerdo particular. se pueden generar escritos para las autoridades correspondientes. El proceso de consulta genera información a la base de datos referente al estado de la averiguación previa con respecto a consultas. El proceso de Aseguramiento obtiene la información para las bases de datos con respecto al tipo de aseguramientos (si es que este existe).

Diligencias

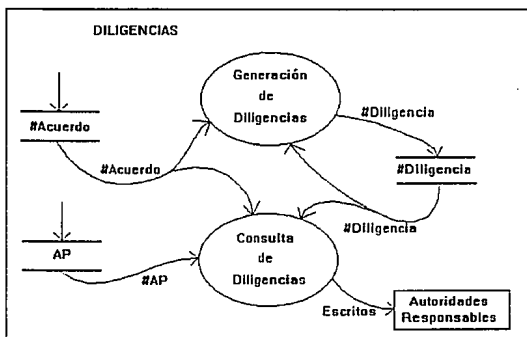


Fig. 2.10 DFD del módulo de Diligencias.

El proceso mostrado en la Fig. 2.10 corresponde al referido como **Diligencias** de la figura 2.4. Consta de dos procesos (Generación de Diligencias y Consulta de Diligencias). En la generación de diligencias se genera el número particular de diligencia propia de cada una, también se pueden generar los escritos y consultar las diligencias ligadas a un Acuerdo. El proceso de consulta genera los escritos y consultas de diligencias ligadas a una Averiguación Previa.

Aseguramientos

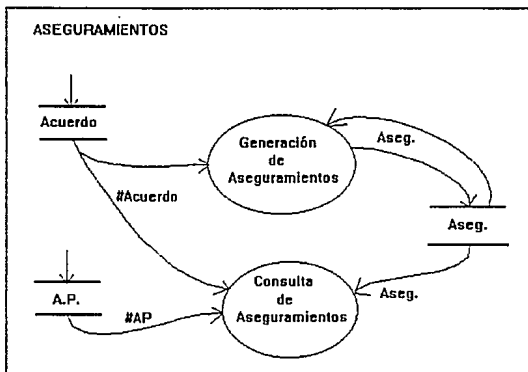


Fig 2.11. DFD del módulo Aseguramientos

EL proceso mostrado en la Fig. 2.11 corresponde al referido como **Aseguramientos** de la figura 2.9. Consta de dos procesos (Generación y Consulta). En la generación de aseguramientos se crea un número único de aseguramiento; también se puede consultar los aseguramientos ligados a un acuerdo determinado. El proceso de consulta muestra los aseguramientos ligados a una Averiguación Previa.

Partes

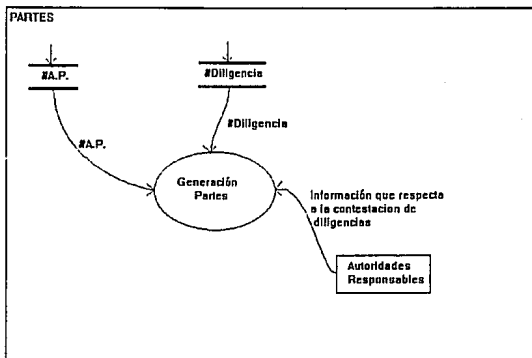


Fig. 2.12 DFD del módulo de Partes.

El proceso de la Fig. 2.12 corresponde al referido como **Partes** de la figura 2.4. Consta de un solo proceso (Generación Partes). En dicho proceso se genera la información para la base de datos que respecta a la contestación de diligencias (Partes) para este proceso se obtiene dicha información de Autoridades Correspondientes y se utilizan para ligarlo a otros el número de averiguación previa y el número de diligencia.

Especificación de Procesos

EMPIZA PROCEDIMIENTO Generación Partes
PRECONDICION Existencia de AP Relacionada
MIENTRAS Cierto * Siempre para una misma AP*
Activa Despliega CveDil
SI CveDil Desplegada
Despliega texto
Despliega Pendiente
SI CveDil Seleccionada
SI información con respecto a la contestación de esta Diligencia existe
Pendiente asigne el mismo valor u otro
Activa Despliega NumParte y Opcion Nueva
SI NumParte Desplegada
Despliega Texto
SI Opcion Nuevo Seleccionada
Generar Parte con Valores Ap y CveDil heredados
SI NumParte Seleccionada
Despliega Texto
Activa opción de imprimir
FIN SI
FIN SI
FIN SI
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación de Diligencia
PRECONDICION Existencia de Ap y Acuerdo relacionado
MIENTRAS Cierto *Siempre para una misma AP y Acuerdo*
Activa Opción Nuevo y Despliega CveDil * Solamente diligencias del Acuerdo*
SI CveDil Desplegada
Despliega texto
SI CveDil Seleccionada
Despliega Texto
Activa Opción Imprimir
FIN SI
SI Opción Nuevo Seleccionada
Genera Diligencia con Avprev y #Ac heredados
FIN SI
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Consulta de Diligencia
PRECONDICION Existencia de Ap y Diligencia Relacionada
MIENTRAS Cierto *Siempre para una misma AP *
Activa Despliega CveDil * Diligencias de la AP*
SI CveDil Desplegada
Despliega texto
SI CveDil Seleccionada
Despliega Texto
Activa Opción Imprimir
FIN SI
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación de Aseguramientos
PRECONDICION Existencia de Ap y Acuerdo relacionado
MIENTRAS Cierto *Siempre para una misma AP y un mismo Acuerdo*
Activa Opción Nuevo y Despliega Desp_Aseg * Solamente diligencias del Acuerdo*
SI Opción Nuevo Seleccionada
Genera Aseg con #AP y #Acuerdo heredados
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Consultade Aseguramientos
PRECONDICION Existencia de Ap relacionado
MIENTRAS Cierto *Siempre para una misma AP*
Activa Despliega Desp_Aseg * Solamente diligencias del Acuerdo*
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Consulta
PRECONDICION Existencia de AP relacionada
MIENTRAS Cierto *Siempre par una misma AP*
Despliga opción Consulta
SI Opción Consulta Seleccionada
Consulta se le asigna un valor verdadero
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación de Acuerdo
PRECONDICION Existencia de Ap relacionado
MIENTRAS Cierto *Siempre para una misma AP*
Activa Opción Nuevo y Despliega #Acuerdo

Si #Acuerdo Desplegada
Despliega texto
SI #Acuerdo Seleccionado
Despliega Texto
Activa Opción Imprimir
FIN SI
SI Opción Nuevo Seleccionada
Genera Acuerdo con Avprev heredada
FIN SI
FIN SI
Activa Opción Consulta, Aseguramiento, Diligencia
SI Opción Consulta Seleccionada
Activa PROCEDIMIENTO Consulta
FIN SI
SI Opción Diligencia Seleccionada
Activa PROCEDIMIENTO Generación de Diligencias
FIN SI
SI Opción Aseguramiento Seleccionada
Activa PROCEDIMIENTO Generación de Aseguramientos
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación #Testigo
PRECONDICION Existencia de AP relacionada
MIENTRAS Cierto *Siempre para una misma AP*
Activa Opción Nuevo y Despliega Nombre *de Testigo*
SI Nombre seleccionado
Activa Opción Datos Generales y Declaraciones
SI Opción Datos Generales Seleccionada
Activa PROCEDIMIENTO Datos Generales del Testigo
FIN SI
SI Declaraciones Seleccionada
Activa PROCEDIMIENTO Declaraciones del Testigo
FIN SI
FIN SI
SI Opción Nuevo Seleccionada
Genera IdenDato con Avprev heredada
Activa PROCEDIMIENTO Datos Generales del Testigo
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Datos Generales del Testigo
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS Cierto *Siempre para un mismo IdenDato*
Completa Datos Generales * del Testigo*
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Declaraciones del Testigo
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS CIERTO *Siempre para un mismo IdenDato*
Activa Opción Nuevo y Despliega FechaDec
SI FechaDec Desplegada
Despliega Texto
SI FechaDec Seleccionada
Despliega Texto
Activa Opción de Imprimir
FIN SI
SI Opción Nuevo Seleccionada
Genera Declaración con IdenDato heredado
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación #Denunciante
PRECONDICION Existencia de AP relacionada
MIENTRAS Cierta *Siempre para una misma AP*
Activa Opción Nuevo y Despliega Nombre *de Denunciante*
SI Nombre seleccionado
Activa Opción Datos Generales y Declaraciones
SI Opción Datos Generales Seleccionada
Activa PROCEDIMIENTO Datos Generales del Denunciante
FIN SI
SI Declaraciones Seleccionada
Activa PROCEDIMIENTO Declaraciones del Denunciante
FIN SI
FIN SI
SI Opción Nuevo Seleccionada
Genera IdenDato con Avprev heredada
Activa PROCEDIMIENTO Datos Generales del Denunciante
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Datos Generales del Denunciante
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS Cierta *Siempre para un mismo IdenDato*
Completa Datos Generales * del Denunciante*
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Declaraciones del Denunciante
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS CIERTO *Siempre para un mismo IdenDato*
Activa Opción Nuevo y Despliega FechaDec
SI FechaDec Desplegada
Despliega Texto
SI FechaDec Seleccionada
Despliega Texto
Activa Opción de Imprimir
FIN SI
SI Opción Nuevo Seleccionada
Genera Declaración con IdenDato heredado
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación #Indiciado
PRECONDICION Existencia de AP relacionada
MIENTRAS Cierto *Siempre para una misma AP*
Activa Opción Nuevo y Despliega Nombre *de Indiciado*
SI Nombre seleccionado
Activa Opción Datos Generales , Delitos y Declaraciones
SI Opción Datos Generales Seleccionada
Activa PROCEDIMIENTO Datos Generales del Indiciado
FIN SI
SI Declaraciones Seleccionada
Activa PROCEDIMIENTO Declaraciones del Indiciado
FIN SI
SI Delitos Seleccionado
Activa PROCEDIMIENTO Delitos Ligados a Indiciado
FIN SI
FIN SI
SI Opción Nuevo Seleccionada
Genera IdenDato con Avprev heredada
Activa PROCEDIMIENTO Datos Generales del Indiciado
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

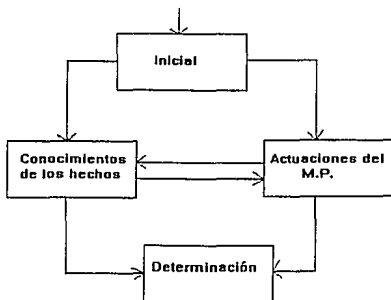
EMPIEZA PROCEDIMIENTO Datos Generales del Indiciado
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS Cierto *Siempre para un mismo IdenDato*
Completa Datos Generales * del Indiciado*
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Declaraciones del Indiciado
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS CIERTO *Siempre para un mismo IdenDato*
Activa Opción Nuevo y Despliega FechaDec
SI FechaDec Desplegada
Despliega Texto
SI FechaDec Seleccionada
Despliega Texto
Activa Opción de Imprimir
FIN SI
SI Opción Nuevo Seleccionada
Genera Declaración con IdenDato heredado
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Delitos Ligados a Indiciado
PRECONDICION Existencia de IdenDato relacionado
MIENTRAS CIERTO *Siempre para un mismo IdenDato*
Activa Opción Nuevo y Despliega CveDel
SI CveDel Desplegada
Despliega NomDel
SI CveDel Seleccionada
Despliega NomDel
Despliega LeyDel
FIN SI
SI Opción Nuevo Seleccionada
Genera Delito con IdenDato heredado
FIN SI
FIN MIENTRAS
FIN PROCEDIMIENTO

EMPIEZA PROCEDIMIENTO Generación de la Determinación
 PRECONDICION Existencia de una AP relacionada
 MIENTRAS Cierto *Siempre para una misma AP*
 Activa Opción Nuevo y Despliega FechaDet
 SI FechaDet Desplegada
 Despliega Texto
 SI FechaDet Seleccionada
 Despliega Texto
 Activa Opción Imprimir
 FIN SI
 SI Opción Nuevo Seleccionada
 Genera Determinación con Avprev heredada
 Pregunta tipo *El tipo de la Determinación*
 SI tipo es igual Consignado con detenido
 MIENTRAS Cierto *Para una misma AP
 Activa Despliega Nombre *Indiciado para una AP*
 SI nombre Desplegado
 Desplegar EstCons *Consignado o no*
 SI nombre Seleccionado
 Pregunta EstCons
 SI EstCons es cierto
 MIENTRAS Cierto
 Activa Opción Nuevo y
 Despliega CveDel
 SI CveDel Desplegada
 Despliega EstCons *del
 delito*
 SI CveDel Seleccionada
 Pregunta EstCons
 FIN SI
 SI Opción Nuevo Seleccionada
 Genera Delito con IdenData
 heredado
 EstCons se le asigna verdadero
 FIN SI
 FIN MIENTRAS
 FIN SI
 FIN MIENTRAS
 FIN SI
 FIN SI
 FIN SI
 FIN MIENTRAS
 FIN PROCEDIMIENTO

Diagrama de Transición de Estados del Sistema SAAP



El diagrama de transición de estados del sistema distingue cuatro estados principales: un estado inicial, dos intermedios (Conocimientos de los Hechos y Actuaciones del M.P.), y uno final (Determinación). El estado inicial nace cuando se presenta un hecho, considerado como delictuoso, y dicho hecho es del conocimiento del M.P.; en esos momentos teóricamente inicia una Averiguación Previa. Posteriormente el M.P. puede proceder a registrar las diferentes declaraciones de los diferentes denunciante y testigos como sus datos generales, todo esto con el objetivo de obtener un esclarecimiento de los hechos (esto se representa en la figura en el Estado de Conocimiento de los Hechos). El M.P. puede realizar diferentes acciones en su carácter de autoridad como peticiones de peritajes, de orden de aprehensión etc. dichas acciones el sistema las almacena (se resaltan en el estado de Actuaciones del M.P.). Cabe mencionar que tanto el Estado de Conocimiento de los Hechos como el Estado de Actuaciones del M.P. se pueden llevar a cabo en forma paralela como de hecho se muestra en la figura. La última etapa se refiere a la Determinación, la cual consiste en un escrito en la que se define la resolución de una A.P.

Diccionario de Datos del Sistema de Averiguaciones Previas

#Acuerdo = Fecha_Ac+ Hora_Ac

#Ap = Avprev

Acuerdo= Avprev + #Acuerdo + Texto * corresponde a la información del acuerdo*

Agenc= 2{número}2 * corresponde al número de clave de la agencia respectiva*

Alias= 1{caracter}35 * representa un pseudónimo de la persona*

Año = [2{número}2] *corresponde al número de año se expresa con dos números*

AP= Avprev + MP + Fecha_AP + Hora_Ap * corresponde a la información principal de la AP*

Apellido paterno= 1{caracteres}25

Aseg= Avprev + #Acuerdo+ Tipo + Desp_Aseg

Autoridades Responsable * Entidades Externas que proporcionan información al sistema*

Avprev= Deleg + Agenc + Mesa + Consec + Año + Mes * Clave de Averiguación Previa*

Calle = 1{caracter}35 * corresponde a la calle y número de la dirección*

Cantidad = 1{número}3 * cantidad de lo asegurado*

Caracteres= [A-Z|a-z|número|'|á|é|i|ó|ú|ü|]

Cargo= 1{caracter}35 * representa el cargo de la persona si dicha persona es funcionario*

Clasificación = 1{caracter}20 * nombre de lo asegurado*

Colonia = 1{caracter}25 * corresponde al nombre de la colonia en la dirección*

Comunicación del Delito * acto mediante el cual es sistema genera una nueva averiguación o toma una ya existente*

Consec=3{número}3 * corresponde a un número consecutivo que se reinicializa cada mes*

Consulta = caracter *Caracter lógico que determina si una A.P. esta o no en consulta*
 CP= 5{número}5 * corresponde al número del código postal*
 Cve_Nom = 5{caracter}5 *Clave de un individuo relacionada*
 CveDel = 8{caracter}8 * clave de un delito relacionada*
 CveDil = 5{caracter}5 * clave de la diligencia relacionada*
DatoDec = FechaDec + HoraDec
 Datos Generales= IdenDato+ nombre + |cargo |+ |dependencia| + |alias |+ nacion +sex
 +fechaNac + |ocup|+edocivil + direccion+Escol
 DatosParte = CveDil + #Acuerdo
 Declaraciones = FechaDec + HoraDec + Texto
 Deleg= 2{número}2 * corresponde al número de clave de la delegación respectiva*
 Delegación = 1{caracter}30 *corresponde al nombre de la delegación en la dirección*
 Delito = Nomdel + CveDel + LeyDeL+ EstCons
 Denunciante= datos generales
 Dependencia= 1{caracter}35 * representa la dependencia donde trabaja dicha persona*
 Desp_Aseg = Clasificación+ Cantidad + Medida
 Dest = 1{caracter}22 * Destino de la Diligencia*
 Determ= Avprev + FechaDet + HoraDet + Tipo + Fin_Ap + Texto
 Día = | 1- 31| * representa el número correspondiente al día del mes*
 Diligencia = Avprev + #Acuerdo + CveDil+ FechaDil + HoraDil +Tipo +Dest +
 Pendiente + Texto * información con respecto a diligencias*
 Direccion = calle + colonia + |delegación| + edo + país + cp
 Edo= 1{caracter}35 * correspondiente al nombre del estado en la dirección*
 Edocivil = [C|S] * determina si la persona es casada o soltera*
 Escol= 1{caracter}35 * corresponde al grado de escolaridad que tiene la persona*

Escritos= texto * corresponde a una declaración impresa*

EstCons = caracter * Campo lógico que determina si un indiciado es o no consignado y por cual delito*

Fecha = Día + - + Mes + - +Año

Fecha_Ac = fecha * fecha de inicio del acuerdo*

Fecha_Ap = fecha * Fecha de inicio de la Averiguación Previa*

Fecha_Det = fecha * fecha de la determinación*

Fecha_par= fecha * fecha del parte*

Fechedec = fecha * corresponde a la fecha de la declaración*

FechaDil= Fecha * Fecha de la Diligencia*

FechaNac= fecha * correspondiente a la fecha de Nacimiento*

Fin_Ap = caracter * identificador que determina si ha finalizada la A.P.*

Hor= [0-24|0-12] corresponde al número de hora en el día

Hora= Hor + : + Min +: + Seg

Hora_Ac = Hora * hora de inicio del acuerdo*

Hora_Ap = Hora * Hora de inicio de la Averiguacion Previa*

Hora_Det = hora * Hora de la determinación*

HoraDec= Hora * corresponde a la hora de la declaración*

HoraDil = Hora * hora de la Diligencia*

IdenDato = Avprev + Cve_Nom

Indiciado= datos generales + EstCons

Información que corresponde a la contestación de Diligencias * corresponde a información necesaria con respecto de las partes*

Involucrados Primarios= 1{{testigo}}1{{denunciante}}1{{indiciado}}

LeyDel = 5{caracter}5 *clave correspondiente a un articulo de ley relacionada a un delito*

Medida = 1{caracter}10 * unidad de medida utilizada*

Memo= 1{caracter} * corresponde a un número indeterminado de caracteres; ejemplo una carta*

Mes= |1-12| * representa el número correspondiente al mes del año*

Mesa=2{número}2 * corresponde al número de máquina asignada dentro de la agencia*

Mín=|0-59| corresponde al número de minutos

MP= 1{caracter}40 * Nombre y/o Clave del Agente del Ministerio Público*

Nación = 1{caracter}35 * representa el nombre de la nacionalidad de dicha persona*

Nombre de Pila = 1{caracteres}25 * se limita el nombre a 25 caracteres*

Nombre= 1{caracteres}25

Nombre= nombre de pila + apellido paterno + apellido materno

NomDel = 1{caracter}250 *nombre de un delito relacionado*

Número= |0-9|

Numparte= 1{caracter}10 * Número de Parte*

Ocup= 1{caracter}35 * corresponde a la ocupación de la persona*

País=1{caracter}25 * corresponde al nombre de la dirección*

Parte= Avprev + #Acuerdo + CveDil + Numpart + Fecha_par

Pendiente = caracter * identificador que determina si la Diligencia esta pendiente o no*

Seg=|0-59| corresponde al número de segundos

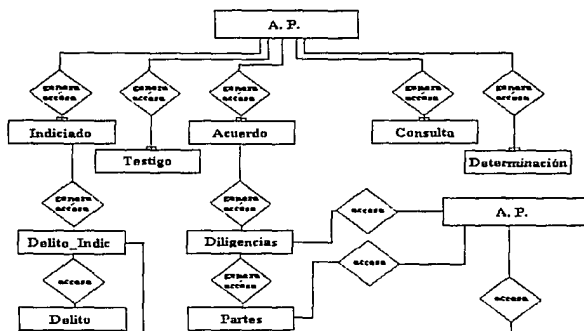
Sex= [M|F] * especifica si el sexo de la persona es masculino o femenino*

Testigo= datos generales

Texto= memo* corresponde a una declaración, diligencia, determinación o acuerdo*

Tipo = 1{caracter}20 * identifica el tipo de aseguramiento, diligencia o Determinación*

Diagrama Entidad-Relación



Características Operativas

Especificación de Software

En la programación del sistema se utilizará Clipper Ver. 5.2., lenguaje de programación correspondiente a la Tercera Generación, enfocado en el manejo de Bases de Datos. Debido a la gran cantidad de ventajas que presentan, se pretendía usar un lenguaje 4GL como UNIFACE, INFORMIX, ORACLE, o SYBASE. Sin embargo, la etapa de Implementación del sistema rebasó en tiempo a la adquisición de dichos lenguajes, razón por la cual se seleccionó Clipper.

El sistema requiere un computador personal AT o superior, 497 KB de memoria RAM convencional para correr adecuadamente, monitor VGA y 2 MB de disco duro como mínimo para ser instalado, sistema Operativo DOS Ver. 3.3 o posterior.

Especificación de Hardware

El equipo prototipo en el que se utilizará el sistema es el siguiente:

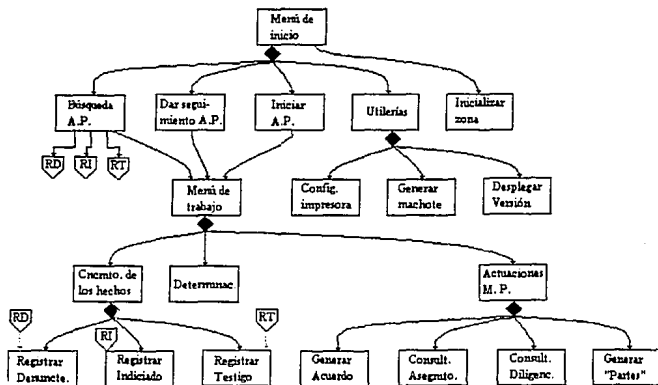
<i>CONCEPTO</i>	<i>CARACTERISTICAS</i>
Procesador	80486 SX
B U S:	16 Bits ISA
Velocidad:	25 MHz
Memoria R A M:	8 MB EXP. A 16 MB
Disco Duro:	120 MB. 17 ms
Disco Flexible:	3.5" 1.4MB
Monitor :	SUPER VGA COLOR 14" PANTALLA ANTIRREFLEJANTE
Slots:	3 ISA
Puertos:	2 SERIALES Y 1 PARALELO
Controlador de Disco Duro	IDE
Mouse:	400 DPI C/Tapete
Teclado:	Español 101 teclas
Sistema Operativo:	MS-DOS 6.0 Windows 3.1 Español

Capítulo IV

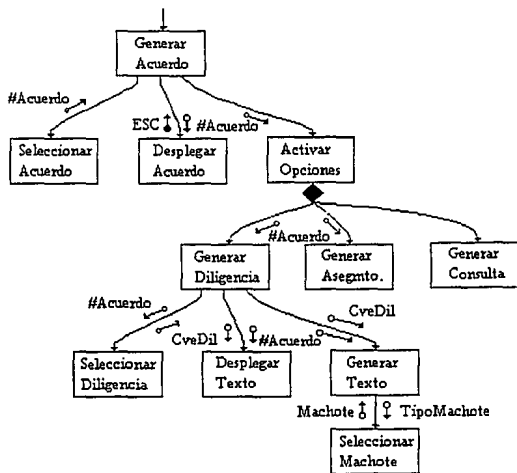
Diseño del Sistema

Cartas de Estructura

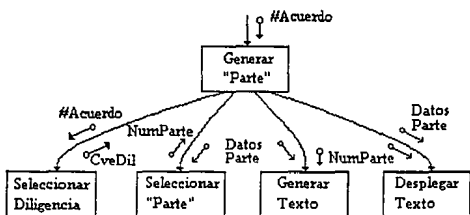
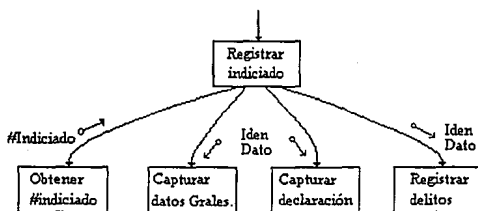
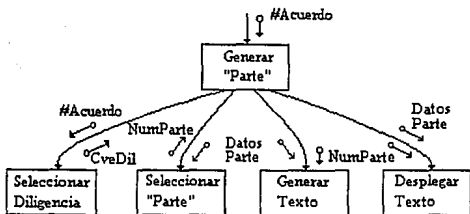
A continuación se presentarán las cartas de estructura del sistema, en cuanto a las miniespecificaciones de los mismos, será suficiente con la especificación de procesos que se obtuvo en la fase de análisis. Los módulos a los que no se halla hecho referencia anteriormente serán completados con la miniespecificación correspondiente. Debido a la magnitud del sistema, a que las mismas personas que realizaron el análisis, realizaron también el diseño e incluso la implementación y a que el lenguaje de programación Clipper proporciona gran cantidad de rutinas y funciones predefinidas no se consideró necesario incluir módulos de entrada y salida de datos.

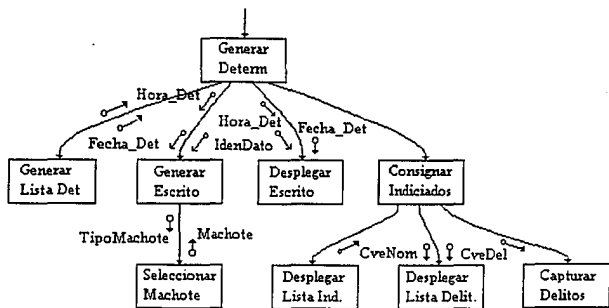
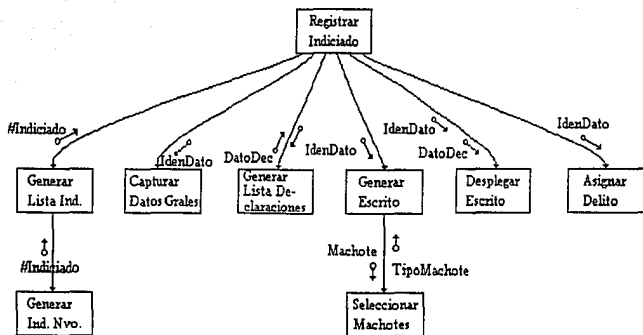


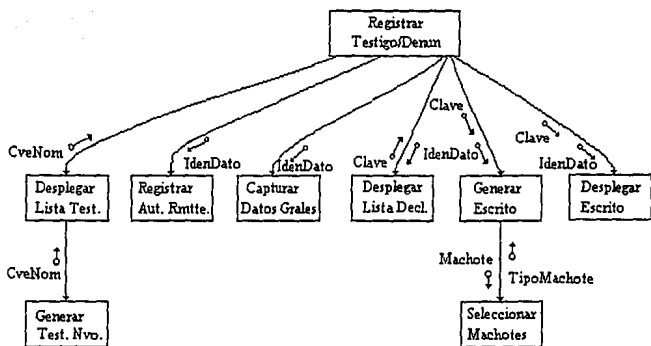
En los diagramas anteriores se describe la estructura de menús que se utilizará en el sistema; el primero presenta distintas opciones que serán utilizadas para acceder a una averiguación previa dada y adicionalmente otorga la opción de utilizar algunas utilerías, como la generación de machotes y la configuración de la impresora. La estructura encabezada por el módulo "Menú de trabajo" organiza las diversas acciones que se podrán realizar dentro de una averiguación previa en particular.



Generar Acuerdo. Después de desplegar un acuerdo existente, es posible activar las opciones necesarias para manipular diligencias, aseguramientos y consultas.







Mini-especificación de los módulos

Módulo: Menú de Inicio

Propósito: Proporcionar un menú de selección para inicializar e ingresar al sistema

Detalles Funcionales:

1. Desplegar pantalla inicial.
2. Si se presiona <F1> en los primeros 3 segundos invoca la función Inicializar_Zona
3. Presentar menú con las siguientes opciones: Iniciar, Seguimiento, Búsqueda, Utilerías e invocar la función correspondiente (Iniciar_AP, Seguimiento, Búsqueda, Utilerías).

Módulo: Inicializar_Zona

Propósito: Establecer la Delegación, Agencia y Mesa donde estará funcionando la computadora que corre el sistema.

Detalles Funcionales:

1. Preguntar mediante un browse el nombre de la Delegación y la agencia, de acuerdo al catalogo correspondiente (DELEDOS.DBF y AGENCIAS.DBF respectivamente).
2. Capturar de pantalla el número de agencia (entero, 2 posiciones).
3. Guardar los datos en el archivo WW:MEM.

Módulo: Iniciar_AP

Propósito: Generar un número de A. P. único para dar inicio a una averiguación previa.

Retorna: #AP.

Detalles Funcionales:

1. Recuperar el número de Delegación Agencia y Mesa de WW.MEM.
2. Capturar de pantalla el número de A. P. proporcionado por el usuario.
3. Componer la llave primaria #AP

Módulo: Dar_Seguimiento

Propósito: Obtener #AP existente para continuar trabajando dentro de una A. P.

Retorna: #AP

Detalles Funcionales:

1. Recuperar el número de Delegación Agencia y Mesa de WW.MEM.

2. Pedir al usuario la parte faltante del #AP
3. Componer la llave primaria #AP.

Módulo: Búsqueda_AP

Propósito: Localizar una A. P. dado el dato de algún participante

Retorna: #AP.

Detalles Funcionales:

1. Presentar Menú: "Por Denunciante, Por Indiciado, Por Testigo, Completa".
2. El usuario deberá proporcionar datos parciales del participante y la búsqueda se hará en la tabla correspondiente.
3. Los datos que concuerden con el dato parcial serán desplegados en un browse para que el usuario elija el participante adecuado o la A. P. correspondiente.
4. Si se elige un participante, llamar el módulo de Registro de participante correspondiente, por ejemplo, para indiciado se invocará Registrar_Indiciado
5. Si se elige un número #AP será el valor de retorno de la función.

Módulo: Utilerías

Propósito: Generar un menú cuyas opciones habiliten funciones adicionales del sistema (Configurar la impresora, Generar machotes y desplegar la versión del sistema)

Detalles Funcionales:

1. Presentar el menú descrito arriba e invocar la función correspondiente: "Configurar, Gen_Machotes, Versión".

Módulo: Menú_dc_Trabajo

Propósito: Generar un menú cuyas opciones habiliten las operaciones que se han de realizar sobre una averiguación previa.

Var. Entrada: #AP

Detalles Funcionales:

1. Desplegar la pantalla con al menú correspondiente, haciendo llamadas a los módulos especificados en la carta de estructura.
2. Activar la opción correspondiente.

Módulo: Configuración_de_Impresión

Función: Establecer márgenes y la clase de impresora que se utilizará en la impresión.

Detalles Funcionales:

1. Las opciones de impresora serán impresora láser H. P. y Epson de matriz de puntos.
2. Los márgenes configurables son el superior (en renglones) y el izquierdo (en caracteres). El texto generado en el sistema tiene una longitud fija (60 caracteres).
3. Debe existir opción para numerar las hojas.

Módulo: Generación_de_Machotes

Propósito: Generar machotes que puedan ser utilizados como plantillas para generar documentos dentro de la A. P.

Detalles Funcionales:

1. Presentar un menú para que el usuario elija el tipo de machote que desea: Declaración, Acuerdo, Diligencia, Parte, Determinación.
2. El usuario proporcionará el nombre del machote, será el medio para identificarlo, además del tipo de machote.

3. Capturar el escrito y almacenarlo en MACHOTES.DBF junto con el tipo (letra en negrita de las opciones del menú) y el nombre proporcionado por el usuario.

Módulo: Versión

Propósito: Desplegar la versión del sistema.

Detalles funcionales:

1. Simplemente despliega la versión del sistema, la cual estará almacenada como constante dentro del mismo módulo.

Módulo: Machotes

Propósito: Copiar una machote dentro del documento que el usuario está generando.

Var. Entrada: Tipo_Machote.

Retorna: Un string o cadena.

Detalles Funcionales:

1. Presentar un browse que contiene los nombres de los machotes disponibles, del tipo especificado y las primeras líneas del machote que se encuentre iluminado dentro del browse.
2. Se retornará un string que contenga el machote que fue seleccionado.

Diseño de la base de datos

Además del diseño de la arquitectura del sistema, es necesario generar el diseño de la base de datos. A continuación será presentado el mismo incluyendo las pruebas de normalización de las tablas las cuales están basadas en el diagrama entidad-relación existente. Las claves utilizadas dentro en la columna *Clase* son las siguientes: **PK**: LLave primaria, **NN**: No Nulo, **AS**: Asignados por el sistema, **FK**: LLave foránea, **NU**: No única. En el caso de que la columna *Descripción* incluya las claves válidas, estas claves estarán incluidas en el código del programa.

AP				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK	Num	2	Del. estatal de la P. G. R.
Agencia	PK	Num	2	Numero de Agencia en la Deleg. corresp.
Mesa	PK	Num	2	Mesa de trabajo dentro de la Agencia
NumAP	PK	Character	14	Numero de A.P. que proporciona el M.P.
Paterno		Character	22	Apellido Paterno del M. P. responsable.
Materno		Character	22	Apellido Materno del M. P. responsable.
Nombre		Character	22	Nombre del M. P. responsable.
Fecha	NN, AS	Date	8	Fecha de inicio de la Av. Prev.
Hora	NN, AS	Character	8	Hora de inicio de la Av. Prev.
FinAP		Logical	1	Indica si la AP ha concluido

Acuerdo				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK, FK	Num	2	*
Agencia	PK, FK	Num	2	*
Mesa	PK, FK	Num	2	*
Num/P	PK, FK	Character	14	*
Fecha Ac	PK, AS	Date	8	Fecha en que se realizó el Acuerdo
Hora Ac	PK, AS	Character	8	Hora en que se realizó el Acuerdo
Texto		Memo	Variable	Texto del Acuerdo

Dilig				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK, FK	Num	2	*
Agencia	PK, FK	Num	2	*
Mesa	PK, FK	Num	2	*
NumAP	PK, FK	Character	14	*
FHAC	PK	Character	16	Fecha Ac+Hora Ac
Cve Dil	PK, AS	Character	5	Cve. Dilig., se reinicia en cada acuerdo
Fecha	PK, AS	Date	8	Fecha de realización de la diligencia
Hora	PK, AS	Character	8	Hora de realización de la diligencia
Tipo		Character	35	Tipo de diligencia
Dest		Character	35	Nombre del destinatario
Texto		Memo	Variable	Texto de la diligencia
Parte		Logical	1	Indica si la diligencia fue contestada, C:totalmente P:parcialmente, N:no Cont.

Parte				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK, FK	Num	2	*
Agencia	PK, FK	Num	2	*
Mesa	PK, FK	Num	2	*
NumAP	PK, FK	Character	14	*
FHAC	PK, FK	Character	16	*
Cve Dil	PK, FK	Character	5	*
Fecha	PK	Date	8	Fecha de realización del <i>parte</i>
Hora	PK	Character	8	Hora de realización del <i>parte</i>
Texto		Memo	10	Texto del <i>parte</i>

Aseg				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK FK NU	Num	2	*
Agencia	PK FK NU	Num	2	*
Mesa	PK FK NU	Num	2	*
NumAP	PK FK NU	Character	14	*
FHAC	PK FK NU	Character	16	*
Clasif		Character	20	Tipo de aseguramiento
Cantidad		Num	9.3	Cantidad de lo asegurado
Unidad		Character	10	Unidad de medida de lo asegurado

Indiciado				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK, FK	Num	2	*
Agencia	PK, FK	Num	2	*
Mesa	PK, FK	Num	2	*
NumAP	PK, FK	Character	14	*
Clave	PK	Character	4	Num. indiciado dentro de la A. P.
Patern		Character	22	Apellido Paterno del indiciado
Matern		Character	22	Apellido materno del indiciado
Nombre		Character	22	Nombre de pila del indiciado
Nación	FK	Numeric	3	Nacionalidad del indiciado (catalogo)
Sex		Character	1	Sexo del indiciado
FechaNac		Date	8	Fecha de nacimiento del indiciado
Escol		Character	1	Escolaridad del indiciado
Especial		Character	1	F:funcionario I:Indígena N:nada
Cargo	FK	Numeric	2	Si es funcionario, qué cargo tiene
Depend	FK	Numeric	2	Dependencia a la que pertenece
Ocup		Character	20	Ocupación del individuo
ClvOcup	FK	Character	1	Clave de la ocupación [A] Artesano [B] Estudiante [C] Ayudante [D] Campesino [E] Chofer [F] Comerciante [G] Empleado [H] Estudiante [I] Hogar [J] Obrero [K] Oficio [L] Profesionista [M] Técnico [Z] Otro
EdoCivil		Character	1	Estado civil S:soltero C:casado V:viudo D:divorciado, U:Unión Lib. N:Concubinato
Calle		Character	35	Calle donde vive el indiciado
Colonia		Character	22	Colonia donde vive el indiciado
Deleg		Character	22	Delegación donde vive el indiciado
CP		Character	5	Código Postal
Alias		Character	22	Alias del indiciado
Tel		Character	20	Teléfono(s) del indiciado
Detenido		Logical	1	¿El indiciado está detenido?
Consignado		Logical	1	¿El indiciado está consignado?

Consulta				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK FK	Num	2	*
Agencia	PK FK	Num	2	*
Mesa	PK FK	Num	2	*
NumAP	PK FK	Character	14	*
Fecha Env	PK, AS	Date	8	Fecha de Envío a consulta
Fecha Rec		Date	8	Fecha de Recepción
TIPO		Num	1	Tipo de Consulta: Acumulado[1], NEAP Archivo[2], Reserva[3], Incompetencia[4]; Cons. S/Detenido [5], Cons. C/Detenido[6]
Num Of		Character	10	Numero de oficio de la respuesta.
Aut		Character	1	¿La consulta fue autorizada?

Declara				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK FK	Num	2	*
Agencia	PK FK	Num	2	*
Mesa	PK FK	Num	2	*
NumAP	PK FK	Character	14	*
Clave	PK, AS	Character	4	Clave del declarante
Fecha	AS	Date	8	Fecha en que se tomó la declaración
Hora	AS	Character	8	Hora en que se tomó la declaración
Texto		Variable	10	Texto de la declaración

Delind				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK FK	Num	2	*
Agencia	PK FK	Num	2	*
Mesa	PK FK	Num	2	*
NumAP	PK FK	Character	14	*
ClavInd	PK, AS	Character	10	Clave del indiciado. reinicia en cada AP
ClavDel	FK	Character	10	Clave del delito
Inicial		Logical	1	¿Delito por el que es perseguido inicialmente?
Consignado		Logical	1	¿El detenido fue consignado?

Dentest				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK FK	Num	2	*
Agencia	PK FK	Num	2	*
Mesa	PK FK	Num	2	*
NumAP	PK FK	Character	14	*
Clave	PK AS	Character	4	Clave del denunciante o del testigo, se reinicia en cada A P
Patern		Character	22	Apellido paterno del Denun. o testigo
Matern		Character	22	Apellido materno del Denun. o testigo
Nombre		Character	22	Nombre del denunciante o testigo
Nación		Character	20	Nacionalidad del denunciante o testigo
Sex		Character	1	Sexo del denunciante o testigo [M. F]
FechaNac		Date	8	Fecha de nacimiento del Den. o testigo
Ocup		Character	25	Ocupación del Denun. o testigo
EdoCivil		Character	1	Estado civil del Denun o testigo
Direcc		Character	45	Dirección del Denun o testigo
CP		Character	5	Código Postal del Denun o testigo
Tel		Character	20	Teléfono del Denun o testigo
Cargo		Character	45	Cargo de la autoridad remitente
Dependencia	FK	Num	2	Dependencia donde labora la Aut. Remitente.
AutRem		Logical	1	¿Se trata de una Autoridad Remitente?

Determ				
Campo	Clase	Tipo	Longitud	Descripción
Delegación	PK FK	Num	2	*
Agencia	PK FK	Num	2	*
Mesa	PK FK	Num	2	*
NumAP	PK FK	Character	14	*
FechaDet	PK, AS	Date	8	Fecha en que se generó la determinación
HoraDet	PK, AS	Character	8	Hora en que se generó la determinación
Tipo		Numeric	1	Tipo de determinación (Igual a tipo de consulta)
Texto		Character	Variable	Texto de la determinación
Fin AP		Logical	1	Campo descontinuado
Respaldo		Logical	1	Campo descontinuado

Machotes				
Campo	Clase	Tipo	Longitud	Descripción
Nombre	PK, NU	Character	16	Nombre del machote
Texto		Memo	Variable	Texto del machote
Tipo	NN	Char	1	Tipo de machote; [D: Diligencia, A: Acuerdo, E: Declaración, P: Parte, T: Determinación]

Una vez definida la estructura de la base de datos se procederá a analizarla para conocer su solidez, descubrir posibles fallas y estar en posición de sugerir mejoras. Las siguientes tablas, muestran la base de datos con información, para facilitar así su observación.

AP

Del	Agencia	Mesa	NumAP	Paterno	Materno	Nombre
02	01	01	0001-II	PEREZ	ORTIZ	JOSE
02	01	01	0002-II	PEREZ	ORTIZ	JOSE
02	01	01	0003-I	SANCHEZ	PIN	PEDRO

Fecha	Hora	Fin AP
19/08/94	16:38:19	.F.
20/08/94	20:00:02	.F.
24/08/94	15:10:12	.F.

Acuerdo

Del	Agencia	Mesa	NumAP	Fecha Ac	Hora Ac	Texto
02	01	01	0001-II	20/08/94	14:00:12	(texto)
02	01	01	0001-II	20/08/94	14:30:00	(texto)
02	01	01	0003-I	24/08/94	17:18:22	(texto)

① ① ① ①

Dilig

Del	Agencia	Mesa	NumAP	FHAC	Cve Dil	Fecha
02	01	01	0001-II	20/08/9414:00:12	D0001	20/08/94
02	01	01	0001-II	20/08/9414:30:00	D0002	20/08/94
02	01	01	0003-I	24/08/9417:18:22	D0001	24/11/93

②

Hora	Tipo	Dest	Texto	Parte
14:31:12			(texto)	C
09:32:12			(texto)	P
10:12:12			(texto)	N

②

②

Parte

Del	Agencia	Mesa	NumAP	FHAC	Cve Dil	Fecha
02	01	01	0001-II	20/08/9414:00:12	D0001	20/08/94
02	01	01	0001-II	20/08/9414:30:00	D0001	22/08/94

Hora	Texto
10:30:00	(texto)
10:22:12	

Aseg

Del	Agencia	Mesa	NumAP	FHAC	Clasif
02	01	01	0001-II	20/08/9414:00:12	COCAINA
02	01	01	0001-II	20/08/9414:30:00	METRALLETA

②

Cantidad	Unidad
12.000	KILOGRAMO
3.000	PIEZA

②

Indiciado

Del	Agencia	Mesa	NumAP	Clave	Paterno	Materno
02	01	01	0001-II	00001	PEREZ	ORTIZ
02	01	01	0001-II	00002	PEREZ	ORTIZ
02	01	01	0003-I	00001	SANCHEZ	PIN

Nombre	Nacion	Sex	FechaNac	Escol	Especial	Cargo	Depend
PASCUAL	100	M	19/08/68	A	.F.		
JOSE	100	M	20/08/44	A	.F.		
PEDRO	100	M	18/08/61	C	.T.	28	20

Ocup	ClvOcup	EdoCivil	Calle	Colonia
OBRERO	J	C	FRESNO 12	FLORIDA
CAMPESINO	D	C	PINO 15	AMERICA
SECRETARIO	L	C	LONDRES 21	TORRES

⑤

⑤

Deleg	CP	Alias	Tel	Detenido	Consignado
TLAHUAC	07220	PERRO	5 55 55 55	.T.	.F.
XOCHIMILCO	08963		6 66 66 66	.F.	.F.
COYOACAN	04350	TURCO	2 22 22 22	.F.	.F.

Consulta

Del	Agencia	Mesa	NumAP	Fecha Env	Fecha Rec	Tipo
02	01	01	0001-II	02/09/94		4

Num Of	Aut
	.F.

Delind

Del	Agencia	Mesa	NumAP	ClavInd	ClavDel
02	01	01	0001-II	0001	10341100
02	01	01	0001-II	0001	10330001
02	01	01	0003-I	0001	12300003

Inicial	Consignado
.T.	.F.
.F.	.F.
.T.	.F.

Declara

Del	Agencia	Mesa	NumAP	Clave	Fecha
02	01	01	0001-II	D001	19/08/94
02	01	01	0001-II	0001	19/08/94
02	01	01	0003-I	D001	01/12/94

Hora	Texto
10:23:22	(texto)
11:00:09	(texto)
09:03:31	(texto)

Dentest ©

Del	Agencia	Mesa	NumAP	Clave	Patern	Matern	Sex
02	01	01	0001-II	D001	PEREZ	ORTIZ	M
02	01	01	0002-II	T001	LOPEZ	LOPEZ	M
02	01	01	0003-I	D001	SANCHEZ	PIN	M

Nombrc	Nación	FechaNac	Ocup	EdoCivil
JOSE	100	19/08/68	EMPRESARIO	C
PEDRO	100	20/08/44	AGRICULTOR	C
JUAN	100	18/08/61		C

Dirección	CP	TEL
HOMERO 16, COL. POLANCO, CUAHUTEMOC., D. F.	07220	5 55 55 55
OHIO 223, COL. ROSEDAL, COYOACAN, D. F.	08963	6 66 66 66
TURIN 76, COL. TORRES LIND. G. A. MADERO D. F.	04350	2 22 22 22

Cargo	Dependencia	AutRem
		.F.
		.F.
COMANDANTE	25	.T.

Determ

Del	Agencia	Mesa	NumAP	FechaDet	HoraDet	Tipo
02	01	01	0001-II	29/12/94	13:54:00	4

Texto	FinAP	Resplado
(texto)		

③

③

Machotes

Nombre	Texto	Tipo
ASEG.	(texto)	D
QUIMICO	(texto)	D
INICIO	(texto)	A

Observaciones

- ❶ Aparentemente podría utilizarse un llave primaria menos extensa, sin embargo, la idea original es que las bases de datos se concentren en una computadora central y por tanto se optó por recurrir a dicha llave.
- ❷ En este caso existe una evidente violación a la primera forma normal, no obstante, simplifica enormemente el manejo interno de las bases de datos sin un perjuicio que sea de consideración ya que nunca se explota de manera separada dicho campo.
- ❸ Estos campos dejaron de ser útiles debido a cambios en la evolución del sistema y se conservan por compatibilidad con las primeras versiones.
- ❹ Se considera que es más conveniente utilizar un catálogo separado con una clave para describir los atributos en cuestión.

- ⑤ Se detecta una violación a la tercera forma normal, que sin embargo no fue corregida debido a que el campo "Ocup" no tiene un patrón definido.
- ⑥ Originalmente la entidad "Dentest" era idéntica tanto para denunciantes como para testigos, pero con la evolución del sistema surgieron pequeñas diferencias entre unos y otros, de tal forma que sería mejor crear entidades separadas.

Capítulo V

Implementación del Sistema

Programación

Como ya se mencionó anteriormente, el lenguaje utilizado para la programación del sistema, el cual favorece la Programación de tipo Estructurada, técnica de programación mediante la cual se crean diversas cajas negras anidadas; entendiéndose por cajas negras bloques de programación los cuales tienen un propósito específico y que generalmente tienen una entrada y una salida que pueden ser usadas por otras cajas negras. El uso del modelo relacional es factible ya que se pueden implementar llaves del sistema en diferentes variables (operación transparente en los lenguajes 4GL) logrando así la intercomunicación relación entre las diferentes tablas.

Empaquetación

Los módulos del sistema, descritos en el Diagrama Estructural, se empaquetaron en seis Unidades de Carga estas son : UNIPRIN.PRG, UNIAC.PRG, UNIDIL.PRG, UNIDET.PRG, BUSQUEDA.PRG, MACHOTES.PRG.

En UNIPRIN.PRG se encuentra los módulos de más alta jerarquía, el módulo correspondiente a Conocimientos de los hechos, así como funciones de propósito general que son llamadas con una alta frecuencia. El hecho que sea este el paquete principal permite que se le de preferencia en cuanto al uso de la memoria, por lo tanto, esas rutinas de alta frecuencia estarán prácticamente siempre en memoria y de esta manera evitar que se tenga que subir a la misma después de cada llamada. En UNIAC.PRG se encuentra los módulos correspondientes a Acuerdos, Consulta y Aseguramientos. En UNIDIL.PRG se encuentra el módulo correspondiente a Diligencias y Partes. En UNIDET.PRG se encuentra el Módulo correspondiente a la Determinación. En BUSQUEDA.PRG se localiza el módulo correspondiente a Búsqueda y finalmente el MACHOTES.PRG el módulo correspondiente a Machotes.

Es importante aclarar que cuando existe memoria suficiente (lo que generalmente ocurre en todos los casos) realmente todos los archivos se convierten en una sola Unidad de Carga. La división en estos programas se debe a la previsión para aquellos remotos casos en donde la memoria convencional no sea lo suficiente para poder ejecutar todos ellos. En estos casos existe la posibilidad para utilizar "OVERLAYS" en donde nuestras las diferentes Unidades de Carga compartirán la memoria de acuerdo al orden de preferencia que se fija. El sistema, sin embargo, tiene un tamaño adecuado (365 Kb) y es por esta razón que nunca fue necesario la utilización de "OVERLAYS".

La Codificación empleada fue del tipo "Top Down", usándose además, técnicas propias de la programación estructurada en donde cada procedimiento es concebido como una caja negra haciendo énfasis en sus entradas y sus salidas. En el sistema existen algunos procedimientos que funcionan independientemente, esto es, no es necesario una entrada específica para que estos puedan ejercer su tarea, simplemente son llamadas y estas ejecutan su procedimiento sin regresar ningún valor.

Se prefirió codificar primero UNIPRIN.PRG, ya que aquí se encuentran los módulos de más alta jerarquía, como el menú principal, de donde se llaman todos los demás módulos. al igual que la generación de número de averiguación previa. El cual es indispensable para prácticamente todos los demás módulos. Dentro del mismo UNIPRIN.PRG se encuentra el módulo correspondiente a Conocimiento de los Hechos, este tiene la característica de tener submódulos Aferente y Eferentes, por esta razón se le dio prioridad en sus codificación, incluyendo a la codificación de sus submódulos (implementación de tipo paraguas). Paralelamente codificado fueron UNIAC.PRG y UNIDET.PRG ya que módulos que se encuentran en el mismo nivel, independientes el uno del otro. UNIDIL.PRG es subordinado de alguna manera a UNIAC.PRG, causa por la que fue codificado después. Al final en la codificación corresponde paralelamente a MACHOTES.PRG y a BUSQUEDA.PRG ya que son módulos suplementario y de bajo nivel.

En cuanto los aspectos de la programación que se les dieron una mayor importancia fueron:

- **CORRECTIVIDAD** Es absolutamente prioritario realizar satisfactoriamente todos los requerimientos planteados en el Análisis para poder esperar una aceptación del sistema. La Adecuabilidad es siempre importante, pero en un sistema como este es vital, ya que a la mínima discrepancia con respecto al Análisis la aceptación del sistema tiende a disminuir prácticamente a cero.

- **PRODUCTIVIDAD** La división de módulos entre programadores diferentes, tomando en cuenta especificaciones de organización fijadas en el diseño, permitió aumentar la capacidad de producción de diferentes procedimientos y módulos en un tiempo previamente fijado.
- **MANTENIMIENTO** Las técnicas de programación estructurada empleadas en la codificación siempre fueron pensadas para tener una alta Cohesión y disminuir el Acoplamiento, de esta manera los códigos de programación son, en muchos casos independientes entre sí y en general fácil de entender; estos factores facilitan su mantenimiento.

Pruebas

La técnica de "Top Down" utilizada en la codificación hace prácticamente obligatorio que las pruebas del sistema se realicen con la misma técnica. Cuando un módulo se termina de codificar, en ese punto se realizan paralelamente dos tipos de prueba: Pruebas de Funcionamiento y Pruebas de Recuperación. El objetivo de esas pruebas es tener una certeza tanto de la Correctividad de los módulos como de su Robustez. En el caso de un resultado desfavorable, este se analizaba para clasificarlo en: error de codificación, de diseño o bien de análisis (este proceso de clasificación muchas veces es inmediato) de esta manera podemos ejercer una retroalimentación en cualquiera de esos puntos.

En las Pruebas de Funcionamiento se le alimentaba al sistema con las entradas fijadas en el Análisis y este debía de producir el resultado esperado en su salida. En todos los casos las salidas deberán de ser las anticipadas, de no serlo así habría un error. En las pruebas de Recuperación se producían diferentes tipos de anomalías como entradas que no fueran las planteadas en el análisis, crear altas sin los nombres requeridos, apagar la máquina en plena sesión etcétera. En este tipo de pruebas se esperaba que este pudiera reconocer las entradas erróneas y que no se crearan registros en blanco, todo con el fin de no alimentar información errónea o incompleta en las bases.

Una de las más importantes retroalimentaciones al sistema se debió a una prueba de Recuperamiento: el apagar la máquina en plena sesión causaba una pérdida de la

información. Por las características de los documentos que se manejan en las averiguaciones previas, estos no pueden ser modificados; razón por la cual una vez que el usuario guarde definitivamente² la información con respecto a un documento, éste es guardado y no podrá ser modificado en el futuro. Esto constituía un gran riesgo, ya que un documento puede llegar a ser muy grande, al igual que el tiempo invertido en este (dos a tres horas) y como el usuario no puede guardar el documento sino hasta terminarlo (si le interesa salvarlo por supuesto), en caso de tener una falla de energía eléctrica, en el momento en que se produce un documento, este se perdería en su totalidad. Esto fue solucionando diseñando respaldos internos cada tres minutos y borrando estos cuando se registra una salida normal de un documento. En caso de un error estos respaldos pueden ser accedidos, continuando así con su elaboración, evitando la pérdida de información.

A pesar de que no se efectuaron formalmente Pruebas de Eficiencia se detectaron dos problemas de este tipo para tener consideración: Que hacer cuando el disco duro se llene y el espacio en disco que los campos de longitud variable ("MEMO") ocupan en las bases, ya que estos crecen de una manera innecesaria en cada modificación hecha a los mismos.

Si el disco duro se llenará en plena sesión se produciría un error en ese momento; un error que devolvería el control al Sistema Operativo y muy probablemente existiría una pérdida de información. Además ese error se presentaría en todo momento que se intentará entrar nuevamente al sistema (si no se libera espacio en disco). La solución de ese problema se obtuvo de esta manera: lo primero que calcula el sistema es la existencia de espacio en disco necesario, esto lo determina calculando que exista, por lo menos el doble de espacio del archivo más grande de la base de datos más 500Kb libres en el disco duro. En caso de que no exista despliega un mensaje de que no hay espacio en disco duro necesario y que por favor lo libere; en caso contrario entra al menú principal del sistema.

El espacio innecesario producido por los archivos de longitud variable se resolvió depurando la base antes de entrar a sesión. Esto consiste en copiar todos los archivos de la base a una temporal y de ahí nuevamente a la base borrando el temporal, con esto solamente se copia la información útil de los campos "MEMO" y se elimina la basura. Además de esta base temporal se tiene copia la base en un respaldo que se borra cuando

²Definitivamente porque cuando se guarda se tiene la opción de regresar a editar, de imprimir el documento y de guardar definitivamente, ese menú estará siempre activo al menos que se desee salir sin salvar o guardar definitivamente.

la transacción termina, de esta manera, si se presenta un error de energía, se recupera la base del respaldo cuando se entra de nuevo al sistema.

Instalación

Cada delegación de la Procuraduría General de la República fue dotada, entre otras cosas, con 5 computadoras personales. Es evidente que el equipo no es suficiente para cubrir las necesidades básicas de ninguna delegación, ya que estas son utilizadas para realizar labores administrativas y en la mayoría de los casos no es posible destinar ningún equipo para las mesas investigadoras. Por esta razón la implementación del sistema se ve enormemente limitada.

Para facilitar la instalación del S.A.A.P. fue necesario elaborar un programa instalador el cual modifica los archivos AUTOEXEC.BAT y CONFIG.SYS cuando estos lo requieren, permite crear un directorio de trabajo con el nombre que se desee o aceptar el directorio propuesto "C:\SAAP". Si el nombre del directorio ya existe el sistema lo menciona y pide que se verifique si este se desea sobre escribir. El programa instalador verifica que exista espacio suficiente en el disco duro para poder utilizar el sistema (busca por un espacio mayor a 2MBytes) si este espacio no existe despliega un mensaje de que no es suficiente y termina la instalación. En el directorio de trabajo se bajan todos los archivos necesarios para el que el sistema pueda funcionar.

Capacitación

Debido a implicaciones políticas y de infraestructura fue difícil llevar a cabo un plan adecuado de capacitación que de cualquier forma se realizará de la siguiente manera:

- Se elaborará un manual para el usuario y una presentación destinada al delegado estatal.
- Se organizarán tres equipos de dos personas cada uno, los cuales harán visitas a cada una de las delegaciones estatales de la República con la

intención de presentar el sistema a delegados y agentes del Ministerio Público.

- Se capacitará a los agentes del Ministerio Público así como a el personal secretarial que los auxilia en la elaboración de las averiguaciones previas. La capacitación se impartirá durante tres días y se cubrirán dos turnos diarios de tres horas cada uno, con la intención de adiestrar un mayor número de personas (se recomienda como máximo a dos alumnos por equipo).

Resultados de la Implementación

El sistema de Averiguaciones Previas tiene en general muy buena aceptación, muchas personas se entusiasmaron principalmente por las utilidades del mismo, como la generación y uso de machotes en documentos, las cuales se traducen en una mayor productividad en su trabajo. Es muy importante recalcar que si bien hubo algunas sugerencias para modificar el sistema, éstos no implicaban en absoluto la modificación de su análisis inicial.

Existen tres tipos principales de obstáculos que impiden una implementación más exitosa:

1. La falta de equipo se presenta como la mayor queja de parte de las delegaciones. Los equipos enviados no bastan para sustituir todas las máquinas de escribir usadas en las 283 mesas con que cuenta esta Institución. A pesar de que exista el deseo y el entusiasmo de implementarlo de una manera total se tiene que hacer parcialmente, en tanto se adquiere el equipo necesario.
2. La poca experiencia entre diferentes usuarios en cuanto el uso de computadoras hace difícil su cambio de manera de trabajar. Esto es más notorio en personas de edad avanzadas, sin embargo el interés mostrado por estas personas tiende a ser muy alto.
3. La falta de confianza manifestada por algunas personas hacia un sistema que catalogan exclusivamente de control que viene del centro. Afortunadamente este caso se da con menos frecuencia.

Este sistema ha probado su eficacia, su análisis puede ser usado de manera íntegra como modelo para las nuevas versiones del mismo. El sistema requiere una mejor infraestructura y mayor apoyo para poder implementar con éxito tanto la versión vigente (no red) como futuras versiones que tengan mayores alcances (que contemple una red nacional de información).

Capítulo VI

Conclusiones

El hecho de realizar un sistema formal, basados en la aplicación de una metodología deja una vasta experiencia en varios aspectos. Experiencia que va tanto del aspecto técnico, la aplicación de la metodología misma, como del aspecto político, el contacto y organización con las diferentes personas involucradas.

En el aspecto técnico el uso de una metodología se observó no sólo como necesario sino como fundamental; el hecho de crear un Modelo Esencial permite fijar las metas funcionales del sistema, la organización de los diferentes procesos para cumplir con esas metas funcionales y de funciones complementarias fijadas tanto en el análisis como en el diseño. Por otro lado, las técnicas de Implementación estudiadas permiten una mejor estructuración y una división tangible de las tareas producidas por el sistema dejando a su vez un testimonio confiable del desarrollo del sistema para su futuro mantenimiento.

Todas las etapas de la metodología son recomendables para un mejor integración de las mismas, estas a la postre se traducirán en mejores y más sólidos resultados. En el desarrollo del sistema S.A.A.P., sin embargo, esa metodología se aplicó de una manera flexible ya que no se llevaron a cabo todos los pasos descritos en la metodología en la cual se basó el sistema. La razón de esto es muy sencilla: la metodología presupone la existencia de grupos bien definidos de Analistas, Diseñadores y Programadores enfocados en la elaboración de sistemas. Se presenta además una constante recomendación de que el grupo de Análisis y el de Diseño solamente interactúen entre sí cuando sea imperativo y por lo tanto, no se mezclen tareas del Análisis en el Diseño y viceversa. Cuando un sólo grupo de gente realiza esas dos funciones (como fue el caso en el sistema S.A.A.P.) existe la tendencia de no ser tan formal en ese aspecto, esto se debe a que existen tareas muy semejantes tanto en el Análisis como en el Diseño. El hecho de que un analista sea también diseñador hace muy difícil que se pueda separar efectivamente estas dos actividades, éste tenderá a ser influenciado por el Diseño desde el momento mismo del Análisis. Un ejemplo de esto es la especificación de procesos vista en el análisis, si el analista es también el diseñador tenderá a realizar su Especificación de Procesos muy parecido a la Especificación de los Módulos descritas en el diseño. Es natural, si este fuera el caso, de querer eliminar una de estas dos especificaciones ya que parecería una repetición del mismo trabajo. Existen Módulos, sin embargo que son ajenos a cualquiera de los procesos descritos; esos módulos si

deberán de ser especificados. Este tipo de flexibilidad es válido porque no interfiere el objetivo principal de la metodología que es el dirigir un sistema de una manera ordenada y coherente a través de sus diferentes etapas de Análisis, Diseño e Implementación.

Las técnicas sugeridas, no sólo se aplicaron exitosamente, sino que produce un verdadero cambio en la forma de trabajar, desde pequeños proyectos hasta los grandes sistemas se pueden efectuar con una mayor robustez y claridad. Muchas técnicas se pueden emplear tanto en macroproyectos como en microproyectos, e inclusive algunas se pudieran exportar a otros tipos de sistemas (la técnica "Top Down" corresponde a uno de estos casos). El conocimiento y la adopción de esta metodología, incluyendo todas las técnicas sugeridas dejan una profunda huella que se manifestará en trabajos posteriores.

En el aspecto humano el sistema produjo, quizá, todavía mayores experiencias. El contacto con todo tipo de personas que mostraron diferentes actitudes con respecto al sistema fue la causa de esto. En la Procuraduría General de la República hay una especial presión en la obtención de resultados inmediatos, un sistema como el S.A.A.P. sin embargo, no ofrece resultados con tanta rapidez. La existencia de otros sistemas de Averiguaciones Previas que al parecer carecen de una buena planeación y dan la sensación de improvisación, muy probablemente son el resultado de esa presión. Afortunadamente el S.A.A.P. tuvo el tiempo suficiente para poder desarrollar su Ciclo de Vida, tuvo especialmente un muy sólido Análisis, esto porque hubo la verdadera fortuna de poder ser asesorados por verdaderos expertos en el campo del Derecho, este solo hecho basta para que resalte de los otros. Lo sólido de su Análisis se constató en las diferentes visitas y presentaciones en diversos lugares de la República, nunca hubo ningún problema en cuanto a la manera de modelar lo que representa la Averiguación Previa. Sin embargo, por razones de infraestructura, por el tiempo de su desarrollo, por los diferentes cambios presentados en la Procuraduría o por la presión previamente citada, la prioridad que se le dio a este sistema no fue la más alta y el equipo encargado de su desarrollo fue muy pequeño, no obstante se crearon la bases para una implementación grande y fructuosa.

A lo largo de la evolución del presente proyecto se ha podido vislumbrar el gran potencial que un desarrollo como el Sistema Automatizado de Averiguaciones Previas representa, ya que es capaz por sí mismo de mejorar hasta en un 70% el tiempo de creación de una averiguación previa —En contraste con el uso de una máquina de escribir—. Este sistema es el primer paso hacia el cambio en la forma tradicional de trabajo del Ministerio Público Federal y puede convertirse en una herramienta tremendamente poderosa. sin embargo, es necesario considerar todas las actividades que

se realizan dentro de la Procuraduría General de la República para constituir un sistema integral que revolucione la dinámica de trabajo dentro de esta institución.

Bibliografía

CÓDIGO FEDERAL DE PROCEDIMIENTOS PENALES. Tercera Edición. Editorial Themis. Diciembre 1992.

Colín Sánchez, Guillermo. DERECHO MEXICANO DE PROCEDIMIENTOS PENALES. Editorial Porrúa. Decimocuarta Edición. México D.F 1980.

Date, C. J.. AN INTRODUCTION TO DATABASE SYSTEMS. Third edition, Addison Wesley Publishing Company. Massachusetts, EUA. 1981

Diario Oficial, REGLAMENTO DE LA LEY ORGÁNICA DE LA PROCURADURÍA GENERAL DE LA REPÚBLICA. Marzo 1993.
E.U.A. 1989.

Fleming, C Candace; Von Halle Barbara. HANDBOOK OF RELATIONAL DATABASE DESIGN. Addison Wesley Publishing Co. New York, E.U.A. 1979

Meilir, Page-Jones. The Practical Guide to STRUCTURED SYSTEMS DESIGN. Second edition. Yourdon Press Computing Series. Englewood Cliffs, New Jersey, EUA. 1988.

Prorise, Jeff. DOS 5, MEMORY MANAGMENT WITH UTILITIES. ZD Press, EUA 1992

Spence, Rick. CLIPPER 5.2 POWER PROGRAMMERS GUIDE. Sybex, Alameda Cal., EUA. 1993

Stonebraker, Michel. READINGS IN DATABASE SYSTEMS. Second edition. Morgan Kaufmann Publishers. Sn. Mateo California, EUA. 1994

Toby, Fry. DESIGN OF DATABASE STRUCTURES, Editorial Prentice Hall, E.U.A., 1982.

Whitten. Bentley, Barlow. SYSTEM ANALYSIS & DESIGN METHODS. Editorial Irwin, Segunda Edición. E.U.A., 1989.

Yourdon, Edward; Constantine, Larry. STRUCTURED DESIGN FUNDAMENTALS OF THE DISCIPLINE OF COMPUTER PROGRAM AND SYSTEM DESIGN. Prentice Hall. Englewood Cliffs New Jersey, E.U.A. 1979.

Yourdon Edward. MODERN STRUCTURED ANALYSIS. Editorial Prentice Hall. Englewood Cliffs New Jersey, E.U.A. 1982.