UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

PROGRAMACIÓN ASP Y XML APLICADA A LA CREACIÓN DE UNA PÁGINA WEB DE CONSULTA DE INFORMACIÓN HISTÓRICA

TESINA

PARA OBTENER EL TÍTULO DE:

LIC. EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA.

SERGIO ROMÁN SÁNCHEZ MARTÍNEZ

Asesor: Fís. Mat. Jorge Luis Suárez Madariaga

Fecha: Febrero 2007.

ÍNDICE

PROGRAMACIÓN ASP Y XML APLICADA A LA CREACIÓN DE UNA PÁGINA WEB DE CONSULTA DE INFORMACIÓN HISTÓRICA

Introducción

En el área del desarrollo web, se han desarrollado herramientas de programación que facilitan al programador el desarrollo de páginas Web. Un ejemplo de estas herramientas es Microsoft InterDev, a través de la cual se desarrollan aplicaciones para la Web.

Una de las tecnologías que se emplean en el desarrollo web es el Lenguaje Extensible de Marcas (XML, eXtensible Markup Language), siendo una herramienta para la manipulación e intercambio de datos.

Otra tecnología son las Páginas de Servidor Activas (ASP, Active Server Pages), que son utilizadas para crear aplicaciones web dinámicas.

En el desarrollo de páginas Web existe la necesidad de hacer uso de una base de datos, por ejemplo, para un órgano gubernamental que desee presentar su información en una red. Las bases de datos constituyen parte importante en la estructura de páginas web, una base de datos mal construida provoca que el sitio web no cumpla con los objetivos por los cuales fue creado.

Los manejadores relacionales de bases de datos son programas que proporcionan soporte y herramientas para la administración de bases de datos.

De acuerdo a las necesidades de las empresas o instituciones públicas, el diseño y construcción de una página web tiene variaciones. Es útil para las Instituciones públicas contar con información sobre sus operaciones o llevar series de datos históricas de las mismas y que esta información se encuentre disponible, esto para ayudar al personal normativo o directivo en la toma de decisiones.

Si la organización cuenta con una intranet, cabe la posibilidad de crear páginas web para acceder a la información de una forma rápida y cómoda.

Con base en lo anterior, surge el interés en la elaboración de la presente tesina, la cual tiene por objetivo:
Dar a conocer los conceptos para la construcción de una página web de consulta de información histórica, utilizando el lenguaje XML, páginas ASP, así como el uso de bases de datos relacionales.

La página web desarrollada esta dirigida al personal normativo de la Dirección de Prestaciones Económicas y Sociales, del Instituto Mexicano del Seguro Social. Esta página consiste en un filtro de consulta de información histórica de la Dirección de Prestaciones Económicas y Sociales.

La tesina se divide en 4 partes, a continuación se presenta un resumen de cada una:

I. XML

Se dan a conocer los orígenes de XML, sus características y ventajas. Se define el concepto de documento XML, se hace una introducción al Estándar XML así como una referencia a la gramática Extended Backus-Naur Form (EBNF). Se definen los conceptos de prólogo y ejemplar, así como los conceptos de documento válido y documento bien formado, se da a conocer cuales son los elementos y atributos que componen un documento XML, así como las reglas que siguen para ser procesados exitosamente. Se hace una introducción al concepto de Hojas de Estilo en Cascada (CSS, Cascading Style Sheets), así como del lenguaje de Hojas de Estilo Extensible (XSL, eXtensible Style Sheet Language).

II. ASP

Se exponen las características de las Páginas de Servidor Activas (ASP, Active Server Pages). Se explican los conceptos de script del lado de servidor y script del lado del cliente. Se describen los objetos más usuales utilizados en ASP. Se describen los pasos para crear un sitio web usando Internet Information Server IIS y se presenta el uso de Microsoft InterDev para crear páginas ASP. Se incluye una referencia a los lenguajes HTML y JavaScript para la creación de algunos elementos que se usarán para crear la página Web.

III. Transformación de datos con Microsoft SQL Server

Se hace una introducción al manejador relacional de bases de datos SQL Server. Se hace una referencia de algunas funciones del manejador Microsoft SQL Server. Se da a conocer un procedimiento para realizar una migración de datos de hojas de cálculo de Microsoft Excel, y se hace una descripción para crear vistas de consulta.

IV. Creación de una página Web de consulta de información histórica

Se presenta el diseño y desarrollo de la página web, usando XML, ASP y el manejador relacional de bases de datos Microsoft SQL Server. Se presenta el diseño de la base de datos, los requerimientos de la página web y el desarrollo de la misma.

Durante el desarrollo de este trabajo, he tenido el apoyo de muchas personas, entre las cuales cito las siguientes:

A mis padres.

A mi hermana Cecilia.

A los Sinodales:

- Jorge Luis Suárez Madariaga.
- Andrés Hernández Balderas.
- Alejandro Rubio Perez.
- Gustavo Gudiño Ramírez.
- Javier Rosas Hernández.

Al Instituto Nacional de Ecología.

Al equipo del INE:
Judith Jaramillo Lopez.
Mauricio “Mau” Barrales Garcia.
Alejandro “Alex” Rubio Perez.

Al Licenciado Hiram Jiménez Arcos.

Al Instituto Mexicano del Seguro Social.

I. XML

Lenguaje Extensible de Marcas (eXtensible Markup Language)

I. XML

En esta primera parte se cubren las principales características del lenguaje XML, así como los conceptos empleados en la parte de programación de la página web.

I.1 – Origen de XML

El lenguaje XML (eXtensible Markup Language, Lenguaje Extensible de Marcas) es una versión simplificada de un metalenguaje de marcado descriptivo conocido como SGML (Standard Generalized Markup Language, Lenguaje Estructurado Generalizado de Marcado).

SGML es un metalenguaje utilizado en grandes aplicaciones industriales. SGML proporciona un modo consistente y preciso de aplicar etiquetas para describir las partes que componen un documento, permitiendo el intercambio de documentos entre diferentes plataformas.

El lenguaje SGML es un lenguaje descriptivo de metamarcado, debido a que es posible elaborar otros lenguajes. Los lenguajes descriptivos incluyen información sobre la estructura lógica del texto, indicando párrafos, citas, listas, entre otros elementos.

Sin embargo, el manejo de SGML es difícil y de elevado costo. Las funciones proporcionadas por SGML son incómodas y complicadas para proporcionar eficientemente información en la Web.

Un grupo del CERN (Conseil Européen por la Recherche Nucléaire) guiado por Tim Berners Lee, creó una aplicación de SGML conocida como HTML (Hiper Text Markup Language, Lenguaje de Marcado de Hipertexto), que fue desarrollado en 1991 para describir páginas web.

Una aplicación de SGML se define como el conjunto de elementos de propósito general, utilizado para un determinado tipo de documento, incluyendo reglas que especifican la organización de los elementos, así como otras funcionalidades.

HTML proporciona un conjunto fijo de elementos predefinidos que se emplea para marcar una página web de propósito general. Algunos de estos elementos son los encabezados, párrafos, listas, tablas, imágenes y vínculos.

Sin embargo, HTML sigue siendo inadecuado para definir muchos tipos de documentos. Algunos ejemplos en que HTML no es adecuado son:

- Documentos que no contengan únicamente los componentes típicos. HTML no cuenta con elementos para representar ecuaciones matemáticas o partituras musicales.
- Bases de datos. Se usan páginas HTML para almacenar información estática de una base de datos. Si se desea filtrar, buscar u ordenar con la información de otra forma,

cada elemento tendría que ser etiquetado individualmente. HTML carece de elementos para realizar lo anterior.
- Documentos con estructura jerárquica. HTML no indica claramente la estructura de un texto que maneje información estructurada, ordenada en niveles de detalle.

El World Wide Web Consortium (W3C), organismo encargado de mantener los estándares de la Web, decidió crear un grupo de trabajo llamado Grupo de Trabajo XML. Jon Bosak, uno de los expertos en SGML, participó en este grupo. El proceso para obtener la primer recomendación de XML duró 2 años.

I.2 – Estándar XML

XML es un estándar internacional desarrollado por un Grupo de Trabajo de XML, conocido como el Comité de Revisión Editorial de SGML, formado bajo el auspicio del World Wide Web Consortium (W3C) en 1996. La versión 1.0 de XML tiene su fundamento en la recomendación de XML REC-xml-19980210, que fue terminada en febrero de 1998.

La recomendación de la versión 1.0 de XML se encuentra disponible en la página de Internet: *http://www.w3c.org /TR/1998/REC-xml-19980210.*

La última versión de la recomendación XML 1.0 está disponible en la siguiente dirección: *http://www.w3.org/TR/REC-xml*.

La traducción al español se encuentra en la página de internet:
http://www.sidar.org/recur/desdi/traduc/es/xml/xml1/index.html

No se cubrirá a fondo la Recomendación XML, sólo se abordarán los conceptos principales que constituyen la base de XML.

La recomendación de la versión 1.0 de define al lenguaje XML de la siguiente manera:

“El Lenguaje Extensible de Marcas, abreviado XML, describe una clase de objetos de datos llamados documentos XML y describe parcialmente el comportamiento de los programas de computadora que los procesan. XML es un "perfil de aplicación" o una forma restringida de SGML [ISO 8879]. Por construcción, los documentos XML son documentos SGML conformados”.

I.2.1 Tecnologías XML

XML cuenta con tecnologías que son usadas en cualquier documento XML, permitiendo agregarle mejoras. Algunos ejemplos de estas tecnologías son:

- XSL (eXtensible Style Language, Lenguaje Extensible de Hojas de Estilo). Funciona como un lenguaje avanzado para crear hojas de estilos. Es capaz de transformar, ordenar y filtrar datos XML.
- XML Schemas. Los esquemas son una forma de definir la estructura, el contenido y la semántica que serán aplicadas a un documento XML.
- XLink : Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en

documentos
- XML Linking Language: Está integrado por Xpath, Xlink y Xpointer.
- Xpath Lenguaje de Rutas XML: Es un lenguaje para acceder a partes de un documento XML. Identifica las partes de un documento XML, como sus atributos y elementos.
- Xpointer: Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.
- Xquery proporciona un modo flexible de consulta para extraer datos de los documentos en el Web.

I.2.2 Gramática EBNF

La sintaxis de XML, definida en la Recomendación XML, esta basada en reglas de producción de la gramática EBNF (Extended Backus-Naur Form). La gramática EBNF fue desarrollada por John Backus y Peter Naur en 1960.

El libro de Gutiérrez Rodríguez, Abraham y Raúl Martínez González "**XML A TRAVES DE EJEMPLOS"**, contiene una introducción a esta gramática, al final del capítulo se proporcionan los datos de esta obra.

Preliminares

La gramática EBNF esta formada por un conjunto de reglas, llamadas de producción, y cada regla de producción define un fragmento específico de sintaxis.

En la Recomendación XML, las características y elementos de XML están definidos por reglas de producción, las cuales están identificadas con un número entre las llaves "[" y "]".

La Recomendación XML incluye todas las reglas de producción que rigen este estándar.

Las reglas de producción se componen de tres partes: un símbolo, la cadena *"::="* y una expresión.
Ejemplo:

Símbolo ::= expresión.

La expresión define los valores que tomar el símbolo. La cadena *"::="* equivale a "*se define como*".

A continuación, se presenta un ejemplo sencillo, con la regla de producción que define al símbolo Númeroentero, al cual se le asocia la expresión *("1"* | *"3"* | *"5"* | *"6")*.

Númeroentero ::= "1" | *"3"* | *"5"* | *"6"*

El símbolo Númeroentero toma los valores de 1, o 3, o 5, o 6. La barra vertical es un operador de alternativa o también un "o" lógico.

Cadenas constantes

Las cadenas constantes son fundamentales en la gramática, se expresan mediante comillas simples y comillas dobles. Las palabras clave de XML son cadenas constantes. Las palabras clave son cadenas constantes que el computador interpreta de manera especial.

Un ejemplo de cadenas constantes lo encontramos en la regla de producción [23] de la Recomendación XML, dentro de ella intervienen dos cadenas constantes, las cuales son '<*?xml*' y '*?>*'. Se observa que las cadenas constantes se encuentran delimitadas por comillas simples.

Caracteres

En la Recomendación XML los espacios en blanco XML son definidos con el símbolo *S.*

Un espacio en blanco en la gramática no es lo mismo que un espacio en el documento XML, por ejemplo:
Númeropar::= "2".
Númeroimpar::= "1".
Números::= Númeropar Númeroimpar.

El valor de *Números* es igual a 21, no importando cuantos espacios en blanco estén en la gramática. Al incluir el símbolo S, como se muestra a continuación:

Números::= Númeropar S Númeroimpar.

el valor que toma *Números* es 2 1.

Al añadir el símbolo *S*, se expresa que se incluyen tantos espacios en blanco como se desee.

Operadores de repetición

Dentro de una regla de producción, se incluyen caracteres de repetición, los cuales son:
?: el carácter de cierre de interrogación se emplea para indicar una o ninguna repetición.
+: se usa para indicar una o más repeticiones.
**:* indica cero o más repeticiones.

En este ejemplo el espacio en blanco *S* aparece solo una vez:
Números::= Númeropar S Númeroimpar.

En este otro, el espacio *S* tiene la opción de aparecer una o ninguna vez:
Números::= Númeropar S? Númeroimpar.

En este caso, existe un espacio *S* o más de uno:
Números::= Númeropar S+ Númeroimpar.

Si se desea que las cadenas este separadas por cero o más espacios, se utiliza la siguiente regla:
Números::= Númeropar S Númeroimpar.*

Paréntesis

Los paréntesis se utilizan para agrupar, para formar expresiones a la que normalmente se les aplica un operador.

Por ejemplo, tenemos la siguiente regla de producción:
Númeropar::= "2" | *"4"* | *"6"* | *"8"*
Número::= Númeropar +

En esta regla de producción, indica la formación de cadenas con al menos un número par. Una forma de expresar lo anterior es:
Númeropar::= ("2" | *"4"* | *"6"* | *"8") +*
Número:: = Númeropar.

I.3 – Características de XML

Las características más importantes de XML son:

- XML se ha diseñado para trabajar sobre Web, se utiliza con protocolos y mecanismos como HTTP, MIME y URL.
- Cualquier documento conforme con XML, es conforme con SGML.
- Sencillez en elaborar documentos en XML.
- Las marcas de XML son legibles.
- XML es un estándar internacional.
- XML es extensible, adaptable a las necesidades de cada desarrollador.
- El usuario define cuantas etiquetas necesita y como serán organizadas en el documento.

I.3.1 - Ventajas de XML

A continuación, se menciona las ventajas que ofrece XML:

- Independencia de los datos respecto de las aplicaciones.
- Información sobre la información, crea un contexto para los datos.
- Elementos para describir la estructura de un documento.
- Capacidad para pasar información a las aplicaciones.

HTML continúa siendo el lenguaje de marcado más popular en el mundo, no obstante HTML contiene limitaciones, entre ellas se mencionan las siguientes:

- HTML no esta orientado hacia la estructura y la semántica de un documento, sino hacia la presentación.
- HTML es un lenguaje de marcado, diseñado para mostrar y enlazar documentos en un navegador, mediante un conjunto determinado y fijo de etiquetas.
- Las etiquetas en HTML son escasas, y no permite la creación de nuevas etiquetas, no ofrece información contextual acerca del documento.
- HTML esta basado en ASCII, por lo que al agregar caracteres de otros idiomas resulta complicado.

XML elimina estas limitaciones y proporciona una gran variedad de aplicaciones prácticas, algunas de ellas son:

- Almacenamiento en Bases de Datos. Se usa XML para etiquetar cada uno de los campos de información dentro de cada registro de una base de datos. Al integrar los datos en un documento XML se ordenan, buscan y procesan los datos de muchas formas distintas.
- Intercambio de noticias e información, utilizando estándares web abiertos (XMLNews).
- Creación de partituras musicales con MusicML (Music Markup Language).
- Formato de fórmulas matemáticos y contenido científico en la Web. (MathML, Mathematical Markup Language).
- Estructuración de documentos. La estructura de árbol en los documentos XML es ideal para marcar la estructura de documentos como novelas, poesía y obras de teatro. La marcación de XML posibilita que el software represente el documento en un formato, localice, extraiga y manipule la información del documento de muchas formas.

Desventajas de XML

Una de las desventajas de XML radica en el soporte de XML dentro del navegador web del usuario, es decir, que el navegador pueda visualizar el contenido de una página web en XML.

Además:

- No es recomendable en sitios Web con enormes cantidades de datos, en los cuales la velocidad de recuperación de datos y la seguridad resultan cruciales.
- XML no resulta aconsejable como sustituto de HTML.
- Si las etiquetas de inicio y cierre del elemento XML no coinciden, el documento XML no es visualizado en el navegador.

1.3.2 – Marcado en XML

El marcado en XML se retomará en el subtema I.5.2 “Marcado en XML”.

I.4 – Documentos válidos y bien formados

Existen dos tipos de documentos XML, uno es documento XML válido y el otro es documento XML bien formado.

Documentos bien formados

Los documentos XML bien formados son todos los que cumplen las especificaciones del lenguaje respecto a las reglas sintácticas que rigen a XML, sin estar sujetos a una Definición de Tipo de Documento (DTD, Document Type Definition). Los documentos XML tienen una estructura jerárquica estricta, los documentos bien formados la cumplen.

Esta estructura es similar a una estructura de un árbol. El documento XML tiene un solo elemento raíz, y de este se derivan otros elementos, y estos a su vez tienen otros elementos.

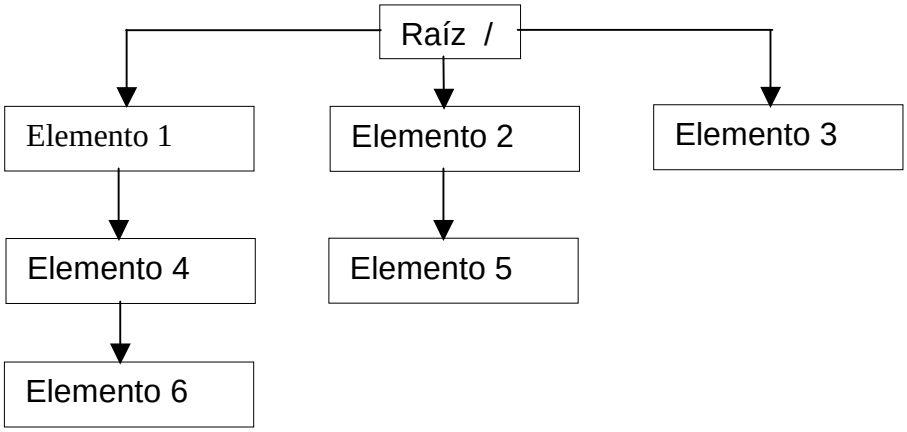

Figura 1.1

Como se aprecia en la Figura 1.1, el elemento raíz se encuentra arriba de la jerarquía, del elemento raíz se derivan los demás elementos o elementos hijo. En este ejemplo, el elemento raíz contiene a otros elementos o nodos hijo, los cuales son los elementos 1, 2 y 3. El elemento 1 y el 2 tienen a su vez, un nodo hijo. (Elemento 4 y 5 respectivamente); y del elemento 4 se deriva otro elemento (Elemento 6).

La regla de producción [1], definida en la recomendación XML, define a un documento XML bien formado como:

*document ::= prolog element Misc**

Las reglas de producción se encuentran en la recomendación XML.

La regla indica que un documento XML bien formado esta integrada por un prólogo (prolog), elementos (element) y un conjunto Misc (que esta formado por un comentario, instrucciones de procesamiento o un espacio en blanco) . Los elementos contenidos en el conjunto Misc se repiten cero o más veces.
La regla de producción de Misc es la [27]:

Misc ::= Comment | PI | S

Donde:
Comment = Comentario:
PI = Instrucción de procesamiento.
S = Espacio en blanco

Documentos válidos

Los documentos XML válidos son aquellos que, además de estar bien formados, siguen una estructura y una semántica determinada por una Declaración del tipo de documento (DTD, Document Type Definition). Los elementos del documento XML y su estructura jerárquica se ajustan al DTD.

El DTD tiene su origen en SGML, el DTD contiene la definición de los objetos que van a formar parte de un documento XML, además de que el DTD es una guía para la construcción del documento XML. La declaración de tipo documento aparece antes que el primer elemento en el documento.

La Declaración del Tipo de Documento se enuncia en las reglas de producción [28] y [29].

I.5 – Partes que integran un documento XML

La estructura de un documento XML esta conformada por marcas y datos de caracteres. Sin embargo, existe una clasificación de mayor nivel. Esta clasificación descompone un documento XML en dos partes, las cuales son el prólogo y el ejemplar.

I.5.1 – El Prólogo y el Ejemplar

El prólogo es opcional, pero se recomienda su uso ya que facilita un proceso fiable y robusto del contenido dentro del ejemplar.

La función del prólogo es la de añadir información sobre el documento; incluye el número de versión de XML, el tipo de codificación de caracteres, si contiene un DTD y si contiene instrucciones de procesamiento. El prólogo aparece al principio del documento XML.

La estructura del prólogo esta definida en la regla de producción [22]:

prolog ::= XMLDecl? Misc (doctypedecl Misc*)?*

Donde:
XMLDecl = Declaración XML.
doctypedecl = Declaración de tipo de documento.

Declaración XML

La regla de producción [23] define la declaración XML:

XMLDecl ::= '<?xml' VersionInfo EncodingDecl? SDDecl? S? '?>'

Esta regla define a los elementos VersiónInfo, EncodingDecl y SDDecl. Estos elementos son los símbolos que especifican las declaraciones de versión, codificación y de documento autónomo respectivamente. El signo de interrogación fuera de una cadena constante, define opcionalidad.

Un ejemplo de declaración XML es la siguiente:

<?xml version="1.0" encoding="ISO-8859-1">

Esta declaración de XML indica que es un documento XML, además proporciona el número de versión de la recomendación XML que usa, en este caso el documento se adapta a la primera recomendación XML del W3C.

La versión de un documento XML esta expresada en la Recomendación XML con la regla de producción [24]:

VersionInfo ::= S 'version' Eq (' VersionNum ' | " VersionNum ")

Dentro de la declaración XML, se incluyen los atributos opcionales, como son *encoding* y *standalone.*

El atributo encoding especifica el conjunto de caracteres que se emplean en el documento XML. La codificación por defecto es el código ASCII de 7 bits (UTF-8). Esta codificación no acepta acentos y otros caracteres. Por otra parte, la codificación de 8 bits ISO-8859, incluye acentos y caracteres como la ñ. También se usan otras codificaciones como UTF-16 y US-ASCII. La codificación se define en la regla de producción [80]:

EncodingDecl ::= S 'encoding' Eq ("" EncName "" | '"' EncName '"')

El atributo standalone le indica al procesador si el documento XML necesita definiciones externas, como un DTD. En el caso de que necesite una definición externa, el valor de standalone es no. Su valor por defecto es yes. Lo anterior se encuentra definido en la regla de producción [32].

SDDecl ::= S 'standalone' Eq ((" ' " ('yes' | 'no') "") | (' " ' ('yes' | 'no') ' " '))

Primero existe un espacio vació *S* seguido de la cadena standalone. Posteriormente se encuentra el símbolo *Eq*, y se tienen los valores "*yes"* o "*no",* los cuales constituyen un conjunto de 2 opciones. La primer opción consiste en usar comillas simples para delimitar el valor "*yes"* o "*no"* de la cadena standalone, y en la segunda opción se usan comillas dobles.

El símbolo *EQ* esta definido en la regla de producción [25].

Eq ::= S? '=' S?

El símbolo *Eq* toma un espacio en blanco o ningún espacio en blanco, seguido del carácter "=" para terminar con un espacio en blanco o ninguno.

Un ejemplo en donde un documento XML es declarada una definición externa es la siguiente:

<?xml version="1.0" standalone='no'?>

Otros elementos del prólogo

Los siguientes elementos son opcionales y se colocan después de la declaración XML:

La declaración de tipo de documento DTD define el tipo y estructura del documento. Al ser incluido en el documento XML, el DTD va a continuación de la declaración de XML. Otra forma de declarar un DTD es a través de un archivo externo.

Las instrucciones de procesamiento son aquellas que le dicen al procesador XML que información pasar a la aplicación. Una instrucción para añadir una hoja de estilo a un

documento XML es una instrucción de procesamiento.

El procesador XML es un módulo de software o librería que lee el documento XML y proporciona acceso al contenido o estructura del mismo a otro módulo de software, llamado aplicación.

La aplicación trabaja en conjunción con el procesador XML y su función es manipular y mostrar el contenido del documento. Esta aplicación es diferente a una aplicación XML.

Las funciones que cumple la declaración de XML son:

- Marca el documento como texto XML.
- Contiene la declaración de versión.
- Informa sobre la codificación empleada.
- Incluye una declaración de documento autónomo.

El ejemplar

Esta es la parte del documento XML que contiene todos los datos y el marcado asociado.
El ejemplar esta constituido por elementos, atributos, comentarios e instrucciones para el procesador, por ejemplo:

<Libros>
    <Medicina>Anatomía Humana</Medicina>

    <Computación>ASP .NET </Computación>

    <Superación>FISH</Superación>
</Libros>

I.5.2 – Marcado en XML

Un documento XML esta compuesto por marcas o etiquetas y datos carácter. El marcado se define como “ciertas señales con un propósito definido que se añade a un texto para ayudar a su procesado”.

Una marca o etiqueta es un texto delimitador que indica la estructura de un documento, por ejemplo etiquetas de inicio y cierre de elementos, etiquetas de elemento vacío, comentarios, declaraciones de tipo de documento e instrucciones de procesamiento. Las etiquetas están delimitadas por los símbolos “<” y “>”. El resto se denomina datos de caracteres, y se denomina como la verdadera información del documento.

El marcado en XML sigue unas sencillas reglas:

- El documento XML tiene un elemento de nivel superior (elemento raíz). Todos los demás elementos están anidados dentro de el.
- Los elementos están adecuadamente anidados. Si un elemento comienza dentro de otro elemento, finaliza dentro del mismo elemento.

- Cada elemento tiene una etiqueta de inicio y una etiqueta de cierre.
- El nombre de una etiqueta de inicio coincide con el nombre de la etiqueta de cierre.
- El nombre de las etiquetas distingue entre mayúsculas y minúsculas. Por ejemplo, el nombre de region es diferente a Region o a REGION.

El nombre que aparece en la etiqueta de inicio y en la etiqueta de cierre, dentro de los caracteres “<” y “>”, se conoce como tipo o identificador genérico del elemento.

Por ejemplo:

<libros> Medicina </libros>

El identificador genérico es “libros”

La diferencia que distingue a las etiquetas de inicio y las etiquetas de cierre es la cadena “</”, la cual precede al identificador genérico del elemento.

En las etiquetas de inicio y de cierre, no van espacios en blanco entre los caracteres “<”, “>”, “</” ó “>”.

La etiqueta de inicio esta definida en la regla de producción [40]:

STag ::= '<' Name (S Attribute) S? '>'*

Mientras que la etiqueta de cierre esta definida por la regla de producción [42]:

ETag ::= '</' Name S? '>'

Ejemplo:

(marca de inicio) *<region>*
(contenido) *<nombre> Norte </nombre>*
(contenido) *<nombre> Sur </nombre>*
(marca de fin) *</region>*

La etiqueta de inicio es *<region>*, y su etiqueta de cierre es *</region>*. Dentro de region, se encuentran otros elementos, los cuales tienen una etiqueta de inicio *<nombre>* y una etiqueta de cierre *</nombre>*.

Si se desea presentar los caracteres dentro del paréntesis (<, >, &, ‘,”,), se usan las referencias a entidades generales predefinidas, las cuales se muestran a continuación:

Entidad	Carácter
&amp	&
&lt	<
&gt	>
&apos	‘
&quot	“

I.5.3 - Elementos

Un elemento es la unidad lógica básica con capacidad para representar la estructura lógica y la semántica de un documento XML. Los elementos de un documento XML contienen la verdadera información del documento XML, y dentro de los cuales se definen otros elementos.

Las funciones principales de los elementos son:

- Especificar un contexto de aplicación para la información.
- Definir y marcar el contenido.
- Proporcionar información sobre la estructura lógica del documento.
- Añadir información sobre el significado del contenido de un elemento en concreto.
- Delimitar las distintas partes de un documento, separándolas y posibilitando su extracción informática.
- Agrupar otros elementos bajo un mismo denominador lógico.

La estructura lógica del documento es la disposición de sus elementos en un documento XML.

Un elemento esta formado por tres partes: una etiqueta de inicio, una etiqueta de cierre y un contenido situado entre ambas etiquetas, es obligatorio que todos los elementos tengan una etiqueta de inicio y una de cierre.

El identificador genérico cumple las siguientes reglas:

- El nombre comienza con una letra o un guión bajo (_), seguido por cero o más letras, dígitos, puntos, guiones o guiones bajos.
- La recomendación XML indica que los elementos que comiencen con el prefijo XML están reservados para estandarización.
- El uso de los dos puntos (:) en un nombre de elemento esta reservado para designar espacios de nombres.
- El nombre de la etiqueta de inicio y el nombre de la etiqueta de cierre son iguales.
- Los caracteres en blanco, las comas, el punto y coma y el signo de admiración, no forman parte de un nombre.

Los elementos están adecuadamente anidados, si un elemento comienza dentro de otro elemento, también finaliza dentro del mismo elemento.
Ejemplo:

Elemento raíz
<libros>
    <computación>
        <Internet> JavaScript </Internet> Elemento anidado
        <servidores> Solaris 9</servidores> Elemento anidado
    </computación>
Fin de elemento raíz
</libros>

El elemento *Internet* y *servidores* comienzan y terminan dentro del elemento *computación*, si la etiqueta de cierre del elemento servidores se encuentre fuera del elemento computación, al momento de visualizar el documento XML se producirá un error.
El elemento raíz agrupa el contenido del ejemplar del documento XML. El elemento raíz es obligatorio, y el documento XML tiene un único elemento raíz.

Ejemplo:
<libros> Elemento raíz.
<computación>
<Internet> JavaScript </Internet
<servidores> Solaris 9</servidores>
</computación>
</libros> Fin de elemento raíz.

El elemento raíz es la etiqueta *libros,* la cual contiene a todos los demás elementos.

El analizador XML recorre los datos de caracteres de un elemento en busca de marcas XML. No se puede insertar los caracteres “<” o “&”, ya que el analizador interpreta a “<” como el inicio de un elemento anidado y a “&” como el comienzo de una entidad o una referencia a carácter. Para insertar estos caracteres dentro de los datos de caracteres, se usan las referencias a caracteres.

En un documento XML se utilizan elementos vacíos, los cuales carecen de contenido.

La etiqueta de elemento vacío está formada por el carácter “<”, seguido de un identificador genérico, para terminar la etiqueta se incluyen espacios en blanco opcionales seguidos de la cadena “/>”. Una etiqueta de inicio y una de fin, sin ninguna información entre ellas, es un elemento vacío.

Un elemento vacío es empleado para indicar a la aplicación XML que realice una acción, o represente un objeto.

A continuación, se presenta un ejemplo de documento XML:

<?xml version="1.0" ?>
Encabezado del documento XML
<bibliografia>
Etiqueta de inicio del elemento raíz
<libro>
Etiqueta de inicio del elemento XML
<autor>Comisión Europea</autor>
Elemento XML anidado
<titulo>Análisis monetario en países de la Unión Europea</titulo>
Elemento XML anidado
<fecha>Enero 2000</fecha>
Elemento XML anidado
</libro>
Etiqueta de cierre del elemento
</bibliografia>

Etiqueta de cierre del elemento raíz

I.5.4 – Atributos y comentarios

Los atributos son objetos que se utilizan para definir diversas propiedades del elemento, que no necesariamente tienen que ser mostradas como una categoría o una indicación de presentación.

La regla de producción [41] define al atributo como:

Attribute ::= Name Eq AttValue

El atributo consta de dos partes. La primera es la propiedad del elemento y la segunda es el valor de esa propiedad. El formato para los atributos es el siguiente:

Propiedad del elemento = "valor"

La adición de atributos constituye una forma alternativa de incluir información en un elemento. Sin embargo, los atributos no contienen subelementos ni subatributos, no se organizan en ninguna jerarquía y no reflejan una estructura lógica, por lo que tienen una capacidad de representación reducida en comparación con los elementos.

Los atributos, al igual que los elementos, cumplen con las siguientes reglas:

- Los nombres de los atributos cumplen las mismas restricciones que siguen los nombres de los elementos.
- Los atributos solo se usan en etiquetas de inicio y de elemento vacío.
- Un determinado nombre de atributo solo aparece una vez en la misma etiqueta de inicio o etiqueta de elemento vacío.
- Para especificar el valor de un atributo, este se escribe entre comillas simples o dobles.
- El caracter "<" no se incluye en un atributo, por que podría interpretarse como el comienzo de una etiqueta.
- Dentro de la etiqueta de inicio de un elemento no aparece dos referencias al mismo atributo.
- La asignación del valor del atributo se realiza a través de un caracter "=". A ambos lados del caracter igual aparecen tantos espacios en blanco como se desee o ninguno.
- Si un elemento tiene varios atributos estos se separan entre si, como mínimo, por un espacio en blanco. Entre los posibles atributos y el caracter ">" se colocan espacios en blanco opcionales.

El siguiente ejemplo presenta un documento XML que contiene atributos:

<?xml version="1.0" ?>

<informe-altas>
<empleado codigo="emp_5643A">Ricardo Fuentes</empleado>
El atributo es *codigo="emp_5643A"*

<empleado codigo="emp_5658F">Juan Valverde</empleado>
El atributo es *codigo="emp_5658F"*
</informe-altas>

En este documento, la etiqueta empleado lleva el atributo “codigo”, que proporciona información sobre un código asignado a cada empleado. El atributo se encuentra en la etiqueta de inicio de cada elemento, y su valor va entre comillas.

Comentarios

Los comentarios están definidos en la regla de producción [15] de la Recomendación XML.

Comment ::= '<!--' ((Char - '-') | ('-' (Char - '-'))) '-->'*

Los comentarios son importantes, por que aportan información útil para el manejo del documento XML.

Los comentarios no se consideran datos caracter, y no son tomados en cuenta al procesar un documento XML.

Para insertar un comentario se utiliza la cadena “<*!*--“ para abrirlo, y la cadena “-->” para cerrarlo.

I.7 – Hojas de Estilo

XML no predefine propiedades de visualización para elementos específicos, por lo tanto se necesita un método que permita modificar la forma en que se visualizan los documentos. Las hojas de estilo proporcionan dicho método.

Las hojas de estilo permiten especificar atributos como colores, márgenes, alineación de elementos, tipos y tamaños de letras, entre otros. Además, se emplean hojas de estilo como patrones o páginas maestras de forma que múltiples páginas tengan el mismo aspecto.

Una hoja de estilo es un archivo que contiene instrucciones para dar formato a los elementos de una página web en HTML.

XML maneja dos tipos de hojas de estilo: las Hojas de Estilo en Cascada (CSS, Cascading Style Sheets) y el Lenguaje de Hojas de Estilo Extensible (XSL, eXtensible Style Sheet Language) . Las hojas de estilo CSS han sido utilizadas con HTML, mientras que las hojas de estilo XSL es un nuevo estándar, que esta enfocado a transformar documentos XML complejos a otros más simples.

I.7.1 – Hojas de estilo CSS

Las hojas de estilo en cascada (CSS, Cascading Style Sheets) es un estándar desarrollado

por el World Wide Web Consortium, para hacer que las páginas web tuvieran mayores capacidades a la hora de ser visualizadas, sin necesidad de añadir al HTML nuevas etiquetas de formato.

El W3C definió a estos documentos como “Hojas de Estilo en Cascada”, para manejar la apariencia de las páginas por medio de múltiples estilos, y el navegador sigue las reglas en cascada para determinar la prioridad y resolver conflictos. Las hojas CSS funcionan de manera similar sobre un documento XML que con HTML.

Las principales ventajas de las hojas CSS son:

- posibilidades adicionales para el formato / presentación.
- un mayor control sobre el documento.
- mayor facilidad en la personalización de documentos.
- Las definiciones del formato del documento se colocan en archivos separados y se aplican a una serie de documentos.

Las desventajas de las hojas CSS son:

- Soporte irregular por parte de los navegadores.
- Ciertas propiedades que funcionan en un navegador no funcionan en otros navegadores.
- Soporte irregular en versiones diferentes de un mismo navegador.
- Para algunos navegadores resulta inaccesible leer algunas propiedades, por ejemplo las que afectan la posición o visibilidad de los elementos.

Las funcionalidades de las hojas de estilo CSS permiten:

- Modificar el formato de cada una de las páginas web que componen un sitio sin necesidad de modificar individualmente cada una de las etiquetas que las constituyen.
- Reduce el desorden de etiquetas dentro de un documento.
- Permite el uso de variaciones de diseño a través de sus clases.

Elementos y propiedades

Una hoja de estilo CSS contiene un conjunto de reglas de estilo que se aplican a los elementos que componen un documento XML o HTML que se desee visualizar.
Estas reglas de estilo consisten en dos elementos principales:

- El selector. Es elemento al cual se va a aplicar el estilo.
- La declaración: Describe las propiedades que describen el estilo.

El selector más simple dentro de un documento XML es cualquiera de las etiquetas definidas por el usuario, en el caso de HTML; son las etiquetas propias de lenguaje. Este selector estará seguido por una lista de propiedades que se aplicaran a dicha etiqueta, contenidas dentro de las llaves "{" y "}".

Cada propiedad se identifica por un par del tipo:

Nombre_propiedad: valor.

Las diferentes propiedades a aplicar a una misma etiqueta se separan por punto y coma.

Se usan unidades de medida para establecer los valores de las propiedades, algunas de ellas son:

Unidad	Descripción
Pt	Punto
pc	Picas
mm	Milímetros
cm	Centímetros
In	Pulgadas
Px	Píxeles
Em	La altura de la fuente actual del elemento
Ex	La altura de una x minúscula en la fuente actual del elemento

En la obra de Gunter Born titulada **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML,** se encuentran estas unidades de medida. Al final de este capitulo se encuentran la referencia completa.

A continuación se dan a conocer las principales propiedades para elaborar una hoja de estilo CSS. En el libro de Gunter Born, **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML,** se encuentran descritas estas y otras propiedades.

Font-family: Indica al navegador el tipo de letra con el que se quiere que sea mostrado un elemento concreto. Esta regla utiliza principalmente dos tipos de valores: el nombre de un tipo de letra en concreto (Arial, Times New Roman), o bien el nombre de una de las familias de tipos de letras como sans-serif. Esta regla permite incluir varios valores separados por una coma, de forma que si el navegador no reconoce un tipo o una familia de letras, pase al siguiente valor definido. Por ejemplo:

Body (Elemento de HTML)
{ font-family: arial,helvetica,serif; }

En el libro de Gunter Born, **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML,** se encuentra un listado de las fuentes más utilizadas en las hojas de estilo en cascada.

Font-size: Fija el tamaño de presentación de las fuentes, y permite indicar este tamaño en varios sistemas de medida, entre ellos pixeles (px), porcentaje y puntos por pantalla (pt).

Ejemplo:

font-size: 150%

Font-weight: Indica el peso de la letra, contiene uno de los siguientes valores:
normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |900
donde normal = 400, Bold (negrita) = 700. Ejemplo:

h1
{ font-weight: bold; }

Font-style: Aborda el estilo del tipo de letra. Esta propiedad toma los valores de "normal", "italic", o "oblique", siendo "normal" el valor predeterminado. Ejemplo:

H1
{ font-style: normal; }

Font: Este nombre representa una forma abreviada para las propiedades "font-style", "font-variant", "fontweight", "font-size", "line-height" y "font -family". Se usa la definición de font en lugar de las propiedades ya mencionadas. Ejemplo:

Body
{ font: 12pt sans-serif normal }

Background: Con este nombre se abrevian las propiedades "background-color", "background-image", "background-repeat", "background-attachment" y "background-position". Ejemplo:

Body
{ background: #ffffff; }

Margin-left: Permite definir el margen izquierdo, con el que se formateará el elemento sobre el que se aplique dicho elemento CSS. Los valores incluyen números, porcentajes o el valor auto (siendo 0 el valor estándar).
Ejemplo:

Body
{ margin-left:10px; }

Margin-top: Esta propiedad define la distancia al lado superior en forma de valor ó porcentaje. Ejemplo:

form
{ margin-top: 5px }

Margin-bottom: Establece el margen inferior a través de valores numéricos, porcentajes o valores auto. Ejemplo:

form
{ margin-bottom: 5em; }

Line-height: Especifica la altura de la fila como separación entre dos lineas base de dos filas. El valor predeterminado es normal, es posible usar porcentajes. Ejemplo:

P
{ line-height: 120%; }

Vertical-align: Propiedad que define la alineación vertical. Los valores que toma son "baseline", "sub", "super", "top", "texttop", "middle", "bottom" y datos porcentuales. Ejemplo:

button
{ vertical-align: middle; }

Text-align: Alineación horizontal de elementos de texto. Los valores que toma son: left, right, center o justify. Ejemplo:

Body
{ text-align: center; }

Width: Ancho de la zona del elemento. Los posibles valores son: numérico, porcentaje o auto. Ejemplo:

image
{ width:80px }

Padding-bottom: Define la separación respecto al lado inferior del marco. Toma valores de porcentaje o numérico. Su valor predeterminado es cero. Ejemplo:

Body
{ padding-bottom: 10px; }

Padding-top: Describe la separación al lado superior del cuadro, por ejemplo, la distancia entre un texto y su marco. Se usan valores numéricos o porcentajes. Ejemplo:

Body
{ padding-to: 7px. }

I.7.2 – Hojas de estilo XSL

El World Wide Consortium esta desarrollando un lenguaje de estilo, denominado XSL (eXtensible Style Sheet Language, Lenguaje de hojas de estilo extensible).

Una hoja de estilo XSL es más potente que una hoja de estilo CSS; por que al usar las hojas de estilo en cascada CSS, el procesador toma el documento XML y lo decora con los formatos definidos en las hojas CSS; mientras que al usar las hojas XSL el procesador toma el documento XML, lo examina y en función de las instrucciones contenidas dentro de la hoja XSL, lo transforma en un nuevo documento.

Las ventajas que ofrecen las hojas de estilo XSL son:

- proporciona el control completo sobre la salida.
- permite seleccionar los datos XML que se quieren visualizar.
- presentar estos datos en cualquier orden o disposición y añadir.
- modificar información con total libertad.
- XSL proporciona acceso a todos los componentes XML, como son los elementos, atributos, comentarios e instrucciones de procesamiento.
- permite ordenar y filtrar los datos XML.

Las desventajas de las hojas de estilo XSL son:

- Su utilización es más compleja.
- Consume cierta memoria y capacidad de proceso, pues se construye un árbol con el contenido del documento.

Las hojas de estilo tienen requerimientos más estrictos para visualizar documentos XML. Cuando se utilizan hojas de estilo para visualización a través de un navegador se recomienda tener en cuenta las siguientes limitaciones:

- La hoja de estilo tiene como salida una hoja HTML.
- La hoja de estilo se aplica al elemento raíz del documento fuente.
- La hoja de estilo XSL comparte su esquema URL y el nombre del servidor con el documento XML fuente. La descarga de hojas de estilo XSL sigue las mismas políticas de seguridad que la descarga de entidades externas, por tanto, leer una hoja de estilo de un dominio diferente produce una violación de acceso.

Dentro del concepto de XSL se esconden la definición y uso de tres lenguajes relacionados con el estándar, estos son:

- El lenguaje XSL, es decir, los objetos de formato descritos con vocabulario XML.
- El lenguaje XSLT o XSL Transforming, que es un lenguaje de marcas que describe la forma en que XSL permite la transformación del documento XML original. XSLT toma un documento XML (denominado fuente) y lo transforma en una versión diferente del documento. (denominado árbol resultado), basándose en los filtros y patrones contenidos dentro de la hoja XSL.
- El lenguaje Xpath o XML Path Language, que es un lenguaje utilizado para direccionar fragmentos de documentos XML, este lenguaje es utilizado por XSLT (también por Xpointer y Xlink) para describir expresiones y caminos locales que nos permite crear transformaciones XSL avanzadas.

Plantillas y elementos.

Dentro de la estructura de un documento XML, se manejan plantillas, elementos y sus atributos.

El primer elemento que aparece en el documento XSL es *<xsl:stylesheet>:*

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

Aquí se declaran las plantillas y demás elementos.

</xsl:stylesheet>

El elemento *</xsl:stylesheet>* es el elemento raíz e identifica al documento como hoja de estilo XSL. Este elemento es uno de los elementos XSL de propósito especial utilizados en las hojas de estilo.

Todos los elementos XSL pertenecen al espacio de nombres xsl, por los que es necesario poner el nombre xsl antes de cada nombre de elemento. Este espacio de nombres se define dentro de la etiqueta de inicio como:

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

Esta definición permite utilizar el espacio de nombres “xsl” dentro de cualquier elemento de la hoja de estilo.

El atributo *version* indica el número de versión de XSL utilizada.

Una plantilla especifica las transformaciones que serán aplicadas al documento XML. El elemento documento *xsl:stylesheet* de una hoja de estilo XSL contiene uno o mas elementos de plantillas XSL.

La estructura básica de las plantillas XSL consiste en un elemento de inicio, un cuerpo que compone la plantilla y que se utilizará en caso de que se produzca la selección sobre un nodo del documento XML y un elemento de cierre.

La estructura general es la siguiente:

<inicio de plantilla>

Cuerpo de la plantilla

</cierre de plantilla>

El formateo de las etiquetas XML usando XSL funciona por selección de elementos dentro del documento XML para escoger la plantilla apropiada. La plantilla escogida especifica el formato de salida a aplicar en ese elemento.

Dentro de una hoja XSL, existe una plantilla que se aplica al elemento raíz del documento, así como reglas específicas para los diferentes etiquetas y reglas por defecto que se aplican a todas aquellas etiquetas que no son gobernadas por otras reglas.

Una plantilla contiene dos tipos de elementos:

- Elementos XML que representan código HTML. Cada uno de estos elementos siguen las mismas reglas de construcción de un documento XML.
- Elementos XSL: El navegador distingue entre el código HTML con el de XSL, por que las instrucciones XSL va precedidas por la referencia “xsl”.

Además de las plantillas, un documento XSL cuenta con patrones y acciones:

- Un patrón es la parte de una regla XSL que especifica la etiqueta XML que será formateada.
- Una acción es la parte de una regla XSL que especifica como el patrón es formateado.

El atributo match es utilizado para especificar a cual elemento se aplica la plantilla (el atributo match es análogo al selector de una regla CSS). El valor del atributo match se denomina patrón.
Toda hoja de estilo XSL contiene exactamente una plantilla cuyo atributo match tenga el valor de "/".

Es posible incluir una o más plantillas adicionales con instrucciones para visualizar subelementos específicos de la estructura del elementos del documento XML, cada una de estas plantillas adicionales contiene un patrón que defina la subrama concreta a la que se refiere.

Los patrones XSL son una sintaxis simple y concisa para identificar nodos en un documento XML, basándose el nombre de estos, su contenido y su contexto. Los nodos son las etiquetas que se encuentran en el documento XML. Cada elemento XML es comparado con los patrones definidos en el documento XSL, y cuando coinciden, el elemento XML es formateado de acuerdo a las reglas establecidas en la plantilla.

La estructura básica de un patrón XSL es jerárquica. Es posible restringir el número de elementos con los que coincida el patrón, incluyendo un filtro rodeado por corchetes angulares "[" y "]".
El patrón raíz "/" no representa el elemento raíz del documento XML, sino que representa al documento completo.

Elementos XSL

A continuación se da la descripción de los elementos XSL usados en el presente trabajo. El libro **THE XSL COMPANION** del autor Bradley Neil, presenta un estudio más a fondo de las hojas de estilo XSL. La referencia completa se proporciona al final de este capítulo.

Elemento <xsl:template>

Atributos:
name = nombre cualificado
match = contiene un patrón XSL
mode = permite que el elemento sea procesado múltiples veces produciendo resultados diferentes. Solo aparece si también lo hace el atributo match.

Funcionamiento: Comprueba los nodos del documento XML que se ajusta al patrón dado por su atributo match. Una vez que se encuentre un nodo o elemento del documento que encaje con el patrón usado, aplica la plantilla definida para ese patrón.

Ejemplo:
<xsl:template match="patrón">
Conjunto de instrucciones a realizar
</xsl:template>

Elemento <xsl:apply-templates>

Atributos:
select: contiene un patrón XSL.
mode: permite que el elemento sea procesado múltiples veces produciendo resultados diferentes. Solo aparece si también lo hace el atributo match. Los elementos *<xsl:apply-templates>* que contiene este atributo solo son lanzados desde elementos *<xsl:template>* cuyo atributo mode coincide.

Funcionamiento: Selecciona un conjunto de nodos utilizando la consulta definida en su atributo select. De no existir dicho atributo, se seleccionan todos los hijos del nodo actual.

Ejemplo:
<xsl:apply-templates select="nodo">
</xsl: apply-templates>

ó

<xsl:apply-templates select="nodo"/>

Elemento <xsl:value-of>

Atributos:
select: contiene un patrón XSL. El valor por defecto es ".", el cual hace referencia al nodo actual.

Funcionamiento. Da la posibilidad de insertar los contenidos de los mismos. Devuelve la primera instancia resultado de la consulta sobre el contexto actual que encaje con el valor de su atributo select.

Ejemplo:
<xsl:value-of select="algun_valor" />

Elemento <xsl:for-each >

Atributos:
select: Contiene un patrón XSL. El valor por defecto seleccionada todos los hijos del nodo actual.

Funcionamiento:
Da la posibilidad de ser más específicos en los patrones de selección, permitiendo que se trabaje con todas las instancias (nodos) resultado de la consulta contenida en su atributo select. Establece el contexto para una iteración, sobre los nodos resultado de una consulta.

Ejemplo:
<xsl:for-each select="patron_de_selección">
Conjunto de instrucciones a ejecutar
</xsl:for-each>

La siguiente parte de la tesina se presenta las Páginas de Servidor Activas ASP, las cuales se emplearon para crear el formulario que acompaña a la página web.

REFERENCIAS DE LA PARTE I

LIBROS CONSULTADOS

1.- Gutiérrez Rodríguez, Abraham. Raúl Martínez González. **XML A TRAVES DE EJEMPLOS**. Alfaomega Ra-Ma. Colombia. 2001

2. - Bradley Neil. **THE XSL COMPANION**. Addison-Wesley. Great Britain. 2000.

3.- Gunter, Born. **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML.** Marcombo. 2001

4.- Vazquez Rodríguez, Adolfo **NAVEGAR EN INTERNET XML**. Alfaomega Ra-Ma. México 2002.

5.- Young, Michael J. **APRENDA XML YA**. Traducción de Vuelapluma, S.L. McGraw-Hill. España. 2000

PÁGINAS DE INTERNET CONSULTADAS.

Recomendación XML (en inglés)

a) http://www.w3c.org
Sitio Oficial del World Wide Web Consortium.

b) http://www.w3c.org /TR/1998/REC-xml-19980210.
Versión 1.0.

c) http://www.w3.org/TR/REC-xml.
Última versión.

Recomendacion XML (en español)

d) http://www.sidar.org/recur/desdi/traduc/es/xml/xml1/index.html

Última actualización: 30/06/2006
Traducción: Carlos Benavides y Emmanuelle Gutiérrez.
Editores: Carlos Benavides, Emmanuelle Gutiérrez y Restrepo
Colaborador: Charles McCathieNevile

Hojas de estilo en cascada CSS

e) http://www.sidar.org/recur/desdi/mcss/manual/intro.php

Última actualización: 30/06/2006
Traducción: Carlos Benavides y Emmanuelle Gutiérrez.
Editores: Carlos Benavides, Emmanuelle Gutiérrez y Restrepo
Colaborador: Charles McCathieNevile

XML

f) http://manuales.dgsca.unam.mx/xml/Desventajas.htm

Año: 2006.
Sitio WEB desarrollado e implementado por la Dirección General de Servicios de Cómputo Académico
Director General: Dr. Alejandro Pisanty Baruch
Dirección de Sistemas: Mtro. Juan Voutssas Márquez
Subdirección de Comunicación: Lic. José Antonio Sánchez Yllañez
Subdirección de Servicios Web: L.I. Luz Ma. Ramírez Romero
Diseño Gráfico, adaptación HTML y JavaScript: L. D.G. Raúl Ramírez Sánchez
Jefa de Departamento: L.A. Nancy Escorcia Martínez

g) http://www.adobe.com/es/devnet/dreamweaver/articles/xml_overview_03.html

Fecha de creación: 8 August 2005.
Adobe Systems Incorporated

h) http://www.dcc.uchile.cl/~rbaeza/inf/xml.html

Rediseño del sitio: Newtenberg Ltd.
Última actualización: 13/7/2006 9:33
Autor: Ricardo Baeza-Yates

Gramática EBNF

i) http://www.garshol.priv.no/download/text/bnf.html

j) http://es.wikipedia.org/wiki/Backus-Naur_form

II. ASP

Páginas de Servidor Activas (Active Server Pages)

II. ASP

II.1 - Definición de ASP

Las Páginas de Servidor Activas (ASP, Active Server Pages) es una tecnología desarrollada originalmente por Microsoft, que son usadas para crear aplicaciones dinámicas e interactivas para la Web. Son páginas web que son preprocesadas por el servidor antes de que se envíen al explorador web del cliente.

Una página ASP contiene scripts del lado del cliente y scripts del lado del servidor, estos últimos son procesados por el servidor Web antes de ser mandados al navegador del usuario. Su empleo se enfoca principalmente a la creación de aplicaciones interactivas destinadas a sitios web.

La tecnología ASP apareció por primera vez (versión 1.0) con el servidor Internet Information Server 3.0 de Microsoft en Diciembre de 1996. La versión 4.0 de IIS incluye la versión 2.0 de ASP, y la versión 5.0 de IIS, distribuida con Windows 2000 incluye ASP 3.0.
La versión 6.0 de Internet Information Server (IIS) se ejecuta en Microsoft Windows Server 2003.

En la página web *http://support.microsoft.com/kb/325889/es* se encuentra una guía para configurar el IIS, además proporciona los pasos para migrar de la versión 5.0 y 5.1 a la versión 6.0.

En la página web *http://support.microsoft.com/kb/224609/es* se encuentra un listado en el que aparecen las versiones de IIS.

En un principio, Microsoft daba soporte para ejecutar páginas ASP, mediante sus servidores Personal Web Server e Internet Information Server. Con el paso del tiempo, compañías como ChiliSoft extendieron el soporte de ASP a otros servidores como Apache, los servidores de Netscape y el servidor web de O'Reilly. Otras tecnologías que compiten con ASP son: ColdFusion de Allaire, JavaServer Pages de Sun Microsystems y PHP esta última de libre distribución bajo Open System.

II.1.1 - Características y ventajas de ASP

Algunas de las características de las páginas ASP son:

- Es posible combinar código HTML en la misma página.
- Se escriben páginas ASP con un simple editor de textos.
- ASP permite usar componentes escritos en otros lenguajes (C++, Visual Basic, Delphi)

que se llaman desde los guiones ASP.
- Los scripts ASP se programan en JavaScript, VBScript, Perlscript y Rexx.

Algunas de las ventajas que ofrecen las páginas ASP son:

- Acceso a bases de datos de una forma sencilla y rápida.
- Las páginas se generan dinámicamente mediante el código de scripts.
- El código de script se ejecuta en el servidor, y no depende del navegador que se use.

Algunas de las desventajas de ASP son:

- La evolución del lenguaje depende de Microsoft
- Difícilmente es portado a otras plataformas. Existen aplicaciones como Chilisoft que permite transportar cierta funcionalidad de ASP a Linux, sin embargo, existen limitaciones, por ejemplo cuando las páginas ASP en Linux pretenden usar componentes COM / ActiveX, las cuales son nativas de Microsoft.
- A menudo es necesario adquirir componentes y en ocasiones pagar por ellos, si es que se necesita cierta funcionalidad no provista por ASP.
- El desarrollador se encuentra sujeto a las directivas de Microsoft. Por ejemplo, en un inicio Microsoft decidió no proveer de un servidor de web a Windows XP Home Edition.

En la tecnología ASP, el concepto de aplicación es definido como el sitio web junto con los archivos ASP que se encuentren en los directorios que el sitio web utilice. Una aplicación comienza cuando se realiza la primer petición de un cliente, de una página alojada en el servidor web.

En ASP existe un archivo llamado global.asa, el cual se encuentra ubicado en el directorio raíz de la aplicación. Dentro de este archivo se declaran los objetos Application y Session; además se inicializan los objetos y variables globales de la aplicación o de la sesión, y proporcionar manipuladores de sucesos para eventos como el inicio de la aplicación ASP.

II.2 - Scripts del lado del cliente y scripts del lado del servidor

Los scripts son programas que permiten la ejecución de código generalmente asociado a eventos. Estos eventos se generan por medio de las acciones que realiza un usuario sobre la página web, por ejemplo cuando se pulsa un botón.

En los lenguajes de scripts se denomina host al navegador web o al motor de páginas ASP. El host es la aplicación en donde se ejecuta el código. Los lenguajes de script solo tienen acceso a los recursos que proporciona la aplicación host.

Los scripts más utilizados en páginas ASP son VBScript y JavaScript. VBScript es un subconjunto de Visual Basic y proporciona un gran número de funciones propias de Visual Basic, mientras que JavaScript es un lenguaje de programación semejante a Java y C/C++ creado por Netscape.

Scripts del lado del cliente

Los scripts del lado del cliente son programas incrustados en el texto HTML. Estos programas se envían al navegador web del cliente, donde son compilados y ejecutados. El navegador del usuario es el host para este tipo de scripts.

Los scripts del lado de cliente están incrustados en los documentos HTML entre las etiquetas "<script>" y "</script>", la etiqueta de inicio lleva una referencia al lenguaje de escritura de scripts que se desee utilizar, mediante la instrucción language="nombre_del_lenguaje".

Por ejemplo, si se desea utilizar VBScript:

<script language="VBScript">
Instrucciones del script.
</script>

Manejando JavaScript, la referencia es de la siguiente manera:

<script language="JavaScript">
Instrucciones del script.
</script>

Algunas de las ventajas de los scripts del lado del cliente son:

- Menor carga en el servidor. La validación de los datos se hace en el lado del cliente.
- Mayor interactividad con la página web.
- No requiere tiempo adicional de descarga, al no tener que esperar a recibir respuesta del servidor.
- Muestra y oculta zonas de una página sin tener que recargar la página.

Por otro lado, algunas de sus desventajas son:

- El código esta expuesto al usuario
- Es necesario que el explorador disponga de un motor de scripts.

Scripts del lado del servidor

Los scripts del lado del servidor se ejecutan en el servidor web. En este caso el host es el motor ASP. El navegador nunca ve el contenido de los scripts del lado de servidor, sino la salida que produce.

El código ASP aparece entre *"<%"* y *"%>"*. Cuando un cliente solicita una página ASP a un servidor Web, éste procesa completamente todo el código que se encuentre entre *"<%"* y *"%>"* antes de enviar la salida al cliente. Por ejemplo:

<%
Instrucciones ASP que son procesadas en el servidor.
%>

El lenguaje predeterminado para el servidor es VBScript. En ocasiones es necesario definir el

lenguaje del lado del servidor. Una forma de cambiar el lenguaje de scripts del lado del servidor es definirlo directamente en la página ASP.

Para especificar el lenguaje de secuencias de comandos de una página ASP, se utiliza la directiva *"@Languaje"*, por ejemplo:

<%@Language=VBSCRIPT%>

Si se deseara utilizar JavaScript en lugar de VBScript, es de la siguiente manera:
<%@Language=JAVASCRIPT%>

Para establecer el lenguaje de scripts en un bloque de código especifico, se utiliza la etiqueta <script> y el atributo language, por ejemplo:

<Script Language=JavaScript Runat=server>
Instrucciones del script
</script>

Dentro de las ventajas de los scripts del lado del servidor se encuentran:

- Se aprovechan los recursos de una maquina central potente.
- Se crea un sitio independiente del explorador, porque todo el código se ejecuta en el servidor web.

Se llevan a cabo tareas que no son posibles con scripts del lado de cliente, como mantener y presentar la forma en que los múltiples usuarios están utilizando el sitio web en un momento dado.

Por otro lado, algunas de sus desventajas son:

- Es necesario que el servidor web permita ejecutar scripts de lado del servidor.
- Un script mal escrito hace que el servidor quede bloqueado.
- Requiere que los recursos del servidor estén disponibles para varios usuarios simultáneamente.

II.3 - Objetos y métodos

ASP contiene seis objetos integrados en el motor ASP. Cada objeto posee una serie de métodos y eventos. A continuación se describen los objetos y se mencionan los métodos y eventos utilizados en la elaboración de la página web.

Los objetos ASP son:

- Request.
- Response.
- Server.
- Session.
- Application.
- ASPError.

Para acceder y manejar bases de datos, ASP cuenta con objetos de bases de datos ActiveX (ADO, ActiveX Database Objects), los cuales se encuentran en la librería ADODB.

Objeto Request

Su función es obtener valores que el navegador del cliente pasa al servidor durante una petición al servidor web, mediante este objeto se accede a toda la información de la petición, por ejemplo la dirección IP del cliente o el tipo de navegador.

Su sintaxis es: *Request.Coleccion(variable).*

Algunas de las colecciones de Request son:

- QueryString: permite acceder a la información contenida dentro de los formularios HTML.
- Form: Es utilizada para acceder a la información cuando los datos son enviados desde un formulario.

En la colección Form, es posible el valor asociado a una variable a través de Request.form(Nombre variable). Las variables son los identificadores asociados al atributo NAME de los elementos incluidos en el formulario.

Objeto Response

Su función es la de enviar datos de salida al cliente que accede al sitio web, los datos son de cualquier clase, por ejemplo, código HTML.

Su sintaxis es *Response.NombredelMetodo.*

Algunos de los métodos de Response son:

- End: Detiene el proceso de una página ASP y devuelve el resultado de las instrucciones ejecutadas hasta ese momento.
- Redirect: Indica al navegador que conecte con otra dirección URL, con esto el cliente es redirigido a otra página.
- Write: Escribe una variable en el resultado HTML actual, en forma de cadena. Esta cadena nos permite enviar código HTML, javascript o cualquier cadena al cliente.

Objeto Server

Este objeto proporciona el acceso a los métodos y propiedades del servidor, y con ello trabajar directamente con sus características.

Su sintaxis es: *server.nombredelmétodo.*

Un método de Server es CreateObject, el cual permite crear una instancia de un componente instalado en el servidor permitiendo crear objetos, por ejemplo se crean objetos de acceso a datos.

Los objetos creados con CreateObject tienen alcance dentro de la página que se esta

procesando, al finalizar la ejecución de la página ASP, dichos objetos se destruirán automáticamente.

Objeto Session

Se utiliza para almacenar la información necesaria para una determinada sesión de usuario. Una sesión comienza cuando un usuario solicita su primera página en el servidor de web. Las variables que almacena el objeto Session permanecen durante toda la sesión de usuario. El servidor web crea automáticamente un objeto Session cuando el cliente solicita por primera vez una página web a la aplicación. El servidor destruye el objeto Session cuando la sesión acaba o es abandonada.

Su sintaxis es: *Session("nombre") = valor*

Algunos de los eventos de Session son:

- Session_OnStart: Se produce cuando un cliente solicita su primera página dentro de la aplicación.
- Session_OnEnd: Se produce cuando un cliente abandona una sesión, o cuando se supera el tiempo de espera.

Objeto Application

Este objeto nos permite compartir información entre todos los clientes de una misma aplicación web.

Su sintaxis es: *Application.nombredelmetodo*

Existe una diferencia entre los métodos y los eventos de un objeto. Los métodos son acciones propias de la naturaleza del objeto, mientras que los eventos responden a la interacción del objeto con otros objetos.

Algunos de los eventos de Application son:

- Application_OnStart. Este evento se produce cuando el primer cliente entra en la aplicación web y se produce antes del evento Session_OnStar.
- Application_OnEnd: Este evento se produce cuando el ultimo cliente sale de la aplicación web, y se produce después del evento Session_OnEnd.

Los scripts correspondientes se declaran en el archivo global.asa.

Objeto ASPError

A través del objeto ASPError, se obtiene información del error que se ha producido durante la ejecución de una página ASP.

Objetos de bases de datos ActiveX

Los objetos de base de datos ActiveX (ADO, ActiveX Database Objects) son una herramienta

de acceso de datos de Microsoft para conseguir un acceso universal a los datos, con un gasto mínimo a través de un proveedor de base de datos OLE (Object Linking and Embedding, Vinculación e incrustación de objetos). Los objetos de acceso a datos (DAO; Data Access Objects) y objetos de datos remotos (RDO, Remote Data Objects) son herramientas mas antiguas que proporcionan un acceso similar.

La tecnología ADO se creó originalmente para su uso desde aplicaciones de Internet y se emplea directamente en secuencias de comandos de servidor y en componentes ASP. Los objetos ADODB proporcionan un acceso flexible a todo tipo de bases de datos.

Dentro de ADO, existen los siguientes objetos:

- Connection.
- Command.
- Parameter.
- RecordSet.
- Field.
- Error.
- Property.

Se cubrirán los objetos Connection, RecordSet y Field, por que fueron los objetos que se utilizarón en el desarrollo de la página web.

Objeto Connection

Este objeto se utiliza para establecer una conexión de la aplicación web con una base de datos. Para usar este objeto, se hace una instanciación, para ello se utiliza la siguiente sentencia:

Set obj_conn = Server.CreateObject("ADODB.Connection")

Propiedades de Connection

ConnectionString: Con esta propiedad, se pasa una cadena de texto al objeto, indicando la base de datos a utilizar, el proveedor, y si es necesario, el usuario y el password de la base de datos. Por ejemplo:

Cn.ConnectionString = "Provider=SQLOLEDB.1; Server=nombre_del_servidor; database = HistoricoBD; user id=nombre_de_usuario; password=contraseña;"

Los componentes de la cadena son:

- Provider: Proveedor OLE DB (En este ejemplo, el proveedor es de una base de datos SQL Server).
- Server: El nombre del servidor.
- Database: El nombre de la base de datos.
- User id: identificador de usuario válido para la base de datos.
- Password: la contraseña del usuario, en caso de no existir, se deja en blanco.

Es recomendable tomar medidas de seguridad para proteger la cadena de conexión, y de esta manera proteger información confidencial como el nombre de usuario y contraseña.

Se recomienda no incrustar la cadena de conexión en el código, sino utilizar un archivo externo para su almacenamiento, por ejemplo un archivo plano.

La cadena de conexión tiene una propiedad denominada Persist Security Info, el valor predeterminado de Persist Security Info es false.

Si el valor de Persist Security Info se establece en true, se permite obtener información de seguridad confidencial, incluyendo el Id. de usuario y la contraseña de la conexión que esté abierta.

Si el valor de Persist Security Info es false, se contribuye a que un origen que no sea de confianza no tenga acceso a la información de la conexión, y contribuye a que no se guarde en disco ningún tipo de información de la cadena de conexión.

Al establecer Persist Security Info como false, la información de la cadena de conexión estará protegida al realizar la conexión.

Ejemplo:
Cn.ConnectionString = "Provider=SQLOLEDB.1;Data Source=nombre_del_origen_de_datos; Initial Catalog = HistoricoBD; user id=sa; password=; PersistSecurity Info= False "

Métodos de Connection
Método Open: El método Open realiza la conexión real con la base de datos, tomando los parámetros previamente configurados. Este método también permite insertar directamente la propiedad ConnectionString, la cual es ejecutada después de crear el objeto Connection

Sintaxis: *Obj_conn.Open.*

Método Close: cierra la conexión con la base de datos con el que el objeto Connection se queda disponible para ser utilizado con otra base de datos.

Sintaxis: *Obj_conn.Close.*

Destrucción del objeto: sirve para liberar memoria y perder todo vinculo con la base de datos.

Sintaxis: *Set Obj_conn = Nothing*

Propiedades del objeto Connection
Provider: Indica el nombre del proveedor para un objeto Connection.

Sintaxis: *Obj_Conn.Provider = "nombredelproveedor".*

Objeto RecordSet
Con este objeto tendremos acceso a la base de datos, ya sea a través de las tablas, de vistas o de consultas que se realicen.

Los objetos RecordSet se emplean para manipular datos de un proveedor. Cuando se

emplea ADO, los datos se manipulan casi siempre utilizando objetos RecordSet. Todos los objetos RecordSet se construyen utilizando registros (filas) y campos (columnas).

Para utilizar este objeto, antes se crea un objeto Connection, ya que este será quien indique al RecordSet de que base de datos tiene que recoger las tablas. Una vez seleccionado una tabla, el objeto RecordSet apuntara al primer registro de la misma. A través de sus métodos y propiedades, podremos ir recorriendo el resto de registros y consultando sus campos.

El procedimiento para instanciar este objeto es muy similar al del objeto Connection:

Set obj_rs = Server.CreateObject("ADODB.RecordSet")

Se asigna a la variable obj_rs un objeto. A continuación se obtiene una instancia del objeto RecordSet mediante el método CreateObject, se indica la librería ADODB, y se finaliza con el objeto a instanciar, en este caso el objeto RecordSet.

Métodos de RecordSet

Método Open: una vez configurado las propiedades del objeto RecordSet, se procede a llenarlo con los datos que se quieran conseguir. Como parámetro se le pasa el nombre de la tabla a seleccionar o una consulta SQL, por ejemplo:

Obj_rs.Open Source, ActiveConnection.

El parámetro Source es una expresión que proporciona al evaluarse una orden sql, un nombre de tabla o una llamada a un procedimiento almacenado.

El parámetro ActiveConnection proporciona al evaluarse un nombre de variable de un objeto Connection válido, o ser un valor de tipo String que contenga parámetros ConnectionString.

Método MoveNext: El RecordSet se mueve un registro hacia delante.

Sintaxis: *Obj_rs.MoveNext*

Método Close: Cierra el RecordSet. Esto hace que desaparezcan todos los datos del mismo.

Sintaxis: *Obj_rs.Close.*

Destrucción del objeto: utilizaremos la misma sentencia que en el objeto Connection, es decir:

Sintaxis: *Set Obj_rs = nothing*

Colección Fields

La colección Fields contiene todos los campos que tiene un registro. Para conocer el valor de un campo, se utiliza cualquiera de las siguientes instrucciones:

Obj_rs.Fields(Número_del_campo) o
Obj_rs.Fields(Nombre_del_campo)

Propiedades de la colección Fields
Name: Se utiliza para conocer el nombre de un campo.

Sintaxis: *Obj_rs.Fields(Número_de_campo).Name*
ú *Obj_rs.Fields(Nombre_del_campo).Name*

Count: se usa para conocer el número de campos del registro.

Sintaxis: *Obj_rs.Fields.Count.*

Propiedades de RecordSet
Propiedad ActiveConnection: Se asigna a esta propiedad el objeto Connection que se haya creado. Esta es la forma de decir al objeto RecordSet la base de datos con la que va a trabajar.
Sintaxis: *Obj_Rs.ActiveConnection = Obj_Conn.*

EOF: Devuelve true cuando la posición de registro actual se encuentra después del último registro del RecordSet.
Sintaxis: *Obj_rs.EOF.*

BOF: Devuelve true cuando la posición del registro actual esta antes del primer registro del RecordSet.
Sintaxis: *Obj_rs.BOF.*

Objeto Field
La función del objeto Field es representar una columna de datos con un tipo de datos común. Un objeto RecordSet tiene una colección de Fields hecha de objetos Field. Cada uno de estos objetos se corresponde con una columna del objeto RecordSet. Se emplea la propiedad Value de los objetos Field para fijar o devolver datos para el registro actual.

Propiedades del objeto Field
Name: indica el nombre de un registro.
Value: Indica el valor asignado a un registro.

Objeto Command
Es una definición de un comando especifico que se intenta ejecutar sobre una fuente de datos.

Objeto Parameter
Representa un parámetro o argumento asociado con un objeto de comando basado en una consulta o procedimiento almacenado parametrizado.

Objeto Error
Contiene detalles acerca de los errores de acceso a datos correspondientes a una única operación en la que el proveedor esta involucrado.

Comandos y funciones en ASP

ASP cuenta con muchas funciones. Dentro del código de la página Filtro.asp, se utilizaron las siguientes instrucciones:

Trim: Elimina espacios dentro de una cadena.
Sintaxis: Trim(cadena).

Len: Devuelve la cantidad de caracteres de cadena.
Sintaxis: Len(cadena).

Mid: Devuelve la longitud de caracteres de cadena, comenzando en la posición de inicio. Cuando la longitud es mayor que la cantidad de caracteres que se quedan en la cadena, devuelve el resto de la cadena. Si no se especifica longitud, devuelve el resto de la cadena comenzando en la posición inicial especificada.

Sintaxis: Mid(cadena,inicio, longitud).

Almacenamiento en búfer en ASP

La salida que ve el cliente, es enviada de dos formas diferentes: con almacenamiento en búfer o sin almacenamiento en búfer. La salida al cliente sin búfer se envía inmediatamente, mientras que la salida al cliente con búfer se envía hasta que la secuencia de comandos está terminada o hasta que se da una instrucción especial para enviarla.
Con el almacenamiento en búfer, todas las salidas al cliente se almacenan en un búfer en el servidor y se envían juntas.

Para indicar si se desea activar el búfer, se emplea la siguiente instrucción:
Response.Buffer = True.

Para desactivar el búfer se utiliza:
Response.Buffer = False.

II.4 - Estructuras de control

Dentro de la programación de la página web, existen casos en los que se requieren de instrucciones que permitan hacer comparaciones entre variables o repeticiones acordes a una condición. Las instrucciones que permiten consultar condiciones y además permiten ejecutar repetitivamente una o varias instrucciones, se denominan estructuras de control.

ASP emplea el lenguaje VBScipt, el cual contiene tres tipos de estructuras de control:

- Instrucciones de bifurcación.
- Bucles.
- Subrutinas.

Instrucciones de bifurcación

En este tipo de instrucciones, se comprueba si una condición es cierta o falsa, después en función del resultado, ejecuta una porción de código y se ignora otra porción, estas instrucciones son:

- If....Then....Else: Son instrucciones condicionales entre dos opciones, las cuales quedan establecidas por una condición que devuelve un subtipo Boolean.
- Select Case: Son instrucciones condicionales entre varias opciones.

Por ejemplo, para la estructura de la instrucción If....Then:

```
if (condición) then
        bloque de instrucciones
End if
```

Por otra parte, para la instrucción If....Then....Else, su estructura es la siguiente:

```
if (condición) then
        bloque de instrucciones
Else            //En caso de que no cumpla la condición
        Bloque de instrucciones
End if
```

Para la instrucción select case:

```
Select case
Case "condición 1"
        Instrucciones
Case "condición 2"
        Instrucciones
Case "condición 3"
        Instrucciones
End select
```

Bucles

Un bucle permite ejecutar un conjunto de instrucciones de manera repetitiva. Las instrucciones de bucle en VBScript se agrupan en dos tipos:

- Instrucciones tipo For....Next: estas instrucciones se emplean cuando se conocen el número de iteraciones que se realizan.
- Instrucciones tipo While: este tipo de instrucciones se utiliza cuando no se sabe de antemano el número de repeticiones que hay que realizar. El número de repeticiones se controla mediante una expresión que se evalúa en cada iteración.

Dentro del tipo While, tenemos dos variantes: el bucle Do While ....Loop y el bucle Do....Loop While

Para el caso del bucle Do While.....Loop, la instrucción indica que se itere mientras que la

condición inicial se cumpla.

En el caso del bucle del tipo Do....Loop While, se ejecuta al menos una vez las instrucciones contenidas en el bucle.

La estructura de la instrucción tipo For....Next es:

For valor_inicial to valor_final
Bloque de instrucciones
Next

Mientras que para Do While....Loop y Do....Loop While es:

Do while (condicion)
Bloque de instrucciones
Loop.

Do
Bloque de instrucciones
Loop while (condición)

Otra estructura de control son las subrutinas, las cuales permiten agrupar instrucciones que podrán ejecutarse desde diferentes lugares de un programa. De esta manera se evita la repetición del código y se facilita su utilización. En VBScript las subrutinas se definen de la siguiente manera:

Sub nombrederutina
Instrucciones
End Sub

ASP.NET

ASP.NET es un entorno de programación que se ejecuta en un servidor web para producir y administrar de forma dinámica formularios Web.

ASP.NET es compatible con la tecnología ASP 3.0 (tradicional), con las tres excepciones siguientes:

- Objeto Request(): ASP devuelve una matriz de cadenas; ASP.NET devuelve una cadena.
- Objeto Request.QueryString(): ASP devuelve una matriz de cadenas; ASP.NET devuelve una cadena.
- Objeto Request.Form(): ASP devuelve una matriz de cadenas; ASP.NET devuelve una cadena.

Las páginas ASP.NET incorporan varios cambios semánticos con respecto a las páginas ASP 3.0, por ejemplo, las páginas ASP.NET sólo admiten un único lenguaje, mientras que ASP 3.0 permite el uso de varios lenguajes en una única página.

En ASP.NET las funciones se declaran en bloques que contengan la instrucción *"<script runat=server>"* mientras que en ASP 3.0 se declaran funciones de página dentro de bloques <% %>

ASP.NET utiliza una extensión de nombre de archivo distinta (.aspx en lugar de .asp), una configuración independiente y una biblioteca denominada Common Language Runtime.

Algunas de las nuevas características de ASP:NET que se pueden aprovechar son:

- Rendimiento y escalabilidad mejorados
- Almacenamiento de resultados en caché y seguridad personalizada
- Controles de páginas de formularios Web
- Infraestructura de servicios Web de XML

ASP.NET conserva la mayoría de las características de ASP 3.0, no ha sido posible mantener el 100% de la compatibilidad entre las dos tecnologías al cambiar de plataforma. Salvo aplicaciones con una gran complejidad, la migración de ASP 3.0 a ASP.NET no tiene mayores dificultades.

II.5 – Uso de Internet Information Server para crear un sitio web

Para procesar una página ASP no existe restricciones en el lado del cliente, sin embargo del lado del servidor es necesario un servidor web que procese páginas ASP. Además, el sistema operativo Windows utiliza el archivo ASP.DLL para interpretar el código ASP.

ASP es una herramienta de desarrollo web y a la vez un entorno, y por tanto requiere de un servidor web. Uno de los servidores que procesa ASP es el Internet Information Server (IIS). En el libro **ADMINISTERING IIS 5.0** de los autores Santry Patrick y Tulloch Mitch, se profundiza el estudio del IIS.

IIS tiene sus funciones limitadas en equipos con sistemas operativos como Windows 2000, por lo que no es buena opción si se desea crear sitios con gran cantidad de usuarios, sin embargo, en sistemas operativos como Windows 2000 Server, IIS posee la potencia de un servidor web.

La administración de Internet Information Server se realiza a través de la herramienta Microsoft Management Console (MMC). Esta consola unifica la interfaz de utilización de los distintos programas de administración de recursos de la red. MMC es una herramienta consolidada para administrar el servidor desde una única interfaz.

La figura 2.1 muestra la consola MMC de Windows XP.

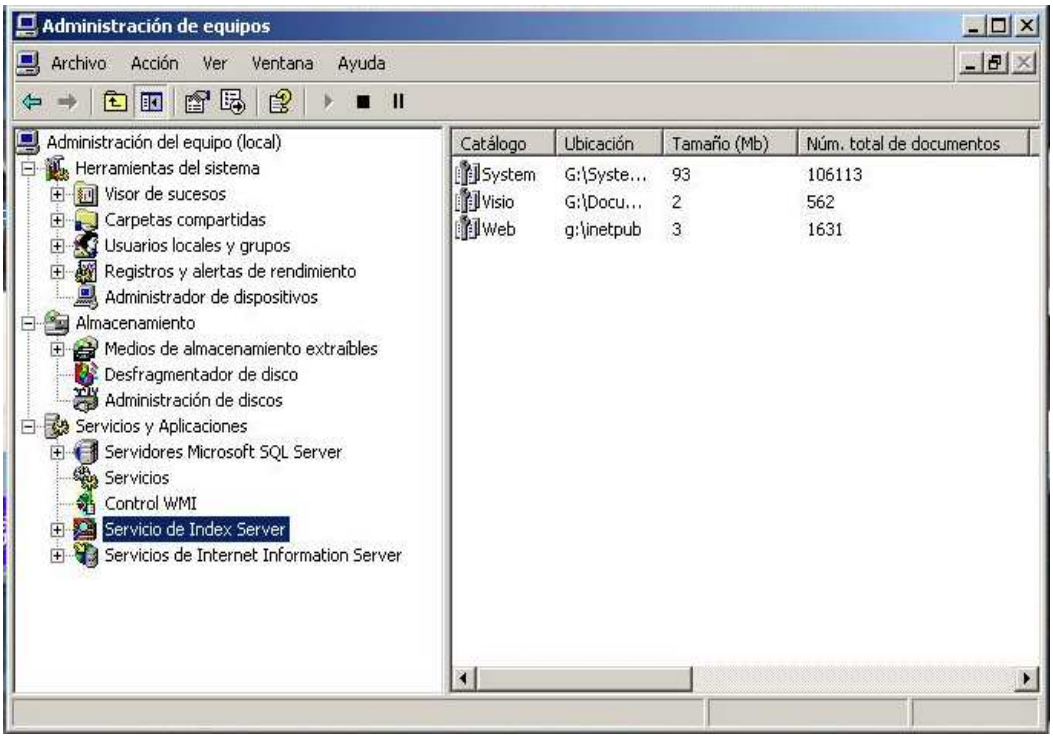

Figura 2.1

La interfaz de la consola de gestión se divide en tres áreas principales. En la parte superior se encuentra la barra de menú y la barra de herramientas.

En la parte izquierda se encuentra el panel de ámbito. El ámbito representa el servicio con el que se este interactuando, se presenta como un árbol denominado espacio de nombres, que se presenta en forma descendente desde el nivel raíz. Cada uno de los módulos representa un servicio o un conjunto de servicios dentro del espacio de nombres.
A medida que se van seleccionando los nodos del espacio de nombres, van apareciendo elementos dentro del panel de resultados, ubicado en el lado derecho de la MMC. Los elementos que aparezcan en el panel de resultados dependen de la selección en el espacio de nombres

IIS integra tecnologías de acceso a bases de datos dentro de MDAC (Microsoft Data Access Component). Se hacer uso de ODBC (Open DataBase Conectivity), ADO (Microsoft Actives Data Objects) y Microsoft Remote Data Services.

Microsoft Index Server (MIS) es una aplicación capaz de generar un índice de los contenidos y las propiedades de las páginas de un sitio web administrado por IIS. La interfaz con que actúan los usuarios esta basada en formularios HTML. Las respuestas procesadas se envían al usuario en forma de páginas Web.
Cada sitio web posee un único directorio raíz. Este directorio es la ubicación predeterminada del sitio en el servidor web y contiene como mínimo la página predeterminada que aparece en el navegador cuando el navegador accede al sitio web.

El directorio raíz se denomina Inetpub. Dentro de este directorio existen otros directorios, algunos de los cuales son:

- Iissamples: Archivos con ejemplos que muestran las capacidades de IIS.
- Mailroot: Contiene directorios para enviar y recibir mail.
- Scripts: En este directorio se localizan los scripts que pertenecen al sitio web.
- Wwwroot: El directorio base por defecto para el servicio de publicación WWW. Es en este directorio en donde se colocan todos los archivos que se desean publicar en web.

II.6 – Uso de Microsoft InterDev 6 para crear páginas ASP

Microsoft InterDev 6 es una herramienta de desarrollo de aplicaciones para la web. Proporciona un ambiente de desarrollo con facilidades para el programador.
Algunas de las características de Visual InterDev son:

- Desarrollo rápido de aplicaciones.
- Potencia en el desarrollo de la parte de servidor.
- Compatibilidad con los últimos estándares para la web.

Visual InterDev se centra en el desarrollo rápido de aplicaciones (RAD, Rapid Application Developed). RAD se centra en el diseño de aplicaciones de forma rápida y eficiente usando métodos visuales. Además, proporciona para cada página una propiedad que especifica el lenguaje de escritura de scripts predeterminado para los scripts del lado del cliente y del lado del servidor.

El libro de **EDICIÓN ESPECIAL VISUAL INTERDEV 6,** de los autores Banick Steve y Michael Morrison se hace un estudio más a detalle sobre las características y componentes de este software. Al final del capítulo se proporciona la referencia completa.

Modo local y modo maestro

InterDev trabaja en modo local y modo maestro. En modo maestro, las páginas web son almacenadas en el directorio raíz del servidor; mientras que en modo local, se crea una copia de los archivos y se almacenan en un directorio diferente, asignado por InterDev o aquel que el usuario defina.

La ventaja de trabajar en modo local, es por que el usuario realiza las modificaciones a su página web, sin alterar la copia que esta disponible en web. Al trabajar en modo maestro, el usuario corre el riesgo de cometer un error y causar problemas en la página publicada en el servidor.

II.7 – Breve referencia a HTML y JavaScript

Dentro de la referencia a HTML, se abarcará la creación de un formulario, así como la descripción de algunos elementos que lo conforman; además se incluye una introducción al lenguaje JavaScript.

Formulario

Un formulario es un conjunto de elementos que permite a las aplicaciones Web comunicarse con los clientes. Esta comunicación consiste en una transferencia de datos, que el cliente introduce en un documento HTML.

Los formularios están compuestos por textos y controles. Los controles son cuadros de texto, botones de opción, casillas de verificación, entre otros.

Los formularios se dividen en dos tipos, de acuerdo al método de procesamiento de información:

- Formularios dirigidos a una persona: en este tipo de formularios, la información es devuelta encapsulando un mensaje de correo electrónico que se envía a una dirección personal, normalmente la del creador de la página web.

- Formularios dirigidos a un servidor: estos formularios se caracterizan porque la información es retornada a un programa especializado que reside en el servidor de procedencia del documento HTML, y que procesa los datos del formulario.

Los métodos para enviar información a través de los formularios son GET y POST.

En el método GET, toma la información del formulario, concatena los datos que componen el formulario formando una cadena de consulta con los pares “Elemento = valor”, y cada par va separado por el símbolo &, después esta cadena se anexa al final de la URL puesta en el elemento action del formulario.

El método POST es el que se usa generalmente para enviar formularios. Es empleado para formularios dirigidos a personas.

La estructura de un formulario es la siguiente:

1. Etiqueta inicial.
2. Cuerpo del formulario.
3. Botón de envío o de borrado.
4. Etiqueta de cierre.

El cuerpo del formulario esta integrado uno o varios elementos, los cuales son:

- Cuadro de texto. Es un cuadro con espacio para una línea de texto, en donde se escriben cadenas de texto, con o sin desplazamiento lateral y / o cifrado mediante asteriscos *.
- Área de texto. Se utiliza para la inserción de un texto de múltiples líneas.
- Lista de selección: Se presenta diversas opciones de una lista, permitiendo la selección de una de ellas.
- Casilla de verificación: Permite la confirmación de la opción asociada, mediante un clic del ratón. Si existe más de una, el formulario permite la selección de múltiples casillas.
- Botón de radio: Permite la selección de una única opción de todas las demás.

Otros elementos del formulario son:

- Botón de envío: Pulsando este botón se envían los datos del formulario al servidor o a la persona señalada. Su inclusión es obligatoria.
- Botón de borrado: Con este botón se realiza la limpieza del formulario, borrando el contenido de cada elemento, para volver a introducir mas datos. Su inclusión es opcional.

Para crear un formulario, se utiliza la etiqueta inicial <FORM> y al final del mismo se emplea la etiqueta de cierre </FORM>.

La etiqueta de apertura del formulario <FORM> suele ir acompañada de una serie de atributos que indican las características del formulario. Los atributos son:

- Name: identifica el elemento. Todo elemento del formulario tiene una variable de identificación, y esto se hace con el atributo name.
- Action: indica a donde es enviada la información.
- Method: determina la manera en que se codifican y envían los datos del cliente al servidor.

Ejemplo:

<FORM name=frmFiltro action="Filtro.asp" method=post>

El formulario se llama frmFiltro, los datos serán enviados a la página Filtro.asp, y usa el método POST.

A continuación se presenta los elementos correspondientes a las listas de selección y botón de envío, por que fueron utilizados en la construcción del formulario de la página web.

En la obra de Gunter Born **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML,** hay un apartado que explica más a detalle los elementos que componen un formulario. Al final de este capítulo se encuentra la bibliografía completa.

Listas de selección

En las listas de selección, el cliente elige una opción o más opciones dentro de todas las propuestas. La estructura de una lista de selección, se compone de:

- Etiqueta de apertura “<select>”.
- Opciones de la lista “<option>”.
- Etiqueta de cierre “</select>”.

Las etiquetas <select> se definen siempre dentro de las etiquetas <FORM>.

Dentro de las etiquetas “<select>”, se definen las opciones que contendrá la lista. Esto se hace con la etiqueta <option>. En la etiqueta <option>, se define el atributo value, el cual establece el valor que la lista devuelve al servidor al enviar los datos del formulario.

El texto que aparece entre las etiquetas <option> será el nombre de la opción el que el cliente vea en el navegador. Por ultimo, cada opcion se cierra con la etiqueta </option>
Por ejemplo:

<select name =”nombre”>
<option value=”otros valores”> opciones </option>
</select>

Las listas permiten realizar selecciones múltiples. Para ello se utiliza el atributo MUTIPLE, el cual indica que el menú permite múltiples selecciones de opciones. Se indica el numero de opciones que se muestran simultáneamente en la pantalla.

Botón de envío

A través de este botón, se envían los datos del formulario al servidor. Estos botones utilizan la etiqueta input, el cual permite pasar un conjunto de variables con sus respectivos valores en un formulario. En el caso del botón envío, se configura con el valor “submit”, por ejemplo:

<input type = “submit” value=”enviar”>

En la página *http://www.osmosislatina.com/lenguajes/html/formas.htm,* se encuentran descritos otros tipos de *input types.*

JavaScript

JavaScript es un lenguaje orientado a objetos, estos objetos poseen propiedades y sobre ellos se aplican métodos. Además, permite reaccionar a eventos.

El núcleo de cualquier programa JavaScript son los objetos. El programa aplica métodos concretos sobre los objetos, además de leer y definir sus propiedades.

Objetos

JavaScript contiene objetos propios definidos, algunos de ellos son:

- Window: Es el objeto más importante, permite acceder a la ventana del navegador y por tanto al documento.
- Document: este objeto engloba todo el documento HTML. Para escribir algo en el documento html, se emplean los métodos del objeto document.
- Form: un formulario es un subobjeto de un documento. Con este objeto se accede a los elementos de cualquier formulario.
- Elements: es una clase formada por todos los objetos que forman parte de un formulario: botones, cuadros de texto, cuadros de lista, etc.

Propiedades

Los objetos de JavaScript están dotados de propiedades. Por ejemplo, el objeto document

contiene una serie de propiedades como el color de fondo. El acceso a propiedades se realiza indicando el objeto al que pertenece la propiedad, posteriormente se le agrega al objeto la propiedad, separándolos por medio de un punto. Por ejemplo, para acceder a la propiedad del color de fondo se utiliza la siguiente instrucción:

Color = document.bgcolor;

Por otra parte, si se desea definir el color, se asigna un nuevo valor a la propiedad, por ejemplo:

Document.bgcolor =”naranja”

Métodos

JavaScript soporta una serie de métodos que se aplican a los objetos. El objeto y el método van separados por un punto. Por ejemplo, el objeto document contiene el método createElement, el cual crea una nueva instancia de un objeto para una etiqueta específica. Su sintaxis es:

document.createElement(tag).

Eventos

Existen determinados elementos de HTML que soportan eventos como el pulsar el mouse, o una tecla. En el atributo de evento indica el comando JavaScript con el que reacciona el evento. Un ejemplo de atributo de evento es onClick, el cual se produce al pulsar un elemento con el ratón.

Definición de funciones

JavaScript usa funciones, estas se ejecutan hasta que se activan explícitamente. La forma de declarar una función es:

Funtion name()
{ Instrucciones de la función.
Return }

La función se introduce con la palabra clave *Funtion*. A cada función se le asigna un nombre unívoco. El nombre va seguido por paréntesis, dentro de los cuales van los parámetros. Las instrucciones de la función van entre llaves { y }. Para terminar la función, se incluye la instrucción return en el código de JavaScript. Los valores que devuelve la función se producen en la forma return valor.

Para activar la función, solo se escribe su nombre seguido de paréntesis, es decir, *nombrefuncion()*

Declaración de variables

Para declarar variables se utiliza la palabra var, esta se antepone al nombre de la variable, por ejemplo:

var nombrevariable
ó
var nombreVariable = expresion

Creación de objetos.
Para crear una instancia de un objeto se emplea la palabra new. Para utilizar el objeto, primero se define el tipo de objeto. La sintaxis de creación de un nuevo objeto es:

nombreObjeto = new nombretipoobjeto (parámetros)

La siguiente parte presenta las principales características del manejador de bases de datos SQL Server, el cual fue empleado para almacenar los datos de la página web.

REFERENCIAS DE LA PARTE II

LIBROS CONSULTADOS

1. - Banick Steve, Michael Morrison. **EDICIÓN ESPECIAL VISUAL INTERDEV 6.** Prentice Hall. Madrid 1999.

5.- Bobadilla Sancho, Jesús. Alcocer Jarabo, Alejandro. Manzaneque Sánchez Luis Rodríguez. **ACTIVE SERVER PAGES (ASP 3.0). INICIACIÓN Y REFERENCIA.** Osborne McGrawHill. España 2002.

2.- Gunter, Born. **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML.** Marcombo. 2001

3.- Power Shelley. **DESARROLLO DE COMPONENTES ASP.** Anaya Multimedia. Madrid 2001.

4.- Santry, Patrick. Tulloch Mitch. **ADMINISTERING IIS 5.0.** McGraw Hill. U.S.A. 2000.

PÁGINAS DE INTERNET CONSULTADAS.

Actualización de IIS versión 5 a versión 6
a) http://support.microsoft.com/kb/325889/es
2006 Microsoft Corporation

Desventajas de ASP
b) http://www.prosumedia.com.mx/documento.php?id_not=6

Listado de las versiones de IIS
c) http://support.microsoft.com/kb/224609/es
2006 Microsoft Corporation

Migración de ASP 3.0 a ASP.NET
d) http://es.gotdotnet.com/quickstart/aspplus/doc/migrationoverview.aspx

Tipos de input types
e) http://www.osmosislatina.com/lenguajes/html/formas.htm
Actualizado : 2005/10/21

III. Migración de datos con Microsoft SQL Server

III. Migración de datos con Microsoft SQL Server

En este apartado se presenta la descripción de los elementos utilizados del manejador de bases de datos Microsoft SQL Server.

Al momento de desarrollar la página web, la Dirección de Prestaciones Económicas y Sociales (DPES), utilizaba el manejador SQL Server 2000. En este trabajo se utilizarón los manejadores SQL Server 2000 y SQL Server 2005 Express Edition.

La DPES requirió migrar parte de su información histórica almacenada en archivos Excel hacia el manejador de bases de datos SQL Server, por este motivo se presenta el proceso de migración de archivos Excel a el manejador SQL Server.

III.1 - Introducción

Microsoft SQL Server es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR o RDBMS, Relational Database Management System) diseñado para trabajar con grandes cantidades de información y tiene la capacidad de cumplir con los requerimientos de proceso de información para aplicaciones comerciales y sitios Web.

La versión actual de este manejador de bases de datos es SQL Server 2005. Dentro del siguiente subtema "*Versiones de Microsoft SQL Server"*, se proporciona un panorama general de este manejador.

Una forma efectiva de almacenar datos es mediante las bases de datos relacionales, las cuales están basadas en la aplicación de la teoría matemática de los conjuntos al problema de la organización de los datos. En una base de datos relacional, los datos están organizados en tablas.

Una tabla es un objeto de la base de datos que contiene información. Una tabla está compuesta por columnas (atributos) y filas (tuplas). Cada columna representa un dato aislado y en bruto que por sí solo no brinda información, por lo tanto estas columnas se agrupan y forman una fila para obtener conocimiento acerca del objeto tratado en la tabla.

Una vista es un objeto definido por una consulta, la vista muestra un conjunto de columnas y filas de datos con un nombre. Sin embargo en la vista no existen datos, estos son obtenidos desde las tablas especificadas en el diseño de la vista, ya que si la información contenida en las tablas cambia, los cambios también serán aplicados a la vista.

Los tipos de datos especifican que tipo de valores son permitidos en cada una de las columnas que conforman la estructura de la fila, para ello SQL Server ofrece un conjunto de tipos de datos predefinidos.

Versiones de Microsoft SQL Server

Microsoft SQL Server 2000

Microsoft SQL Server 2000 está disponible en las siguientes versiones:

- Versión Enterprise: soporta todas las características de Microsoft SQL Server 2000. Esta edición esta dirigida a empresas que implementan medianas y grandes bases de datos, las cuales brindan recursos a soluciones web, tienen un alto índice de trabajo transaccional, y cuentan con soporte para datawarehouse.

- Versión Estándar: es ideal para aplicaciones que necesiten brindar información a grupos de trabajos o departamentos dentro de una organización.

- Versión Personal: soporta todas las características de Microsoft SQL Server 2000 Standard Edition, excepto la replicación transaccional, además no se encuentra disponible el full text search cuando se instala sobre Windows Me y Windows 98.

- Versión Developer Edition: es una edición para desarrolladores que emplean Microsoft SQL Server como su origen de datos. Soporta todas las características de SQL Server 2000, además de un conjunto de herramientas gráficas para la configuración de idiomas.

- Versión Enterprise Evaluation Edition: soporta todas las características de Microsoft SQL Server 2000, a excepción de las herramientas gráficas para configuración del lenguaje. Esta edición es funcional durante un período de 120 días.

- Versión Microsoft SQL Server 2000 Desktop Engine: es una versión del motor de base de datos relacional de Microsoft SQL Server 2000. Esta edición es empleada para aquellas aplicaciones que no requieran la implementación de tareas administrativas para el cliente.

Microsoft SQL Server 2005

SQL Server 2005 es la más reciente versión del manejador de bases de datos de Microsoft. El objetivo de Microsoft es constituir a Microsoft SQL Server 2005 como una de las versiones con mayor variedad de características, las cuales incluyen mejoras en la administración de información, productividad en el desarrollo y rendimiento de las aplicaciones a nivel empresarial.

Microsoft ha rediseñado la familia de productos SQL Server 2005 para satisfacer las necesidades de cada segmento de sus clientes con las siguientes versiones:

- Express
- Workgroup
- Standard
- Enterprise.

En la dirección web *http://www.microsoft.com/spain/sql/productinfo/features/compare-features.mspx*, se encuentra más información sobre las versiones de SQL 2005.

Se utilizó el servidor SQL Server 2005 Express Edition., por que una versión compacta,

gratuita, requiere menos memoria ram que las otras versiones de SQL Server 2005 y es compatible con el sistema operativo Windows XP Profesional.
Cabe hacer mención que para instalar SQL Server 2005 Express Edition, fue necesario realizar las siguientes actualizaciones:

- Windows XP Service Pack 2.
- .NET Framework 2.0.
- Windows Installer 3.1.

La descarga de SQL 2005 Express Edition en español esta disponible en la siguiente página: *http://www.microsoft.com/downloads/details.aspx?familyid=220549B5-0B07-4448-8848-DCC397514B41&displaylang=es*

Los requisitos de instalación de SQL Server 2005 (para todas las versiones), se encuentra disponible para su consulta en la dirección *http://msdn2.microsoft.com/es-es/library /ms143506.aspx* ó en la dirección *http://download.microsoft.com/download/a/2/3/a23083ba-88d4-4e89-b9fb-dfd3b618bbdb/RequirementsSQLEXP2005.htm.*

Para administrar el servidor SQL 2005, se instaló el software "Microsoft SQL Server Management Studio Express", el cual puede descargarse gratuitamente de la página de Microsoft.

En la página web *http://msdn2.microsoft.com/es-es/library/ms165690.aspx,* se encuentra más información sobre el SQL Server Management Studio.

La base de datos de la página web fue construida en el servidor SQL Server 2000, se manejó SQL Server 2000 por que la versión del servidor SQL Server 2005 Express Edition no incluye la herramienta de transformación de datos "Servicio de Transformación de Datos (DTS, Data Transformation Services)", esta herramienta se manejó para realizar la migración de los datos de la Dirección de Prestaciones Económicas y Sociales.

Las características que no están incluidas SQL Server 2005 Express Edition se encuentran en la página web:
https://www.microsoft.com/spanish/msdn/articulos/archivo/261205/voices/sseoverview.mspx#E2FAC.

En la página *https://www.microsoft.com/spanish/msdn/articulos/archivo/261205/voices/sseoverview.mspx#E2FAC* y en *http://msdn2.microsoft.com/es-es/library/ms143719.aspx* se presenta una guía de instalación de SQL Server 2005, así como información adicional acerca de este manejador de bases de datos.

SQL Server 2005 Express es compatible con los mismos proveedores nativos y administrados que el resto de las versiones de SQL Server 2005. Esto conlleva de que una aplicación escrita para SQL Server Express funciona con otras ediciones de SQL Server.

Instancias de usuario

Las instancias de usuario son una característica de SQL Server Express que ofrece la

posibilidad de tratar las bases de datos como archivos. SQL Server 2005 instala por defecto la instancia denominada SQLEXPRESS.
Las instancias de usuario permiten que un usuario que no es administrador adjunte y administre la base de datos de SQL Server Express. Para ello, se crea una copia privada de SQL Server que se ejecute en el contexto de seguridad del usuario que abre la conexión a la base de datos.

El usuario que abre la conexión se convierte en la cuenta de servicio para la instancia de usuario y por consiguiente, tiene todos los permisos de administrador del sistema para la base de datos.

Estos permisos permiten que un usuario de Windows distinto del administrador tenga permisos de administrador de base de datos.

Dichos permisos permiten además al usuario crear todos los objetos de base de datos necesarios.

En la página *https://www.microsoft.com/spanish/msdn/articulos/archivo/261205/voices/sseoverview.mspx#E2FAC* esta disponible más información sobre las instancias de SQL Server 2005.

Archivos físicos de la base de datos

Un archivo de base de datos es un archivo del sistema operativo. Una base de datos se distribuye en por lo menos dos archivos: un archivo para datos y otro para el registro de las transacciones (log).

Microsoft SQL Server permite los siguientes tipos de archivos:

- Archivos de datos primarios: Toda base de datos tiene un archivo de datos primario que realiza el seguimiento de todos los demás archivos, además de almacenar datos. Por convenio este archivo tiene la extensión MDF.

- Archivos de datos secundarios: Una base de datos tiene cero o varios archivos de datos secundarios. Por convenio la extensión recomendada para los archivos de datos secundarios es NDF.

- Archivos de registro (LOG): Todas las bases de datos por lo menos tendrán un archivo de registro que contiene la información necesaria para recuperar todas las transacciones que suceden sobre la misma. Por convenio la extensión de este archivo es LDF.

Toda base de datos creada en Microsoft SQL Server tienen un archivo de base de datos primario (.mdf) y uno para el Log de Transacciones (.ldf). Además tiene archivos de datos secundarios (.ndf). Al crearse una base de datos, se crea una copia de la base de datos Model y se incluye en la nueva base de datos. El Log de Transacciones lleva toda la información necesaria para la recuperación de la base de datos en una eventual caída del sistema.

III.2 - El Administrador Corporativo (SQL Server 2000)

El Administrador Corporativo es una herramienta que agrupa toda la configuración del servidor y funcionalidad de mantenimiento, así como el diseño y el mantenimiento de las bases de datos; además es un completo conjunto de interfaces graficas para todas las funciones de administración de Microsoft SQL Server 2000.
El administrador corporativo esta construido como complemento para la consola de administración de Microsoft (MMC).

Mediante el Administrador Corporativo, se realiza una serie de tareas, como registrar un servidor, configurar esquemas de mantenimiento, programación y duplicación de bases de datos.

A continuación se presenta un panorama general del Administrador Corporativo.

Registrar servidores

Antes de usar el Administrador Corporativo, es necesario registrar el servidor o los servidores con los que se van a trabajar. Esto se hace mediante el asistente para registro de servidores o con un cuadro de diálogo en el que se incluye la información necesaria.

Una vez registrado al menos un servidor, se usan las herramientas del Administrador Corporativo. Mediante el panel de la izquierda de la ventana, se accede a grupos de servidores, servidores y bases de datos, permitiendo seleccionar el servidor, la base de datos o la tabla con los que se pretende trabajar.

El administrador corporativo cuenta con las siguientes herramientas:

Servicio de Transformación de Datos: permite recuperar información de un origen de datos, realizar transformaciones sencillas o complejas y almacenarlos en otro origen de datos, como una base de datos SQL

Administración: proporciona acceso a distintas áreas claves de la administración del servidor, se accede a funciones necesarias para gestionar y administrar los servidores. Dentro de Administración, se encuentran las siguientes funciones:

- Agente SQL Server: es el responsable de ejecutar trabajos programados, supervisar la actividad y enviar mensajes de alerta a los operadores configurados sobre el estado del servidor y los trabajos programados.

- Copia de seguridad: permite crear, eliminar y cambiar el nombre de los dispositivos de copias de seguridad. También realiza copias de seguridad de bases de datos. Aquí es donde se encuentran los contenidos de los dispositivos de copias de seguridad que se han creado.

- Actividad: La actividad de los usuarios proporciona información sobre aquellos que están conectados al servidor. Con esta característica, se descubre que usuarios están conectados, desde que equipo están conectados y que aplicaciones están usando

para conectarse al servidor.

- Planes de mantenimiento de base de datos: Permite establecer un servidor de mantenimiento programado con solo seguir los pasos del asistente. El asistente incluye tareas como comprobar la integridad, reorganizar los datos y copias regulares de las bases de datos y los registros de transacción.

- Registros de transacción de SQL Server: Permiten observar lo que ha estado sucediendo en el servidor desde la ultima vez que se inicio. Proporciona un buen historial para poder diagnosticar la mayoría de los problemas que tienen lugar en el entorno.

Seguridad: Permite configurar usuarios, inicios de sesión, roles del servidor y permisos asociados para las bases de datos. Los inicios de sesión es donde se establecen las configuraciones para conectarse al servidor de todos aquellos que tengan permiso para hacerlo. El inicio de sesión contiene un nombre, una contraseña y la base de datos a acceder.

Creación de vistas

Una vista se usa de forma parecida a una tabla. Una vista se define creando una instrucción de selección para recuperar columnas de una o más tablas de la base de datos.

Existen dos usos principales de las vistas. La primera es simplificar uniones complejas estableciendo una vista que presenta los datos al usuario o desarrollador de una forma simplificada y no normalizada. La segunda es incrementar la seguridad permitiendo acceder a vistas de columnas de las tablas y no a las tablas subyacentes de las vistas.

Para crear una nueva vista, se selecciona el nodo Bases de Datos del panel izquierdo, al desplegarse el árbol se selecciona la base de datos que se usará, se despliega un subárbol con opciones de la base de datos. Se pulsa con el botón derecho sobre el nodo Vista, al aparecer el menú emergente se selecciona la opción "Nueva vista", con la cual se activa el cuadro de diálogo que permite diseñar una vista para la base de datos.

En la parte superior del cuadro de diálogo, se agrega la tabla o el conjunto de tablas que se usarán en la vista y las relaciones que las unen.

La segunda parte permite seleccionar las columnas que se devolverán o se usarán en el criterio de la selección de la consulta.

La tercera parte muestra el código SQL generado para la consulta; en esta sección es posible agregar código SQL.

La parte final es el panel resultante para la consulta que muestra los datos devueltos cuando se ejecuta la consulta en la vista.

SQL Server Management Studio
(SQL Server 2005)

SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, configurarlos, administrarlos y desarrollarlos. Esta incluido en SQL Server 2005 excepto en la versión Express Edition, el SQL Server Management Studio para la versión Express Edition esta disponible para su descarga gratuita en la página de Microsoft.

SQL Server Management Studio combina un grupo de herramientas gráficas con una serie de editores de secuencias de comandos.

SQL Server Management Studio combina las características del Administrador Corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un entorno único.

Habilitación de protocolos para conexiones remotas. (SQL Server 2005 Express Edition)

De forma predeterminada, SQL Server 2005 Express Edition y SQL Server 2005 Developer Edition no permiten conexiones remotas., por lo que al intentar conectar desde un equipo remoto con una instancia de SQL Server 2005 Express Edition, se recibe un mensaje de error.

Para que la página web se conectara con el servidor SQL Server 2005, se configuró para permitir conexiones remotas. Se ejecutaron los siguientes pasos:

1.- Se abrió la carpeta SQL Server 2005, Herramientas de configuración, SQL Server Configuration Manager.

2.- Se abrió el nodo Protocolos de cliente.

3.- En la lista de protocolos, se seleccionó con el botón secundario en los protocolos deshabilitados, a continuación se seleccionó la opción Habilitar (Figura 3.1).

4.- Se procedió a reiniciar el sistema para que los cambios tomaran efecto.

Figura 3.1

Creación de un usuario en SQL Server 2005 Express Edition.

Se creó un usuario en el servidor SQL Server para realizar las conexiones a la base de datos y de esta forma no utilizar el usuario *sa*.

Los pasos que se siguieron fueron:

1.- Se abrió el SQL Server Management Studio Express, dentro del Explorador de objetos se abrió el nodo Seguridad, Inicio de Sesión.

2.- Con el botón derecho del mouse se desplegó el submenú, y se seleccionó el nodo “Nuevo inicio de sesión” (figura 3.2)
.

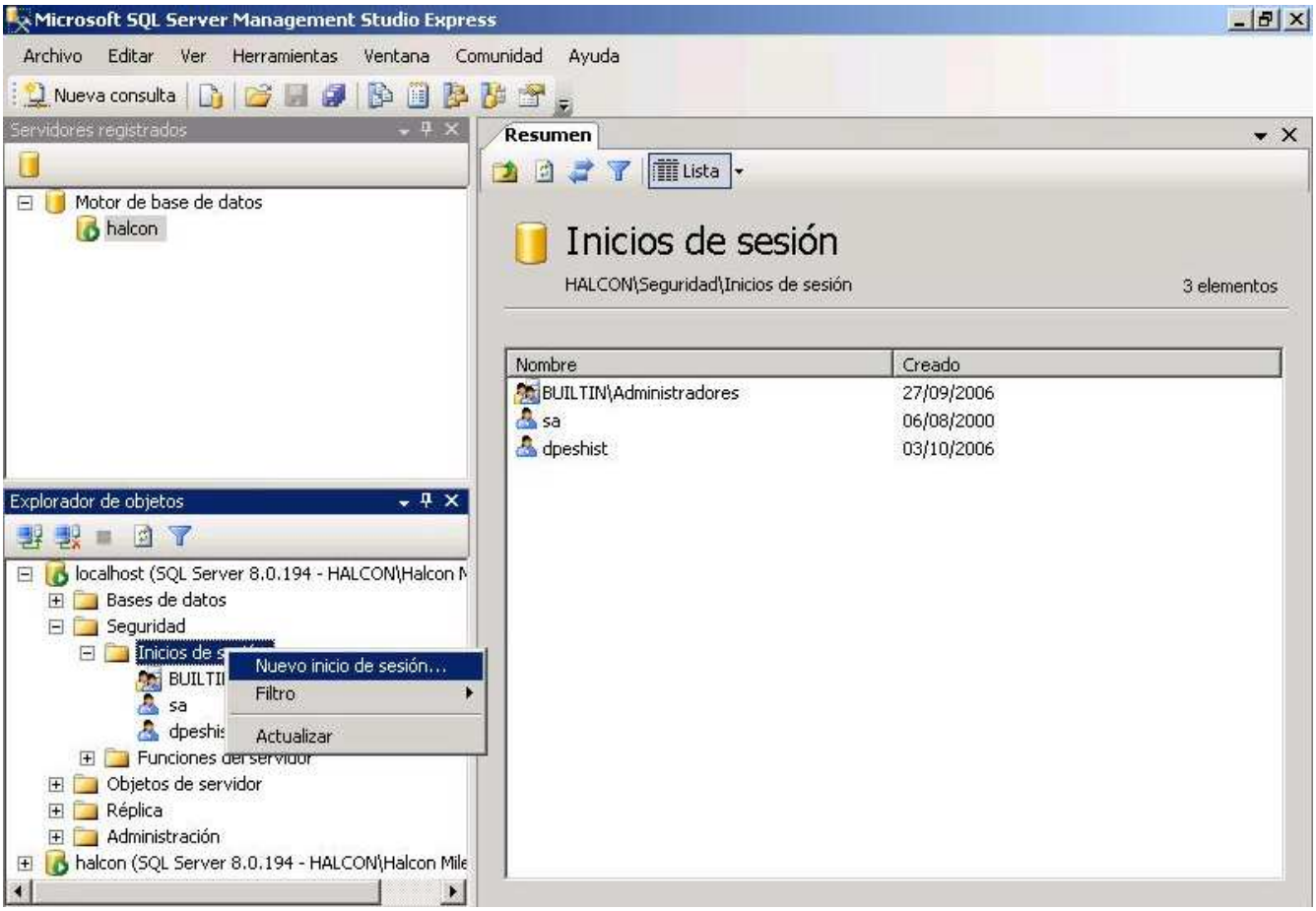

Figura 3.2

3.- En el cuadro de diálogo “Propiedades de inicio de sesión” (Figura 3.3) se mostraron las opciones para crear un usuario.

Los valores que se propusieron fueron:

- Nombre de inicio de sesión: se propuso el nombre de "historicodpes".
- Modo de Autenticación: Se eligió la autenticación de SQL Server.
- Contraseña: se propuso la contraseña "dpesc905hbd"
- Base de datos predeterminada: se seleccionó la base HistoricoBD.

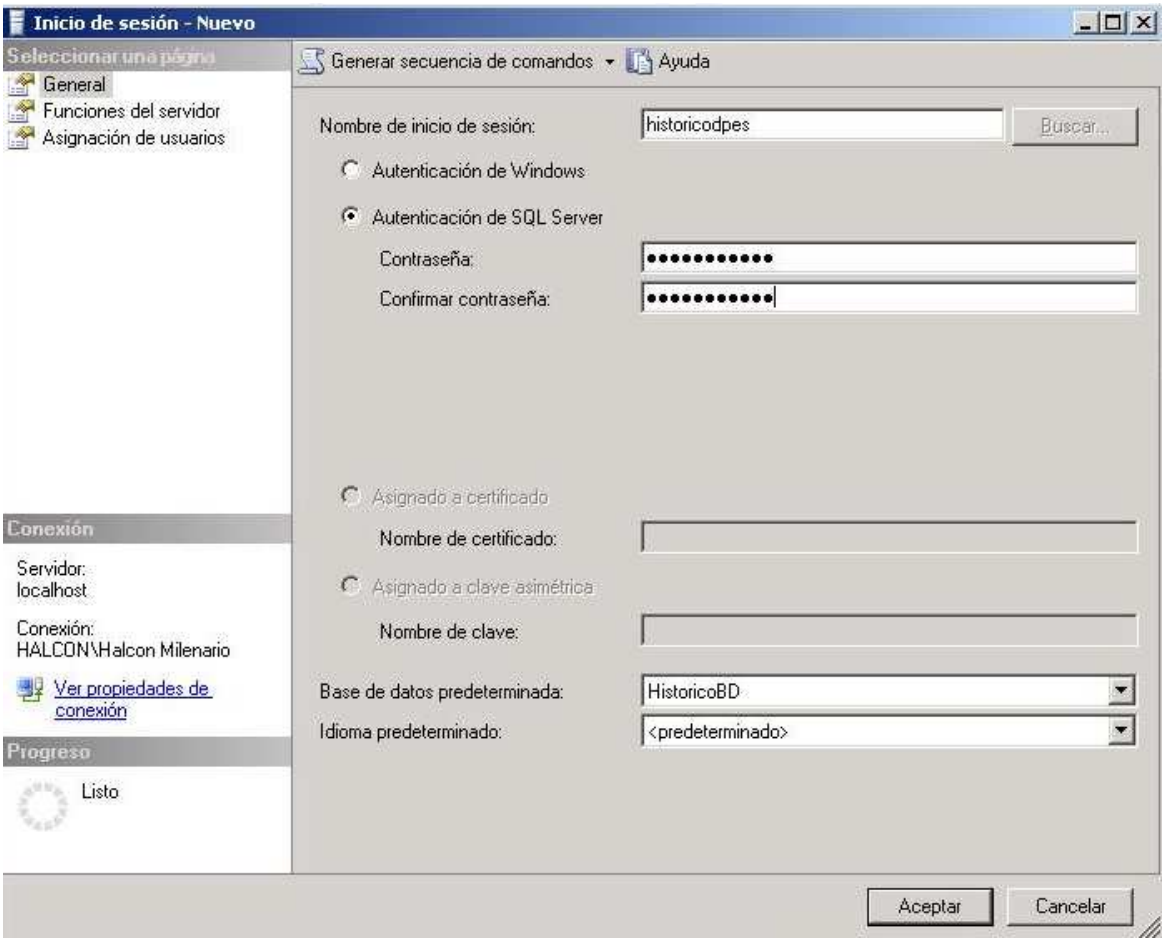

Figura 3.3

4.- Se seleccionó la opción “Asignación de usuarios” (Figura 3.4) en donde se asignó la base HistoricoBD al usuario historicodpes, además se asignó el permiso db_datareader (lectura de datos) para el usuario historicodpes.

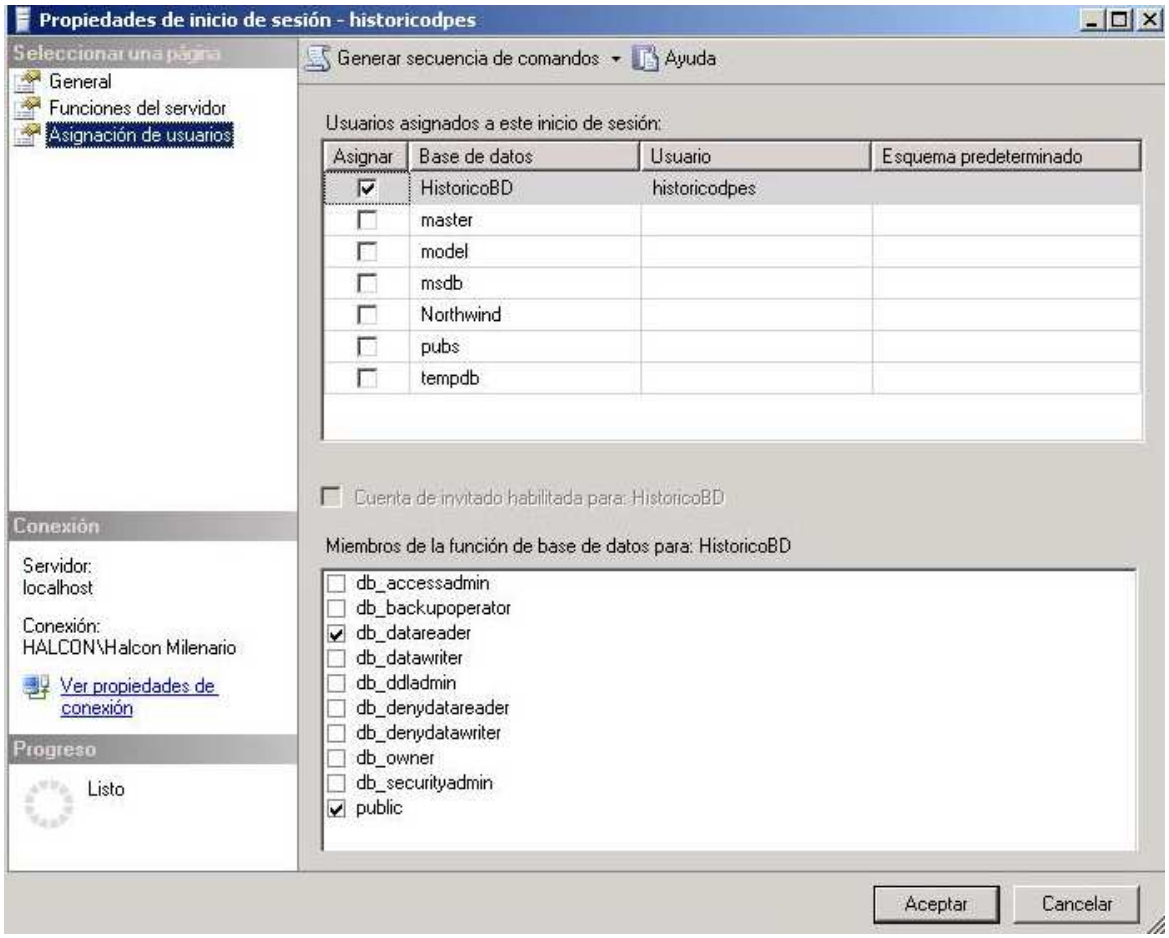

Figura 3.4

Creación de un usuario DSN

Se creó un usuario DSN, el cual se incluyó dentro de la cadena de conexión para efectuar la conexión con el usuario historicodpes.

Se ejecutaron los siguientes pasos:

1.- Se accedió al Panel de Control, Herramientas Administrativas, Orígenes de datos (ODBC), con el botón agregar se desplegó cuadro de diálogo “Crear nuevo origen de datos”. (Figura 3.5).

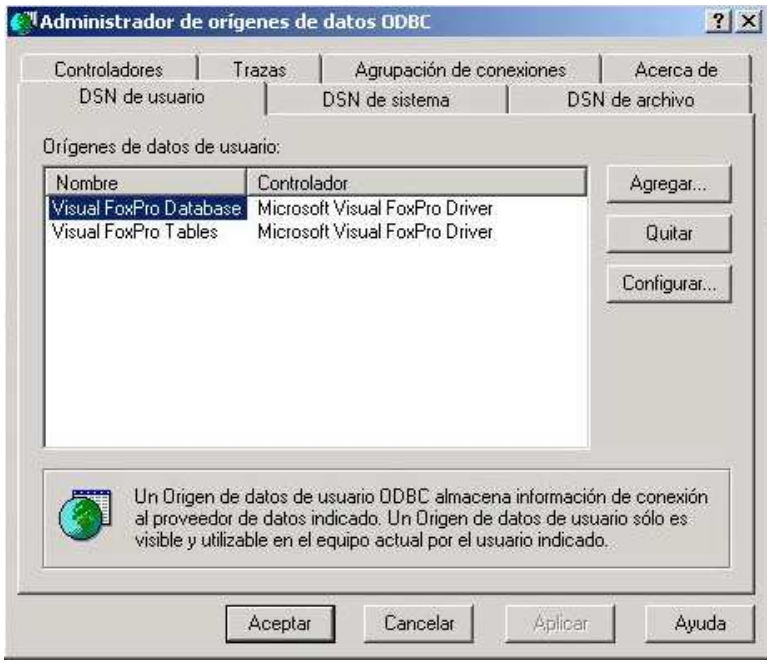

Figura 3.5

2.- De la lista de controladores se seleccionó el controlador SQL Native Client.(Figura 3.6).

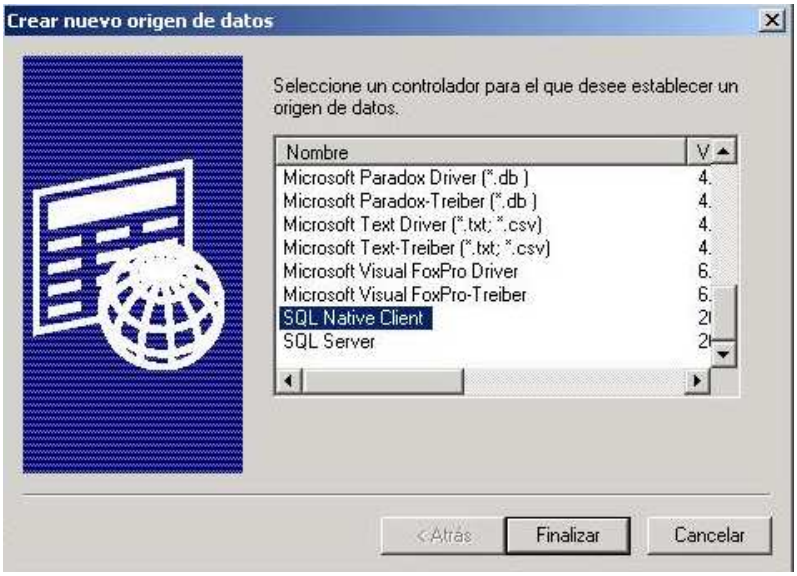

Figura 3.6

En el cuadro de diálogo "Crear un nuevo origen de datos para SQL Server" se definieron las siguientes características (Figura 3.7):

- Nombre: Historicobd
- Descripción: Conexión a la base de datos Histórico BD
- Servidor: localhost

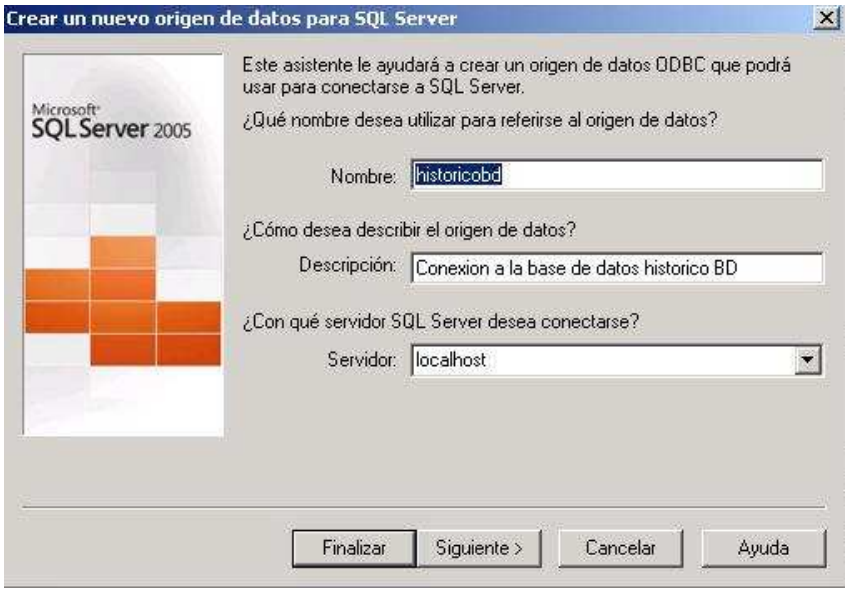

Figura 3.7

En el siguiente cuadro se seleccionó la opción de autenticación “Con la autenticación de SQL Server, mediante un id de inicio de sesión y una contraseña escritos por el usuario”.

En id de inicio de sesión se colocó el usuario que se creó en el manejador SQL Server 2005; y en contraseña se asignó la contraseña que se propuso para este usuario.
Si el usuario fue aceptado, se mostrará el siguiente cuadro de diálogo, de lo contrario aparecerá un mensaje de error de conexión.

En el siguiente cuadro de diálogo, se seleccionó la opción "Establecer la siguiente base de datos como predeterminada" (Figura 3.8).

Figura 3.8

En el último cuadro de diálogo se oprimió el botón finalizar. (Figura 3.9)

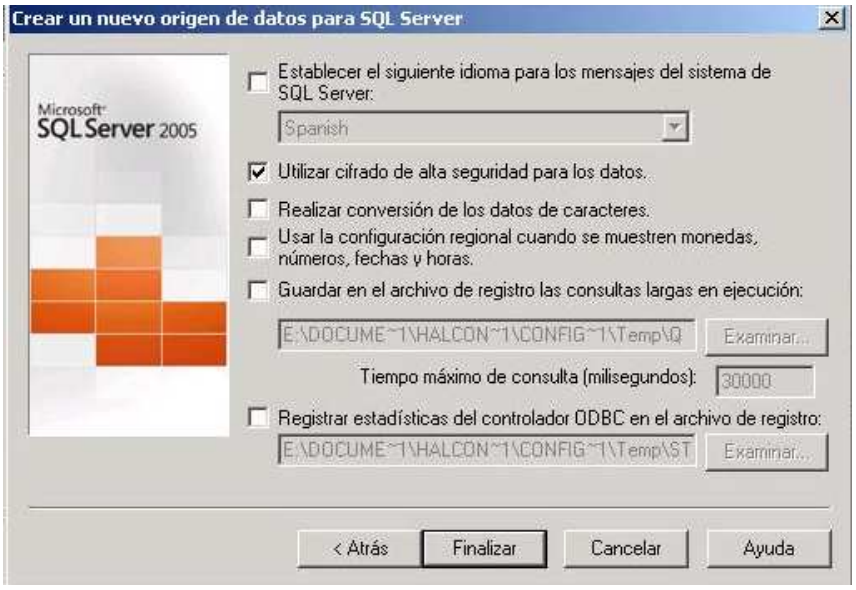

Figura 3.9

Se desplegó un cuadro con un resumen de las propiedades de la conexión (Figura 3.10).

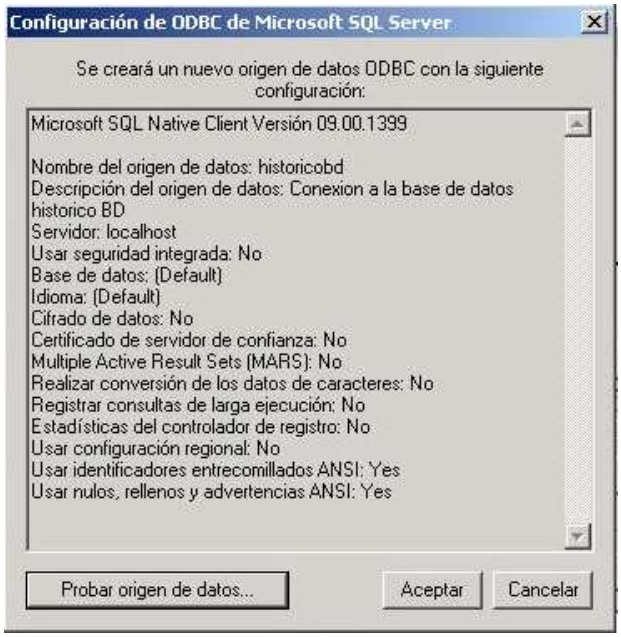

Figura 3.10

Al oprimir el botón "Probar origen de datos“ se mostró el resultado de las pruebas de conectividad". Si no se pudo conectar al servidor, aparecerá un mensaje de error (Figura 3.11).

Figura 3.11

III.3 - Servicios de transformación de datos (DTS) (SQL Server 2000).

Los Servicios de Transformación de Datos (DTS, Data Transformation Services), proporciona una serie de herramientas que permiten transformar, extraer, consolidar datos de múltiples fuentes y copiar los datos a uno o varios destinos.

Los paquetes DTS son conjuntos organizados de conexiones, tareas DTS, transformaciones DTS y restricciones de flujo de trabajo. Se utilizan para almacenar tareas DTS para su reutilización, contienen uno o más pasos que se ejecutan secuencialmente o en paralelo. Los paquetes se transfieren de un servidor a otro para ayudar a automatizar las tareas de mantenimiento.

Elementos de paquetes DTS

Cada paquete contiene uno o más de los siguientes elementos:

Tareas: Cada tarea es un único paso definido en el paquete. Una tarea define un único elemento de trabajo que se realiza como parte del movimiento o transformación de datos o como un trabajo que se va a ejecutar. Algunos ejemplos de tareas comunes son:

- Transformación de datos: selecciona los datos de una conexión fuente, organizar las columnas para un conjunto de transformaciones y enviar datos transformados a la conexión de destino.

- Importar y exportar datos: utilizada para importar datos desde archivos de texto, base de datos Microsoft Access y otras fuentes a bases de datos SQL.

Flujo de trabajo: los pasos DTS y las restricciones de precedencia imponen el orden de los

elementos de trabajo en un paquete DTS. El Diseñador DTS se utiliza para diseñar gráficamente el flujo de trabajo de los paquetes. Los pasos representan las unidades de ejecución de un paquete DTS. Estos pasos definen el orden en el que se ejecutan las tareas cuando se ha ejecutado el paquete.

Conexiones: la finalización con éxito de una tarea DTS se basa en la habilidad para conseguir conectar los datos de origen y los de destino. La arquitectura utilizada por DTS para realizar conexiones es OLE DB, que proporciona acceso a una variedad de fuentes de datos.

Algunas de las fuentes de datos que admiten actualmente una conexión OLE DB son:

- Microsoft SQL Server.
- Microsoft Access.
- Oracle.
- ODBC Data Sources.
- Microsoft Excel.
- HTML.
- Archivos de texto en varios formatos.

Diseñador DTS

El diseñador DTS permite crear paquetes DTS con una amplia funcionalidad implementando gráficamente el Modelo de objetos DTS.

Se accede al Diseñador DTS desde el Administrador corporativo pulsando el botón derecho del ratón sobre el nodo de “Servicio de transformación de Datos” y seleccionando “Nuevo paquete”, o desde el menú Acción seleccionar “Nuevo paquete”. (Ver Figura 3.12).

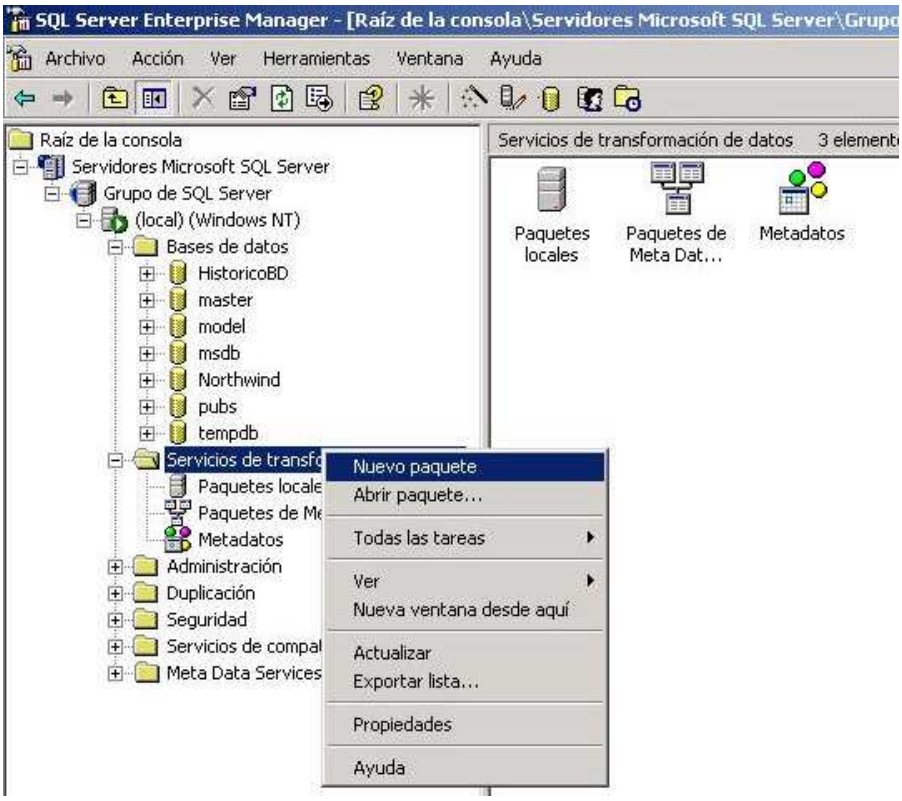

Figura 3.12

El diseñador DTS proporciona una forma de utilizar una interfaz intuitiva para crear y editar paquetes mostrando las tareas disponibles en una paleta. La primer paleta denominada Conexión, permite configurar y establecer las conexiones para el paquete. (Figura 3.13).

La segunda paleta denominada Tareas, contiene las tareas disponibles para agregar al paquete. Cuando se selecciona una tarea, se agrega a la ventana de diseño y se presenta un cuadro de diálogo que permite definir las propiedades de la nueva tarea.
(Figura 3.13).

Figura 3.13

Los iconos que representan las conexiones aparecerán en la hoja de diseño. Asimismo, aparecerán las tareas con sus correspondientes iconos.(Figura 3.13).

Una vez creado el paquete, existen varias opciones para guardarlo, por ejemplo, guardarlo en una tabla de SQL Server, en meta data services de SQL, en un archivo de almacenamiento estructurado o en un archivo de Visual Basic. La opción predeterminada para guardar paquetes es en SQL Server.

Creación de un paquete usando el diseñador DTS

A continuación se da una descripción para crear un paquete DTS con el diseñador DTS:

1.- Al acceder al Administrador Corporativo, se abre el nodo del servidor sobre el que se está trabajando, después se abre el nodo "Servicios de Transformación de Datos".

2.- Para acceder al diseñador DTS, en la opción "Paquetes Locales " se pulsa el botón derecho, en el menú emergente seleccionar "Nuevo paquete". Otra opción es abrir el menú Acción, Nuevo Paquete.

3.- Si se usan archivos de Excel como origen, se selecciona de la paleta Conexión el icono de Excel. Aparecerá el cuadro de diálogo "Propiedades de conexión", el cual agrega una conexión a un origen de datos (Figura 3.14). Dentro de las opciones disponibles se encuentran:

- Nueva conexión: se asigna un nombre a una nueva conexión con el origen de datos.
- Origen de datos: proporciona una lista de proveedores OLE DB. En este caso estará seleccionado el proveedor de Microsoft Excel 97-2000.
- Nombre de archivo: especifica el nombre de archivo y la ruta de acceso del archivo o la base de datos que almacena los datos que se van a importar.

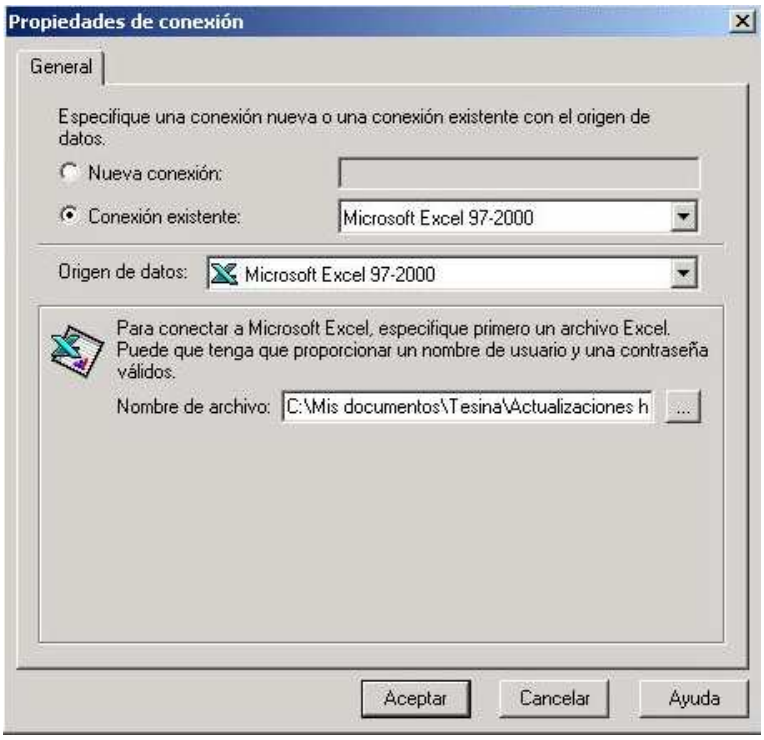

Figura 3.14

Al pulsar el botón aceptar, el icono de Excel aparecerá en la hoja de diseño del diseñador DTS.

4.- De la paleta de conexión, se busca el icono que asocia la conexión con el destino, para este caso se busca el icono de Microsoft OLE DB Provider for SQL Server. Aparecerá nuevamente el cuadro de diálogo "Propiedades de Conexión" (Figura 3.15), el cual es similar al anterior, salvo por las siguientes opciones:

- Servidor: especifica el nombre del servidor que almacena el origen de datos.
- Utilizar autenticación de Windows: especifica que el paquete utilizará autenticación de Windows para iniciar sesiones en una instancia de SQL Server.
- Usar la autenticación de SQL Server: especifica que el paquete utilizará autenticación de SQL Server para iniciar una sesión en una instancia de SQLServer.
- Nombre de usuario: Especifica un nombre de usuario para la conexión de la base de datos.
- Contraseña. especifica una contraseña para la conexión de la base de datos.
- Base de datos: Muestra un listado con las bases de datos del servidor SQL Server especificado, en esta lista buscamos la base de datos de destino de los datos.

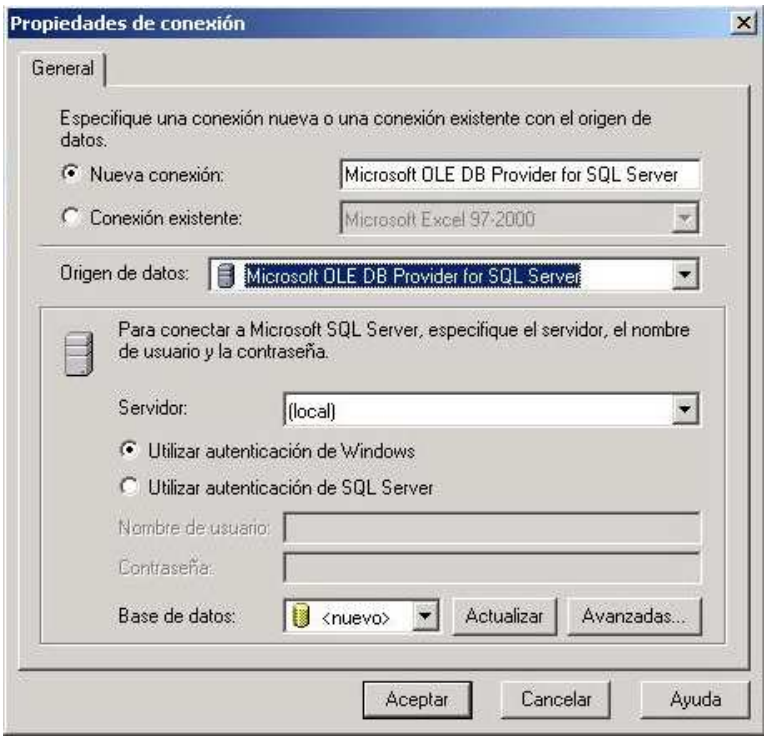

Figura 3.15

El icono de la conexión de SQL Server aparecerá en la hoja de diseño del diseñador DTS.

5.- Se selecciona el icono de Excel, el cual representa el origen de datos; en la paleta "Tarea" se selecciona "Tarea Transformar Datos", representada por una flecha negra sobre un engrane. En la hoja de diseño aparecerá un puntero con el mensaje "Seleccione una conexión de destino". Con este puntero seleccionamos la conexión de SQL Server, con ello se dibuja una flecha que une las dos conexiones. Otro camino es seleccionar directamente la tarea Transformar Datos, al llevar el puntero a la hoja de diseño, aparecerá un mensaje solicitando que se seleccione el origen, y posteriormente se solicita seleccionar el destino.

6.- Sobre la flecha se pulsa el botón derecho del ratón, en el menú emergente se selecciona Propiedades (Figura 3.16). Aparecerá el cuadro de diálogo "Propiedades de la tarea Trasformar Datos", que contiene las opciones para configurar la transformación.

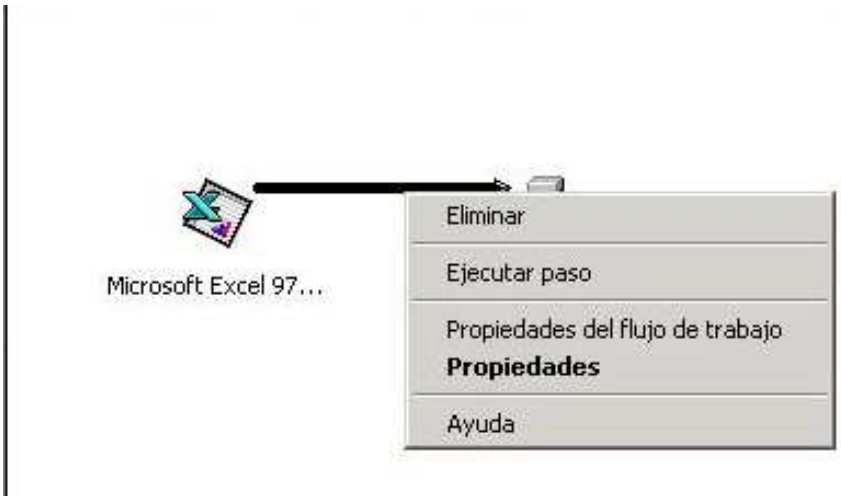

Figura 3.16

En la ficha Origen (Figura 3.17) aparecen las siguientes opciones:

- Descripción: Se escribe una descripción de esta tarea.
- Tabla o vista: Se selecciona una tabla o vista del origen de datos especificado en la conexión. En este caso, es posible seleccionar una hoja especifica del archivo de Excel.
- Consulta SQL: Permite especificar que una instrucción SQL recupere los datos desde el origen de datos.
- Vista previa: Muestra los datos de la tabla de origen seleccionada. En este caso, se visualizan los datos del archivo Excel (Figura 3.18).

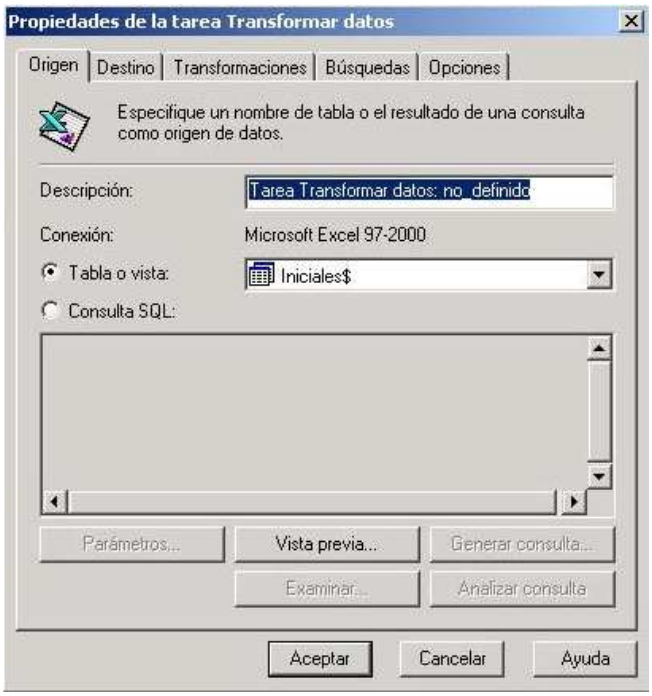

Figura 3.17

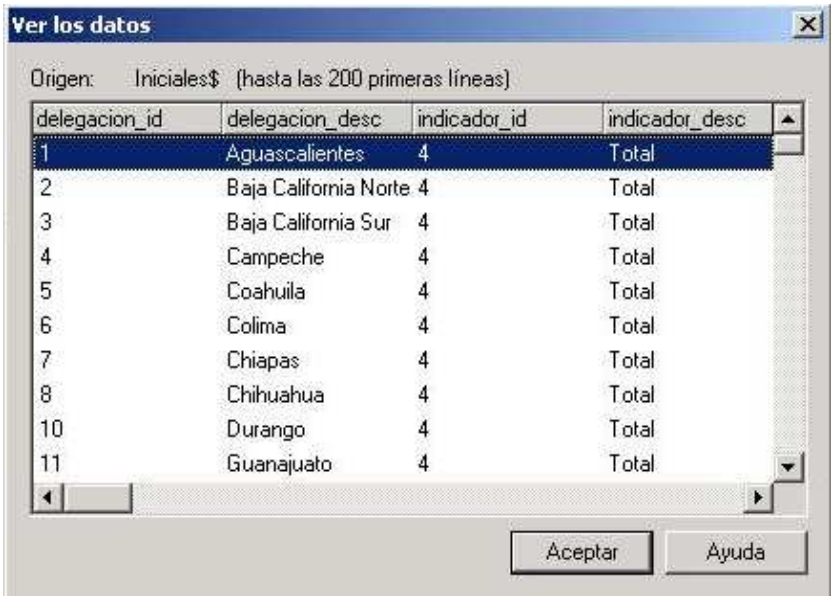

Figura 3.18

En la ficha Destino se configura la tabla de destino o la ubicación de almacenamiento para los resultados de la tarea. Las opciones de esta ficha son:

Nombre de la tabla: permite seleccionar de la lista el destino de los datos de la transformación.

Crear: Especifica que se crea como destino una nueva tabla. Al pulsar sobre el botón crear, aparece un cuadro de diálogo con instrucciones SQL para crear la tabla.

Figura 3.19

La instrucción CREATE TABLE (Figura 3.19) permite definir el nombre de la tabla, el cual va entre las llaves "[" y "]". Enseguida se define el nombres, el tipo, el tamaño y si acepta valores Null:

- Nombre: muestra los nombres de las columnas de destino.
- Tipo: muestra el tipo de datos nativo del destino mediante asignaciones de tipos de datos OLE DB.
- Aceptar Null: Permite ver si las columnas de destino admiten valores NULL.
- Tamaño: Muestra el ancho de las columnas de destino, donde se pueda aplicar.

En la ficha Transformaciones se selecciona y asigna gráficamente las columnas de origen y de destino que forman cada transformación. Las asignaciones de columnas se representan en esta ficha mediante flechas que conectan columnas de los cuadros Origen y Destino. De forma predeterminada, la tarea Transformar datos asigna todas las columnas mediante asignaciones de columna en relación una a una. Se usará esta ficha para verificar que cada nombre de columna del origen le corresponda un nombre de columna similar en el destino.

Las opciones que presenta esta ficha son:

- Nombre: Muestra el nombre de la transformación seleccionada. Presenta un listado que contiene los nombres de la transformación; si no se hace, se mostrará <ninguno seleccionado>. De manera predeterminada, se asigna a las transformaciones el nombre DTS_Transformation_n, donde n es la posición ordinal de la transformación asignada.
- Tipo: Muestra el tipo de la transformación de columna para la asignación seleccionada. Si se crea una transformación nueva, este valor se omite.
- Nuevo: Muestra el cuadro de diálogo “Crear una transformación nueva”, en el que se selecciona el tipo de transformación de columna.
- Eliminar: Elimina una transformación seleccionada del listado de transformaciones.
- Origen: Muestra gráficamente las columnas de los datos de origen.
- Destino: Muestra gráficamente las columnas de los datos de destino.
- Seleccionar todo: Selecciona todas las columnas de origen y todas las columnas de destino.
- Eliminar todo: Elimina todas las transformaciones.

6.- Una vez configurada la transformación, se selecciona la flecha de Transformación de Datos usando el botón derecho del ratón; en el menú emergente se selecciona la opción "Ejecutar paso", en caso de que no existan problemas, aparecerá un cuadro de diálogo avisando que se ha efectuado correctamente la transformación.

Para corroborar que los datos han sido migrados, se busca dentro de la base de datos de destino el nombre de la tabla creada por el DTS, al abrirla se verifica si los datos han sido migrados correctamente, o si ha surgido algún problema durante la migración.

La siguiente y última parte, se presenta el desarrollo de la página web, empleando las páginas Active Server Pages, el lenguaje XML y el manejador de bases de datos SQL Server (versiones 2000 y 2005 Express Edition).

REFERENCIAS DE LA PARTE III

LIBROS CONSULTADOS

1.- Dalton Patrick, Whitehead Paul. **LA BIBLIA DE SQL SERVER 2000.** Ediciones Anaya Multimedia. España 2002.

2.- Pérez López, César. **DOMINE MICROSOFT SQL SERVER 2000. ADMINISTRACIÓN Y ANÁLISIS DE BASES DE DATOS.** Alfaomega Ra-Ma. España.

PÁGINAS DE INTERNET CONSULTADAS

Comparación de características de SQL 2005.
a) http://www.microsoft.com/spain/sql/productinfo/features/compare-features.mspx
2006. Microsoft Corporation.

Descarga de libros en pantalla de SQL Server 2005 Express Edition.
b) http://www.microsoft.com/downloads/details.aspx?familyid=BE6A2C5D-00DF-4220-B133-29C1E0B6585F&displaylang=es.
2006. Microsoft Corporation.

Descarga de SQL Server 2005 Express Edition.
c) http://www.microsoft.com/downloads/details.aspx?familyid=220549B5-0B07-4448-8848-DCC397514B41&displaylang=es.
2006. Microsoft Corporation.

Información de SQL Server 2005.
d) http://www.microsoft.com/latam/technet/productos/servers/sql/2005/default.mspx.
2006. Microsoft Corporation.

Información de SQL Server 2005 Express Edition
e) https://www.microsoft.com/spanish/msdn/articulos/archivo/261205/voices/sseoverview.mspx #E2FAC
Fecha de publicación: 26 de Diciembre de 2005.
Microsoft Corporation.

Instalación de SQL Server 2005.
f) http://msdn2.microsoft.com/es-es/library/ms143719.aspx
Ultima actualización: 5 de diciembre de 2005.
Microsoft Corporation.

Panorama General de SQL Server 2005 Express Edition.
g) http://www.microsoft.com/spanish/msdn/articulos/archivo/100904/voices/10PanoramaSQLServer2005ExpressEdition.asp
Autor: Rajesh George
Junio 2004
Microsoft Corporation.

Requisitos de SQL Server 2005 (Todas las versions).
h) http://msdn2.microsoft.com/es-es/library/ms143506.aspx
Ultima actualización: 5 de diciembre de 2005.
Microsoft Corporation.

Requisitos de SQL Server 2005 Express Edition.
i) http://download.microsoft.com/download/a/2/3/a23083ba-88d4-4e89-b9fb-dfd3b618bbdb/RequirementsSQLEXP2005.htm.
2006. Microsoft Corporation.

SQL Server Management Studio.
k) http://msdn2.microsoft.com/es-es/library/ms165690.aspx
2006 Microsoft Corporation.

IV. Creación de una página Web

IV. Creación de una página Web de consulta de información histórica

IV.1 - Contexto de la página web

La página web que se presenta como aplicación, esta dirigida al personal normativo de la Dirección de Prestaciones Económicas y Sociales del Instituto Mexicano del Seguro Social.

Antes de presentar el diseño de la página, se hará una presentación del Instituto Mexicano del Seguro Social, así como de la Dirección de Prestaciones Económicas y Sociales.

Dentro de las funciones primordiales del estado mexicano se encuentra la seguridad social. El Instituto Mexicano del Seguro Social se ha constituido como la principal Institución del Sistema de Seguridad Social en México.

El Instituto esta dividido en varias direcciones, cada una presta servicios de acuerdo a la función asignada. Una de estas direcciones es la Dirección de Prestaciones Económicas y Sociales.

El organigrama completo del Seguro Social se encuentra en la siguiente dirección: *http://www.imss.gob.mx/NR/rdonlyres/DCE558B7-E500-4D89-B308-9037C76ECFA3/0/org_dpes_06_2006.pdf*

Debido a que la Dirección de Prestaciones Económicas y Sociales no contaba con un lugar en donde se concentre la información histórica referente a los Certificados de Incapacidad Subsidiados por Ramo de Seguro Tramitados de los Indicadores de Prestaciones Económicas, se requirió contar con un apartado en donde la información se encuentre disponible y centralizada.

Para ello, se desarrolló una página web dentro de la intranet de la Dirección de Prestaciones Económicas y Sociales, esta página es un filtro de información histórica, donde el usuario hace consultas sobre la información de los Certificados de Incapacidad Subsidiados por Ramo de Seguro Tramitados, seleccionando las opciones que la página presenta.

Esta página web maneja información referente a las prestaciones denominadas subsidios, los cuales se dividen en:

- Riesgo de Trabajo
- Enfermedad General
- Maternidad.

IV.1.1 - Misión del Instituto Mexicano del Seguro Social

La Misión del Instituto Mexicano del Seguro Social es otorgar a los trabajadores mexicanos y a sus familias la protección suficiente y oportuna ante contingencias tales como la enfermedad, la invalidez, la vejez o la muerte.

La protección se extiende no sólo a la salud, sino también a los medios de subsistencia, cuando la enfermedad impide que el trabajador continúe ejerciendo su actividad productiva, ya sea de forma temporal o permanente.

IV.1.2 - Dirección General

El artículo 267 de la Ley del Seguro Social, establece la figura del Director General. El artículo se menciona como sigue:

Artículo 267.- El Director General será nombrado por el Presidente de la República debiendo ser mexicano por nacimiento que no adquiera otra nacionalidad y estar en pleno goce y ejercicio de sus derechos civiles y políticos. (Artículo reformado publicado en el Diario Oficial de la Federación 23-01-1998).

En el artículo 268, menciona las atribuciones del Director General, algunas de sus atribuciones son:

- Presidir las sesiones de la Asamblea General y del Consejo Técnico.
- Ejecutar los acuerdos del propio Consejo.
- Presentar anualmente al Consejo Técnico el balance contable y el estado de ingresos y gastos.

IV.1.3 - Honorable Consejo Técnico

En el artículo 263 se define al Honorable Consejo Técnico, un fragmento de este artículo se menciona a continuación:

Artículo 263. El Consejo Técnico es el órgano de gobierno, representante legal y el administrador del Instituto y estará integrado hasta por doce miembros, correspondiendo designar cuatro de ellos a los representantes patronales en la Asamblea General, cuatro a los representantes de los trabajadores y cuatro a los representantes del Estado, con sus respectivos suplentes y el Ejecutivo Federal cuando lo estime conveniente, podrá disminuir a la mitad la representación estatal.

El artículo 264 de la Ley del Seguro Social describe las atribuciones del Consejo Técnico, algunas de estas son:

- Decidir sobre las inversiones de las reservas y demás recursos del Instituto, con sujeción a lo previsto en esta Ley y sus reglamentos, excepto los provenientes del seguro de retiro, cesantía en edad avanzada y vejez.
- Convocar a Asamblea General ordinaria o extraordinaria.
- Discutir y aprobar el proyecto de presupuesto de ingresos y egresos del Instituto que someta a su consideración el Director General, así como autorizar adecuaciones al presupuesto aprobado. (Fracción reformada publicada en el Diario Oficial de la Federación 20-12-2001).

IV.1.4 - División geográfica de la República Mexicana

El Instituto mexicano del Seguro Social, ha divido a la Republica mexicana en 37 delegaciones.

IV.2 - Dirección de Prestaciones Económicas y Sociales

IV.2.1 - Creación de la Dirección de Prestaciones Económicas y Sociales

El 11 de enero de 1995, el Consejo Técnico autorizó la creación de la Dirección de Prestaciones Económicas y Sociales.

La Dirección de Prestaciones Económicas y Sociales se constituye en el órgano encargado de establecer y aplicar acciones normativas, de asesoría, de apoyo técnico y para la instrumentación de la evaluación en el ámbito nacional, con el propósito de que los procesos operativos para el otorgamiento y control de las prestaciones económicas, servicios de guardería, de prestaciones sociales y los servicios de ingreso que estipula la Ley y sus reglamentos, se realicen con oportunidad, eficiencia y sentido humano.

Las prestaciones económicas y sociales cobran especial relevancia en el contexto de la Seguridad Social Integral, por que con estos servicios se busca:

- Proteger los medios de subsistencia.
- Fomentar la salud, prevenir enfermedades y accidentes.
- Contribuir a los niveles de vida de la población derechohabiente.

IV.2.2 - Objetivo y marco jurídico de la Dirección de Prestaciones Económicas y Sociales

El objetivo de la Dirección de Prestaciones Económicas y Sociales es la de proporcionar en el ámbito nacional, los servicios que otorga el Instituto relativos a las prestaciones económicas, guarderías, prestaciones sociales y servicios de ingreso, conforme a las disposiciones que al respecto establece la Ley del Seguro Social y reglamentos.

Son muchas leyes las que componen la base legal que sustenta a la Dirección de Prestaciones Económicas y Sociales, estas leyes se pueden consultar en el capítulo 4 del *Manual de Organización de la Dirección de Prestaciones Económicas y Sociales.*

Esta manual se encuentra disponible en formato pdf en la siguiente dirección:
http://www.imss.gob.mx/NR/rdonlyres/2A59A4B6-A756-4195-9DA6-AB057A4DAF35/0/manual_pes_2003_09.pdf.

IV.2.3 - Estructura de la Dirección de Prestaciones Económicas y Sociales

La Dirección de Prestaciones Económicas y Sociales tiene dependencia directa de la Dirección General, contando con una estructura normativa integrada por cuatro Coordinaciones, en las que se delimitan las responsabilidades y funciones para normar, evaluar y controlar el otorgamiento de los servicios a su cargo. Las Coordinaciones son:

- Coordinación de Prestaciones Económicas.
- Coordinación de Guarderías.
- Coordinación de Prestaciones Sociales.
- Coordinación de Servicios de Ingreso.

A continuación se presenta el organigrama de la Dirección de Prestaciones Económicas y Sociales.

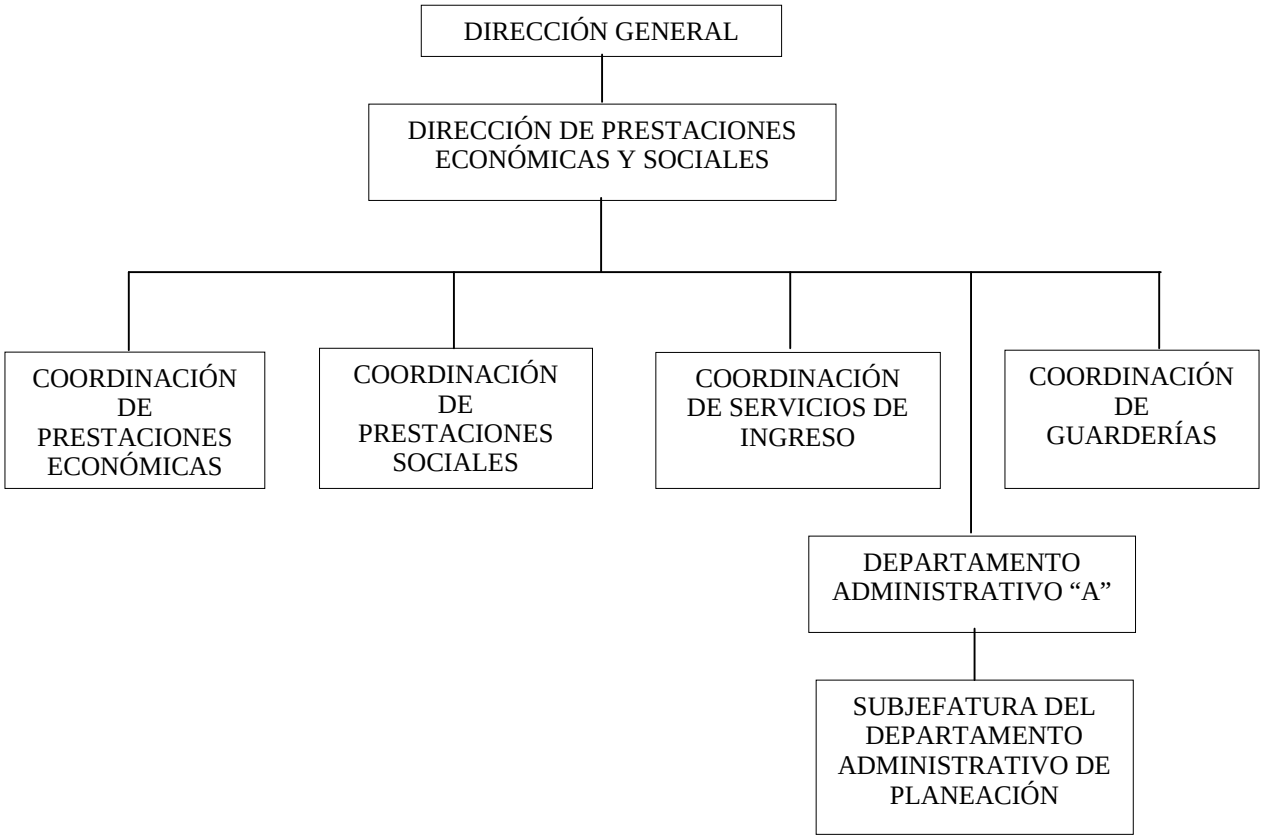

Figura 4.1

IV.2.4 - Políticas de la Dirección de Prestaciones Económicas y Sociales

Dentro de las políticas de la Dirección de Prestaciones Económicas y Sociales se encuentran:

- Dará cumplimiento a los lineamientos y disposiciones gubernamentales e institucionales que se emitan en materia de servicios de prestaciones económicas, guarderías, prestaciones sociales, de distribución o comercialización de bienes e imagen institucional.

- Emitirá la normatividad para la organización y funcionamiento de los servicios de

prestaciones económicas, guarderías, prestaciones sociales y de servicios de ingreso.

- Informará de manera permanente a la Dirección General el desarrollo y avance de los planes y programas a su cargo.

IV.2.5 - Atribuciones y funciones de la Dirección de Prestaciones Económica y Sociales

En el Artículo 77 del Reglamento de Organización Interna del Instituto Mexicano del Seguro Social, se especifican las atribuciones de la Dirección de Prestaciones Económicas y Sociales, algunas de estas son:

- Planear, dirigir y normar las acciones relacionadas con el otorgamiento de las prestaciones en dinero, servicio de guarderías, prestaciones sociales institucionales y otros servicios y prestaciones a su cargo, comprendiendo los ingresos de dichas prestaciones y servicios, y en su caso, evaluar sus resultados.

- Emitir lineamientos para la administración de los ingresos y recursos que se generen en las unidades operativas de prestaciones sociales institucionales y en las de otros servicios y prestaciones a su cargo.

- Dictar disposiciones, lineamientos y criterios de observancia general y obligatoria para las instancias y unidades operativas de prestaciones económicas, guarderías, prestaciones sociales institucionales, así como de otras prestaciones a su cargo, y regular la recopilación, evaluación y sistematización de la información que al respecto se genere.

Dentro de las funciones de la Dirección de Prestaciones Económicas y Sociales se encuentran:

- Planear y dirigir las acciones para el otorgamiento de las prestaciones económicas, servicio de guarderías, prestaciones sociales y servicios de ingreso, que garanticen el bienestar individual y colectivo de la población derechohabiente del Instituto Mexicano del Seguro Social.

- Proponer al C. Director General, los proyectos de iniciativas o reformas de leyes, reglamentos, decretos y acuerdos en materia de subsidios y ayudas, pensiones y rentas vitalicias; otorgamiento y expansión del servicio de guarderías; bienestar social, desarrollo cultural, deporte y cultura física, adiestramiento técnico y capacitación para el trabajo.

- Establecer la norma, criterios, políticas y lineamientos para regular la administración de los ingresos y recursos que se generen en las unidades operativas de prestaciones sociales y de servicios de ingreso.

IV.2.6 - Prestaciones Económicas

En la Ley del Seguro Social, se establece el otorgamiento de prestaciones económicas, las cuales son pensiones, subsidios y ayudas.

Las prestaciones económicas se dividen en tres grupos:

- Pensiones: total o parcial, invalidez, viudez, orfandad, ascendientes, vejez, cesantía en edad avanzada.
- Subsidios: Riesgos de trabajo, Enfermedad general, Maternidad.
- Ayudas: Gastos de funeral, gastos de matrimonio.

Subsidios
Los subsidios constituyen la prestación económica que se otorga al asegurado inhabilitado para trabajar a consecuencia de una incapacidad temporal, derivada de un riesgo de trabajo, enfermedades o accidentes no profesionales.

Certificado de Incapacidad
El certificado de incapacidad temporal para el trabajo, es el documento médico legal que expide en los formatos oficiales, el médico del Instituto al asegurado para hacer constar la incapacidad temporal para el trabajo.

Los certificados de incapacidad se dividen en 2 tipos:
- Certificados iniciales: son los que se expiden por primera vez.
- Certificados subsecuentes: son los certificados de incapacidad que se expiden después del certificado inicial o de primera vez, así como todos los siguientes que se expidan.

El Instituto divide a los trabajadores en 2 grupos:
- Trabajadores que trabajan en el Instituto (Trabajadores IMSS).
- Trabajadores que no pertenecen al Instituto (Trabajadores no IMSS).

Descripción de subsidios

Subsidio por Riesgo de Trabajo
Es la prestación en dinero que se otorga al asegurado imposibilitado para trabajar por un riesgo de trabajo (accidentes y enfermedades a consecuencia del trabajo), en sustitución del salario y se otorga por el 100% del ingreso del trabajador a la fecha del siniestro, a partir del primer día de la incapacidad y durante 52 semanas máximo.

Subsidio por Enfermedad
Es la prestación en dinero que se otorga al asegurado imposibilitado para trabajar por enfermedades o accidentes no profesionales, en sustitución del salario y se paga por el 60% del último salario diario de cotización, a partir del cuarto día del inicio de la incapacidad y hasta 52 semanas, previo dictamen médico cabe la posibilidad de prorroga por 26 semanas adicionales como máximo.

Subsidio por Maternidad
Es la prestación en dinero que se otorga a la asegurada imposibilitada para trabajar por embarazo, en sustitución del salario, y se paga por el 100% del ultimo salario diario cotizado, durante los 42 días previos al parto y 42 días posteriores al mismo(prenatal y postnatal) en los que por prescripción médica y legal descansar la trabajadora.

IV.3 - Creación de la página web

IV.3.1 - Antecedentes

A través de la intranet del Instituto Mexicano del Seguro Social se accede a la intranet de la Dirección de Prestaciones Económicas y Sociales. En la intranet de la Dirección de Prestaciones Económicas y Sociales, se encuentra el apartado denominado "Comunidad de la Dirección de Prestaciones Económicas y Sociales". En esta comunidad se almacena información sobre las actividades de la Dirección de Prestaciones Económicas y Sociales. Esta comunidad solo es accesible dentro de la intranet del Instituto Mexicano del Seguro Social

En la página de inicio de la Comunidad de la Dirección de Prestaciones Económicas y Sociales, se encuentra el enlace llamado Información Estadística. Este enlace lleva a un archivo de Microsoft Excel, el cual contiene la información histórica referente a prestaciones de pensiones, ayudas, subsidios, importes de subsidios, importes de pensiones e importes de ayudas.

Los datos que se almacenan en este archivo no están adecuadamente ordenados, por lo que el usuario tiene que localizar los datos entre las hojas del libro de Excel. Al encontrar los datos, el usuario realiza una copia y pega la información en otra hoja de Excel, hasta completar la información. Este proceso consume mucho tiempo, y la búsqueda de los datos resulta tediosa y complicada. Con el avance del tiempo, este archivo crece; lo que dificulta aun más su manejo.

La Dirección de Prestaciones Económicas y Sociales consideró necesario el desarrollo de un apartado dentro de su intranet, con el propósito de presentar la información histórica de una forma mas amigable, accesible y rápida.

El objetivo de que sea página web consiste en que a futuro no solo el personal normativo de la DPES tenga acceso a la red, sino además, que delegados y personal normativo de otros estados de la República Mexicana tengan acceso a esta información, accediendo a través del portal del IMSS.

Además, la DPES considera que el público en general tenga acceso a esta información, esto se realizará posteriormente, por que se tienen que desarrollar otros módulos con otro tipo de información.

Previamente se desarrolló una página ASP, en la cual se presentan varias opciones de consulta de información histórica. A continuación se da una breve presentación de estas páginas:

La página se denominó Indicadores.asp. Esta página contiene las siguientes opciones:

1.- Certificados de incapacidad subsidiados por ramo de seguro (Trabajadores No IMSS).
2.- Certificados de Incapacidad que no produjeron subsidio, por ramo de seguro.
3.- Ayudas e indemnizaciones.
4.- Pensiones.
5.- Derechohabientes, Asegurados y Pensionados.
6.- Indicadores para el control y evaluación de programas y procesos generales.

La siguiente imagen presenta la página Indicadores.asp.

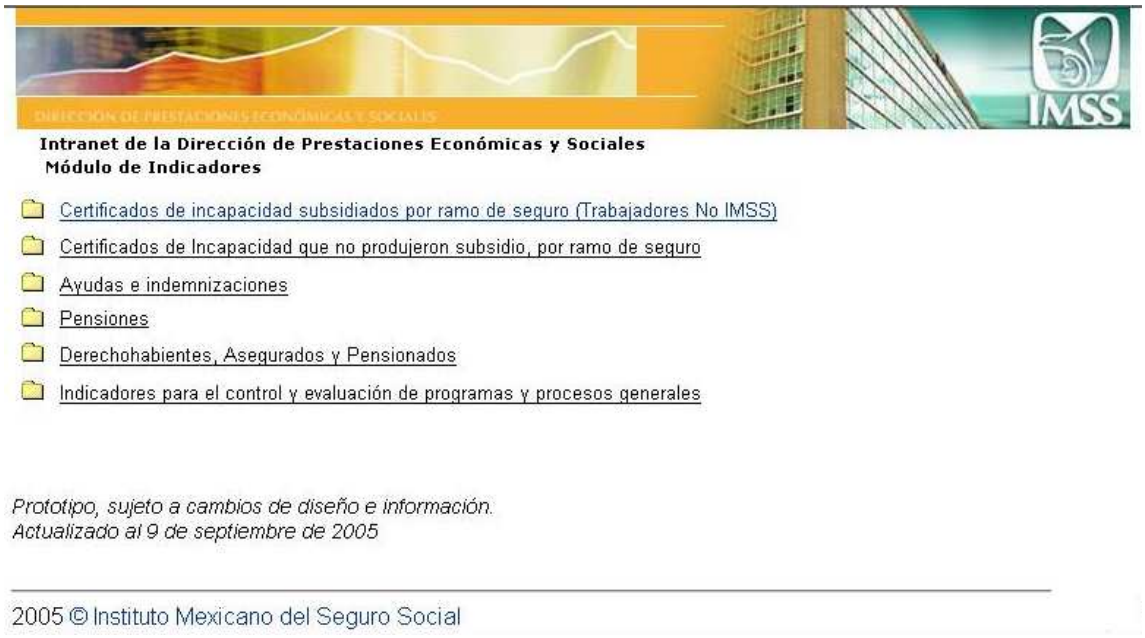

Figura 4.2

La página web de consulta que se desarrolló, se encuentra ubicada en el indicador Certificados de incapacidad subsidiados por ramo de seguro (Trabajadores No IMSS), la página de filtro se denominó filtro.asp, la cual contiene las opciones que el usuario selecciona para consultar la información que requiera.

Diagrama de Flujo de Datos para la página web

La función del siguiente diagrama es la de representar las actividades y los procesos que realizara la página web.

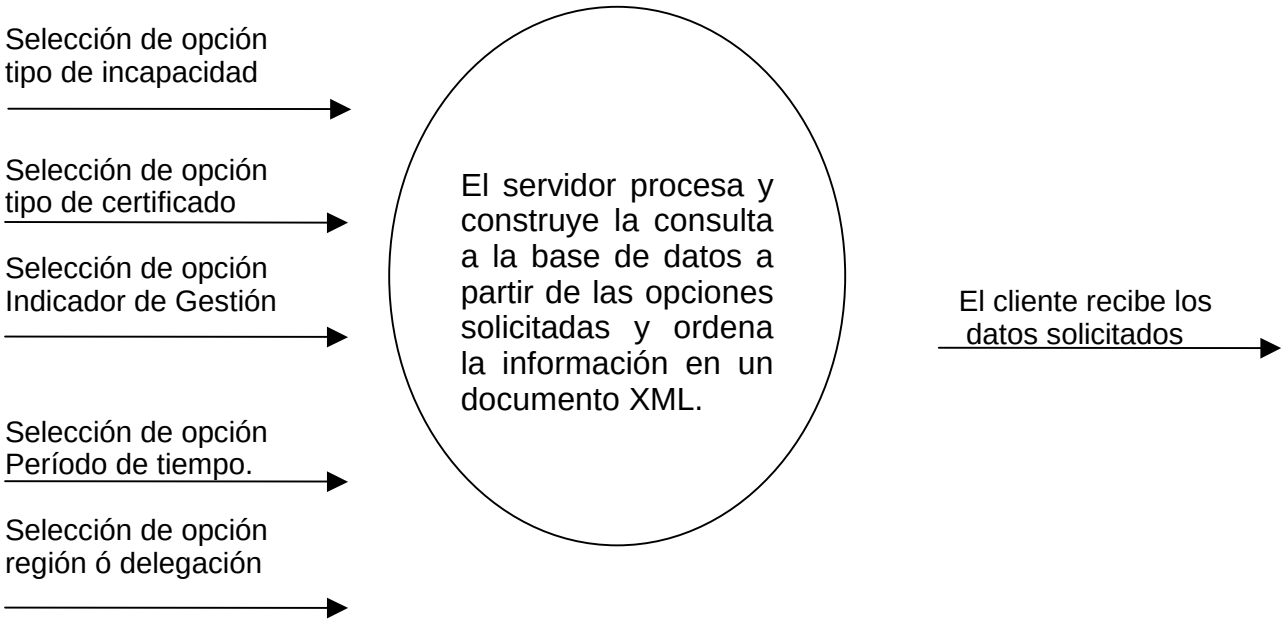

Figura 4.3

Requerimientos de la página web

El personal normativo de la Dirección de Prestaciones Económicas y Sociales desea realizar consultas sobre los datos de acuerdo a los siguientes criterios:

La página contendrá un cuadro con opciones de selección para el siguientes Indicador de Gestión:

- Numero de certificados expedidos a nivel Delegación con periodo anual.

Tipo de incapacidad: los tipos de incapacidad que seleccionar el usuario son:

- Riesgos de Trabajo.
- Enfermedad General.
- Maternidad.

Tipo de certificado: los tipos de certificado disponibles son:

- Certificados Iniciales.
- Certificados Subsecuentes.

Para el Indicador de Gestión Número de certificados expedidos a nivel Delegación con periodo anual se tienen disponibles dos listados. El primer listado contiene el nombre de las

delegaciones que integran el territorio nacional. El segundo listado contiene los años.
Al momento de desarrollar la página web, se contaba con información de los Indicadores de Gestión comprendida en el siguiente período:

- Indicador de Gestión a nivel Delegación: del año 1999 hasta Junio del 2005.

Sin embargo, no existe inconveniente si se desea agregar información reciente o hacer alguna modificación a la misma.

Además, se tiene en consideración el desarrollo de la página web, por lo que las opciones de esta página web aumentaran con el paso del tiempo.

Presentación de resultados

El usuario al realizar su consulta, solo escoge un tipo de certificado. En todas las demás opciones, el usuario elige una o mas opciones.

La presentación del Indicador de Gestión Número de certificados expedidos a nivel Delegación con periodo anual es la siguiente:
En una columna, se presenta el nombre de las Delegaciones que el usuario solicite.
Los datos serán agrupados de acuerdo a su tipo de incapacidad y además, estos son ordenados en columnas de acuerdo al año o años solicitados por el usuario.

En la Figura 4.4 se presenta el diseño de la presentación de resultados del Indicador de Gestión Número de certificados a nivel Delegación con período anual:

	Incapacidad 1				**Incapacidad 2**			
Delegación	**año 1**	**Año 2**	**Año 3**	**año 4**	**año 1**	**año 2**	**año 3**	**año 4**
Delegación 1	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad
Delegación 2	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad
Delegación 3	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad
Delegación 4	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad	cantidad

Figura 4.4

IV.3.2 –Diseño de la base de datos

La Dirección de Prestaciones Económicas y Sociales, a través del Área de Recursos Informáticos, proporcionó la información de los certificados expedidos de cada Indicador de Gestión, esta información fue recibida en hojas de cálculo de Excel.

Una solución rápida es la de acomodar los datos dentro de una nueva hoja de Excel, migrarlos y construir las base de datos directamente. Esta acción causa errores de inconsistencia a los datos, debido a que se tiene considerado ir aumentando las opciones de consulta dentro de la página web, además de agregar nueva información a la base de datos.

El primer paso es la elaboración del diagrama entidad-relación, el cual esta basado en el modelo entidad / vínculo que introdujo Chen en 1976, con base en este diagrama se construirán las tablas en donde se almacenará la información. Para ello se creó un nuevo

archivo de Excel, en donde se organizó los datos; los datos ordenados se migraron al servidor SQL Server mediante el Servicio de Transformación de Datos de SQL Server, y al final se elaborarán las vistas que usará la página web para realizar las consultas.

El modelo entidad-relación esta compuesto por entidades, relaciones y atributos. Cada entidad tiene atributos, y las tablas están relacionadas con otras tablas. Las entidades darán origen a las tablas en el manejador de la base de datos; los atributos serán las columnas de la entidad y las relaciones serán las llaves foráneas que estarán conectadas a otras tablas.

En la Figura 4.7 se presenta el modelo entidad-relación, basado en el modelo entidad / vínculo de Chen.

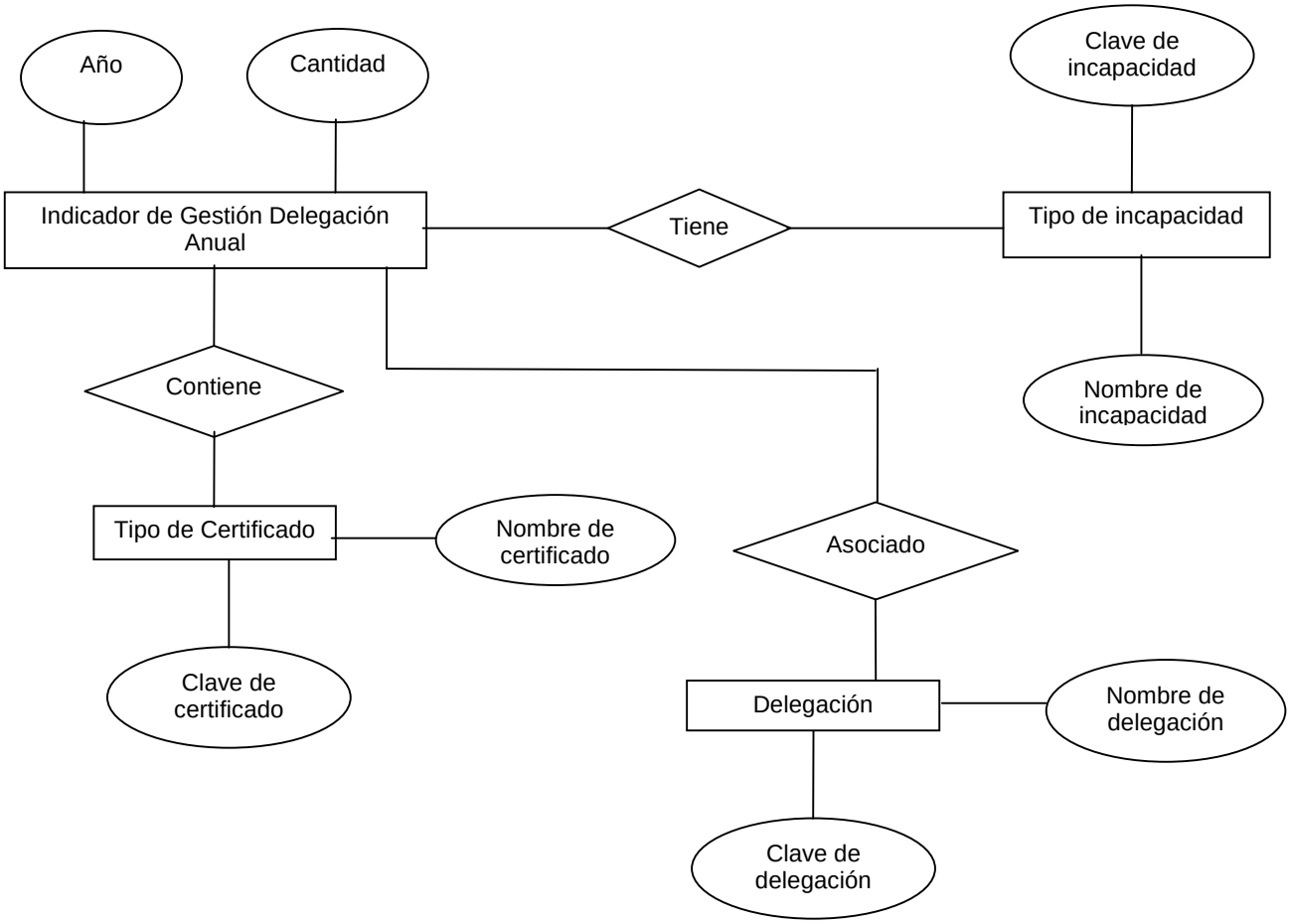

Figura 4.7

A continuación se presentan los diseños de las tablas por los cuales se agruparon los datos dentro de un nuevo archivo de Excel, el cual se denominó “BDImss”, la información contenida en este archivo será migrada al manejador SQL Server; también se presentan los campos que serán denominados llaves primarias o llaves foráneas. Cada diseño fue colocado en una hoja del archivo de Excel.

Estos diseños se elaboraron sólo una vez, por lo que no es necesario rediseñarlos.

Tabla delegación anual (Figura 4.8).

ano	Id_incapacidad	Id_certificado	Id_delegacion	cantidad

Figura 4.8

Tipo de clave: Compuesta.
Número de atributos que integran la clave primaria: cuatro atributos.
Campos que integran la clave primaria: año, Id_incapacidad, Id_certificado e Id_delegacion.
Los campos Id_incapacidad, Id_certificado e Id_delegación son llaves foráneas.

Tabla incapacidad (Figura 4.9)

Id_incapacidad	Incapacidad
1	Riesgos de Trabajo
2	Enfermedad General
3	Maternidad

Figura 4.9

Tipo de clave: Simple
Número de atributos que integran la clave primaria: 1 atributo.
Campos que integran la clave primaria: Id_incapacidad.

Tabla certificado (Figura 4.10)

Id_certificado	Certificado
1	Iniciales
2	Subsecuentes

Figura 4.10

Tipo de clave: Simple
Número de atributos que integran la clave primaria: 1 atributo.
Campos que integran la clave primaria: Id_certificado.

Tabla delegación (Figura 4.11)

id_delegación	Delegación
1	Aguascalientes
2	Baja California Norte
3	Baja California Sur
4	Campeche

5	Coahuila
6	Colima
7	Chiapas
8	Chihuahua
9	Durango
10	Guanajuato
11	Guerrero
12	Hidalgo
13	Jalisco
14	México Ote.
15	México Pte.
16	Michoacán
17	Morelos
18	Nayarit
19	Nuevo León
20	Oaxaca
21	Puebla
22	Querétaro
23	Quintana Roo
24	San Luis Potosí
25	Sinaloa
26	Sonora
27	Tabasco
28	Tamaulipas
29	Tlaxcala
30	Veracruz Norte
31	Veracruz Sur
32	Yucatán
33	Zacatecas
34	Noroeste
35	Noreste
36	Suroeste
37	Sureste

Figura 4.11

Tipo de clave: Simple
Número de atributos que integran la clave primaria: 1 atributo.
Campos que integran la clave primaria: Id_delegacion.

Todas las tablas presentadas cumplen con las primeras tres formas normales. Por ejemplo la tabla Certificado (Figura 4.12):

Id_certificado	Certificado
1	Iniciales
2	Subsecuentes

Figura 4.12

La Primera Forma Normal dice: “Todos los atributos tener un sólo valor para cada instancia”. Cada atributo de la tabla “certificado” tiene un valor único para cada ocurrencia de la entidad.

La Segunda Forma Normal se define como “Todo atributo debe ser dependiente del identificador único completo”. En la tabla “certificado”, el atributo certificado depende en su totalidad de la clave primaria (id_certificado).

La Tercera Forma Normal dice “Ningún atributo no-clave puede ser dependiente de otro atributo no-clave”. En la tabla “certificado” el atributo certificado es un atributo no clave, el cual no depende de otro atributo no-clave, por lo que se cumple esta forma normal.

Migración a la Base de Datos SQL Server

Antes de proceder con la migración de los datos, se creará una base de datos en el manejador SQL Server 2000, esta base almacenará los datos migrados.

Para ello, en el Administrador Corporativo se selecciona el nodo Bases de Datos, con el botón derecho del mouse se accede al menú emergente, en donde se selecciona la opción *Nueva base de datos*, otro camino es seleccionar el nodo mencionado, y en la barra de menú seleccionar *Acción* \ *Nueva base de datos*.

Aparecerá un cuadro de diálogo con tres pestañas, dentro del cual se definen las propiedades de la nueva base de datos, para la base de datos del presente trabajo se usó las opciones predeterminadas por SQL Server 2000 salvo el nombre de la base, el cual se propuso como HistoricoBD; este nombre se propuso por que en esta base se almacenó datos históricos, no obstante pudo ser cualquier otro, por ejemplo “HistoricoIMSS”.

Una vez creada esta base de datos, el siguiente paso es llevar los datos del nuevo archivo de Excel, al manejador SQL Server. Esto se hará con la herramienta Servicios de Transformación de Datos DTS.

En el administrador corporativo se abrió el nodo Servicios de Transformación de Datos, se seleccionó la opción “Paquetes locales”. Por cada hoja de Excel que se migró a SQL Server 2000, se creó un paquete DTS. Estos paquetes DTS fueron creados con el diseñador DTS.

Para crear un nuevo paquete se pulsa el botón derecho sobre la opción paquetes locales y se escoge la opción “Nuevo paquete”. Esto abrirá el Diseñador DTS, dentro del cual se establecerá el origen de datos y el destino de estos.

Al seleccionar el icono de Excel, aparecerá el cuadro de diálogo en donde se definirá la ruta de acceso al archivo de Excel.

Por otra parte, al seleccionar el icono *de Microsoft OLE DB Provider for SQL Server* se despliega el cuadro de diálogo en donde se define el tipo de autenticación, el cual será definido como “Utilizar autenticación de SQL Server”; se define el nombre de usuario y la contraseña. Para definir una contraseña segura, se recomienda combinar en la misma números y letras.

La lista desplegable ubicada en la parte inferior del cuadro de diálogo, muestra las bases

disponibles que el usuario tiene acceso; se localiza la base de datos denominada HistoricoBD, la cual fue el destino de los datos.

Una vez que los iconos de origen y destino aparecen en la hoja de diseño, se procede a seleccionar la Tarea Transformar Datos, la cual tiene su origen en el icono que representa el origen de datos, es decir, el archivo de Excel que contiene los datos históricos. El nombre propuesto para este archivo Excel fue “BDImss”.

Pulsando el botón derecho del mouse, en el menú emergente se selecciona la opción Propiedades, desplegándose el cuadro de diálogo “Propiedades de la tarea transformar datos”.

Dentro de la pestaña Origen, se encuentra una lista desplegable, en donde se escoge la hoja de Excel que se desee migrar; si se desea consultar como se almacenan los datos en el manejador, se pulsar el botón vista previa, en donde aparecen los primeros 100 registros de los datos a migrar.

En la pestaña Destino se realizó las operaciones de creación de tabla y creación de sus columnas(atributos).. Por defecto SQL Server propone una estructura para crear la tabla; esta estructura se modificó y para ello, se pulsó sobre el botón Crear, el cual despliega un cuadro de diálogo con comandos SQL; estos comando serán los que indiquen a SQL Server que creen la tabla con sus columnas.

La instrucción CREATE TABLE crea una tabla en la base de datos seleccionada; el nombre de la tabla viene entre las llaves “[” y “]”. SQL Server propone un nombre para la tabla que se desee crear; en cada hoja de excel migrada se cambiarán los nombres predefinidos por los que se presentan en la figura 4.17:

Nombre de la hoja de Excel	**Nombre de la tabla**
Delegación Anual	deleg_anual
Certificado	Certificado
Incapacidad	Incapacidad
Delegación	Delegación

Figura 4.13

El nombre de las columnas mostradas en el cuadro de diálogo corresponden a los que se encuentren en la hoja de Excel que se este migrando.

Los tipos de datos se definirán dentro del cuadro de diálogo; enseguida del nombre de la columna viene el tipo de dato (float, int, char).

De la figura 4.14 a la figura 4.17 se da un listado con los nombres de columnas para cada tabla, y sus respectivos tipos de datos:

Tabla Certificado

Nombre de columna	Tipo de dato
id_certificado	int
Certificado	char

Figura 4.14

Tabla deleg_anual

Nombre de columna	Tipo de dato
Año	Int
id_certificado	Int
id_incapacidad	Int
id_delegación	Int
Cantidad	Float

Figura 4.15

Tabla delegación

Nombre de columna	Tipo de dato
id_delegación	Int
Delegación	Char

Figura 4.16

Tabla incapacidad

Nombre de columna	Tipo de dato
id_incapacidad	int
Incapacidad	Char

Figura 4.17

Definidos los tipos de datos y los nombres de las tablas, se accede a la tercera pestaña, en donde no es necesario hacer modificación alguna, sin embargo en caso de que no se accede a esta pestaña, SQL Server solicitará que se defina al menos una transformación de esta pestaña.

Una vez que se ha realizado las definiciones correspondientes, se procede a ejecutar la migración, esto se realiza pulsando el botón derecho del mouse sobre la flecha “Tarea Transformar Datos” seleccionando la opción “Ejecutar Paso”; en caso de que no exista ningún error, aparecerá un mensaje indicando que el paso se ejecutó correctamente, en caso contrario aparecerá un mensaje de error.

De esta forma, en la base de datos HistoricoBD aparecerán las tablas que se migraron mediante la herramienta DTS.

Asignación de llaves primarias y construcción de vistas

Una vez creadas las tablas, se procede a la asignación de las llaves primarias para cada tabla mediante el Diseñador de Tablas. El Diseñador de Tablas es una herramienta visual que permite diseñar y visualizar una tabla individual de la base de datos que se este utilizando.

Se accede al Diseñador de tablas mediante el menú “Acción-Diseñar tabla” o bien, pulsando el botón derecho del mouse sobre la tabla y seleccionar la opción Diseñar Tabla.
El Diseñador de Tablas muestra en su parte superior una cuadrícula, en la que cada fila describe una columna de la tabla seleccionada. La cuadrícula muestra características de cada columna de la tabla, como son: nombre de columna, tipo de datos, longitud y Permitir valores nulos.

La parte inferior del Diseñador de tablas muestra otras características de la columna de datos seleccionada. Para cada columna de la tabla, se definió un tipo de dato.
Para asignar llaves primarias, se seleccionan los renglones que serán llaves primarias, posteriormente se selecciona el icono “Establecer clave principal” representado por una llave, con este procedimiento los renglones seleccionados son establecidos como llaves primarias de la tabla.

Usando el Diseñador de tablas, se procede a establecer las propiedades de cada columna, esto se realizará para todas las tablas de la base de datos.

De la figura 4.18 a la figura 4.21, se proporcionan los diseños completos de las tablas que componen la base de datos HistóricoBD, así como un listado completo sobre las columnas de cada tabla con sus respectivos valores dentro del Diseñador de Tablas.

Tabla Certificado

Nombre de columna	***Llave primaria***	***Tipo de datos***	***Longitud***	***Permitir valores nulos***
id_certificado	Si	Int	4	No
certificado	No	char	15	No

Figura 4.18

Tabla deleg_anual

Nombre de columna	***Llave primaria***	***Tipo de datos***	***Longitud***	***Permitir valores nulos***
año	Si	Int	4	No
id_certificado	Si	Int	4	No
id_incapacidad	Si	Int	4	No
id_delegación	Si	Int	4	No
cantidad	No	Float	8	No

Figura 4.19

Tabla delegacion

Nombre de columna	*Llave primaria*	*Tipo de datos*	*Longitud*	*Permitir valores nulos*
Id_delegación	Si	Int	4	No
Delegación	No	Char	45	No

Figura 4.20

Tabla incapacidad

Nombre de columna	*Llave primaria*	*Tipo de datos*	*Longitud*	*Permitir valores nulos*
id_incapacidad	Si	Int	4	No
Incapacidad	No	Char	25	No

Figura 4.21

Una vez diseñadas las tablas de la base de datos, procedemos a elaborar las vistas, las cuales son utilizadas por la aplicación web.

En SQL Server es posible crear vistas a través del Lenguaje Estructurado de Consulta SQL, sin embargo se usó la herramienta gráfica Diseñador de Vistas.

El Diseñador de vistas esta formado por cuatro paneles, los cuales son:

- Panel Diagrama.
- Panel Cuadrícula.
- Panel SQL.
- Panel Resultados.

La función del Panel Diagrama es mostrar las tablas que se están consultando. Las tablas están representadas por rectángulos, las cuales muestran en su interior sus respectivas columnas. Las relaciones entre las tablas son representadas por líneas que unen los rectángulos.

El Panel Cuadrícula contiene una cuadrícula con un formato similar al de una hoja de Excel; en la cual se especifican opciones como las columnas de datos que se han de mostrar, las filas que se han de seleccionar, la agrupación de las filas, entre otras.

La función del Panel SQL es mostrar la instrucción SQL de la consulta o de la vista. La creación de vistas implica el uso de SQL, sin embargo el manejador SQL Server las crea por el usuario. Dentro de este panel es posible modificar las instrucciones SQL que el Diseñador de Vistas utilizó, o también se escriben instrucciones SQL personalizadas. Por ejemplo, si se desea escribir una consulta la cual no es posible diseñar con el Diseñador de Vistas, se escriben las instrucciones SQL para crear la consulta en este panel.

Por último, se encuentra el Panel Resultados, el cual muestra una cuadrícula con los datos recuperados por la vista. Este panel muestra el contenido de la vista. Se usó el Panel Diagrama para construir las vistas, por que es un panel que proporciona una interfaz gráfica en donde se aprecia el diseño de la vistas.

Panel Diagrama para crear una vista

Para crear vistas, en el Administrador Corporativo se accede a la base de datos (HistoricoBD), se selecciona el icono Vistas y se selecciona del menú emergente la opción Nueva Vista.

En el Panel Diagrama, se selecciona la opción agregar tabla del menú emergente; aparecerá un cuadro de diálogo en donde se muestran todas las tablas disponibles de la base de datos. A continuación se elige el nombre de las tablas y se pulsa el botón agregar; una vez escogidas las tablas se cierra el cuadro de diálogo. En caso de que las tablas seleccionadas estén relacionadas, se mostrará su relación.

El siguiente paso es elegir el nombre de las columnas que aparecerán en la vista, seleccionando las casillas de verificación a un lado de cada nombre de columna; finalmente se ejecutará la vista y el Panel Resultados mostrará el contenido de la misma.

Para guardar la vista, se oprime el icono guardar, asignándole un nombre:
La vista que se creó corresponden con un Indicador de Gestión “Certificados Expedidos a Nivel Delegación”

Indicador de Gestión	Nombre de la Vista
Certificados Expedidos a Nivel Delegación	v_delegación_anual.

Figura 4.22

En la tabla “deleg_anual” la columna “id_incapacidad” es llave foránea de la tabla incapacidad, para unir la información de la tabla “deleg_anual” con la información asociada a la llave foránea id_incapacidad (que es llave primaria de la tabla incapacidad), se procedió a crear una vista que proporcione información de la tabla "deleg_anual" y de la tabla “incapacidad” de manera conjunta.

Esta vista proporciona la información que será consultada en la página web.

Para la vista v_deleg_anual, se seleccionó sus tablas correspondientes y las columnas que aparecerán, las cuales se dan a continuación:

Vista v_delegacion_anual.

Nombre de la tabla: deleg_anual.	Columnas: año, cantidad.
Nombre de la tabla: incapacidad.	Columnas: id_incapacidad,incapacidad
Nombre de la tabla: delegación.	Columnas: delegacion.
Nombre de la tabla: certificado.	Columnas: certificado

IV.3.3 – Fase 1. Programación en ASP

El desarrollo de la página Web fue dividido en dos etapas, la primera consistió en el diseño del formulario en la página Filtro.asp, la configuración del archivo global.asa, así como el

diseñó del código para efectuar las consultas a la base de datos.

La segunda etapa abarcó el desarrollo de las páginas de resultado donde el usuario ve desplegada su información. En la construcción del formulario, se utilizó la herramienta Microsoft InterDev 6.

El primer paso es elaborar un proyecto en InterDev, el cual se denominó HistoricoBD.

En el archivo global.asa de HistóricoBD, se definieron las siguientes propiedades:

Propiedades del script: *<SCRIPT LANGUAGE=VBScript RUNAT=Server>*

Instrucciones para el inicio de Sesión, dentro del cual se define las propiedades de conexión a la base de datos:

```
Sub Session_OnStart
    Set con = server.CreateObject("ADODB.Connection")
    con.Provider=" SQLNCLI "
    con.ConnectionString = "Provider=SQLNCLI; Data Source=halcon; Initial
    Catalog=HistoricoBD; user id= historicodpes; password= dpesc905hbd "
    con.Open
    set session("cn") = con
End Sub
```

Instrucciones para el cierre de la conexión:

```
Sub Session_OnEnd
Set session("cn") = nothing
End Sub
```

Fin de script:
</SCRIPT>

A continuacón se presenta el código ASP utilizado para la elaboración de las listas.
No se proporciona todo el código, sino los elementos más importantes.
El código completo se encuentra en el anexo 2.

Lista que contiene las opciones para los tipos de certificado

Conexión a la base de datos:
<% Set con = server.CreateObject("ADODB.Connection")

Proveedor para SQL Server 2005:
con.Provider = "SQLNCLI"

Cadena de conexión:
con.ConnectionString = "dsn=historicobd; user id=historicodpes;password=dpesc905hbd;"
con.Open

Definición de la variable “certi”. La variable certi contiene la instrucción SQL para hacer la consulta a la tabla certificado:
*certi = "Select * from Certificado"*

Creacion del objeto recordset:
set rs = Createobject("ADODB.recordset")
rs.Open certi,session("cn")

Ciclo While para llenar la lista de certificados:
do while rs.EOF <> true %>

Asigna el valor del *rs(0)* (la primer columna de la tabla certificado) como value:
<OPTION value= <%Response.Write rs(0) %> >

Asigna el valor del rs(1) (la segunda columna de la tabla certificado). Este valor se muestra en la lista:
<%Response.Write rs(1) %> </OPTION><%

Mueve el opbjeto recordset un registro hacia delante:
rs.MoveNext

Cierre del ciclo While:
loop

Cierre del objeto Recordset:
rs.Close %>

Lista que contiene las opciones para el tipo de Indicador de Gestión

Definición de la opcion del Indicador de Gestión Delegacion Anual
<OPTION value=pointDelegacional >Año 1999-2005 Delegacional Anual</OPTION>

Lista que contiene las opciones para los tipos de Incapacidad

Definición de la variable “inca”. La variable inca contiene la instrucción SQL que consulta a la tabla incapacidad:
*<% inca = "Select * from incapacidad"*

Creación del objeto recordset:
set rs = Createobject("ADODB.recordset")
rs.Open inca,session("cn")

Ciclo While para llenar la lista de certificados:
do while rs.EOF <> true %>

Asigna el valor del rs(0) (la primer columna de la tabla incapacidad):
<OPTION value= <%Response.Write rs(0) %> >

Asigna el valor del rs(1) (la segunda columna de la tabla incapacidad):

<%Response.Write rs(1) %></OPTION>

Mueve el objeto recordset un registro hacia adelante:
<% rs.MoveNext

Cierra el ciclo While:
loop

Cierra el objeto Recordset:
rs.Close %>

Lista que contiene las opciones para los nombres de las delegaciones

Definición de la variable “deleg”. La variable deleg contiene la instrucción SQL que consulta a la tabla delegación:
<% deleg = "Select delegacion from delegacion"

Creación del objeto recordset:
set rs = Createobject("ADODB.recordset")
rs.Open deleg,session("cn")

Ciclo While para llenar la lista de delegaciones:
do while rs.EOF <> true %>

Asigna el valor del rs(0) (la primer columna de la tabla delegación)
<OPTION value= <%Response.Write rs(0) %> >

Asigna el valor del rs(0) (la primer columna de la tabla delegación)
<%Response.Write rs(0) %></OPTION>

Mueve el objeto recordset un registro hacia adelante:
<% rs.MoveNext

Cierra el ciclo While:
loop

Cierra el objeto Recordset:
rs.Close %>

Lista que contiene las opciones para los años

Definición de la variable “ano”. La variable ano contiene la instrucción SQL que consulta a la tabla delegación, obteniendo los años. Se utiliza la instrucción distinct para que no existan valores repetidos:
<% ano = "Select distinct año from deleg_anual"

Creación del objeto recordset:
set rs = Createobject("ADODB.recordset")

rs.Open ano,session("cn")

Ciclo While para llenar la lista de certificados:
do while rs.EOF <> true %>

Asigna el valor del rs(0) (corresponde al valor de los años).
<OPTION value= <%Response.Write rs(0) %> >

Asigna el valor del rs(0) (corresponde al valor de los años).
<%Response.Write rs(0) %></OPTION>

Mueve el objeto recordset un registro hacia adelante:
<% rs.MoveNext

Cierra el ciclo While:
loop

Cierra el objeto Recordset:
rs.Close %>

Para que el personal de la Dirección ubique más rápido las opciones, cada lista llevará un texto descriptivo, el cual se da a conocer a continuación.

- Para la lista tipo de certificado: “Seleccione el tipo de certificado”.
- Para la lista Indicador de Gestión: “Seleccione el Indicador de Gestión”.
- Para la lista de incapacidades: “Seleccione la incapacidad”.
- Para las ultimas dos listas: “Seleccione los períodos de tiempo y las delegaciones”

En una línea aparte, se definieron las instrucciones para un script programado en JavaScript.
Definición del lenguaje del script:
<script language="javascript">

La estructura del script es la siguiente:

Se definió la función: function Valida().

La función Valida() permite lanzar un mensaje al usuario, para que seleccione al menos una opción de cada lista.

Para el Indicador de Gestión Delegacion, la función valida() queda establecida como:

function valida()

{
Si el Indicador de Gestión seleccionado es igual a Nacional Anual
if (frmFiltro.lstAlcance.value == "pointDelegacional")
{ Si no hay opciones de incapacidad seleccionadas
if (frmFiltro.lstIndicadores.selectedIndex == -1)
{ Despliega el mensaje

```
alert(" Seleccione una incapacidad");
return false;       }

Si no existen opciones de años seleccionadas.
if (frmFiltro.lstFiltro1.selectedIndex == -1)
{          Despliega el mensaje
alert(" Seleccione una delegación");
return false;
}       }          }
```

Para concluir, se ejecutó el navegador de Internet para ver la salida del formulario. La figura 4.23 muestra el formulario requerido.

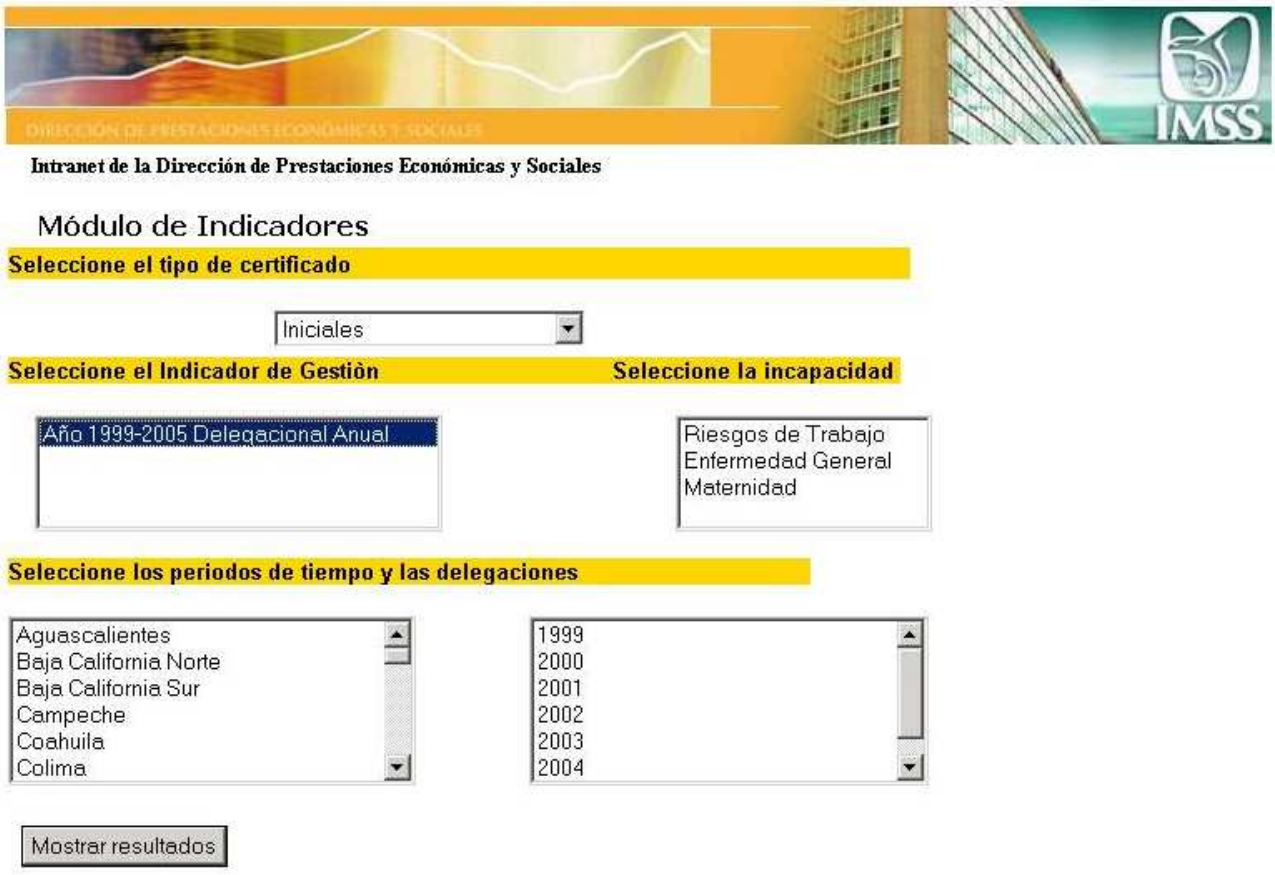

Figura 4.23

Construido el formulario, se agregó el código ASP, el cual se encarga de recibir y procesar las opciones solicitadas por el usuario, convirtiéndolas en una consulta SQL que es enviada al manejador de bases de datos, y el resultado de la misma es enviado a su respectiva página de resultado

Estructura del código ASP

Al inicio se encuentra la siguiente instrucción condicional, que verifica si se ha seleccionado

el tipo de certificado iniciales o el certificado subsecuentes
if Request("lstcbovariables") = "1" or Request("lstcbovariables") = "2" then

Esta instrucción engloba a las opciones que corresponden al Indicador de Gestión Delegación Anual.

A cada Indicador de Gestión le corresponde un condicional, el cual evalúa si el Indicador de Gestión fue seleccionado, y contiene las instrucciones para el armado de las consultas SQL,

Se propuso el diseño de dos consultas, las cuales consistieron en:

- La primer consulta regresa todos los datos solicitados por el usuario.
- La segunda consulta solo devuelve un determinado conjunto de datos, los cuales varían de acuerdo al Indicador de Gestión que se este consultando.

Dentro de cada Indicador de Gestión, existen estructuras iterativas, las cuales descomponen la solicitud almacenada en el objeto Request(nombre del elemento del formulario), y construyen determinado segmento de la consulta SQL.

Se presenta como ejemplo un segmento de código asp, el cual contiene un ciclo iterativo "For", el cual se utilizó para construir la porción de la consulta SQL referente a las delegaciones solicitadas:

La variable iCont contiene la longitud de las opciones solicitadas
iCont=len(Request("lstfiltro1"))

La variable cadena es igual al contenido del objeto Request sobre el objeto lstfiltro1
cadena=Request("lstfiltro1")

El siguiente ciclo permite analizar el contenido de la variable cadena a través de la función mid()
Desde 1 hasta el valor de iCont
for i = 1 to iCont

Si el caracter analizado es diferente al caracter de separación, es decir diferente de coma.
if mid(cadena,i,1) <> "," then

La variable token almacena el caracter analizado previamente
Token = token & mid(cadena,i,1)
else

Instrucción para formar el segmento de la consulta SQL sobre las delegaciones solicitados
sql = sql & Trim("delegacion=")& Trim(Token) & " or "
Token = ""
end if

Si la variable i es igual al valor de iCont entonces

if i = iCont then

Instrucción para la construcción final del segmento SQL para los años.
sql = sql & Trim("delegación=") & Trim(Token)
end if
next

Se presenta el código mediante el cual se elaboró la consulta SQL final, correspondiente al Indicador de Gestión Delegación Anual:
Condicional que evalúa si el usuario ha seleccionado el tipo de certificado iniciales
if *request("lstcbovariables")= 1 then*

Construcción de la primer consulta SQL concatenando todos los segmentos que se construyeron en los ciclos iterativos:
session("query4") = "select año,incapacidad,certificado,delegacion,cantidad from v_delegacion_anual where certificado='iniciales' and (" & sql & ")" & " and " & segmen & " and (" & sql2 & ") order by incapacidad,delegacion"

Construcción de la segunda consulta SQL concatenando todos los segmentos
session("query5") = "select distinct delegacion from v_delegacion_anual where (" & sql & ")" & "and" & segmen & " order by delegacion"

Condición que evalúa si el usuario ha seleccionado el tipo de certificado subsecuentes
elseif request("lstcbovariables")= 2 then

Construcción de la primer consulta SQL concatenando todos los segmentos:
session("query4") = "select año,incapacidad,certificado,delegacion,cantidad from v_delegacion_anual where certificado='subsecuentes' and (" & sql & ")" & " and " & segmen & " and (" & sql2 & ") order by incapacidad,delegacion"

Construcción de la segunda consulta SQL concatenando todos los segmentos:
session("query5") = "select distinct delegacion from v_delegacion_anual where (" & sql & ")" & "and" & segmen & "order by delegacion"

end if

IV.3.4 – Fase 2. Programación en ASP y XML

La segunda etapa consistió en elaborar la página de resultado con su respectiva hoja de estilo XSL; la cual en conjunto muestra el o los datos solicitados por el usuario, y además cuenta con determinadas características en la presentación de los mismos.

Los códigos generados para la generación de los documentos XML son compactos, por lo que se darán a conocer los códigos fuente para generar los documentos XML con su respectivo funcionamiento.

Código para generar el documento XML en la página rdelegacion.ASP del Indicador de Gestión Delegación Anual

Se da a conocer el código fuente para la página rdelegacion.ASP.

Definición del lenguaje del servidor.
<%@ Language=VBScript %>

Definición de activación del búfer
<%Response.Buffer = true%>

Definición de directivas para el documento XML y XSL.
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="delegacionanual.xsl"?>

Nodo Raíz: *<delegacionanual>*

Etiqueta XML para el nodo delegacion1. Este bloque crea las etiquetas XML con los nombres de las delegaciones solicitadas
<delegacion1>

Definición de la conexión a la base de datos
<%
set rs = Createobject("ADODB.recordset")
rs.Open session("query5"),session("cn")
do while rs.EOF <> true
%>

Etiquete XML que contiene los nombres de las delegaciones.
<nombre><%Response.Write rs(0) %></nombre>
<%

rs.MoveNext
loop

rs.Close
%>

Cierre de la etiqueta XML
</delegacion1>

Definición de la segunda conexión.
<%
set rsdos = Createobject("ADODB.recordset")
rsdos.Open session("query4"),session("cn")

do while rsdos.EOF <> true
%>

Etiqueta de inicio delegacion2:

<delegacion2>

Creación de las etiquetas XML.
Etiqueta XML para los años.
<año><%Response.Write rsdos(0)%></año>

Etiqueta XML para el tipo de incapacidad.
<incapacidad><%Response.Write rsdos(1)%></incapacidad>

Etiqueta XML para el tipo de certificado
<certificado><%Response.Write rsdos(2)%></certificado>

Etiqueta XML para el nombre de la delegación.
<nombre><%Response.Write rsdos(3)%></nombre>

Etiqueta XML que muestra la cantidad de certificados.
<cantidad><%Response.Write FormatNumber(rsdos(4),[0]) %></cantidad>

Cierre de la etiqueta delegación 2
</delegacion2>

<%
rsdos.MoveNext
loop %>

</delegacionanual>

Construcción de las hojas de estilo XSL

A continuación, se presenta la estructura propuesta para las hojas de estilo XSL para cada uno de los documentos XML. Se hace la aclaración de que no se presenta el código fuente completo, esto con el propósito de presentar las principales instrucciones XSL.

Cabe mencionar que las hojas de estilo XSL sólo procesan el documento XML y no se conectan a la base de datos. Por que en este trabajo la función de las hojas de estilo XSL es la de formatear el documento XML.

Estructura general de la hoja de estilo.

La hoja XSL tiene las siguientes instrucciones:

Definición del documento XML y definición de la hoja de estilo XSL.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:template match="/">

    Contenido XSL
    </xsl:template>
```

El contenido XSL consiste en la definición de las plantillas y restricciones que se manejen para procesar el documento XSL.

En la página de resultado existe una columna que muestra los nombres de las delegaciones seleccionadas; las demás columnas tendrán los datos solicitados agrupados por el tipo de incapacidad.

Para ello se construye una tabla principal, la cual esta formada por cuatro columnas; en la primer columna estará la columna que identifica a los nombres de las delegaciones seleccionadas, la segunda columna esta destinada a los datos correspondientes a la incapacidad Riesgos de Trabajo, la tercera columna esta destinada a la incapacidad Enfermedad General, y la cuarta columna corresponde al incapacidad de Maternidad. Esta tabla será construida dentro de la hoja de estilo XSL.

Estructura de la tabla principal:

Delegaciones	Incapacidad Riesgo de Trabajo	Incapacidad Enfermedad General	Incapacidad Maternidad

Figura 4.24

Instrucciones para la primer columna en la hoja de estilo.

A continuación se presenta el código empleado para esta columna:

Se crea la tabla*:<table cellpadding="1" cellspacing="1" border="1">*

Encabezado de la tabla: *Año*

Instrucción XSL que aplica para cada elemento del nodo seleccionado:
<xsl:for-each select= "//delegacion1/ nombre">;

Instrucción XSL para mostrar todos los valores del nodo seleccionado:
<xsl:value-of select="." />

Cierre de instrucción:*</xsl:for-each>*

Cierre de la tabla: *</table>*

Instrucciones para clasificar los datos.

A continuación se presentan las instrucciones XSL que se establecieron para procesar los nodos XML.
Para el documento XML, esta es la estructura básica del nodo delegacion2

<delegacion2>
<año> elemento </año>

<incapacidad> elemento </incapacidad>
<certificado> elemento </certificado>
<cantidad> elemento </cantidad>
</delegacion2>

Se presenta la instrucción XSL que realiza la comparación del valor de la etiqueta incapacidad del nodo delegacion2:

<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo']">

En esta instrucción procesará todos los nodos *"delegacion2"* que cumplan con esta restricción. La restricción XSL esta compuesta por:

El nombre del nodo a procesar: *//delegacion2.*

La condición a evaluar: *[incapacidad='Riesgos de trabajo']*

La propuesta de ordenamiento fue por incapacidad y por año, para ello se utiliza el siguiente bloque de instrucciones:

Condición que cumplir el nodo:
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2000]">

Construcción de la tabla:
<td><table cellpadding="1" cellspacing="1" border="1">

Título de la tabla:
<tr><td bgcolor="yellow">2000</td> </tr>

Plantilla XSL que contiene la restricción:
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2000]" />

Cierre de tabla:
</table></td>

Cierre de la condición:
</xsl:if>

Con la etiqueta *<xsl:apply-templates select "nombre del nodo[año] "/>,* se define una plantilla que será aplicada sólo a las etiquetas de año que corresponda. Se presenta el siguiente ejemplo:

<xsl:template match="//delegacion2[año = 1999]">
<xsl:apply-templates select="cantidad" />
</xsl:template>

IV.4 - Presentación de Resultados

A continuación se presenta la página de resultado para el Indicadores de Gestión Delegación Anual (Figura 4.25).

Intranet de la Dirección de Prestaciones Económicas y Sociales

Módulo de Indicadores

Certificados de incapacidad subsidiados (Trabajadores No IMSS)---> Número de certificados de incapacidad subsidiados---> Delegacion Anual

	Riesgos de Trabajo			Enfermedad General			Maternidad		
Delegacion	1999	2000	2001	1999	2000	2001	1999	2000	2001
Aguascalientes	5,716	5,837	5,111	14,399	14,763	12,548	2,920	3,135	3,428
Baja California Norte	8,075	6,664	11,176	56,842	58,153	49,410	13,312	14,519	14,587
Baja California Sur	214	295	113	5,648	5,963	5,913	1,099	1,230	1,320
Campeche	971	797	663	5,187	5,385	5,554	824	897	971
Coahuila	8,397	9,841	10,673	82,085	76,088	62,878	7,464	8,410	8,532
Colima	601	501	739	6,509	6,538	6,012	855	979	1,052

Figura 4.25

REFERENCIAS DE LA PARTE IV

PÁGINAS DE INTERNET CONSULTADAS

Dirección de Prestaciones Económicas y sociales
http://www.imss.gob.mx/IMSS/IMSS_SITIOS/DPES
IMSS 2006.

Facultades de la Dirección de Prestaciones Económicas y Sociales
http://www.imss.gob.mx/IMSS/IMSS/IMSS_ORG/Organigrama/DN/DPES/DPES_Facult_2003_12.htm
IMSS 2006.

Manual de la Dirección de Prestaciones Económicas y Sociales
http://www.imss.gob.mx/NR/rdonlyres/2A59A4B6-A756-4195-9DA6-AB057A4DAF35/0/manual_pes_2003_09.pdf
IMSS 2006.

Marco normativo del IMSS
http://www.imss.gob.mx/IMSS/IMSS/IMSS_REG/
IMSS 2006.

Organigrama del Instituto Mexicano del Seguro Social
http://www.imss.gob.mx/NR/rdonlyres/DCE558B7-E500-4D89-B308-9037C76ECFA3/0/org_dpes_06_2006.pdf
IMSS 2006.

Organización de la Dirección de Prestaciones Económicas y sociales
http://www.imss.gob.mx/IMSS/IMSS_SITIOS/DPES/DPES_ORG/
IMSS 2006.

Subsidios
http://www.imss.gob.mx/IMSS/IMSS_SITIOS/DPES/DPES_INF/Subsidios/subsidios_home_001_2003_12.htm
IMSS 2006.

CONCLUSIONES

1.- El presente trabajo se presenta bajo la opción de tesina, ya que se aplican conocimientos de bases de datos y programación, presentando una aportación de práctica profesional.

2.- La aportación de este trabajo fue el desarrollo de una aplicación web para la Dirección de Prestaciones Económicas y Sociales del Instituto Mexicano del Seguro Social. La aplicación consiste en un formulario con opciones de consulta de información histórica.
La DPES realizaba sus consultas sobre un archivo de Excel; con el uso de esta aplicación la DPES hará sus consultas de una forma más rápida y menos tediosa.

3.- El alcance de la página web abarca a la Dirección de Prestaciones Económicas y Sociales, sin embargo es posible acceder a esta página a través de la intranet del IMSS.

4.- Es necesario señalar que se tiene contemplado desarrollar otros módulos y agregar más opciones a la página, todo lo anterior será anexado a la página a través del área de Recursos Informáticos de la Dirección de Prestaciones Económicas y Sociales.

5.- XML es un metalenguaje que combinado con otras tecnologías como ASP, proporciona soporte a sitios web, combinando contenido dinámico y satisfaciendo las necesidades o peticiones de los usuarios del sitio web. Por medio de una combinación de instrucciones ASP embebidas en etiquetas XML, se crean documentos XML extensos.

6.- Es posible cambiar hojas de estilo XSL, sin afectar la estructura del documento XML, lo cual se constituye como una ventaja. En un caso concreto, si la Dirección de Prestaciones Económicas y Sociales desea cambiar la presentación de sus datos, sólo es necesario cambiar las instrucciones XSL por aquellas que procesen los documentos XML de acuerdo a la presentación requerida.

7.- Cuando la organización tiene datos almacenados en un formato inadecuado, (Microsoft Excel) la consulta de datos es complicada y tediosa. Microsoft Excel no es un manejador de bases de datos, por lo que no es posible ejecutar consultas o crear vistas de datos en este programa, tampoco es posible llevar un control sobre datos redundantes. Se recomienda usar un sistema manejador de bases de datos para almacenar los datos, por que el manejador proporciona rapidez, consistencia e integridad a los datos que se manejen y almacenen.

8.- En ciertos aspectos ASP.NET y ASP 3.0 tienen sus diferencias, sin embargo las páginas ASP 3.0 tienen compatibilidad con ASP.NET. Resulta interesante hacer una migración de esta misma página a ASP.NET, con el objetivo de apreciar más de cerca sus características, inconvenientes y ventajas.

ANEXOS

Anexo 1.
Código fuente del archivo Global.asa

<!--
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán

Programación ASP Y XML aplicada a la creación de una página web de consulta de información histórica.

Tesina para obtener el título de Matemáticas Aplicadas y Computación

Presenta: Sánchez Martínez Sergio Román.

Asesor: Físico Matemático Jorge Luis Suárez Madariaga

->
‘Definicón del lenguaje del lado del servidor
<SCRIPT LANGUAGE=VBScript RUNAT=Server>

Sub Session_OnStart

‘Código para definir las propiedades de la conexión a la base de datos, como el proveedor, ‘servidor, base de datos, nombre de usuario y contraseña.

con.Provider = "SQLncli" con.ConnectionString = "dsn=historicobd; user id=historicodpes; password=dpesc905hbd;"
con.Open

Set session("cn") = con

End Sub

‘Código para terminar la sesión

Sub Session_OnEnd
set session("cn") = nothing
End Sub

Fin del script
</SCRIPT>

Anexo 2.
Código fuente del archivo Filtro.asp

```
<! --
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán
Programación ASP Y XML aplicada a la creación de una página web de consulta de información histórica.
Tesina para obtener el título de Matemáticas Aplicadas y Computación
Presenta: Sánchez Martínez Sergio Román.
Asesor: Físico Matemático Jorge Luis Suárez Madariaga

->

‘Definición del lenguaje del lado del servidor
<%@ Language=VBScript %>

<%
'Instrucción para definir si se desea activar el búfer.
Response.Buffer = true

‘Instrucción que engloba las opciones para la consulta de los Indicadores de Gestión
‘Nacional, Delegacion y Region

if Request("lstcbovariables") = "1" or Request("lstcbovariables") = "2" then

        ‘Código para el Indicador de Gestión Delegación Anual
        if Request("lstAlcance")="pointDelegacional" then

                ‘Definición de la variable cadena
                cadena=""

                ‘Definición de variables
                iCont=len(Request("lstIndicadores"))
                cadena=Request("lstIndicadores")

                ‘Ciclo para obtener las incapacidades solicitadas
                segmen= "(id_incapacidad="
                        for i = 1 to icont
                                if mid(cadena,i,1)<> "," then
                                        segmen= Trim(segmen) & mid(cadena,i,1)
                                        else
                                        segmen= Trim(segmen) & (" or id_incapacidad=")
                                end if
                        next

                segmen= Trim(segmen) & ")"

                ‘Definición de variables
```

```
iCont=len(Request("lstfiltro1"))
cadena=Request("lstfiltro1")

‘Ciclo para obtener el nombre de las delegaciones solicitadas
for i = 1 to iCont
        if mid(cadena,i,1) <> "," then
                Token= token & mid(cadena,i,1)
        else
        sql = sql & Trim("delegacion='") & trim(Token) & trim("'") & " or "
        Token=""
        end if

        if  i = icont then
                'sql="Select * from indicador1 where"
                sql = sql & Trim("delegacion='") & trim(Token) &  trim("'")

        end if
 next

‘Definición de variables
iCont=len(Request("lstfiltro2"))
cadena=Request("lstfiltro2")

‘Ciclo para obtener los años solicitados
for i = 1 to iCont
          if mid(cadena,i,1) <> "," then
                Token2 = token2 & mid(cadena,i,1)
            else
                sql2 = sql2 & Trim("año=")& Trim(Token2) & " or "
        Token2 = ""
        end if
        if  i = iCont then
                sql2 = sql2 & Trim("año=") &  Trim(Token2)
        end if
 next


‘Construcción de la consulta SQL si el usuario escogió certificados iniciales
 if request("lstcbovariables")= 1 then

‘Consulta 1
session("query4") = "selectaño,incapacidad,certificado,delegacion,cantidad from
v_delegacion_anual where certificado='iniciales' and (" & sql & ")" & " and " & segmen & " and
(" & sql2 & ") order by incapacidad,delegacion"

‘Consulta 2
session("query5") = "select distinct delegacion from v_delegacion_anual where (" & sql & ")"
& "and" & segmen & " order by delegacion"

‘Construcción de la consulta SQL si el usuario escogió certificados subsecuentes
```

```
 'Consulta 1
session("query4") = "select año,incapacidad,certificado,delegacion,cantidad from v_delegacion_anual where certificado='subsecuentes' and (" & sql & ")" & " and " & segmen & " and (" & sql2 & ") order by incapacidad,delegacion"

'Consulta 2
session("query5") = "select distinct delegacion from v_delegacion_anual where (" & sql & ")" & "and" & segmen & "order by delegacion"
                        end if

            'Redirección a la página de resultado
             Response.Redirect("rdelegacion.ASP")

            'Término de ejecución
            Response.End

        end if

end if  %>

<HTML><HEAD>

<script language="javascript">
// Función Valida
function valida()
{
//Condicional: Si en el valor de la lista de Indicadores de Gestión es igual a Delegación Anual
if (frmFiltro.lstAlcance.value == "pointDelegacional")
{
                //Si no se seleccionó una incapacidad
                if (frmFiltro.lstIndicadores.selectedIndex == -1)
                        {
                        //Muestra un mensaje
                        alert("Debe seleccionar una incapacidad");
                        return false;
                        }
                //Si no se seleccionó una delegación
                 if (frmFiltro.lstFiltro1.selectedIndex == -1)
                        {
                        //Muestra un mensaje
                        alert("Debe seleccionar una delegacion");
                        return false;
                        }

                //Si no se seleccionó un año
                if (frmFiltro.lstFiltro2.selectedIndex == -1)
                        {
                        //Muestra un mensaje
                        alert("Debe seleccionar un año");
                        return false;  }  }
```

```
return true;
}

Fin del script
</script>

</HEAD> <BODY>

<!-- Propiedades del Formulario ->
<form id=frmFiltro name=frmFiltro action="Filtro.asp" method=post>

<!—Definición de la tabla ->
<table cellSpacing="0" cellPadding="1" width="75%" border="0">
<tr><td>

<!—Imagen del encabezado ->
<FONT color=black><IMG height=83 alt ="" src="images/Encabezado.JPG" width=761 >
</FONT></td></tr><tr><td><P>

<!—Texto ->
<FONT color=black>
<strong><font size=2>Intranet de la Dirección de Prestaciones Económicas y Sociales</font>
</strong></FONT>
</P>

<!—Texto ->
<P><STRONG><FONT face=Verdana color=black>Módulo de Indicadores </FONT>
</STRONG></P></td></tr></table>

<!—Texto ->
<table border=0 style="WIDTH: 770px; HEIGHT: 381px"><tr><td><P>
<FONT style="BACKGROUND-COLOR: gold">
<FONT face=Arial><FONT size=2>
<STRONG> PASO 1: Seleccione el tipo de certificado</STRONG>
</FONT></FONT></FONT></P><P>

<!-- Lista que contiene las opciones para los tipos de certificado ->
<SELECT id=lstcbovariables style="WIDTH: 187px" name=lstcbovariables>

<!-- Código ASP para consultar a la base de datos los tipos de certificado ->
<%
        ‘Definición de la conexión a base de datos
        Set con = server.CreateObject("ADODB.Connection")

        'Proveedor para SQL Server 2005
        con.Provider = "SQLncli"

        'Cadena de conexión
        con.ConnectionString = "dsn=historicobd; user id=historicodpes;
        password=dpesc905hbd;"
```

```
con.Open

'Consulta para la lista de opciones de certificado
certi = "Select * from Certificado"

‘Creación del objeto recordset
set rs = Createobject("ADODB.recordset")
rs.Open certi,session("cn")

'Ciclo While para llenar la lista de certificados
'Mientras la propiedad sea diferente de EOF
do while rs.EOF <> true
%>

<!--Asigna el valor de la primer columna de la tabla certificado como value->
<OPTION value= <%Response.Write rs(0)
%> >

<!--Asigna el valor de la segunda columna de la tabla
certificado. Este valor se muestra en la lista ->
<%Response.Write rs(1) %></OPTION>

<%
'Mueve el recordset una posición hacia adelante
rs.MoveNext

'Cierre del ciclo While
loop

'Cierre del objeto Recordset
rs.Close %>

<!—Etiqueta de cierre ->
</SELECT>

</P></td></tr><tr><td><P><STRONG><FONT style="BACKGROUND-COLOR: gold"
face=Arial size=2> Seleccione el Indicador de Gestiòn Seleccione la incapacidad
</FONT></STRONG></P>
<P>

‘Opciones para los Indicadores de Gestión
<SELECT id=lstAlcance style="WIDTH: 245px; HEIGHT: 78px" onclick="return
Alcance();" size =2 name=lstAlcance>

<OPTION value=pointDelegacional >Año 1999-2005 Delegacion Anual</OPTION>
</SELECT>

<!-- Opciones para la lista de Incapacidades ->
<SELECT style="WIDTH: 154px; HEIGHT: 78px" multiple size=2 name=lstIndicadores
id=lstIndicadores>
```

```
<%
'Definición de la consulta a la base de datos
inca = "Select * from incapacidad"

'Definición del recordset
set rs = Createobject("ADODB.recordset")
rs.Open inca,session("cn")

'Ciclo While
do while rs.EOF <> true
%>

'Llenado de la lista de incapacidad
<OPTION value= <%Response.Write rs(0)
%> ><%Response.Write rs(1) %></OPTION>
<%

'Mueve el recordset una posición hacia adelante
rs.MoveNext
loop

'Cierre del recordset
rs.Close %>

</SELECT>

</P></td></tr><tr><td> <P><STRONG><FONT style="BACKGROUND-COLOR: gold"
    face=Arial size=2>Paso 4: Seleccione los periodos de tiempo y las delegaciones o regiones</FONT></STRONG></P> <P>

<!-- Lista de delegaciones -->
<SELECT id=lstFiltro1 style="WIDTH: 245px; HEIGHT: 107px" multiple size=2 name=lstFiltro1>

<%
'Consulta sql
deleg = "Select delegacion from delegacion"
set rs = Createobject("ADODB.recordset")

'Definición
rs.Open deleg,session("cn")
do while rs.EOF <> true %>

'Llenado de la lista
<OPTION value= <%Response.Write rs(0)
%> ><%Response.Write rs(0) %></OPTION><%

'Mueve el recordset una posición hacia adelante
rs.MoveNext
```

```
    'Fin de ciclo
    loop

    'Fin del recordset
    rs.Close %>

'Cierre de lista
</SELECT>

<!-- Lista de años   ->
<SELECT id=lstFiltro2 style="WIDTH: 240px; HEIGHT: 107px" multiple size=2 name=lstFiltro2>
<%
    'Consulta para años
    ano = "Select distinct año from deleg_anual"

    'Definición de recordset
    set rs = Createobject("ADODB.recordset")
    rs.Open ano,session("cn")

    'Ciclo While
    do while rs.EOF <> true %>

    'Llenado de lista
    <OPTION value= <%Response.Write rs(0)
    %> ><%Response.Write rs(0) %></OPTION>
    <!—Mueve el recordset una posicion  ->
    <% rs.MoveNext

    'Fin de ciclo
    loop

    'Cierra el recordset
    rs.Close %>

Cierre de lista
</SELECT></P> <P></P>

<!-- Botón enviar -->
<INPUT id=cmdEnviar onClick="return valida();" style="LEFT: -1px; WIDTH: 124px; TOP: 0px; HEIGHT: 25px" type="submit" size=65 value="Mostrar resultados" name=cmdEnviar>
</td></tr></table>

<!—Fin de formulario ->
</form>
</BODY></HTML>
```

Anexo 3.
Código fuente del archivo rdelegacion.

```
<!--
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán

Programación ASP Y XML aplicada a la creación de una página web de consulta de información histórica.

Tesina para obtener el título de Matemáticas Aplicadas y Computación

Presenta: Sánchez Martínez Sergio Román.

Asesor: Físico Matemático Jorge Luis Suárez Madariaga
->
<!—Definición del lenguaje del lado del servidor ->
<%@ Language=VBScript %>

<%Response.Buffer = true%>

<!-- Definición del documento XML ->
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="delegacionanual.xsl"?>

<!-- Nodo Raíz ->
<delegacionanual>

<!-- Nodo delegación 1 ->
<delegacion1>

<%

‘Definición del objeto recordset
set rs = Createobject("ADODB.recordset")
rs.Open session("query5"),session("cn")

‘Ciclo while para el llenado de la etiqueta “Nombre de la delegación”
do while rs.EOF <> true %>
<nombre><%Response.Write rs(0) %></nombre>
<%rs.MoveNext

‘Fin de ciclo
loop rs.Close %>

<!--  Etiqueta de cierre ->
</delegacion1>

<%
```

```
‘Definición del segundo recordset
set rsdos = Createobject("ADODB.recordset")
rsdos.Open session("query4"),session("cn")

‘Ciclo While para el llenado de las etiquetas XML
do while rsdos.EOF <> true %>

<!— Etiqueta de inicio  ->
<delegacion2>

<!—Elementos XML->
<año><%Response.Write rsdos(0)%></año>
<incapacidad><%Response.Write rsdos(1)%></incapacidad>
<certificado><%Response.Write rsdos(2)%></certificado>
<nombre><%Response.Write rsdos(3)%></nombre>
<cantidad><%Response.Write FormatNumber(rsdos(4),[0]) %></cantidad>

<!—Etiqueta de cierre  ->
</delegacion2>

<%
‘Mueve el objeto Recorset una posición
rsdos.MoveNext

‘Fin de ciclo
loop %>

<!-- Cierre de etiqueta. ->
</delegacionanual>
```

Anexo 4
Código fuente del archivo delegacionanual.xsl

<!--
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán

Programación ASP Y XML aplicada a la creación de una página web de consulta de información histórica.

Tesina para obtener el título de Matemáticas Aplicadas y Computación

Presenta: Sánchez Martínez Sergio Román.

Asesor: Físico Matemático Jorge Luis Suárez Madariaga

->

<!—Encabezado de documento XML ->
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!—Inicio de hoja de estilo ->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">

<!—Título y encabezado de la página web ->
<title>Instituto Mexicano del Seguro Social</title>
<table><tr><IMG height="83" alt="" src="images/Encabezado.JPG" width="761" />
</tr><tr><td>Intranet de la Dirección de Prestaciones Económicas y Sociales
</td></tr>

<tr><td>Módulo de Indicadores </td>
</tr><tr/><tr/><tr/><tr/> </table>

Certificados de incapacidad subsidiados (Trabajadores No IMSS)---> Número de certificados de incapacidad subsidiados---> Delegación Anual
*

*

<!—Tabla en donde aparecen los años ->
<table cellpadding="1" cellspacing="1" border="0">
<tr><td>Delegacion
<table cellpadding="1" cellspacing="1" border="0">
<tr><td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">
Delegacion</td></tr>

<!—Instrucción XSL para mostrar los nombres de las delegaciones ->
<xsl:for-each select="//delegacion1/nombre">

```
<tr><td bgcolor="e0e0e0" nowrap="1">
<xsl:value-of select="." />
</td></tr></xsl:for-each>
</table></td></tr></table></td><td>
```

<!-- Instrucción XSL para seleccionar la incapacidad Riesgo de trabajo->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo']">
<b>Riesgos de Trabajo</b>
<table cellpadding="1" cellspacing="1" border="1">
<tr>
```

<!—Instrucción XSL para seleccionar el año 1999 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=1999]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">1999</td> </tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=1999]" />
</table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2000 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2000]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2000</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2000]" />
</table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2001 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2001]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2001</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2001]" />
</table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2002 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2002]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2002</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2002]" />
</table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2003 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2003]">
<td><table cellpadding="1" cellspacing="1" border="1">
```

```
<tr><td bgcolor="yellow">2003</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2003]" />
</table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2004 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2004]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2004</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2004]" />
</table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2005 que pertenece a la incapacidad Riesgos de trabajo-->

```
<xsl:if test="//delegacion2[incapacidad='Riesgos de trabajo'][año=2005]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td nowrap="1" bgcolor="yellow">Enero-Junio 2005</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Riesgos de trabajo'][año=2005]" />
</table></td></xsl:if>

</tr></table></xsl:if>

</td><td>
```

<!—Instrucción XSL para seleccionar la incapacidad Enfermedad General-->

```
<xsl:if test="//delegacion2[incapacidad='Enfermedad General']">
<b>Enfermedad General</b>
<table cellpadding="1" cellspacing="1" border="1">
<tr>
```

<!—Instrucción XSL para seleccionar el año 1999 que pertenece a la incapacidad Enfermedad General-->

```
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=1999]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">1999</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=1999]"
/></table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2000 que pertenece a la incapacidad Enfermedad General -->

```
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=2000]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2000</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=2000]"
/></table></td></xsl:if>
```

<!—Instrucción XSL para seleccionar el año 2001 que pertenece a la incapacidad Enfermedad General -->
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=2001]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2001</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=2001]" /></table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2002 que pertenece a la incapacidad Enfermedad General -->
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=2002]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2002</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=2002]" /></table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2003 que pertenece a la incapacidad Enfermedad General -->
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=2003]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2003</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=2003]" /></table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2004 que pertenece a la incapacidad Enfermedad General -->
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=2004]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2004</td>
</tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=2004]" /></table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2005 que pertenece a la incapacidad Enfermedad General -->
<xsl:if test="//delegacion2[incapacidad='Enfermedad General'][año=2005]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td nowrap="1" bgcolor="yellow">Enero-Junio 2005</td></tr>
<xsl:apply-templates select="//delegacion2[incapacidad='Enfermedad General'][año=2005]" /></table></td></xsl:if>

</tr></table></xsl:if>
</td><td>

<!—Instrucción XSL para seleccionar el año la incapacidad Maternidad-->
<xsl:if test="//delegacion2[incapacidad='Maternidad']">

```
<b>Maternidad</b>
<table cellpadding="1" cellspacing="1" border="1">
<tr>

<!—Instrucción XSL para seleccionar el año 1999 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=1999]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">1999</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=1999]" />
</table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2000 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=2000]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2000</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=2000]" />
</table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2001 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=2001]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2001</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=2001]" />
</table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2002 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=2002]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2002</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=2002]" />
</table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2003 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=2003]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2003</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=2003]" />
</table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2004 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=2004]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td bgcolor="yellow">2004</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=2004]" />
```

```
</table></td></xsl:if>

<!—Instrucción XSL para seleccionar el año 2005 que pertenece a la incapacidad Maternidad
-->
<xsl:if test="//delegacion2[incapacidad='Maternidad'][año=2005]">
<td><table cellpadding="1" cellspacing="1" border="1">
<tr><td nowrap="1" bgcolor="yellow">Enero-Junio 2005</td>
</tr><xsl:apply-templates select="//delegacion2[incapacidad='Maternidad'][año=2005]" />
</table></td></xsl:if>

</tr></table></xsl:if>
</td></tr></table>
<br/><br/><br/> <p>

<span style="FONT-STYLE: italic">Prototipo, sujeto a cambios de diseño e información.
<br/> Actualizado al 9 de septiembre de 2005
</span></p><br /> <hr width="98%" align="left" /> <div class="IMSS" />
<span class="IMSS">2005 <a href="http://www.imss.gob.mx" target="_self">©</a>
<a href="http://www.imss.gob.mx">Instituto Mexicano del Seguro Social</a> </span>

<!-- Fin de plantilla ->
</xsl:template>

<!—Plantilla para el año 1999 ->
<xsl:template match="//delegacion2[año = 1999]">
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!—Plantilla para el año 2000 ->
<xsl:template match="//delegacion2[año = 2000]">
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!—Plantilla para el año 2001 ->
<xsl:template match="//delegacion2[año = 2001]">
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!—Plantilla para el año 2002 ->
<xsl:template match="//delegacion2[año = 2002]">
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!—Plantilla para el año 2003 ->
<xsl:template match="//delegacion2[año = 2003]">
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!—Plantilla para el año 2004 ->
<xsl:template match="//delegacion2[año = 2004]">
```

```
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!—Plantilla para el año 2005 ->
<xsl:template match="//delegacion2[año = 2005]">
<tr><td><xsl:apply-templates select="cantidad" />
</td></tr></xsl:template>

<!-- Fin de la hoja de estilo ->
</xsl:stylesheet>
```

REFERENCIAS

REFERENCIAS DE LA PARTE I

LIBROS CONSULTADOS

1.- Gutiérrez Rodríguez, Abraham. Raúl Martínez González. **XML A TRAVES DE EJEMPLOS**. Alfaomega Ra-Ma. Colombia. 2001

2. - Bradley Neil. **THE XSL COMPANION**. Addison-Wesley. Great Britain. 2000.

3.- Gunter, Born. **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML.** Marcombo. 2001

4.- Vazquez Rodríguez, Adolfo **NAVEGAR EN INTERNET XML**. Alfaomega Ra-Ma. México 2002.

5.- Young, Michael J. **APRENDA XML YA**. Traducción de Vuelapluma, S.L. McGraw-Hill. España. 2000

PÁGINAS DE INTERNET CONSULTADAS.

Recomendación XML (en inglés)

a) http://www.w3c.org
Sitio Oficial del World Wide Web Consortium.

b) http://www.w3c.org /TR/1998/REC-xml-19980210.
Versión 1.0.

c) http://www.w3.org/TR/REC-xml.
Última versión.

Recomendacion XML (en español)

d) http://www.sidar.org/recur/desdi/traduc/es/xml/xml1/index.html

Última actualización: 30/06/2006
Traducción: Carlos Benavides y Emmanuelle Gutiérrez.
Editores: Carlos Benavides, Emmanuelle Gutiérrez y Restrepo
Colaborador: Charles McCathieNevile

Hojas de estilo en cascada CSS

e) http://www.sidar.org/recur/desdi/mcss/manual/intro.php

Última actualización: 30/06/2006
Traducción: Carlos Benavides y Emmanuelle Gutiérrez.
Editores: Carlos Benavides, Emmanuelle Gutiérrez y Restrepo
Colaborador: Charles McCathieNevile

XML

f) http://manuales.dgsca.unam.mx/xml/Desventajas.htm

Año: 2006.
Sitio WEB desarrollado e implementado por la Dirección General de Servicios de Cómputo Académico
Director General: Dr. Alejandro Pisanty Baruch
Dirección de Sistemas: Mtro. Juan Voutssas Márquez
Subdirección de Comunicación: Lic. José Antonio Sánchez Yllañez
Subdirección de Servicios Web: L.I. Luz Ma. Ramírez Romero
Diseño Gráfico, adaptación HTML y JavaScript: L. D.G. Raúl Ramírez Sánchez
Jefa de Departamento: L.A. Nancy Escorcia Martínez

g) http://www.adobe.com/es/devnet/dreamweaver/articles/xml_overview_03.html

Fecha de creación: 8 August 2005.
Adobe Systems Incorporated

h) http://www.dcc.uchile.cl/~rbaeza/inf/xml.html

Rediseño del sitio: Newtenberg Ltd.
Última actualización: 13/7/2006 9:33
Autor: Ricardo Baeza-Yates

Gramática EBNF

i) http://www.garshol.priv.no/download/text/bnf.html

j) http://es.wikipedia.org/wiki/Backus-Naur_form

REFERENCIAS DE LA PARTE II

LIBROS CONSULTADOS

1. - Banick Steve, Michael Morrison. **EDICIÓN ESPECIAL VISUAL INTERDEV 6.** Prentice Hall. Madrid 1999.

5.- Bobadilla Sancho, Jesús. Alcocer Jarabo, Alejandro. Manzaneque Sánchez Luis Rodríguez. **ACTIVE SERVER PAGES (ASP 3.0). INICIACIÓN Y REFERENCIA.** Osborne

McGrawHill. España 2002.

2.- Gunter, Born. **COMPENDIUM HTML CON XHTML, DHTML, CSS, XML, XSL Y WML.** Marcombo. 2001

3.- Power Shelley. **DESARROLLO DE COMPONENTES ASP.** Anaya Multimedia. Madrid 2001.

4.- Santry, Patrick. Tulloch Mitch. **ADMINISTERING IIS 5.0.** McGraw Hill. U.S.A. 2000.

PÁGINAS DE INTERNET CONSULTADAS.

Actualización de IIS versión 5 a versión 6
a) http://support.microsoft.com/kb/325889/es
2006 Microsoft Corporation

Desventajas de ASP
b) http://www.prosumedia.com.mx/documento.php?id_not=6

Listado de las versiones de IIS
c) http://support.microsoft.com/kb/224609/es
2006 Microsoft Corporation

Migración de ASP 3.0 a ASP.NET
d) http://es.gotdotnet.com/quickstart/aspplus/doc/migrationoverview.aspx

Tipos de input types
e) http://www.osmosislatina.com/lenguajes/html/formas.htm
Actualizado : 2005/10/21

REFERENCIAS DE LA PARTE III

LIBROS CONSULTADOS

1.- Dalton Patrick, Whitehead Paul. **LA BIBLIA DE SQL SERVER 2000.** Ediciones Anaya Multimedia. España 2002.

2.- Pérez López, César. **DOMINE MICROSOFT SQL SERVER 2000. ADMINISTRACIÓN Y ANÁLISIS DE BASES DE DATOS.** Alfaomega Ra-Ma. España.

PÁGINAS DE INTERNET CONSULTADAS

Comparación de características de SQL 2005.

a) http://www.microsoft.com/spain/sql/productinfo/features/compare-features.mspx
2006. Microsoft Corporation.

Descarga de libros en pantalla de SQL Server 2005 Express Edition.
b) http://www.microsoft.com/downloads/details.aspx?familyid=BE6A2C5D-00DF-4220-B133-29C1E0B6585F&displaylang=es.
2006. Microsoft Corporation.

Descarga de SQL Server 2005 Express Edition.
c) http://www.microsoft.com/downloads/details.aspx?familyid=220549B5-0B07-4448-8848-DCC397514B41&displaylang=es.
2006. Microsoft Corporation.

Información de SQL Server 2005.
d) http://www.microsoft.com/latam/technet/productos/servers/sql/2005/default.mspx.
2006. Microsoft Corporation.

Información de SQL Server 2005 Express Edition
e) https://www.microsoft.com/spanish/msdn/articulos/archivo/261205/voices/sseoverview.mspx #E2FAC
Fecha de publicación: 26 de Diciembre de 2005.
Microsoft Corporation.

Instalación de SQL Server 2005.
f) http://msdn2.microsoft.com/es-es/library/ms143719.aspx
Ultima actualización: 5 de diciembre de 2005.
Microsoft Corporation.

Panorama General de SQL Server 2005 Express Edition.
g) http://www.microsoft.com/spanish/msdn/articulos/archivo/100904/voices/10PanoramaSQLServer2005ExpressEdition.asp
Autor: Rajesh George
Junio 2004
Microsoft Corporation.

Requisitos de SQL Server 2005 (Todas las versions).
h) http://msdn2.microsoft.com/es-es/library/ms143506.aspx
Ultima actualización: 5 de diciembre de 2005.
Microsoft Corporation.

Requisitos de SQL Server 2005 Express Edition.
i) http://download.microsoft.com/download/a/2/3/a23083ba-88d4-4e89-b9fb-dfd3b618bbdb/RequirementsSQLEXP2005.htm.
2006. Microsoft Corporation.

SQL Server Management Studio.
k) http://msdn2.microsoft.com/es-es/library/ms165690.aspx
2006 Microsoft Corporation.

REFERENCIAS DE LA PARTE IV

PÁGINAS DE INTERNET CONSULTADAS

Dirección de Prestaciones Económicas y sociales
http://www.imss.gob.mx/IMSS/IMSS_SITIOS/DPES
IMSS 2006.

Facultades de la Dirección de Prestaciones Económicas y Sociales
http://www.imss.gob.mx/IMSS/IMSS/IMSS_ORG/Organigrama/DN/DPES/DPES_Facult_2003_12.htm
IMSS 2006.

Manual de la Dirección de Prestaciones Económicas y Sociales
http://www.imss.gob.mx/NR/rdonlyres/2A59A4B6-A756-4195-9DA6-AB057A4DAF35/0/manual_pes_2003_09.pdf
IMSS 2006.

Marco normativo del IMSS
http://www.imss.gob.mx/IMSS/IMSS/IMSS_REG/
IMSS 2006.

Organigrama del Instituto Mexicano del Seguro Social
http://www.imss.gob.mx/NR/rdonlyres/DCE558B7-E500-4D89-B308-9037C76ECFA3/0/org_dpes_06_2006.pdf
IMSS 2006.

Organización de la Dirección de Prestaciones Económicas y sociales
http://www.imss.gob.mx/IMSS/IMSS_SITIOS/DPES/DPES_ORG/
IMSS 2006.

Subsidios
http://www.imss.gob.mx/IMSS/IMSS_SITIOS/DPES/DPES_INF/Subsidios/subsidios_home_001_2003_12.htm
IMSS 2006.