



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURIA Y ADMINITRACION

**GRAFICACIÓN DE INDICADORES ECONÓMICOS BAJO
TECNOLOGIAS WAMP**

**DISEÑO DE UN SISTEMA O PROYECTO PARA UNA
ORGANIZACIÓN**

YAIR ALBERTO RAMÍREZ MUNIVE



MÉXICO, D.F.

2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURIA Y ADMINISTRACION

**GRAFICACIÓN DE INDICADORES ECONÓMICOS BAJO
TECNOLOGIAS WAMP**

**DISEÑO DE UN SISTEMA O PROYECTO PARA UNA
ORGANIZACIÓN
QUE PARA OBTENER EL TÍTULO DE:**

LICENCIADO EN INFORMÁTICA

PRESENTA

YAIR ALBERTO RAMÍREZ MUNIVE

ASESOR:

DRA. GRACIELA BRIBIESCA CORREA



MÉXICO, D.F.

2010

Dedico este Proyecto a quienes me han brindado su apoyo y consejos y en los momentos más difíciles me alentaron a seguir adelante, anhelando que siempre me preparara para enfrentarme a la vida.

Hoy se han culminado nuestros esfuerzos y mis deseos iniciándose así una nueva etapa en mi vida, en la que siempre estarán en mi corazón. Por ello doy gracias...

A ustedes Abuelitos Hermenegildo y Rosa, por cuidar de mi y forjar mi vida como a otro de sus hijos. Por ser Papá y Mamá después de la escuela y por todo su amor y cariño.

A la memoria de mis Abuelitos Juan y Catalina, que desde el cielo nos han bendecido y cuidado, intercediendo día a día por nosotros.

A mi Papá, Juventino, mi mejor amigo y compañero de vida. Por ser modelo de vida y siempre creer en mí. Caballero Noble lleno de bondad y sabiduría.

A mi Mamá Emma, por apoyarme cuando nadie lo hizo, por creer en mí cuando todos desconfiaban, por perdonarme cuando nadie entendía. Por darme tu vida, tu amor y tú espacio.

A mis hermanos Christopher y Tannia, mis estrellas de la suerte. Por el apoyo y paciencia que me han tenido. Por dar soporte a mis ideas y creer conmigo imposibles.

A Cinthya, mi amiga y compañera de vida. Por haberme dado el regalo más bello de vida, la noticia más importante en mi vida, la dicha de ser Padre.

A mi Hijo Erick, la razón de mi existir, la persona más importante en mi vida. Mil gracias porque muchas cosas de mi cambiaron con tu llegada. Ilusiones trajiste a nuestras vidas y junto con ella la dicha para mí de saber el significado de la palabra "PAPA".

A mis profesores "por explicarme que para tener la cabeza en el cielo, hay que tener primero los pies en la tierra"...

Por todo esto a Dios agradezco por haber sido parte de mi vida.

INTRODUCCIÓN	4
I DESCRIPCION Y DEFINICIÓN DE CONCEPTOS	7
I .1 INTERNET	7
I .2 TCP/IP EL PROTOCOLO DE COMUNICACIONES	7
I .3 WORLD WIDE WEB.....	8
I .4 HIPER TEXT TRANSFER PROTOCOL.....	9
I .5 TECNOLOGÍAS CLIENTE/SERVIDOR.....	10
I .6 APACHE2TRIAD.....	10
I .7 HYPER TEXT MARKUP LENGUAJE.....	12
I .7.1 Elementos	12
I .7.2 Códigos básicos en HTML	13
I .8 CASCADING STYLE SHEETS	15
I .8.1 Tipos de Estilo.....	16
I .9 BASES DE DATOS	17
I .9.1 Tipos de bases de datos	17
I .9.1.1 Según la variabilidad de los datos almacenados.....	18
I .9.1.1.1 Bases de datos estáticas.....	18
I .9.1.1.2 Bases de datos dinámicas.....	18
I .9.1.2 Según el contenido.....	18
I .9.1.2.1 Bases de datos bibliográficas.....	18
I .9.1.2.2 Bases de datos de texto completo.....	19
I .9.1.2.3 Directorios	19
I .9.1.2.4 Bases de datos o "bibliotecas" de información Biológica	19
I .9.2 Modelos de bases de datos.....	19
I .9.2.1 Bases de datos jerárquicas	20
I .9.2.2 Base de datos de red	20
I .9.2.3 Base de datos relacional	21
I .9.2.4 Bases de datos multidimensionales	22
I .9.2.5 Bases de datos orientadas a objetos.....	22
I .9.2.6 Bases de datos documentales	23
I .9.2.7 Base de datos deductivas	23
I .9.2.8 Gestión de bases de datos distribuida.....	24
I .10 JAVA SCRIPT	24
I .10.1 Incluir Java Script en una página	24
I .11 SSH.....	26
I .11.1 Seguridad	26
I .11.2 Telnet, problemas de seguridad y SSH.....	26
I .12 APACHE.....	28
I .12.1 Uso de Apache	29
I .13 MYSQL.....	30
I .13.1 Características principales de MySQL.....	30
I .13.2 Conexión a un servidor MYSQL	31
I .13.3 Uso y manipulación de una Base de Datos.....	32
I .13.4 Crear una Base de Datos	34
I .13.5 Crear una Tabla	35
I .13.6 Ingresar Datos	38

I .13.7 Consulta de información de una tabla	40
I .13.8 Actualizar la información de una tabla.....	40
I .13.9 Eliminar la información de una tabla.....	41
I .13.10 Eliminar toda la información de una tabla.....	41
I .13.11 Eliminar una tabla.....	41
I .14 PHP.....	41
I .14.1 “Hola Mundo”.....	42
I .14.2 Uso de Variables con PHP	43
I .14.3 Uso de Operadores con PHP	45
I .14.3.1 Operadores Aritméticos.....	45
I .14.3.2 Operadores de Comparación	47
I .14.3.3 Operadores Lógicos	48
I .14.4 Uso de Instrucciones de programación con PHP	50
I .14.4.1 Sentencias Condicionales	50
I .14.4.1.1 Sentencia IF – ELSE	50
I .14.4.1.2 Sentencia SWITCH - CASE.....	51
I .14.4.2 Bucles.....	53
I .14.4.2.1 Sentencia WHILE	53
I .14.4.2.2 Sentencia FOR.....	55
I .14.5 ENVIAR y RECIBIR datos con PHP	57
II ANTECEDENTES	61
II .1 IIEc (Instituto de Investigaciones Económicas)	61
II .1.1 Fundación IIEc	61
II .1.2 Autonomía	61
II .1.3 Actualidad.....	62
II .2 METODOLOGÍA A USAR	65
II .3 ANÁLISIS DEL SITIO WEB DINAMICO	67
II .3.1 Análisis de Requerimientos	67
II .3.2 Detección de Necesidades.....	68
II .3.3 Soluciones Sugeridas.....	69
II .3.3.1 Aclaración y Comprensión del Proyecto.....	69
II .3.3.1.1 Interfaz de Administrador.....	70
II .3.3.1.2 Interfaz de Usuario	71
II .3.3.2 Objetivo General del Sitio Web sugerido.....	72
III PROPUESTA DE DESARROLLO, ESTRUCTURA DE BASE DE DATOS Y ESTRUCTURA DEL SITIO.	73
III .1 ESTRUCTURA DE ARCHIVOS Y ALMACENAJE	73
III .2 DISEÑO Y ESTRUCTURA DE LA BASE DE DATOS.....	74
III .2.1 Determinar la Finalidad	74
III .2.2 Determinar las tablas que se necesitan para la Base de Datos	75
III .2.3 Determinar los campos que se necesitan para las Tablas	76
III .2.4 Identificar los campos con valores exclusivos	77
III .2.5 Determinar las Relaciones entre las Tablas	77
III .2.6 Precisar el Diseño	78
III .2.7 Introducir datos y crear otros objetos de la Base de Datos	78

III .2.8 Estructura final de la Base de Datos	79
III .3 DISEÑO DE LA ESTRUCTURA DEL SITIO WEB.	80
III .3.1 Elegir Herramientas de Trabajo.....	80
III .3.2 Estructura de Encabezado y Hoja de estilos	81
III .3.3 Estructura de la programación del Sitio.....	82
III .3.3.1 Módulos de Conexión.....	82
III .3.3.1.1 Conexiones a la Base de Datos	82
III .3.3.2 Modulo de Inserción	84
III .3.3.2.1 Estructura de Formularios	84
III .3.3.2.2 Guardar valores enviados en la Base de Datos	86
III .3.3.3 Módulos de Consulta.....	88
III .3.3.3.1 Consultar información a la Base de Datos.....	88
III .3.3.3.2 Imprimir los valores consultados a la Base de Datos	89
III .3.3.4 Módulos de Graficación.....	89
III .3.3.4.1 JpGraph.....	89
III .3.3.4.2 Generar Gráficos mediante JpGraph.....	91
IV DESARROLLO E IMPLEMENTACION DEL SITIO WEB PROPUESTO PARA LA GRAFICACION DE INDICADORES ECONOMICOS.....	94
IV.1 Modulo 1 - Menú de Usuario.....	94
IV.2 Modulo 2 - Consulta y Graficación del valor de distintas mezclas del valor de petróleo al día.....	98
IV.3 Modulo 3 - Consulta e Imprime el valor de distintas mezclas del valor de petróleo por día.	104
IV.4 Modulo 4 - Búsqueda y Consulta de los valores de distintas mezclas de petróleo por día.	106
IV.5 Modulo 5 - Formulario de Ingreso para la sesión de Administrador.....	114
IV.6 Modulo 6 - Menú de Usuario Administrador.....	117
IV.7 Modulo 7 - Formulario de Ingreso para nuevos registros del precio por barril de petróleo al día.....	122
IV.8 Modulo 8 - Consulta, Imprime y modifica el valor de distintas mezclas del valor de petróleo por día.	128
IV.9 Modulo 9 - Consulta, Imprime y elimina el valor de distintas mezclas del valor de petróleo por día.	131
CONCLUSIONES	145
ANEXOS	148
ANEXO 1 – INSTALACIÓN DE APACHE2TRIAD.....	148
ANEXO 2 – CREAR UN SERVIDOR DE PRUEBAS EN DEAMWEAVER.....	154
GLOSARIO DE TÉRMINOS TÉCNICOS.....	162
BIBLIOGRAFÍA	169

INTRODUCCIÓN

El Instituto de Investigaciones Económicas (IIEc) fue creado para apoyar académicamente a la Escuela Nacional de Economía en un doble aspecto: complementar las cátedras con los medios indispensables para que los estudiantes pudieran realizar sus trabajos estadísticos y contables, así como orientarlos en las tareas de investigación y, primordialmente, realizar investigaciones sobre la problemática económica nacional.

El Instituto cuenta con una importante producción de libros individuales y colectivos, la mayoría publicados en coedición con universidades nacionales e internacionales, editoriales privadas y dependencias gubernamentales. En el año 2008 se aprobó e inicio el desarrollo del Proyecto Revista Digital el cual tendrá como objetivo convertirse en un medio de divulgación y espacio de análisis interpretativo relacionado con la problemática económica mundial actual, enfocada a la comunidad académica, estudiantil y público interesado; así como satisfacer sus necesidades de información económica.

El tema programado para la publicación del número cero es “Reforma Energética en México” en el cual se encuentra basado el desarrollo de este proyecto.

Es este motivo por el cual el Instituto tiene la necesidad de desarrollar la Plataforma Telemática ReDIIEc la cual tiene como función apoyar al área de investigación del Instituto, llevando el control de las publicaciones realizadas, los artículos dictaminados de publicación periódica (cuatrimestralmente), artículos de coyuntura, materiales diversos para un Acervo (Reseñas, notas periodísticas, documentales, etc.) y el control de indicadores económicos así como su graficación.

En este trabajo de titulación se diseñará una solución basada en el control de indicadores económicos de petróleo tomando en cuenta las etapas de desarrollo

de sistemas que le permitirán al usuario hacer uso de este sistema de manera fácil y eficaz, haciendo más sencilla la actualización del sitio.

El Objetivo de este trabajo es:

Manipular los indicadores de petróleo, dando de alta nuevos registros, habilitando la eliminación y modificación de los mismos, diseñando búsquedas avanzadas y personalizadas de registros y graficar el valor de distintas mezclas del valor de petróleo al día. Todo esto a través de la creación de un sitio Web dinámico.

El alcance de este trabajo de titulación es manipular los indicadores de petróleo a través de un sitio Web dinámico funcional para el Instituto de Investigaciones Económicas contemplando las fases del desarrollo de sistemas para el control de Indicadores Económicos de petróleo y graficación de su valor al día. Así mismo proponer el desarrollo de otros indicadores económicos tales como PIB, BMV, monedas nacionales y devaluaciones etc.

En el capítulo 1, se hace una breve descripción de los conceptos que se utilizan para realizar este trabajo de titulación, tales como: Internet, Tecnologías Cliente-Servidor, *Apache2triad*, *HTML*, *CSS*, *Java Script*, *SSH*, Bases de Datos, *MYSQL*, y *PHP*.

En el capítulo 2, se mencionan los antecedentes del Instituto de Investigaciones Económicas y se describe como se llevó a cabo el análisis, detección de necesidades y soluciones sugeridas.

En el capítulo 3, se presenta una propuesta de desarrollo a las soluciones sugeridas, definiendo la estructura del sitio y la estructura de la base de datos.

En el capítulo 4, Se muestra el desarrollo final y la implementación del sitio Web dinámico. Estableciendo la estructura, ubicación de elementos y descripción detallada de los módulos.

I DESCRIPCIÓN Y DEFINICIÓN DE CONCEPTOS

1.1 INTERNET

Es el resultado de la interconexión de miles de computadoras en todo el mundo. Todas ellas comparten los protocolos de comunicación, es decir que todos hablan el mismo lenguaje para ponerse en contacto unas con otras. Los servicios básicos ofrecidos ahora por Internet son correo electrónico, noticias en red, acceso a computadoras remotas y sistemas de adquisición de datos y la capacidad para transferir información entre computadoras remotas.

El protocolo utilizado para esta gran red es llamado *TCP/IP* (Transfer Control Protocol) el cual se encarga de contabilizar la transmisión de datos entre computadoras y registrar si hay o no errores, mientras que *IP* (Internet Protocol) es el que realiza realmente la transferencia de datos.

1.2 TCP/IP EL PROTOCOLO DE COMUNICACIONES

Una red existe cuando hay dos o más computadoras conectadas de forma que puedan compartir y pasar información entre ellas. Cada una de estas máquinas se denomina *Host* o nodo de la red. Si proporciona un servicio específico, tal como la verificación de una contraseña esta computadora se denominará Servidor.

Los nodos de una red siguen un conjunto de reglas, denominadas Protocolos para intercambiar información, que a su vez sirve también para definir los servicios que pueden estar disponibles en una computadora. Hay muchos tipos diferentes de protocolos, aunque los más habituales proporcionan conexiones *TCP/IP* que permiten que los usuarios se conecten a Internet.

El protocolo de comunicaciones *TCP/IP* sirve como núcleo de Internet. Este protocolo de comunicaciones permite conectar computadoras que utilizan distintos sistemas operativos.

Trabaja a nivel de capa de red y de transporte en la clasificación del modelo de la *ISO/OSI*.

Una dirección *IP* es un número formado por cuatro octetos de la siguiente forma xxx.xxx.xxx.xxx donde cada xxx representa un número decimal entre 0 y 255 e identifica en forma única a cada dispositivo conectado a la gran red, por ejemplo 132.248.18.65 identifica una red y un *host* dentro de esa red.

Como a las personas les es difícil manejarse con números, se manejan mediante nombres que la red se encarga de traducir a direcciones *IP*, así el nombre completo de una máquina puede ser "fca.unam.mx".

1.3 WORLD WIDE WEB

La visualización de una página Web de la World Wide Web normalmente comienza tecleando la *URL* de la página en el navegador Web, o siguiendo un enlace de hipertexto a esa página o recurso. En ese momento el navegador comienza una serie de comunicaciones, transparentes para el usuario, para obtener los datos de la página y visualizarla. Cuando se utiliza bastante en empresas

El primer paso consiste en traducir la parte del nombre del servidor de la *URL* en una dirección *IP* usando la base de datos distribuida de Internet conocida como *DNS*. Esta dirección *IP* es necesaria para contactar con el servidor Web y poder enviarle paquetes de datos.

El siguiente paso es enviar una petición *HTTP* al servidor Web solicitando el recurso. En el caso de una página Web típica, primero se solicita el texto *HTML* y luego es inmediatamente analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros ficheros que formen parte de la página.

Las estadísticas de popularidad de un sitio Web normalmente están basadas en el número de 'páginas vistas' o las 'peticiones' de servidor asociadas, o peticiones de fichero, que tienen lugar.

Al recibir los ficheros solicitados desde el servidor Web, el navegador renderiza la página tal y como se describe en el código *HTML*, el *CSS* y otros lenguajes Web. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla.

1.4 HIPER TEXT TRANSFER PROTOCOL

HTTP define la manera en que los desarrolladores Web interactúan con los servidores Web. *HTTP* utiliza *TCP/IP* el protocolo de Red en Internet. Utilizando protocolos estándar como ese, el usuario podrá comunicarse con cualquier servidor Web desde distintos navegadores Web y en espera de respuesta.

El corazón del *HTTP* reside en un mensaje de solicitud y un mensaje de respuesta. El usuario escribe la ubicación de elemento en el cuadro de texto dirección y dicho documento es localizado y devuelto al navegador como respuesta *HTTP* estándar.

1.5 TECNOLOGÍAS CLIENTE/SERVIDOR

La tecnología llamada Cliente /Servidor es actualmente utilizada en casi todas las aplicaciones administrativas e Internet/Intranet. Bajo este esquema, un servidor es un ordenador remoto, en algún lugar de una red, que proporciona información según se le solicite. Mientras que un cliente funciona en su computadora local, se comunica con el servidor remoto y pide a éste información.

Típicamente, un único servidor atiende a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente. Los sistemas Cliente-Servidor pueden ser de muchos tipos, pues esto depende principalmente de las aplicaciones instaladas en el propio servidor. Entre otros, existen: servidores de impresión mediante los cuales los usuarios comparten impresoras, servidores de archivos con los que los clientes comparten discos duros, servidores de bases de datos donde existe una única base de datos que es consultada por los clientes y puede o no ser modificada por ellos y servidores Web que utilizan también la tecnología Cliente/Servidor, aunque añaden aspectos nuevos y propios a la misma.

Los servidores web quizás actualmente son los de mayor importancia. Un servidor web es aquel que permite a los clientes compartir datos, documentos y multimedia en formato web. Aunque parte de la tecnología Cliente/Servidor, el servidor Web aporta ventajas adicionales a un servidor tradicional. (Williams & Lane, 2001)

1.6 APACHE2TRIAD

Apache2triad es una distribución de software del tipo *WAMP* que integra algunos de los servidores y de los intérpretes de código libre más populares para desarrollar en un entorno web y proporcionar un *webhosting*, entre su contenido más notable está:

Servidores:

- Apache (servidor web)
- MySQL (servidor de bases de datos)
- PostgreSQL (servidor de bases de datos)
- XMail (servidor de correo electrónico)
- SlimFTPd (servidor ftp)

Intérpretes:

- PHP (Lenguaje interpretado)
- Perl (Lenguaje interpretado)
- Python (Lenguaje interpretado)
- Interfaces Gráficas de Usuario:
- Apache2TriadCP (Panel de Control)
- phpMyAdmin (GUI para MySQL)
- phpPgAdmin (GUI para PostgreSQL)
- phpSQLiteAdmin (GUI para SQLite)
- AWStats (webserver monitor)
- UebiMiau (cliente de correo electrónico)
- PHPXmail (GUI para XMail)
- PHPsFTPd (GUI para SlimFTPd)

Apache2triad se puede llamar un sistema de infraestructura de Internet, porque contiene todos los servidores, intérpretes, e interfaces usuarios ya configurados y listos para ser utilizados. (Apeche2triadNet)

I.7 HYPER TEXT MARKUP LENGUAJE

HTML, es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. *HTML* se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). *HTML* también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *Script* (por ejemplo *Java Script*), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de *HTML*.

HTML también es usado para referirse al contenido del tipo de *MIME Text/html* o todavía más ampliamente como un término genérico para el *HTML*, ya sea en forma descendida del *XML* (como *XHTML* 1.0 y posteriores) o en forma descendida directamente de *SGML* (como *HTML* 4.01 y anteriores). (Wikipedia)

Por convención, los archivos de formato *HTML* usan la extensión *.htm* o *.html*.

I.7.1 Elementos

Los elementos son la estructura básica de *HTML*. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento *HTML*. Un elemento generalmente tiene una etiqueta de inicio (p.ej. <nombre-de-elemento>) y una etiqueta de cierre (p.ej. </nombre-de-elemento>).

Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (p.ej. <nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>). Algunos elementos, tales como
, no tienen contenido ni llevan una etiqueta de cierre. Debajo se listan varios tipos de elementos de marcado usados en *HTML*.

El marcado estructural describe el propósito del texto. Por ejemplo, `<h2>FCA</h2>` establece a "FCA" como un encabezamiento de segundo nivel. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores Web han estandarizado el formato de los elementos. Un formato específico puede ser aplicado al texto por medio de hojas de estilo en cascada.

El marcado presentacional describe la apariencia del texto, sin importar su función. Por ejemplo, `negrita` indica que los navegadores Web visuales deben mostrar el texto en negrita, pero no indica qué deben hacer los navegadores Web que muestran el contenido de otra manera (por ejemplo, los que leen el texto en voz alta).

En el caso de `negrita` e `<i>itálica</i>`, existen elementos que se ven de la misma manera pero tienen una naturaleza más semántica: `énfasis fuerte` y `énfasis`. Es fácil ver cómo un lector de pantalla debería interpretar estos dos elementos. Sin embargo, son equivalentes a sus correspondientes elementos presentacionales: un lector de pantalla no debería decir más fuerte el nombre de un libro, aunque éste esté en *itálicas* en una pantalla. La mayoría del marcado presentacional ha sido desechada con *HTML* 4.0, en favor de Hojas de estilo en cascada. (Factoría de Internet)

I .7.2 Códigos básicos en HTML

`<html>`: define el inicio del documento *HTML*, le indica al navegador que lo que viene a continuación debe ser interpretado como código *HTML*.

`<head>`: define la cabecera del documento *HTML*, esta cabecera suele contener información sobre el documento que no se muestra directamente al usuario. Como por ejemplo el título de la ventana del navegador. Dentro de la cabecera `<head>` podemos encontrar:

- `<title>`: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana
- `<link>`: para vincular el sitio a hojas de estilo o iconos Por ejemplo:`<link rel="stylesheet" href="/style.css" type="text/css">`
- `<style>`: para colocar el estilo interno de la página, ya sea usando CSS, *Java Script* u otros lenguajes similares. No es necesario colocarlo si se va a vincular a un archivo externo usando la etiqueta `<link>`

`<body>`: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo `<body>` podemos encontrar numerosas etiquetas. A continuación se indican algunas a modo de ejemplo:

- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`: encabezados o títulos del documento con diferente relevancia.
- `<table>`: define una tabla
- `<tr>`: fila de una tabla
- `<td>`: celda de datos de una tabla
- `<a>`: Hipervínculo o enlace, dentro o fuera del sitio Web. Debe definirse el parámetro de pasada por medio del atributo `href`. Por ejemplo: `FCA` se representa como FCA)
- `<div>`: área de la página
- ``: imagen. Requiere del atributo `src`, que indica la ruta en la que se encuentra la imagen. Por ejemplo: ``
- ``, ``, ``: Etiquetas para listas.
- ``: texto en negrita (*Etiqueta descartada. Se recomienda usar la etiqueta ``*)
- `<i>`: texto en cursiva
- `<u>`: texto subrayado

La mayoría de etiquetas deben cerrarse como se abren, pero con una barra ("/") tal como se muestra en los siguientes ejemplos:

- `<table><tr><td>Contenido de una celda</td></tr></table>`
- `<script>Código de un script integrado en la página</script>`

1.8 CASCADING STYLE SHEETS

Las hojas de estilo en cascada (CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en *HTML* o *XML* (y por extensión en *XHTML*). El *W3C* (*World Wide Web Consortium*) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de *CSS* es separar la *estructura* de un documento de su *presentación*.

Por ejemplo, el elemento de *HTML* `<H1>` indica que un bloque de texto es un encabezamiento y que es más importante que un bloque etiquetado como `<H2>`. Versiones más antiguas de *HTML* permitían atributos extra dentro de la etiqueta abierta para darle formato (como el color o el tamaño de fuente).

No obstante, cada etiqueta `<H1>` debía disponer de la información si se deseaba un diseño consistente para una página y, además, una persona que lea esa página con un navegador pierde totalmente el control sobre la visualización del texto.

Cuando se utiliza *CSS*, la etiqueta `<H1>` no debería proporcionar información sobre cómo va a ser visualizado, solamente marca la estructura del documento. La información de estilo separada en una hoja de estilo, especifica cómo se ha de mostrar `<H1>`: color, fuente, alineación del texto, tamaño y otras características no visuales como definir el volumen de un sintetizador de voz, por ejemplo. (Tomas, 2001)

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento *HTML*. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

I .8.1 Tipos de Estilo

CSS proporciona tres caminos diferentes para aplicar las reglas de estilo a una página Web:

1. **Una hoja de estilo externa**, que es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código *HTML* de la página Web. Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página *HTML* de la estructura básica de la página.
2. **Una hoja de estilo interna**, que es una hoja de estilo que está incrustada dentro de un documento *HTML*. (Va a la derecha dentro del elemento <head>). De esta manera se obtiene el beneficio de separar la información del estilo, del código *HTML* propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra, (esta posibilidad es difícil de ejecutar si se desea para guardar las copias sincronizadas). En general, la única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se está enviando algo a la página web.
3. **Un estilo en línea**, que es un método para insertar el lenguaje de estilo de página, directamente, dentro de una etiqueta *HTML*. Esta manera de proceder no es excesivamente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código se convierte en una tarea larga, tediosa y poco elegante de resolver el problema de la programación de la página.

Este modo de trabajo se podría usar de manera ocasional si se pretende aplicar un formateo con prisa, al vuelo. No es todo lo claro, o estructurado, que debería ser, pero funciona. (WebEstilo)

1.9 BASES DE DATOS

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos. (Piattini, 2001)

Existen programas denominados sistemas gestores de bases de datos, *SGBD*, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal(*LOPD*).

1.9.1 Tipos de bases de datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación: (SitePoint, 2003)

I .9.1.1 Según la variabilidad de los datos almacenados

I .9.1.1.1 Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

I.9.1.1.2 Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

I .9.1.2 Según el contenido

I .9.1.2.1 Bases de datos bibliográficas

Solo contienen un surrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

I .9.1.2.2 Bases de datos de texto completo

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

I .9.1.2.3 Directorios

Un ejemplo son las guías telefónicas en formato electrónico.

I .9.1.2.4 Bases de datos o "bibliotecas" de información Biológica

Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

Aquellas que almacenan secuencias de nucleótidos o proteínas.

Las bases de datos de rutas metabólicas

Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas

Bases de datos clínicas

Bases de datos bibliográficas (biológicas)

I .9.2 Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos son:

I .9.2.1 Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

I .9.2.2 Base de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

I .9.2.3 Base de datos relacional

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es *SQL*, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años '80 (1980-1989) la aparición de *dBASE* produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que *dBase* no utilizaba *SQL* como lenguaje base para su gestión. (DuBois, 2002)

I .9.2.4 Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP.

Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos multidimensional podría serlo también en una base de datos relacional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

I .9.2.5 Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz.

Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Se está trabajando en *SQL3*, que es el estándar de *SQL92* ampliado, que soportará los nuevos conceptos orientados a objetos y mantendrá compatibilidad con *SQL92*. (Guiarte Multimedia)

I .9.2.6 Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Taurus es un sistema de índices optimizado para este tipo de bases de datos.

I .9.2.7 Base de datos deductivas

Un sistema de base de datos deductivos, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática. (Interactive)

I .9.2.8 Gestión de bases de datos distribuida

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.

I .10 JAVA SCRIPT

Java Script es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje *Java* y el lenguaje *C*.

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas *Java Script* van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos.

I.10.1 Incluir Java Script en una página

Existen distintos modos de incluir lenguaje *Java Script* en una página.

La forma más frecuente de hacerlo es utilizando la directiva `<script>` en un documento *HTML* (se pueden incluir tantas directivas `<script>` como se quiera en un documento). El formato es el siguiente:

```
<script language="Javascript 1.3">
```

El atributo lenguaje hace referencia a la versión de *JavaScript* que se va a utilizar en dicho script.

Otro atributo de la directiva `script` es `src`, que puede usarse para incluir un archivo externo que contiene *JavaScript* y que quiere incluirse en el código *HTML*.

```
<script language="JavaScript" src ="archivo.js"> </script>
```

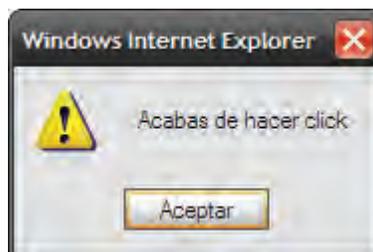
El archivo externo simplemente es un archivo del texto que contiene código *JavaScript*, y cuyo nombre acaba con la extensión `js`.

Puede incluirse también código *JavaScript* como respuesta a algún evento:

```
<input type="submit" onclick="alert('Acabas de hacer click');return false;" value="Click">
```



El *Script* anterior creara un Botón con la leyenda `Click` y un evento `OnClick`, esto quiere decir que al dar click sobre el botón mostrará una alerta con la leyenda `Acabas de hacer click`.



Los scripts pueden incluirse como comentarios para asegurar que su código no es "visto" por navegadores viejos que no reconocen *JavaScript* y así evitar errores.

```
<script....>  
<!--  
Código JavaScript  
!-->  
</script>
```

Para validar el código en *HTML* es necesario incluir los comentarios *HTML* `<!--` y `-->` para que no se interpreten los símbolos `<` y `>` como inicio y fin de etiqueta.

I.11 SSH

SSH (Secure SHell, en español: intérprete de órdenes seguro) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de *X* para poder ejecutar programas gráficos si tenemos un Servidor *X* (en sistemas *Unix*) corriendo.

Además de la conexión a otras máquinas, *SSH* nos permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones *FTP* cifradas), gestionar claves *RSA* para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante *SSH*.

I.11.1 Seguridad

SSH trabaja de forma similar a como se hace con *telnet*. La diferencia principal es que *SSH* usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible y ninguna tercera persona pueda descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión; aunque es posible atacar este tipo de sistemas por medio de ataques de *REPLAY* y manipular así la información entre destinos.

I.11.2 Telnet, problemas de seguridad y SSH

Telnet (*TELEcommunication NETwork*) es el nombre de un protocolo de red (y del programa informático que implementa el cliente), que sirve para acceder mediante una red a otra máquina, para manejarla remotamente como si estuviéramos sentados delante de ella.

Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23.

Telnet sólo sirve para acceder en modo terminal, es decir, sin gráficos, pero fue una herramienta muy útil para arreglar fallos a distancia, sin necesidad de estar físicamente en el mismo sitio que la máquina que los tenía. También se usaba para consultar datos a distancia, como datos personales en máquinas accesibles por red, información bibliográfica, etc.

Aparte de estos usos, en general telnet se ha utilizado (y aún hoy se puede utilizar en su variante *SSH*) para abrir una sesión con una máquina *UNIX*, de modo que múltiples usuarios con cuenta en la máquina, se conectan, abren sesión y pueden trabajar utilizando esa máquina. Es una forma muy usual de trabajar con sistemas *UNIX*.

Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajan por la red como texto plano (cadenas de texto sin cifrar). Esto facilita que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas, y así acceder él también a todas esas máquinas. (Sitepoint, 2003)

Por esta razón dejó de usarse, casi totalmente, hace unos años, cuando apareció y se popularizó el *SSH*, que puede describirse como una versión cifrada de telnet (actualmente se puede cifrar toda la comunicación del protocolo durante el establecimiento de sesión (RFC correspondiente, en inglés) si cliente y servidor lo permiten, aunque no se tienen ciertas funcionalidad extra disponibles en *SSH*).

1.12 APACHE

El servidor *HTTP Apache* es un servidor web *HTTP* de código abierto para plataformas *Unix* (*BSD*, *GNU/Linux*, etc.), *Windows*, *Macintosh* y otras, que implementa el protocolo *HTTP/1.1* y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular *NCSA HTTPd 1.3*, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu *Apache* fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además *Apache* consistía solamente en un conjunto de parches a aplicar al servidor de *NCSA*. Era, en inglés, a *patchy server* (un servidor "parcheado").

El servidor *Apache* se desarrolla dentro del proyecto *HTTP Server (httpd)* de la *Apache Software Foundation*.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, *Apache*, es el servidor *HTTP* más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan *PHP* como módulo de *Apache*. (Sklar, 2004)

I .12.1 Uso de Apache

Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a *Apache*, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones *LAMP*, junto a *MySQL* y los lenguajes de programación *PHP/Perl/Python* y *Ruby*.

Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere application server. Mac OS X integra *Apache* como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. *Apache* es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene *Apache* instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de *Apache*, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de *Apache* en orden de pre visualizar y probar código mientras éste es desarrollado.

Microsoft Internet Information Services (IIS) es el principal competidor de *Apache*, así como *Sun Java System Web Server* de Sun Microsystems y un anfitrión de otras aplicaciones como *Zeus Web Server*. Algunos de los más grandes sitios web del mundo están ejecutándose sobre *Apache*. (Apache.org)

La capa frontal (*front end*) del motor de búsqueda Google está basado en una versión modificada de *Apache*, denominada Google Web Server (GWS). Muchos proyectos de Wikimedia también se ejecutan sobre servidores web *Apache*.

1.13 MYSQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. *MySQL AB* desarrolla *MySQL* como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la *GNU GPL* para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en *ANSI C*.

Al contrario de proyectos como *Apache*, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, *MySQL* es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. (Welling & Thomson, 2002)

1.13.1 Características principales de MySQL

- Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación *SQL*.

- Es *Open Source*. El código fuente de *MySQL* se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia *GPL* para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de software que la usa.

I .13.2 Conexión a un servidor MYSQL

Para conectarse al servidor, usualmente se necesita de un nombre de usuario (login) y de una contraseña (password), y si el servidor al que se desea conectar está en una máquina diferente de la nuestra, también se necesita indicar el nombre o la dirección *IP* de dicho servidor. Una vez que se conocen estos tres valores, se podrá realizar la conexión de la siguiente manera:

```
shell> mysql -h NombreDelServidor -u NombreDeUsuario -p
```

Cuando se ejecuta este comando, se nos pedirá que proporcionemos también la contraseña para el nombre de usuario que se está usando.

Si la conexión al servidor *MySQL* se pudo establecer de manera satisfactoria, recibiremos el mensaje de bienvenida y estaremos en el *prompt* de *mysql*:

```
shell>mysql -h NombreDelServidor -u NombreDeUsuario -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 5563 to server version: 3.23.41
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```

Este *prompt* indicará que *mysql* está listo para recibir comandos.

Algunas instalaciones permiten que los usuarios se conecten de manera anónima al servidor corriendo en la máquina local. Si es el caso de nuestra máquina, debemos de ser capaces de conectarnos al servidor invocando a *mysql* sin ninguna opción:

```
shell> mysql
```

Después de que se ha conectado de manera satisfactoria, podemos desconectarnos en cualquier momento al escribir "quit", "exit", o presionar CONTROL+D.

I .13.3 Uso y manipulación de una Base de Datos

Teniendo cientos de Maestros en la UNAM, deseamos tener registros de los datos de cada uno de los maestros de la FCA. Podemos hacer esto al crear tablas que guarden esta información, para que posteriormente la consulta de estos datos sea bastante fácil y de manera muy práctica. Esta sección muestra cómo crear una base de datos, crear una tabla, incorporar datos en una tabla, y recuperar datos de las tablas de diversas maneras.

La base de datos "FCA" será muy simple (deliberadamente), pero no es difícil pensar de situaciones del mundo real en la cual una base de datos similar puede ser usada.

Primeramente usaremos la sentencia *SHOW* para ver cuáles son las bases de datos existentes en el servidor al que estamos conectados:

```
mysql> SHOW DATABASES;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| mysql    |
```

```
| test     |
```

```
+-----+
```

```
2 rows in set (0.00 sec)
```

```
mysql>
```

Es probable que la lista de bases de datos que veamos sea diferente en nuestro caso, pero seguramente las bases de datos "mysql" y "test" estarán entre ellas.

En particular, la base de datos "mysql" es requerida, ya que ésta tiene la información de los privilegios de los usuarios de MySQL. La base de datos "test" es creada durante la instalación de MySQL con el propósito de servir como área de trabajo para los usuarios que inician en el aprendizaje de MySQL.

Se debe anotar también que es posible que no veamos todas las bases de datos si no tenemos el privilegio *SHOW DATABASES*. Se recomienda revisar la sección del manual de MySQL dedicada a los comandos *GRANT* y *REVOKE*.

Si la base de datos "test" existe, hay que intentar acceder a ella:

```
mysql> USE test
```

```
DATABASE changed
```

```
mysql>
```

Observar que *USE* no requiere el uso del punto y coma, aunque si se usa éste, no hay ningún problema. El comando *USE* es especial también de otra manera: éste debe ser usado en una sola línea.

Podríamos usar la base de datos "*test*" (si tenemos acceso a ella) para los ejemplos que vienen a continuación, pero cualquier cosa que hagamos puede ser eliminada por cualquier otro usuario que tenga acceso a esta base de datos. Por esta razón, es recomendable que preguntemos al administrador *MySQL* acerca de la base de datos que podemos usar. Supongamos que deseamos tener una base de datos llamada "*fca*" a la cual sólo nosotros tengamos acceso, para ello el administrador necesita ejecutar un comando como el siguiente:

```
mysql> GRANT ALL on fca.* TO MiNombreUsuario@MiComputadora  
-> IDENTIFIED BY 'MiContraseña';
```

En donde *MiNombreUsuario* es el nombre de usuario asignado dentro del contexto de *MySQL*, *MiComputadora* es el nombre o la dirección *IP* de la computadora desde la que nos conectamos al servidor *MySQL*, y *MiContraseña* es la contraseña que se nos ha asignado, igualmente, dentro del ambiente de *MySQL* exclusivamente. Ambos, nombre de usuario y contraseña no tienen nada que ver con el nombre de usuario y contraseña manejados por el sistema operativo (si es el caso).

I .13.4 Crear una Base de Datos

Si el administrador creó la base de datos al momento de asignar los permisos, podemos hacer uso de ella. De otro modo, nosotros debemos crearla:

```
mysql> USE fca  
ERROR 1049: Unknown database 'fca'  
mysql>
```

El mensaje anterior indica que la base de datos no ha sido creada, por lo tanto necesitamos crearla.

```
mysql> CREATE DATABASE fca;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE fca  
Database changed  
mysql>
```

Bajo el sistema operativo *Unix*, los nombres de las bases de datos son sensibles al uso de mayúsculas y minúsculas (no como las palabras clave de *SQL*), por lo tanto debemos de tener cuidado de escribir correctamente el nombre de la base de datos. Esto es cierto también para los nombres de las tablas.

Al crear una base de datos no se selecciona ésta de manera automática; debemos hacerlo de manera explícita, por ello usamos el comando *USE* en el ejemplo anterior. La base de datos se crea sólo una vez, pero nosotros debemos seleccionarla cada vez que iniciamos una sesión con *mysql*.

I.13.5 Crear una Tabla

Por ahora nuestra Base de datos ha sido creada sin contenido alguno, como lo indica el comando *SHOW TABLES*:

```
mysql> SHOW TABLES;  
  
Empty set (0.00 sec)
```

En principio, necesitamos una tabla que contenga un registro para cada una de nuestro Maestros. Ésta puede ser una tabla llamada Maestros y debe contener por lo menos el nombre de cada uno de nuestros maestros. Debido a que el nombre en sí puede variar y ser muy largo en algunos casos, la tabla debe contener otros campos de información para almacenar el Apellido Paterno y Apellido Materno.

Así mismo, también será necesario contar con información para localizar a cada Maestro y mantener contacto con él. Esto se puede agregando ciertos campos como, teléfono, correo y dirección.

¿Y qué sucede con la edad? Esto puede ser también de interés, pero no es una buena idea almacenar este dato en la base de datos. La edad cambia conforme pasa el tiempo, lo cual significa que debemos de actualizar los registros frecuentemente. En vez de esto, es una mejor idea guardar un valor fijo, tal como la fecha de nacimiento. Entonces, cuando necesitemos la edad, la podemos calcular como la diferencia entre la fecha actual y la fecha de nacimiento. *MySQL* proporciona funciones para hacer operaciones entre fechas, así que no hay ningún problema.

Al almacenar la fecha de nacimiento en lugar de la edad podremos generar felicitaciones y así mantener cercanía con nuestro Maestro.

Usaremos la sentencia *CREATE TABLE* para indicar como estarán conformados los registros de nuestros Maestros.

```
CREATE TABLE `maestros` (  
  `nombre` varchar(50) NOT NULL,  
  `apellido_paterno` varchar(50) NOT NULL,  
  `apellido_materno` varchar(50) default NULL,  
  `email` varchar(50) NOT NULL,  
  `telefono` varchar(50) default NULL,  
  `direccion` varchar(50) default NULL,  
  `cp` int(5) default NULL,  
  `fecha` varchar(25) default NULL,  
);
```

VARCHAR es una buena elección para estos campos ya que los valores que almacenarán son de longitud variable. No es necesario que la longitud de estas columnas sea la misma, ni tampoco que sea de 50. Se puede especificar cualquier longitud entre 1 y 255, lo que se considere más adecuado.

Si resulta que la elección de la longitud de los campos que hemos hecho no resultó adecuada, *MySQL* proporciona una sentencia *ALTER TABLE* que nos puede ayudar a solventar este problema.

Para el campo *cp* los valores que almacenará no serán de longitud variable ya que el código postal siempre es conformado por 5 números, es por eso que *INT 5* es una buena elección.

El uso del tipo de dato *DATE* para la fecha puede resultar obvio, pero en la experiencia personal, *VARCHAR* permitirá almacenar año, mes, día, hora, minutos y segundos. *PHP* contiene funciones como *SUBSTR* que devuelve la porción de cadena especificada por los parámetros comienzo y longitud.

Ahora que hemos creado la tabla, la sentencia *SHOW TABLES* debe producir algo como:

```
mysql> SHOW TABLES;
```

```
+-----+
```

```
| Tables_in_fca |
```

```
+-----+
```

```
| maestros      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql>
```

Para verificar que la tabla fue creada como nosotros esperábamos, usaremos la sentencia *DESCRIBE*:

```
mysql> DESCRIBE fca;
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(50)	YES		NULL	
apellido_materno	varchar(50)	YES		NULL	
apellido_paterno	varchar(50)	YES		NULL	
email	varchar(50)	YES		NULL	
telefono	varchar(30)	YES		NULL	
direccion	varchar(50)	YES		NULL	
cp	int(5)	YES		NULL	
fecha	varchar(25)	YES		NULL	

```
8 rows in set (0.00 sec)
```

Así mismo podremos hacer uso de la sentencia *DESCRIBE* en cualquier momento, por ejemplo, si olvidamos los nombres ó el tipo de las columnas en la tabla.

I.13.6 Ingresar Datos

Para ingresar datos en una Tabla existen dos formas de hacerlo, mediante la sentencia *INSERT* y *LOAD DATA*.

Para hacer uso de la sentencia *LOAD DATA* se debe crear un archivo de texto "maestros.txt" que contenga un registro por línea con valores separados por tabuladores, cuidando que el orden de las columnas sea el mismo que utilizamos en la sentencia *CREATE TABLE*.

Para valores que no conozcamos podemos usar valores nulos (*NULL*). Para representar estos valores en nuestro archivo debemos usar *W*.

Para cargar el contenido del archivo en la tabla maestros se usará el siguiente comando:

```
mysql> LOAD DATA LOCAL INFILE "maestros.txt" INTO TABLE maestros;
```

La sentencia *LOAD DATA* nos permite especificar cuál es el separador de columnas, y el separador de registros, por default el tabulador es el separador de columnas (campos) y el salto de línea es el separador de registros, que en este caso son suficientes para que la sentencia *LOAD DATA* lea correctamente el archivo "maestros.txt".

Si lo que deseamos es añadir un registro a la vez, entonces debemos hacer uso de la sentencia *INSERT*. En la manera más simple, debemos proporcionar un valor para cada columna en el orden en el cual fueron listados en la sentencia *CREATE TABLE*.

```
mysql> INSERT INTO maestros
```

```
->VALUES('Graciela','Bribiesca','Correa','gbribiesc@servidorr.unam.mx','2000-03-30',NULL);
```

Se debe notar que los valores de cadenas y fechas deben estar encerrados entre comillas. También, con la sentencia *INSERT* podemos insertar el valor *NULL* directamente para representar un valor nulo, un valor que no conocemos.

De este ejemplo, debemos ser capaces de ver que es un poco más la tarea que se tiene que realizar si inicialmente cargamos los registros con varias sentencias *INSERT* en lugar de una única sentencia *LOAD DATA*.

I .13.7 Consulta de información de una tabla

La sentencia *SELECT* es usada para obtener la información guardada en una tabla. La forma general de esta sentencia es:

```
SELECT LaInformaciónQueDeseamos FROM DeQueTabla WHERE  
CondiciónASatisfacer;
```

Aquí, *LaInformaciónQueDeseamos* es la información que queremos ver. Esta puede ser una lista de columnas, o un *** para indicar "todas las columnas". *FROM DeQueTabla* indica el nombre de la tabla de la cual vamos a obtener los datos. La clausula *WHERE* es opcional. Si está presente, la *CondiciónASatisfacer* especifica las condiciones que los registros deben satisfacer para que puedan ser mostrados.

La manera más simple de la sentencia ***SELECT*** es cuando se recuperan todos los datos de una tabla:

```
mysql> SELECT * FROM maestros;
```

Esta forma del ***SELECT*** es útil si deseamos ver los datos completos de la tabla, por ejemplo, para asegurarnos de que están todos los registros después de la carga de un archivo.

I .13.8 Actualizar la información de una tabla

Una vez que hay datos en la tabla, podríamos tener la necesidad de modificar los mismos. Para hacerlo, utilizamos el comando *UPDATE*. La sintaxis para esto es,

```
UPDATE nombre_tabla SET columna = nuevo valor WHERE condición;
```

I .13.9 Eliminar la información de una tabla

A veces podemos desear deshacernos de los registros de una tabla. Para ello, utilizamos el comando *DELETE FROM*. La sintaxis para esto es,

```
DELETE FROM nombre_tabla WHERE condición;
```

I .13.10 Eliminar toda la información de una tabla

Que debemos hacer si se desea eliminar todos los datos de una tabla pero no de la tabla en sí. Para esto, podemos utilizar el comando *TRUNCATE TABLE*. La sintaxis para *TRUNCATE TABLE* es

```
TRUNCATE TABLE "nombre_tabla"
```

I .13.11 Eliminar una tabla

A veces podemos decidir que necesitamos eliminar una tabla en la base de datos por alguna razón. De hecho, sería problemático si no podemos hacerlo ya que esto crearía una pesadilla de mantenimiento para *DBA*. Afortunadamente, *SQL* nos permite hacerlo, ya que podemos utilizar el comando *DROP TABLE*. La sintaxis para *DROP TABLE* es

```
DROP TABLE "nombre_tabla"
```

I .14 PHP

PHP es un lenguaje de programación que permite el uso de variables, sentencias condicionales, *bucles*, funciones y que a diferencia de otros lenguajes se ejecuta en el servidor. Por esto nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. Entendiendo la estructura el programa *PHP* es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página *HTML*. (Sklar, 2004)

Al ser *PHP* un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas *PHP* funcionen, el servidor donde están alojadas debe soportar *PHP*.

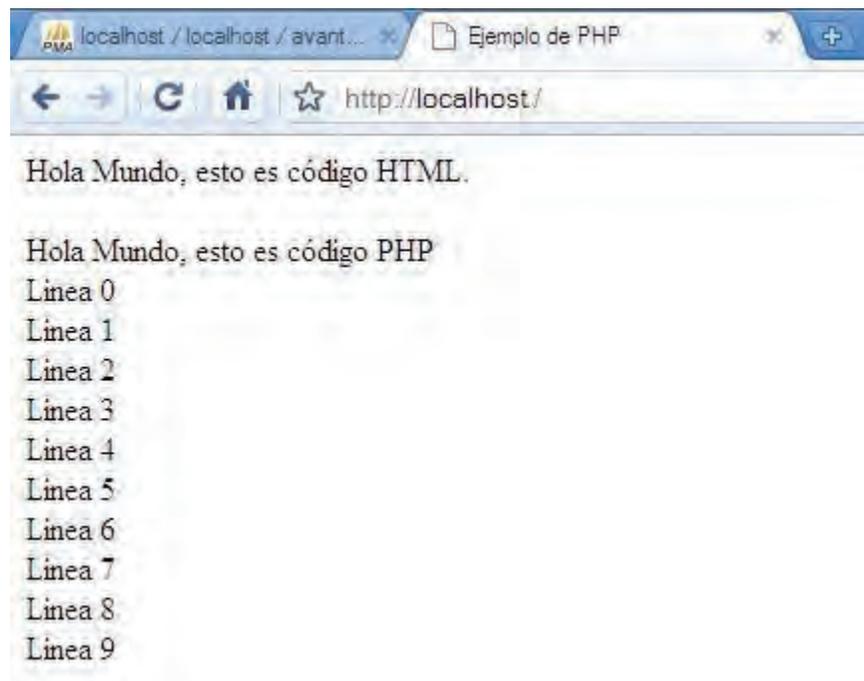
I .14.1 “Hola Mundo”

La ventaja que tiene *PHP* sobre otros lenguajes de programación que se ejecutan en el servidor, es que nos permite intercalar las sentencias *PHP* en las páginas *HTML*, es un concepto algo complicado de entender si no se ha visto nunca como funciona unas páginas *PHP* o *ASP*.

Para ejemplificar cómo es posible intercalar código *HTML* con código *PHP*, creemos una aplicación que ejecute primero el código *HTML* y enseguida el código *PHP* con un ciclo *FOR* que imprima el número de línea.

```
<!-- Mi Primer Aplicación PHP -->
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
    Hola Mundo, esto es código HTML.
    <BR><BR>
    <?php echo " Hola Mundo, esto es código PHP <br>";
    for($i=0;$i<10;$i++)
    {
        echo "Linea ".$i."<br>";
    }
    ?>
</body>
</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



I.14.2 Uso de Variables con PHP

Una variable es un contenedor de información, en el que podemos meter números enteros, números decimales y caracteres. El contenido de las variables se puede leer y se puede cambiar durante la ejecución de una página *PHP*.

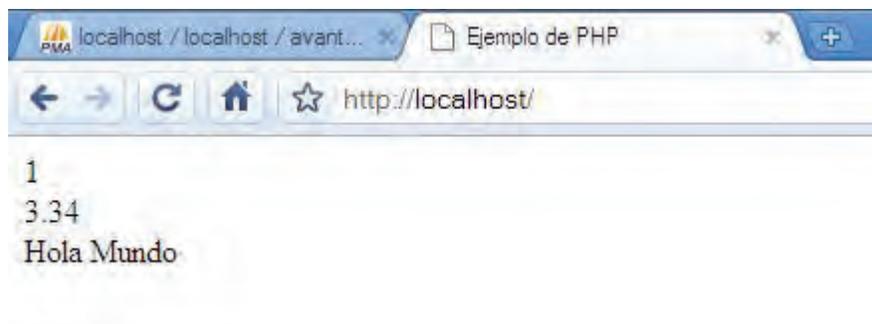
Existen 2 tipos de variables, las variables locales que solo pueden ser usadas dentro de funciones y las variables globales que tienen su ámbito de uso fuera de las funciones, podemos acceder a una variable global desde una función con la instrucción `global nombre_variable;`

En *PHP* todas las variables comienzan con el símbolo del dólar `$` y no es necesario definir una variable antes de usarla. Tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres.

Para ejemplificar el uso de variables creamos una aplicación en donde definamos tres variables, \$a, \$b y \$c y con la instrucción ECHO hemos impreso el valor que contenían, insertando un salto de línea entre ellas.

```
<!-- Uso de variables con PHP -->
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
    <?php
    $a = 1;
    $b = 3.34;
    $c = "Hola Mundo";
    echo $a,"<br>",$b,"<br>",$c; ?>
</body>
</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



I .14.3 Uso de Operadores con PHP

I .14.3.1 Operadores Aritméticos

Los operadores de *PHP* son muy parecidos a los de *C* y *JavaScript*, si se conocen estos lenguajes resultara más fáciles de reconocer.

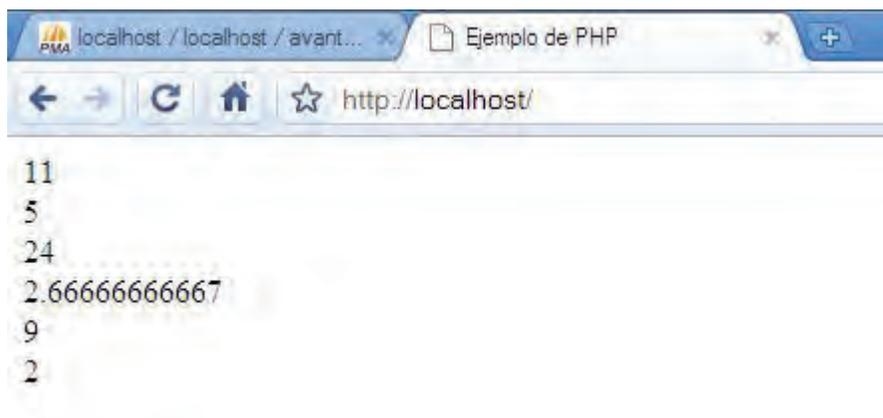
Estos son los operadores que se pueden aplicar a las variables y constantes numéricas.

Operador	Nombre	Ejemplo	Descripción
+	Suma	5 + 6	Suma dos números
-	Resta	7 - 9	Resta dos números
*	Multiplicación	6 * 3	Multiplica dos números
/	División	4 / 8	Divide dos números
%	Módulo	7 % 2	Devuelve el resto de dividir ambos números, en este ejemplo el resultado es 1
++	Incremento	\$a++	Suma 1 al contenido de una variable.
--	Disminución	\$a--	Resta 1 al contenido de una variable.

Para ejemplificar el uso de Operadores Numéricos crearemos una aplicación en la cual se definirán dos variables A y B y se les realizaran operaciones de suma, resta, multiplicación, división, de Módulo, incremento y disminución.

```
<!-- Uso de operadores aritméticos con PHP -->
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
    <?php
    $a = 8;
    $b = 3;
    echo $a + $b,"<br>";
    echo $a - $b,"<br>";
    echo $a * $b,"<br>";
    echo $a / $b,"<br>";
    $a++; echo $a,"<br>";
    $b--; echo $b,"<br>";
    ?>
</body>
</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



I .14.3.2 Operadores de Comparación

La comparación entre valores permitirá al usuario hacer una correcta toma de decisiones si los valores son correctamente utilizados. Algunos de estos operadores de *PHP* son:

Operador	Nombre	Ejemplo	Devuelve cierto (1) cuando:
==	Igual	\$a == \$b	\$a es igual \$b
!=	Distinto	\$a != \$b	\$a es distinto \$b
<	Menor que	\$a < \$b	\$a es menor que \$b
>	Mayor que	\$a > \$b	\$a es mayor que \$b
<=	Menor o igual	\$a <= \$b	\$a es menor o igual que \$b
>=	Mayor o igual	\$a >= \$b	\$a es mayor o igual que \$b

Para ejemplificar el uso de Operadores Numéricos crearemos una aplicación en la cual se definirán tres variables A, B y C donde se realizaran comparaciones entre cada variable.

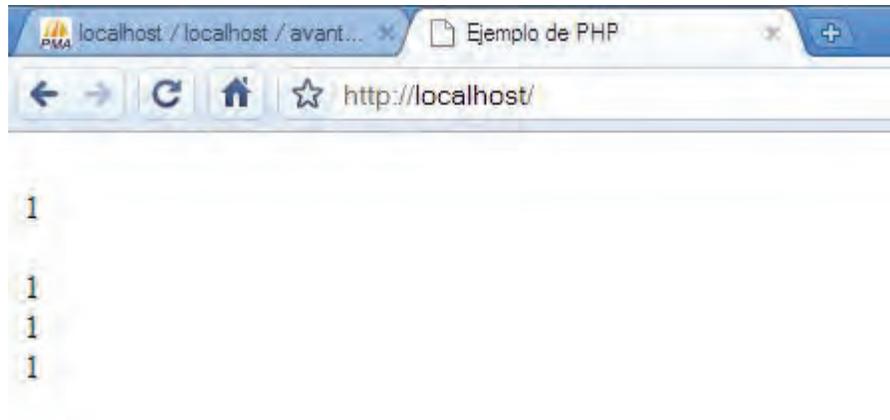
```
<!-- Uso de operadores de comparación con PHP -->
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
    <?php
    $a = 8;
    $b = 3;
    $c = 3;
    echo $a == $b,"<br>";
    echo $a != $b,"<br>";
    echo $a < $b,"<br>";
    echo $a > $b,"<br>";
```

```

    echo $a >= $c, "<br>";
    echo $b <= $c, "<br>";
    ?>
</body>
</html>

```

Ejecutado este ejemplo, mostrará lo siguiente:



I .14.3.3 Operadores Lógicos

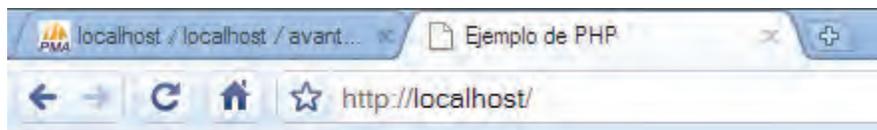
Los operadores lógicos son usados para evaluar varias comparaciones, combinando los posibles valores de estas.

Operador	Nombre	Ejemplo	Devuelve cierto (1) cuando:
&&	Y	(7>2) && (2<4)	Cuando ambas condiciones son verdaderas.
And	Y	(7>2) and (2<4)	Cuando ambas condiciones son verdaderas.
	O	(7>2) (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera.
Or	O	(7>2) or (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera.
!	No	!(7>2)	Niega el valor de la expresión.

Para ejemplificar el uso de Operadores Numéricos crearemos una aplicación en la cual se definirán tres variables A, B y C donde se realizaran comparaciones entre cada variable.

```
<!-- Uso de operadores lógicos con PHP -->
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
<?php
    $a = 8;
    $b = 3;
    $c = 3;
    echo ($a == $b) && ($c > $b), "<br>";
    echo ($a == $b) || ($b == $c), "<br>";
    echo !($b <= $c), "<br>";
?>
</body>
</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



1

I .14.4 Uso de Instrucciones de programación con PHP

I .14.4.1 Sentencias Condicionales

Las sentencias condicionales nos permiten ejecutar o no unas ciertas instrucciones dependiendo del resultado de evaluar una condición. Las más frecuentes son las instrucciones IF y SWITCH.

I .14.4.1.1 Sentencia IF – ELSE

La sentencia if ejecuta una serie de instrucciones u otras dependiendo de la condición que le pongamos. Probablemente sea la instrucción más importante en cualquier lenguaje de programación.

```
<?php
if (condición) {
    Sentencias a ejecutar cuando la condición es cierta
} else {
    Sentencias a ejecutar cuando la condición es cierta
}
?>
```

Para ejemplificar el uso de la sentencia IF crearemos una aplicación en la cual se definirán dos variables A y B donde se realizarán condiciones entre cada variable.

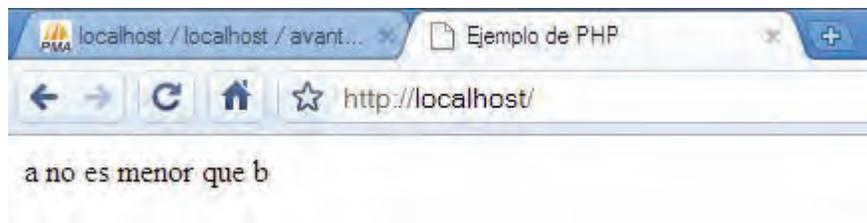
```
<!-- Uso de sentencia IF con PHP -->
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
    <?php
    $a = 8;
    $b = 3;
    if ($a < $b) {
        echo "a es menor que b";
    }
    ?>
```

```

        } else {
            echo "a no es menor que b";
        }
    ?>
</body>
</html>

```

Ejecutado este ejemplo, mostrará lo siguiente:



I .14.4.1.2 Sentencia SWITCH - CASE

La sentencia switch es similar a una serie de sentencias IF en la misma expresión.

En muchas ocasiones, se quiere comparar la misma variable (o expresión) con muchos valores diferentes, y ejecutar una parte de código distinta dependiendo del valor de dicha variable (o expresión). Para eso sirve la sentencia switch.

La sintaxis de la sentencia switch es la siguiente:

```

switch(expr){
    case valor1:
        sentencia1;
        break;
    case valor2:
        sentencia2;
        break;
    .
    .
    default:
        sentencia;
}

```

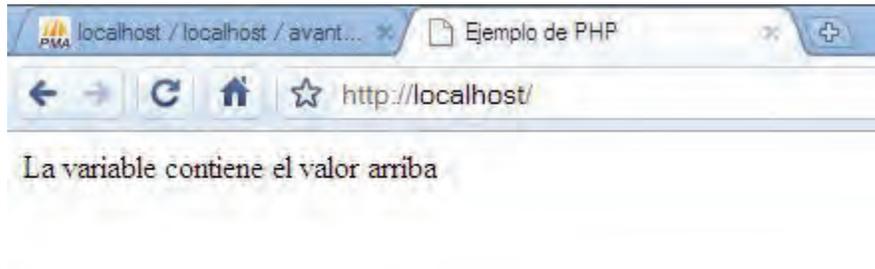
Para ejemplificar el uso de la sentencia SWITCH crearemos una aplicación en la cual se definirán dos variables A y B donde se realizaran condiciones entre cada variable.

```
<!-- Uso de sentencia SWITCH/CASE con PHP --><html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
<?php
    $posicion = "arriba";
    switch($posicion) {
        case "arriba":    // Bloque 1
            echo "La variable contiene";
            echo " el valor arriba";
            break;
        case "abajo":// Bloque 2
            echo "La variable contiene";
            echo " el valor abajo";
            break;
        default:    // Bloque 3
            echo "La variable contiene otro valor";
            echo " distinto de arriba y abajo";
    }
?>
```

```
</body>
```

```
</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



I .14.4.2 Bucles

I .14.4.2.1 Sentencia WHILE

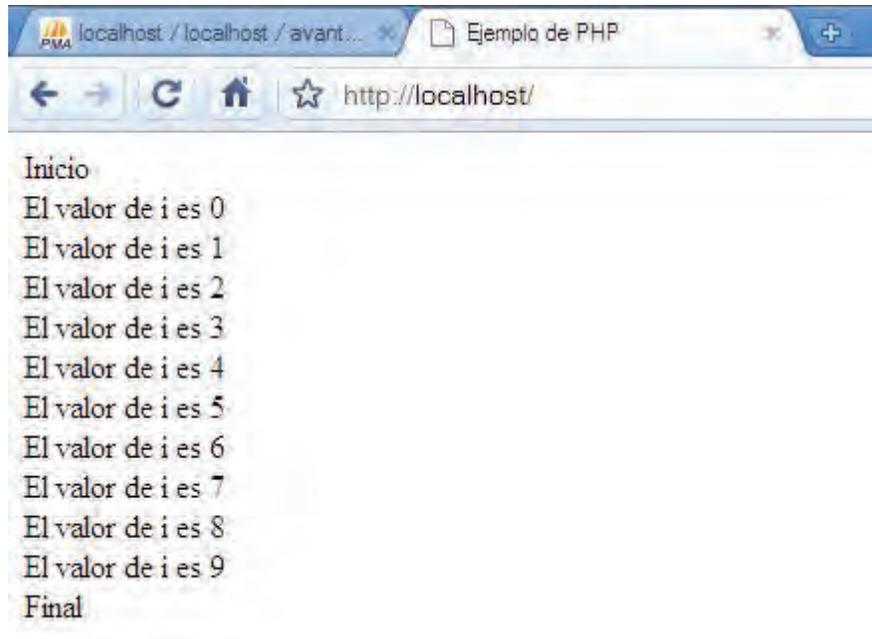
Mientras la condición sea cierta se reiterará la ejecución de las instrucciones que están dentro del while. La sintaxis de la sentencia WHILE es la siguiente:

```
<?php  
while (condición)  
{  
    instrucciones a ejecutar  
}  
?>
```

Para ejemplificar el uso de la sentencia SWITCH crearemos una aplicación en la cual se declarara la variable "i" y se inicializara con el valor 0 y se ejecutara una sentencia hasta que se cumpla cierta condición. Hasta que "i" valga 10.

```
<!-- Uso de sentencia WHILE con PHP --><html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
    Inicio<BR>
    <?php
        $i=0;
        while ($i<10)
        {
            echo "El valor de i es ", $i,"<br>";
            $i++;
        }
    ?>
    Final<BR>
</body>
</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



I.14.4.2.2 Sentencia FOR

La instrucción for es la instrucción de *bucles* más completa. En una sola instrucción nos permite controlar todo el funcionamiento del bucle.

El primer parámetro del for, es ejecutado la primera vez y sirve para inicializar la variable del bucle, el segundo parámetro indica la condición que se debe cumplir para que el bucle siga ejecutándose y el tercer parámetro es una instrucción que se ejecuta al final de cada iteración y sirve para modificar el valor de la variable de iteración.

La sintaxis de la sentencia FOR es la siguiente:

```
<?php
for (inicial ; condición ; ejecutar en iteración)
{
    instrucciones a ejecutar
} ?>
```

Para ejemplificar el uso de la sentencia FOR crearemos una aplicación en la cual se declarara la variable "i" y se inicializara con el valor 0 y se ejecutara una sentencia hasta que se cumpla cierta condición. Hasta que "i" valga 10.

```
<!-- Uso de sentencia FOR con PHP --><html>

<head>

    <title>Ejemplo de PHP</title>

</head>

<body>

Inicio<BR>

<?php

    for($i=0 ; $i<10 ; $i++)

    {

        echo "El valor de i es ", $i,"<br>";

    }

?>

Final<BR>

</body>

</html>
```

Ejecutado este ejemplo, mostrará lo siguiente:



I .14.5 ENVIAR y RECIBIR datos con PHP

Es importante recordar que todo bloque de código *PHP* debe estar dentro de `<?php`(para abrir) y `?>`(para cerrar) o simplemente de una forma más corta `<?<CÓDIGO_PHP> ?>`, muy parecido a las etiquetas en *ASP* `<%<CÓDIGO_ASP> %>`. Para ejemplificar el envío y recepción de datos, crearemos una aplicación que reciba datos de un usuario y el sistema envíe una respuesta en pantalla sobre los datos escritos en una misma página *PHP* mediante una condición:

```
<html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
<div align="left">

<?php
if (isset($_POST['muestra'])) //Inicia condicion
```

```

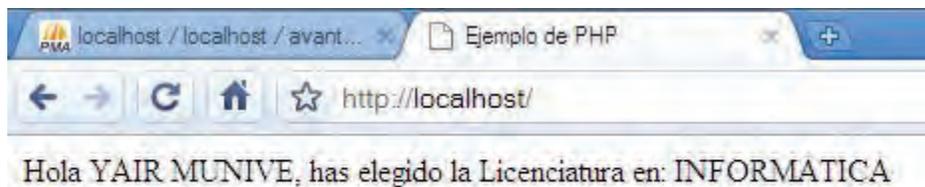
    {
        echo 'Hola ' . $_POST['nombre'] . ', has elegido la Licenciatura en: ' .
$_POST['licenciatura']. ";
    } else { //Si no la cumple inicia siguiente condición
?>
</div>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'];?>">
    <div align="left">Bienvenido Maestro de la FCA
    <br><br>¿Cuál es su nombre?
    <input type="text" name="nombre" />
    <br><br>¿Elige una Licenciatura?
    <select name="licenciatura">
        <option value="NULL">-----</option>
        <option value="INFORMATICA">INFORMATICA</option>
        <option value="CONTADURIA">CONTADURIA</option>
        <option value="ADMINISTRACION">ADMINISTRACION</option>
    </select>
    <br>
    <br>
    <input type="submit" name="muestra" value="Enviar" />
    </div>
</form>
<div align="left">
    <?php
        } // Fin de condicion else
?>
    </div>
</body>
</html>

```

Ejecutado este ejemplo, mostrará lo siguiente:



Ingresando la Información que requiere esta aplicación, mostrará lo siguiente:



En este código es posible observar las siguientes características:

- Las variables enviadas por un formulario utilizando el método POST, son recibidas en el lenguaje dentro de la matriz `$_POST`, lo cual facilita la obtención de este tipo de datos. Este mismo método es utilizado por el lenguaje para todas las fuentes de información en una aplicación web, tales como cookies en la matriz `$_COOKIES`, variables de URL en `$_GET` (que en formularios puede servir para guardar los datos), variables de sesión utilizando `$_SESSION`, y variables del servidor y del cliente por medio de la matriz `$_SERVER`.
- El código *PHP* está incrustado dentro del *HTML* e interactúa con el mismo, lo que permite diseñar la página Web en un editor común de *HTML* y añadir el código dinámico dentro de las etiquetas `<?php ?>`.

- El resultado muestra y oculta ciertas porciones del código *HTML* en forma condicional.
- Es posible utilizar funciones propias del lenguaje para aplicaciones Web como `htmlspecialchars()`, que convierte los caracteres que tienen algún significado especial en el código *HTML* o que podrían desplegarse erróneamente en el navegador como acentos o diéresis, en sus equivalentes en formato *HTML*.

Todas las variables en *PHP* llevan el símbolo de dólar como prefijo (ejemplo: `$variable1`, `$variable2`, `$variable3...`, `$variableN`), los tipos de variable no es necesario declararlas, ya que a medida que se usan en la aplicación el servidor reconoce de que tipo son. Para mostrar una cadena (en inglés string) debe estar dentro de comillas dobles o simples (ejemplo: "Hola Mundo", 'Lo que quiero mostrar'). Cabe destacar que si se desea mostrar el símbolo " o ' debe encerrarse en el otro tipo de comillas ("...'...", '...'...') o usarse un escape (`\'`, `\"`).

Toda línea de instrucción siempre termina en un punto y coma (;), al igual que el lenguaje C.

Para insertar un comentario de una sola línea, debe empezar por `//` o por `#`. El resto de la línea es tratado entonces como un comentario. Para insertar un bloque de comentario, de una o más líneas, se utiliza la combinación `/*` y `*/`, por ejemplo: `/* <COMENTARIOS> */`

II ANTECEDENTES

II.1 IIEc (Instituto de Investigaciones Económicas)

II.1.1 Fundación IIEc

El Instituto de Investigaciones Económicas de la Universidad Nacional Autónoma de México (UNAM) se fundó a fines de noviembre de 1940 gracias a las gestiones del maestro Jesús Silva Herzog, director de la Escuela Nacional de Economía (ENE), ante el rector el doctor Gustavo Baz. El instituto se dividió en dos departamentos, el de Investigaciones Económicas y el de Laboratorios, cuyo antecedente inmediato fue el Laboratorio de Organización e Investigación Industrial, creado en 1939 cuando el licenciado Mario Souza dirigía la ENE. El propósito de la nueva institución era apoyar las labores docentes y asesorar a los estudiantes en la práctica profesional de la investigación económica. Pasados los años, la presencia del Instituto se amplió, además de colaborar en la enseñanza de la economía y la dirección de tesis, fungió como órgano de consulta en la revisión y formulación de los planes de estudio de la Escuela de Economía y en la elaboración de su reglamento interno, y tuvo a su cargo la publicación de la revista *Investigación Económica* –órgano oficial de la Escuela– y la organización de los cursos de invierno.

II.1.2 Autonomía

El Instituto fue creado en 1940 como parte de la Escuela Nacional de Economía por iniciativa del entonces director de dicha Escuela, el profesor y doctor Jesús Silva Herzog.

El Instituto se dividió en dos departamentos: el Departamento de Laboratorios, que quedó bajo la dirección del profesor Manuel Bravo Jiménez y el Departamento de

Investigaciones, que quedó a cargo del connotado investigador Miguel Othón de Mendizábal.

La creación del Instituto tuvo lugar en una época de grandes problemas, ya que eran los años de la segunda guerra mundial, antecedida por la Gran Depresión de 1929-1932 y la intensificación de la lucha social que propició en México el ascenso del gobierno revolucionario del general Lázaro Cárdenas. En este contexto se hacía necesario que los economistas realizaran investigaciones y análisis que permitieran entender mejor la realidad y encontrar solución a los problemas que se enfrentaban.

Sin embargo, la terminación de la segunda guerra mundial en 1945, la reconversión a tiempos de paz y los programas de industrialización que se pusieron en marcha plantearon nuevos problemas a los que era necesario encontrar explicación y solución.

La agudización del conflicto entre atender la gran carga de docencia y la necesidad primordial de realizar investigaciones condujo a que los investigadores plantearan ante el rector de la UNAM la conveniencia de independizar al Instituto respecto a la Escuela de Economía, gestiones que no tuvieron éxito hasta 1967-1968, cuando el Instituto logró adquirir la situación de Instituto independiente.

II.1.3 Actualidad

El Instituto de Investigaciones Económicas (IIEc) cuenta con una importante producción de libros individuales y colectivos, la mayoría publicados en coedición con universidades nacionales e internacionales, editoriales privadas y dependencias gubernamentales. En el año 2004 se inició la publicación cuatrimestral de la revista *ECONOMÍAUnam* en coedición con la Facultad de Economía, la FES-Acatlán y la FES-Aragón. *Problemas del Desarrollo*, *Revista Latinoamericana de Economía*, creada en 1969, es el órgano oficial del Instituto.

A partir del año 2006, además de la versión mexicana, se publica la versión Cono Sur, coeditada con el Consejo Latinoamericano de Ciencias Sociales y cuya distribución abarca más de 18 países de América Latina, Canadá y España.

Anualmente el Instituto organiza seminarios en los que se debaten temas de vanguardia de la realidad económica nacional e internacional; en ellos participan especialistas del mismo Instituto, de otras instituciones académicas –públicas y privadas-, representantes del gobierno y de la sociedad civil. Asimismo, el Instituto mantiene convenios de intercambio académico con instituciones de educación superior de diversos países del mundo, lo que ha fortalecido su presencia y liderazgo en el ámbito de su competencia.

El IIEc desarrolla programas académicos con los posgrados de Economía y de Estudios Latinoamericanos, lo cual incrementa cada año su plantilla de tutores. Sus académicos también contribuyen en otros posgrados de la UNAM, como docentes, participantes en comités tutorales y evaluadores de los programas de especialidad, función que se amplía a diferentes universidades del país.

El Instituto, por tradición, fomenta el reconocimiento a la investigación económica, particularmente al otorgar el Premio Anual de Investigación Económica “Maestro Jesús Silva Herzog” y el Premio “Doctor Ernest Feder”. El primero se entrega en el marco del Seminario de Economía Mexicana y el segundo, en el Seminario Internacional de Economía Agrícola.

Algunos de sus académicos han sido distinguidos con el Premio Universidad Nacional, la Distinción Universidad Nacional para Jóvenes Académicos y el premio Sor Juana Inés de la Cruz.

En la actual administración, el trabajo académico se ha organizado en torno a 14 unidades de investigación: Economía Mundial, Industrial, del Sector Primario, Urbana y Regional, Aplicada, del Trabajo y de la Tecnología, Fiscal y Financiera, de la Educación, la Ciencia y la Tecnología, del Sector Energético, Historia Económica, Economía Política del Desarrollo, del Medio Ambiente, del

Conocimiento y Desarrollo, y Estudios Hacendarios y del Sector Público. Además, en su estructura académica se encuentra el departamento de Análisis Macroeconómico de Coyuntura y Prospectiva, con especial énfasis en el desarrollo de la economía mexicana.

Los representantes de cada unidad de investigación integran el Consejo Académico, cuya función primordial es construir la política académica, con la ulterior aprobación del Consejo Interno. Ambos consejos coadyuvan con la dirección del Instituto para elaborar el programa anual de investigación, y para establecer los lineamientos académicos necesarios para el funcionamiento óptimo de todos sus programas institucionales. También se realizan actividades de extensión a través del Centro de Educación Continua, responsable de llevar a cabo diplomados presenciales y a distancia sobre temas de actualidad.

La estructura académica permite, hoy más que nunca, la formación de recursos humanos de alto nivel, en distintas modalidades: prestadores de servicio social, becarios nacionales y extranjeros (algunos de ellos provenientes de la Secretaría de Relaciones Exteriores) y otros de proyectos con financiamiento: PAPIIT, PAPIME, CONACYT.

Cada jueves se transmite por Radio UNAM, el programa "Momento Económico", espacio plural y abierto que difunde y recupera debates sobre aquellos temas de coyuntura de interés nacional e internacional que involucran a la sociedad mexicana.

El 20 de febrero de 2007 se inauguró la nueva sede del Instituto, sita en el circuito Mario de la Cueva s/n, en la Ciudad de la Investigación en Humanidades, dentro de la Ciudad Universitaria, contando con instalaciones de vanguardia que incluyen 120 cubículos, seis salas de trabajo, 12 salas de apoyo a la investigación, salas de lectura y acervo bibliotecario, librería, auditorio, dos salas de diplomados y seminarios, una sala de videoconferencias, dos aulas para docencia, una sala de cómputo y otra para impartir cursos propios, una sala de consejo interno y oficinas para la dirección y para las tres secretarías (Académica, Técnica y Administrativa).

Con su quehacer fundamental de investigación y diversas actividades académicas como: publicaciones, seminarios, coloquios, mesas redondas, talleres, presentaciones de libros, cursos, conferencias y otras, el IIEc cumple con los altos fines universitarios y con los retos y compromisos que la sociedad actual plantea.

II.2 METODOLOGÍA A USAR

En el análisis y diseño de sistemas la metodología más comúnmente empleada es la que se conoce como Ciclo de vida clásico, la cual sugiere un enfoque sistemático secuencial en etapas e análisis, diseño, generación de código, pruebas y mantenimiento.

Para la elaboración del proyecto de un sitio Web la tendencia es utilizar la metodología de ciclo de vida, debido a que obliga a planificar cada paso desde el principio, lo que puede resultar contraproducente debido a que en un proyecto Web existen muchas incertidumbres sobre lo que realmente se necesita, como pueden ser puntos de vista sobre diseño gráfico y en algunos casos diseño e implementación de nuevos códigos. (Tomas, 2001)

Por lo que la elección del ciclo de vida como metodología para el desarrollo de este sitio Web o sistema, no resulta muy adecuado si se toma en cuenta que cada grafica deberá ser evaluada por los investigadores y encargados de este proyecto y cada uno pedirá modificaciones respecto a la propuesta desarrollada.

Por lo que la opción viable considerando los continuos cambios que sufrirá el proyecto es basarse en una mezcla de las metodologías de ciclo de vida con el de Prototyping (Prototipo) y la metodología en espiral.

El paradigma de la construcción de prototipos comienza con la recolección de requisitos. El desarrollador y los clientes (bien sean internos o externos) definen los objetivos globales, identifican los requisitos y las áreas donde es necesaria mas definición sugiriendo un diseño rápido.

El diseño rápido se centra en una presentación de los aspectos que serán visibles para el cliente/usuario y se utiliza para refinar los requisitos del desarrollo. La iteración ocurre cuando el prototipo se va modificando para satisfacer las necesidades del cliente.

El modelo en espiral es un modelo de procesos de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con el aspecto lineal del ciclo de vida clásico. El modelo en espiral se desarrolla en una serie de versiones incrementales.

Este modelo se divide en actividades de marco de trabajo, denominadas regiones de tareas que pueden ser comunicación con el cliente y planificación, análisis de riesgo, construcción de prototipos y evaluación.

La planeación de un sitio de acuerdo a esta metodología, muchas veces no especifica la etapa de prueba y publicación debido a que solo se enfoca en el modo en que se realizará el diseño del sitio propuesto y siempre se desconoce en qué momento o número de vueltas a la espiral tendrá la versión final del sitio.

Por lo que para la etapa de pruebas y publicación para este proyecto se incluirán en la etapa de evaluación de nuestra espiral. En esta prueba se incluirá a cualquier persona para que realice pruebas de contenido, diseño y visualización o simplemente de su Visto Bueno. (Pressman, 2002)

Una vez que se hayan detectado u corregido los posibles errores de navegación, contenido y programación se procederá con su evolución iteradita de dicho ciclo.

El mantenimiento del sitio será diario debido a que los indicadores económicos de petróleo cambian constantemente de acuerdo a su precio de compra y venta, por lo que una vez obtenido el valor del precio del barril, este deberán ser actualizado en la base de datos por el administrador del sitio.

II.3 ANÁLISIS DEL SITIO WEB DINAMICO

II.3.1 Análisis de Requerimientos

Hoy en día el marco del Programa Académico del Instituto de Investigaciones Económicas de la Universidad Nacional Autónoma de México, se plantea la creación de una Revista Digital habilitando al usuario una sección de apoyo y aprendizaje contemplando el use de graficas dinámicas de indicadores de petróleo.

La aparición de Tecnologías Telemáticas (TT), que integran cómputo, informática y comunicaciones, han generado un cambio sustancial de la base material de la sociedad a un ritmo acelerado y la penetración que han tenido se refleja en un efecto económico que propicia la interdependencia global.

Por otra parte los indicadores son estadísticas, serie estadística o cualquier forma de indicación que nos facilita estudiar dónde estamos y hacia dónde nos dirigimos con respecto a determinados objetivos y metas, así como evaluar programas específicos y determinar su impacto.

Problemáticas tales como la brecha digital, la cibercultura y la geografía digital, que describen rupturas sociales debido a las diferencias entre la disponibilidad, acceso, uso, apropiación y en su caso adaptación de las TT, son algunos de los temas que preocupan ahora a la sociedad.

Ante este entorno, organismos internacionales plantean la adopción y apropiación de estas tecnologías, a las comunidades de aprendizaje y la gestión del conocimiento como estrategias para incrementar la capacidad del individuo y la sociedad para aprovechar las oportunidades y enfrentar las amenazas que representa este ambiente.

II.3.2 Detección de Necesidades

La creación y desarrollo del sistema que permita manipular los indicadores debe considerar los siguientes aspectos: (mx.yahoo)

- Ser específicos, es decir, estar vinculados con los fenómenos económicos, sociales, culturales o de otra naturaleza sobre los que se pretende actuar; por lo anterior, se debe contar con objetivos y metas claros, para poder evaluar qué tan cerca o lejos nos encontramos de los mismos y proceder a la toma de decisiones pertinentes. Deben mostrarse especificando la meta u objetivo a que se vinculan y/o a la política a la que se pretende dar seguimiento; para cumplirlas se recomienda que los indicadores sean pocos.
- Ser explícitos, de tal forma que su nombre sea suficiente para entender si se trata de un valor absoluto o relativo, de una tasa, una razón, un índice, etc., así como a qué grupo de población, sector económico o producto se refieren y si la información es global o está desagregada por sexo, edad, años o región geográfica.
- Estar disponibles para varios años, con el fin de que se pueda observar el comportamiento del fenómeno a través del tiempo, así como para diferentes regiones y/o unidades administrativas.
- La comparabilidad es un insumo que permite fomentar el desarrollo social o económico de acuerdo con lo que tenemos respecto a los demás o a otros momentos; de hecho, nociones socioeconómicas como producción, pobreza y trabajo sólo son comprensibles en términos relativos.

- Deben ser relevantes y oportunos para la aplicación de políticas, describiendo la situación prevaleciente en los diferentes sectores de gobierno, permitiendo establecer metas y convertirlas en acciones.
- Ser claros, de fácil comprensión para los miembros de la comunidad, de forma que no haya duda o confusión acerca de su significado, y debe ser aceptado, por lo general, como expresión del fenómeno a ser medido.

II.3.3 Soluciones Sugeridas

La programación del sitio Web dinámico contempla una interfaz amigable para el usuario desde el punto de vista de la navegación, es decir desde cualquier parte del sitio puede regresar al menú de inicio y volver a solicitar cualquier operación o movimiento. Esto en función a dos tipos de interfaces distintas. La primera interfaz será de carácter Administrativo, la cual deberá ser manipulada por una única persona que será llamada Administrador; la segunda interfaz, será la del Usuario, la cual podrá ser manipulada por cualquier persona que pueda tener acceso al sitio.

Dentro de la facilidad que debe presentar al usuario es que este pueda identificar mediante un menú, cuales son las operaciones que se le permiten realizar. Ser agradable a la vista y pueda ser actualizado de manera sencilla, según los valores de las mezclas de Petróleo al día.

II.3.3.1 Aclaración y Comprensión del Proyecto

El diseño de este sistema se centra en la graficación de valores de las distintas mezclas de petróleo al día. A su vez las mezclas de petróleo a utilizar serán las siguientes:

- West Texas
- Mezcla Mexicana
- Olmeca
- Istmo
- Maya

Siendo el valor para cada uno de ellos el precio en Dólares por Barril.

Cada registro contendrá el precio por barril para cada mezcla y la fecha para la cual se da de alta el registro. A su vez cada registro será identificado con un ID que le permitirá a los usuarios hacer búsquedas, eliminar, modifica y crear registros.

Por otra parte es necesario hacer una distinción entre los dos tipos de interfaz a desarrollar señalando diferencias, ubicando a cada una de estas de tal manera que se identifiquen y diferencien claramente una de otra.

II.3.3.1.1 Interfaz de Administrador

El desarrollo de esta interfaz contempla un nivel básico de seguridad que le permita al usuario administrador identificarse mediante un usuario y contraseña; esto con la finalidad de mantener segura la información y evitar ser modificada por cualquier persona.

Una vez identificado el usuario como administrador ingresara a un menú en donde se le brindaran las siguientes opciones:

- **Ingresar nuevos registros.-** Esta opción le permitirá al administrador registrar día con día los valores a las distintas mezclas de petróleo por día.
- **Ingresar registros anteriores.-** Esta opción le permitirá al administrador ingresar valores a las distintas mezclas de petróleo a registros anteriores ó atrasados.

- **Modificar registros.-** Esta opción le permitirá al administrador modificar cada uno de los valores que tenga el registro por día.
- **Eliminar registros.-** Esta opción le permitirá al administrador eliminar cualquier registro por día.
- **Búsqueda general.-** Esta opción le permitirá al administrador modificar y eliminar registros mediante una búsqueda general. El administrador deberá ingresar un año de búsqueda y podrá visualizar los meses que comprende ese año. Posteriormente deberá elegir un mes para ese año y así visualizar los 30 registros para ese mes.
- **Búsqueda avanzada.-** Esta opción le permitirá al administrador modificar y eliminar un registro mediante una búsqueda avanzada. El administrador deberá ingresar el año el mes y el día para poder visualizar el registro.

II.3.3.1.2 Interfaz de Usuario

El desarrollo de esta interfaz será diseñada para que cualquier usuario pueda navegar de manera sencilla.

Por razones de seguridad y para mantener la veracidad de la información se deshabilitaran las opciones de modificación y eliminación de registros para este tipo de usuario.

El menú para los usuarios contendrá las siguientes opciones:

- **Graficar.-** Esta opción le permitirá al usuario visualizar a manera de grafica el comportamiento que ha sufrido los precios por barril de petróleo de cada una de las mezclas. Esta opción graficara de acuerdo a los cinco registros anteriores a la fecha actual.

- **Valores.-** Esta opción le permitirá al usuario visualizar a manera de consulta, la fecha de cada registro y podrá acceder a cada una de ellas para visualizar los precios por barril de petróleo de las distintas mezclas correspondientes a ese día.
- **Búsqueda general.-** Esta opción le permitirá al usuario visualizar a manera de consulta, registros mediante una búsqueda general. El usuario deberá ingresar un año de búsqueda y podrá visualizar los meses que comprende ese año. Posteriormente deberá elegir un mes para ese año y así visualizar los 30 registros para ese mes.
- **Búsqueda avanzada.-** Esta opción le permitirá al usuario visualizar a manera de consulta, un registro mediante una búsqueda avanzada. El usuario deberá ingresar el año el mes y el día para poder visualizar el registro.

II.3.3.2 Objetivo General del Sitio Web sugerido

Ser un medio de difusión que permita a los usuarios clarificar, definir, de forma más precisa, verificar situaciones de cambio o resultado, evaluar, estimar o demostrar progreso con respecto al precio por barril de petróleo a distintas mezclas establecidas.

III PROPUESTA DE DESARROLLO, ESTRUCTURA DE BASE DE DATOS Y ESTRUCTURA DEL SITIO.

Una vez que se ha realizado la planeación del sitio Web, se ha definido la organización lógica y se tiene idea de cuál será la apariencia del sitio, se procede a desarrollar los prototipos para que estos sean analizados y aprobados por las diferentes personas que tomarán decisiones respecto a si es correcto el manejo que se da de los indicadores económicos tomando en cuenta los factores anteriormente mencionados.

III.1 ESTRUCTURA DE ARCHIVOS Y ALMACENAJE

Un sitio Web es como una pequeña oficina, dependiendo de la organización que tenga será el nivel de eficacia y eficiencia que tenga al resolver un conflicto. Si somos ordenados sabremos donde guardar nuestros documentos y donde consultarlos cuando necesitemos de ellos. Lo mismo pasa con la programación. Un buen programador siempre debe considerar ordenar su código y debe ser minucioso al guardar y acomodar sus archivos. La estructura del sitio se verá reflejada en el cómo y en donde se guardaran los archivos.

Configurado nuestro servidor local (léase Anexo 1) ubicaremos el lugar donde guardaremos nuestros archivos. La ruta será `C:\apache2triad\htdocs` en donde crearemos una carpeta llamada Indicadores la cual contendrá toda la información para que nuestro sitio Web funcione.

Dentro de la carpeta indicadores que llamaremos Raíz, agregaremos subcarpetas o Subdirectorios que nos permitirán manejar y manipular de manera más rápida y cómoda nuestros archivos.

III.2 DISEÑO Y ESTRUCTURA DE LA BASE DE DATOS.

Uno de los puntos más importantes a considerar en el diseño de una base de datos, es la definición de tablas y las relaciones entre éstas. Al momento de diseñar es fundamental realizar un análisis previo del problema a resolver y posteriormente, realizar un correcto diseño inicial de la base de datos a utilizar.

Debe tenerse una visión general de las estructuras que almacenarán los datos que se encuentren en el sistema. Como último paso se construirán los mecanismos que manipularán estas estructuras, como lo son las consultas, los formularios y otros objetos que ayuden a manejar la base de datos. (DuBois, 2002)

Se debe hacer un diseño de cada tabla, determinar los campos que contienen y definir las relaciones entre éstas. Esto no quiere decir que una tabla no pueda modificarse en cualquier momento, pero esta acción puede conllevar a la destrucción de la información previamente almacenada.

Otro punto que debe tomarse en cuenta antes de modificar una tabla, es que la mayoría de los elementos de la base de datos se basan en ellas, por ejemplo, las consultas, los formularios, los informes, etc. Por lo tanto las alteraciones en las tablas tienen una repercusión directa sobre los elementos que la manipulan.

III.2.1 Determinar la Finalidad

El primer punto que fue considerado para la realización de la Base de Datos es que esta mezclara el diseño Estático y Dinámico. Por un lado a nivel Usuario esta base de datos será de sólo lectura, utilizadas primordialmente para consultar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

Por otro lado a nivel Administrador esta Base de Datos se desarrollará bajo un carácter dinámico, ya que el Administrador tendrá permisos para modificar la información almacenada, realizar operaciones como actualización y consulta de datos.

III.2.2 Determinar las tablas que se necesitan para la Base de Datos

Determinar las tablas que se requieren, es uno de los puntos más complicado del proceso de diseño de la base de datos.

Esto se debe a que los resultados que se desean obtener de la base de datos (los informes que se desean imprimir, los formularios que desean utilizar, las preguntas para las que se desean respuestas) no proporcionan pistas necesariamente acerca de la estructura de las tablas que los producen.

Al diseñar las tablas, se recomienda dividir la información teniendo en cuenta los siguientes principios fundamentales:

- Una tabla no debe contener información duplicada (redundante) y la información no debe duplicarse entre las tablas. Cuando cada elemento de información está almacenado en una tabla, se actualiza en un solo lugar. Esto resulta más eficiente y elimina la posibilidad de que existan entradas duplicadas que contengan información diferente. Por ejemplo, probablemente se desee almacenar de una lista de clientes, la dirección y el número de teléfono de cada uno de ellos, una sola vez, en una sola tabla.

- Cada tabla debe contener información sobre un asunto. Cuando cada tabla contiene hechos sólo sobre un asunto, se puede mantener la información acerca de cada asunto, independientemente de otros asuntos. Por ejemplo, se pueden almacenar las direcciones de los clientes en una tabla diferente de los pedidos de los clientes, de modo que se pueda eliminar un pedido y conservar a la vez la información sobre el cliente correspondiente.

III.2.3 Determinar los campos que se necesitan para las Tablas

Cada tabla contiene información acerca del mismo asunto, y cada campo de una tabla contiene hechos individuales sobre el asunto de la tabla. Por ejemplo, una tabla de clientes puede incluir los campos del nombre de la empresa, la dirección, la ciudad, el país y el número de teléfono.

Al determinar los campos para cada tabla, se deben tener en cuenta las siguientes sugerencias:

- Relacionar cada campo directamente con el asunto de la tabla
- No incluir datos derivados ni calculados (datos que son el resultado de una expresión).
- Incluir toda la información que se necesite.
- Almacenar información en sus partes lógicas más pequeñas (por ejemplo, Nombre y Apellidos, en lugar del Nombre completo).
- Seleccionar el tipo de datos o el tamaño de campo más pequeño que sea apropiado para los datos del campo

III.2.4 Identificar las campos con valores exclusivos

Para poder conectar información almacenada en tablas independientes (por ejemplo, para conectar a un cliente con todos los pedidos del cliente), cada tabla de la base de datos debe incluir un campo o un conjunto de campos que identifiquen de forma exclusiva cada registro individual de la tabla. Este campo o conjunto de campos lo llamaremos Llave primaria.

La potencia de un sistema de base de datos reside en su capacidad para encontrar y reunir rápidamente información almacenada en tablas independientes mediante consultas, formularios e informes. Para ello, cada tabla debe incluir un campo o un conjunto de campos que identifique exclusivamente cada registro almacenado en la tabla. Esta información se denomina clave principal de la tabla. Una vez designada una clave principal de una tabla, con el fin de garantizar su exclusividad. (Pougue & O'Really, 2007)

III.2.5 Determinar las Relaciones entre las Tablas

Una vez creadas las tablas diferentes para cada tema de la base de datos y que se han identificado los campos de clave principal, se necesita de una forma de indicarle a Access cómo debe volver a combinar esa información relacionada de un modo significativo.

El primer paso de este proceso es definir relaciones entre las tablas. Una vez realizada esta operación, se pueden crear consultas, formularios e informes para mostrar información de varias tablas a la vez. Por ejemplo, los campos de varias tablas deben coordinarse de modo que muestren información acerca de un mismo tema. Esta coordinación se lleva a cabo mediante las relaciones entre las tablas.

Una relación hace coincidir los datos de los campos clave (normalmente un campo con el mismo nombre en ambas tablas). En la mayoría de los casos, estos campos coincidentes son la clave principal de una tabla, que proporciona un identificador único para cada registro, y una clave externa de la otra tabla.

III.2.6 Precisar el Diseño

Una vez diseñadas las tablas, los campos y las relaciones que se necesitan, es el momento de estudiar el diseño y detectar los posibles fallos que puedan surgir. Es más sencillo cambiar el diseño de la base de datos en su propia definición, que una vez que se hayan rellenado las tablas con datos.

Luego de todo esto se podrá utilizar *MYSQL* para crear las tablas, especificar las relaciones entre las tablas e introducir algunos registros de datos en cada tabla. También es buen momento para buscar duplicaciones de datos innecesarias y eliminarlos.

III.2.7 Introducir datos y crear otros objetos de la Base de Datos

Cuando se considere que la estructura de las tablas cumple los objetivos de diseño descritos anteriormente, es el momento de comenzar a diseñar el sistema que podrá ingresar, modificar, consultar datos, formularios y módulos que desee.

III.2.8 Estructura final de la Base de Datos

Mediante un sistema de gestión de base de datos relacional, *MYSQL*, implementaremos la Base de Datos para nuestro sistema.



The screenshot shows the MySQL interface for a database named 'acervo' on localhost. It displays the structure of two tables: 'i_petroleo' and 'user'.

Campo	Tipo	Nulo	Predeterminado
<u>id_i_petroleo</u>	int(10)	Sí	NULL
west_texas	decimal(5,2)	Sí	NULL
mezcla_mexicana	decimal(5,2)	Sí	NULL
olmeca	decimal(5,2)	Sí	NULL
itsmo	decimal(5,2)	Sí	NULL
maya	decimal(5,2)	Sí	NULL
fecha	varchar(20)	Sí	NULL

Campo	Tipo	Nulo	Predeterminado
username	varchar(15)	Sí	
password	varchar(15)	Sí	

El nombre de la Base de Datos será *Acervo* y contendrá dos tablas. *i_petroleo* y *user*.

La tabla *i_petroleo* contendrá 7 campos encargados de almacenar los datos históricos del valor del petróleo por día. La llave principal que nos permitirá manipular y realizar operaciones será el campo *id_i_petroleo*; la estructura de dicho campo será de tipo entero de 10, solo permitirá números enteros positivos [*UNSIGNED*], y se completará el con ceros a la izquierda [*ZEROFILL*], no podrá admitir valores nulos (*NOT NULL*) y se auto incrementara automáticamente; esta última opción nos asegurara que ningún registro se guardara con el mismo identificador o *ID* y cada uno será único.

West_texas, *mezcla_mexicana*, *olmeca*, *istmo* y *maya* serán los campos encargados de almacenar el precio o valor de barril de petróleo por día. La estructura de estos campos será de tipo decimal (5,2) y el campo fecha que almacenara el día de creación de registro y permitirá identificar por día el precio o valor del barril de petróleo.

La tabla *user* contendrá 2 campos que permitirán al sistema identificar al usuario administrador mediante los campos *username* y *password*. La estructura de estos campos será de tipo *varchar* de 15.

III.3 DISEÑO DE LA ESTRUCTURA DEL SITIO WEB.

III.3.1 Elegir Herramientas de Trabajo

La primera acción a seguir es definir la estructura *HTML* del sitio Web, es decir los marcos que estructuran la pantalla. La finalidad del uso de un marco o frame es dividir la ventana del explorador en múltiples regiones o zonas independientes que permitan al usuario una fácil navegación.

Para la elaboración de los documentos *CSS*, *HTML* y *PHP*, se utilizara como herramienta de trabajo cualquier editor que permita su elaboración y modificación, como por ejemplo *EditPlus* o *Dreamweaver*. En este caso desarrollaremos con *Dreamweaver*.

Dentro de las ventajas que ofrece *Dreamweaver* es que permite ver en modo grafico la forma en que se va estructurando nuestro documento y también es posible modificar manualmente el código.

Para poder hacer pruebas y desarrollar nuestro sistema debemos instalar previamente *Apache2triad* que configurará lo necesario para crear nuestro sistema (léase Anexo 1). Después de ser instalado utilizaremos *dreamweaver* y lo primero que haremos será configurar un servidor local de pruebas. (Léase Anexo 2)

Para programar una página o un sistema debemos siempre tener en cuenta la simplificación y la legibilidad del código, esto para hacer más fácil la modificación de los documentos y hacerlo más legible para cualquier persona que necesite y tenga los permisos adecuados para ver, modificar o agregar código.

III.3.2 Estructura de Encabezado y Hoja de estilos

Manteniendo siempre la ideología de simplificar código, crearemos un encabezado o *header* que será agregado en cada documento *PHP* que trabajemos o utilicemos.

Comenzaremos con crear un archivo *PHP* llamado `header.php` el cual almacenaremos en la carpeta `GENERALES` en la siguiente ruta:
`C:\apache2triad\htdocs\indicadores\`



El *Header* le permitirá visualizar al usuario el nombre del Proyecto, del Instituto y de la UNAM, contendrá una función *PHP* que permitirá visualizar la fecha actual y un botón que permita al usuario iniciar sesión. El inicio de sesión será exclusivo para administradores del sitio Web.

La estructura que se utiliza para el *header* es un margen de 0 pixeles (*px*), esto quiere decir que no existirán espacios entre el monitor y nuestra página de Internet.

Esta característica la podemos delimitar en nuestra hoja de estilos, para lo cual necesitaremos crearla y así poder adjuntarla en nuestro *header*. En este caso será conveniente adjuntar nuestra hoja de estilos en el *header*, ya que el *header* será incluido en todas nuestros documentos de *PHP* y así mismo se incluirán todos los estilos que se generen en el desarrollo del sistema.

Para esto crearemos un archivo .CSS el cual será llamado style_iiec en la carpeta style_iiec contenida en la siguiente ruta C:\apache2triad\htdocs\indicadores\generales.

El código que permitirá adjuntar la hoja de estilos al *Header* se compondrá por la ubicación o ruta del enlace hacia este, el código quedará de la siguiente manera.
<link href="style_iiec/style_iiec.css" rel="stylesheet" type="text/css" />

III.3.3 Estructura de la programación del Sitio

III.3.3.1 Módulos de Conexión

III.3.3.1.1 Conexiones a la Base de Datos

Primero es recomendable separar los datos de conexión con las funciones de conexión en sí, de modo que si estos cambian sea más fácil y rápido cambiarlos en todos los *PHP* en los que haga falta.

Por este motivo se definirá un archivo que contenga los datos de conexión y al igual que nuestro *HEADER*, este será incluido en todas los documentos *PHP* que se requiera.

El archivo se llamará *ACERVO.PHP*, ubicado en el subdirectorio *GENERALES* dentro de la carpeta *CONNECT*, el cual contendrá los siguientes datos:

- Nombre del *Host* a la cual nos deseamos conectar.
- Nombre de la Base de Datos a la cual nos deseamos conectar.
- Nombre del Usuario con el cual nos deseamos conectar.
- Contraseña con la cual nos podremos conectar.

El *HOST* en nuestro caso será *LOCALHOST*, debido a que las pruebas y la configuración realizada con *APACHE2TRIAD* hará que nuestra computadora sea un servidor local de pruebas.

La Base de Datos que se definió en el capítulo anterior es ACERVO, que será por medio de la cual podemos realizar nuestra conexión.

El nombre de Usuario dado por default al configurar e instalar *Apache2triad* es el usuario *ROOT*.

La contraseña por la cual nos podremos conectar será la que nosotros hayamos establecido en la configuración de *Apache2triad*. Debemos recordar que esta contraseña es global y será ocupada por todos los componentes de instalación de *Apache2triad*.

El código de conexión para *ACERVO.PHP* es el siguiente:

```
<?php
    $hostname_acervo = "localhost";
    $database_acervo = "acervo";
    $username_acervo = "root";
    $password_acervo = "*****";
    $acervo = mysql_pconnect($hostname_acervo, $username_acervo,
    $password_acervo) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

Este archivo se incluirá a cada uno de los archivos *PHP* en donde se requiera la conexión o manejo de Base de Datos ACERVO mediante el siguiente código:

```
<?php include("../generales/Connections/acervo.php"); ?>
```

“../” será incluido cuando se desee indicar la salida o retroceso de un subdirectorio.

III.3.3.2 Modulo de Inserción

III.3.3.2.1 Estructura de Formularios

Una vez establecida la base de datos y determinados los parámetros de conexión al servidor, es momento de hacer pruebas e ingresar registros a nuestra base de datos mediante formularios. Para realizar la inserción a una base de datos dentro de nuestro sistema es necesario diseñar un formulario y dentro del mismo documento *PHP* se debe hacer uso de la instrucción *INSERT* para almacenar nuestros datos. La estructura del formulario para nuestro sistema se integrará siempre de la siguiente manera:

```
<form action="<?php echo $editFormAction; ?>" method="post"
name="i_petroleo" id="i_petroleo" />

<input name="west_texas" type="text" id="west_texas"/>

<input name="mezcla_mexicana" type="text" id="mezcla_mexicana"/>

<input name="olmeca" type="text" id="olmeca"/>

<input name="istmo" type="text" id="istmo"/>

<input name="maya" type="text" id="maya"/>

<input type='hidden' name="fecha" id="fecha" value="<?
echo(date("Y/m/d")); ?>" />

<input type="submit" name="Submit" value="Enviar" />

<input type="reset" name="reset" id="reset" value="Limpiar" />
```

En este código podemos observar lo siguiente:

form action El atributo “*action*” indica el tipo de acción que va a realizar el formulario. Si lo que queremos es que la información sea enviada a un programa que la gestione o a otro documento *PHP*, debemos indicarle en la acción, la *url* del archivo donde se encuentra ubicado el programa o documento *PHP* que la

gestionará. Siendo así se escribirá de la siguiente manera: `<form action="dirección completa del archivo que la gestionará" ...> </form>`.

`"<?php echo $editFormAction; ?>"` Indicará a la aplicación que los resultados ingresados por el usuario serán enviados, recibidos y guardados por el mismo documento *PHP*.

method Mediante este atributo le indicamos al formulario la forma en la que el formulario será enviado. Existen dos valores posibles: *GET* y *POST*. El método indicado será *POST*.

name y *id* serán atributos que permitirán identificar y nombrar los campos que se envíen y mediante los mismos identificaremos la información que se enviará.

type="text" Se utiliza para la entrada de cadenas de texto corto, como por ejemplo nombre de personas, números, fechas o diversos datos que se puedan expresar en una línea de texto. En este caso será utilizado para introducir el precio por barril de petróleo al día.

input type='hidden' En este caso no se muestra ningún campo para la entrada de datos al usuario, pero un valor especificado es enviado junto con el formulario. Se suele usar para transmitir información de estado ó control ó para enviar algún tipo de información que no debe ser variada en el formulario, pero sí debe ser enviada junto a este.

date("Y/m/d") se usa para expresar fechas y horas en formato estándar *GMT*, y entre los paréntesis podemos agregar distintos comandos para que nos muestre un determinado formato y orden. El formato de fecha que enviamos será AÑO/MES/DIA en que se envía formulario.

type="submit" El atributo *VALUE* especifica una etiqueta no editable que se mostrará en el botón de envío. Lo normal es que este botón no envíe datos, pero si se indica el atributo *NAME* con un nombre de variable será enviada la variable

con el valor de *VALUE*. Esto puede ser útil si se incluyen distintos botones de *SUBMIT* para distinguir cual fue pulsado.

input type="reset" Este botón se usa para volver a los valores por defecto todos los elementos del formulario, borrando todos los datos introducidos por el usuario.

III.3.3.2 Guardar valores enviados en la Base de Datos

Cuando queremos procesar el envío y almacenamiento de datos en una misma hoja deberemos utilizar el atributo *action="<?php echo \$editFormAction; ?>"* el cual será recibido en la misma hoja.

Para que esto sea posible debemos implementar el siguiente código:

```
$editFormAction = $_SERVER['PHP_SELF'];  
  
if (isset($_SERVER['QUERY_STRING'])) {  
  
    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);  
  
}
```

Donde la primer línea de código guarda el nombre del archivo que se está ejecutando en *\$editFormAction*.

La segunda línea de código pregunta si existen variables en la *url* Ej: *index.php?nombre=YairMunive*. Donde *QUERY_STRING* tiene el valor de "nombre=YairMunive".

Y finalmente si la condición es cierta agrega las variables de *URL* a la variable *\$editFormAction*, reemplazando los caracteres por sus correspondientes en *html*.

De esta manera ya es posible diseñar una aplicación que ejecute un *Query* de inserción a la base de Datos y estos a su vez sean almacenados.

La estructura diseñada para este sistema se hará de la siguiente manera:

```
$insertSQL = sprintf("INSERT INTO i_petroleo (west_texas,  
mezcla_mexicana, olmeca, istmo, maya, fecha)  
  
VALUES ("$_POST['west_texas'].'", "$_POST['mezcla_mexicana'].'",  
"$_POST['olmeca'].'", "$_POST['istmo'].'", "$_POST['maya'].'");  
  
mysql_select_db($database_acervo, $acervo);  
  
$Result1 = mysql_query($insertSQL, $acervo) or die(mysql_error());
```

En este código podemos observar lo siguiente:

Sprintf es una función que da formato al *Query* y permite almacenarlo en la variable *\$insertSQL*.

\$_POST permitira recibir los datos del formulario. Para recibir estos valores solo necesitaremos usar la variable *\$_POST* definida en el método de envío de formulario *METHOD* y agregar entre corchetes el campo que deseamos imprimir. Ejemplo ['*mezcla_mexicana*'].

mysql_select_db establece la base activa que estará asociada con el identificador de enlace especificado. Si no se especifica un identificador de enlace, se asume el último enlace abierto. Si no hay ningún enlace abierto, la función intentará establecer un enlace como si se llamara a *mysql_connect*()).

mysql_query envía una consulta a la base de datos *\$acervo* activa en el servidor asociado con el identificador_de_enlace *\$query_Recordset1* que hace referencia a una variable que contiene el formato del *Query* de búsqueda.

III.3.3.3 Módulos de Consulta

III.3.3.3.1 Consultar información a la Base de Datos

Para realizar las consultas a una base de datos dentro de nuestro sistema solo es necesario realizar el *Query* de búsqueda dentro de nuestro código y ejecutar la búsqueda. La estructura de búsqueda para nuestro sistema se integrará siempre de la siguiente manera:

```
$query_Recordset1 = sprintf("SELECT * FROM i_petroleo WHERE fecha = '$date'");  
  
$Recordset1 = mysql_query($query_Recordset1, $acervo) or die(mysql_error());  
  
$row_Recordset1 = mysql_fetch_assoc($Recordset1);  
  
$totalRows_Recordset1 = mysql_num_rows($Recordset1);
```

En este código podemos observar lo siguiente:

Sprintf es una función que da formato al Query y permite almacenarlo en la variable *\$query_Recordset1*.

mysql_query envía una consulta a la base de datos *\$acervo* activa en el servidor asociado con el identificador_de_enlace *\$query_Recordset1* que hace referencia a una variable que contiene el formato del Query de búsqueda.

mysql_fetch_assoc devuelve una matriz asociativa que corresponde a la fila recuperada, o *FALSE* si no hay más filas disponibles de la variable de búsqueda *\$Recordset1* y a su vez es almacenado en la variable *\$row_Recordset1*.

mysql_num_rows regresa el número de filas obtenido de la variable de búsqueda *\$Recordset1*. Este comando es válido solo para las sentencias *SELECT*.

Cabe señalar que la variable *\$date* contiene el año mes y día de la función *DATE*;

III.3.3.3.2 Imprimir los valores consultados a la Base de Datos

Una vez hecha nuestra consulta en la base de datos, podremos imprimir los resultados obtenidos. Para imprimir estos valores solo necesitaremos usar la variable `$row_Recordset1` definida en nuestra estructura de búsqueda y agregar entre corchetes el campo que deseamos imprimir. Ejemplo ['mezcla_mexicana'].

III.3.3.4 Módulos de Graficación

III.3.3.4.1 JpGraph

JpGraph es una librería de clases orientadas a objetos para *PHP* (recomendada para *PHP* 4.3.0), para la creación dinámica de imágenes. Con *JpGraph* se pueden crear gráficas complejas con un mínimo de código y con un control muy detallado. Con el único requerimiento de que se tenga instalado el soporte para la librería *GD Graphics library* (<http://www.boutell.com/gd/>).

Características:

- Se pueden crear imágenes (gráficas) que son amigables para la web, pues en promedio una imagen de un tamaño de 300 * 200 ocupa alrededor de 2K y en general las imágenes rara vez ocuparán más de 4-5K.
- *JpGraph* cuenta con soporte para *GD1* y *GD2*, la librería auto detecta que versión del *GD* se tiene instalada.
- Generación automática de imágenes del lado del cliente con la posibilidad de modificar las graficas de forma fácil.
- Soporte para adjuntar texto a las imágenes con párrafos multilínea y soporte para fuentes *TTF* (tipos de letra).
- Soporte para *alpha-blending* (grado de traslucidez).
- Contiene más de 200 banderas de países.
- Soporte para complejas gráficas de Gantt.
- Escalas flexibles.

- Soporta formatos gráficos de tipo: PNG, GIF y JPG y puedes seleccionar el mejor formato para una gráfica. Aunque esto depende de la instalación que se tenga de *PHP* y de la librería que se esté utilizando.
- Soporta gráficas con imágenes de fondo (*background image*).
- Soporte para graficas de barras horizontales.
- Soporte para gráficas de tipo científico.
- Se pueden ocultar (por tiempo) las gráficas para no saturar el servidor de *HTTP*.
- Cuenta con auto escala para dar un mejor valor estético a las imágenes generadas.
- Total soporte para dar una escala manual a las gráficas con un gran control de posición.
- Diferentes estilos de relleno para las graficas.
- Soporte para generar las graficas al vuelo o como un proceso batch en el servidor.
- Amplía documentación con 150 páginas de tutorial y una extensa referencia de clase.

JpGraph cuenta con licencia *QPL 1.0 (Qt Free Licensee)* Para usos no comerciales, software libre (*Open source*) y usos educacionales, y una licencia *Professional (JpGraph Professional)* para usos comerciales. Básicamente esto significa que si quien lo ocupa no desarrolla software libre y tiene ganancias financieras con ello, para *JpGraph*, esto es considerado uso comercial. Cabe mencionar que la versión profesional incluye funcionalidades adicionales. (Asial)

III.3.3.4.2 Generar Gráficos mediante JpGraph

Luego de haber descargado los archivos fuente de esta librería en *JpGraph Download* creamos nuestro archivo *PHP* en cual incluimos primero el archivo *jpgraph.php* el cual contiene las funciones básicas y luego el archivo *jpgraph_line.php* que contiene las funciones específicas para la creación de gráficos de línea.

```
<?php

// Incluimos los archivos necesarios

include("inc/jpgraph.php");

include("inc/jpgraph_line.php");

// Creamos el array de datos

$ydata = array(11, 3, 8, 12, 5, 1, 9, 13, 5, 7);

// Creamos un nuevo grafico de 350x250

$graph = new Graph(350, 250, "auto");

$graph->SetScale( "textlin");

// Creamos el grafico basado en el array

$lineplot = new LinePlot($ydata);

$lineplot->SetColor("blue");

// Agregamos el grafico a la imagen

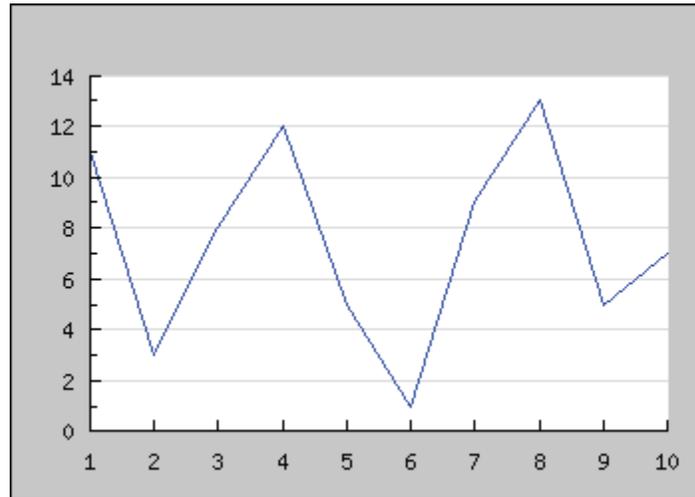
$graph->Add( $lineplot);

// Mostramos la imagen

$graph->Stroke();

?>
```

Podemos notar que hemos creado primero la imagen de 350x250 y con los ejes auto escalados, luego creamos un objeto que contiene el gráfico de líneas, posteriormente agregamos este gráfico a la imagen principal y la enviamos al cliente mediante *Stroke*.



Si deseamos personalizar los gráficos podemos agregar título al gráfico, nombre los ejes X y Y, así mismo podemos agregar más arreglos para comparaciones de gráficos. La siguiente aplicación ejemplificara lo anterior:

```
include("inc/jpgraph.php");  
include("inc/jpgraph_line.php");  
  
$ydata1 = array(11, 3, 8, 12, 5, 1, 9, 13, 5, 7);  
$ydata2 = array(1, 19, 15, 7, 22, 14, 5, 9, 21, 13);  
  
$graph = new Graph(350, 250, "auto");  
$graph->SetScale( "textlin");  
$graph->img->SetMargin(40, 20, 20, 40);  
$graph->title->Set("JpGraph Demo");
```

```
$graph->xaxis->title->Set("Altura");
```

```
$graph->yaxis->title->Set("Total");
```

```
$lineplot1 = new LinePlot($ydata1);
```

```
$lineplot1->SetColor("blue");
```

```
$lineplot2 = new LinePlot($ydata2);
```

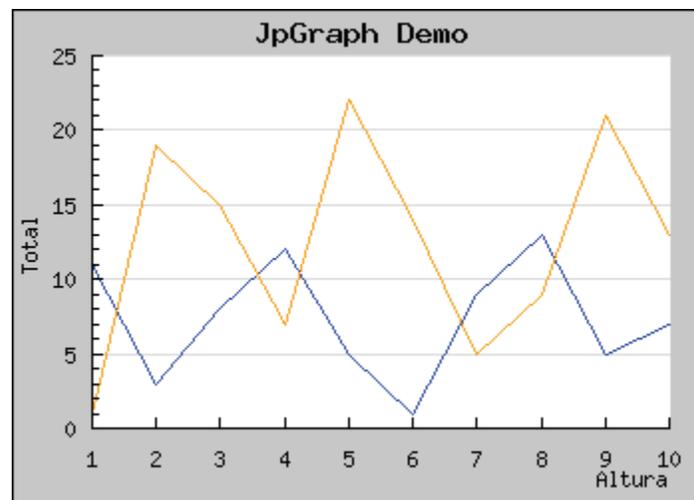
```
$lineplot2->SetColor("orange");
```

```
$graph->Add($lineplot1);
```

```
$graph->Add($lineplot2);
```

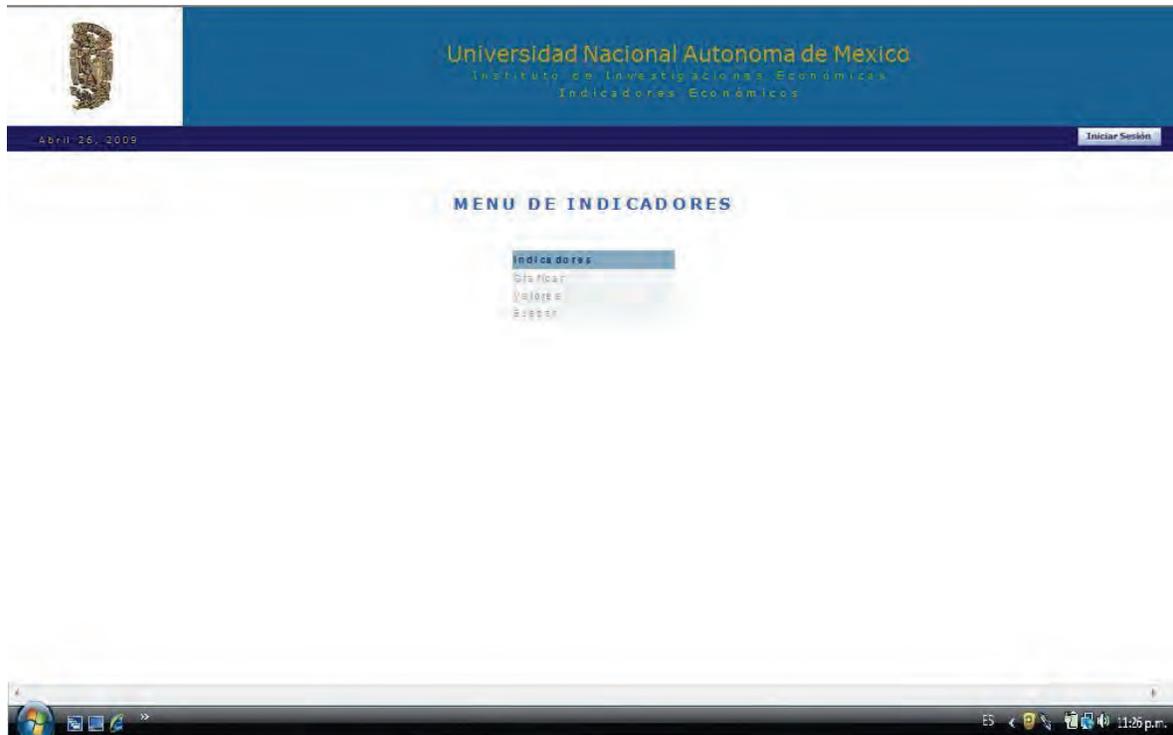
```
$graph->Stroke();
```

Ejecutado este ejemplo, mostrará lo siguiente:

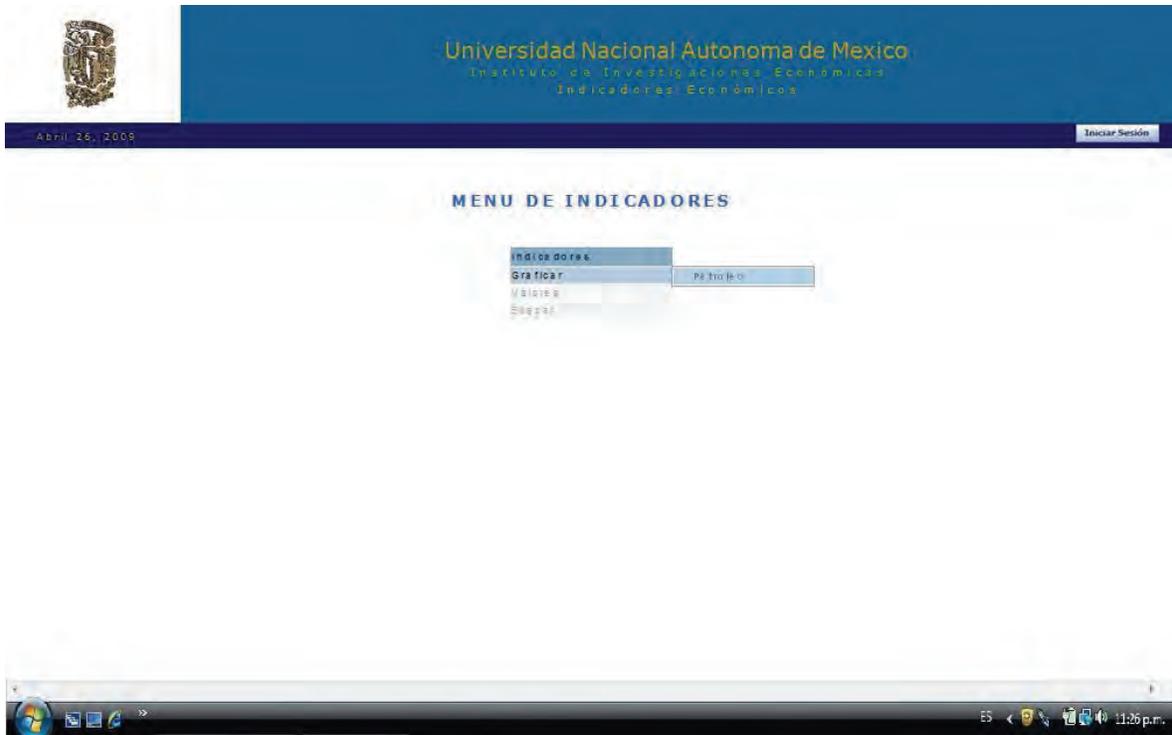


IV DESARROLLO E IMPLEMENTACION DEL SITIO WEB PROPUESTO PARA LA GRAFICACION DE INDICADORES ECONOMICOS.

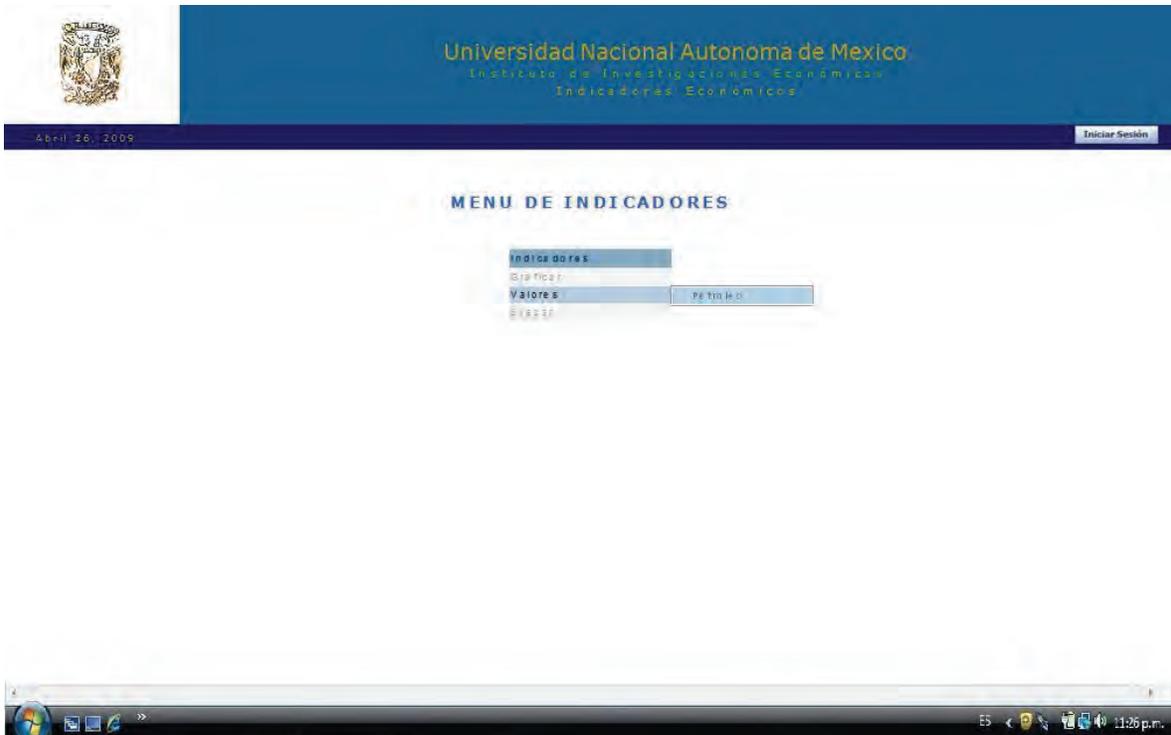
IV.1 Modulo 1 - Menú de Usuario



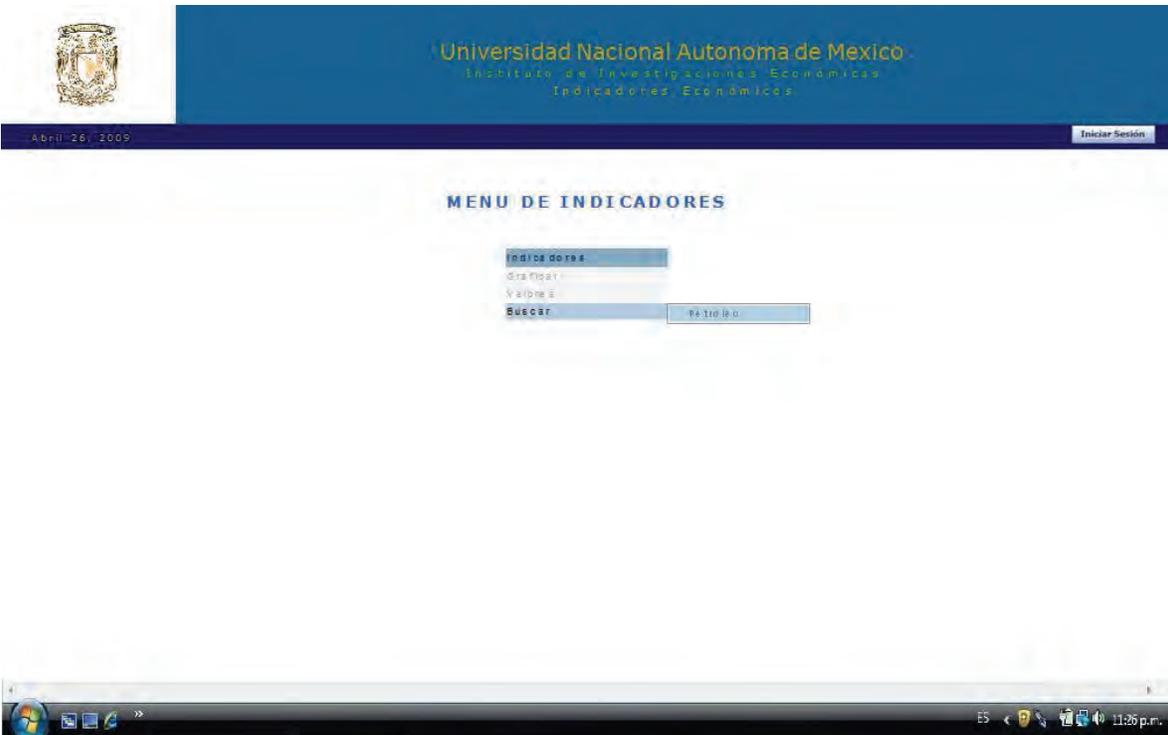
```
//=====
// Archivo: INDEX.PHP
// Id: PHP Modulo Usuario, Menú de Usuario.
// Modulo1. Interfaz de Menú de Usuario
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción:
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```



```
//=====
// Archivo: INDEX.PHP
// Id: PHP Modulo Usuario, Menú de Usuario.
// Modulo1. Interfaz de Menú de Usuario
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción GRAFICAR - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```



```
//=====
// Archivo: INDEX.PHP
// Id: PHP Modulo Usuario, Menú de Usuario.
// Modulo1. Interfaz de Menú de Usuario
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción VALORES - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```



```
//=====
// Archivo: INDEX.PHP
// Id: PHP Modulo Usuario, Menú de Usuario.
// Modulo1. Interfaz de Menú de Usuario
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción BUSCAR - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```

IV.2 Modulo 2 - Consulta y Graficación del valor de distintas mezclas del valor de petróleo al día.

Universidad Nacional Autónoma de México
Instituto de Investigaciones Económicas
Indicadores Económicos

Abril 26, 2009 Iniciar Sesión

Valor de Petróleo al día

Coincidencias Encontradas: 1

Lista de precios al día: 2009/04/25

- West Texas Int: 46.00
- Mezcla Mexicana: 56.00
- Olimeca: 76.00
- Istmo: 26.00
- Maya: 36.00

Valor de Petróleo al Día 2009/04/26

Petróleo/Día	2009/04/21	2009/04/22	2009/04/23	2009/04/24	2009/04/25
West Texas Int	64.00	54.00	74.00	44.00	46.00
Mezcla Mexicana	65.00	55.00	75.00	45.00	56.00
Olimeca	67.00	57.00	77.00	47.00	76.00
Istmo	62.00	52.00	72.00	42.00	26.00
Maya	63.00	53.00	73.00	43.00	36.00

Regresar

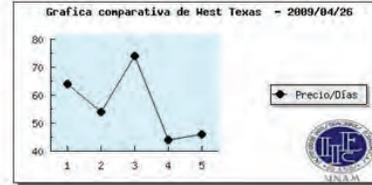
```
//=====
// Archivo: SHOWPETROLEOS_GRAFICA.PHP
// Id: PHP Modulo Usuario, Visualiza y Compara Mezclas de Petróleo.
// Modulo2. Interfaz Grafica para la consulta y graficación de valores.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivos adjuntos: GRAFICA.PHP. / Id: Imprime Grafica de Valores
// Ubicación de archivo adjunto: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Muestra el valor de las 5 mezclas de petróleo al día, compara el
// valor de cada mezcla y grafica los valores mediante una fecha de referencia.
// Fecha de Referencia: Fecha del día - 5
//=====
```



Valor de Petroleo al día

Coincidencias Encontradas: 1

Lista de precios al día	2009/04/25
West Texas Int:	46.00
Mezcla Mexicana:	56.00
Olmecca:	76.00
Istmo:	26.00
Naya:	36.00



Petroleo/Dias	2009/04/21	2009/04/22	2009/04/23	2009/04/24	2009/04/25
West Texas Int:	64.00	54.00	74.00	44.00	46.00
Mezcla Mexicana:	65.00	55.00	75.00	45.00	56.00
Olmecca:	67.00	57.00	77.00	47.00	76.00
Istmo:	62.00	52.00	72.00	42.00	26.00
Naya:	63.00	53.00	73.00	43.00	36.00

Regresar

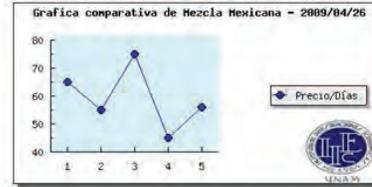
```
//=====
// Archivo: SHOWPETROLEOS_GRAFICA_WEST.PHP
// Id: PHP Modulo Usuario, Visualiza y compara la mezcla WEST_TEXAS.
// Modulo2. Interfaz Grafica para la consulta y graficación de valores.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivos adjuntos: G_WEST.PHP.
// Id: Imprime Grafica de Valores de la mezcla WEST_TEXAS
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra el valor de la mezcla WEST_TEXAS y grafica los valores
// mediante una fecha de referencia. G_WEST.PHP es incluido como imagen.
// Fecha de Referencia: Fecha del día - 5
//=====
```



Valor de Petroleo al día

Coincidencias Encontradas: 1

Lista de precios el día	2009/04/25
West Texas Int:	46.00
Mezcla Mexicana:	56.00
Olimex:	76.00
Istmo:	26.00
Maya:	36.00



Petroleo/Días	2009/04/21	2009/04/22	2009/04/23	2009/04/24	2009/04/25
West Texas Int	64.00	54.00	74.00	44.00	46.00
Mezcla Mexicana	65.00	55.00	75.00	45.00	56.00
Olimex	67.00	57.00	77.00	47.00	76.00
Istmo	62.00	52.00	72.00	42.00	26.00
Maya	63.00	53.00	73.00	43.00	36.00

Regresar

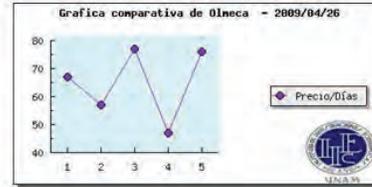
```
//=====
// Archivo: SHOWPETROLEOS_GRAFICA_MEXICANA.PHP
// Id: PHP Modulo Usuario, Visualiza y compara la mezcla MEXICANA.
// Modulo2. Interfaz Grafica para la consulta y graficación de valores.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivos adjuntos: G_MEXICANA.PHP.
// Id: Imprime Grafica de Valores de la mezcla MEXICANA
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra el valor de la mezcla MEXICANA y grafica los valores
// mediante una fecha de referencia. G_MEXICANA.PHP es incluido como imagen.
// Fecha de Referencia: Fecha del día - 5
//=====
```



Valor de Petroleo al día

Coincidencias Encontradas: 1

Lista de precios al día	2009/04/25
West Texas Int:	46.00
Mezcla Mexicana:	56.00
Olmecca:	76.00
Istmo:	26.00
Naya:	36.00



Petroleo/Días	2009/04/21	2009/04/22	2009/04/23	2009/04/24	2009/04/25
West Texas Int:	64.00	54.00	74.00	44.00	46.00
Mezcla Mexicana:	65.00	55.00	75.00	45.00	56.00
Olmecca:	67.00	57.00	77.00	47.00	76.00
Istmo:	62.00	52.00	72.00	42.00	26.00
Naya:	63.00	53.00	73.00	43.00	36.00

Regresar

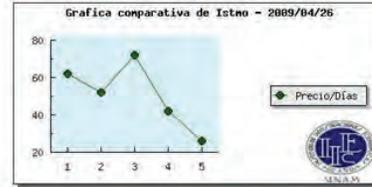
```
//=====
// Archivo: SHOWPETROLEOS_GRAFICA_OLMECA.PHP
// Id: PHP Modulo Usuario, Visualiza y compara la mezcla OLMECA.
// Modulo2. Interfaz Grafica para la consulta y graficación de valores.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivos adjuntos: G_OLMECA.PHP.
// Id: Imprime Grafica de Valores de la mezcla OLMECA
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra el valor de la mezcla OLMECA y grafica los valores
// mediante una fecha de referencia. G_OLMECA.PHP es incluido como imagen.
// Fecha de Referencia: Fecha del día – 5
//=====
```



Valor de Petroleo al día

Coincidencias Encuentros: 1

Lista de precios al día	2009/04/25
West Texas Int:	46.00
Mezcla Mexicana:	56.00
Olimeca:	76.00
Istmo:	26.00
Maya:	36.00



Petroleo/Días	2009/04/21	2009/04/22	2009/04/23	2009/04/24	2009/04/25
West Texas Int:	64.00	54.00	74.00	44.00	46.00
Mezcla Mexicana:	65.00	55.00	75.00	45.00	56.00
Olimeca:	67.00	57.00	77.00	47.00	76.00
Istmo:	62.00	52.00	72.00	42.00	26.00
Maya:	63.00	53.00	73.00	43.00	36.00

Regresar

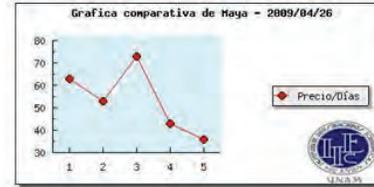
```
//=====
// Archivo: SHOWPETROLEOS_GRAFICA_ISTMO.PHP
// Id: PHP Modulo Usuario, Visualiza y compara la mezcla ITSMO.
// Modulo2. Interfaz Grafica para la consulta y graficación de valores.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivos adjuntos: G_ISTMO.PHP.
// Id: Imprime Grafica de Valores de la mezcla ITSMO
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra el valor de la mezcla ITSMO y grafica los valores
// mediante una fecha de referencia. G_ITSMO.PHP es incluido como imagen.
// Fecha de Referencia: Fecha del día – 5
//=====
```



Valor de Petroleo al día

Coincidencias Encontradas: 1

Lista de precios al día	2009/04/25
West Texas Int:	46.00
Mezcla Mexicana:	56.00
Dimeca:	76.00
Isimo:	26.00
Maya:	36.00



Petroleo/Dias	2009/04/21	2009/04/22	2009/04/23	2009/04/24	2009/04/25
West Texas Int:	64.00	54.00	74.00	44.00	46.00
Mezcla Mexicana:	65.00	55.00	75.00	45.00	56.00
Dimeca:	67.00	57.00	77.00	47.00	76.00
Isimo:	62.00	52.00	72.00	42.00	26.00
Maya:	63.00	53.00	73.00	43.00	36.00

Regresar

```
//=====
// Archivo: SHOWPETROLEOS_GRAFICA_MAYA.PHP
// Id: PHP Modulo Usuario, Visualiza y compara la mezcla MAYA.
// Modulo2. Interfaz Grafica para la consulta y graficación de valores.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivos adjuntos: G_MAYA.PHP.
// Id: Imprime Grafica de Valores de la mezcla MAYA
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra el valor de la mezcla MAYA y grafica los valores
// mediante una fecha de referencia. G_MAYA.PHP es incluido como imagen.
// Fecha de Referencia: Fecha del día – 5
//=====
```

IV.3 Modulo 3 - Consulta e Imprime el valor de distintas mezclas del valor de petróleo por día.

```
//=====
// Archivo: SHOWPETROLEOS.PHP
// Id: PHP Modulo Usuario, Visualiza el valor de las mezclas de petróleo por día.
// Modulo3. Interfaz Grafica para la consulta de todos los registros insertados.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra un contador del total de registros insertados y el valor o
// precio almacenado para cada una de las mezclas por fecha de inserción,
// ordenados de manera descendente.
//
// El Número total de registros será mostrado por grupos de 10 en 10 y se incluirán
// iconos para adelantar y atrasar las páginas mostradas, así como ir a la primera
// y última página
//=====
```



Valor de Petroleo al día

Coincidencias Encontradas: 125

Lista de precios al día	2009/04/30
West Texas Int:	74.00
Nezcla Mexicana:	75.00
Olimexa:	77.00
Istmo:	72.00
Maya:	73.00

Lista de precios al día	2009/04/29
West Texas Int:	74.00
Nezcla Mexicana:	75.00
Olimexa:	77.00
Istmo:	72.00
Maya:	73.00

Lista de precios al día	2009/04/28
West Texas Int:	74.00
Nezcla Mexicana:	75.00
Olimexa:	77.00
Istmo:	72.00
Maya:	73.00

Lista de precios al día	2009/04/27
West Texas Int:	74.00
Nezcla Mexicana:	75.00
Olimexa:	77.00
Istmo:	72.00
Maya:	73.00

Lista de precios al día	2009/04/26
West Texas Int:	64.00
Nezcla Mexicana:	65.00
Olimexa:	67.00
Istmo:	62.00
Maya:	63.00

Lista de precios al día	2009/04/25
West Texas Int:	46.00
Nezcla Mexicana:	46.00
Olimexa:	46.00
Istmo:	26.00
Maya:	36.00

Lista de precios al día	2009/04/24
West Texas Int:	44.00
Nezcla Mexicana:	45.00
Olimexa:	47.00
Istmo:	42.00
Maya:	43.00

Lista de precios al día	2009/04/23
West Texas Int:	74.00
Nezcla Mexicana:	75.00
Olimexa:	77.00
Istmo:	72.00
Maya:	73.00

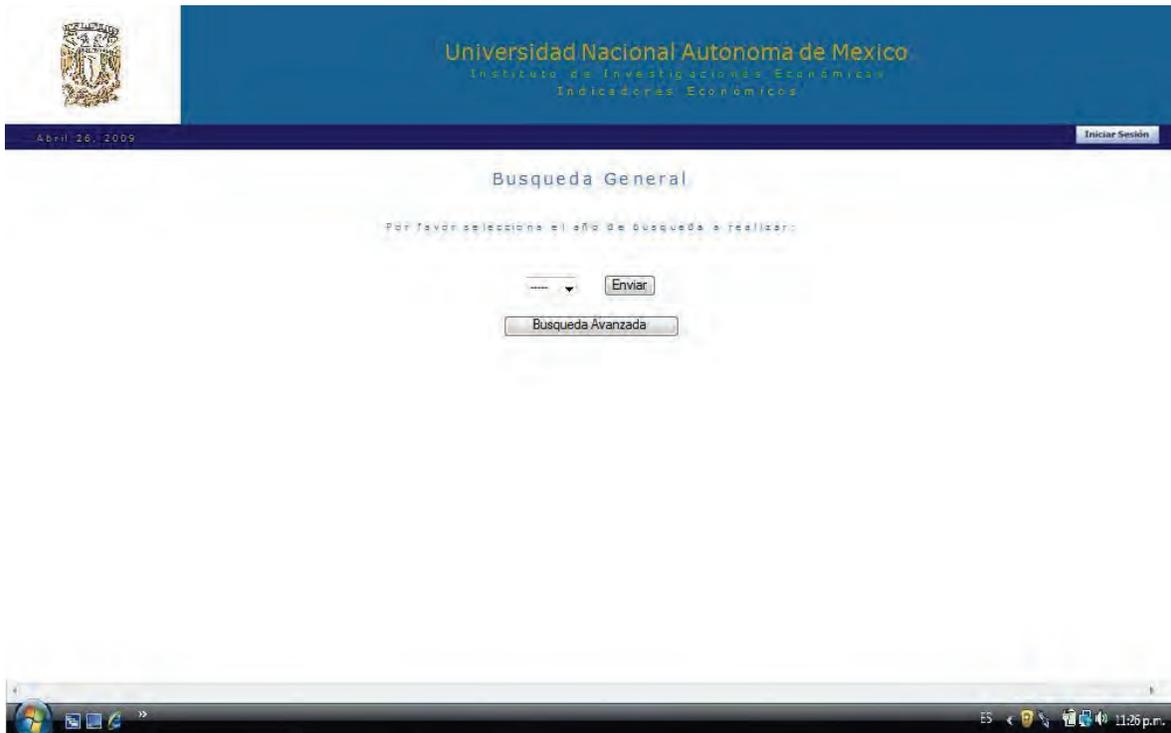
Lista de precios al día	2009/04/22
West Texas Int:	54.00
Nezcla Mexicana:	55.00
Olimexa:	57.00
Istmo:	52.00
Maya:	53.00

Lista de precios al día	2009/04/21
West Texas Int:	64.00
Nezcla Mexicana:	65.00
Olimexa:	67.00
Istmo:	62.00
Maya:	63.00

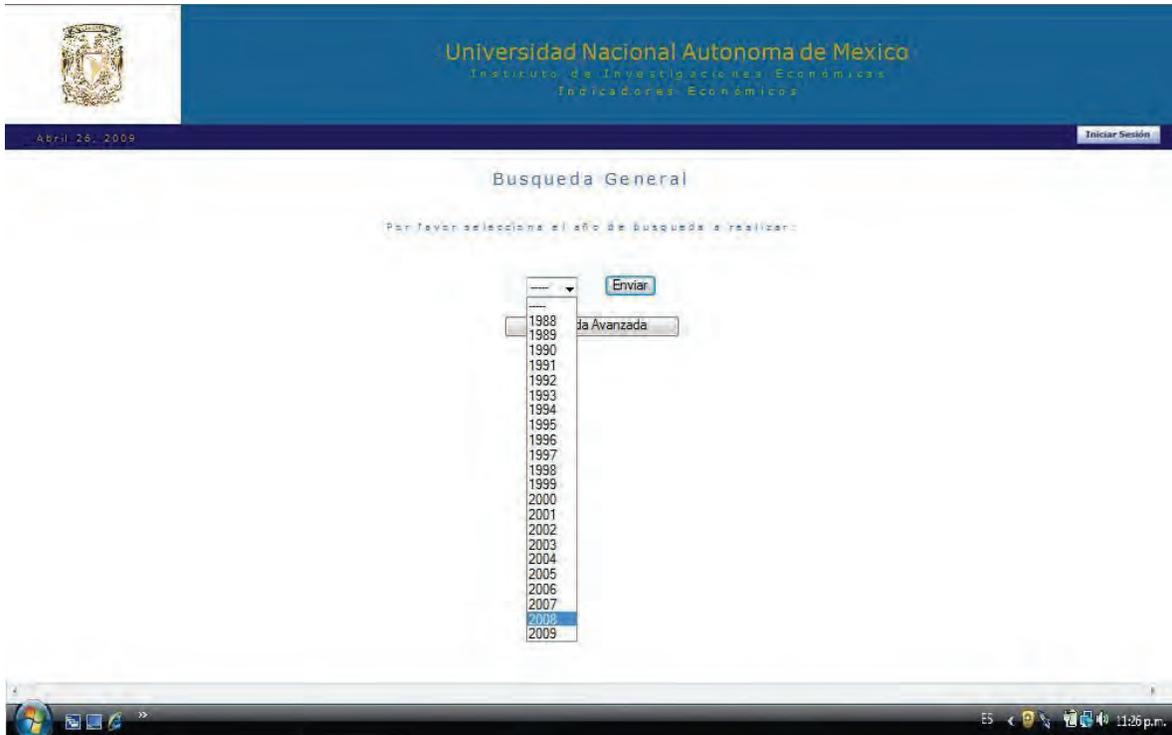


Regresar

IV.4 Modulo 4 - Búsqueda y Consulta de los valores de distintas mezclas de petróleo por día.



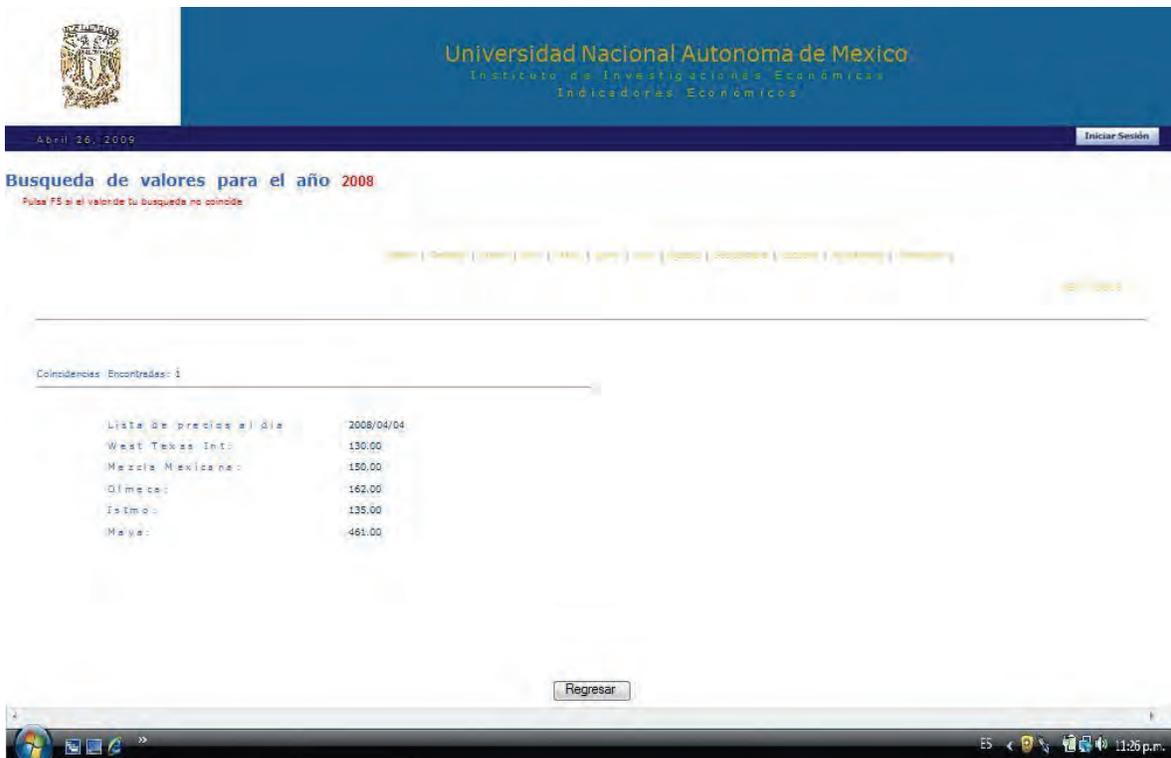
```
//=====
// Archivo:   BUSC_GENERAL.PHP
// Id:   PHP Modulo Usuario, Realiza búsquedas generales por año.
// Modulo4. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivo de Recepción: BUSQUEDA.PHP.
// Id: Recibe el valor ó fecha seleccionada para la búsqueda General
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Permite la selección de un año, para realizar la búsqueda general
// y muestra un botón ó vínculo hacia la búsqueda avanzada.
//=====
```



```
//=====
// Archivo:   BUSC_GENERAL.PHP
//
// Descripción: Al usuario le será permitido seleccionar el año de búsqueda
// (1988 – 2009) o de lo contrario acceder mediante un botón o vínculo a la
// búsqueda avanzada.
//
// Vínculo de búsqueda avanzada: BUSC_AVANZADA.PHP
//
//=====
```



```
//=====
// Archivo:   BUSQUEDA.PHP
//
// Si no existen registros: Se mostrara la siguiente leyenda cuando aún no se
// hayan ingresado registros algunos para el año elegido. << No hay registros >>.
//
//=====
//
// Si existen registros: Primeramente se mostrara el número total de coincidencias
// encontradas para el año elegido, es decir, existirá un número máximo por año,
// que será de 365 registros, uno diario.
//
// El Número total de registros será mostrado por grupos de 10 en 10 y se incluirán
// iconos para adelantar y atrasar las páginas mostradas, así como ir a la primera
// y última página
//=====
```

```
//=====
// Archivo:   BUSQUEDA.PHP
//
// Descripción: El Número total de registros será mostrado por grupos de 10 en 10
// y se incluirán iconos para adelantar y atrasar las páginas mostradas, así como ir
// a la primera y última página.
//
// Así mismo para hacer más eficiente y rápida la búsqueda de valores, se incluirá
// un menú en la parte superior derecha que permitirá al usuario realizar sus
// búsquedas por mes de acuerdo al año ingresado. Así se mostraran 30 registros
// por mes, mostrados por grupos de 10.
//=====
```



Busqueda Avanzada

Por favor seleccione la Fecha de búsqueda a realizar:

año:

Enviar



```
//=====
// Archivo:   BUSC_AVANZADA.PHP
// Id:   PHP Modulo Usuario, Realiza búsquedas Avanzadas.
// Modulo4. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivo de Recepción: R_AVANZADA.PHP.
// Id: Recibe el valor ó fecha seleccionada para la búsqueda avanzada
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Permite la selección de una fecha específica, para realizar la
// búsqueda avanzada.
//=====
```



Busqueda Avanzada

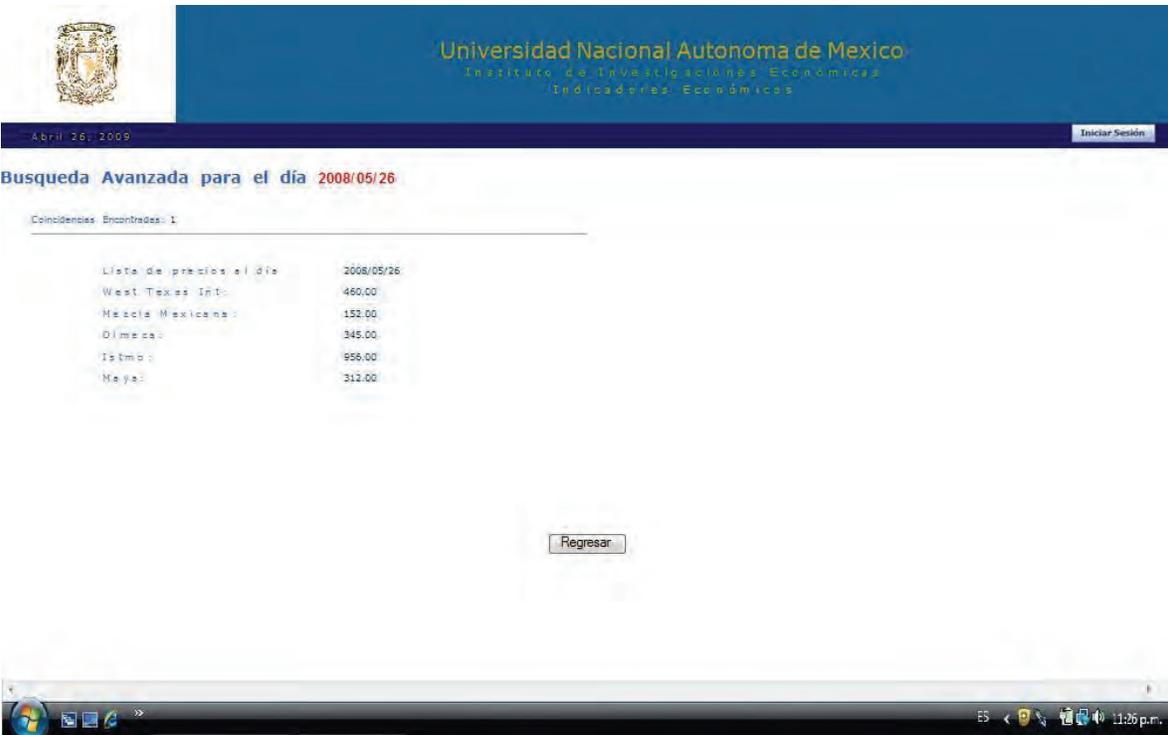
Por favor seleccione la Fecha de búsqueda a realizar:

año: 2008 05 26

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

Enviar

```
//=====
// Archivo: BUSC_AVANZADA.PHP
// Id: PHP Modulo Usuario, Realiza búsquedas Avanzadas.
// Modulo4. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivo de Recepción: R_AVANZADA.PHP.
// Id: Recibe el valor ó fecha seleccionada para la búsqueda avanzada
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Los valores de búsqueda se dividirán en tres campos AÑO el cual
// contendrá valores de 1988 hasta 2009, MES el cual contendrá los 12 meses del
// año y DIA el cual contendrá los 30 días de cada mes.
//=====
```

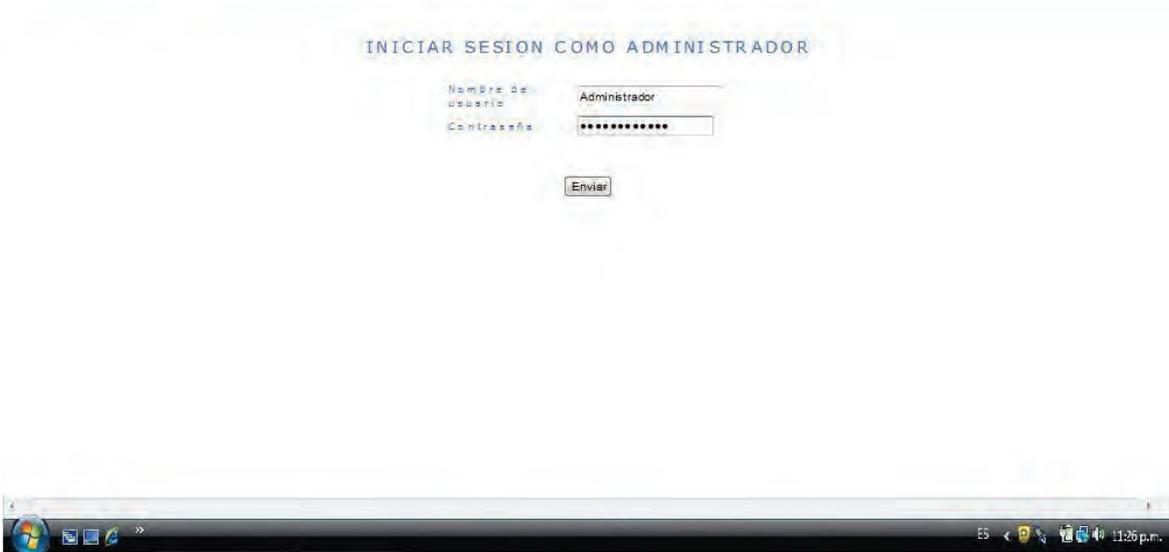


```
//=====
// Archivo de Recepción: R_AVANZADA.PHP.
// Id: Recibe el valor ó fecha seleccionada para la búsqueda avanzada
// Modulo4. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Si la fecha introducida en el archivo BUSC_AVANZADA.PHP se
// encuentra dentro de la base de datos, el sistema la mostrara junto con los
// precios por barril de cada una de las mezclas ingresadas para ese día.
//=====
```

IV.5 Modulo 5 - Formulario de Ingreso para la sesión de Administrador.



```
//=====
// Archivo:   INGRESAR.PHP
// Id:   PHP Modulo Usuario, Formulario de ingreso a la Sesión de Administrador.
// Modulo5. Interfaz de Menú de Usuario
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: El Usuario podrá tener acceso a este modulo desde cualquier otro
// con tan solo presionar el botón de inicio de sesión ubicado en parte inferior
// derecha del header adjunto a cada documento PHP creado.
//=====
```



```
//=====
// Archivo:   INGRESAR.PHP
// Id:   PHP Modulo Usuario, Formulario de ingreso a la Sesión de Administrador.
// Modulo5. Interfaz de Menú de Usuario
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Inicio de Sesión Exitoso: Se considerará exitoso si la contraseña y el nombre de
// usuario son exactos a los almacenados en la BD. El usuario será diseccionado
// al Modulo de Administración
//
// Inicio de Sesión Erróneo: El usuario permanecerá en la misma página si no
// ingresa los datos correctos.
//=====
```

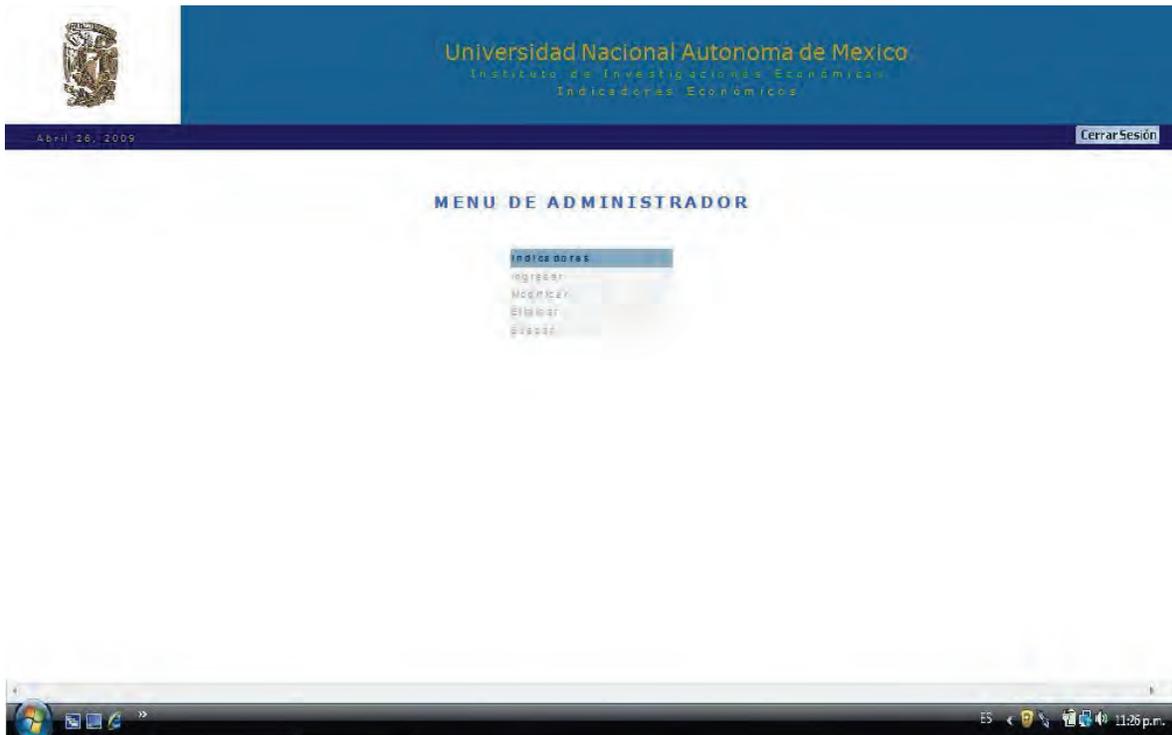


Usted no está autorizado para ver esta página en un momento será redireccionado.



```
//=====
// Archivo:   ERROR.PHP
// Id:   PHP Modulo administrador, Pagina de error de ingreso no autorizado
// Modulo5. Interfaz de Aviso y redirección al no ser reconocido como
// administrador.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario intentara ingresar a cualquiera de los documentos de
// Administración sin ser autenticado antes por su Usuario y Contraseña, el
// sistema deber impedir el acceso y solo permitirlo en caso de reconocer los
// parámetros de inicio de sesión. La redirección se realizara al documento
// INGRESAR.PHP
//=====
```

IV.6 Modulo 6 - Menú de Usuario Administrador



```
//=====
// Archivo: INDEX_ADMIN.PHP
// Id: PHP Modulo Administración, Menú de Usuario Administrador.
// Modulo6. Interfaz de Menú de Usuario Administrador
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción:
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```



MENU DE ADMINISTRADOR

Indicadores	
Ingresar	PETROLEO
Modificar	
Eliminar	
Búsqueda	



```
//=====
// Archivo: INDEX_ADMIN.PHP
// Id: PHP Modulo Administración, Menú de Usuario Administrador.
// Modulo6. Interfaz de Menú de Usuario Administrador
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción INGRESAR - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```



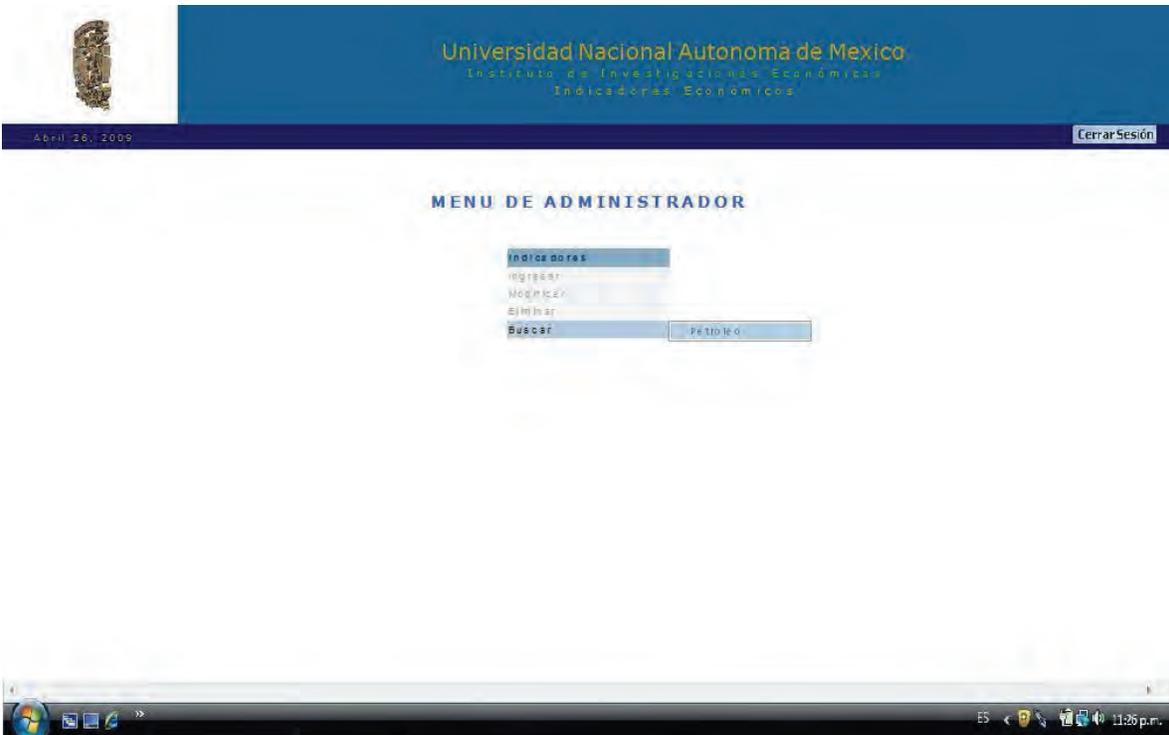
MENU DE ADMINISTRADOR



```
//=====
// Archivo: INDEX_ADMIN.PHP
// Id: PHP Modulo Administración, Menú de Usuario Administrador.
// Modulo6. Interfaz de Menú de Usuario Administrador
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción MODIFICAR - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```

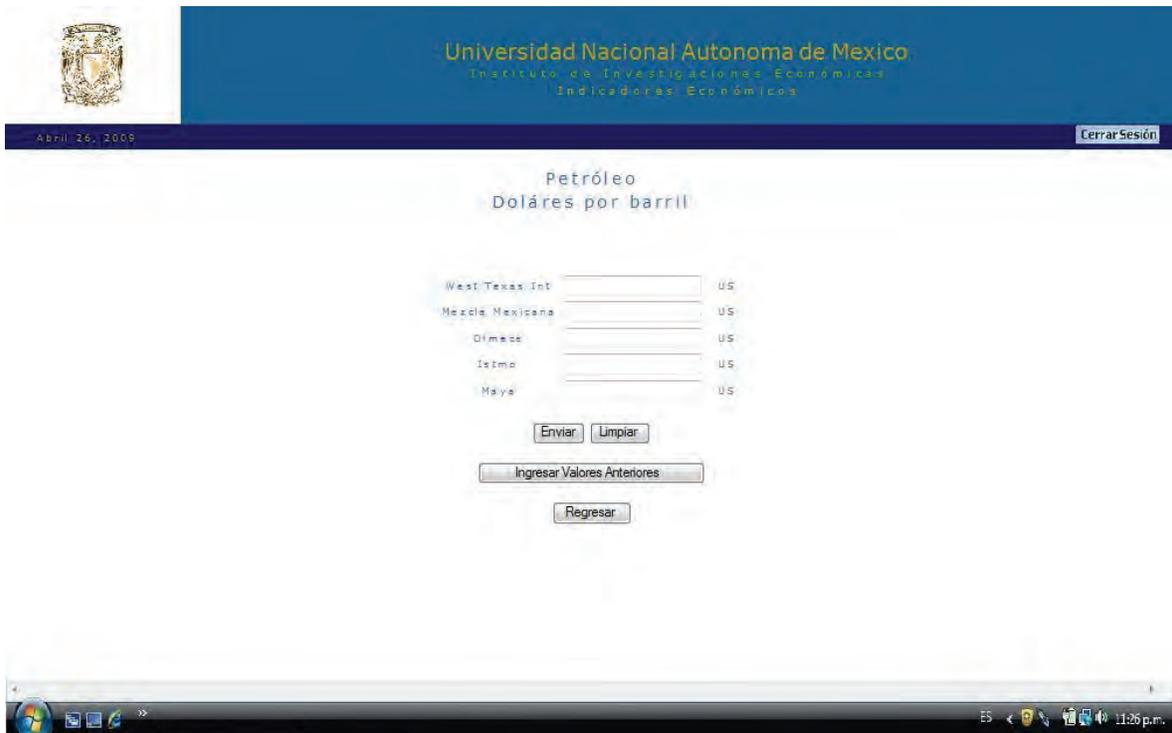


```
//=====
// Archivo: INDEX_ADMIN.PHP
// Id: PHP Modulo Administración, Menú de Usuario Administrador.
// Modulo6. Interfaz de Menú de Usuario Administrador
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción ELIMINAR - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```

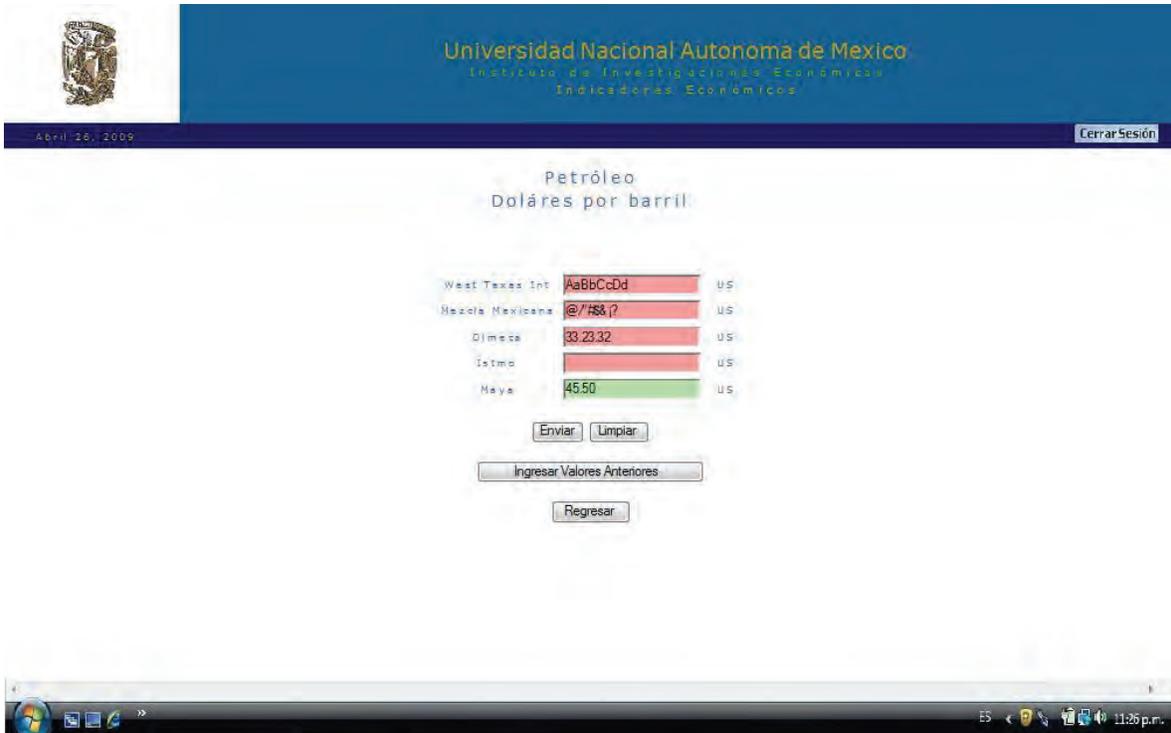


```
//=====
// Archivo: INDEX_ADMIN.PHP
// Id: PHP Modulo Administración, Menú de Usuario Administrador.
// Modulo6. Interfaz de Menú de Usuario Administrador
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Despliegue de menú, opción BUSCAR - PETROLEO
// ORC_JS, JavaScript Class Framework
// version:3.00.71113
// Copyright 2007 by SourceTec Software Co.,LTD
// For more information, see:www.sothink.com
//=====
```

IV.7 Modulo 7 - Formulario de Ingreso para nuevos registros del precio por barril de petróleo al día.



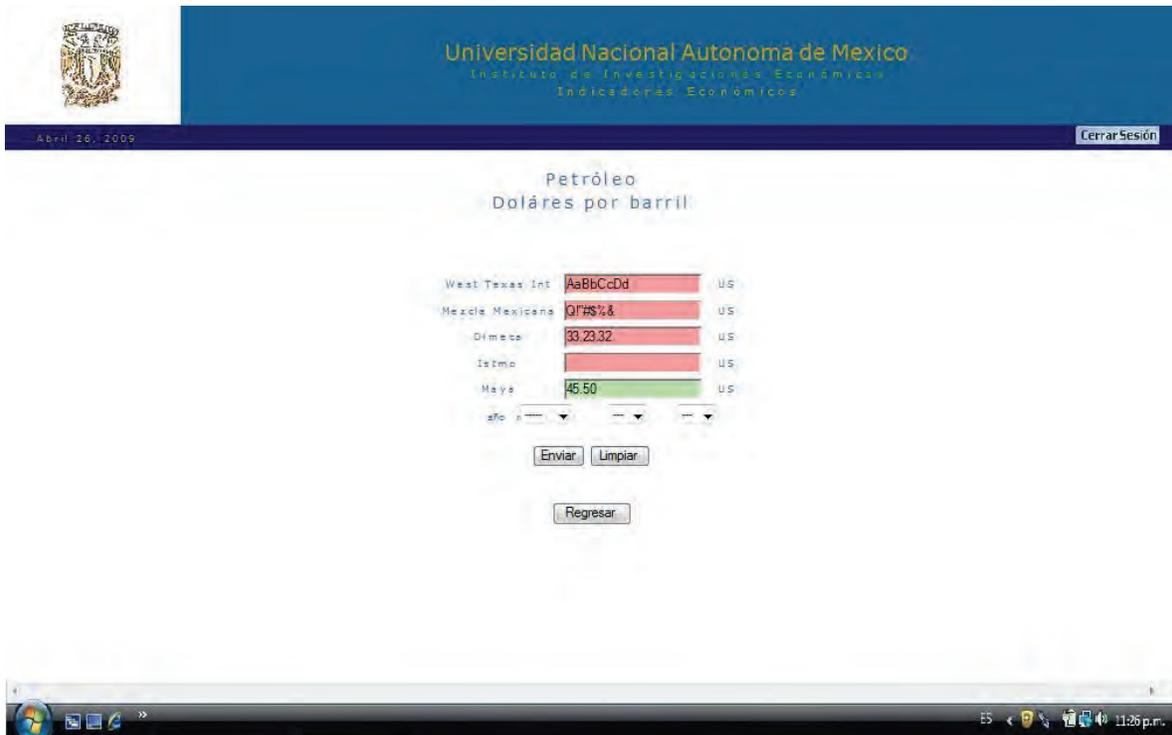
```
//=====
// Archivo:    PETROLEO.PHP
// Id:    PHP Modulo Administrador, Ingresar el registro y los valores del día.
// Modulo7. Interfaz Grafica para ingresar los valores correspondientes al día.
// Modificado y Creado:    2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Permite el envío de los precios por barril de petróleo al día, esto
// quiere decir que se enviara la fecha del día automáticamente y el usuario hará
// uso de este formulario una vez al día.
//=====
```



```
//=====
// Archivo:   PETROLEO.PHP
// Id:   PHP Modulo Administrador, Ingresa el registro y los valores del día.
// Modulo7. Interfaz Grafica para ingresar los valores correspondientes al día.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: El formulario deberá ser validado antes del envío. No se aceptará
// Letras (AaBbCcDdEe), no se aceptará caracteres especiales ("/'#%&i?"), no se
// aceptarán cantidades erróneas (33.29.32), no se aceptarán espacios ni
// formularios sin contenido.
//=====
```



```
//=====
// Archivo:   PETROLEO_CAPT_1.PHP
// Id:   PHP Modulo Administrador, Ingresar el registro y los valores del día.
// Modulo7. Interfaz Grafica para ingresar los registros de cualquier día.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Permite el envío de los precios por barril de petróleo para cualquier
// día, esto quiere decir que se enviara la fecha del día que nosotros
// seleccionemos. Esta opción fue diseñada para hacer un historial completo e
// ingresar registros de días anteriores.
//=====
```



```
//=====
// Archivo:   PETROLEO_CAPT_1.PHP
// Id:   PHP Modulo Administrador, Ingresa el registro y los valores del día.
// Modulo7. Interfaz Grafica para ingresar los valores correspondientes al día.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: El formulario deberá ser validado antes del envío. No se aceptará
// Letras (AaBbCcDdEe), no se aceptará caracteres especiales ("!/#%&i?"), no se
// aceptarán cantidades erróneas (33.29.32), no se aceptarán espacios ni
// formularios sin contenido.
//=====
```



```
//=====
// Archivo:   INSERTADO.PHP
// Id:   PHP Modulo administrador, Pagina de ingreso satisfactorio
// Modulo7. Interfaz de Aviso y redirección al ser almacenado correctamente un
// registro como administrador.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario ingresara un registro cualquiera y es satisfactorio el
// proceso de inserción, el sistema dará un aviso al usuario administrador y
// realizara una redirección al documento INSERTADO.PHP
//=====
```



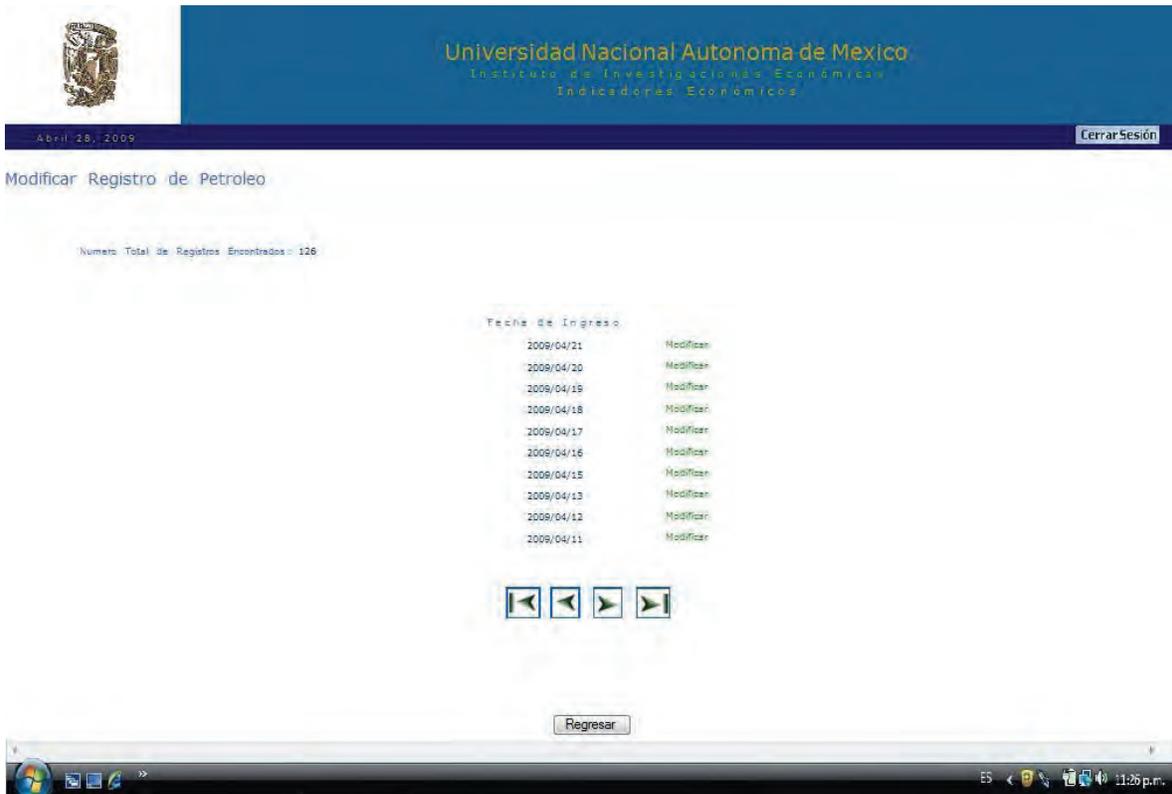
Esta Fecha de Registro ya ha sido Almacenada..

« INGRESE OTRO REGISTRO »



```
//=====
// Archivo:   ERROR.PHP
// Id:   PHP Modulo administrador, Pagina de error de ingreso repetido
// Modulo7. Interfaz de Aviso y redirección al no ser reconocido como
// administrador.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario intenta ingresar un registro repetido, el sistema
// alertara al usuario de que la fecha que intenta ingresar ya ha sido almacenada
// con anterioridad.
//=====
```

IV.8 Modulo 8 - Consulta, Imprime y modifica el valor de distintas mezclas del valor de petróleo por día.



```
//=====
// Archivo: SHOWPETROLEO_MOD.PHP
// Id: PHP Modulo Administrador, Visualiza los registros insertados por día.
// Modulo8. Interfaz Grafica para la consulta y modificacion de los registros
// insertados.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra un contador y la fecha de inserción para cada registro
// ordenados de manera descendente. Para cada registro se muestra un vínculo
// para poder ser modificados. El Número total de registros será mostrado por
// grupos de 10 en 10 y se incluirán iconos para adelantar y atrasar las páginas
// mostradas, así como ir a la primera y última página.
//=====
```



Modificar precios de petroleo

Id[_petroleo]	00.00000147
West Texas Int	64.00 US
Mezcla Mexicana	65.00 US
Olmeca	67.00 US
Istmo	62.00 US
Maya	63.00 US

Actualizar registro

Regresar



```
//=====
// Archivo: MODIFICARPETROLEO.PHP
// Id: PHP Modulo Administrador, Visualiza el valor de las mezclas de petróleo
// por día.
// Modulo8. Interfaz Grafica para la modificación del registro seleccionado.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra un contador del total de registros insertados y el valor o
// precio almacenado para cada una de las mezclas por fecha de inserción,
// ordenados de manera descendente.
// El Número total de registros será mostrado por grupos de 10 en 10 y se incluirán
// iconos para adelantar y atrasar las páginas mostradas, así como ir a la primera
// y última página
//=====
```

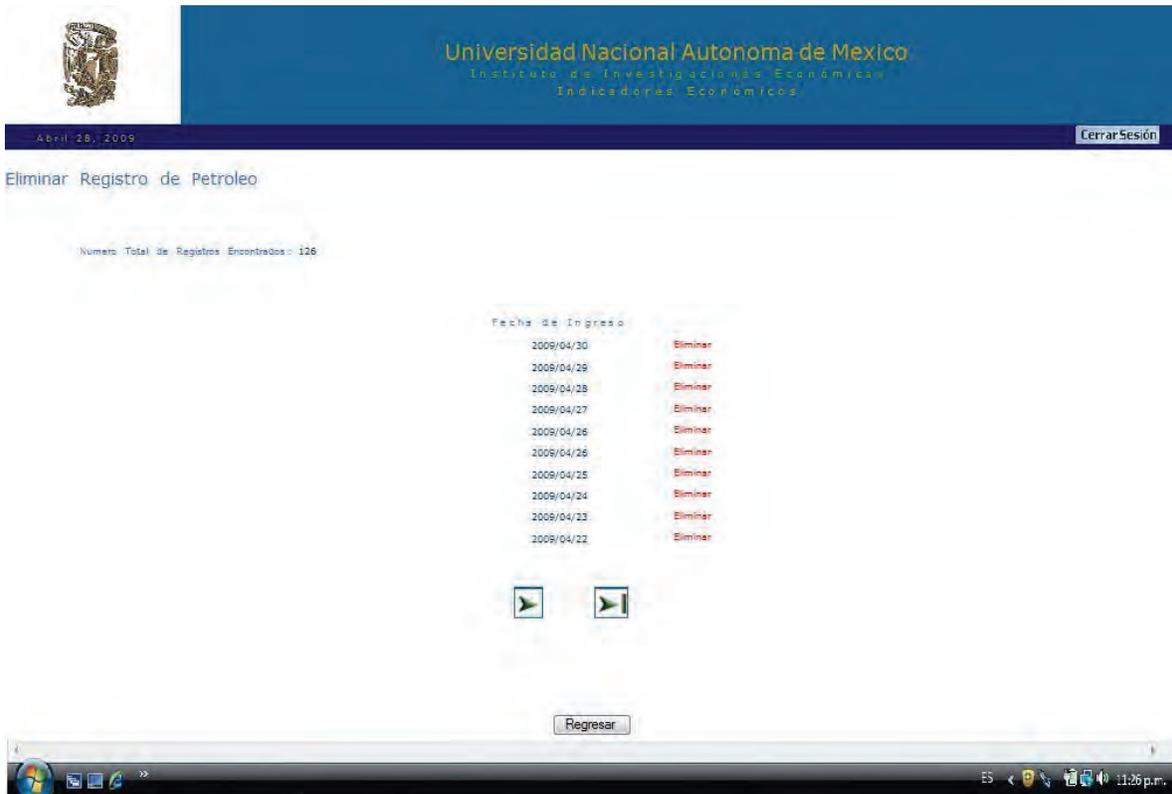


El Registro ha sido Almacenado
ESPERE UN MOMENTO...



```
//=====
// Archivo: INSERTADO.PHP
// Id: PHP Modulo Administrador, Pagina de ingreso satisfactorio
// Modulo8. Interfaz de Aviso y redirección al ser almacenado correctamente un
// registro como administrador.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario ingresara un registro cualquiera y es satisfactorio el
// proceso de inserción, el sistema dará un aviso al usuario administrador y
// realizara una redirección al documento INSERTADO.PHP
//=====
```

IV.9 Modulo 9 - Consulta, Imprime y elimina el valor de distintas mezclas del valor de petróleo por día.



```
//=====
// Archivo:   PETROLEO_DEL.PHP
// Id:   PHP Modulo Administrador, Visualiza los registros insertados por día.
// Modulo9. Interfaz Grafica para la consulta y eliminación de los registros
// insertados.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra un contador y la fecha de inserción para cada registro
// ordenados de manera descendente. Para cada registro se muestra un vinculo
// para poder ser eliminados. El Número total de registros será mostrado por
// grupos de 10 en 10 y se incluirán iconos para adelantar y atrasar las páginas
// mostradas, así como ir a la primera y última página.
//=====
```



El Registro ha sido Eliminado
ESPERE UN MOMENTO.....

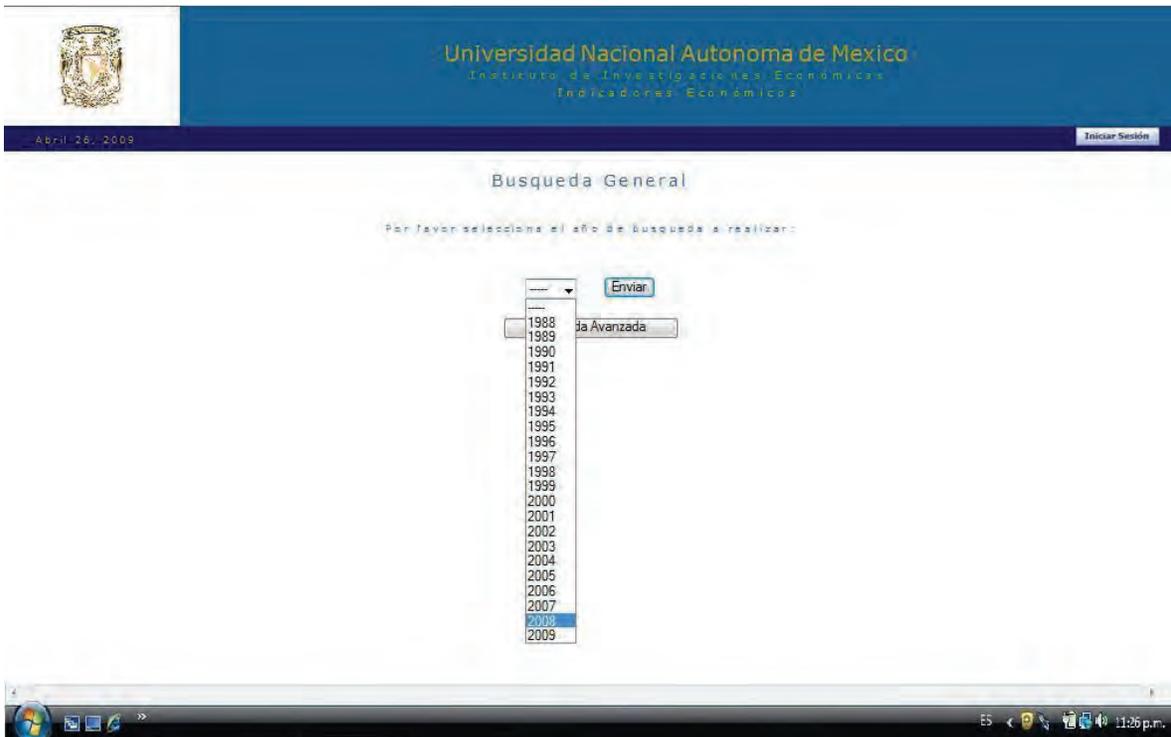


```
//=====
// Archivo: ELIMINADO.PHP
// Id: PHP Modulo Administrador, Pagina de eliminación satisfactoria
// Modulo9. Interfaz de Aviso y redirección al ser eliminado correctamente un
// registro como administrador.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario eliminara un registro cualquiera y es satisfactorio el
// proceso de eliminación, el sistema dará un aviso al usuario administrador y
// realizara una redirección al documento ELIMINADO.PHP
//=====
```

IV.10 Modulo 10 - Búsqueda y Consulta de los valores de distintas mezclas de petróleo por día.



```
//=====
// Archivo:   BUSC_GEN_ADMIN.PHP
// Id:   PHP Modulo Administrador, Realiza búsquedas generales por año.
// Modulo10. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivo de Recepción: R_GEN_ADMIN.PHP
// Id: Recibe el valor ó fecha seleccionada para la búsqueda General
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Permite la selección de un año, para realizar la búsqueda general
// y muestra un botón ó vínculo hacia la búsqueda avanzada.
//=====
```



```
//=====
// Archivo:   BUSC_GEN_ADMIN.PHP
//
// Descripción: Al Administrador le será permitido seleccionar el año de búsqueda
// (1988 – 2009) o de lo contrario acceder mediante un botón o vínculo a la
// búsqueda avanzada.
//
// Vínculo de búsqueda avanzada: BUSC_AVA_ADMIN.PHP
//=====
```



Busqueda de valores para el año

Pulse F5 si el valor de tu búsqueda no coincide



NO HAY REGISTROS

Regresar



```
//=====
// Archivo:   R_GEN_ADMIN.PHP
//
// Si no existen registros: Se mostrara la siguiente leyenda cuando aún no se
// hayan ingresado registros algunos para el año elegido. << No hay registros >>.
//
//=====
//
// Si existen registros: Por un lado se mostrara el número total de coincidencias
// encontradas para el año elegido, es decir, existirá un número máximo por año,
// que será de 365 registros, uno diario. Por otra parte se mostrarían las opciones
// de modificación y eliminación para cada registro mostrado
//
// El Número total de registros será mostrado por grupos de 10 en 10 y se incluirán
// iconos para adelantar y atrasar las páginas mostradas, así como ir a la primera
// y última página
//=====
```




Busqueda de valores para el año 2008

Pulse F5 si el valor de su búsqueda no coincide



Coincidencias Encontradas: 5

Lista de precios a día	2008/04/04	Modificar	Eliminar
West Texas Int:	130.00		
Mezcla Mexicana:	150.00		
Qlmico:	162.00		
Isimco:	135.00		
Maya:	461.00		

Regresar



```
//=====
// Archivo: BUSQUEDA.PHP
//
// Descripción: El Número total de registros será mostrado por grupos de 10 en 10
// y se incluirán iconos para adelantar y atrasar las páginas mostradas, así como ir
// a la primera y última página.
//
// Así mismo para hacer más eficiente y rápida la búsqueda de valores, se incluirá
// un menú en la parte superior derecha que permitirá al usuario realizar sus
// búsquedas por mes de acuerdo al año ingresado. Así se mostrarán 30 registros
// por mes, mostrados por grupos de 10; habilitando las opciones de modificación
// y eliminación para cada registro mostrado.
//
//=====
```



Busqueda Avanzada

Por favor seleccione la Fecha de búsqueda a realizar:

año: mes: día:

Enviar



```
//=====
// Archivo:   BUSC_AVA_ADMIN.PHP
// Id:   PHP Modulo Administrador, Realiza búsquedas Avanzadas.
// Modulo10. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivo de Recepción: R_AVA_ADMIN.PHP
// Id: Recibe el valor ó fecha seleccionada para la búsqueda avanzada
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Permite la selección de una fecha específica, para realizar la
// búsqueda avanzada.
//=====
```



Busqueda Avanzada

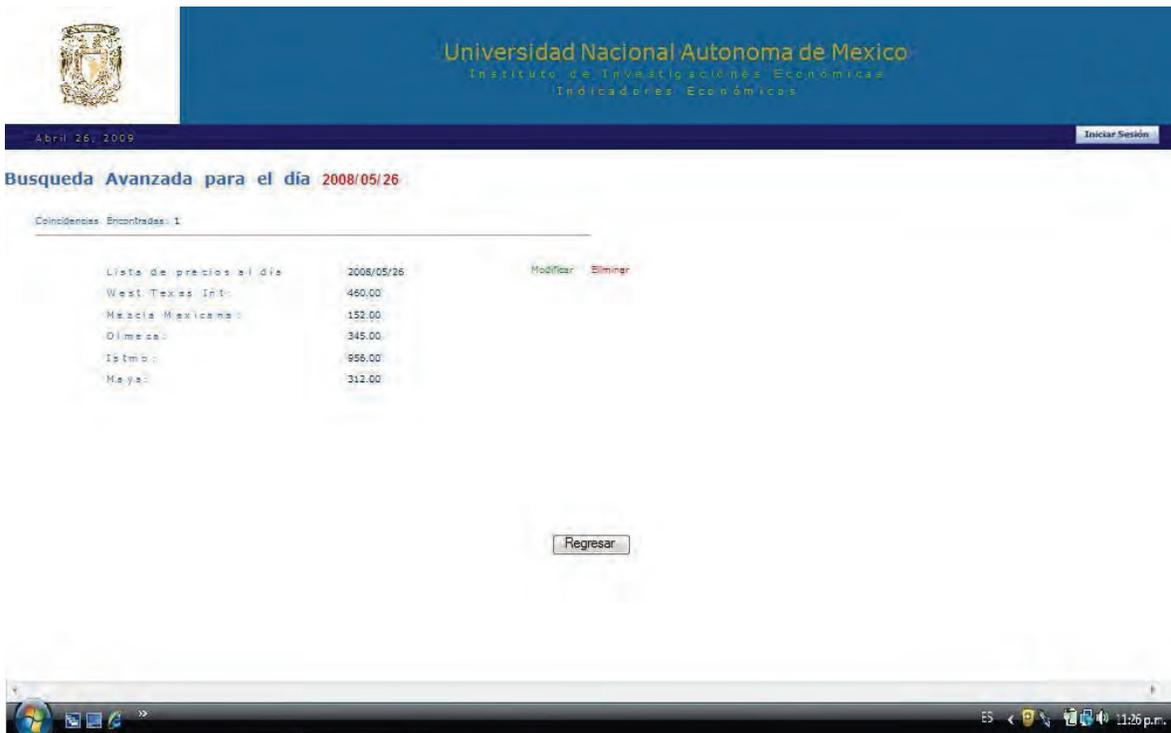
Por favor seleccione la Fecha de búsqueda a realizar:

año: 2008 mes: 05 día: 26

Enviar

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

```
//=====
// Archivo:   BUSC_AVA_ADMIN.PHP
// Id:   PHP Modulo Administrador, Realiza búsquedas Avanzadas.
// Modulo10. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Archivo de Recepción: R_AVA_ADMIN.PHP
// Id: Recibe el valor ó fecha seleccionada para la búsqueda avanzada
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Los valores de búsqueda se dividirán en tres campos AÑO el cual
// contendrá valores de 1988 hasta 2009, MES el cual contendrá los 12 meses del
// año y DIA el cual contendrá los 30 días de cada mes.
//=====
```



```
//=====
// Archivo de Recepción: R_AVANZADA.PHP.
// Id: PHP Modulo Administrador, Recibe el valor ó fecha seleccionada para la
// búsqueda avanzada.
// Modulo10. Interfaz Grafica para búsqueda de valores y selección avanzada.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo adjunto:
// C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Si la fecha introducida en el archivo BUSC_AVANZADA.PHP se
// encuentra dentro de la base de datos, el sistema la mostrara junto con los
// precios por barril de cada una de las mezclas ingresadas para ese día.
//=====
```



Modificar precios de petroleo

Id[_petroleo]	00.00000147
West Texas Int	64.00 US
Mezcla Mexicana	65.00 US
Olmeca	67.00 US
Istmo	62.00 US
Maya	63.00 US

Actualizar registro

Regresar



```
//=====
// Archivo: MODIFICARPETROLEO.PHP
// Id: PHP Modulo Administrador, Visualiza el valor de las mezclas de petróleo
// por día.
// Modulo10. Interfaz Grafica para la modificacion del registro seleccionado.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema\petroleo
//
// Descripción: Muestra un contador del total de registros insertados y el valor o
// precio almacenado para cada una de las mezclas por fecha de inserción,
// ordenados de manera descendente.
// El Número total de registros será mostrado por grupos de 10 en 10 y se incluirán
// iconos para adelantar y atrasar las páginas mostradas, así como ir a la primera
// y última página
//=====
```



El Registro ha sido Almacenado
ESPERE UN MOMENTO...



```
//=====
// Archivo:   INSERTADO.PHP
// Id:   PHP Modulo Administrador, Pagina de ingreso satisfactorio
// Modulo10. Interfaz de Aviso y redirección al ser almacenado correctamente un
// registro como administrador.
// Modificado y Creado:   2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario ingresara un registro cualquiera y es satisfactorio el
// proceso de inserción, el sistema dará un aviso al usuario administrador y
// realizara una redirección al documento INSERTADO.PHP
//=====
```



```
//=====
// Archivo: ELIMINADO.PHP
// Id: PHP Modulo Administrador, Pagina de eliminación satisfactoria
// Modulo10. Interfaz de Aviso y redirección al ser eliminado correctamente un
// registro como administrador.
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario eliminara un registro cualquiera y es satisfactorio el
// proceso de eliminación, el sistema dará un aviso al usuario administrador y
// realizara una redirección al documento ELIMINADO.PHP
//=====
```



PROYECTO DE INDICADORES
Gracias por su visita.
DESCONECTANDO ...



```
//=====
// Archivo: SALIR.PHP
// Id: PHP Modulo Administrador, Pagina de salida del sistema
// Modulo10. Interfaz de Aviso y redirección al ser desconectado del sistema de
// administración
// Modificado y Creado: 2008-06-13 Yair Munive - yair.unam@gmail.com
// Ubicación de archivo: C:\apache2triad\htdocs\indicadores\sistema
//
// Descripción: Si el usuario da click en los botones de cerrar sesión, el sistema
// dará un aviso al usuario administrador y realizara una redirección al documento
// SALIR.PHP, que desconectará al usuario del sistema.
//=====
```

CONCLUSIONES

La programación de aplicaciones Web constituye una tarea multidisciplinaria en la cual es necesario contar con conocimientos de informática, programación y diseño gráfico.

Los conocimientos de informática son necesarios para saber diseñar e implementar soluciones basadas en sistemas de información, que faciliten la toma de decisiones y agilicen las operaciones propias de dicha institución. Esta labor debe adaptarse a los avances tecnológicos y a los cambios socioeconómicos tanto en el plano nacional como internacional.

Así mismo lograr la satisfacción de las necesidades y logro de objetivos, a través del aprovechamiento adecuado de los recursos informáticos, mediante la planeación, la organización, la dirección y el control de las actividades y labores que nos permitirán cumplir con nuestros objetivos.

Los conocimientos en programación, no solo permitirán especificar de manera precisa sobre los datos que debe operar una computadora, como deben ser almacenados o transmitidos y que acciones debe tomar bajo alguna circunstancia, sino que también nos permitirá crear programas, mediante la aplicación de procedimientos lógicos, a través de: El análisis, desarrollo y diseño lógico del programa para resolver un problema en particular; La escritura de la lógica del programa empleado un lenguaje de programación específico; El ensamblaje o compilación del programa; La prueba y depuración del programa; y el desarrollo de la documentación.

Y finalmente los conocimientos en diseño gráfico, siendo de gran importancia, proyectar, coordinar, seleccionar y organizar un conjunto de elementos para producir y crear objetos visuales destinados a comunicar mensajes específicos a grupos determinados. Los aspectos básicos a considerar en un diseño son: El Lenguaje visual, que permite saber comunicar el mensaje adecuado, con los recursos oportunos, dependiendo del grupo de personas o público al que vaya

dirigido el mensaje; La comunicación, que permite conocer los procesos de comunicación, para poder captar los mensajes que el diseño ha de comunicar; La percepción visual, que permite estar informados, la manera en que las personas vemos y percibimos lo que vemos. Aspectos tan importantes, como nuestro campo visual, el recorrido de la vista, el contraste, la percepción de las figuras, fondos y trayectoria de la luz; Y la administración de recursos, que permite conocer todos los recursos de los que se dispone, y aplicarlos lo mejor posible.

El realizar un análisis para crear un sitio Web dinámico funcional que permitiera manipular los indicadores de petróleo para el Instituto de Investigaciones Económicas, significó: detectar las necesidades a solucionar, sugerir una alternativa para solucionar estas necesidades, definir cuáles eran los objetivos de este nuevo proyecto y determinar las herramientas de trabajo.

La fase de diseño del sitio Web comprendió el definir cuestiones lógicas como era el tipo de navegación para usuarios y administradores, la estructura del servidor de archivos y base de datos que sirviera para el funcionamiento y almacenaje del sitio.

La generación de código correspondió a lo que se estudió en el presente trabajo del sitio Web, donde se puede concluir que antes de realizar cualquier hoja de contenido es importante conocer la estructura lógica que tendrá el sitio Web, las funciones que realizará. El siguiente paso será realizar el desarrollo de éste, es decir comenzar elaborando la estructura física de los documentos que integran el sitio, para decidir qué elementos se encontrarán visibles para el usuario y el administrador, cuáles serán los elementos de exploración y donde estarán ubicados para facilitar la navegación a lo largo del sitio, el tipo de vínculos a utilizar, así como la elección de estilos que permita establecer uniformidad a lo largo de todos los documentos que integran el sitio, pero que a su vez permita un fácil lectura a través de la elección de una dimensión y fuentes apropiados.

Definidos la estructura del sitio Web y el modo en que ésta va a controlar la navegación y desplegar el contenido, es posible comenzar la generación de contenido, es decir, implementar Querys de Base de Datos, documentos *PHP*, menús de navegación, imágenes y animaciones.

Finalmente, puedo concluir que un sitio publicado requiere de continuas actualizaciones de contenido, por lo que en este proyecto es necesaria la intervención de un usuario administrador. Son muchos quienes cometen el error de subestimar este aspecto de vital importancia para toda iniciativa seria en Internet.

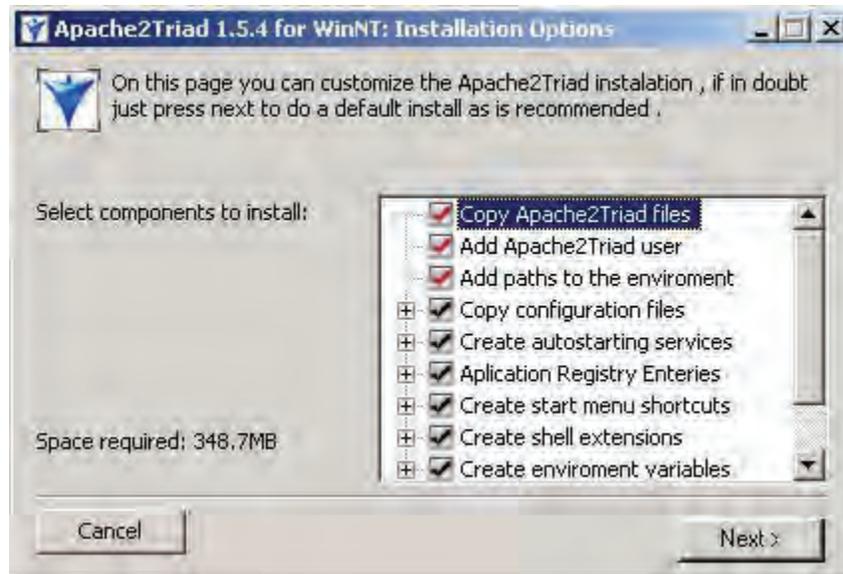
ANEXOS

ANEXO 1 – INSTALACIÓN DE APACHE2TRIAD

Una vez seleccionado el archivo de instalación “*apache2triad1.5.4.exe*” que se encuentra en el disco de instalación, lo ejecutamos y a continuación nos aparecerá la siguiente ventana y seleccionaremos la opción Run.



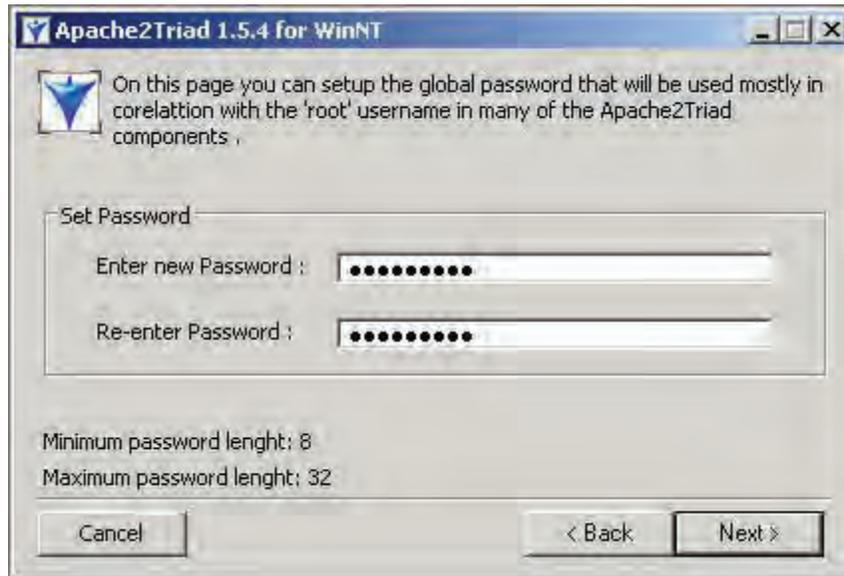
En nuestra siguiente ventana se mostrara una lista de los componentes que queremos instalar y a continuación pulsaremos el botón Next. (En nuestro caso no modificaremos algo, solo daremos click en Next).



A continuación se muestra la Ruta en donde se guardaran las configuraciones y carpetas para poder trabajar con nuestro sistema. Por default nos aparece *C:\apache2triad* y trabajaremos con esa misma dirección.



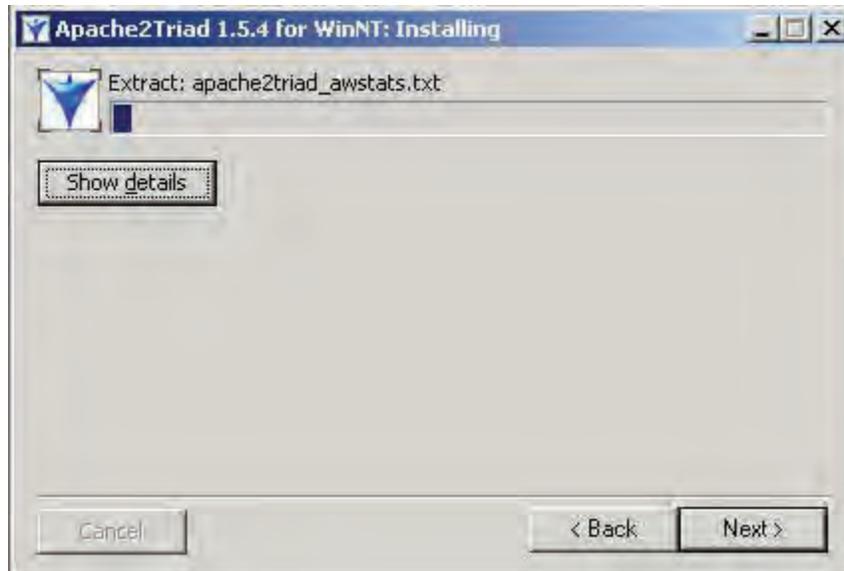
A continuación registraremos una contraseña global para poder trabajar con todas nuestras aplicaciones y es recomendable anotarla y guardarla en un lugar donde podamos acceder a ella en caso de olvidarla.



A continuación se muestra la pantalla de Licencia, una vez estando de acuerdo con los términos pulsaremos el botón Next.



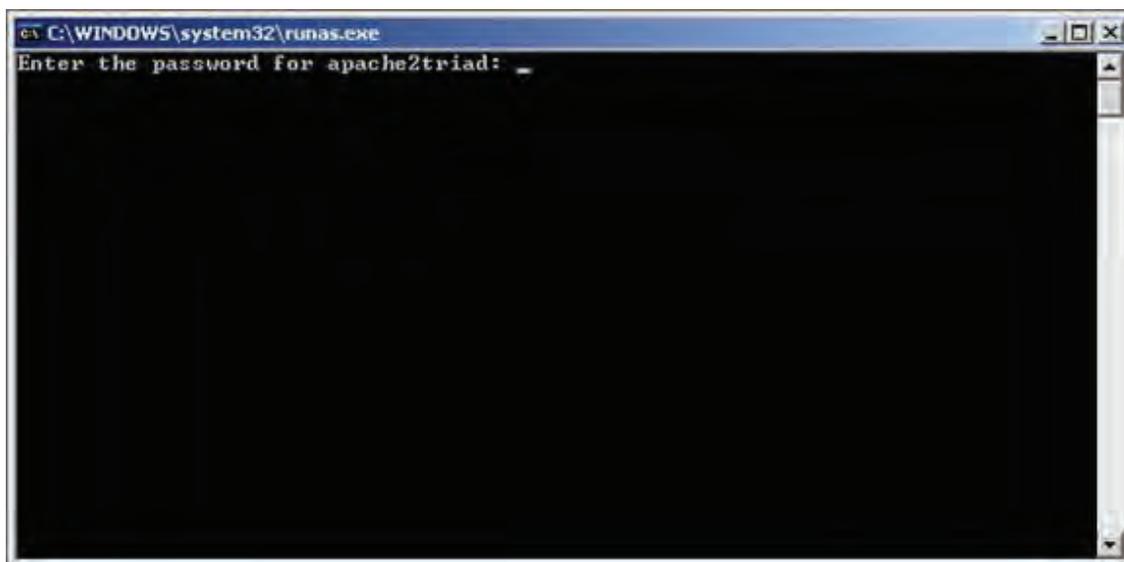
Ya casi por terminar nuestra intervención en la instalación, se nos mostrara una pantalla de avance de la instalación de *Apache2triad*, esperaremos a que termine de instalarse.



Después de haber instalado los componentes de *Apache2triad*, pulsaremos ok para hacer la configuración final del mismo.



Esta siguiente pantalla, la cual de muestra en modo consola, nos pedirá que ingresemos la contraseña global que establecimos en pasos anteriores para que podamos manejar y asegurar nuestra información mediante esta.



Una vez ingresada nuestra contraseña el sistema creara directorios, archivos de configuración, etc. Proceso en el cual no es necesaria nuestra intervención.

```
C:\apache2triad\pgsql\bin\initdb.exe -U root -D C:\apache2triad\pgsql\data
creating directory C:/apache2triad/pgsql/data ... ok
creating directory C:/apache2triad/pgsql/data/global ... ok
creating directory C:/apache2triad/pgsql/data/pg_xlog ... ok
creating directory C:/apache2triad/pgsql/data/pg_xlog/archive_status ... ok
creating directory C:/apache2triad/pgsql/data/pg_clog ... ok
creating directory C:/apache2triad/pgsql/data/pg_subtrans ... ok
creating directory C:/apache2triad/pgsql/data/pg_twophase ... ok
creating directory C:/apache2triad/pgsql/data/pg_multixact/members ... ok
creating directory C:/apache2triad/pgsql/data/pg_multixact/offsets ... ok
creating directory C:/apache2triad/pgsql/data/base ... ok
creating directory C:/apache2triad/pgsql/data/base/1 ... ok
creating directory C:/apache2triad/pgsql/data/pg_tblspc ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 1000
creating configuration files ... ok
creating template1 database in C:/apache2triad/pgsql/data/base/1 ... ok
initializing pg_authid ... ok
enabling unlimited row size for system tables ... ok
initializing dependencies ... ok
creating system views ... ok
loading pg_description ... ok
creating conversions ... ok
setting privileges on built-in objects ...
```

Finalmente se nos mostrara la siguiente pantalla y pulsaremos el botón OK para que el sistema pueda ser reiniciado. Se recomienda guardar cualquier aplicación, archivos o programas en uso antes de reiniciar el sistema, ya que esto puede causar perdida de algunos datos.

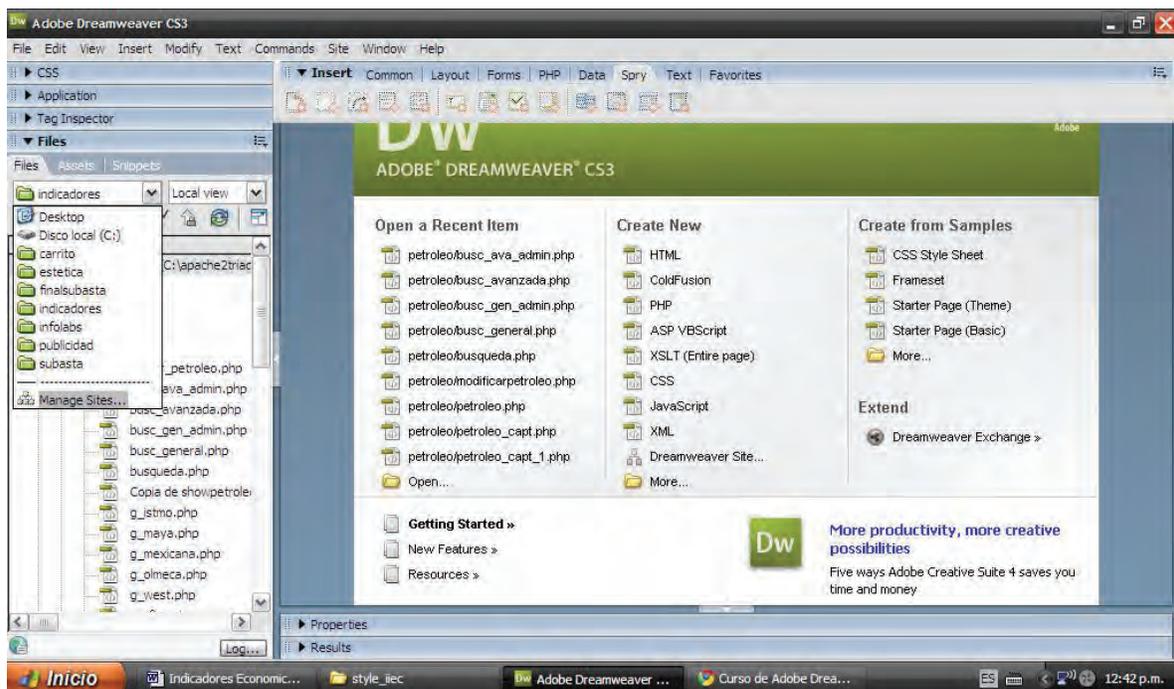


Finalizado este proceso se modificara automáticamente las aplicaciones y podremos trabajar de manera local. Para comprobar que la aplicación fue realizada con éxito iniciaremos nuestro Explorador o Navegador Web y escribiremos la siguiente dirección <http://localhost/> y visualizaremos el índice de *APACHE2TRIAD*.

ANEXO 2 – CREAR UN SERVIDOR DE PRUEBAS EN DEAMWEAVER

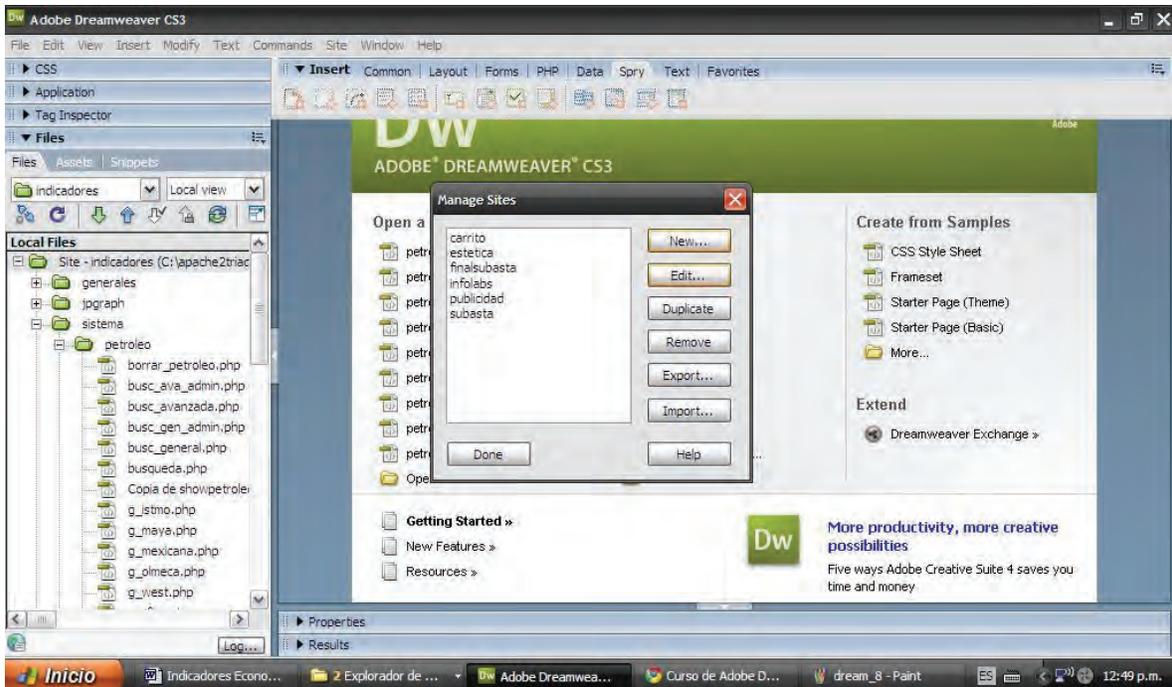
Crear un servidor local de pruebas nos facilitara el uso y ejecución de nuestro código. Simplemente bastara con Presionar F12 y nuestro código se ejecutara automáticamente y podremos visualizar nuestra aplicación web en el Explorador que tengamos predeterminado.

Lo primero que se debe realizar para crear nuestro servidor de pruebas, será seleccionar en el menú de Files la opción Manage Sites ...



Una vez seleccionado aparecerá una nueva ventana que nos permitirá crear, editar, duplicar, remover, importar y exportar sitios de prueba.

Una vez aclarado seleccionaremos la opción New y crearemos un nuevo sitio.

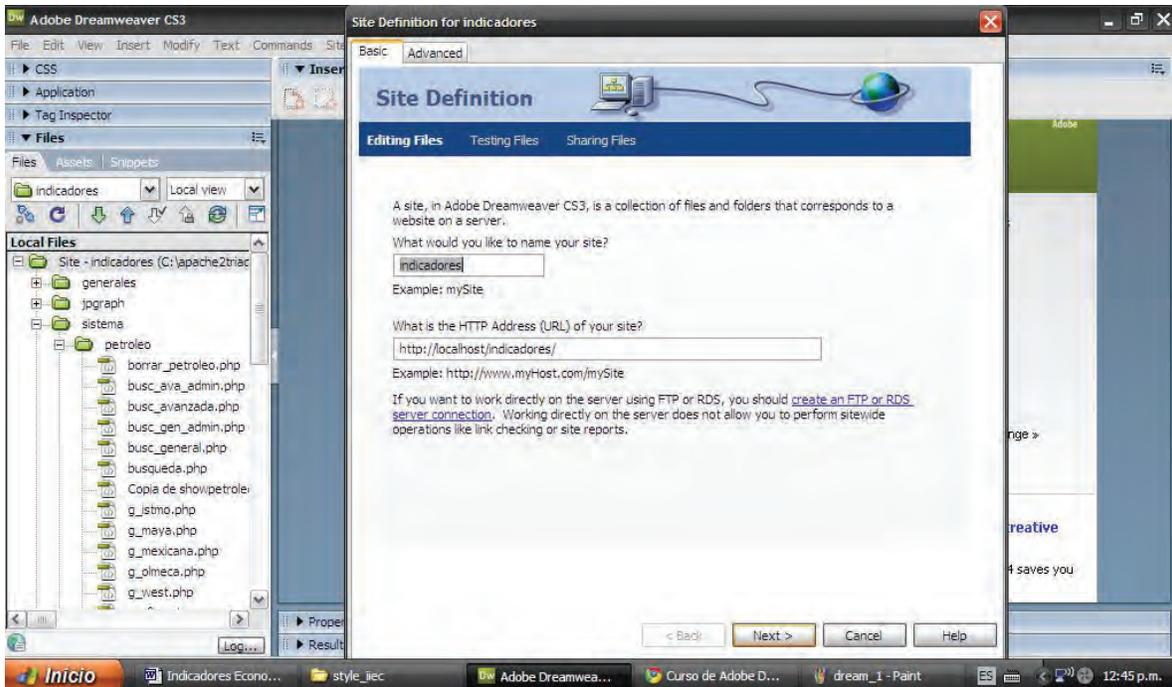


Seleccionada la opción New aparecerá una opción dentro de la cual seleccionaremos la opción de Site o sitio.

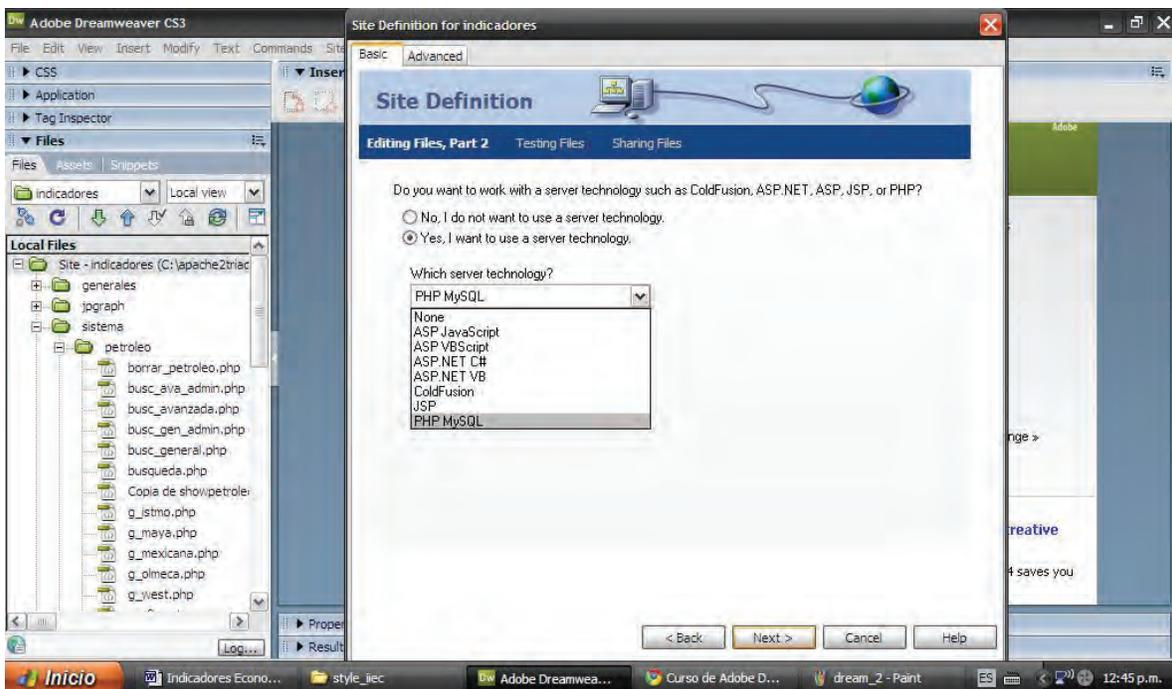
En la siguiente pantalla ingresaremos las definiciones para nuestro nuevo sitio. En el primer campo pondremos el nombre de nuestro nuevo sitio que vamos a crear. Yo recomiendo que el nombre del nuevo sitio sea el de la carpeta raíz que creamos que en este caso la llamamos indicadores.

En el siguiente campo indicaremos la *URL* o *http*, dirección desde la cual visualizaremos nuestro sistema. En este caso nuestra dirección será *http://localhost/indicadores/*

Posteriormente pulsaremos el botón Next o siguiente para continuar con nuestra configuración.



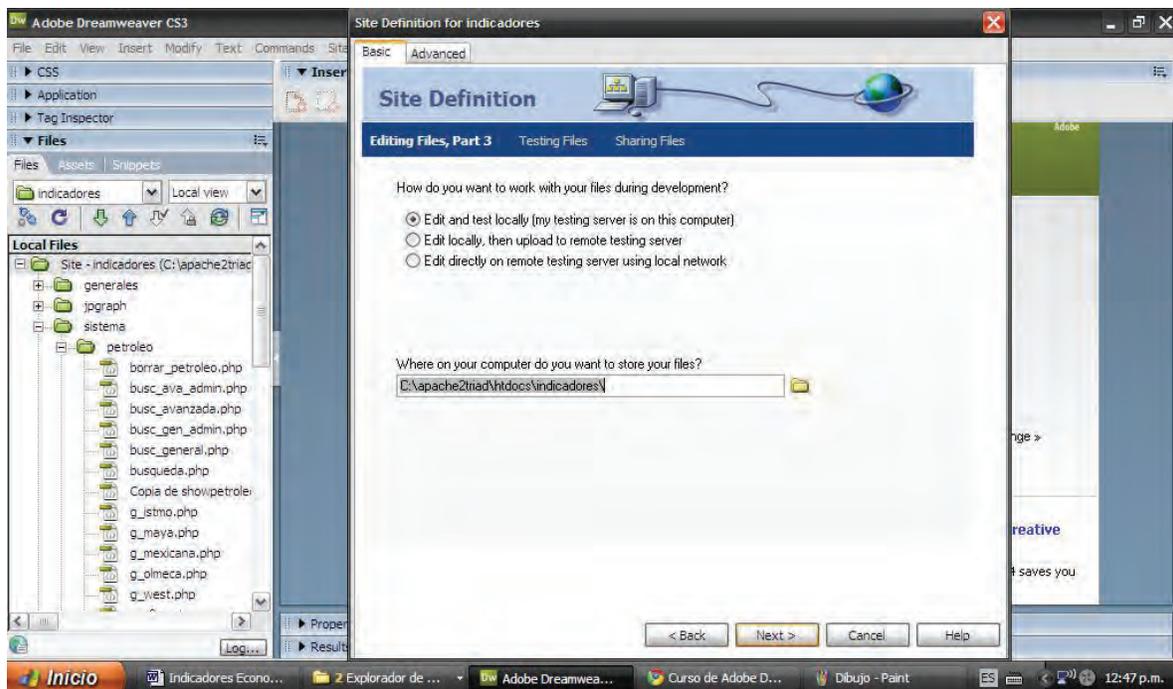
En el paso 2 para nuestra configuración, seleccionaremos la opción Si, quiero usar un servidor de tecnología. Una nueva opción emergerá dentro de la cual seleccionaremos la opción *PHP* y *MYSQL*. Que serán las tecnologías que utilizaremos para el desarrollo de nuestro sistema.



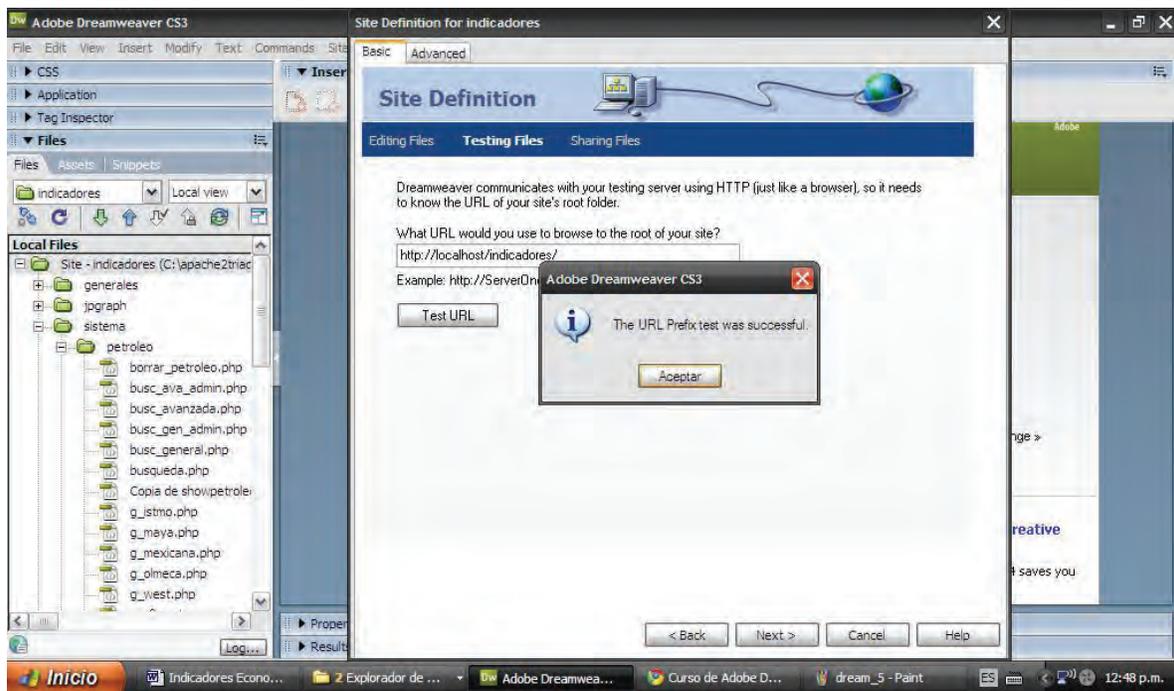
Pulsado el botón siguiente, en la parte tres de la configuración de nuestro sitio, podremos elegir la opción de cómo queremos trabajar nuestros archivos durante el desarrollo.

Para este caso seleccionaremos la opción Editar y probar localmente (El servidor de pruebas se encuentra en mi computadora).

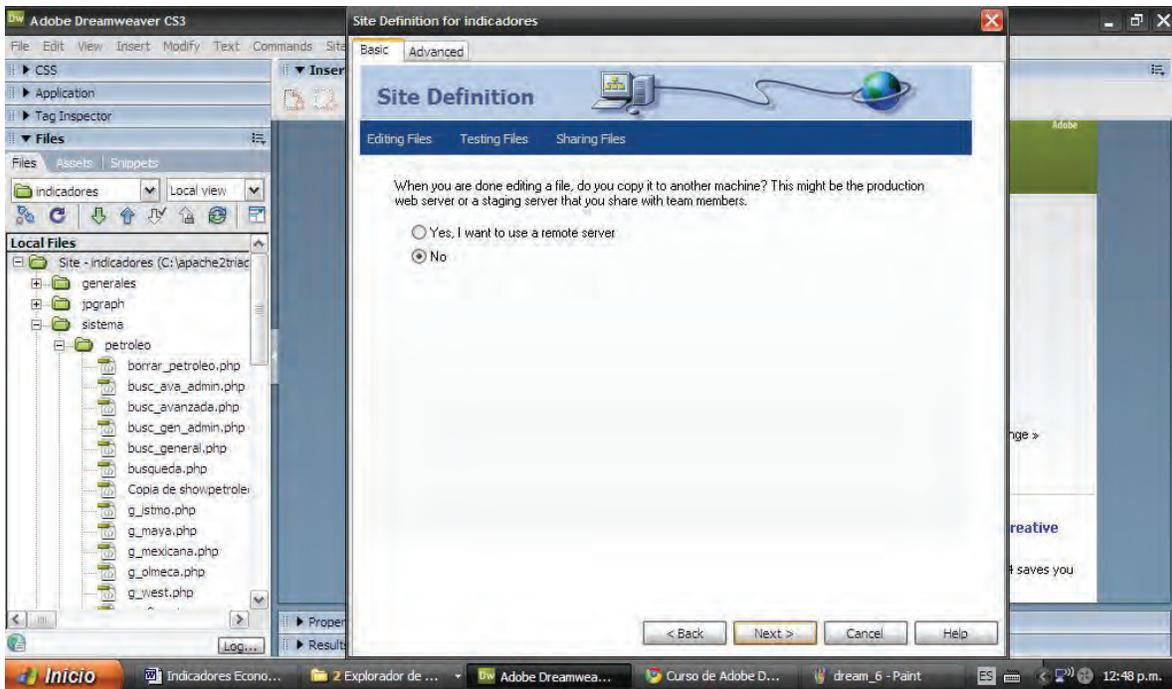
Ahora le especificaremos al programa en donde guardaremos y almacenaremos nuestros archivos mediante la ruta donde se encuentran. En este caso la ruta será `C:\apache2triad\htdocs\indicadores\`



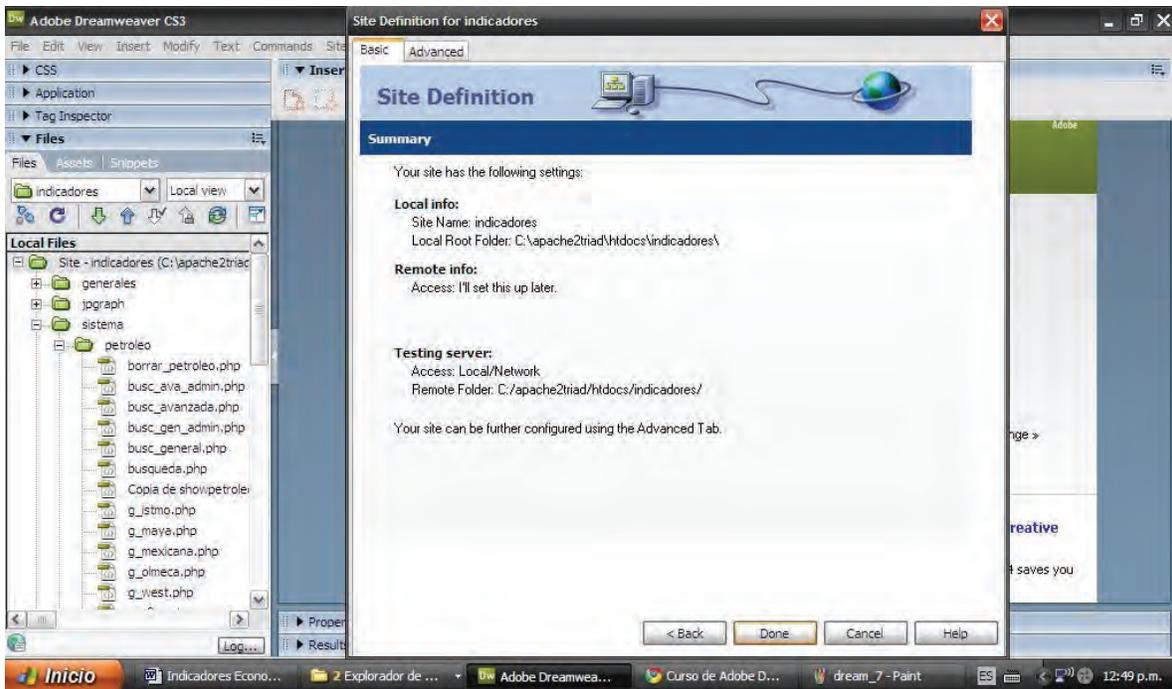
Seleccionada la opción siguiente, se ingresará a la opción de prueba de archivos. Dreamweaver se comunica con el servidor de pruebas mediante *HTTP* así que necesitara la *URL* de nuestro sitio. En este caso la *URL* será *http://localhost/indicadores/* inmediatamente haremos una prueba al *URL* pulsando el botón *TEST URL*. Si no existe algún error nos indicará que el prefijo de la *URL* es correcto.



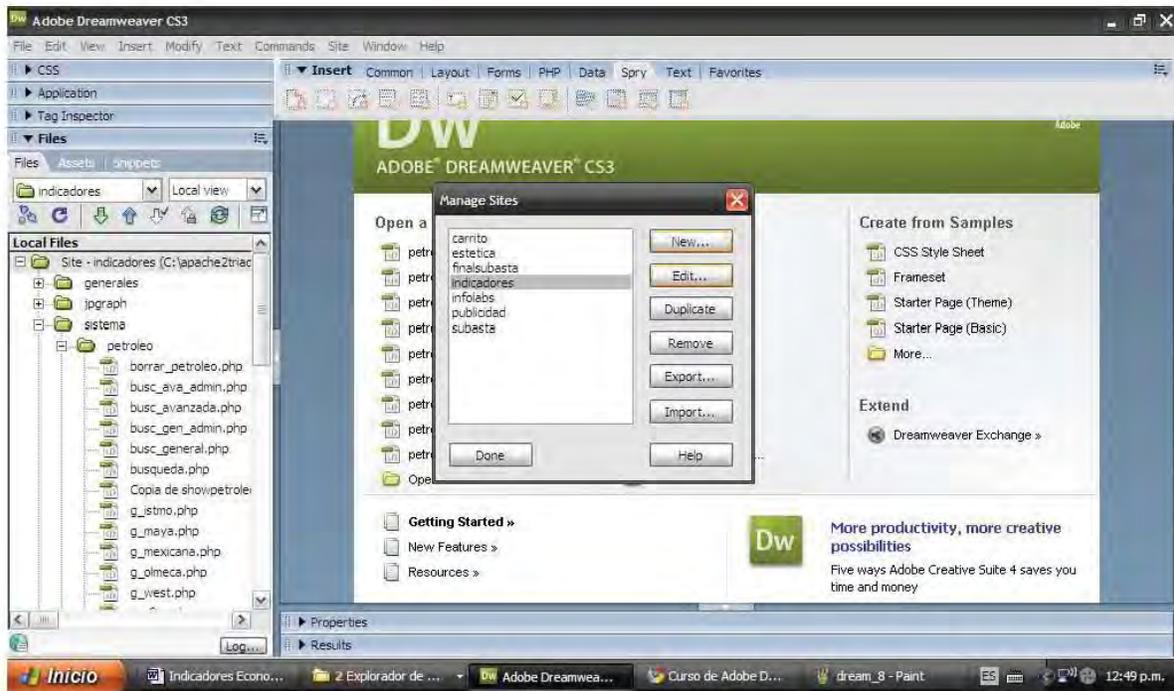
Pulsado el botón siguiente en la nueva ventana tendremos la opción de elegir si por cada vez que guardemos un documento guarde una copia en el servidor o en otra máquina. Para efectos de este sistema seleccionaremos la opción No.



Finalizado este proceso nos mostrará la información que configuramos para nuestro sitio. La información Local, la Información Remota y la información del servidor de pruebas serán mostradas. Si no existe error alguno seleccionaremos la opción Done o hecho.



Como proceso final Dreamweaver nos mostrara la primera pantalla de configuración en donde ahora será mostrado el sitio nuevo que hemos creado, una vez seleccionado Dreamweaver abrirá todos los subdirectorios y archivos que hemos guardado y almacenado.



GLOSARIO DE TÉRMINOS TÉCNICOS

Alpha-blending: Es una combinación convexa de dos colores permitiendo los efectos de transparencia en los gráficos de una computadora.

El valor de ALPHA en los rangos de código de color es de 0.0 a 1.0, donde 0.0 representa la transparencia total de un color y 1.0 la opacidad total de un color.

ANSI C: El "Lenguaje de Programación C" ANSI X3.159-1989 se conoce a menudo como ANSI C, o a veces como C89 (para distinguirla de C99).

Apache: Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.11 y la noción de sitio virtual.

Sirve para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Aplicacion LAMP: Plataforma para desarrollo y ejecución de aplicaciones web dada por el uso en conjunto de Linux, Apache, MySQL y PHP. El nombre LAMP se debe justamente a las iniciales de estas 4 tecnologías.

Atributo: Representa una propiedad de interés de una entidad. Los atributos se describen en la estructura de la base de datos empleando un modelo de datos.

Ejemplo: la entidad FCA tienen los siguientes atributos: Contaduría, Administración e Informática.

Backbone: Principales conexiones troncales de Internet. Está compuesta de un gran número de routers comerciales, gubernamentales, universitarios y otros de gran capacidad interconectados que llevan los datos a través de países, continentes y océanos del mundo.

Bucle: Sentencia que se realiza repetidas veces a un trozo aislado de código, hasta que la condición asignada a dicho bucle deje de cumplirse. Es utilizado para hacer una acción repetida sin tener que escribir varias veces el mismo código, lo que ahorra tiempo, deja el código más claro y facilita su modificación en el futuro.

Campo: Espacio de almacenamiento para un dato en particular. En las bases de datos, un campo es la mínima unidad de información a la que se puede acceder; un campo o un conjunto de ellos forman un registro, donde pueden existir campos en blanco, siendo éste un error del sistema.

Conexión Remota: Es una tecnología de conexión a través de internet, la cual permite trabajar con su computadora de forma remota.

CSS: Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

dBase: Primer Sistema de gestión de base de datos usado ampliamente para microcomputadoras, publicado por Ashton-Tate para CP/M, y más tarde para Apple II, Apple Macintosh, UNIX [1], VMS [2], e IBM PC bajo DOS donde con su legendaria versión III Plus se convirtió en uno de los títulos de software más vendidos durante un buen número de años.

Direccionamiento IP: Proceso mediante el cual Un Router envía los paquetes desde la red origen a la red destino utilizando el protocolo IP. Los paquetes deben incluir un identificador tanto para la red origen como para la red destino.

DNS: Sistema de nombre de dominio (en inglés Domain Name System, DNS) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado al internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

Eficiencia: Capacidad para lograr un fin empleando los mejores medios posibles

Eficacia: Capacidad de lograr el efecto que se desea o se espera, sin que priven para ello los recursos o los medios empleados

Entidad: Es la representación de un objeto o concepto del mundo real que se describe en una base de datos.

Etiqueta: (Tag). Marca o marcas que se dejan en un texto para que luego sean interpretadas, generalmente para realizar alguna acción sobre el mismo texto marcado. Por ejemplo, el lenguaje HTML se basa en la aplicación de múltiples etiquetas para dar formato y darle características a un texto. Este texto es interpretado y mostrado por un navegador.

Front end: Responsable de recoger entradas de los usuarios, y ser procesadas de tal manera que cumplan las especificaciones para que el back-end pueda usarlas. Front-end hace referencia al estado inicial de un proceso. Contrasta con back-end, que se refiere al estado final de un proceso. La conexión entre front-end y el back-end es un tipo de interfaz.

GNU GPL: Licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Host: Host o anfitrión es una computadora que funciona como el punto de inicio y final de las transferencias de datos. Más comunmente descrito como el lugar donde reside un sitio web. Un host de Internet tiene una dirección de Internet única (dirección IP) y un nombre de dominio único o nombre de host.

Http: (HyperText Transfer Protocol). Protocolo usado para acceder a la Web (WWW). Se encarga de procesar y dar respuestas a las peticiones para visualizar una página web. Además sirve para el envío de información adicional como el envío de formularios con mensajes, etc.

Indicador Economico: Sirve para indicar la situación de un aspecto económico particular en un momento determinado en el tiempo.

Informática: Ciencia aplicada que abarca el estudio y aplicación del tratamiento automático de la información, utilizando dispositivos electrónicos y sistemas computacionales. También está definida como el procesamiento automático de la información.

Interfaz: Parte de un programa que permite el flujo de información entre un usuario y la aplicación, o entre la aplicación y otros programas o periféricos. Esa parte de un programa está constituida por un conjunto de comandos y métodos que permiten estas intercomunicaciones.

Internet: Conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

Intérprete: Programa o sistema capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel.

ISO/OSI: Modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection) fue el modelo de red descriptivo creado por la Organización Internacional para la Estandarización lanzado en 1984. Es decir, fue un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

Java Script: Lenguaje de programación interpretado, o sea, no requiere compilación. Es utilizado especialmente en páginas web embebido en el código HTML o similares. La mayoría de los navegadores pueden interpretar los códigos JavaScript incluidos en las páginas web.

MIME: (Multipurpose Internet Mail Extensions - Extensiones de Correo Internet Multipropósito). Serie de especificaciones dirigidas al intercambio transparente de todo tipo de archivos a través de Internet.

MySQL: Sistema de gestión de base de datos relacional, multihilo y multiusuario de código abierto.

Navegador: (Del inglés, web browser) Programa que permite visualizar la información que contiene una página web (ya esté esta alojada en un servidor dentro de la World Wide Web o en uno local).

Nodo de red: Es un punto de conexión, ya sea de redistribución (como un router, un switch o un gateway) o de destino (computadoras, servidores) para la transmisión de datos, que tiene una dirección IP

NSFNET: National Science Foundation's Network. La NSFNET comenzó con una serie de redes dedicadas a la comunicación de la investigación y de la educación. Fue creada por el gobierno de los Estados Unidos (a través de la National Science Foundation), y fue reemplazo de ARPANET como backbone de Internet. Desde entonces ha sido reemplazada por las redes comerciales.

PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica

Píxel: (acrónimo del inglés picture element, "elemento de imagen") Es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

Procedimiento: Modo de ejecutar determinadas acciones que suelen realizarse de la misma forma, con una serie común de pasos claramente definidos, que permiten realizar una ocupación o trabajo correctamente.

Protocolo: Conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales. En su forma más simple, un protocolo puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación.

Red: Interconexión de computadoras para compartir información, recursos y servicios. Esta interconexión puede ser a través de un enlace físico (alambrado) o inalámbrico.

Registro: (también llamado fila o tupla) Representa un ítem único de datos implícitamente estructurados en una tabla.

Renderizar: Proceso de generar una imagen desde un modelo. Acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla.

Script: Guión o conjunto de instrucciones que permiten la automatización de tareas, creando pequeñas utilidades. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto. También un script puede considerarse una alteración o acción a una determinada plataforma.

Sentencia: Línea de código en algún lenguaje de programación. Un programa está constituido por múltiples sentencias de programación, lo que es llamado código fuente.

Servidor: Computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.

Servidor de NCSA: Servidor web desarrollado originalmente en el National Center for Supercomputing Applications por Robert McCool y una lista de colaboradores. El desarrollo del NCSA HTTPd se suspendió en 1998, pero el código sobrevivió durante un tiempo en manos del Proyecto Apache, el cual es utilizado actualmente por dos terceras partes de los servidores web de Internet. Prácticamente todo el código de NCSA se ha ido reescribiendo progresivamente en versiones de Apache.

Servidor Web: Programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

SGBD: Gestor de base de datos o (en inglés database management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

SGML: Siglas de Standard Generalized Markup Language o "Lenguaje de Marcado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos. La Organización Internacional de Estándares (ISO) normalizó este lenguaje en 1986. El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.

Sistema: Es un todo organizado y complejo; un conjunto o combinación de cosas o partes que forman un todo complejo o unitario. Es un conjunto de objetos unidos por alguna forma de interacción o interdependencia.

SQL: (Structured Query Language) lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella. Es un lenguaje de cuarta generación (4GL).

SSH: (Secure SHell, en español: Intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.

Tabla: Tipo de modelado de datos, donde se guardan los datos recogidos por un programa.

Tecnología Telemática: Disciplina científica y tecnológica que surge de la evolución y fusión de la telecomunicación y de la informática.

Telnet: Protocolo y nombre del sistema que permite conectarse a un host o servidor en donde el cliente hace de terminal virtual del servidor.

TTF: Conjunto de caracteres con un estilo o modelo gráfico particular.

Tupla: Vease registro

UNIX: Sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

URL: (sigla en inglés de Uniform Resource Locator), es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

Variable: Estructura de datos que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa. Una variable corresponde a un área reservada en la memoria principal del ordenador pudiendo ser de longitud fija o variable

W3C: El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web. Está dirigida por Tim Berners-Lee, el creador original de URL (Uniform Resource Locator, Localizador Uniforme de Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y HTML (Lenguaje de Marcado de HiperTexto) que son las principales tecnologías sobre las que se basa la Web.

WAMP: Acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

Windows, como sistema operativo; Apache, como servidor web; MySQL, como gestor de bases de datos; PHP (generalmente), Perl, o Python, como lenguajes de programación.

WWW: World Wide Web, cuya traducción podría ser Red Global Mundial, es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

XML: Siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

BIBLIOGRAFÍA

Alegsa, D. O. (s.f.). *Diccionario de informatica, internet, tecnologias y computacion on line*. Obtenido de "Alegsa.com.ar": <http://www.alegsa.com.ar/Diccionario/diccionario.php>

Apache.org. (s.f.). *Get Involved*. Obtenido de "The Apache Software Foundation": <http://apache.org/foundation/getinvolved.html>

Apeche2triadNet. (s.f.). *Information*. Obtenido de "Apache2triad Net": <http://apache2triad.net/>

Asial, C. (s.f.). *Most powerful PHP-driven charts*. Obtenido de JpGraph: <http://jppgraph.net/>

Cad.com.mx. (s.f.). *Indicadores Económicos*. Obtenido de "Economía.com.mx": http://www.economia.com.mx/indicadores_economicos.htm

DuBois, P. (2002). *MySQL CookBook*. USA: O'Really.

Factoría de Internet, S. (s.f.). *El Portal del Webmaster*. Obtenido de "WebTaller": <http://www.webtaller.com/>

Guiarte Multimedia, S. (s.f.). *Tu mejor ayuda para aprender a hacer webs*. Obtenido de "DesarrolloWeb": <http://www.desarrolloweb.com/>

Interactive, P. C. (s.f.). *Comunidad de programadores*. Obtenido de "La Web del Programador": <http://www.lawebdelprogramador.com/diccionario/>

Kendall, K., & Kendall, J. (1997). *Análisis y Diseño de Sistemas*. México: Prentice Hall.

mx.yahoo. (s.f.). *Respuestas*. Obtenido de "Yahoo México": <http://mx.answers.yahoo.com/question/index?qid=20080303145200AAh3v1k>

O'Reilly, J. (1994). *Principios de Telecomunicaciones*. México: Addison-Wesley.

Peña, O. (2007). *Dreamweaver CS3*. Madrid: Anaya Multimedia.

PHP, G. d. (2001). *Manual de PHP*.

Piattini, M. (2001). *Fundamentos y Modelos de Bases de Datos*. Madrid: Alfaomega.-Rama.

Pougue, P., & O'Really. (2007). *Dreamweaver CS3, The missing manual*.

- Pressman, R. (2002). *Ingeniería del Software*. España: McGraw-Hill.
- SitePoint. (2003). *The PHP Anthology, Volume I: Foundations*.
- Sitepoint. (2003). *The PHP Anthology, Volume II: Applications*.
- Sklar, D. (2004). *Learning PHP 5*. USA: O'Really.
- The PHP, g. (s.f.). *Hipertext Procesor*. Obtenido de "PHP.net": <http://php.net/>
- Tomas, P. (2001). *Diseños de Sitios Web*. España: McGraw-Hill.
- WebEstilo. (s.f.). *Usabilidad, programación y mucho más*. Obtenido de "WebEstilo": <http://www.webestilo.com/>
- Welling, L., & Thomson, L. (2002). *Desarrollo Web con PHP y MySQL*. Madrid: Anaya.
- Wikipedia. (s.f.). *Wikimedia*. Obtenido de "La enciclopedia Libre": <http://es.wikipedia.org/wiki/Wikipedia:Portada>
- Williams, H., & Lane, D. (2001). *Web Databases Applications with PHP & MySQL*. USA: O'Really.