15882-01
TESIS-01

CINVESTAV-IPN
Biblioteca de Ingeniería Eléctrica
FB0000014043

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
INSTITUTO POLITÉCNICO NACIONAL

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN COMPUTACIÓN

TÉCNICA DE DIAGRAMACIÓN PARA EL DISEÑO ARQUITECTÓNICO DE SISTEMAS TRANSACCIONALES VÍA TECNOLOGÍA WEB

TESIS QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS
EN LA ESPECIALIDAD DE INGENIERÍA ELÉCTRICA

PRESENTA

ING. RAMÓN MONDRAGÓN SOTELO

DIRECTOR DE TESIS:

DR. SERGIO VÍCTOR CHAPA VERGARA

CINVESTAV
IPN
ADQUISICION
DE LIBROS

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

XM
CLASIF.
ADQU
FECHA
PROCED.
$

AGRADECIMIENTOS

Son muchas las personas e instituciones que me apoyaron para concluir mi Maestría en Ciencias.

Comenzaré por mencionar a mis padres: Genara Sotelo Arroyo y Roberto Mondragón Rivero y mis hermanos: Raúl, Martín y Marcelo, que nunca dudaron de mí y que me impulsaron para no detenerme ante ninguna adversidad, gracias por todo el apoyo que me han dado.

Un especial agradecimiento a mi futura esposa Lisette Anahi Flores García por todo el amor y fuerza que me ha dado, por compartir conmigo los momentos buenos y malos. Gracias Princesita.

Al Dr. Sergio Víctor Chapa Vergara por haber aceptado ser mi asesor de tesis y por el apoyo mostrado durante este proceso.

Al Dr. Arturo Díaz Pérez y al Dr. Guillermo Benito Morales Luna por ser mis revisores de tesis y por las aportaciones que hicieron a mi trabajo.

A mis compañeros de generación 1998-2000 por haber compartido conmigo la experiencia de realizar una maestría.

A las personas que laboran en la sección de computación, en especial a Sofía Reza Cruz por todo el apoyo mostrado durante mi instancia en el CINVESTAV-IPN.

A mis compañeros de trabajo por el apoyo mostrado durante esta etapa de mi vida.

Al CONACYT y al CINVESTAV-IPN por haberme dado la oportunidad de realizar mi maestría y por el apoyo económico recibido durante este periodo.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

RESUMEN

Las transacciones son los requerimientos o envíos que se le hacen al sistema (producto de *software*). Pueden ser tan sencillas como transferir un archivo o tan complejas como la manipulación de bases de datos.

Los sistemas transaccionales vía tecnología *web* son precisamente aquellas aplicaciones que utilizan transacciones para su funcionamiento y que además están desarrolladas sobre tecnología web (Internet, intranet, extranet). Esto último proporciona grandes beneficios a las empresas, por ejemplo la publicación y acceso a la información corporativa de forma rápida y eficiente.

Al igual que con otros proyectos de desarrollo de *software*, el ciclo de vida de los sistemas transaccionales vía tecnología *web* debe incluir la descripción detallada de los requisitos, análisis, diseño, construcción y mantenimiento.

Este trabajo de tesis se enfocó en la fase del diseño de los sistemas transacionales vía tecnología *web*.

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. Se define como el proceso de aplicar técnicas y principios con el propósito de describir un dispositivo, un proceso o un sistema con suficiente detalle como para permitir su realización física.

Las técnicas de diagramación de flujo de datos convencionales (la propuesta por Gane & Sarson, la de Dijkstra, Bohm y Jacopini y la de Chen) son utilizadas en el análisis y diseño arquitectónico de productos de *software*, pues ayudan a determinar el flujo de la información a través del sistema; sin embargo, presentan algunas carencias para representar los sistemas transaccionales vía tecnología *web*, tales como la interpretación de *tags*, eventos, operaciones SQL, componentes (Applets, Activex, scripts), etc.

En esta tesis se propone una técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*, con el propósito de subsanar las carencias de las técnicas de diagramación convencionales.

La técnica propuesta incorpora iconos que representan los principales elementos de la tecnología *web* utilizados en los sistemas transaccionales. Incluye también lineamientos para documentar los diagramas generados por la misma, así como un formato de enlace entre los diagramas y la ubicación física de los archivos que formarán parte del código del sistema.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

ÍNDICE

RESUMEN....iii

INTRODUCCIÓN....1

CAPÍTULO 1. TECNOLOGÍA *WEB*....4

1.1. Internet....4
1.2. Intranet....5
1.3. Extranet....6
1.4. Resumen....6

CAPÍTULO 2. SISTEMAS TRANSACCIONALES....8

2.1. Definición de transacción....8
2.2. Propiedades de las transacciones....8
2.3. Procesamiento de transacciones....8
2.4. Sistemas transaccionales vía tecnología *web*....9
2.5. Resumen....11

CAPÍTULO 3. TÉCNICAS DE DIAGRAMACIÓN CONVENCIONALES....12

3.1. Diagrama de flujo de datos de Gane & Sarson....12
3.2. Diagrama de flujo de datos de Dijkstra, Bohm y Jacopini....13
3.3. Diagrama entidad-relación propuesto por Chen....15
3.4. Aplicaciones más frecuentes de las tecnicas convencionales....16
3.5. Otras técnicas de diagramación....17
3.6. Resumen....19

CAPÍTULO 4. APLICACIÓN DE LAS TÉCNICAS DE DIAGRAMACIÓN CONVENCIONALES....20

4.1. Análisis del objeto de aprendizaje actividades....20
4.2. Diseño arquitectónico del *objeto de aprendizaje actividades* utilizando la técnica de Gane & Sarson....22
4.3. Diseño arquitectónico del *objeto de aprendizaje actividades* utilizando la técnica de Dijkstra, Bohm y Jacopini....24
4.4. Diseño arquitectónico del *objetos de aprendizaje actividades* utilizando la técnica de Chen....26
4.5. Comparación de las técnicas de diagramación convencionales aplicadas en un sistema transaccional vía tecnología *web*....28
4.6. Resumen....29

CAPÍTULO 5. TÉCNICA DE DIAGRAMACIÓN PARA EL DISEÑO ARQUITECTÓNICO DE SISTEMAS TRANSACCIONALES VÍA TENCOLOGÍA *WEB*.....30

5.1. Técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*.....30
5.2. Resumen.....36

CAPITULO 6. APLICACIÓN DE LA TÉCNICA PROPUESTA.....37

6.1. Diseño arquitectónico del objeto de aprendizaje actividades utilizando la técnica propuesta.....37
6.2. Construcción del sistema transaccional.....45
6.3. Mantenimiento del sistema transaccional.....48
6.4. Comparación de la técnicas de diagramación propuesta vs las convencionales.....49
6.5. Resumen.....49

CONCLUSIONES.....50

RELACIÓN DE FIGURAS.....52

GLOSARIO.....54

BIBLIOGRAFÍA Y REFERENCIAS URL.....55

La redes computacionales son un claro ejemplo del avance que ha tenido la tecnología informática. Al principio sólo eran redes locales, posteriormente metropolitanas y más tarde surgieron las redes mundiales tales como Internet, cuyo aprovechamiento en aplicaciones empresariales dio como resultados nuevos conceptos: intranets y extranets.

Las intranets son aplicaciones desarrolladas sobre tecnología Internet que corren a nivel de redes locales, es decir, no están abiertas a la red Internet.

Las extranets son aquellas aplicaciones que aprovechan las características que brinda una intranet y la red global Internet.

Son muchas las amenazas a la seguridad que se tienen en tecnología *web* (Internet, intranet y extranet): el acceso indebido a la información, daños al sistema, virus informáticos, etc. Se han adoptado diversos mecanismos de protección para evitarlas, por ejemplo el uso de *firewalls* (conjunto de componentes que funcionan como punto de obstrucción, restringiendo el acceso entre redes), contraseñas, *software* antivirus y encriptación. Pese a estos la seguridad sigue siendo un problema inherente a dicha tecnología [Jen98].

El desarrollo de aplicaciones en tecnología *web* es cada vez más utilizado por las empresas, debido a los múltiples beneficios que ofrece. Algunos de ellos son los siguientes:

- Reduce el tiempo de aprendizaje para utilizar el sistema por parte de los usuarios.
- Ofrece simplicidad en la instalación de aplicaciones.
- Brinda soporte a diferentes tipos de información: texto, imágenes, sonido y vídeo.
- Permite acceso a bases de datos.
- Permite la escalabilidad.
- Es de fácil navegación.
- La actualización de las aplicaciones se realiza sólo del lado del servidor.
- Soporta transacciones.

Las transacciones son los requerimientos o envíos que se le hacen al sistema. Pueden ser tan sencillas como transferir un archivo o tan complejas como la manipulación de bases de datos.

Los sistemas que utilizan las transacciones para su funcionamiento reciben el nombre de sistemas transaccionales, un claro ejemplo de este tipo de sistemas es una aplicación que manipula bases de datos.

Desarrollar un sistema transaccional vía tecnología *web* proporciona grandes ventajas a las empresas, además de las ya mencionadas, destacan la publicación y acceso a la información corporativa de forma rápida y eficaz.

En esta tesis se adopta el concepto de sistemas transaccionales vía tecnología *web* para hacer referencia a las aplicaciones o conjunto de aplicaciones que utilizan bases de datos y que se desarrolla sobre tecnología *web*.

La construcción de un sistema transaccional vía tecnología web es el resultado de su propio ciclo de vida: Definición de requisitos, Análisis, Diseño, Construcción y Mantenimiento.

En esta tesis se pondrá principal atención en la fase del diseño de los sistemas transaccionales vía tecnología *web*.

El diseño del software se sitúa en el núcleo técnico del proceso de ingeniería del mismo; es un proceso que traduce los requisitos en una representación gráfica (diagramas), es decir, el diseño se representa a un alto nivel de abstracción que se puede seguir hasta requisitos específicos de datos funcionales y de comportamiento. A medida que ocurren interacciones del diseño, el refinamiento subsiguiente lleva a representaciones de mucho menor nivel de abstracción. En el diseño se hace uso de las técnicas de diagramación de flujo de datos [Rog98].

Entre las técnicas de diagramación de flujo de datos más utilizadas están las siguientes:

1. Gane & Sarson.
2. Dijkstra, Bohm y Jacopini.
3. Chen.

La primera propuesta se emplea con más frecuencia en la etapa del análisis de sistemas. Uno de sus símbolos principales es el rectángulo que se usa para representar una entidad externa, es decir, un elemento del sistema (*hardware*, una persona, otro programa, etc.) u otro sistema que produzca información para ser transformada por el *software* o que reciba información que se aplica a los datos (o al control) y los cambia de alguna forma. Todas las flechas de un diagrama de flujo de datos deben estar etiquetadas. La línea doble representa un almacén de datos (información almacenada que es utilizada por el *software*); el círculo representa un transformador de información (una función) que reside dentro de los límites del sistema a ser modelado [Fai94].

La segunda propuesta es más utilizada en la etapa del diseño de sistemas. Usa una caja (cuadrado o rectángulo) para indicar un paso del proceso, un rombo para representar una condición lógica y flechas para señalar el flujo de control. La combinación de estos elementos arrojan nuevos elementos como son el case (selección) y los bucles (repeat-until y do-while).

La tercera propuesta (diagrama entidad-relación de Chen) tiene un uso más común para la representación de bases de datos relacionales. Sin embargo, el esquema ha sido aprovechado en el análisis y diseño de productos de *software*. Las entidades se representan mediante rectángulos y las relaciones mediante flechas y rombos. Cada entidad interviene estableciendo una relación con una determinada cardinalidad (número de ocurrencias de un objeto que se relaciona con las de otro).

Las técnicas de diagramación antes mencionadas son de gran utilidad durante la construcción de un producto de *software* y ayudan en gran medida a su mantenimiento; sin embargo, no son completamente eficaces en la representación gráfica de los sistemas transaccionales vía tecnología *web*.

A continuación se mencionan algunas de las carencias que se identificaron:

- Representación de tags.
- Representación de operaciones SQL.
- Representación de componentes (Applets, ActiveX, Scripts, Etc) en paginas html.
- Representación de eventos tales como (onchange, Meta, Target, Submit, Etc.).
- No enlazan los diagramas con la ubicación de los archivos.
- No permiten a un DFD llegar a un nivel bajo de abstracción.
- No enlazan los diagramas con la base de datos y las tablas que se usarán en el sistema.

Para subsanar tales deficiencias, en esta tesis se propone un técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*.

Esta técnica de diagramación incorpora iconos para representar a los principales transformadores de información (*tags*, eventos, componentes, etc.) en tecnología *web*, así como a los elementos más importantes de los sistemas transaccionales (operaciones SQL). Además la técnica propuesta incluye lineamientos para documentar los diagramas generados con la misma y las modificaciones que éstos sufran durante el mantenimiento.

Una característica adicional de la técnica es que realiza un enlace entre los diagramas y los archivos que formarán parte del código. Lo cual facilita la localización de la correspondencia diagramas-código y simplifica el mantenimiento del sistema.

La tesis está estructurada de la siguiente manera: en el capítulo 1 abarco lo relacionado con la tecnología *web*, desde la aparición de Internet, hasta su evolución a extranet. Los sistemas transaccionales es el tema del capítulo 2. En el capítulo 3 muestro técnicas de diagramación convencionales; en el 4 aplico estas técnicas a un sistema transaccional vía tecnología *web;* el 5 es la propuesta de una técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*. El capítulo 6 es la aplicación de la técnica propuesta.

En este capítulo se hablará de Internet, de su evolución a intranet y a extranet; se mencionarán las características más importantes de cada concepto.

1.1. Internet

La característica principal de Internet es la de ser un sistema universal de comunicaciones que permite que cualquier tipo de equipos puedan comunicarse entre sí de forma transparente, mediante el empleo de todo tipo de redes (locales, metropolitanas, extendidas), de tecnologías (*ethernet, token ring*, red telefónica, RDSI, etc.) y de medios físicos de transmisión (fibra óptica, cable de cobre, ondas de radio, etc.). Esto es posible gracias al conjunto de normas y la familia de protocolos TCP/IP Esta denominación engloba más de 100 normas o protocolos abiertos (no dependen de ninguna casa comercial) que se han convertido en el estándar *de facto* para la comunicación entre ordenadores. Cada uno de los protocolos englobados da solución a algún tipo de comunicación entre ordenadores, esto es, definen cómo funcionan aplicaciones concretas, por ejemplo, correo electrónico, transferencia de ficheros, etc. [Boh00].

Actualmente Internet es utilizado para aplicaciones militares, científicas, educacionales, comerciales, etc.

La tecnología no permanece estática. Hablar de Internet es hablar de servicios telemáticos interactivos y multimedia. La red proporciona a los usuarios medios interactivos para encontrar y compartir información.

Los nuevos desarrollos en cuanto a ancho de banda, en muchos casos conducidos por la demanda creciente para telecomunicaciones interactivas cada vez más rápidas, impondrán cambios importantes en Internet y permitirán, junto con el desarrollo de procesadores cada vez más potentes, poder trabajar en entornos multimedia a través de la red. Un papel importante en el incremento del ancho de banda lo desempeña el empleo de la fibra óptica.

Se puede augurar una explosión en los servicios personales, de entretenimiento, educación y formación, con unas cifras de negocio importantes. Los usuarios tendrán además de éstas, otras ventajas, como poder acceder a una gran cantidad de información y grandes posibilidades de comunicación vía correo electrónico, servicio que está evolucionando para poder transmitir voz e imágenes, lo cual tiene una clara aplicación en las relaciones ciudadano-administración (teleadministración) [hVli].

El comercio electrónico es otra de las áreas de negocio con mayor futuro en Internet, pese a que se cuenta con un problema tan importante como el de la seguridad. Es necesario que las aplicaciones incorporen funciones de cifrado, aspecto fundamental en la comunicación empresa-consumidor -por ejemplo, en el caso de enviar números de tarjetas de crédito o información

sobre un producto determinado-. Hay quien asegura que nunca existirá un 100% de seguridad y siempre habrá que asumir un cierto nivel de riesgo similar al del mundo físico [hInt].

En realidad es arriesgado predecir el futuro de Internet. Nuevos avances en *hardware, software* y comunicaciones se desarrollan constantemente, por lo que resulta difícil predecir cuál será la tecnología dominante.

1.2. Intranet

La popularización de Internet ha llevado a muchas empresas a plantearse la construcción de sus sistemas de información basados en los estándares de esa red; lo que en la literatura especializada se conoce con el nombre de intranet. Por lo tanto, las herramientas usadas para crear una intranet son idénticas a las de Internet y las aplicaciones *web*. La diferencia principal es que al acceso a la información publicada está restringido a clientes dentro del grupo de la intranet [Jen98].

Establecer una intranet supone hablar de un potente sistema de comunicación, de reducción de costes, de mayor productividad y calidad, de mejorar la relación con proveedores y clientes, de obtener mayor información. Además, la instalación de una red interna dentro de la empresa supone una reconciliación con cuatro mundos dispersos: sistemas de información y bases de datos (sistemas de compra, finanzas), documentación técnica (planes, *software*, etc.), comunicación (correo electrónico, revistas, etc.) y el mundo exterior [hGeo].

La intranet propone el concepto de usar el paginador de *web* como la interface de información universal. Las ventajas de este nuevo paradigma se mencionan a continuación:

- Reduce el tiempo de aprendizaje de los usuarios.
- Simplifica la instalación de aplicaciones.
- Presenta diferentes tipos de información: texto, gráficas, sonido y vídeo.
- Actúa como "front-end" para las aplicaciones cliente-servidor.
- Permite el acceso a bases de datos.

Una de las principales motivaciones para la adopción de la intranet es que permite a las organizaciones evolucionar de una estrategia de publicación calendarizada a una publicación con base en la demanda [hInt].

Tradicionalmente, las compañías publican una vez al año el manual del empleado. Cualquier cambio de último momento o ajuste importante, sería actualizado hasta el siguiente año. La intranet ofrece dos soluciones a este problema:

- El empleado decide cuándo consultar la información.
- La información puede actualizarse instantáneamente.

1.3. Extranet

La extranet surge a partir de la combinación del entorno cerrado y seguro de intranet y de la red pública Internet.

Existen dos configuraciones básicas de extranet. La primera es arrendar una línea directa, donde una empresa puede tener un completo control físico sobre la línea que enlaza una intranet con otra. La segunda, que es más utilizada, consiste en establecer una extranet mediante un vínculo seguro en Internet. Si se usa la infraestructura existente de intranet e Internet, una extranet se convierte tan sólo en una cubierta lógica, definida solamente por los privilegios de acceso y las tablas de ruteo, y no en una red física completamente nueva. En el segundo esquema, construir una extranet resulta económico y relativamente sencillo [hSea].

Los estándares abiertos de la tecnología de Internet han hecho que en la actualidad la creación y adopción de las extranets sea uno de los conceptos más prometedores de colaboración empresarial. El escenario ideal de una extranet requiere el despliegue uniforme a través de los entornos de la intranet e Internet. De esta manera, un grupo seleccionado de empresas enlazadas puede colaborar utilizando las tecnologías estándares de Internet, disfrutando al mismo tiempo de la confidencialidad y autonomía de un entorno de intranet [Deb99].

Algunas de las aplicaciones de las extranets se muestran a continuación:

- Grupos privados que cooperan con la empresa y que comparten la misma información e ideas.
- Entornos de colaboración donde algunas empresas participan en el desarrollo de una aplicación nueva que ellos pueden usar.
- Programas de formación u otros contenidos educativos que las empresas pueden desarrollar o compartir.
- Listas de catálogos de productos.
- Gestión de proyecto y control para empresas que forman parte de un mismo esquema de trabajo.

Una extranet debe servir para integrar una amplia variedad de bases de datos a la tecnología *web*, generalmente con el uso del navegador *web* como interfaz. El uso de extranets como foros para compartir bases de datos puede volverse un medio eficaz para realizar transacciones empresariales decisivas [Eth00].

A través del uso de *firewalls* físicas y virtuales, contraseñas, encriptación y diversas formas de autentificación del usuario, la tecnología web (Internet, intranets y extranets) debe ser capaz de administrar la seguridad y confiabilidad. [hSum].

1.4. Resumen

En este capítulo se habló sobre la tecnología *web*. El primer concepto que se trató fue Internet, una red de comunicaciones de área amplia, que ha sido aprovechada para entablar comunicación desde diferentes partes del mundo. Internet ofrece múltiples beneficios, entre los más conocidos están los siguientes: acceso a correo electrónico, transferencia de archivos,

recopilación y publicación de información en formato html, centros de noticias, foros de discusión, etc.

Con la aparición de Internet, las empresas se dieron cuenta de los grandes beneficios que ofrecía este nuevo paradigma de comunicación y lo económico que resultaba su implementación en una red local, razón por la cual decidieron transformar sus sistemas cliente/servidor a tecnología Internet, dando origen al concepto de intranet.

Una intranet es un conjunto de aplicaciones implementadas con tecnología Internet en redes locales. En este tipo de aplicaciones se utilizan los navegadores como interfaz de usuario, servidores de páginas *web*, servidores de aplicaciones y sistemas manejadores de bases de datos.

El potencial que dan las intranets en las empresas es superior en relación con el esquema cliente/servidor que antes se tenía. Sin embargo no se podía mantener este ambiente totalmente aislado de Internet, por lo que nace un nuevo concepto: extranet.

Una extranet es la combinación de los esquemas Internet e intranet. Como es de suponerse, la seguridad aquí cobra mayor importancia, ya que la globalización de las aplicaciones puede causar grandes problemas, razón por la cual sólo tendrá acceso personal autorizado.

CAPÍTULO 2. SISTEMAS TRANSACCIONALES

En este capítulo se hablará de transacciones, sistemas transaccionales y del aprovechamiento de tecnología *web* en este tipo de sistemas.

2.1. Definición de transacción

Una transacción es un conjunto de operaciones que comparten un fin común: transformar su estado actual en un nuevo estado de consistencia. Fueron originalmente desarrolladas para ser utilizadas dentro de los sistemas de base de datos, donde se usaban para auxiliar en el mantenimiento de los datos y de las aplicaciones.

Las transacciones ayudan a simplificar la construcción de sistemas confiables a través de procesos que proveen soporte uniforme para invocar y sincronizar operaciones como la manipulación de bases de datos [Rog98].

2.2. Propiedades de las transacciones

Atomicidad. Para que una transacción se ejecute con éxito es necesario que se realicen todas sus operaciones internas; si alguna no se ejecuta, la transacción completa no debe dejar evidencia de sus operaciones, esto es, se realizan todas las operaciones de la transacción o ninguna.

Durabilidad. Si una transacción fue realizada con éxito, los resultados producidos no se perderán aunque existan daños posteriores.

Seriabilidad. La seriabilidad permite que si varias transacciones locales o distribuidas hacen uso concurrente de recursos, cada transacción será ejecutada una tras otra de manera lógica, asegurando que cada entidad haya usado datos consistentes.

Aislamiento. El aislamiento permite que cada entidad oculte los resultados intermedios que producen las operaciones internas de una transacción, de tal manera que otras entidades no los usen ya que serían datos inconsistentes.

2.3. Procesamiento de transacciones

El procesamiento de transacciones consiste en un conjunto de modificaciones a recursos del sistema, donde se define un punto de inicio y un punto de terminación, que definen un bloque de operaciones a realizar. Dentro de este proceso los demás usuarios no pueden hacer modificaciones hasta que los datos sean estables [hArt].

En la figura 2-1 se muestra un diagrama de una transacción normal con inicio, operaciones y fin de la transacción.

Las transacciones pueden depender de otras, es decir, que en el cuerpo de una transacción pueden existir otras tal como se muestra en la figura 2-2. Este tipo de transacciones se denominan *anidadas*.

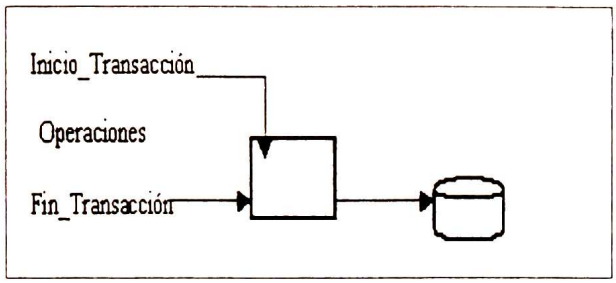


Figura 2-1. Cuerpo de una transacción.

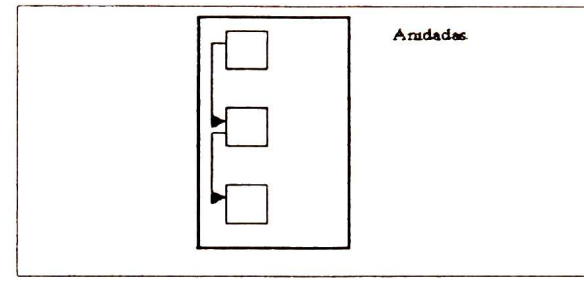


Figura 2-2. Transacción anidada.

2.4. Sistemas transaccionales vía tecnología *web*

En esta tesis se adopta el concepto de sistemas transaccionales como la aplicación o conjunto de aplicaciones que interactúan con bases de datos, es decir, la manipulación de información.

Para dar mayor alcance a este tipo de sistemas han sido implementados en tecnología *web*. Así nace un nuevo concepto: sistemas transaccionales vía tecnología *web*, mismo que será utilizado a lo largo de este trabajo.

Los sistemas transaccionales vía tecnología *web* proporciona grandes beneficios a las empresas. Entre los más comunes se encuentran los siguientes:

- Enriquece los paradigmas de comunicación y colaboración en la empresa.
- Facilita la publicación y el acceso a la información corporativa.
- Permite estandarizar una interface única para acceder a la información y a las aplicaciones corporativas.
- Trae tecnología multimedia al alcance de todos.
- Adaptable a los sistemas de información.
- La interface cliente (*web browser)*, es una herramienta sencilla y poderosa que permite llevar a cabo una serie de operaciones que de otro modo deberían utilizar una serie de programas especiales.
- La actualización de los programas se ejecuta sólo en el lado servidor, evitando el soporte al cliente.
- Los servidores *web* típicamente no necesitan tanto poder computacional como otras aplicaciones.
- Los avances en el estándar html son muy rápidos y van transformando el lenguaje en un ambiente de programación mucho más poderoso.
- La tecnología *web* es una excelente, avanzada, sencilla y atractiva manera de publicar información.

- La tecnología *web* es escalable y puede ser aplicada fácilmente a través de redes de área amplia como también en redes pequeñas y medianas.
- Accesible para la mayoría de las plataformas de cómputo.
- Rápido diseño.
- Integra la estrategia de cómputo distribuido.

Entre las aplicaciones más comunes de un sistema transaccional vía tecnología *web* se encuentran las siguientes:

- Comercio electrónico.
- Servicios al cliente (ejemplo: seguimiento de paquetes postales).
- Servicios financieros.
- Búsqueda de información.
- Acceso remoto a bases de datos.
- Creación de documentos html personalizados y dinámicos.
- Distribución de multimedia.

El esquema común de un sistema transaccional vía tecnología *web* se muestra a continuación:

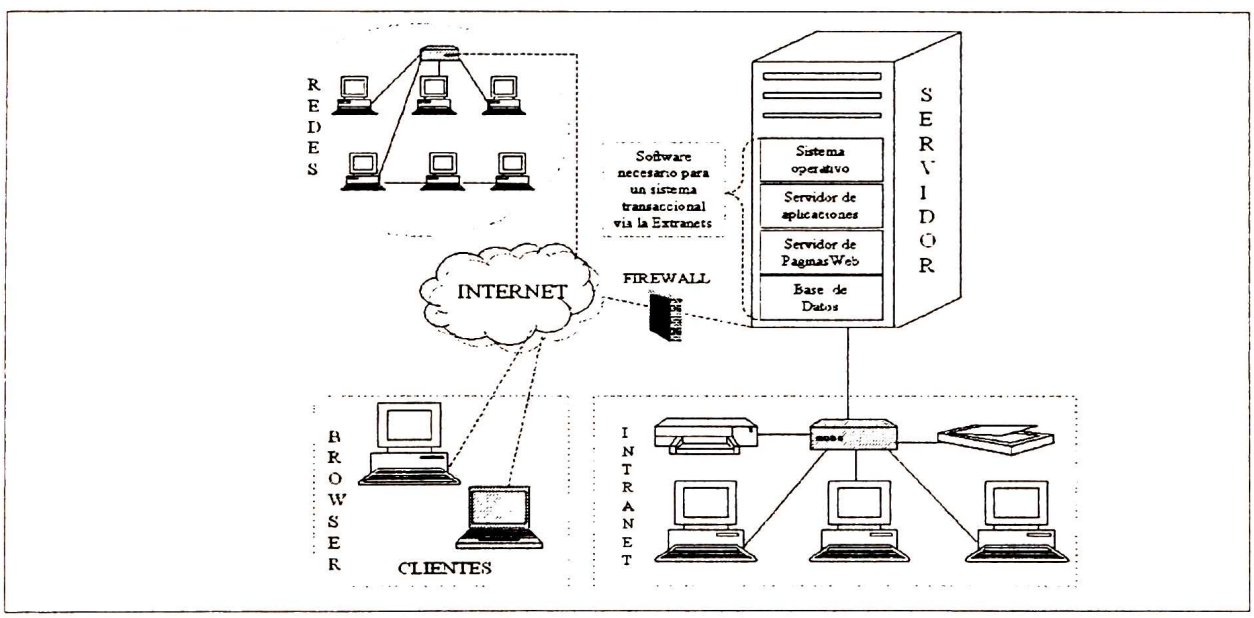


Figura 2-3. Esquema común de un sistema transaccional vía tecnología *web*.

En la figura 2-3 podemos observar los elementos que forman un sistema transaccional vía tecnología *web:* del lado del servidor tenemos el sistema operativo, servidor de aplicaciones, servidor de página *web* y el sistema manejador de bases de datos. Todos ellos trabajan en conjunto para formar la infraestructura (*software*) que soporta este tipo de sistemas transaccionales. Del lado del cliente sólo se necesita una computadora que tenga acceso a Internet. A continuación se describen brevemente estos elementos:

Sistema operativo. Un sistema operativo actúa como gestor de los recursos del sistema computacional, tales como el procesador, la memoria, los archivos y los dispositivos de E/S. En esta función, el sistema operativo lleva la cuenta del estado de cada recurso y decide quién lo obtiene, por cuánto tiempo y en qué momento. El sistema operativo resuelve las peticiones conflictivas de recursos para preservar la integridad [hSun].

Servidor de aplicación. Un servidor de aplicaciones hace posible el acceso a las bases de datos a través de la *web* [hIfw].

Base de datos. Las bases de datos están presentes en numerosas aplicaciones, sobre todo en aquéllas que manejan grandes cantidades de información que deben ser almacenadas y manipuladas de forma rápida y eficiente [hLea].

Servidor web. Este tipo de servidores son los que hacen posible que se puedan publicar páginas en Internet [hLea].

Cliente web. Para establecer la comunicación entre clientes y servidores en la *web* se hace uso del protocolo http. De cara a acceder a los servidores *web* y visualizar documentos de hipertexto, es necesario disponer de un programa cliente capaz de comunicarse con dichos servidores. Estos programas se llaman navegadores (browsers) [hCyb].

Otros Servidores. Otros servidores utilizados en los sistemas transaccionales vía tecnología web proporcionan funcionalidad de correo electrónico, grupos de noticias, transferencia de archivos, etc. Estos servidores se instalan de acuerdo a las necesidades del sistema transaccional.

2.5. Resumen

En este capítulo se definió una transacción como un conjunto de operaciones que comparten un fin común: transformar su estado actual en uno nuevo de consistencia, se mencionaron sus principales características: atomicidad, durabilidad, seriabilidad y aislamiento.

Se introdujo el término sistemas transaccionales vía tecnología *web* como una aplicación o conjunto de aplicaciones que utilizan transacciones (manipulación de bases de datos) para su funcionamiento y que son desarrollados sobre tecnología *web* (Internet, intranet y extranet).

Se mencionaron los principales beneficios (facilidad de publicación y acceso a la información corporativa, tecnología multimedia al alcance de todos, etc.) que proporcionan los sistemas transaccionales vía tecnología *web*, así como sus aplicaciones más comunes (comercio electrónico, acceso remoto a bases de datos, etc.).

Finalmente, se describieron de manera breve los elementos (sistema operativo, servidor de paginas *web*, servidor de aplicaciones, bases de datos, clientes *web* y otros servidores) que forman la infraestructura (*software*) que soportan las aplicaciones transaccionales vía tecnología *web*.

CAPÍTULO 3. TÉCNICAS DE DIAGRAMACIÓN CONVENCIONALES

La información se transforma a medida que fluye por un sistema (*software*) el cual acepta entradas en una gran variedad de formas, aplica elementos de hardware, software y humanos para transformar la entrada en salida, y produce salida en una gran variedad de formas.

El diagrama de flujo de datos (DFD) es una técnica que representa el movimiento de la información y las transformaciones que se aplican a los datos al trasladarse de la entrada a la salida. El DFD es también conocido como grafo de flujo de datos o como diagrama de burbujas. Este tipo de diagrama se puede usar para representar un sistema o *software* a cualquier nivel de abstracción. De hecho los DFD pueden ser divididos en niveles que representen un mayor flujo de información y un mayor detalle funcional [Rog98].

En este capítulo se presentan las principales técnicas de diagramación de flujo de datos.

3.1. Diagramas de flujos de datos de Gane & Sarson

Gane & Sarson propusieron un técnica de diagramación de flujo de datos para modelar el movimiento de la información en productos de *software*. La técnica no proporciona un gran nivel de abstracción y es usada para mostrar de forma general el flujo de los datos a través de un sistema de *software* [Fai94].

Los símbolos que forman la notación de la técnica son el rectángulo que actúa como un consumidor o productor de información que reside fuera del sistema que se va a modelar; el círculo que es un transformador de información y reside dentro del sistema; el flujo de datos es representado por una flecha donde la cabeza de la misma indica la dirección del mismo. Por último el almacenamiento de información o depósito de datos es representado por dos líneas paralelas.

A continuación se muestra la notación básica DFD propuesta por *Gane & Sarson.*

Notación	Descripción
Entidad	Un productor o consumidor de información que reside fuera de los límites del sistema a ser modelado.
Proceso	Un transformador de información (una función) que reside dentro de los límites del sistema a ser modelado.

Notación	Descripción
Flujo	Un flujo de datos (la cabeza de la flecha indica la dirección del mismo).
Almacén de datos	Un depósito de datos que se va a conservar para ser usado por uno o varios procesos; puede ser tan simple como una memoria intermedia o cola, o tan sofisticada como una base de datos relacional.

Figura 3-1. Notación básica DFD de Gane & Sarson.

La sencillez de la notación DFD propuesta en la figura 3-1 es una de las razones por las que esta técnica es ampliamente utilizada. Sin embargo es importante señalar que el diagrama no proporciona ninguna indicación explícita de la secuencia de procesamiento. El procesamiento o la secuencia puede estar implícitamente en el diagrama.

La notación propuesta por Gane & Sarson que se usa para desarrollar un DFD no es suficiente para describir los requisitos del *software*. Por ejemplo, una flecha de un DFD representa un objeto de datos que entra o sale de un proceso y un almacén de datos representa alguna colección organizada de datos, pero ¿cuál es el contenido de los datos implicados en las flechas o en el almacén? Si la flecha (o el almacén) representan una colección de objetos, ¿cuáles son? Finalmente la notación gráfica debe ser ampliada con texto descriptivo.

3.2. Diagramas de flujos de datos de Dijkstra, Bohm y Jacopini

Los fundamentos del diseño procedimental se afianzaron con el trabajo de Dijkstra, Bohn y Jacopini. A finales de los años 60 propusieron el uso de construcciones lógicas con las que podría formarse cualquier programa. Dichas construcciones hacían hincapié en el mantenimiento del dominio funcional, es decir, todas tenían una estructura lógica predecible, se entraba por arriba y se salía por abajo, permitiendo al lector seguir el flujo procedimental más fácilmente [Inc93].

Las construcciones son: la secuencia, la condición y la repetición. La secuencia implementa los pasos de procesamiento esenciales en la especificación de un algoritmo, la condición proporciona los medios para seleccionar el procesamiento basándose en alguna ocurrencia lógica y la repetición proporciona los bucles. Estas tres construcciones son fundamentales para la programación estructurada.

A continuación se muestra la notación gráfica propuesta por Dijkstra, Bohm y Jacopini.

Notación	*Descripción*
Secuencia	La secuencia se representa por una caja (cuadrado, rectángulo) y una línea (flecha) de control. Sirve para indicar un paso del proceso.

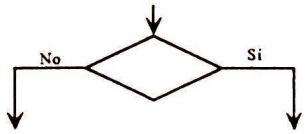
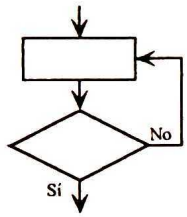
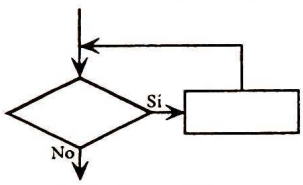
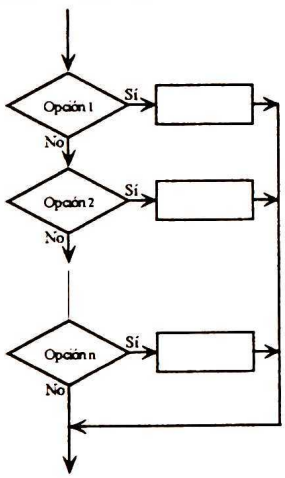
Notación	Descripción
 Condición *(if-then-else)*	La condición, también denotada *if-then-else*, se representa con un rombo de decisión que si es cierto provoca el procesamiento de la parte del *then*, y si es falso, invoca el procesamiento de la parte del *else*.
 Bucle (*repeat-until*)	Un *repeat-until* ejecuta primero la tarea de bucle, después prueba la condición y repite la tarea hasta que falla la condición.
 Bucle (*do-while*)	El *do-while* prueba una condición y ejecuta una tarea de bucle repitiendo mientras se mantenga cierta la condición.
 Selección (*select-case*)	La construcción selección (o *select-case*) es de hecho una extensión de *if-then-else*. Un parámetro se prueba por decisiones sucesivas hasta que ocurre una condición verdadera y se ejecuta un camino de procesamiento de parte case.

Figura 3-2. Notación básica DFD de Dijkstra, Bohm y Jacopini.

Las construcciones estructuradas son fragmentos lógicos que permiten al lector reconocer elementos procedimentales de un módulo en lugar de leer el diseño o el código línea a línea. La comprensión mejora cuando se encuentran formas lógicas fácilmente reconocibles.

Es indiscutible que las herramientas gráficas, tales como los diagramas de flujo de datos proporcionan una excelente forma gráfica que describe muy bien el detalle procedimental. Sin

embargo, si se emplean mal las herramientas gráficas, una imagen incorrecta puede conducir a un *software* erróneo.

3.3. Diagrama entidad-relación de Chen

El modelo entidad-relación (E-R) fue propuesto por Chen a mediados de los años 70 para representar conceptualmente los problemas de *software* y mostrar la visión de un sistema de forma global. Físicamente adopta la forma de un grafo escrito en papel al que se denomina *diagrama entidad-relación* (DER). Sus elementos fundamentales son las entidades y las relaciones [Law91].

Una entidad es todo aquello de lo que se desea almacenar información. Caracteriza a un tipo de objeto, real o abstracto, del problema a modelar. Toda entidad es distinguible del resto de las entidades, tiene nombre y posee *atributos* definidos en un dominio determinado. En el diagrama E-R las entidades se representan mediante rectángulos.

Una *relación* es una asociación o conexión matemática entre varias entidades. Las relaciones también se nombran. Se representan en el diagrama E-R mediante flechas y rombos.

Las conexiones entre entidades y relaciones se establecen mediante una variedad de símbolos especiales que indican la cardinalidad y modalidad.

La cardinalidad es la especificación del número de ocurrencias de un objeto que se relaciona con las de otro.

La cardinalidad ha sido clasificada de la siguiente manera:

- Una a una (1:1).
- Una a muchas (1:M).
- Una a cero o muchas (1:0 o M).
- Muchas a muchas (M:N).
- Cero o muchas a cero o muchas (0 o M : 0 o N).

La modalidad de una relación es cero si no hay necesidad explícita de que ocurra, es decir, si es opcional. La modalidad es 1 si una ocurrencia de la relación es obligatoria.

A continuación se muestran la notación de la técnica de diagramación propuesta por Chen y sus colegas.

Notación	Descripción
Entidad	Representa una entidad.
Etiqueta-relación	Este símbolo debe ser etiquetado con la relación que une a las entidades.

Notación	Descripción
Relación 1: 1	Indica relación de uno a uno entre las entidades que están unidas por este símbolo.
Relación 1 : M	Indica relación de uno a muchos entre las entidades que están unidas por este símbolo.
Relación 1 : 0 o M	Indica relación de uno a cero o muchos entre las entidades que están unidas por este símbolo.
Relación M : N	Indica relación de muchos a muchos entre las entidades que están unidas por este símbolo.
Relación 0 o M : 0 o N	Indica relación de cero o muchos a cero o muchos entre las entidades que están unidas por este símbolo.

Figura 3-3. Notación propuesta por Chen.

El aspecto básico para elaborar un diagrama E-R es la determinación de entidades para lo cual se extraen los nombres comunes de la descripción verbal del sistema y entre ellos se escogen los que aporten información valiosa. Con el resto de nombres se utiliza el sentido común descartando los inútiles. En caso de duda, es mejor incluir una entidad que posteriormente se revele como innecesaria que perder información relevante para el problema.

Hay normas de sentido común a seguir cuando se dibuja un diagrama E-R. La primera es emplear preferentemente líneas rectas en las relaciones y evitar en lo posible que éstas se crucen. Se suele usar nombres para describir las entidades y verbos para las relaciones. Esto es lógico ya que las entidades se ponen en común cuando se realiza alguna acción. Los verbos empleados no necesariamente tienen que ser siempre infinitivos.

3.4. Aplicaciones más frecuentes de las técnicas convencionales

Las técnicas de diagramación convencionales ayudan a simplificar las fases del análisis, diseño y mantenimiento en el desarrollo de un producto de *software*. Pueden representar gráficamente a cualquier sistema de *software*, sin embargo el nivel de abstracción será diferente y en algunos casos demasiado general para cierto tipo de sistemas, como los orientados a tecnología *web* ya que en tales técnicas no se proporcionan características para esta tecnología (Internet, intranet y extranet).

A continuación se presenta una tabla que nos muestra el uso principal de cada técnica descrita en este capítulo.

		Gane & Sarson	Dijkstra y sus colegas (Bohm y Jacopini)	Chen
Fases en el desarrollo de un producto de software	Análisis	1	0	1
	Diseño	0	1	0
	Mantenimiento	1	1	1
Tipos de sistemas	Procedimentales	1	1	1
	Orientados a objetos	1	0	0
	Orientados a tecnología *web*	0	0	0

Figura 3-4. Principal uso de las técnicas convencionales.

En la figura 3-4 Podemos observar que ninguna de estas técnicas fue desarrollada para ser aplicada a sistemas orientados a tecnología *web*, por la cual no logran un alto nivel de abstracción en la fase del diseño arquitectónico.

3.5. Otras técnicas de diagramación

Diagramas de estructuras

Los diagramas de estructuras describen el sistema de programación como una jerarquía de partes y lo muestran gráficamente como un árbol. Las principales aportaciones a esta notación fueron hechas por Constantine, Yourdon y Myers [Ian98].

Un diagrama de estructura muestra las relaciones entre las unidades de programa sin incluir ninguna información acerca de su orden de activación. Su notación es la siguiente:

1. Un rectángulo con el nombre de unidad.
2. Una flecha que conecta a los rectángulos.
3. Una flecha con un círculo, con el nombre de los datos que se pasan entre los elementos del diagrama de estructura. Las flechas con círculos suelen dibujarse paralelas a las flechas que conectan los rectángulos del diagrama.

Diagramas de transición de estados

Este tipo de diagramas es utilizado para representar los cambios de proceso a proceso.

En el contexto de sistemas orientados a objetos deben considerarse dos estados:

1. El estado de cada objeto cuando el sistema ejecuta su función.
2. El estado del sistema observado desde el exterior cuando éste ejecuta su función.

El estado de un objeto adquiere en ambos casos características activas y pasivas. Un estado pasivo es simplemente el estado actual de todos los atributos de un objeto; el activo indica que el objeto entra en una transformación continua o proceso. Para lograr la transición de un objeto de estado activo a otro debe ocurrir un evento (a veces llamado disparador) [Ian98].

La notación está representada de la siguiente manera:

1. Un círculo representa el estado actual.
2. Un doble círculo representa un nuevo estado.
3. Una flecha representa un evento que transforma el estado.
4. Un círculo remarcado representa la acción tomada cuando el estado cambia.

Diagramas orientado a objetos

El diseño orientado a objetos se basa en la idea de utilizar el ocultamiento de información como principal criterio de descomposición y en el concepto de los tipos de datos abstractos. Se ocupa de las entidades del sistema y no de las actividades de procesamientos de datos.

Un objeto es una entidad que tiene un estado y un conjunto definido de operaciones que actúan sobre ese estado, el cual se representa mediante un conjunto de atributos del objeto. Las operaciones asociadas con el objeto proveen servicios a otros objetos (clientes) que los requieren para realizar alguna actividad de cómputo. Los objetos se crean de acuerdo con una definición de la clase a la que pertenece, que sirve como *template* para los mismos. Tal definición incluye la declaración de todos los atributos y servicios asociados con un objeto de esta clase [Ian98].

El ocultamiento de información dentro de los objetos implica que los cambios hechos a uno de ellos no afecta a otros de manera impredecible.

En los diagramas se utilizan principalmente el rectángulo y las flechas como notación básica para denotar los eventos, clases y flujo de procesamiento.

Diagramación de secuencias temporales (diagramas de tiempo)

Es utilizada para mostrar las diferentes etapas de procesamiento (operaciones) y la interrelación de eventos en cierta cantidad de tiempo.

En el diagrama que se genera por esta técnica se especifican las operaciones en un formato dependiente del tiempo que se empleará en cada una de ellas. Cada operación puede incluir a otras dentro de su ejecución [Ian98].

La notación básica de esta técnica de diagramación son las flechas, mismas que indican la operación y el tiempo necesario para su ejecución, así como la posible dependencia de otras operaciones.

Además de las técnicas de diagramación de flujo de datos mencionadas, existen muchas otras que en la mayoría de los casos resultan de la combinación de éstas; sin embargo, ninguna de ellas está orientada a sistemas basados en tecnología *web*.

3.6. Resumen

Existe una gran variedad de técnicas de diagramación (Gane & Sarson, Dijkstra, Bohm y Jacopini, Chen, etc.) que son utilizadas en las etapas de análisis, diseño y mantenimiento de productos de *software*. Sin embargo estas técnicas no son orientadas a sistemas transaccionales vía tecnología *web*, por lo cual no logran definir diagramas a un alto nivel de abstracción para este tipo de sistemas, es decir, no muestran detalladamente el flujo y la transformación de la información en ellos.

CAPÍTULO 4. APLICACIÓN DE LAS TÉCNICAS DE DIAGRAMACIÓN CONVENCIONALES

En este capítulo se aplicarán las técnicas de diagramación convencionales. Para esto se presenta el análisis del sistema transaccional vía tecnología *web objeto de aprendizaje actividades* que será utilizado.

El objeto de aprendizaje actividades es un recurso didáctico que tiene como finalidad proporcionar capacitación a distancia sin la supervisión de un profesor o instructor, puede abarcar cualquier tema y nivel educativo.

4.1. Análisis del objeto de aprendizaje actividades

Los elementos que se sugieren para desarrollar un tema en particular y que formarán parte del *objeto de aprendizaje actividades* son los siguientes:

- *Contenidos.* Son documentos html que tienen como objetivo mostrar información del tema que se está proyectando. Se permitirá hacer referencia a documentos que se encuentren en otros servidores por medio de direcciones url.

- *Preguntas.* Se presentarán a los usuarios para darles seguimiento de acuerdo con el tipo de respuestas que proporcionen. Se incorporarán los dos tipos de preguntas diferentes que se muestran a continuación:

Tipo 1. Solo una opción. Ejemplo.
¿Cuánto es 2+2?
○ 4
○ 5
○ 3
○ 1

Tipo 2. Más de una opción. Ejemplo.
¿Que números son mayores de 5?
☐ 6
☐ 4
☐ 7
☐ 2

- *Evaluación en forma de árbol.* A los usuarios se le presentarán situaciones y posibles alternativas a ellas. Cada opción seleccionada llevará al usuario a una nueva situación y así hasta terminar. El cierre de esta evaluación se hará mediante retroalimentación.

Las especificaciones del sistema se mencionan a continuación:

- El sistema dará un trato diferente a cada usuario que se conecte, dependiendo de la hora en que se conecte, de las opciones que elija, del tema en particular. Dicho control se llevará a cabo mediante el *login* y *password* del usuario, mismos que servirán para dar acceso al sistema y registrar toda acción realizada por el usuario.

- La concurrencia es ilimitada en el sistema, queda sujeta a la capacidad del equipo en que se instale el sistema.

- Deberá tener un historial de elementos visitados y un índice para ir directamente a algún elemento determinado que forme parte del *objeto de aprendizaje actividades.*

- Toda la información que se presente en el *objeto de aprendizaje actividades* estará almacenada en una base de datos.

- El flujo de la información en el *objeto de aprendizaje actividades* dependerá de la ruta que el instructor programe en él. Existen diferentes rutas que pueden ser recorridas por los usuarios, de acuerdo con las decisiones que se tomen.

A continuación se muestra el DFD (diagrama de flujo de datos) del sistema transaccional vía tecnología web (*objeto de aprendizaje actividades*), tomando en cuenta la notación propuesta por Gane & Sarson.

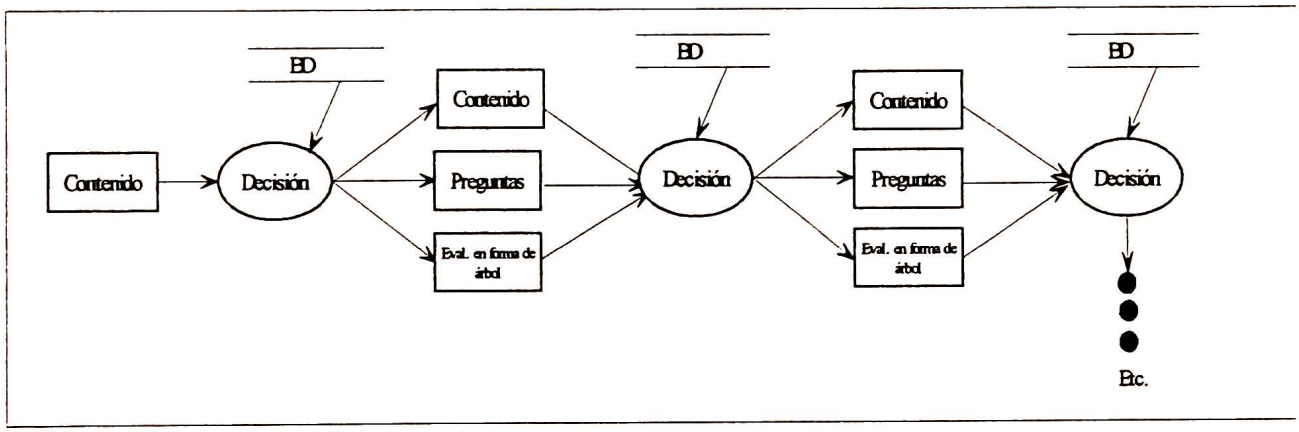


Figura 4-1. DFD del análisis del *objeto de aprendizaje actividades.*

Como podemos observar, un *objeto de aprendizaje actividades* siempre inicia con un contenido, donde posiblemente se muestren las instrucciones a seguir para recorrerlo. Es importante señalar que de uno de estos elementos se puede ir a cualquier otro en la siguiente iteración; la determinación del elemento que seguirá dependerá de las decisiones que el usuario tome en el *objeto de aprendizaje actividades* y de la programación que se hizo para cada una de ellas. Dicha programación reside en la base de datos.

4.2. Diseño arquitectónico del *objeto de aprendizaje actividades* utilizando la técnica de Gane & Sarson

A continuación se presentan los DFD que se generan al utilizar la técnica propuesta por Gane & Sarson.

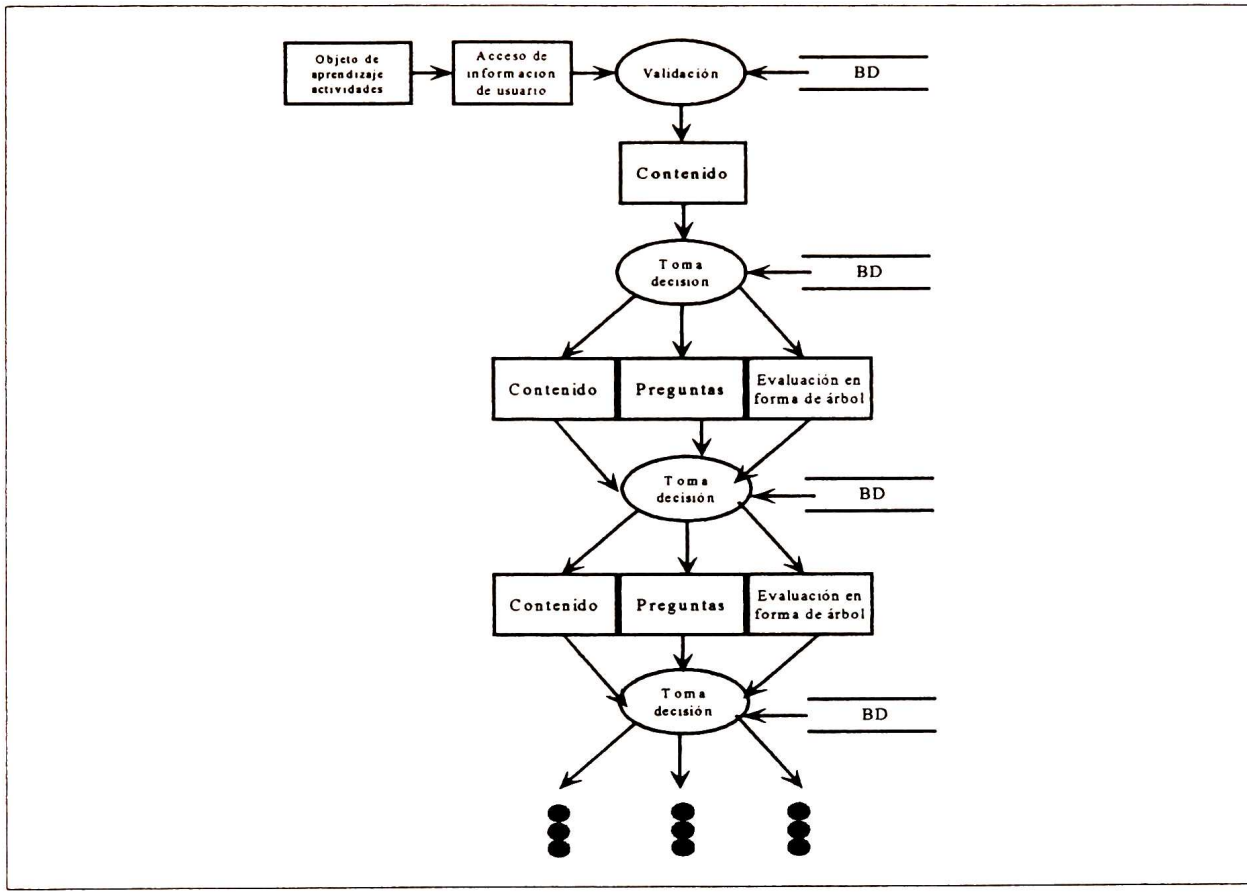


Figura 4-2. DFD del *objeto de aprendizaje actividades.*

A continuación se muestra el DFD de cada uno de los elementos que forman un *objeto de aprendizaje actividades*:

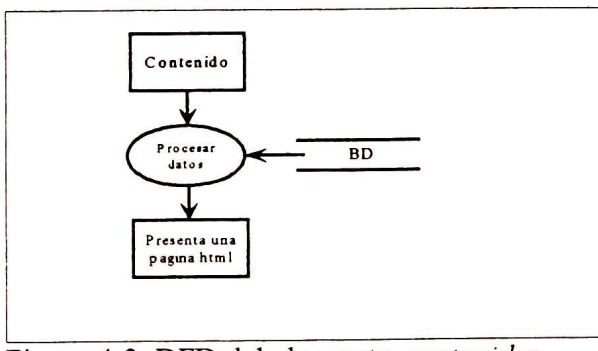


Figura 4-3. DFD del elemento *contenidos.*

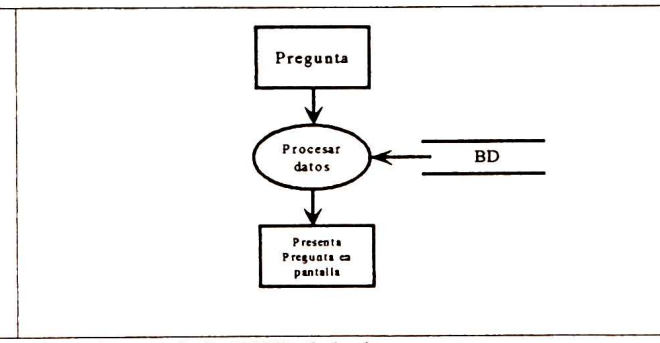


Figura 4-4. DFD del elemento *preguntas.*

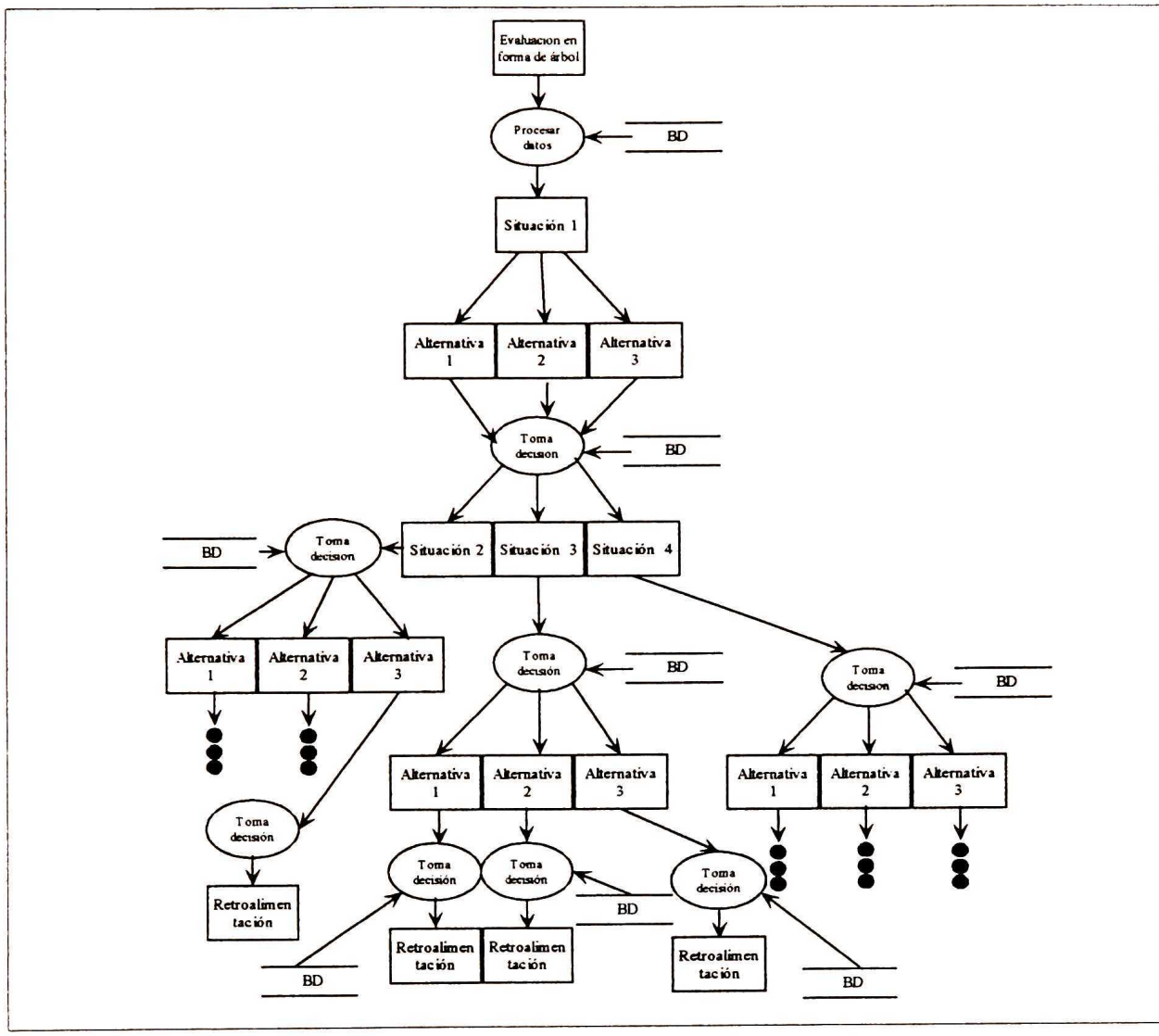


Figura 4-5. DFD del elemento *evaluación en forma de árbol.*

En estos diagramas podemos observar que el nivel de abstracción no es suficiente para entender el procesamiento interno del sistema. La razón es simple pues la notación no proporciona secuencia de procesamiento explícita. Esto hace que los diagramas proporcionen sólo una visión global del funcionamiento del sistema.

Esta técnica de diagramación es más utilizada en la fase del análisis de un sistema ya que ofrece una idea general de lo que se quiere hacer con el sistema sin especificar cómo se va a realizar.

La técnica no proporciona notación para representar el flujo y la transformación de datos en sistemas orientados a tecnología *web*.

4.3. Diseño arquitectónico del *objeto de aprendizaje actividades* utilizando la técnica de Dijkstra, Bohm y Jacopini

A continuación se presentan los DFD que se generan al utilizar la técnica propuesta por Dijkstra, Bohm y Jacopini para el sistema transaccional *objeto de aprendizaje actividades.*

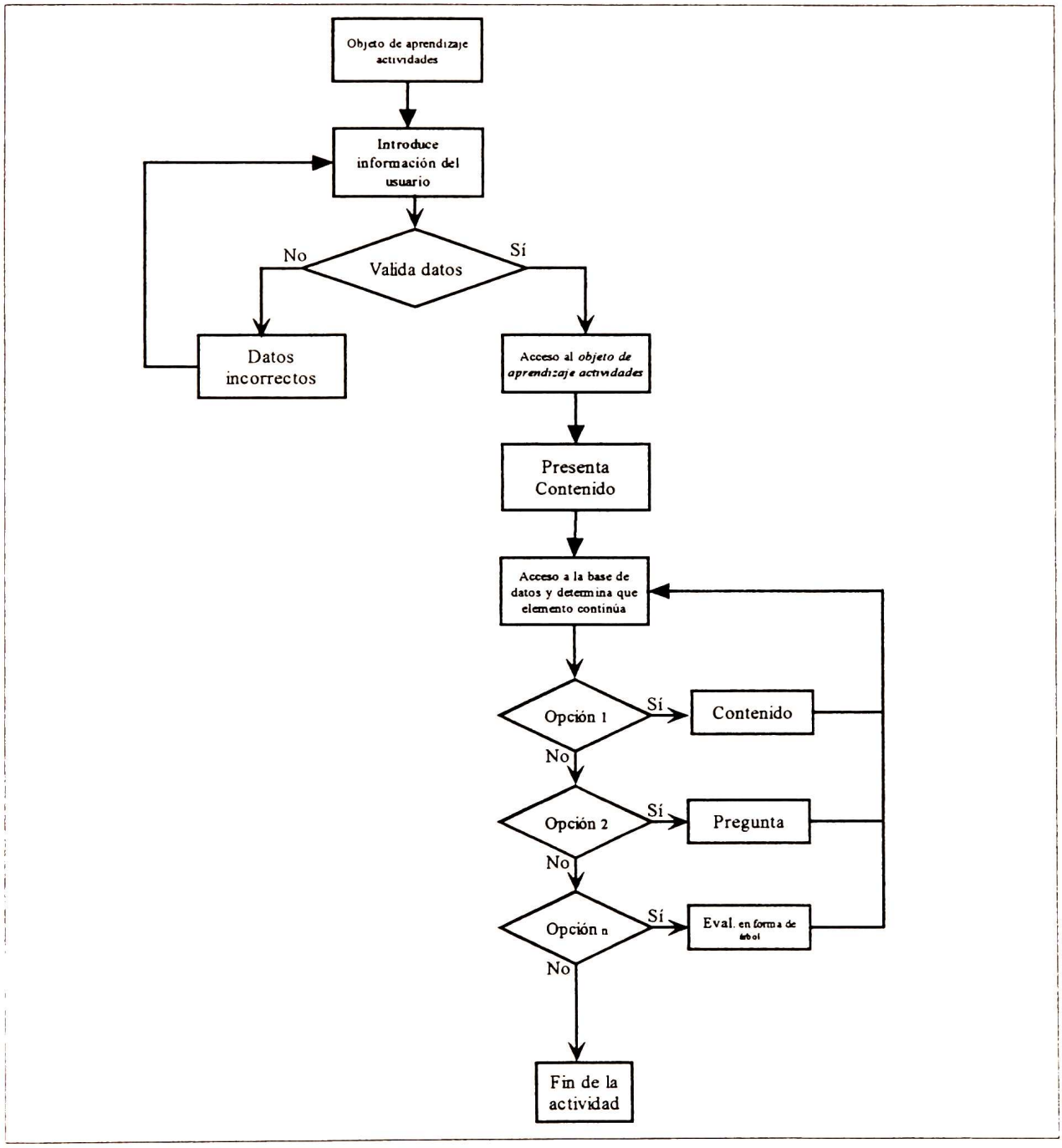


Figura 4-6. DFD del *objeto de aprendizaje actividades.*

A continuación se muestra el diagrama de cada uno de los elementos que forman un *objeto de aprendizaje actividades.*

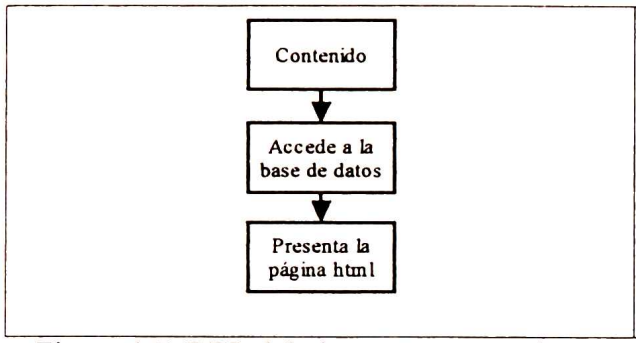


Figura 4-7. DFD del elemento *contenidos.*

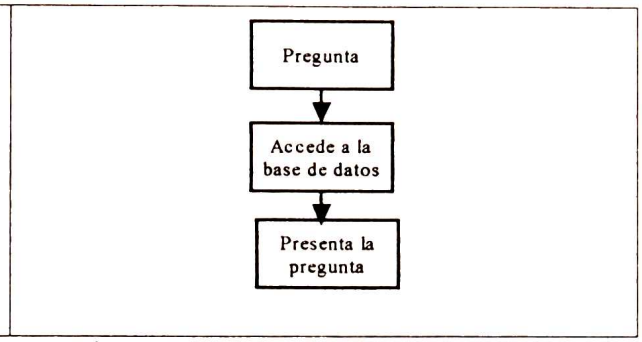


Figura 4-8. DFD del elemento *preguntas.*

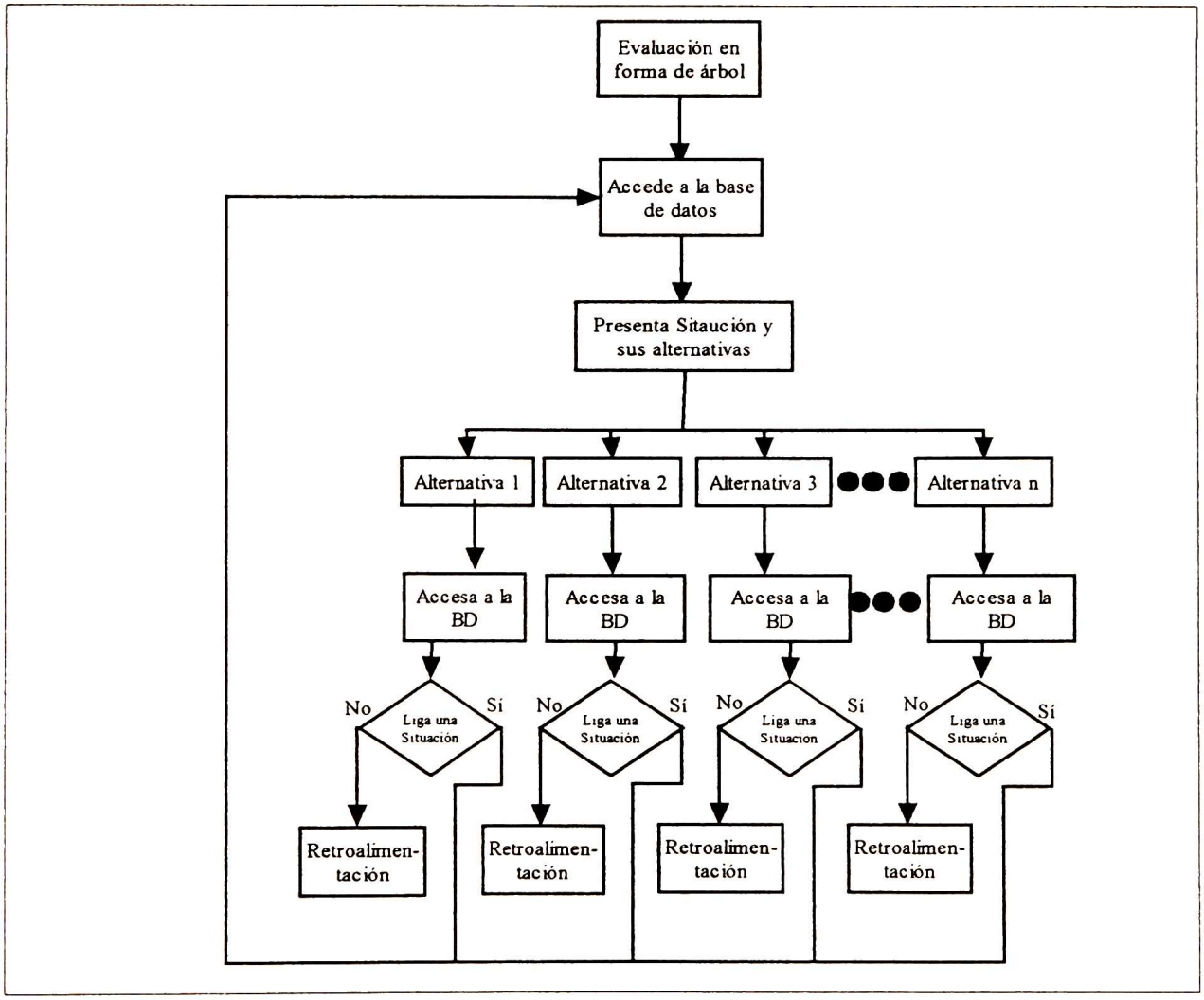


Figura 4-9. DFD del elemento *evaluación en forma de árbol.*

Estos diagramas son generados utilizando la técnica propuesta por Dijkstra, Bohm y Jacopini. En ellos podemos observar un alto nivel de abstracción para representar el flujo y la transformación de datos a través del sistema. Veamos también que no existen símbolos que representen procesos automáticos, operaciones SQL, tags, eventos, componentes, etc. lo cual no permite niveles altos de abastracción en el diseño arquitectónico de sistemas transaccionales vía tecnología *web*.

4.4. Diseño arquitectónico del *objetos de aprendizaje actividades* utilizando la técnica de Chen

A continuación se presenta los DFD que se generan utilizando la técnica propuesta por Chen para el sistema transaccional *objetos de aprendizaje actividades.*

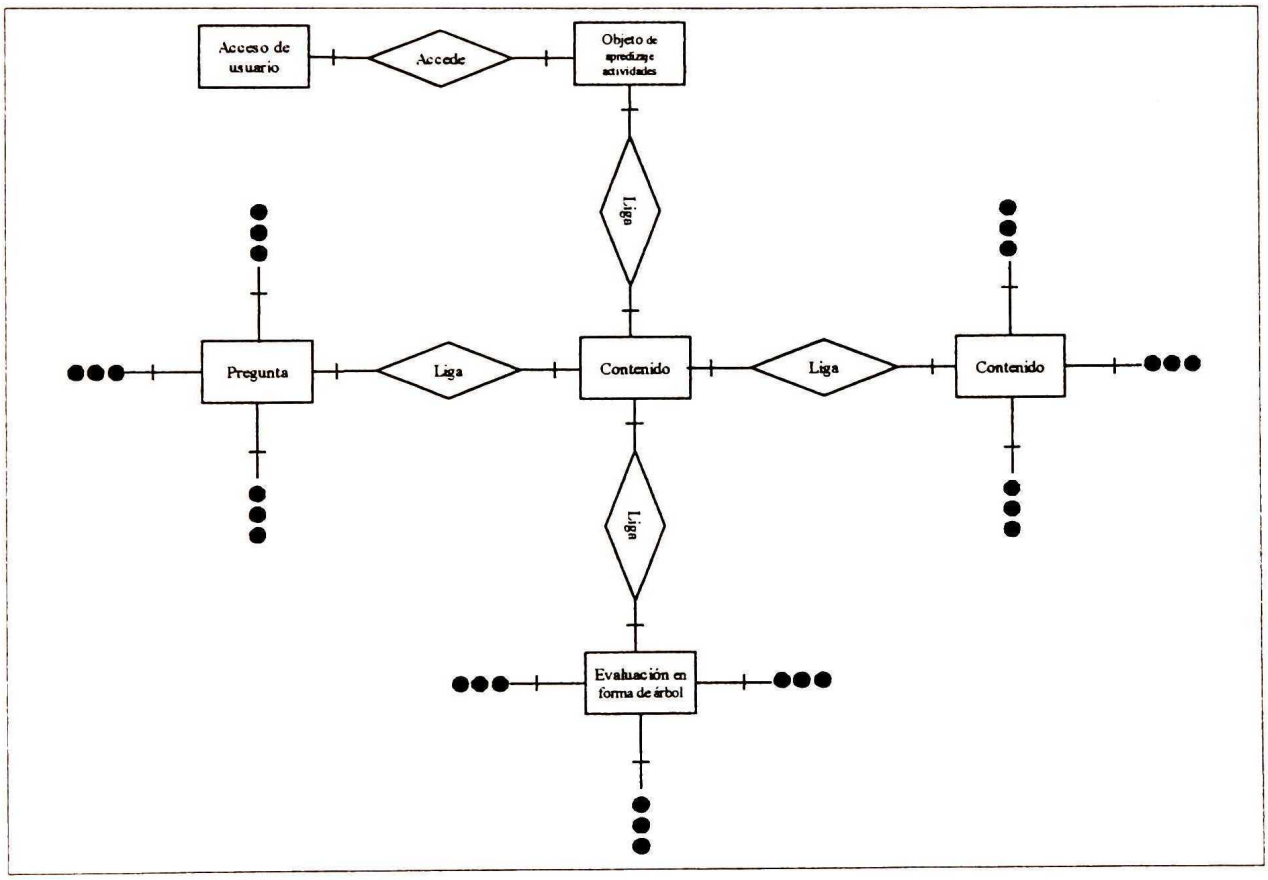


Figura 4-10. DFD del *objeto de aprendizaje actividades.*

A continuación se muestra el diagrama de cada uno de los elementos que forman un *objeto de aprendizaje actividades.*

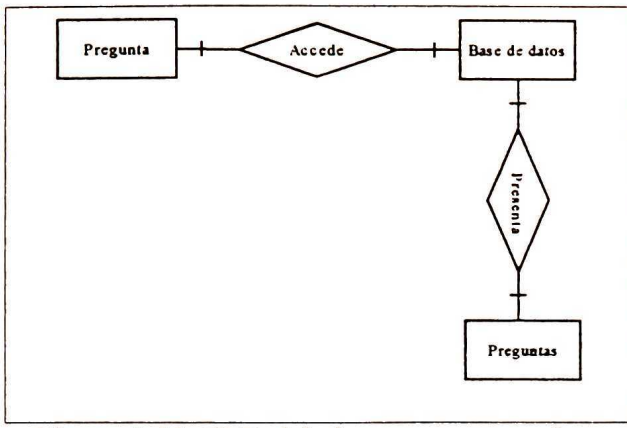


Figura 4-11. DFD del elemento *contenidos.*

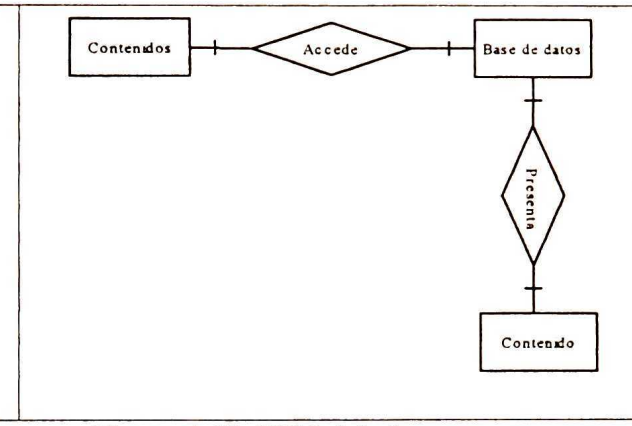


Figura 4-12. DFD del elemento *preguntas.*

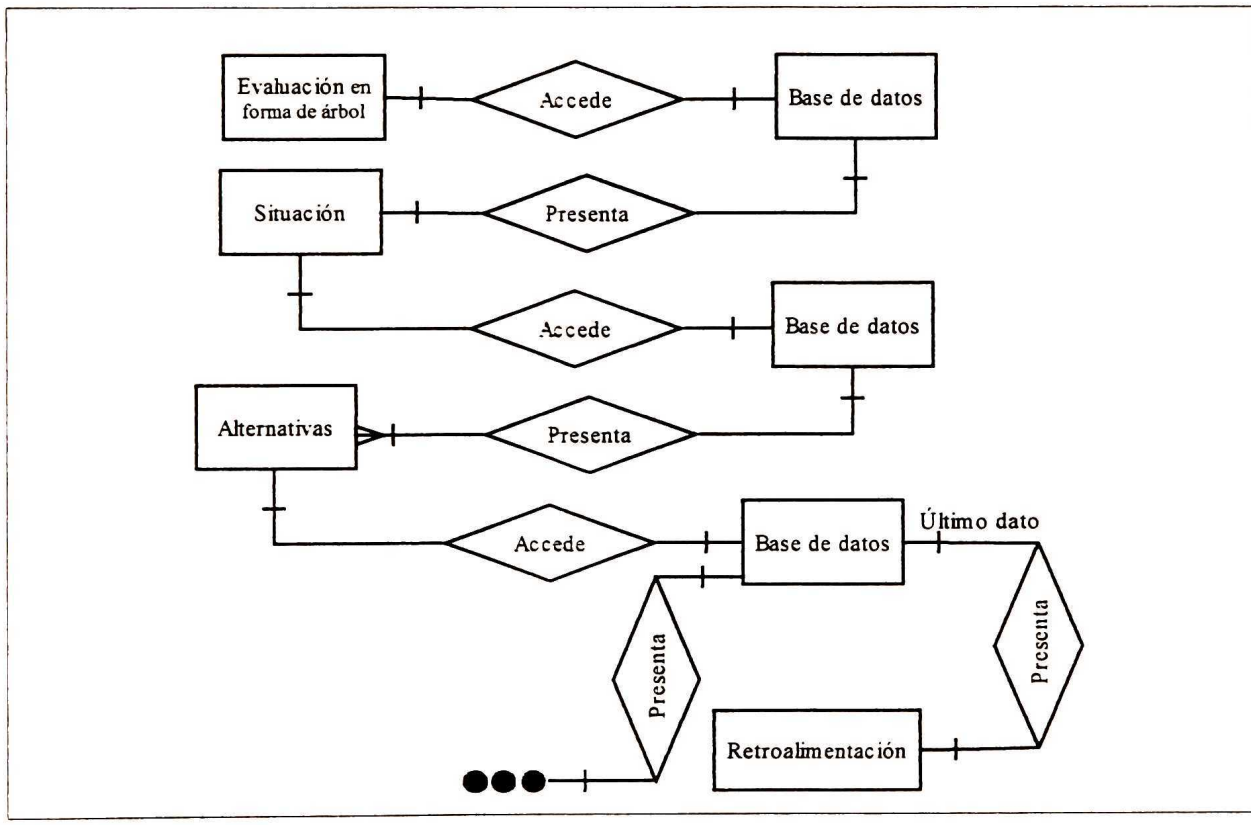


Figura 4-13. DFD del elemento *evaluación en forma de árbol.*

La técnica de Chen es utilizada en estos diagramas para representar el diseño arquitectónico del sistema transaccional *objeto de aprendizaje actividades.* Podemos observar que el nivel de abstracción es muy bajo y que el flujo y la transformación de los datos es mostrada de forma muy general.

La propuesta de Chen es utilizada con mayor frecuencia para diagramar esquemas entidad-relación de bases de datos relacionales, ya que fue propuesta para el modelado de datos.

Por tal razón no proporciona notación para representar elementos de la tecnología *web*, tales como eventos HTML, operaciones SQL, tags, componentes, etc.

Los diagramas generados con esta técnica muestran de forma general el flujo de la información en un sistema transaccional vía tecnología *web* y proporcionan un bajo nivel de abstracción.

4.5.- Comparación de las técnicas de diagramación convencionales aplicadas en un sistema transaccional vía tecnología *web*

En este capítulo hemos aplicado las técnicas de diagramación convencionales a un sistema transaccional vía tecnología *web*, de lo cual hemos detectado limitaciones.

A continuación se muestra una comparación de las técnicas de diagramación convencionales aplicadas en el diseño arquitectónico de un sistema transaccional vía tecnología *web*.

	Gane & Sarson		Dijkstra y sus colegas (Bohm y Jacopini)		Chen	
Representar graficámente	Alto nivel de abstracción	Bajo nivel de abstracción	Alto nivel de abstracción	Bajo nivel de abstracción	Alto nivel de abstracción	Bajo nivel de abstracción
Diseño estructurado	0	1	1	0	0	1
Bases de datos	0	1	0	1	0	1
Eventos HTML	0	0	0	1	0	0
Tags HTML	0	0	0	1	0	0
Componentes (Applets, ActiveX, OCX, etc.)	0	0	0	1	0	0
Operaciones SQL básicas (Insert, Update, Delete, Select)	0	0	0	0	0	0
Trabajos automáticos (Jobs)	0	0	0	1	0	0

Figura 4-14. Comparación de las técnicas de diagramación convencionales.

En la figura 4-14 Observamos que el nivel de abastracción que muestran las técnicas convencionales es bajo en casi todos los rubros y nulo en alguno de ellos.

Con esto se constata que las técnicas convencionales fueron creadas para sistemas que no soportan tecnología *web*.

4.6. Resumen

Al aplicarse las técnicas de diagramación convencionales -descritas en el capitulo 3- se demostró que no fueron creadas para ser usadas en sistemas orientados a tecnología *web*.

Las principales carencias identificadas en las técnicas convencionales se dan en la representación gráfica de operaciones SQL, eventos HTML, tags HTML, procesos automáticos, componentes, scripts y referencia a la base de datos que se va a utilizar.

CAPÍTULO 5. TÉCNICA DE DIAGRAMACIÓN PARA EL DISEÑO ARQUITECTÓNICO DE SISTEMAS TRANSACCIONALES VÍA TECNOLOGÍA WEB

El objetivo del diseñador es producir un modelo o representación de una entidad que se va a construir posteriormente. El proceso mediante el cual se desarrolla el módulo combina la intuición y los criterios basados en la experiencia en la construcción de entidades similares, un conjunto de principios y/o heurísticas que guían la evolución del módulo, un conjunto de criterios que permiten juzgar la calidad y un proceso de iteración (repetición) que lleva como fin último a una representación definitiva del diseño [Mcc00].

El diseño del *software* es la primera de las tres actividades técnicas -diseño, construcción y prueba- necesarias para construir y verificar el programa, cada actividad transforma la información de manera que se obtiene finalmente un *software* válido.

El diseño arquitectónico define la relación entre los principales elementos estructurales del programa y hace uso de los diagramas de flujo de datos (DFD) para representar gráficamente los sistemas que se van a construir [Mol93].

Las técnicas de diagramación de flujo de datos que actualmente existen (Gane & Sarson, Dijkstra, Bohm y Jacopini y la de Chen) presentan algunas carencias para representar sistemas transaccionales vía tecnología *web,* tales como la representación de tags, operaciones SQL, componentes (Applets, ActiveX, Scripts, etc.) en páginas html, eventos HTML (onchange, Meta, Target, Submit, etc.) y el enlace de los diagramas con la ubicación física de los archivos que forman parte del código del sistema.

En este capítulo se presenta una técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*, que subsana las deficiencias encontradas en otras técnicas de diagramación de flujo de datos.

5.1. Técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*

La técnica está orientada a sistemas transaccionales vía tecnología *web* e incorpora representación gráfica a los principales transformadores de información en este tipo de tecnologías, tales como *tags*, operaciones SQL, eventos HTML, componentes, *scripts*, enlace a base de datos, etc.

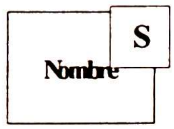
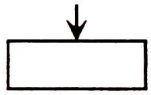
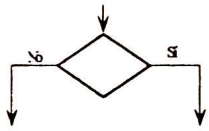
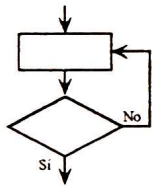
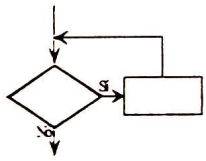
Además la notación incorpora los lineamientos para enlazar los diagramas con la ubicación física de los archivos, esto para agilizar la búsqueda de lo representado por el diagrama.

También se incorpora un esquema de encabezados para los diagramas generados por la técnica.

La notación de la técnica de diagramación propuesta se describe a continuación:

Notación	**Descripción**
Vacío	Indica que no se tiene documento alguno que mostrar en un *frame*.
Layer	Muestra las capas en la programación DHTML. Se debe incluir el nombre de la capa en la parte interna del símbolo.
En construcción	Indica que esta parte del módulo está en construcción. Se debe colocar la leyenda en construcción en la parte interna del símbolo.
Submit	Indica el botón que ejecuta un formulario. Se debe colocar el nombre del botón que ejecuta al formulario sobre el símbolo.
Liga	Indica las ligas (url) contenidas en los documentos. Se recomienda colocar palabras claves sobre el símbolo. Estas palabras claves deben estar relacionadas con el tipo de liga que representa y darnos la idea del tipo de documento al que apunta.
Directorio	Indica los directorios en los que están distribuidos los archivos que forman el proyecto. Se debe colocar el nombre del directorio y un número identificador del mismo en la parte interna del símbolo.
Flujo	Indica el flujo de los datos a través del sistema.
Frame	Indica los *frames*. Se recomienda colocar el nombre del *frame* sobre el símbolo.
Script	Indica el uso de código *javascript* o de un archivo con extensión "js". Se debe colocar el nombre del *script* o del archivo "js" en la parte interna del símbolo.

Notación	Descripción
Documento de *frames*	Representa un documento de *frames*. Se debe colocar el nombre de este archivo en la parte interna superior del símbolo. En la parte interna inferior se debe colocar el número del directorio donde está almacenado el archivo (documento de *frames*).
Documentos	Representa los documentos de cualquier extensión, ya sea html, doc, xls, ppt, etc. Se debe colocar el nombre del documento en la parte interna superior de este símbolo. En la parte interna inferior se debe colocar el número del directorio donde está almacenado el documento.
Frame interno	Indica los *frames* internos. Se recomienda indicar el nombre del *frame* sobre este símbolo.
Tag Meta	Indica el uso del tag *meta* ya sea *refresh* o para ejecutar una página nueva. Se debe colocar el nombre de la página que se ejecuta sobre el símbolo o indicar que es *refresh*.
Inicia Conector	Indica el inicio de un conector. Se deben utilizar letras dentro del símbolo.
Finaliza Conector	Indica la continuación del conector. Se deben utilizar letras dentro del símbolo.
Botón	Indica la utilización de un botón. Se debe colocar el nombre del botón dentro o sobre este símbolo.
Onchange	Indica el uso del evento *onchange*. Se debe colocar el nombre del formulario que procesa este evento.
Formulario o Forma	Indica la utilización de un formulario o forma. Se debe colocar el nombre de este formulario o forma dentro del símbolo.
Lanzar o Ejecutar	Indica la ejecución de un formulario dentro de otro. Se debe colocar el nombre del formulario que se ejecuta sobre el símbolo.
Target	Indica la utilización de la propiedad *target* de algunos objetos. Se debe colocar el nombre del *target* en la parte interna del símbolo.
Contiene	Indica que un elemento contiene a otros elementos.

Notación	Descripción
 Operación SQL	Representa las operaciones SQL. Se debe colocar la inicial de la operación SQL que se realiza S-*Select*, U-*Update*, I-*Insert* y D-*Delete*, además del nombre del *query* o elemento que realiza esta operación.
Componente	Representa los componentes utilizados. Se debe colocar el nombre del componente y el tipo del mismo. Ejemplo del tipo "Applet".
Base-Tabla	Representa la base de datos con la que se está conectando el sistema y la tabla en la que se lleva a cabo la operación.
 Secuencia	Representa el paso de un proceso o un proceso en sí.
 Condición	Representa condición. Aquí se aplica el procesamiento de uno u otro lado dependiendo del valor que arroje la condición evaluada.
 *Repeat-until*	Representa el bucle *repeat-until.* Primero se ejecuta el bucle, después se evalúa la condición. Ese bucle es utilizado cuando se desea que se ejecute al menos una vez su sentencia.
 *Do-while*	Representa el bucle *do-while*. Primero evalúa la condición y después ejecuta el bucle. Se utiliza cuando este último puede o no ejecutarse.

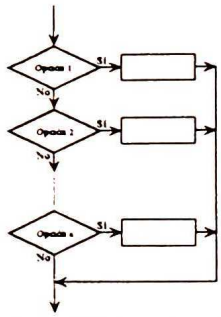
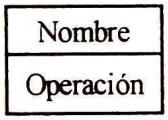
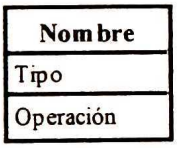
Notación	Descripción
 *Case* (Selección)	Representa una selección (*case*). Es una extensión de la condición y ejecuta la parte donde el valor comparado encuentra su correspondencia. Una vez que lo halla efectúa el procesamiento respectivo y termina su ejecución sin considerar las comparaciones subsecuentes.
Otro *Tag*	Representa los *tags* no definidos con los símbolos anteriores. Se debe colocar el nombre del *tag* y la operación que efectúa.
 Otro Evento	Representa los eventos no definidos con los símbolos anteriores. Se debe colocar el nombre del evento y la operación que efectúa
 Elemento no definido	Representa los elementos no considerados con los símbolos anteriores. Se debe colocar el nombre del elemento, el tipo de elemento y la operación que efectúa.

Figura 5-1. Notación de la técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*.

Esta notación fue implementada como una librería en un programa de diagramación de flujo de datos llamado *SmartDraw*, el cual está disponible en Internet (http://www.smartdraw.com).

A continuación se muestra la interfaz del diagramador seleccionado para implementar la técnica de diagramación propuesta. Se sugiere que se adopte este diagramador para generar los documentos (diagramas) de un sistema transaccional vía tecnología *web*.

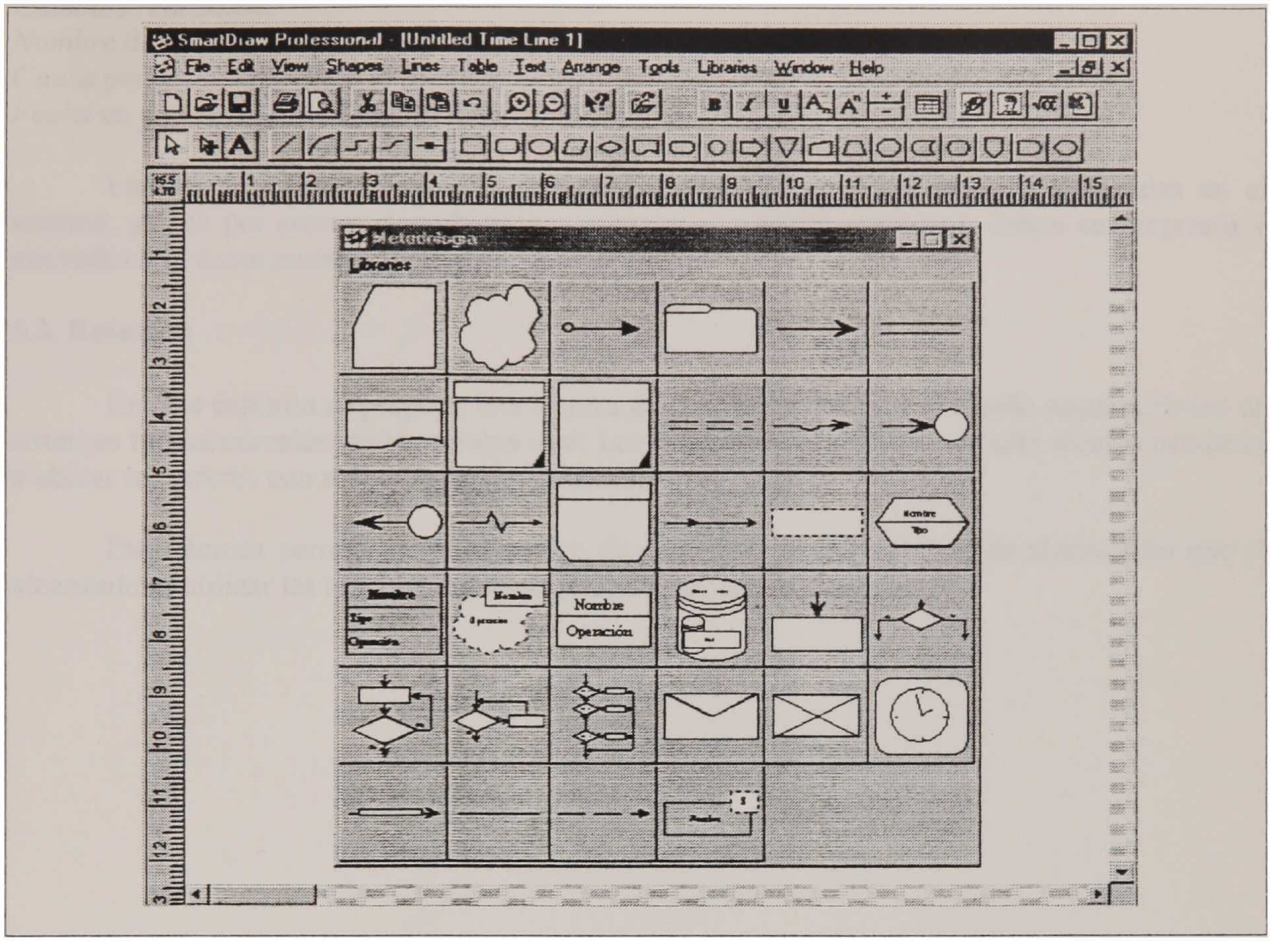


Figura 5-2. Implementación de la técnica de diagramación en una librería de *SmartDraw*.

Cabe hacer mención que esta técnica no está sujeta a ningún *software* en particular y que puede ser implementada en cualquiera que sirva para diagramar.

Cada diagrama generado por esta técnica deberá tener como encabezado la siguiente información:

Módulo que se está diagramando.
Nombre de la persona que diagramó el módulo.
Fecha en que se concluyó el diagrama del módulo.

En esta técnica también se establecen los lineamientos para controlar las modificaciones sobre los diagramas.

Primero se modifican los diagramas afectados, utilizando la simbología de la técnica propuesta.

Después se documentan los diagramas modificados con la siguiente información:

Cambios realizados.
Nombre de la persona que realiza la modificación.
Causa por la que se realiza la modificación.
Fecha en que se realiza la modificación.

Los documentos (diagramas) que fueron afectados por los cambios realizados en el sistema, ya sea por errores descubiertos o por nuevas implementaciones, deben ser impresos y anexados a la documentación existente del sistema.

5.2. Resumen

En este capítulo se propone una técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*. Los diagramas generados con esta técnica ayudarán a ubicar los errores con mayor rapidez y eficacia.

Esta técnica permite la construcción de un DFD con mayor nivel de abstracción que el alcanzado al utilizar las técnicas convencionales descritas en el capítulo 3.

CAPÍTULO 6. APLICACIÓN DE LA TÉCNICA PROPUESTA

En este capítulo se aplicará la técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*.

6.1. Diseño arquitectónico del objeto de aprendizaje actividades utilizando la técnica propuesta

El diseño arquitectónico del sistema transaccional tomará como base el analisis y las especificaciones del sistema *objeto de aprendizaje actividades* descritas en el capítulo cuatro.

El modelo entidad-relación de la base de datos que se empleará en el sistema transaccional vía tecnología *web* se muestra en la siguiente imagen.

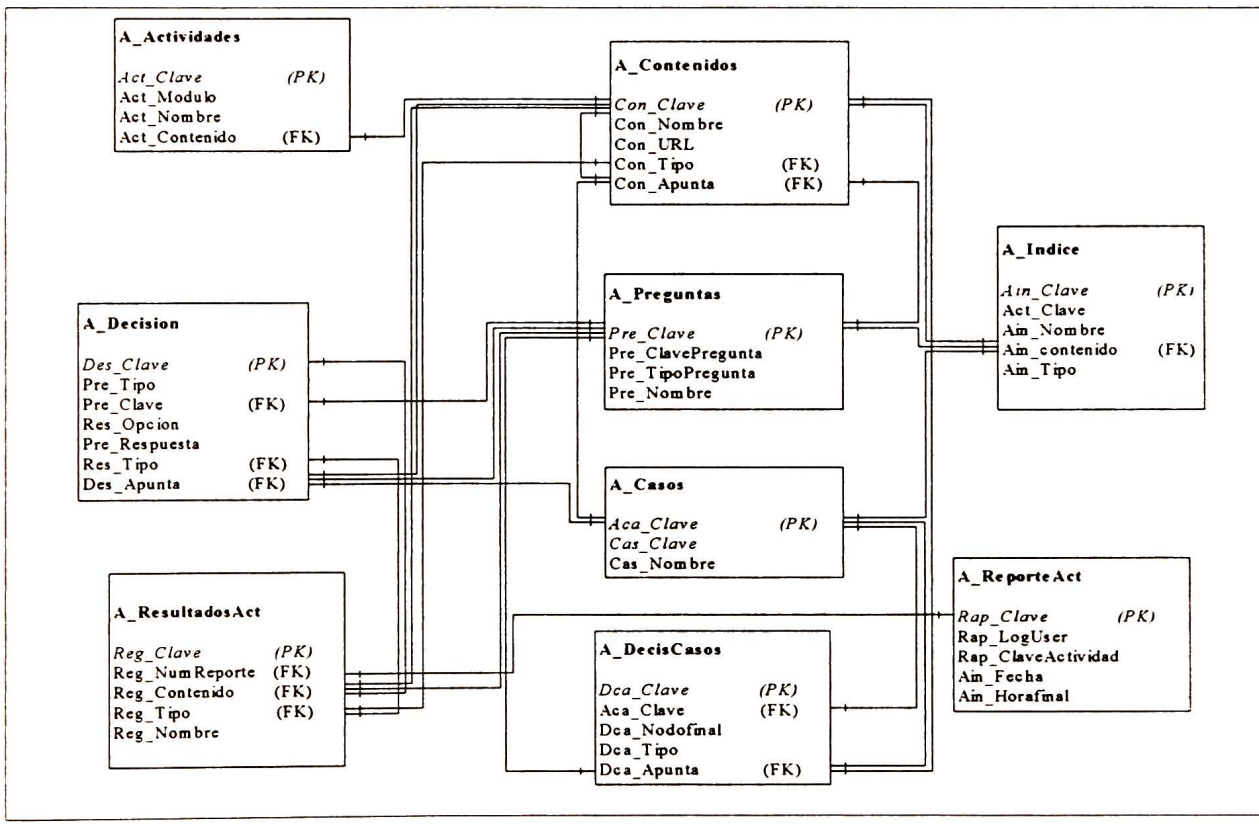


Figura 6-1. Modelo entidad-relación del sistema transaccional *objeto de aprendizaje actividades*.

En la figura 6-1 podemos observar las tablas que forman el modelo entidad-relación de la base de datos del sistema transaccional: las tablas **A_Decision** y **A_DecisCasos** guardan información para la toma de decisiones de la ruta que seguirá el usuario de acuerdo con sus acciones; las tablas **A_ReporteAct** y **A_ResultadosAct** registran la secuencia que el usuario está

siguiendo en el *objeto de aprendizaje actividades*; las tablas **A_Preguntas**, **A_Casos** y **A_Contenidos** almacenan los elementos -o la liga a éstos- que forman parte de un *objeto de aprendizaje actividades*. La tabla **A_Indice** guarda la información para la opción **Índice** del *objeto de aprendizaje actividades* y finalmente la tabla **A_Actividades** hace referencia a los *objetos de aprendizaje actividades*.

A continuación se muestra el diagrama del *objeto de aprendizaje actividades* generado con la técnica de diagramación propuesta en el capítulo anterior.

Primera parte del diagrama:

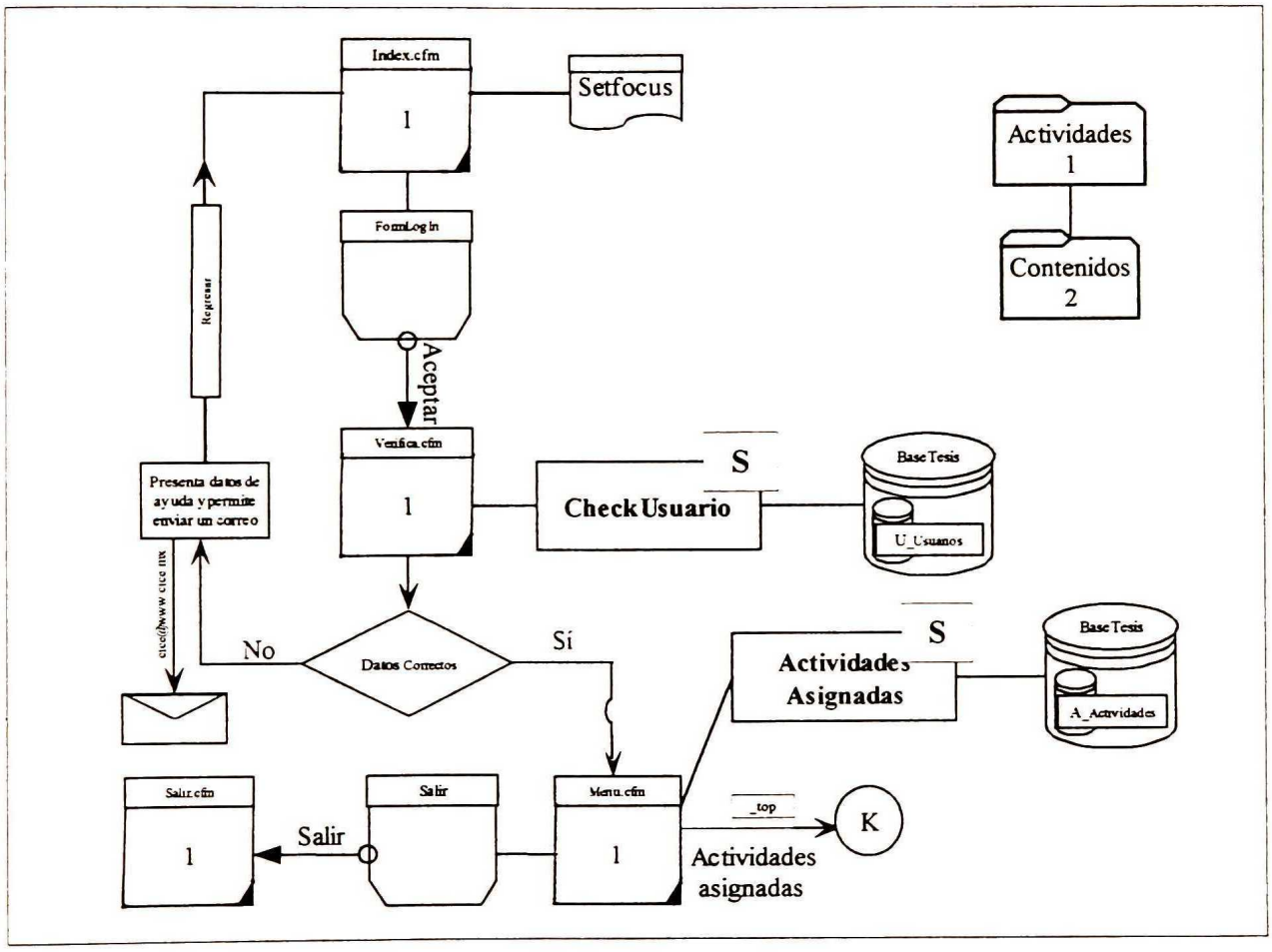


Figura 6-2. Primera parte del diagrama generado con la técnica propuesta.

En la figura 6-2 podemos observar que el *objeto de aprendizaje actividades* inicia con un documento llamado **Index.cfm**, el cual contiene un *script* y un formulario denominados **Setfocus** y **FormLogIn**, respectivamente. Este último recibirá los datos del usuario -*Login* y *Password*- para transferirlos al documento **Verifica.cfm** que comparará los datos del usuario con los obtenidos (Select) de la tabla **U_Usuarios** de la base de datos **BaseTesis** por medio del query **CheckUsuario**. El archivo **Verifica.cfm** compara los datos dando acceso al sistema si éstos son correctos, de lo contrario retroalimenta al usuario dándole dos opciones: enviar un correo por

medio del cual solicite ayuda o regresar a la página principal **Index.cfm** e introducir nuevamente los datos a ser validados.

Si los datos del usuario fueron correctos, se ejecuta automáticamente el archivo **Menu.cfm**, el cual selecciona los ejercicios del *objeto de aprendizaje actividades* de la tabla **A_Actividades** de la base de datos **BaseTesis** por medio del query **Actividades Asignadas**.

En la parte superior derecha podemos observar la estructura de los directorios **Actividades** y **Contenidos** lugares donde se almacenarán los archivos o documentos que formarán parte del sistema. Tales directorios tienen un número identificador que servirá para ser referenciados por los archivos que estén dentro de ellos.

Por último podemos observar que el archivo **Menu.cfm** cuenta con un conector inicial etiquetado con la letra **K** y que nace de una liga URL; cuando el usuario presiona esta liga se inicia el *objeto de aprendizaje actividades seleccionado.*

Segunda parte del diagrama:

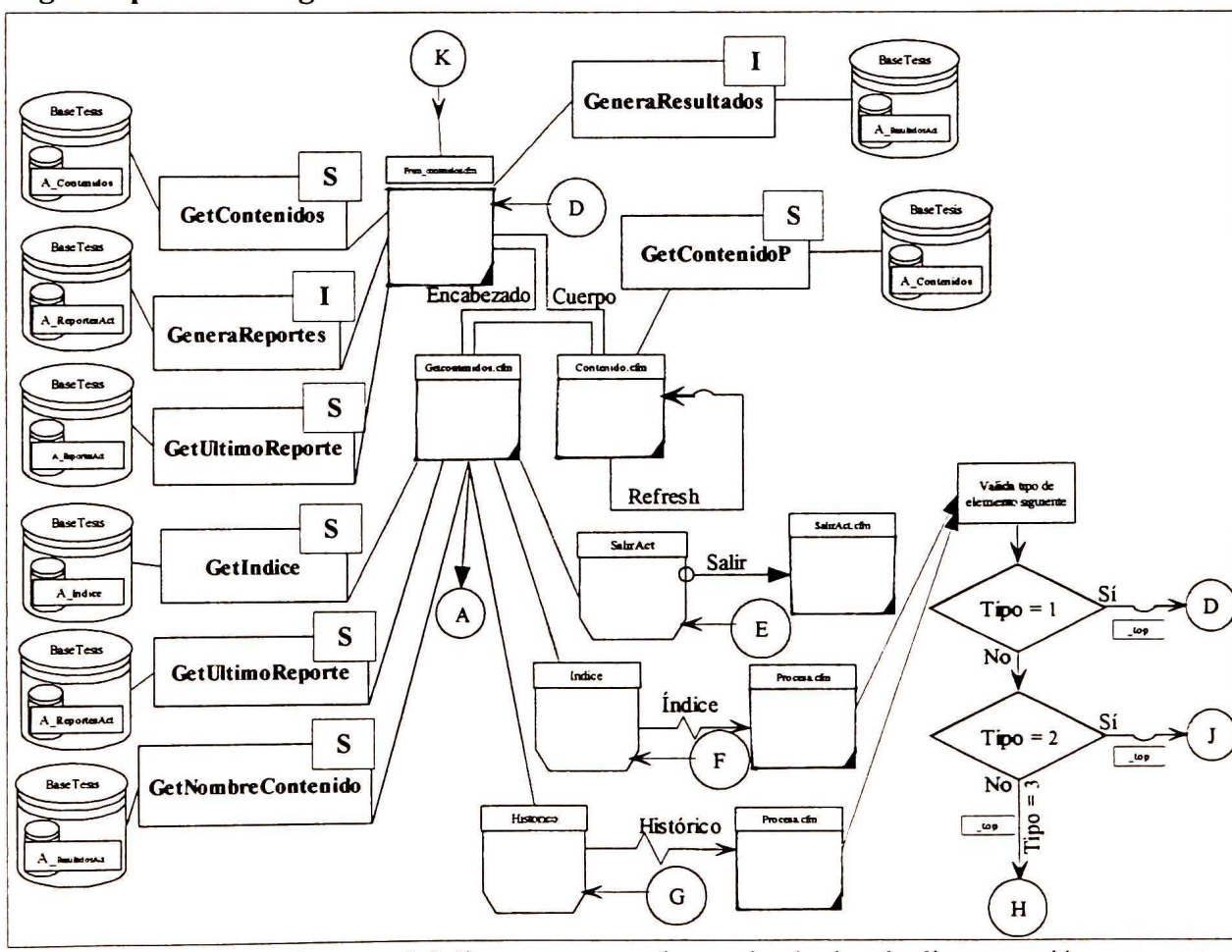


Figura 6-3. Segunda parte del diagrama generado con la técnica de diagramación propuesta.

Podemos observar el conector final etiquetado con la letra **K**, que viene de una liga URL del archivo **Menu.cfm**. Esta liga carga un ejercicio del *objeto de aprendizaje actividades*, el cual inicia con el archivo de *frames* **Fram_Contenidos.cfm**; en él se seleccionan datos de las tablas **A_Contenidos** y **A_ReportesAct** y se insertan datos en la tablas **A_ReportesAct** y **A_ResultadosAct**, todo esto en la base de datos **BaseTesis.** El documento **Fram_Contenidos.cfm** contiene dos *frames*: **Encabezado** y **Cuerpo**, que cargan los archivos **Getcontenidos.cfm** y **Contenido.cfm**, respectivamente.

El archivo **Contenidos.cfm** muestra la información del contenido, obtiene los datos de la tabla **A_contenidos** de la base **BaseTesis**.

El archivo **Getcontenidos.cfm** selecciona datos de las tablas **A_Indice**, **A_ReportesAct** y **A_ResultadosAct** de la base de datos **BaseTesis**. Contiene cuatro formularios: **Contenidos**, **SalirAct**, **Índice** e **Histórico**, cuando se presiona el botón **Salir** del formulario **Salir**, se carga un archivo **SalirAct.cfm** que termina la sesión del usuario en el ejercicio.

Cuando se presiona el botón **Aceptar** del formulario **Contenidos**, se valida la información para ejecutar el siguiente elemento que forma parte del *objeto de aprendizaje actividades* mismo que será explicado en la tercera parte de este diagrama y al cual se hace referencia por medio del conector inicial etiquetado con la letra **A**.

Los formularios **Índice** e **Histórico** son ejecutados por **onchage´s** mismos que cargan al archivo **Procesa.cfm** el cual valida el tipo de elemento que mostrará el ejercicio del *objeto de aprendizaje actividades* en la siguiente iteración. Si el tipo es 1 el elemento siguiente es un *contenido*, si es 2 entonces es una *pregunta* y finalmente si es 3 será una *evaluación en forma de árbol*.

Si el elemento que continúa es un *contenido*, entonces se inicia automáticamente la secuencia en el documento **Fram_Contenidos.cfm** y se siguen los mismos pasos ya mostrados anteriormente.

Si es una *pregunta*, entonces se carga automáticamente el documento de *frames* **Fram_Preguntas.cfm** -que se explicará más adelante-; se hace referencia a él por medio de un conector inicial etiquetado con la letra **D**.

Si es un *evaluación en forma de árbol*, entonces se carga el documento de *frames* **Fram_Casos.cfm** -que se explicará más adelante-; se hace referencia a él por medio de un conector inicial etiquetado con la letra **H**.

Tercera parte del diagrama:

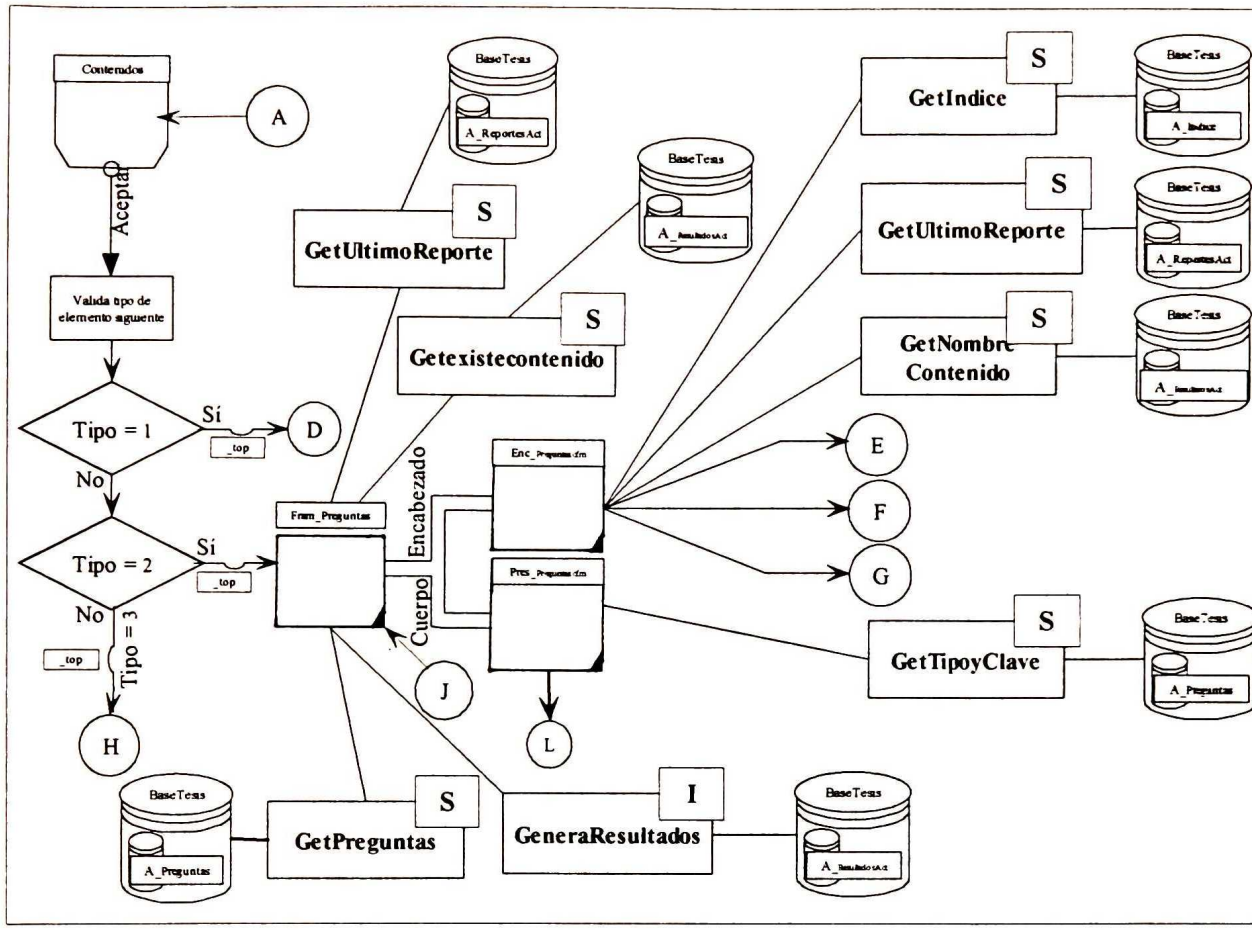


Figura 6-4. Tercera parte del diagrama generado con la técnica de diagramación propuesta.

Cuando se presiona el botón **Aceptar** del formulario **Contenidos**, se determina el elemento siguiente en el *objeto de aprendizaje actividades*. Si el tipo es 1 se carga automáticamente el documento **Fram_contenidos.cfm**, el cual ocupa toda la pantalla como lo indica el target **_top**; si el tipo es 2, entonces se carga automáticamente el documento de *frames* **Fram_Preguntas** el cual contiene dos *frames*: **Encabezado** y **Cuerpo**; en el encabezado se carga el documento **Enc_Preguntas.cfm** y en el cuerpo se carga el documento **Pres_Preguntas**.

El documento de *frames* **Fram_Preguntas.cfm** selecciona datos de las tablas **A_Reportes**, **A_ResultadosAct** y **A_Preguntas** e inserta datos en la tabla **A_ResultadosAct** todo esto en la base de datos **BaseTesis**.

El documento **Enc_Preguntas** carga datos de las tablas **A_Indice**, **A_ReportesAct** y **A_Resultados** de la base de datos **BaseTesis**; además contiene 3 formularios: **SalirAct**, **Índice** e **Histórico** mismos que fueron explicados en la segunda parte de este diagrama.

Cuarta parte del diagrama:

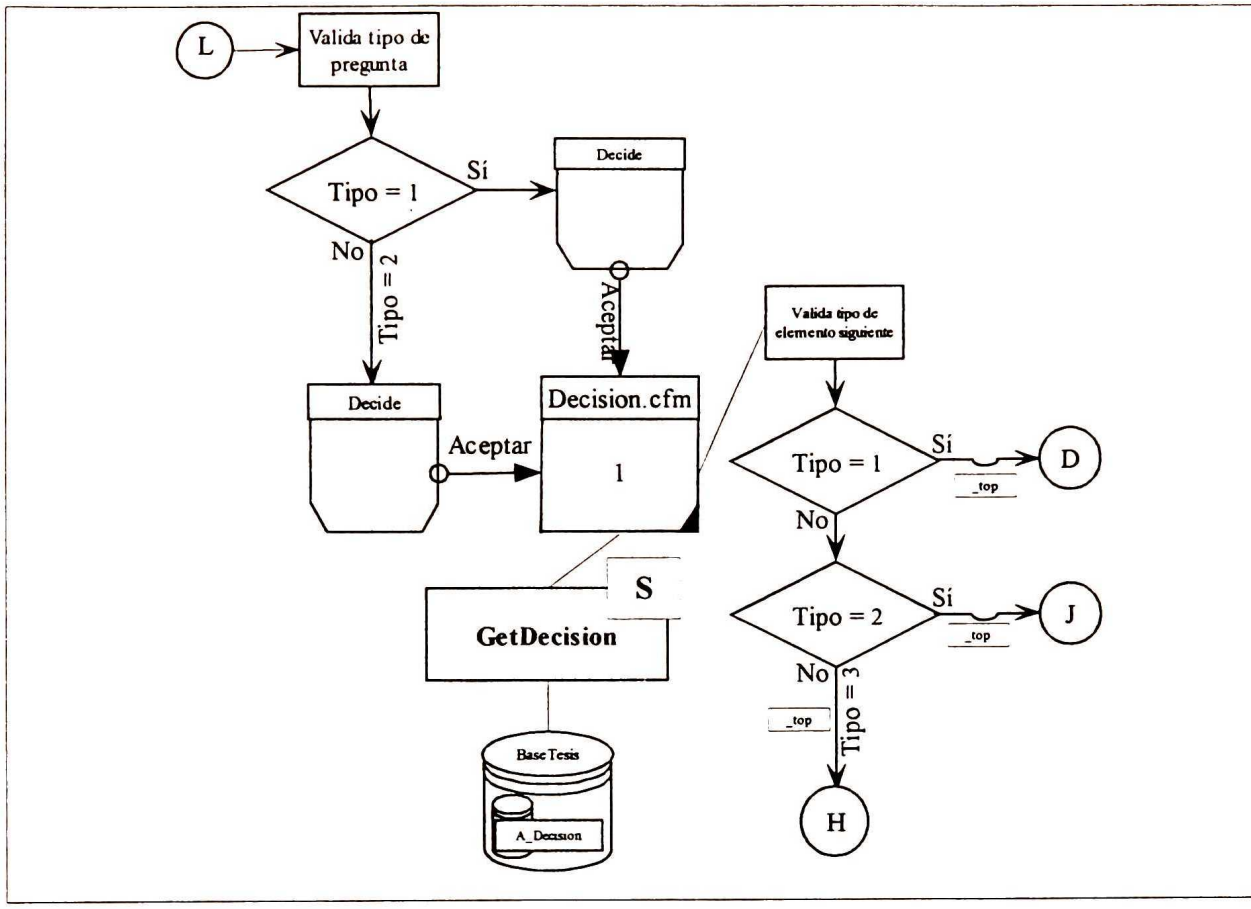


Figura 6-5. Cuarta parte del diagrama generado con la técnica de diagramación propuesta.

El documento **Pres_Preguntas.cfm** selecciona el tipo y clave de la pregunta de la tabla **A_Preguntas** de la base de datos **BaseTesis**, presenta la pregunta y mediante un formulario llamado **Decide** toma la respuesta del usuario y la valida en el archivo **Desicion.cfm** el cual obtiene los datos de la tabla **A_Decision** de la base de datos **BaseTesis** para compararlos con la respuesta del usuario; de acuerdo con la respuesta con secuencia programada en la base de datos se procesa el tipo de elemento que se mostrará a continuación -puede ser un contenido, otra pregunta o una evaluación en forma de árbol-, cargando automáticamente los archivos **Fram_Contenidos.cfm**, **Fram_Preguntas.cfm** y **Fram_Casos.cfm,** respectivamente.

Quinta parte del diagrama:

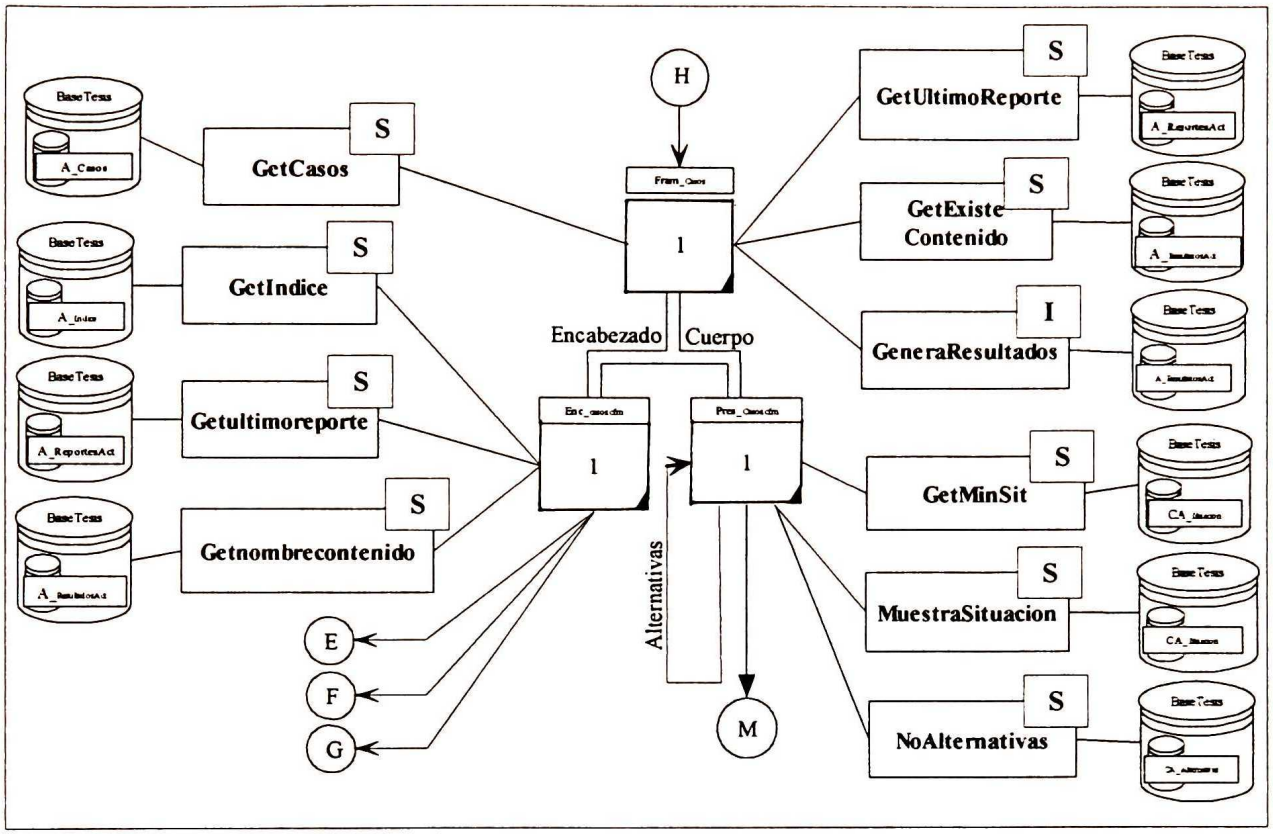


Figura 6-6. Quinta parte del diagrama generado con la técnica de diagramación propuesta.

Cuando el elemento siguiente es una *evaluación en forma de árbol* carga automáticamente el documento de frames **Fram_Casos.cfm** el cual contiene dos frames: **Encabezado** y **Cuerpo** que cargan los documentos **Enc_Casos.cfm** y **Pres_Casos.cfm**, respectivamente.

El documento **Fram_Casos.cfm** selecciona información de las tablas **A_Casos**, **A_ReportesAct** y **A_ResusltadosAct**, inserta información en la tabla **A_ResultadosAct**. Todo esto en la base de datos **BaseTesis.**

El documento **Enc_Casos.cfm** selecciona información de las tablas **A_Indice**, **AreportesAct** y **A_ResultadosAct** de la base de datos **BaseTesis**. Contiene tres formularios: **SalirAct**, **Índice** e **Histórico**, cuyo funcionamiento fue explicado anteriormente.

El documento **Pres_Casos.cfm** selecciona información de las tablas **CA_Situación** y **CA_Alternativas** de la base de datos **BaseTesis**. Presenta la información en un formato de situación y alternativas, donde las alternativas son ligas URL que llaman al documento **Pres_Casos.cfm**. Cuando se han agotado las alternativas en una situación se dice que es terminal y nos llevará a otro elemento (*contenido*, *pregunta* o *evaluación en forma de árbol*) en la siguiente iteración.

Sexta parte del diagrama:

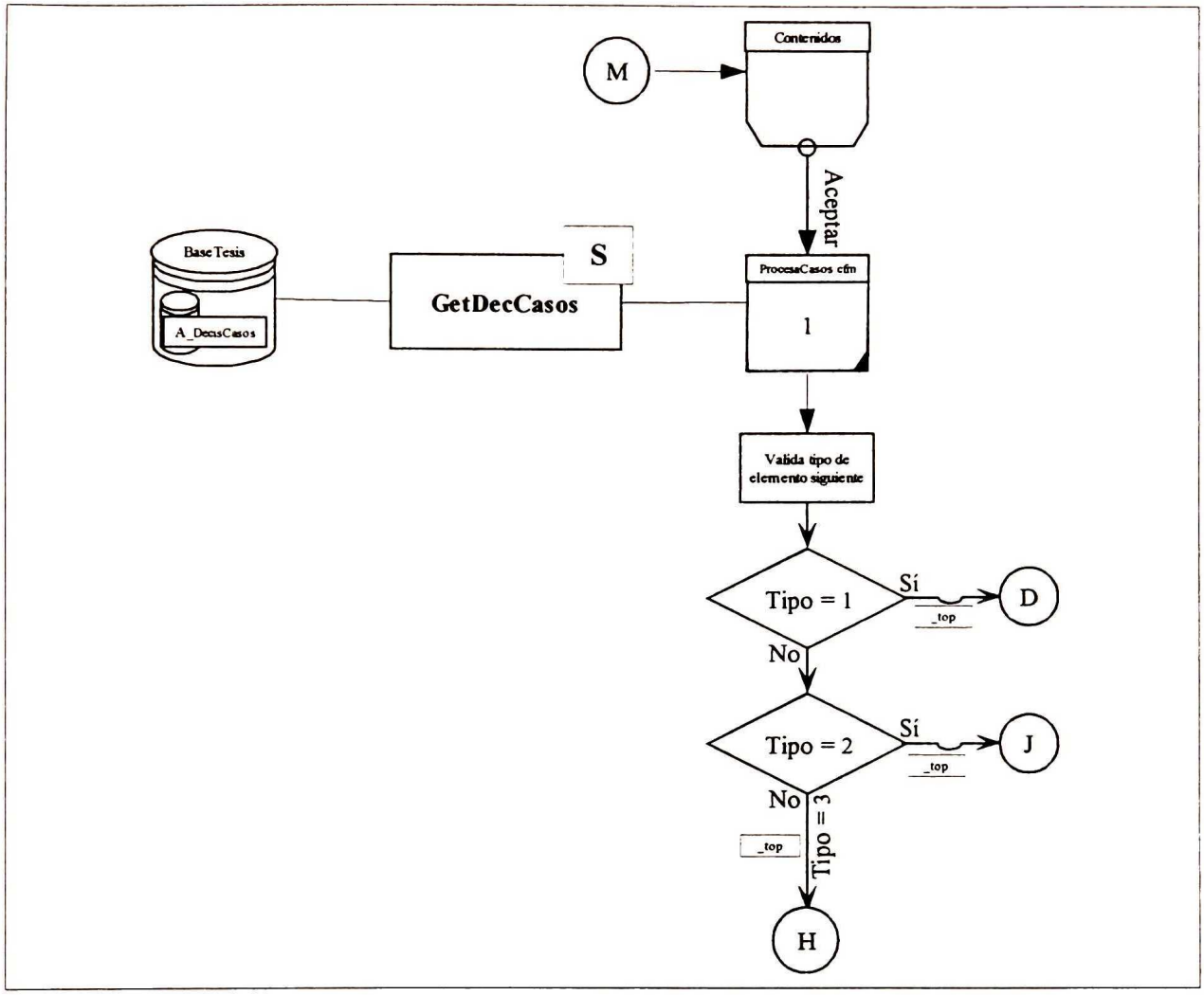


Figura 6-7. Sexta parte del diagrama generado con la técnica de diagramación propuesta.

Podemos observar que el documento **Pres_Casos.cfm** contiene un formulario llamado **Contenidos** el cual ejecuta un archivo **ProcesaCasos.cfm** mismo que obtiene datos de la tabla **A_DesicCasos** de la base de datos **BaseTesis** y ejecuta un proceso para decidir qué elemento continúa en la siguiente iteración del *objeto de aprendizaje actividades*.

6.2. Construcción del sistema transaccional

A continuación se presentan las pantallas que forman parte del sistema transaccional vía tecnología web (*objeto de aprendizaje actividades*).

En esta pantalla se muestra la forma en que un usuario se identificará para poder tener acceso a un *objeto de aprendizaje actividades.*

Los datos que introduce el usuario son validados en la base de datos del sistema.

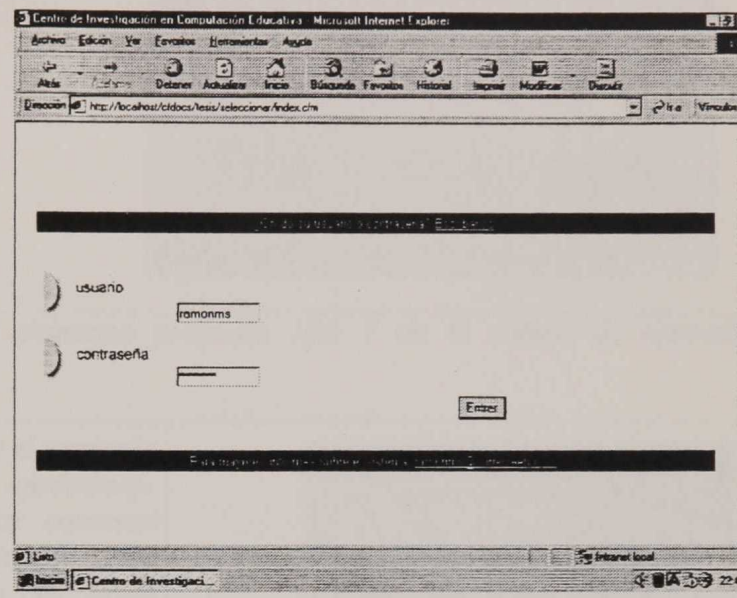


Figura 6-8. Validación de usuario.

En la primera pantalla podemos apreciar que se muestra un contenido. En el análisis y diseño del sistema se especificó que todo *objeto de aprendizaje actividades* inicia con un contenido, a partir de este elemento se puede viajar a cualquiera de los tres elementos que forman parte del *objeto de aprendizaje actividades.*

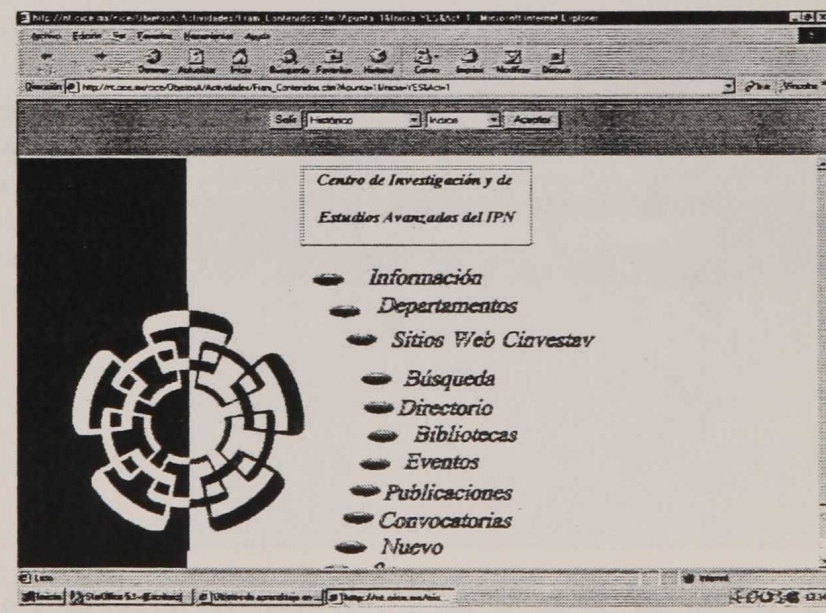


Figura 6-9. Implementación de un contenido en el *objeto de aprendizaje actividades.*

Esta pantalla muestra uno de los dos tipos de preguntas que fueron implementados para el *objeto de aprendizaje actividades*, la secuencia de los elementos es introducida por el instructor y se presenta a partir de las acciones del usuario en el recorrido del *objeto de aprendizaje actividades.*

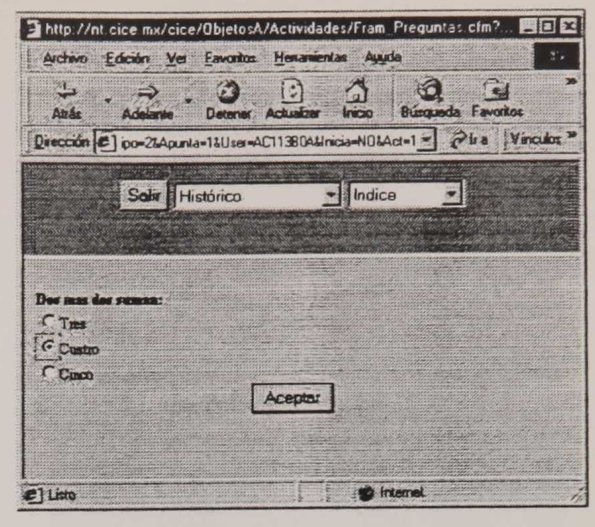


Figura 6-10. Implementación del elemento *pregunta tipo 1* en el *objeto de aprendizaje actividades.*

Para que un usuario pueda pasar al siguiente elemento del *objeto de aprendizaje actividades*, obligatoriamente debe contestar antes la pregunta que se le haya realizado. De no hacerlo, aparecerá un mensaje que indica que la pregunta no se ha respondido.

Figura 6-11. Retroalimentación recibida cuando no se ha respondido una pregunta.

Éste es el segundo tipo de pregunta que fue implementado en el *objeto de aprendizaje actividades* y al igual que el tipo uno, también tiene validación para determinar, con base en la respuesta del usuario, el siguiente elemento que se le presentará.

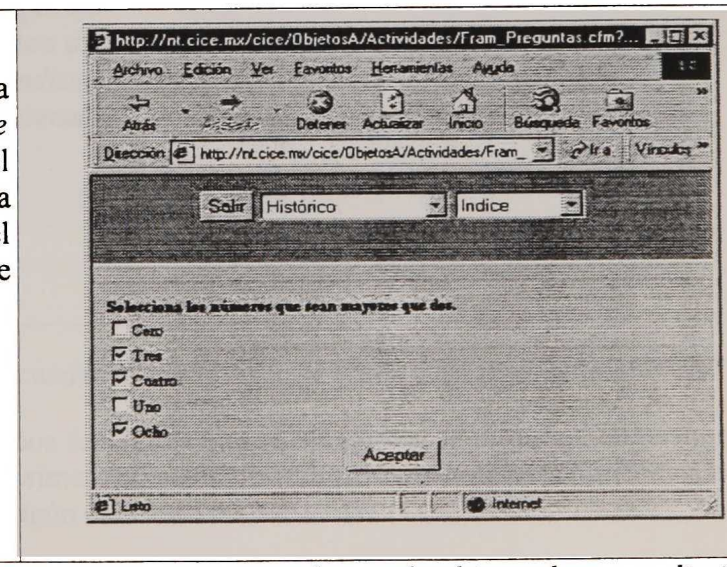


Figura 6-12. Implementación del elemento *pregunta tipo 2* en el *objeto de aprendizaje actividades.*

Éste es el elemento *evaluación en forma de árbol*. Aquí se presentan situaciones y alternativas a los usuarios dependiendo de las respuestas que ellos den. Al final de la *evacuación en forma de árbol*, habrá una retroalimentación.

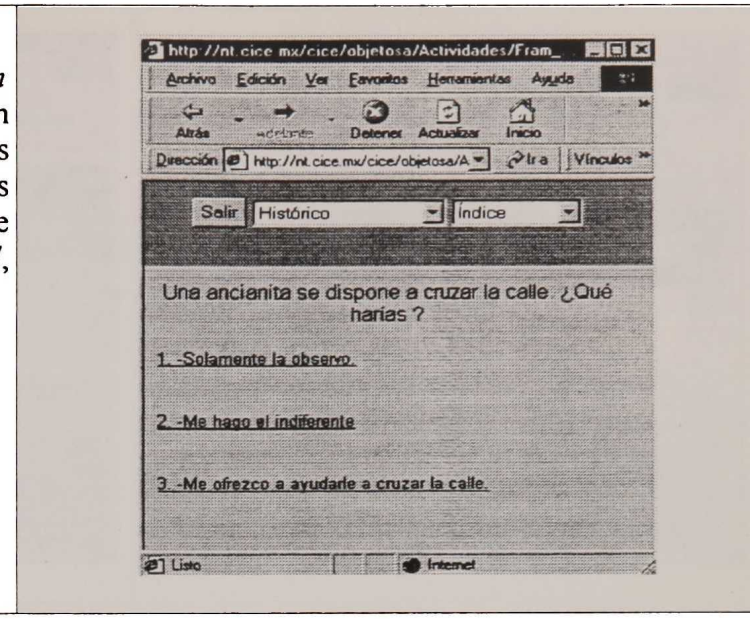


Figura 6-13. Implementación del elemento *evaluación en forma de árbol.*

Esta pantalla muestra la forma en que termina una evaluación en forma de árbol. Podemos ver que se da una retroalimentación al usuario de acuerdo con la ruta que haya realizado. Es importante mencionar que dependiendo de la ruta que haya realizado se tomará la decisión de llevarlo al siguiente elemento que podría ser cualquiera de los tres que forman un *objeto de aprendizaje actividades* como ya se ha mencionado anteriormente.

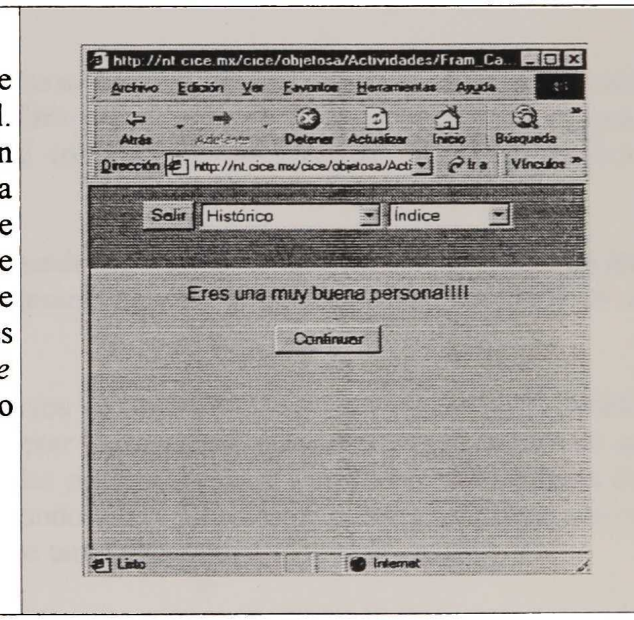


Figura 6-14. Retroalimentación recibida cuando se concluye una *evaluación en forma de árbol.*

Como se ilustra a continuación, dos funciones que se agregaron al *objeto de aprendizaje actividades* son *histórico* e *índice*, en el primero se guarda un historial de los elementos visitados, el segundo sirve para desplazarse hacia algún elemento en particular.

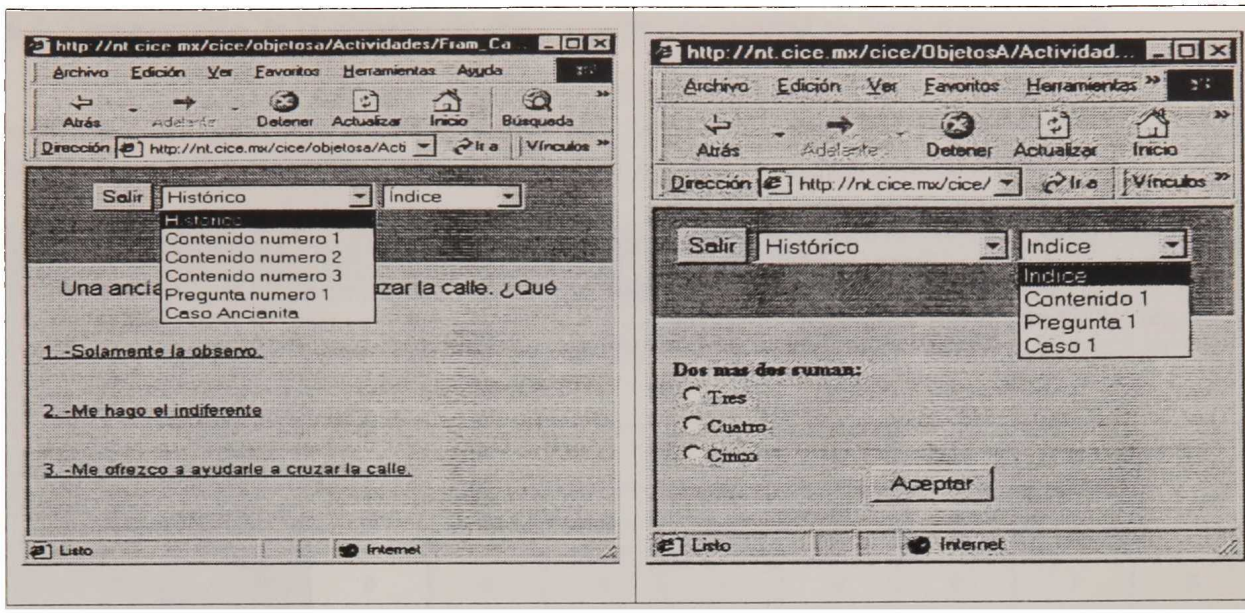


Figura 6-15. Funciones *histórico e índice* implementadas en el *objeto de aprendizaje actividades.*

6.3. Mantenimiento del sistema transaccional

El *software* indudablemente sufrirá cambios después de ser entregado al cliente (una excepción podría ser el *software* empotrado). Esto sucede porque se han encontrado errores, porque el *software* debe ajustarse a cambios del entorno externo o porque el cliente requiere mejoras funcionales o de rendimiento [Arn94].

El mantenimiento del *software* existente puede dar cuenta de más del 60 por ciento de las inversiones efectuadas por una organización de desarrollo, y ese porcentaje sigue ascendiendo a medida que se produce más *software*.

Tan sólo el 20 por ciento de los esfuerzos de mantenimiento se invierte corrigiendo errores. El 80 por ciento restante se dedica a adaptar los sistemas existentes a los cambios de su entorno externo, a efectuar las mejoras solicitadas por los usuarios y a rehacer la ingeniería de las aplicaciones para su posterior utilización. Cuando se considera que el mantenimiento abarca todas estas actividades es fácil ver por qué absorbe tanto esfuerzo.

La técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web* ayudará a encontrar errores y partes del sistema que se quieran modificar, facilitando la reingeniería de estos sistemas.

Cada vez que se modifique el código del sistema se tendrán que modificar también los diagramas de flujo de datos que sirvieron de guía en la construcción del mismo.

6.4. Comparación de la técnicas de diagramación propuesta vs las convencionales

Como se puede apreciar la técnica que en esta tesis se propone está orientada a tecnología *web*, lo cual proporciona grandes ventajas sobre las técnicas convencionales. La técnica propuesta ayuda a crear diagramas de flujo de datos con un mayor nivel de abstracción.

	Gane & Sarson		Dijkstra y sus colegas (Bohm y Jacopini)		Chen		Técnica propuesta	
Representar gráficamente	Alto nivel de abstracción	Bajo nivel de abstracción	Alto nivel de abstracción	Bajo nivel de abstracción	Alto nivel de abstracción	Bajo nivel de abstracción	Alto nivel de abstracción	Bajo nivel de abstracción
Diseño estructurado	0	1	1	0	0	1	1	0
Bases de datos	0	1	0	1	0	1	1	0
Eventos HTML	0	0	0	1	0	0	1	0
Tags HTML	0	0	0	1	0	0	1	0
Componentes (Applets, ActiveX, OCX, etc)	0	0	0	1	0	0	1	0
Operaciones SQL básicas (Insert, Update, Delete, Select)	0	0	0	0	0	0	1	0
Trabajos automáticos (Jobs)	0	0	0	1	0	0	1	0

Figura 6-16. *Comparación de las técnicas propuesta vs convencionales.*

Podemos observar que la técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web* cubre perfectamente el flujo de los principales transformadores de datos que son utilizados en tecnología *web* y sistemas transaccionales.

6.5. Resumen

En este capítulo se puso a prueba la técnica de trabajo propuesta para diagramar el flujo de la información en sistemas transaccionales vía tecnología *web*.

La técnica de diagramación propuesta ofrece grandes ventajas en el diseño arquitectónico de sistemas transaccionales vía tecnología *web*. La notación de la técnica está basada en tecnología *web*, ello hace posible la construcción de diagramas con un alto nivel de abstracción.

CONCLUSIONES

La redes computacionales son un claro ejemplo del avance que ha tenido la tecnología informática. Al principio sólo eran redes locales (LAN), posteriormente metropolitanas (MAN) y más tarde surgieron las redes mundiales (WAN) tales como Internet, cuyo aprovechamiento en aplicaciones empresariales dio como resultado nuevos conceptos: intranets (aplicaciones desarrolladas sobre tecnología Internet que corren a nivel de redes locales, es decir, no están abiertas a la red Internet) y extranets (aplicaciones que aprovechan las características que brinda una intranet y la red global Internet).

El desarrollo de aplicaciones en tecnología *web* es cada vez más utilizado por las empresas, debido a los múltiples beneficios que ofrece: acceso a bases de datos, multimedia al alcance de todos, transacciones, etc.

Las transacciones son los requerimientos o envíos que se le hacen al sistema. Pueden ser tan sencillas como transferir un archivo o tan complejas como la manipulación de bases de datos.

Los sistemas que utilizan las transacciones para su funcionamiento reciben el nombre de sistemas transaccionales; un claro ejemplo de ellos es una aplicación que manipula bases de datos.

En esta tesis se definió un sistema transaccional vía tecnología *web* como una aplicación capaz de interactuar con bases de datos a través de la *web (Internet, intranet y extranet).*

La construcción de un sistema transaccional vía tecnología *web* es el resultado de las fases de desarrollo necesarias en la construcción de cualquier producto de *software*: Definición de requisitos, Análisis, Diseño, Construcción y Mantenimiento.

Este trabajo de tesis se enfocó en la fase del diseño de los sistemas transacionales vía tecnología *web*.

El diseño es el núcleo técnico de la ingeniería del *software*. En él se desarrollan, revisan y documentan refinamientos progresivos de estructuras de datos, arquitectura del programa, interfaces y detalles procedimentales.

Para representar la arquitectura de un sistema transaccional se usan técnicas tales como gráficas de diseño estructurado, diagrama de flujo de datos (DFD), pseudocódigo, etc. Se pueden aplicar muchas de ellas al diseño arquitectónico de sistemas transaccionales vía tecnología *web*. Sin embargo muestran claras carencias en la representación de los principales transformadores de información en los sistemas transaccionales vía tecnología *web*, entre las que destacan la representatividad de *tags*, eventos HTML, operaciones SQL, componentes, scripts, enlace a la base de datos, etc.

Para subsanar tales deficiencias, en esta tesis se propone un técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*, que incorpora iconos para representar a los principales transformadores de información (*tags*, eventos, componentes, *scripts*, enlace a la base de datos, etc.) en tecnología *web*, así como a los elementos más importantes de los sistemas transaccionales (operaciones SQL). Además la técnica propuesta incluye lineamientos para documentar los diagramas generados con la misma y las modificaciones que éstos sufran durante el mantenimiento.

Una característica adicional de la técnica es que realiza un enlace entre los diagramas y los archivos que formarán parte del código, lo que facilita la localización de la correspondencia diagramas-código y simplifica el mantenimiento del sistema.

La notación de la técnica permite encontrar con mayor rapidez los archivos y elementos transformadores de información en los sistemas transaccionales, lo cual agiliza el proceso de reparación o implementación de los sistemas que fueron diagramados con la técnica propuesta.

Se puede concluir que el uso de la técnica propuesta es de gran ayuda en el diseño arquitectónico de sistemas transaccionales vía tecnología *web*, ya que permite la generación de diagramas de flujo de datos con un alto nivel de abstracción, lo que facilita su interpretación, ayudando en la construcción y mantenimiento de los mismos.

RELACIÓN DE FIGURAS

Figura 2-1. Cuerpo de una transacción....9
Figura 2-2. Transacción anidada....9
Figura 2-3. Esquema común de un sistema transaccional vía tecnología *web*....10

Figura 3-1. Notación básica DFD de Gane & Sarson....12
Figura 3-2. Notación básica DFD de Dijkstra, Bohm y Jacopini....13
Figura 3-3. Notación propuesta por Chen y sus colegas....15
Figura 3-4. Principal uso de las técnicas convencionales....17

Figura 4-1. DFD del análisis del *objeto de aprendizaje actividades*....21

Utilizando la técnica de Gane & Sarson
Figura 4-2. DFD del *objeto de aprendizaje actividades*....22
Figura 4-3. DFD del elemento *contenidos*....22
Figura 4-4. DFD del elemento *preguntas*....22
Figura 4-5. DFD del elemento *evaluación en forma de árbol*....23

Utilizando la técnica de Dijkstra, Bohm y Jacopini
Figura 4-6. DFD del *objeto de aprendizaje actividades*....24
Figura 4-7. DFD del elemento *contenidos*....25
Figura 4-8. DFD del elemento *preguntas*....25
Figura 4-9. DFD del elemento *evaluación en forma de árbol*....25

Utilizando la técnica de Chen
Figura 4-10. DFD del *objeto de aprendizaje actividades*....26
Figura 4-11. DFD del elemento *contenidos*....27
Figura 4-12. DFD del elemento *preguntas*....27
Figura 4-13. DFD del elemento *evaluación en forma de árbol*....27

Figura 4-14. Comparación de las técnicas de diagramación convencionales....28

Figura 5-1. Notación de la técnica de diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología *web*....31
Figura 5-2. Implementación de la técnica de diagramación en una librería de *SmartDraw*....35

Figura 6-1. Modelo entidad-relación del sistema transaccional *objeto de aprendizaje actividades*....37
Figura 6-2. Primera parte del diagrama generado con la técnica propuesta....38
Figura 6-3. Segunda parte del diagrama generado con la técnica de diagramación propuesta....39
Figura 6-4. Tercera parte del diagrama generado con la técnica de diagramación propuesta....41
Figura 6-5. Cuarta parte del diagrama generado con la técnica de diagramación propuesta....42

Figura 6-6. Quinta parte del diagrama generado con la técnica de diagramación propuesta.....43
Figura 6-7. Sexta parte del diagrama generado con la técnica de diagramación propuesta.....44
Figura 6-8. Validación de usuario.....45
Figura 6-9. Implementación de un contenido en el *objeto de aprendizaje actividades*.....45
Figura 6-10. Implementación del elemento pregunta *tipo 1* en el objeto de aprendizaje actividades.....46
Figura 6-11. Retroalimentación recibida cuando no se ha respondido una pregunta.....46
Figura 6-12. Implementación del elemento pregunta *tipo 2* en el *objeto de aprendizaje actividades*.....46
Figura 6-13. Implementación del elemento *evaluación en forma de árbol*.....47
Figura 6-14. Retroalimentación recibida cuando se concluye una *evaluación en forma de árbol*.....47
Figura 6-15. Funciones *histórico e índice* implementadas en el *objeto de aprendizaje actividades*.....48
Figura 6-16. Comparación de las técnicas propuesta vs convencionales.....49

DER. Diagrama entidad-relación.

DFD. Diagrama de flujo de datos.

E-mail. Correo que se establece vía electrónica mediante Internet. Cada persona tiene una dirección asignada en su computadora de tal manera que puede enviar y recibir mensajes.

Encriptación. Proceso basado en operaciones lógicas binarias para disfrazar un dato y evitar que sea leído por otra fuente distinta al destino.

Extranet. Es la combinación de una intranet con Internet, con un esquema de seguridad.

Firewall. Sinónimo de dispositivo de *software* o *hardware* encargado de proteger cualquier sistema de la entrada de personas no autorizadas. Regula, según las necesidades, los niveles internos de restricción a la información y autoriza el acceso a cierto tipo de datos.

FTP. Servicio que permite transferir archivos entre sistemas y entre redes remotas con sistemas diversos. De uso común en Internet.

Html (*HyperText Markup Language*). Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores *web*. Admite componentes hipertextuales y multimedia.

Http (*HyperText Transfer Protocol*). Protocolo usado para la transferencia de documentos html.

Internet. Red de redes con base en TCP/IP y acceso público mundial.

Intranet. Red de área amplia con gran infraestructura y acceso privado. Utiliza tecnología *web*.

Protocolo. Conjunto de reglas establecidas para realizar transacciones.

RDSI. Red digital de servicios integrados, es un servicio para transmitir varios tipos de información, texto, imágenes, sonido, etc., mediante la red pública.

Script. Conjunto de caracteres formado por mandatos y secuencias de tecleo, que se utiliza frecuentemente en Internet para automatizar tareas muy habituales, por ejemplo, la conexión a la red (*login*).

SQL. El lenguaje de consulta a la base de datos cliente/servidor más utilizado.

URL (*Uniform Resource Locator*). Sistema unificado de localización de recursos en la red.

Bibliografía

[Arn94] Robert S. Arnold, *Software reengineering.* IEEE Computer Society,1994
[Boh00] Bohdan O. Szuprowiz, *Implementing Enterprise Portals Integration strategies for Intranet, Extranet, and Internet Resources.* Computer Technology Research, 2000.
[Deb99] Deborah Bayles, *Creación de redes entre socios comerciales.* Prentice Hall, 1999.
[Eth00] Ethan Cerami, *Tecnología Push en Internet.* Mc Graw Hill, 2000.
[Fai94]Richard Fairley, Ingeniería de software. Mc Graw Hill, 1994.
[Ian98] Ian Sommerville, Software engineering. Adisson-Wesley,1998.
[Inc93] D. C. Ince, Ingeniería de software. Adisson-Wesley, 1993
[Jen98] Jennifer Stone G., *The 21st Century Intranet.* Forework by Vinton G. CERF, 1998.
[Law91] Shari Lawrence, Software engineering. Mc Millan, 1991
[Mcc00] Steve McConell, *Proyectos Informaticos.* Mc Graw Hill, 2000.
[Mol93] K. H. Möller and D. J. Paulish, Software Metrics. Chapman & Hall, 1993.
[Rog98] Roger S. Pressman, *Ingeniería de software un enfoque práctico.* Mc graw Hill, 1998.

Referencias URL

[hArt] http://www.articlecentral.com
[hCyb] http://www.cyberlink.com.pe
[hGeo] http://www.geocities.com
[hIfw] http://www.infoworld.com
[hInt] http://www.internet.com
[hLea] http://www.learnthenet.com
[hSea] http://search.msn.es
[hSum] http://sunsite.unam.mx
[hSun] http://www.sun.com
[hVli] http://www.vlib.org

Los abajo firmantes, integrantes de jurado para el examen de grado que sustentará el **Sr. Ramón Mondragón Sotelo,** declaramos que hemos revisado la tesis titulada:

"Técnica de Diagramación para el diseño arquitectónico de sistemas transaccionales vía tecnología Web".

Y consideramos que cumple con los requisitos para obtener el Grado de Maestro en Ciencias en la especialidad en Ingeniería Eléctrica opción Computación.

Atentamente,

Dr. Sergio V. Chapa Vergara _____

Dr. Arturo Díaz Pérez _____

Dr. Guillermo B. Morales Luna _____

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL

BIBLIOTECA DE INGENIERIA ELECTRICA
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro antes del vencimiento de préstamo señalado por el último sello.

11 DIC. 2002

DEVOLUCION

AUTOR MONDRAGON SOTELO, R.

TITULO TECNICA DE DIAGRAMACION PARA EL DISEÑO ARQUITECTONICO DE SISTEMAS...

CLASIF. XM 01.1 REGTRO. BI - 15882

NOMBRE DEL LECTOR	FECHA PREST.	FECHA DEVOL.
Sofia Paz	27/06 2001	28/6/01
José A. Conca	11/nov/02	9/12/02