



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

“PÁGINA WEB PARA UNA PYME BASADA EN AJAX”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

ANDRÉS MANUEL JAIRO



ASESOR DE TESIS
M.I. JUAN CARLOS ROA BEIZA

MÉXICO D.F. 2011



Universidad Nacional
Autónoma de México



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.



Agradezco:

A Dios por haberme permitido llegar hasta aquí y haber estado conmigo en todo momento de este trabajo; sin su ayuda jamás lo habría logrado.

A mis padres por el amor que me han dado en toda la expresión de la palabra; fueron mi motivación.

A mi hermanita por su apoyo y por darme ejemplo de alcanzar el objetivo.

A mi facultad de ingeniería, por haberme dado las herramientas para desarrollarme profesionalmente; no darme el pescado sino la caña de pescar.

A mis profesores, de los cuales aprendí no solo el conocimiento académico sino la integridad y calidad humana.

Las palabras nunca resultarán suficientes para decir "GRACIAS"

"...no digo que yo mismo ya lo haya alcanzado; lo que sí hago es olvidarme de lo que queda atrás y esforzarme por alcanzar lo que está delante..."

"No es más grande el que siempre triunfa sino el que jamás se desalienta."



ÍNDICE

CAPÍTULO 1 ENTORNO DEL PROBLEMA	1
1.1 Introducción	1
1.2 Operación del negocio de herrería	2
1.3 Logística y documentos que se generan durante el proceso de venta.	5
1.4 Software y Hardware necesarios para soportar la aplicación	8
CAPÍTULO 2 MARCO TEÓRICO	15
2.1 Comercio electrónico y aplicaciones Web	15
2.1.1 Comercio electrónico	15
2.1.2 Aplicaciones Web.....	22
2.2 Características, ventajas y desventajas de Apache, PHP, JavaScript y XML ...	34
2.2.1 Apache.....	34
2.2.2 PHP.....	44
2.2.3 JavaScript	50
2.2.4 XML.....	56
2.3 Características, ventajas y desventajas de MySQL.....	60
2.4 Características, ventajas y desventajas de AJAX.....	66
2.5 Metodologías y tecnologías similares a AJAX para el desarrollo de aplicaciones Web (RIA's).	77
2.5.1 Silverlight.....	79
2.5.2 OpenLaszlo	80
2.5.3 Mozilla Prism.....	81
2.5.4 Adobe AIR.....	82
2.5.5 JavaFX.....	83
CAPÍTULO 3 ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA.....	85
3.1 Justificación de las herramientas elegidas	85
3.1.1 Herramientas para el Front-End.....	87
3.1.2 Herramientas para el Back-End	90
3.2 Posibles Módulos de la aplicación.	101
3.3 Requisitos.....	105
3.4 Interactividad y capacidad de respuesta en aplicaciones Web.....	112



CAPITULO 4 DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN.....	115
4.1 Metodología a implementar	115
4.2 Diseño y Construcción del Back-End.....	120
4.3 Instalación y configuración del sitio Web	137
4.4 Diseño y construcción del front-end.....	143
4.4.1 Diseño del Fron-end.....	144
4.4.2 Construcción del Front-End.....	148
4.4.2.1 Tienda en línea	148
4.4.2.2 Galería de imágenes	167
4.4.2.3 Diseño y cotización en línea.....	170
4.4.2.4 Información de la PyME	173
4.4.3 Interfaz de administración	176
4.5 Pruebas e integración	184
Conclusiones.....	189
BIBLIOGRAFÍA	190



PRÓLOGO

El objetivo del presente trabajo será proporcionar una solución al proceso de venta de una pequeña empresa dedicada a la fabricación de productos hechos de herrería. Se ofrecerán las ventajas de la colocación del negocio en línea, así como la automatización de algunas cuestiones involucradas en el proceso de venta y fabricación de sus productos.

Esto mediante una página Web que más que sólo una página será una aplicación Web, la cual estará basada en una técnica de programación llamada AJAX, la cual fue elegida para mejorar la eficiencia de dicha aplicación, así también se presentará su justificación.



CAPÍTULO 1

ENTORNO DEL PROBLEMA





CAPÍTULO 1 ENTORNO DEL PROBLEMA

1.1 Introducción

EL uso de las tecnologías de la información, así como las redes de computadoras en los últimos años ha jugado un papel crucial en la integración y prosperidad de la sociedad actual y cada día su incremento es más notable.

Observamos a través de la historia como a partir de la primera aparición de la red de computadoras llamada ARPANET, creada por el MIT (*Massachusetts Institute of Technology, Instituto Tecnológico de Massachusetts*) en 1969 para un uso exclusivamente militar y de seguridad, posteriormente se separa del pentágono en 1983, ese es el momento clave a partir del cual poco a poco el uso de la red se fue extendiendo a las universidades, instituciones, empresas, particulares y prácticamente a todo el mundo logrando un crecimiento de la red de redes interconectadas entre sí que forman la Internet que actualmente conocemos y que ya es un elemento imprescindible tanto en el ámbito económico como en cualquier campo de la actuación humana.

En el ámbito comercial Internet ha ofrecido un aumento en la producción ya que las empresas pueden llegar a conocer mejor los perfiles y necesidades de sus consumidores además de reducir tiempo gastado en las transacciones, habiendo una mayor y más eficiente comunicación, logrando por ejemplo, que los clientes actuales y potenciales puedan tener información de última hora de la empresa.

Es así como Internet es una herramienta imprescindible para que la empresa tenga mayor presencia frente a sus competidores más tradicionales.

Por tanto hoy en día la tendencia en la comercialización de productos es mediante el comercio electrónico ya que ya no existe la necesidad de tener que atraer a los clientes hasta los locales de venta y de esta manera llegar a una mayor cantidad de clientes,



permite también un ahorro de la mayor parte de los costos fijos, así como la renta de la tienda o el pago de la nómina de empleados; además de que una tienda en Internet puede recibir pedidos las 24 horas del día los 7 días a la semana. Aunado a esto los potenciales compradores pueden escoger los productos en la tranquilidad de sus hogares, sin la asistencia o presión de un vendedor.

Con la aparición de las tarjetas de crédito la venta por catálogo en línea tomó un mayor impulso, los catálogos en línea son uno de los mayores productos de las páginas Web dinámicas, ya que en una página estática puede haber contenido flash o un buen diseño pero eso no garantizará que exista un flujo continuo de visitantes o clientes satisfechos que pueden dejar de visitar el sitio por ausencia de actualizaciones o contenidos personalizados, ésta es una de las mayores ventajas de las páginas Web dinámicas, que interactúan con el usuario, recibiendo la información que realmente busca y necesita.

1.2 Operación del negocio de herrería

De acuerdo al título de la presente Tesis se hará una optimización en línea de una PYME dedicada a la herrería en general, los productos que ofrece la pequeña empresa son el resultado de un proceso artesanal con reducida maquinaria automatizada industrialmente ya que la elaboración del producto final se realiza básicamente a mano con técnicas un tanto rudimentarias, posee una administración en su mayor parte empírica. Su posición esta en constante crecimiento.

De esta manera el negocio sobre el cual se realizará el trabajo esta clasificado como una Pequeña y Mediana Empresa (PyME), este tipo de clasificación está determinada por la legislación de cada país, en México de acuerdo a la Secretaría de Hacienda la categoriza básicamente como una empresa que reúne todos los aspectos de una empresa tradicional pero obviamente con características distintivas como son el número de trabajadores, para nuestro país corresponde de 11 a 30 trabajadores para el sector comercio.



Para poder tener un panorama general de la manera como está actualmente operando el negocio, se expondrá el funcionamiento de cada área y lineamiento que están actualmente involucrados en el proceso de producción de la pequeña empresa.

Se cuenta con los siguientes mecanismos para la producción y movimiento del negocio, los cuales clasificamos en cuatro principales áreas:

✦ **Administración y operaciones.** Esta área se ocupa de las cuestiones relacionadas con la empresa, Desde la contratación del personal hasta la compra de la materia prima, el pago del personal, firma de los cheques, horarios del personal, pago de proveedores, control de inventarios tanto de los materiales como de los productos finales, limpieza del local.

✦ **Producción.**

✦ *Diseño.* Es el área que se encarga de la parte artística, se encuentra el personal encargado de realizar los bocetos para los nuevos productos o para el asesoramiento de diseño para el cliente cuyas necesidades no satisfacen los artículos en existencia. Después de la elección del cliente se procede a realizar una cotización con base en las dimensiones, material y complejidad de elaboración del producto. Se establecen las formas de pago considerando los costos de envío e instalación y finalmente el tiempo de entrega del producto terminado.

✦ *Construcción.* Después de haber obtenido un diseño para cierto producto y después de haber completado el proceso de compra se procede a la manufactura del producto. Aquí se complementa con las áreas de acabado, empaque, transporte y colocación.

✦ **Contabilidad y finanzas.** Es el área encargada del registro de todos los movimientos de ingresos monetarios a través del tiempo, además de llevar a



cabo las declaraciones e impuestos de acuerdo a los resultados de los libros de contabilidad que lleva la empresa como emisión de facturas, proyección del negocio por ventas y costos asociados al desarrollo de todo el negocio.

✦ **Mercadeo y ventas.** Este departamento es el que hace el primer contacto con el cliente, poniéndolo en contacto con el personal de la empresa del área administrativa, solicitando una cotización de algún producto de su necesidad, si el producto solicitado se encuentra en existencia y satisface las necesidades del comprador se procede a presentar una cotización del producto indicándole sus características.

Aquí también se involucra el personal responsable de las estrategias de mercado del negocio, tales como la publicidad en forma impresa y en medios electrónicos como Internet. También incluye el personal que atiende en el mostrador a las personas que visitan el local para solicitar un presupuesto de algún producto ofreciéndoles los diferentes catálogos y cotizaciones de los productos ya fabricados o los que puedan ser fabricados de acuerdo a las necesidades particulares del cliente y posteriormente canalizándoles al área de producción.

Es así como de forma general se ha explicado las áreas operativas de la PYME. Debido al tamaño de la misma, algunas áreas se desarrollan por un mismo personal (ver figura 1.2.1).

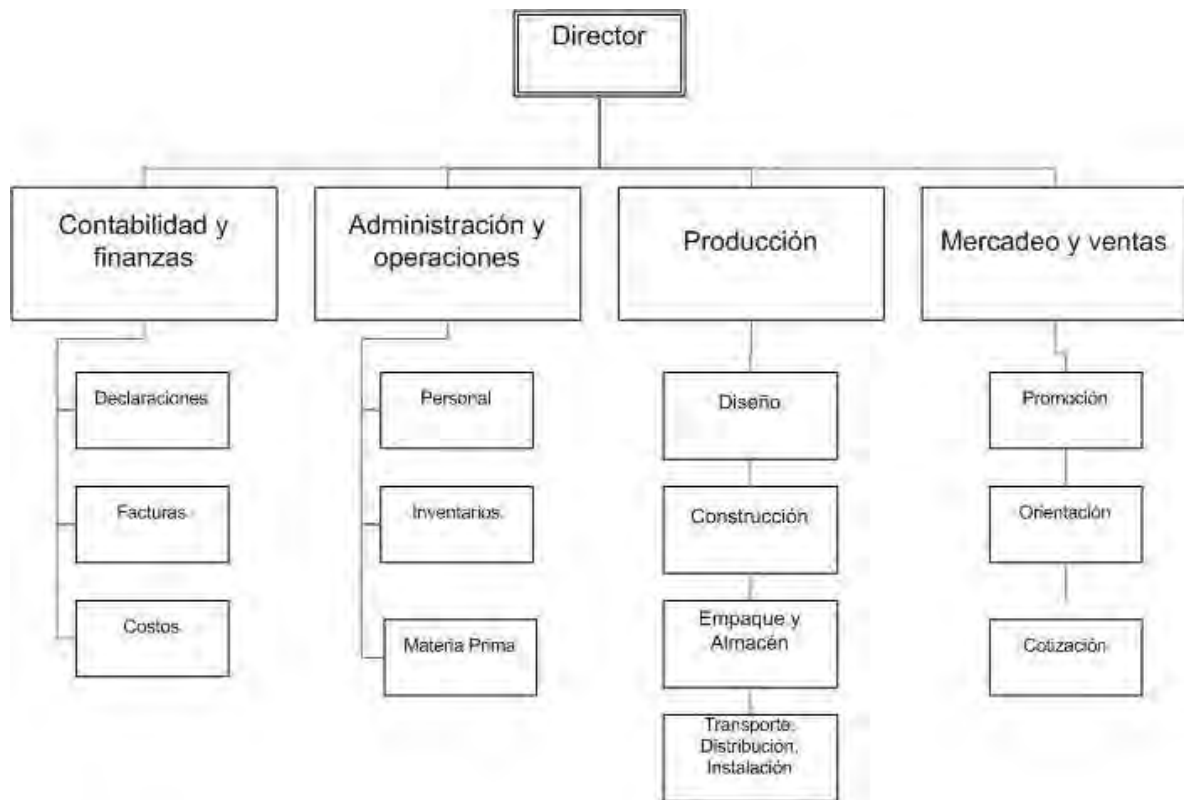


Figura 1.2.1 Operación del negocio de herrería

1.3 Logística y documentos que se generan durante el proceso de venta.

Después de establecer los departamentos involucrados en el proceso de venta a continuación se muestra la logística de operación de la pequeña empresa. (Véase Fig. 1.3.1)

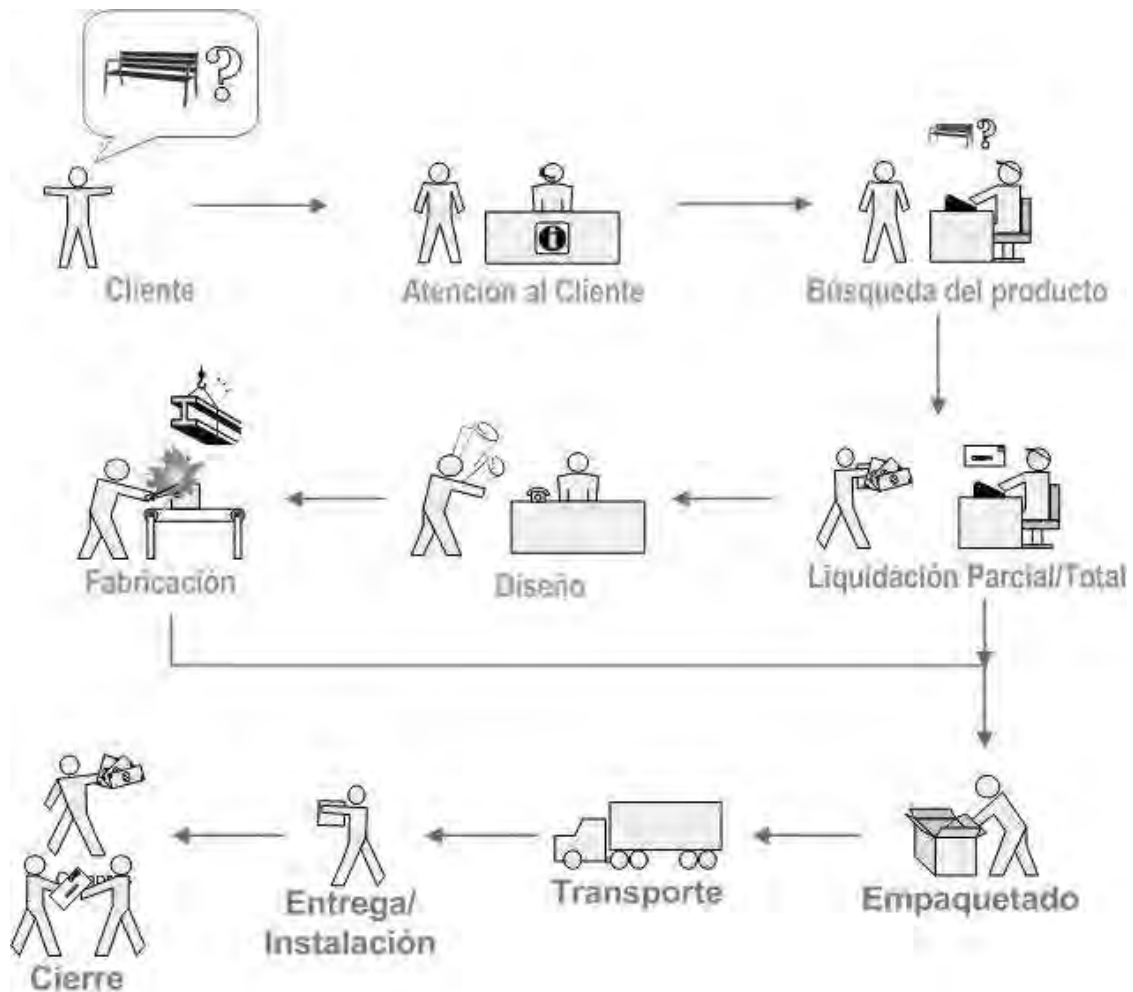


Fig. 1.3.1 Logística de operación de la empresa

A continuación se detalla el proceso:

- Se parte de la necesidad de un cliente que requiere un producto, el cliente visita el local de la empresa para requerir asesoría, solicitando algún producto o la fabricación e instalación, según sea el caso, de algún producto de acuerdo a sus necesidades.
- El personal que atiende al cliente muestra el catálogo impreso de los productos en existencia o los diseños de los productos para los productos solicitados.
- El cliente hace una elección de algún producto y pide la cotización por el producto elegido u otro por fabricar.



-
- El personal que atiende al cliente realiza una búsqueda en el catálogo donde se encuentra el registro de los productos de almacén para verificar la existencia.
 - Si hay en existencia el producto, el personal le muestra las características del mismo, así como costo, formas de pago, plazo de envío, costo de transporte e instalación según el caso. Cuando el producto existe y satisface las necesidades del cliente, el pago es cubierto en su totalidad, de otra manera, el pago se realiza por un anticipo del 50% y se completa en el momento de la entrega e instalación. Aquí se genera una copia del anticipo para el cliente.
 - De lo contrario, si el producto buscado o el diseño no se encuentran en el catálogo, el personal lo canaliza al departamento de diseño para una orientación y una posterior fabricación del producto, así como su correspondiente cotización.
 - Después del acuerdo del producto o diseño sobre pedido, se genera una orden de pago en caja y una orden de fabricación o empaque del producto de la bodega, para el departamento de manufactura.
 - Después de generarse la orden de fabricación, el personal de dicho departamento atiende las siguientes áreas:
 - Almacenamiento de materiales: La fabricación de productos se inicia con la recepción de los insumos en la empresa. Se clasifica y ordena el material por tipo.
 - Transporte al área de proceso. Los materiales seleccionados se transportan al área de producción.
 - Corte de piezas. Se lleva a cabo mediante patrones de acuerdo a la medida que se requiera, según el modelo que pueda ser diseñado por alguien ajeno a la empresa.
 - Unión de piezas. Se unen las piezas con soldadura o tornillos en el caso de trabajos de aluminio.
 - Procesos de barnizado, acabados, etc.
 - Empaque
 - Almacenamiento del producto terminado
 - Clasificación
-



- Después de la fabricación se realiza la entrega y puesta a punto del producto, la transacción finaliza cuando se cubre el costo restante y se firma de conformidad por parte del cliente.

Finalmente se genera la factura que se muestra en la figura 1.3.2.

HERRERIA "GARCIA"
JOSE GPE. GARCIA GALLARDO
 Calz. de Gpe. No. 17, Col. San Rafael Chamapa
 Naucalpan, Edo. de México. CP. 53660
 R.F.C. GAGG-411012-EF3
 R.F.C. HECR-600705-T21

FECHA			NOTA DE VENTA	
DIA 15	MES Dici	AÑO 2010	Nº 249	
NOMBRE <u>Enliso Andres Quinones</u>				
DOMICILIO <u>8 de febrero</u>				
CANT.	CONCEPTO	P. UNITARIO	IMPORTE	
1	VENTANA pc 280 x 135 Tubulan. Rose 1 Hoja Concediza			
<i>[Handwritten signature]</i>				
	Precio	Nº	3000=	
	Ido	Nº	1000=	
	RCS1R	Nº	1000=	
			TOTAL	

CONTRIBUYENTE DEL REGIMEN SIMPLIFICADO

Figura 1.3.2 Ejemplo de factura generada al final del proceso de venta

1.4 Software y Hardware necesarios para soportar la aplicación

Después de analizar las áreas involucradas en el proceso de venta de las cuales se pretende optimizar, se comenzará por revisar los requisitos técnicos tanto de software como hardware para que la aplicación tenga un mejor desempeño.



Para los fines del sitio Web que se construirá, se basará en una técnica de programación Web que ha liderado la Web 2.0 llamada AJAX (***Asynchronous Javascript And XML, JavaScript asíncrono y XML***), que requerirá de ciertas capacidades tanto de software y hardware, que se tratarán más ampliamente en el siguiente capítulo.

Herramientas para el desarrollo de la aplicación:

Dreamweaver CS3. Permite al programador diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web, el propio programa va guiando al usuario en su utilización.

Características:

- ✦ Entorno de desarrollo compatible con PHP, J2EE y Microsoft.NET.
- ✦ Diseño de páginas Web con interfaz gráfica.
- ✦ Barra de herramientas de programación, que facilita las operaciones de programación más usuales
- ✦ Contracción del código, para centrarse sólo en el que se está utilizando.
Barra de herramientas de reproducción de estilos, con la que se puede ver el resultado final.

De acuerdo a la página del fabricante es indispensable contar con los siguientes requisitos del sistema:

Para Windows

- ✦ Procesador Intel® Pentium® 4, Intel Centrino®, Intel Xeon® o Intel Core™ Duo (o compatible)
- ✦ Microsoft® Windows® XP con Service Pack 2 ó Windows Vista™ Home Premium, Business, Ultimate o Enterprise (certificada para ediciones de 32 bits)
- ✦ 512 MB de RAM



- ✦ 40 MB de espacio disponible en el disco duro (se necesita espacio libre adicional durante la instalación)
- ✦ Resolución de pantalla de 1.024 x 768
- ✦ Unidad de DVD-ROM
- ✦ Software QuickTime 7 necesario para funciones multimedia
- ✦ Es necesaria una conexión a Internet o telefónica para activar el producto

Para Macintosh

- ✦ Procesador PowerPC® G4, G5 ó Intel® multinúcleo
- ✦ Mac OS X v10.4.8-10.5 (Leopard)
- ✦ 512 MB de RAM
- ✦ 40 MB de espacio disponible en el disco duro (se necesita espacio libre adicional durante la instalación)
- ✦ Resolución de pantalla de 1.024 x 768
- ✦ Unidad de DVD-ROM
- ✦ Software QuickTime 7 necesario para funciones multimedia
- ✦ Es necesaria una conexión a Internet o telefónica para activar el producto

Extensión Developer Toolbox

Además de *Dreamweaver* se requerirá de una extensión llamada **Developer Toolbox** (**Caja de herramientas de desarrollo**) diseñada para DW (*Dreamweaver*) que permitirá desarrollar la aplicación en menor tiempo. Dicha extensión se tratará en mayor detalle en el capítulo 4.

De igual manera será necesario contar con un mínimo de requisitos para el uso de la extensión, estos también son señalados por el fabricante:

- ✦ Adobe Dreamweaver CS3.
- ✦ Adobe Extension Manager CS3

Servidores compatibles.

- ✦ ASP - Microsoft IIS 5.0 ó superior.



- ✦ PHP - Apache 1.3.x or 2.x / XAMPP.
- ✦ CF - Built in ColdFusion 6 ó 7.

Base de datos compatibles

- ✦ MySQL 4.0 ó superior
- ✦ Microsoft MSSQL Server 2003
- ✦ Microsoft Access 2000 y 2003

Navegadores compatibles

- ✦ Microsoft Internet Explorer 6.0 ó superior
- ✦ Mozilla Firefox 1.5 ó superior
- ✦ Safari 2.0 ó superior

También será necesario contar con una herramienta que permitirá la facilidad de uso del servidor y los lenguajes involucrados, llamada XAMPP; que es de uso gratuito.

XAMPP funciona con los sistemas operativos:

- ✦ Linux
- ✦ Windows
- ✦ Mac OS X
- ✦ Solaris

Además de proporcionar Apache, PHP y MySQL, ofrece otras utilidades de interés, tales como Webalizer, Filezilla FTP Server, Mercury Mail Transport o Zend Optimizer, siendo éste el motivo por el que se ha decantado por su uso frente al paquete Wampp.

Se descargará la versión 1.7.3 para Windows que contiene las siguientes herramientas: Apache 2.2.14 (IPV6 enabled), MySQL 5.1.41 (Community Server) con PBXT engine 1.0.09-rc, PHP 5.3.1 (PEAR), Miniperl 5.10.1, XAMPP Control Version 2.5.8, ApacheFriends Edition, XAMPP CLI Bundle 1.6, XAMPP Port Check 1.5, XAMPP Security 1.1, SQLite 2.8.17, SQLite 3.6.20, OpenSSL 0.9.8l, phpMyAdmin 3.2.4, msmtmp 1.4.19 y Webalizer 2.21-02.



Requisitos de sistema para XAMPP:

- ✦ 128 MB RAM
- ✦ 192 MB libres de espacio en Disco duro.
- ✦ Windows 2000, XP (Server 2003), Vista (Server 2008), 7
- ✦ Todos los sistemas de 32 bit (64 también debería trabajar)

Finalmente para que la aplicación pueda funcionar adecuadamente al momento de su implementación, es necesario contar con un mínimo de requerimientos basados en los datos del fabricante de las herramientas elegidas, en los siguientes aspectos:

Equipo de cómputo

■ En el cliente

● Hardware:

- Memoria RAM: 128 MB RAM (Recomendado: 256 MB RAM o más).
- Disco Duro (HD): 100 MB de espacio en disco.
- Procesador: Pentium 233 MHz (Recomendado: Pentium 500MHz o superior).

● Software:

- Sistema Operativo: Windows, Linux, Mac OS
- Navegadores: Son los programas que permitirán usar el servicio HTTP (protocolo de transferencia de hipertextos), es decir, poder visualizar las páginas Web.

Navegadores Web que permiten AJAX: Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon,



IceWeasel, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores, Google Chrome, Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él. Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores, y el Web Browser para S60 de Nokia tercera generación y posteriores, Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

Navegadores que no permiten AJAX: Opera 7 y anteriores, Microsoft Internet Explorer para Windows versión 4.0 y anteriores, Safari 1.2 y anteriores, Dillo, Navegadores basados en texto como Lynx y Links, Navegadores para personas con capacidades especiales visuales (Braille), Algunos navegadores de teléfonos móviles, Navegador de la PSP.

- Lenguajes de Cliente: JavaScript.

■ En el Servidor:

● Hardware:

- Memoria RAM: 128 MB RAM (Recomendado: 256 MB RAM o más).
- Disco Duro (HD): 100 MB de espacio en disco.
- Procesador: Pentium 233 MHz (Recomendado: Pentium 500MHz o superior).

● Software:

- Sistema Operativo: Windows, Linux, Mac OS.
- Navegadores Web. (Las mismas consideraciones que para el cliente)



- Lenguajes de Servidor como son PHP, ASP, JSP, etc).
- Servidor Web: Apache o en la versión comercial de Windows llamada IIS.

Conexión a Internet

Aunque las pruebas se harán en un servidor local en el que no sea imprescindible una conexión a Internet, cuando el proyecto ya esté funcionando en línea es necesario contar obviamente con una conexión a Internet a través de un proveedor de servicio (ISP o Internet Service Provider). Es necesario que el cliente Web esté conectado a un módem con suficiente ancho de banda, mínimo 56kb.

Unidades

Unidad de CD-ROM (si la instalación del software se hace desde un CD-ROM).

Pantalla

Monitor Super VGA (800 x 600) o de mayor resolución con 256 colores.

Periféricos

Mouse, teclado.



CAPÍTULO 2

Marco teórico





CAPÍTULO 2 MARCO TEÓRICO

2.1 Comercio electrónico y aplicaciones Web

2.1.1 Comercio electrónico

Como se mencionaba en la introducción de la presente tesis, gracias al desarrollo de las nuevas Tecnologías de la Información y Comunicación (TICs), las distancias y el tiempo entre las transacciones de proveedores y consumidores, así como otros factores involucrados en el comercio, han ido mejoradas, siendo esta actividad una de las más importantes en el tráfico de Internet. De esta manera la Web ofrece un nuevo mercado que define lo que ahora es llamado como la "economía digital".

De acuerdo a la OECD (*Organisation for Economic Co-Operation and Development, Organización para la Cooperación y el Desarrollo Económicos*) el Comercio Electrónico es definido como: el proceso de compra, venta o intercambio de bienes, servicios e información a través de las redes de comunicación.

La anterior definición aparentemente es llana sin embargo define lo realmente necesario para comprender el comercio electrónico, es decir, el comercio electrónico no es una nueva tecnología sino una metodología moderna para hacer negocios en la que se suple esa necesidad de las empresas, comerciantes y consumidores por reducir costos, mejorar la calidad de los bienes y servicios, además de mejorar el tiempo de entrega de los bienes o servicios con el uso de la tecnología. Implica además cualquier transacción comercial en la cual las partes involucradas interactúan de manera electrónica en lugar de hacerlo de la manera tradicional con intercambios físicos o trato físico directo.

Es así como el comercio electrónico es el medio para llevar a cabo los cambios tanto en organización como en operaciones en los negocios, dentro de una escala global, logrando así que las compañías sean más eficientes y flexibles en sus operaciones



internas, para así trabajar de una manera más cercana con sus proveedores y estar más pendiente de las necesidades y expectativas de sus clientes.

En la actualidad las empresas ya están comenzando a usar Internet como un nuevo canal de ventas, sustituyendo las visitas personales, correo y teléfono por pedidos electrónicos, ya que gestionar un pedido por Internet cuesta 5% menos que hacerlo por vías tradicionales. Es así como surge entonces el comercio electrónico, como una alternativa de reducción de costos y una herramienta fundamental en el desempeño empresarial.

Asimismo surgen también algunos nuevos inconvenientes relacionados con el comercio electrónico que van desde la validez legal de las transacciones y contratos sin papel, la necesidad de acuerdos internacionales que armonicen las legislaciones sobre comercio, el control de las transacciones internacionales, incluido el cobro de impuestos; la protección de los derechos de propiedad intelectual, la protección de los consumidores en cuanto a publicidad engañosa o no deseada, fraude, contenidos ilegales y uso abusivo de datos personales, hasta otros provocados por la dificultad de encontrar información en Internet, comparar ofertas y evaluar la fiabilidad del vendedor y del comprador en una relación electrónica, la falta de seguridad de las transacciones y medios de pago electrónicos, la falta de estándares consolidados, la proliferación de aplicaciones y protocolos de comercio electrónico incompatibles y la congestión de Internet.

En México se necesita de una normativa jurídica en relación al comercio electrónico y el formato digital. En un contexto mundial, nuestro país se encuentra desactualizado, implicando esto una toma de medidas apremiantes respecto a esto ya que en el país a pesar de esto el comercio electrónico se ha incrementado en forma considerable.

De acuerdo a los estudios hechos por la AMIPCI (***Asociación Mexicana de la Industria Publicitaria y Comercial en Internet***) mejor conocida como Asociación mexicana de Internet, tan solo para el año 2009 las ventas por Internet representaron el



16% del total de ventas (Figura 2.1.1.1). Estos estudios fueron basados en las empresas que operan en México de gran diversidad de giros que van desde las ventas de boletos de avión, artículos de cómputo, espectáculos, cine, accesorios para celulares, electrónica, hospedajes, viajes, etc., y cuyos movimientos mercantiles se encuentran tanto en el sector presencia física como on-line.

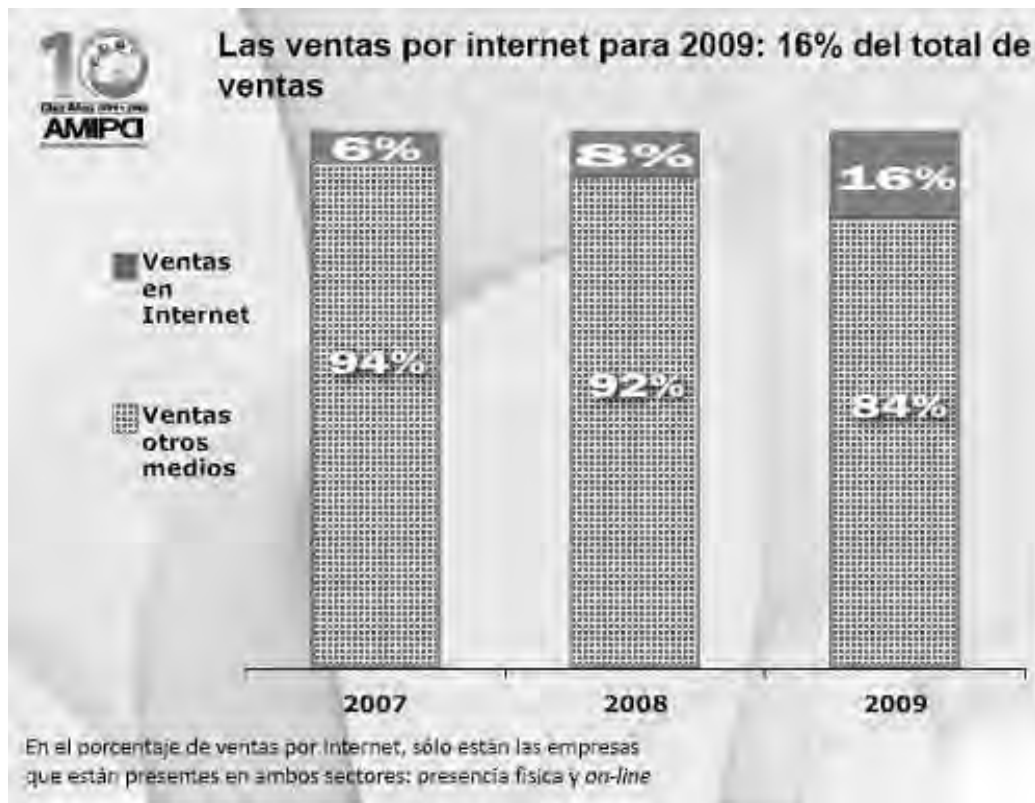


Figura 2.1.1.1 Porcentaje de ventas por Internet en 2009

En México el comercio electrónico se enfoca en primer lugar a los portales de cómputo (44%), en segundo lugar las aerolíneas (37%) y en tercer lugar los portales de hogar y electrodomésticos (33%).

De esta manera el crecimiento esperado del comercio online se ha superado (ver figura Figura 2.1.1.2)



Figura 2.1.1.2 Se superan expectativas de crecimiento

Como también se menciona en la introducción de la Tesis con el surgimiento de la tarjeta de crédito la venta por catálogo en línea tomó un mayor impulso y hoy en día según los estudios de la AMIPCI la tarjeta de crédito (Fig. 2.1.1.3) continúa siendo el medio de pago preferido.



Figura 2.1.1.3 La tarjeta de crédito sigue siendo el medio de pago preferido.

(www.amipci.org.mx)



Con el incremento del ancho de banda y el uso de los dispositivos móviles se ha impulsado el comercio electrónico. Con la combinación del crecimiento en el acceso de banda ancha y el uso de dispositivos móviles cada vez más inteligentes como los teléfonos 3G, se espera que las ventas en el comercio electrónico aumenten significativamente pronto.

Finalmente al mismo tiempo surgen también los derechos de los consumidores que cada vez demandan más confianza por parte de los sitios Web y que al momento de efectuar una compra solicitan los que posean algún sello de confianza por alguna entidad reconocida, los consumidores se inclinan por aquellos sitios recomendados o que gocen de buena reputación.

A continuación se muestra el ciclo del comercio electrónico.

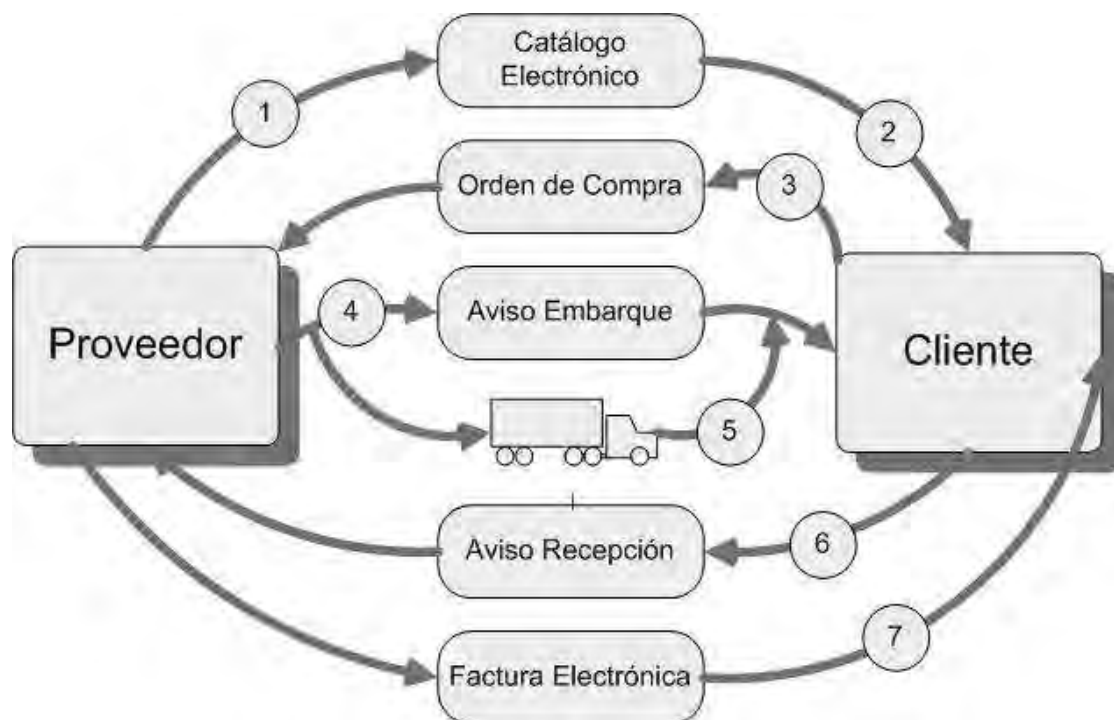


Figura 2.1.1.4 Ciclo del comercio electrónico



Considerando la figura 2.1.1.4, se muestra como primera instancia del ciclo del comercio electrónico la información que ofrece el proveedor al cliente mediante un sistema de información bien estructurado como un catálogo electrónico que ofrezca ayuda para la decisión de compra “remota”, después de haber tomado la decisión el cliente efectúa su orden de compra así como el establecimiento de las cláusulas de compra como envío, empaquetado, donde y como pagar, así como las tecnologías inherentes. Posteriormente el proveedor envía el producto y a la recepción de este por el cliente se notifica al proveedor con un aviso de recepción y con la factura electrónica es como la venta es finalizada.

El éxito del negocio en esta modalidad sigue estando en conocer el mercado y los clientes, varias prácticas y esquemas de personalización basados en el comercio inteligente han evolucionado para ayudar a las compañías de comercio electrónico para mejorar las relaciones con sus clientes y poder suplir sus necesidades de manera óptima.

Ventajas

Para los clientes:

- Permite el acceso a más información. La naturaleza interactiva del Web y su entorno hipertexto permiten búsquedas profundas no lineales que son iniciadas y controladas por los clientes, por lo tanto las actividades de mercadeo mediante el Web están más impulsadas por los clientes que aquellas proporcionadas por los medios tradicionales.
- Facilita la investigación y comparación de mercados. La capacidad del Web para acumular, analizar y controlar grandes cantidades de datos especializados permite la compra por comparación y acelera el proceso de encontrar los artículos.
- Abarata los costos y precios. Conforme aumenta la capacidad de los proveedores para competir en un mercado electrónico abierto se produce una



baja en los costos y precios, de hecho tal incremento en la competencia mejora la calidad y variedad de los productos y servicios.

Para las empresas:

- se puede disminuir el tiempo que se tardan en realizar las transacciones comerciales, incrementando la eficiencia de las empresas.

- Comunicaciones de mercadeo. Actualmente, la mayoría de las empresas utiliza el Web para informar a los clientes sobre la compañía, a parte de sus productos o servicios, tanto mediante comunicaciones internas como con otras empresas y clientes. Esta capacidad ofrece oportunidades sin precedentes para ajustar con precisión las comunicaciones a los clientes individuales, facilitando que éstos soliciten tanta información como deseen.

- Beneficios operacionales. El uso empresarial del Web reduce errores, tiempo y sobrecostos en el tratamiento de la información.

Desventajas

- Entorno empresarial y tecnológico cambiante. Empresas y clientes desean tener flexibilidad para cambiar, según su voluntad, de socios comerciales, plataformas y redes. No es posible evaluar el costo de esto, pues depende del nivel tecnológico de cada empresa, así como del grado deseado de participación en el comercio electrónico. Como mínimo una empresa necesitará una computadora personal con sistema operativo Windows o Mac, un módem, una suscripción a un proveedor de servicios de Internet, una línea telefónica. Una compañía que desee involucrarse más, deberá prepararse para introducir el comercio electrónico en sus sistemas de compras, financieros y contables, lo cual implicará el uso de un sistema para el intercambio electrónico de datos (EDI) con sus proveedores y/o una intranet con sus diversas sedes.



- Privacidad y seguridad. La mayoría de los usuarios no confía en el Web como canal de pago. En la actualidad, las compras se realizan utilizando el número de la tarjeta de crédito, pero aún no es seguro introducirlo en Internet sin conocimiento alguno. Cualquiera que transfiera datos de una tarjeta de crédito mediante el Web, no puede estar seguro de la identidad del vendedor. Análogamente, éste no lo está sobre la del comprador. Quien paga no puede asegurarse de que su número de tarjeta de crédito no sea recogido y sea utilizado para algún propósito malicioso; por otra parte, el vendedor no puede asegurar que el dueño de la tarjeta de crédito rechace la adquisición. Resulta irónico que ya existan y funcionen correctamente los sistemas de pago electrónico para las grandes operaciones comerciales, mientras que los problemas se centren en las operaciones pequeñas, que son mucho más frecuentes.

- Cuestiones legales, políticas y sociales. Existen algunos aspectos abiertos en torno al comercio electrónico: validez de la firma electrónica, no repudio, legalidad de un contrato electrónico, violaciones de marcas y derechos de autor, pérdida de derechos sobre las marcas, pérdida de derechos sobre secretos comerciales y responsabilidades. Por otra parte, deben considerarse las leyes, políticas económicas y censura gubernamentales.

2.1.2 Aplicaciones Web

Con el surgimiento de las aplicaciones Web el crecimiento de comercio electrónico ha superado las expectativas.

Después de revisar algunos conceptos implicados en el comercio electrónico es momento de introducir los conceptos que serán requeridos para poder desarrollar la aplicación que satisfaga los requerimientos.



Por tanto se tratará en esta sección los conceptos importantes de las aplicaciones Web, pero antes de definirlos será necesario tener una comprensión básica de las redes y de World Wide Web.

Transacciones http

HTTP define un conjunto de métodos y encabezados que permiten a los clientes y servidores interactuar e intercambiar información de una manera uniforme y confiable.

En su forma más simple, una página Web es un documento XHTML o HTML, es decir, un archivo de texto simple que contiene marcado, es decir, etiquetas, para describir a un navegador Web cómo mostrar y dar formato a la información del documento. Los documentos XHTML o HTML también pueden contener datos de hipertexto (lo que se conoce como hipervínculos) que vinculan a distintas páginas, o a otras partes de la misma página.

Cuando el usuario activa un hipervínculo, la página Web solicitada se carga en el navegador Web del usuario.

HTTP utiliza URI's que al especificar las ubicaciones de los documentos se llaman URL's (Localizadores uniformes de recursos).

Los URL's comunes se refieren a archivos, directorios u objetos que realizan tareas complejas, como búsquedas en bases de datos y en Internet.

Un URL contiene la información que dirige a un navegador al recurso que el usuario desea utilizar. Cuando el navegador Web recibe un URL, realiza una transacción HTTP simple para obtener y mostrar la página Web que se encuentra en esa dirección.

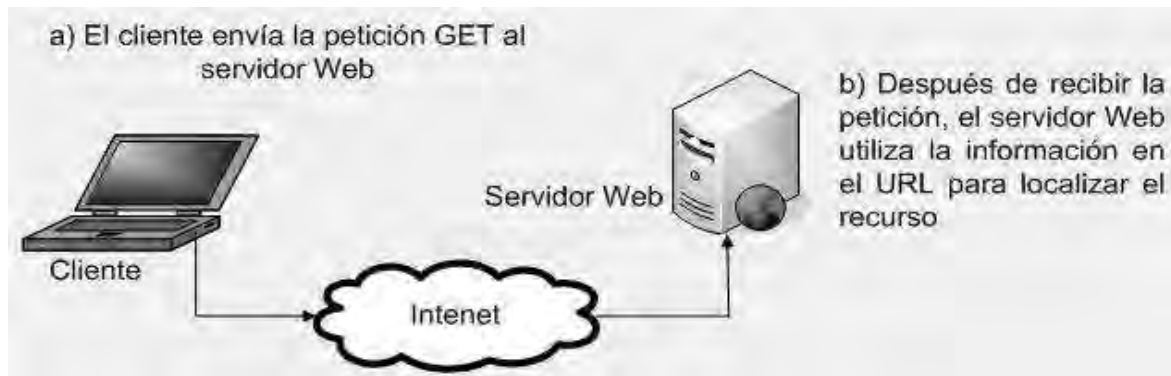


Figura 2.1.2.1 Interacción entre el navegador Web y servidor Web

El navegador Web envía una petición HTTP al servidor de la forma: *GET /ruta/recurso.html HTTP/1.1*, donde GET es un método HTTP, el cual indica que el cliente desea obtener un recurso del servidor. El resto de la petición proporciona el nombre de la ruta del recurso, junto con el nombre del protocolo y el número de versión (HTTP/1.1). Ver figura 2.1.2.1. Cualquier servidor que entienda HTTP (versión 1.1) puede traducir esta petición y responder en forma apropiada.

En la figura 2.1.2.2 se muestran los resultados de una petición exitosa. El servidor responde enviando una línea de texto que indica la versión de HTTP, seguida de un código numérico y una frase que describe el estado de la transacción. Estos códigos de estado HTTP están especificados por el RFC 2616 (ver Tabla 2.1.2.1)

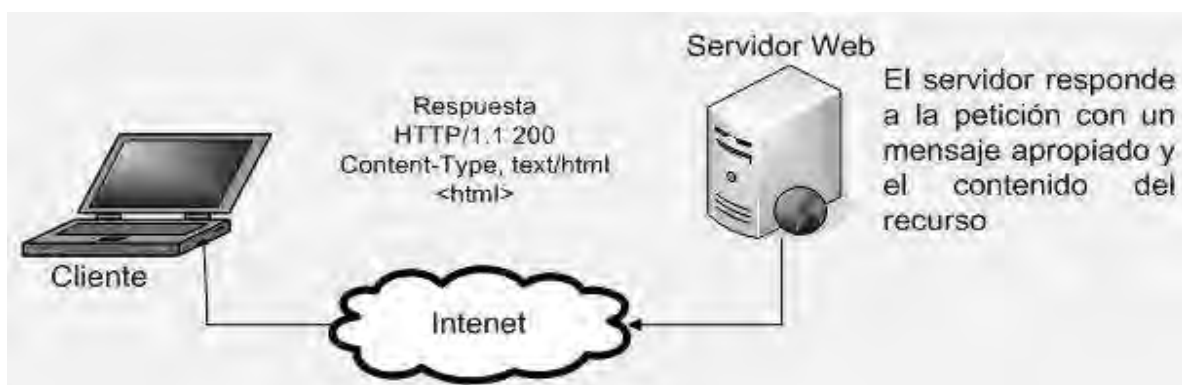


Figura 2.1.2.2 Interacción entre el cliente y el servidor Web: La respuesta HTTP.



Código de status	Descripción
1XX	Respuestas informativas
100	Continúa
101	Conmutando protocolos
102	Procesando (WebDav)
⋮	
2xx	Peticiones correctas
200	OK
201	Creado
202	Aceptada
⋮	
3xx	Redirecciones
301	Movido permanentemente
302	Movido temporalmente
304	No modificado
⋮	
4xx	Errores del cliente
401	No autorizado
403	Prohibido
404	No encontrado
⋮	
5xx	Errores de servidor
500	Error interno
503	Servicio no disponible
507	Almacenamiento insuficiente
⋮	

Tabla 2.1.2.1 Códigos de status HTTP

Después, el servidor envía uno o más encabezados HTTP, los cuales proporcionan información adicional sobre los datos que se van a enviar.

Por ejemplo:

- ✦ Content-type: text/HTML: El servidor está enviando un documento de texto XHTML.
- ✦ Content-type: text/plain: La información enviada es texto que puede mostrarse directamente, sin interpretar el contenido como marcado de XHTML.
- ✦ Content-type : image/jpeg: Indica que el contenido es una imagen JPEG



El encabezado, o conjunto de encabezados, va seguido por una línea en blanco, la cual indica al cliente que el servidor terminó de enviar encabezados HTTP. Después el servidor envía el contenido del documento HTML solicitado. El servidor termina la conexión cuando se completa la transferencia del recurso. El navegador del lado del cliente analiza el marcado de HTML que recibe y despliega (o visualiza los resultados).

Aplicación Web

Con base en lo anterior, una aplicación Web es aquella aplicación que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador o browser, de forma dinámica.

Es una aplicación de software que se codifica en un lenguaje soportado por los navegadores Web como HTML, JavaScript, Java, PHP, etc., en la que se confía la ejecución al navegador.

De esta manera una aplicación Web cuenta con un conjunto de páginas dinámicas y estáticas.

Páginas estáticas. Las páginas estáticas son archivos HTML alojados en un equipo que ejecuta un servidor Web, el servidor Web, que se verá más adelante en detalle, es quien suministra las páginas Web en respuesta a las peticiones de los navegadores Web.

El servidor sencillamente al recibir la petición de una página estática lee la solicitud, localiza la página y la envía al navegador que lo solicitó.

Cuando el servidor Web recibe una petición para mostrar una página Web estática, el servidor la envía directamente al navegador que la solicita (Ver Figura 2.1.2.3).

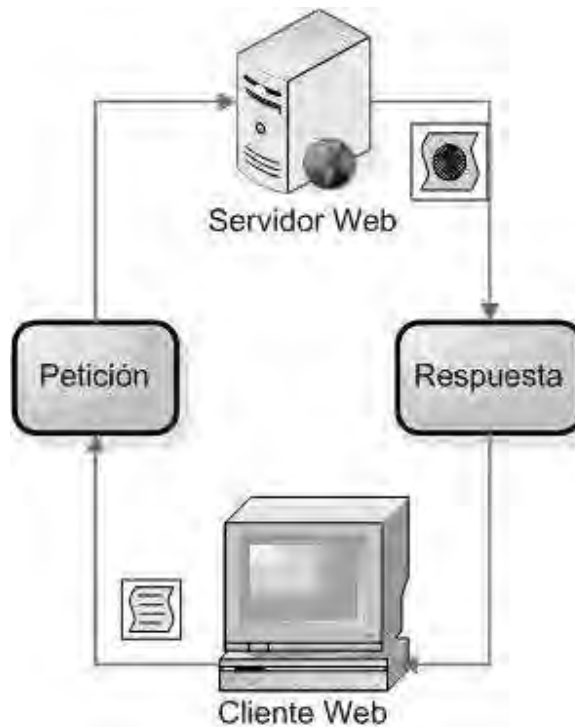


Figura 2.1.2.3 Procesamiento de una página Web estática

Página dinámica: Una página dinámica utiliza lenguajes que a su vez serán interpretados por el navegador, como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Permite una comunicación activa entre usuario y la información.

Cuando el servidor Web recibe una petición para mostrar una página dinámica, transfiere la página a un software especial encargado de finalizar la página. Este software especial se denomina servidor de aplicaciones. El servidor de aplicaciones lee el código de la página, finaliza la página en función de las instrucciones del código y elimina el código de la página. El resultado es una página dinámica que el servidor de aplicaciones devuelve al servidor Web, que a su vez la envía al navegador que la solicita, así lo único que el navegador recibe cuando llega la página es código HTML puro. (ver figura. 2.1.2.4)

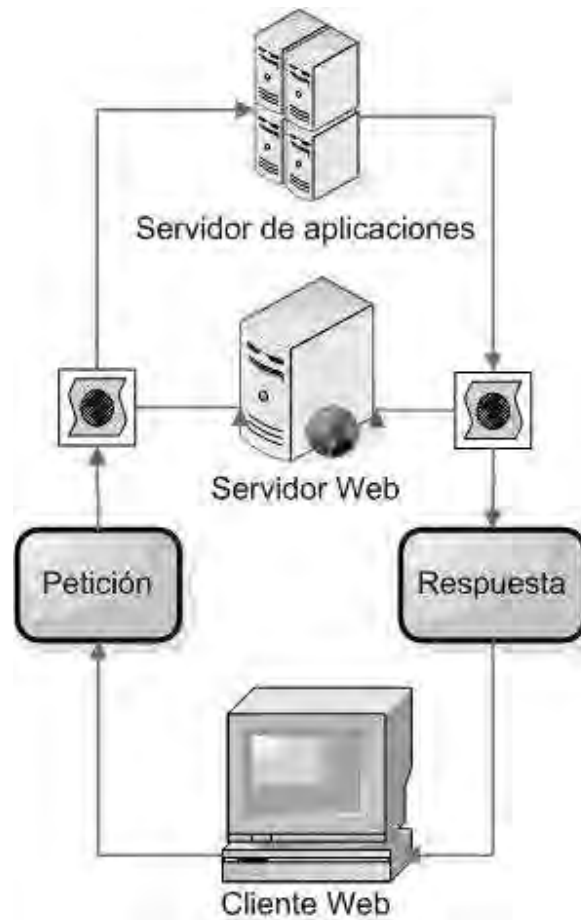


Figura 2.1.2.4 Procesamiento de una página Web dinámica

Funcionamiento de una aplicación Web

Las aplicaciones Web son aplicaciones multinivel (comúnmente conocidas como aplicaciones de n niveles), que dividen su funcionalidad en niveles separados, es decir, agrupaciones lógicas de funcionalidad.

Aunque los niveles pueden localizarse en la misma computadora, por lo general, los niveles de las aplicaciones Web residen en computadoras separadas.

En la figura 2.1.2.5 se presenta la estructura básica de una aplicación Web en tres niveles. Aunque existan muchas variaciones posibles en su forma más común una aplicación Web está estructurada como una aplicación en tres capas.

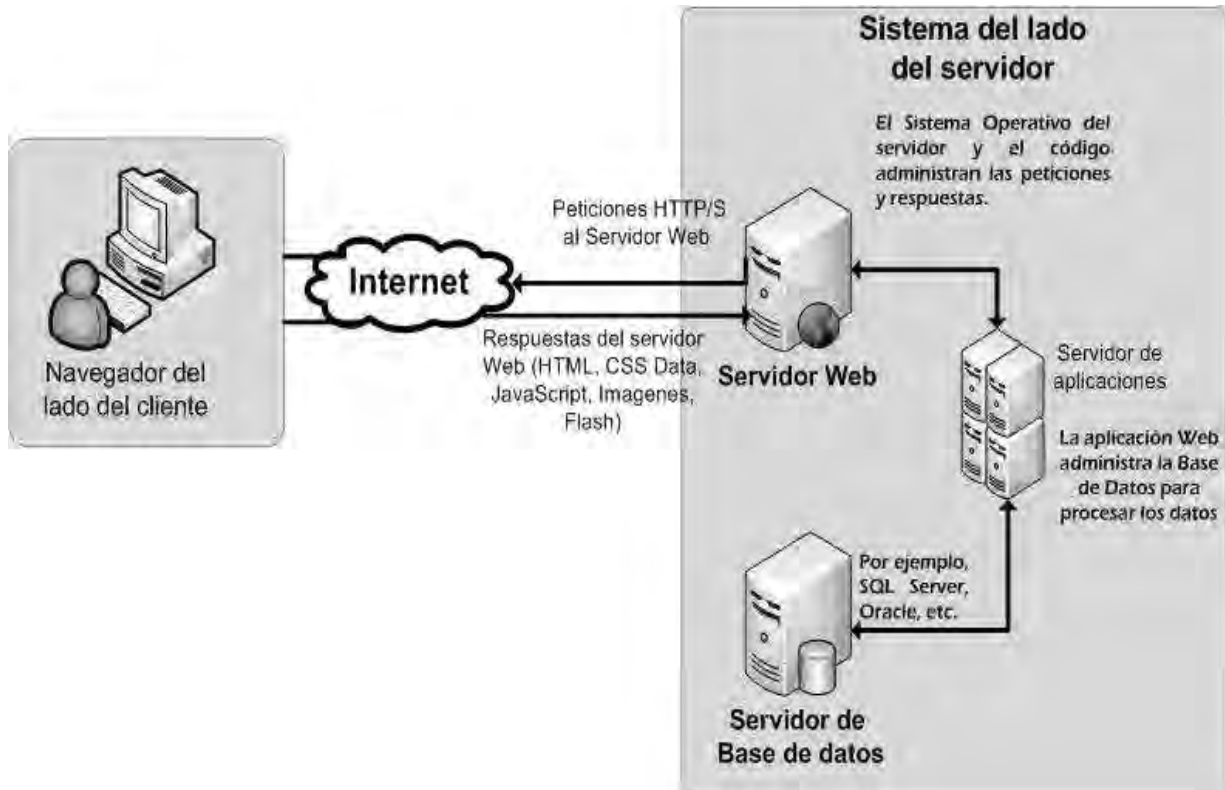


Figura 2.1.2.5 Arquitectura de tres niveles o capas.

Primer Nivel o nivel superior: Es la interfaz de usuario de la aplicación, la cual recopila los datos de entrada y de salida.

Segundo nivel o nivel intermedio: Implementa la lógica de negocios, de controlador y de presentación para controlar las interacciones entre los datos en el nivel de información y los clientes de la aplicación. La lógica de control de este nivel procesa las peticiones de los clientes y obtiene datos de la base de datos. La lógica comercial en el nivel intermedio hace valer las reglas comerciales y asegura que los datos sean confiables antes de que la aplicación servidor actualice la base de datos, o presente los datos a los usuarios. Las reglas comerciales dictan la forma en que los clientes pueden o no acceder a los datos de la aplicación, y la forma en que ésta procesa los datos. Es el motor capaz de usar alguna tecnología Web dinámica (PHP, JSP, ASP, Java, etc.).



Tercer Nivel o nivel inferior: También conocido como nivel de datos o nivel de información, mantiene los datos de la aplicación. Generalmente este nivel almacena los datos en un sistema de administración de bases de datos relacionales (RDBMS). Es donde se encuentra la base de datos que almacena el contenido y datos del cliente como nombres de usuario, contraseñas, números de seguridad social y detalles de tarjetas de crédito.

Proceso del acceso a una base de datos.

Puede utilizar prácticamente cualquier base de datos con la aplicación Web, siempre y cuando se haya instalado el controlador de base de datos correcto en el servidor. Los usuarios interactúan en forma directa con la aplicación a través de la interfaz de usuario, que por lo general es el navegador Web, el teclado y el ratón.

En respuesta a las acciones del usuario, el nivel cliente interactúa con el nivel intermedio para hacer peticiones y obtener datos del nivel de información. Después, el nivel cliente muestra los datos obtenidos para el usuario. El nivel cliente nunca interactúa directamente con el nivel de información. El navegador Web envía peticiones a la capa de en medio que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

En la figura de 2.1.2.6 se muestra el flujo del proceso de acceso a la base de datos en el que se aprecia como la petición inicial es activada por el usuario a través del navegador por Internet al servidor de aplicación Web.

La aplicación Web accede a los servidores de bases de datos para realizar la tarea pedida actualizando y recuperando la información que queda dentro de la base de datos. La aplicación Web entonces presenta la información al usuario a través del navegador.

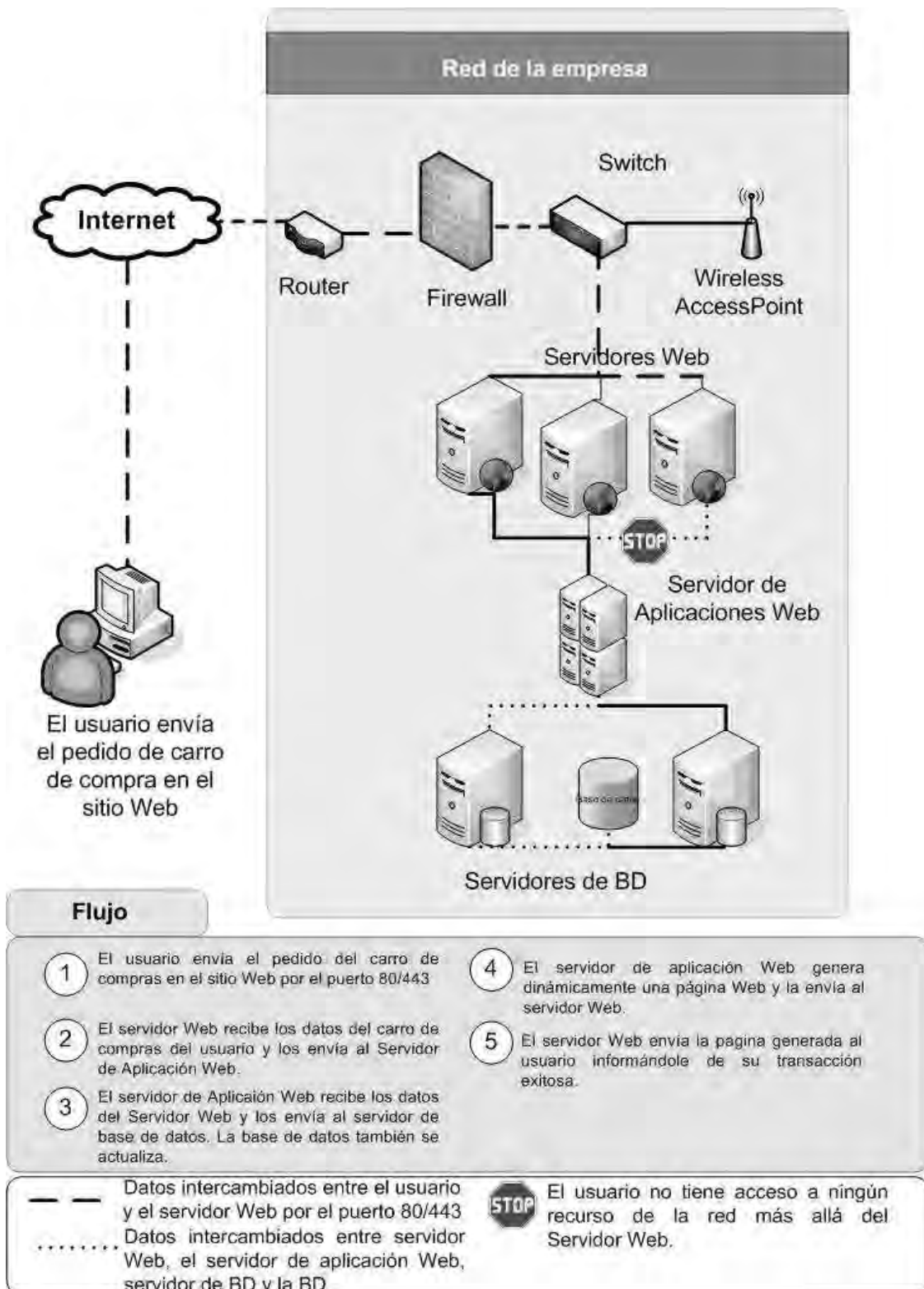


Figura 2.1.2.6 Proceso de acceso a la base de datos



Las aplicaciones Web solicitan al servidor de contenido (esencialmente una base de datos) y dinámicamente genera documentos Web del servidor al cliente. Los documentos son generados en un formato estándar que permite ser soportado por todos los navegadores (ej. HTML o XHTML).

El navegador es una pieza clave ya que es quien interpreta y ejecuta los scripts mientras muestra las páginas solicitadas y contenido, es el cliente universal para cualquier aplicación Web.

A continuación se detallan algunas ventajas y desventajas de las aplicaciones Web que podemos encontrar cuando se desea desarrollar aplicaciones de estas arquitecturas.

VENTAJAS

- Ligeras. Al ser ejecutadas en un navegador Web es práctico y ligero al cliente.
- Portables. Es independiente de la computadora donde se utilice debido a que accede a través de una página Web, solo es necesario contar con una conexión a Internet. Cuando el acceso se realiza mediante teléfonos móviles es necesario un diseño específico de los archivos CSS para no dificultar el acceso a estos usuarios.
- Para actualizarla y mantenerla es más fácil, ya que solo basta con actualizarla en el servidor y no es necesario distribuir e instalar la aplicación en todos los equipos de la red.
- Multiplataforma. Funcionan igual, independiente de la versión del sistema operativo instalado en el cliente. La aplicación Web se escribe una vez y se ejecuta igual en todas partes. Sólo es necesario contar con un navegador compatible y mínimamente actualizado.



-
- Ahorro de tiempo. Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa, eso mejora la rapidez en su funcionamiento.
 - No requiere de “grandes instalaciones”.
 - Actualizaciones inmediatas. Gracias a que el software lo gestiona el propio desarrollador, cuando se acceda a la aplicación Web se está usando la última versión que haya lanzado.
 - No ocupan espacio en el disco duro local.
 - Bajo consumo de recursos. Gracias a que la aplicación no se encuentra instalada en la computadora local muchas de las tareas que realiza la aplicación no consumen recursos de la computadora local ya que se realizan desde otra computadora.
 - Alta disponibilidad. Esto se logra debido a que el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
 - Existe cierto “blindaje” contra los virus ya que los datos están almacenados en el servidor de la aplicación.
 - Colaboración. El acceder y compartir los datos por parte de varios usuarios es más fácil ya que el acceso al servicio se realiza desde una única ubicación.

DESVENTAJAS

- Se ofrecen menos funcionalidades que las aplicaciones de escritorio, o más limitadas.



-
- Es necesario contar con una conexión a Internet permanente (generalmente) y una velocidad y ancho de banda promedio para una óptima navegación.
 - La disponibilidad depende de un tercero, el proveedor de la conexión a Internet, o el que provee el enlace entre el servidor de la aplicación y el cliente.
 - La seguridad de los datos, depende de la seguridad de la aplicación Web y del servidor donde este alojado.
 - Está limitado al espacio donde se guarda la información.
 - Poco soporte a varios idiomas.
 - En ocasiones depende bastante de la configuración de la privacidad del navegador para trabajar eficientemente (ej. JavaScript, cookies, Flash).
 - La estabilidad de la aplicación está sujeta al número de visitas en un mismo lapso de tiempo (sobrecarga del servidor).

2.2 Características, ventajas y desventajas de Apache, PHP, JavaScript y XML

2.2.1 Apache

Uno de los elementos que serán imprescindibles para una aplicación Web es contar con un servidor Web, para poder elegir cual es el que es conveniente para los propósitos del presente trabajo es necesario comprender algunas cuestiones relacionadas y que se muestran a continuación.



Servidor. Un servidor técnicamente es un sistema que proporciona recursos, este sistema es conocido como una computadora que pertenece a una red y que suministra información a otras computadoras, las cuales ejecutan los programas clientes.

Dependiendo el tipo de recursos que suministra es la clasificación del tipo de servidor, por ejemplo, servidor de archivos, de impresión, de correo, de telefonía, de acceso remoto, de bases de datos, o el que por ahora es de interés es el servidor Web.

Servidor Web. El servidor Web es un programa cuya función es atender y responder a las diferentes peticiones de los navegadores, proveyendo de los recursos que se soliciten usando el protocolo http (Hypertext Transfer Protocol,) o el https (Hypertext Transfer Protocol Secure), que es la versión cifrada y autenticada. Este protocolo pertenece a la capa de aplicación del modelo OSI (Open System Interconnection, Modelo de referencia de interconexión de sistemas abiertos) (Figura 2.2.1.1) y que está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (Hypertext Markup Lenguaje):

Textos complejos con enlaces, imágenes, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Este programa llamado servidor generalmente se instala en una computadora conectada en forma permanente en Internet, de esta manera podrá atender las solicitudes de páginas Web y tareas de procesos de datos que los clientes pudieran solicitar en cualquier momento.

Con esto existe la ventaja de la disponibilidad de la información desde cualquier punto de la red y además de que es posible almacenar aplicaciones en el servidor, permitiendo que el cliente no requiera de elementos adicionales para ejecutar la aplicación.

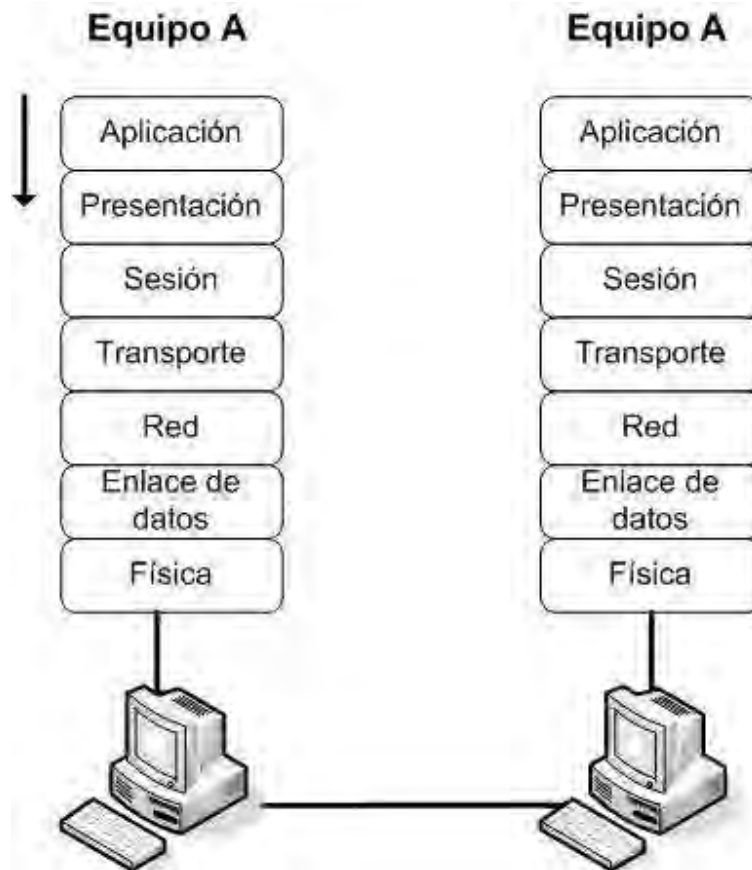


Figura 2.2.1.1 El protocolo HTTP opera en la capa de aplicación del modelo OSI

Gracias a los avances en conectividad y la gran disponibilidad de banda ancha, hoy en día es muy común establecer los servidores Web dentro de la propia empresa, sin tener que recurrir a caros alojamientos en proveedores externos.

Funcionamiento

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía telefónica o digital.

El ciclo que ejecuta un servidor Web básico es el siguiente:



- ✦ El servidor se ejecuta en una computadora manteniéndose a la espera de peticiones por parte de un cliente (navegador), esta espera de peticiones es en el puerto TCP (por defecto es el 80).
- ✦ Recibe petición.
- ✦ Busca recurso.
- ✦ Responde a estas peticiones enviando el recurso y utilizando la misma conexión por la que recibió la petición mostrándose en el navegador o señalándose el respectivo mensaje si se detectó algún error. El servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.
- ✦ El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, colores y disposición de los textos y objetos de la página.
- ✦ Regresa al paso en que se recibe la petición.

Como se vio en la sección 2.1.2 (aplicaciones Web) si existió algún problema de conexión se envía un mensaje de error de estatus http. (Figura 2.1.2.3).

El esquema anterior sólo es válido para páginas estáticas, por tanto no es suficiente actualmente ya que la tendencia es a desarrollar páginas dinámicas, de esta manera es importante que en la elección de un servidor Web se analice el soporte para contenido de tipo dinámico.

De esta manera además de transferir código HTML los servidores Web pueden entregar aplicaciones Web, estas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas http.

- ✦ *Aplicaciones en el lado del cliente:* El cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java “applets” o JavaScript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto,



que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts).

- ✦ Aplicaciones del lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo http.

Cuando se contrata un plan de alojamiento Web con una compañía, esta proporciona un servidor al dueño del sitio para poder alojarlo; al respecto hay dos opciones, optar por un “servicio dedicado”, lo que se refiere a una computadora servidora dedicada exclusivamente al sitio del cliente (para aplicaciones de alta demanda), o un “servidor compartido”, lo que significa que un mismo servidor (computadora + programas servidor) se usará para varios clientes compartiendo los recursos.

Para seleccionar el servidor Web que satisfaga nuestras necesidades se deberán tomar en cuenta los siguientes aspectos:

- ✦ Número de conexiones concurrentes.
- ✦ Número de transacciones por segundo.
- ✦ Soporte para el desarrollo.
- ✦ Seguridad. Que pueda garantizar la integridad (al viajar la información el contenido y el significado no se altere), confiabilidad (el servicio debe estar disponible en todo momento) y confidencialidad (que impida que personas no autorizadas lean y conozcan la información que se transmite) de la información.

Servidores Web más populares:

- ✦ Apache
- ✦ IIS
- ✦ Cherokee

Otros servidores, más simples pero más rápidos, son:

- ✦ lighttpd



✦ tthttpd

El más usado es el Apache del ambiente Unix, seguido por el IIS del sistema Windows Server. Se explica el predominio del servidor Apache porque es gratis, muy estable y confiable, tiene una gran capacidad y existen versiones para todas las plataformas de computación o se lo puede adaptar con facilidad

Servidor Web apache

Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular servidor Web NCSA HTTPd 1.3 cuyo desarrollo se suspendió en 1998, pero más tarde todo el código de Apache fue reescrito por completo. Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA, era en inglés, “a *patchy Server*” (un servidor "parcheado").

Posteriormente todo el soporte de Apache fue provisto por “The Apache Group” o “La Fundación Apache”, logrando la primera aparición de Apache en Abril de 1995. Este servidor se sigue desarrollando como un proyecto de Software libre.

La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP (conforme al protocolo HTTP/1.1) más usado. Alcanzó su máxima porcentaje de mercado en 2005 siendo el servidor empleado en el 72% de los sitios Web en el mundo para esa fecha.

Aunque el porcentaje de su uso ha descendido un poco, para Octubre de 2010, de acuerdo a la empresa Netcraft, empresa británica de servicios de Internet, el servidor



Web Apache sigue siendo el servidor más usado de Internet ocupando el 67% del mercado. Ver Figura 2.2.1.2

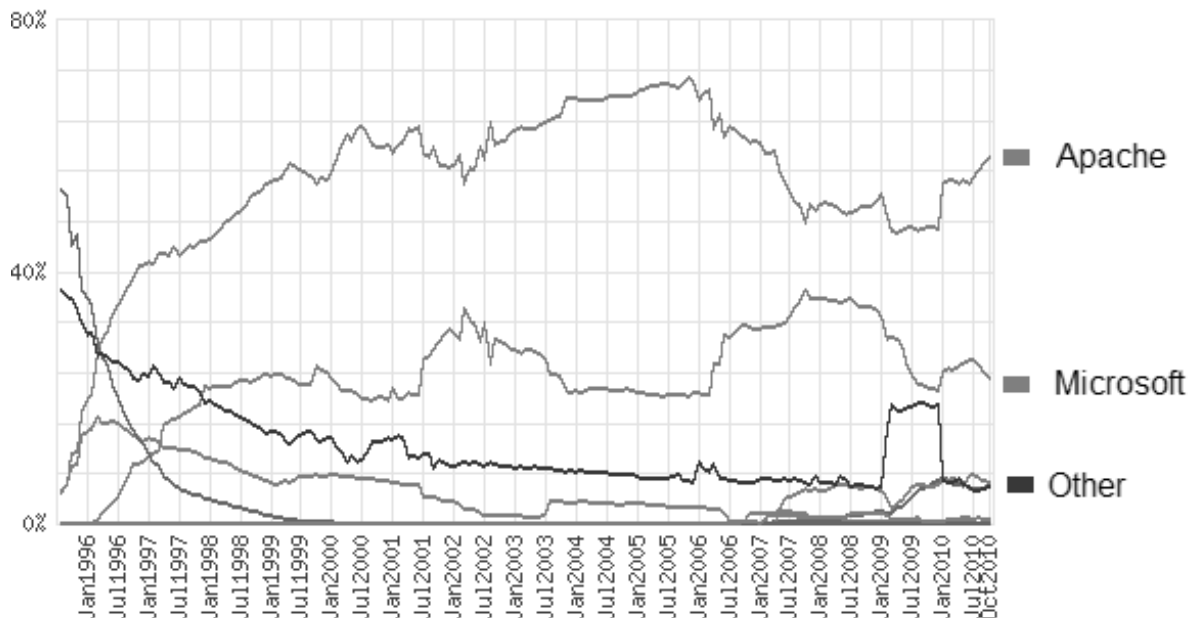


Figura 2.2.1.2 El servidor Apache es el más usado en los sitios Web a nivel mundial

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Es principalmente usado para servir páginas Web estáticas y dinámicas en la WWW. Apache es el servidor Web del popular sistema XAMP, junto con MySQL y los lenguajes de programación PHP/Perl/Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP, etc.

Características

Código abierto. Todo el código fuente de Apache está escrito en C, Apache es una tecnología gratuita de código fuente abierto (*open source*) es el término con el que se conoce al software distribuido y desarrollado libremente, es decir, existe transparencia en la forma de su desarrollo. El API de Apache es de dominio público por lo que pueden



realizarse mejoras al código además de agregarse nuevos módulos al servidor, así como una capacidad de adaptación a necesidades particulares. Existe una gran comunidad de desarrolladores a nivel mundial que desarrollan y lo mejoran gracias a esta realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos, logrando actualizaciones constantes sin necesidad de esperar a que un grupo reducido de desarrolladores resuelva un fallo sino que puede ser resuelta por un grupo más completo. Esto permite producir binarios para plataformas en las que no hay un binario oficial o compilarlo para un mejor rendimiento en una plataforma específica.

Es un servidor basado en procesos, utilizando la técnica pre-fork (consiste en la creación previa de un grupo de procesos y su mantenimiento hasta que sea necesaria su utilización).

Arquitectura modular. Arquitectura en la que existe una sección *core* o base y diversos módulos multiproceso y módulos adicionales que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web, es decir, gracias a esto se puede adaptar a diferentes entornos y necesidades con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos para el desarrollo de módulos específicos. Cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

Administración. Es uno de los elementos principales de cualquier servidor. Apache cuenta con un conjunto de archivos de configuración ampliamente documentados, estos se encuentran en formato ASCII y pueden ser editados con cualquier editor de texto. Si se necesita realizar alguna actualización resulta sencillo respaldar la configuración del servidor. Se puede administrar de forma remota ya que puede operarse y configurarse desde la línea de comandos, así como por medio de interfaces gráficas y un conjunto de herramientas desarrolladas con este propósito.



Extensible: Gracias a que es modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Personalizable, la arquitectura modular de Apache permite construir un servidor hecho a la medida. Además permite la implementación de los últimos y más nuevos protocolos.

Configuración de errores. Cuenta con archivos de configuración específicos que hacen posible configurar el servidor para que ejecute un determinado script cuando ocurra un error en concreto, es decir, Apache permite la creación de *logs* de errores configurados por el administrador. Tiene una alta configuración en la creación y gestión de *logs* que contienen enormes cantidades de información, necesitando una herramienta que realice resúmenes de los datos para facilitar el análisis del contenido, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

VENTAJAS

- ✦ *Eficiencia y estabilidad.* La eficiencia de Apache es lograda gracias a que se encuentra optimizado para consumir la menor cantidad de recursos del sistema en comparación a otros servidores, gracias a esto puede lograr un gran desempeño convirtiéndolo en una aplicación estable y robusta, soportando un mayor número de transacciones.
- ✦ *Multiplataforma.* Apache es soportado por la gran mayoría de plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementan el protocolo HTTP/1.11 y la noción de sitio virtual, es por tanto una aplicación sumamente portable y configurable en diferentes ambientes lo que lo hace prácticamente universal.
- ✦ *Extensibilidad.* Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente



existen muchos módulos para Apache que son adaptables a éste. Desde módulos que permiten generar contenido dinámico (con PHP, Java, Perl, Python), monitoreo del rendimiento, comunicaciones seguras por SSL, creación de servidores virtuales por IP. Otra cuestión importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

- ✦ *Seguridad.* La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas puede en la mayoría de los casos ser abusada solamente por los usuarios locales y no puede ser accionada remotamente. Cuenta con Módulos de autenticación: mod_access, mod_auth y mod_digest, así como Soporte para SSL y TLS. También permite autenticación de base de datos basada en SGBD.
- ✦ *Soporte para los lenguajes.* Apache trabaja con gran cantidad de lenguajes como Perl, Python, tcl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- ✦ *Personalización de los mensajes de error.* Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto y negociación de contenido de esta manera se puede tener un mayor control sobre lo que sucede en el servidor.

Desventajas

- ✦ *Administración compleja.* Como la mayoría de los programas open-source, uno depende de configurar los archivos a mano o tener que instalarse herramientas adicionales para las tareas de administración. Fue criticado por la falta de una interfaz gráfica que ayude en su configuración ya que resultar difícil de



configurar incluso para tareas sencillas y por ello algunos usuarios con poca experiencia tratan de evadir su uso ya que una configuración errónea puede hacer que no funcione adecuadamente el servidor, es necesario contar con la documentación necesaria respecto a los archivos de configuración, esto para poder ajustar correctamente algunos parámetros del servidor.

- ✦ *Falta de integración.* Al ser un producto multiplataforma, el servidor no aprovecha al máximo las posibilidades que ofrece el sistema operativo.
- ✦ *Formatos de configuración no estándar.* Esto dificulta un poco la automatización y el procesamiento de la configuración al no estar basada en formatos más soportados como el XML.
- ✦ *Lentitud de arranque y ejecución.* El servidor es aproximadamente un 20% más lento en su arranque debido a la sobrecarga que la resolución representa para el cargador (loader) y es aproximadamente un 5% más lento en su ejecución bajo algunas plataformas.

2.2.2 PHP

Fue originalmente diseñado en Perl como un complemento para ese lenguaje, con base en la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense Rasmus Lerdorf en el año 1994. El 8 de junio de 1995 fue publicado "*Personal Home Page Tools*" después de que Lerdorf lo combinara con su propio *Form Interpreter* para crear PHP/FI. En 1997 las aportaciones de Zeev Suraski y Andy Gutsman llevaron al desarrollo del PHP versión 3 y fue cuando éste se popularizó de manera definitiva, sobre todo al combinarse con la base de datos MySQL y el servidor Apache. Dado que estos tres productos son Open Source, o código abierto, puede usarse sin limitación alguna por cualquier usuario o empresa que así lo desee. La implementación principal de PHP ahora es producida por *The PHP Group*. Actualmente PHP se usa como acrónimo recursivo de **H**ypertext **P**re-processor (Pre-procesador de hipertexto) y se encuentra instalado en más de 20 millones de sitios Web y en un millón de servidores. De acuerdo a la empresa *tiobe.com*, (dedicada a las



estadísticas de uso de los lenguajes de programación más populares), (ver tabla 2.2.2.1) PHP es ubicado en 2010 en la cuarta posición ocupando el 8.32% del mercado.

Position Oct-10	Position Oct-09	Delta in Position	Programming Language	Rantings Oct-10	Delta Oct-09
1	1	=	Java	18.17%	-0.48%
2	2	=	C	17.18%	0.33%
3	3	↑	C++	9.80%	-0.08%
4	4	↓	PHP	8.32%	-2.03%
5	5	=	(Visual) Basic	5.65%	-3.04%
6	6	=	C#	4.96%	0.55%
7	7	=	Python	4.86%	0.96%

Tabla 2.2.2.1 Estadísticas de uso de lenguajes de programación (ww.tiobe.com).

Incluso con su competidor más cercano de su categoría por la parte de software propietario se encuentra muy por encima. (Ver figura 2.2.2.1)

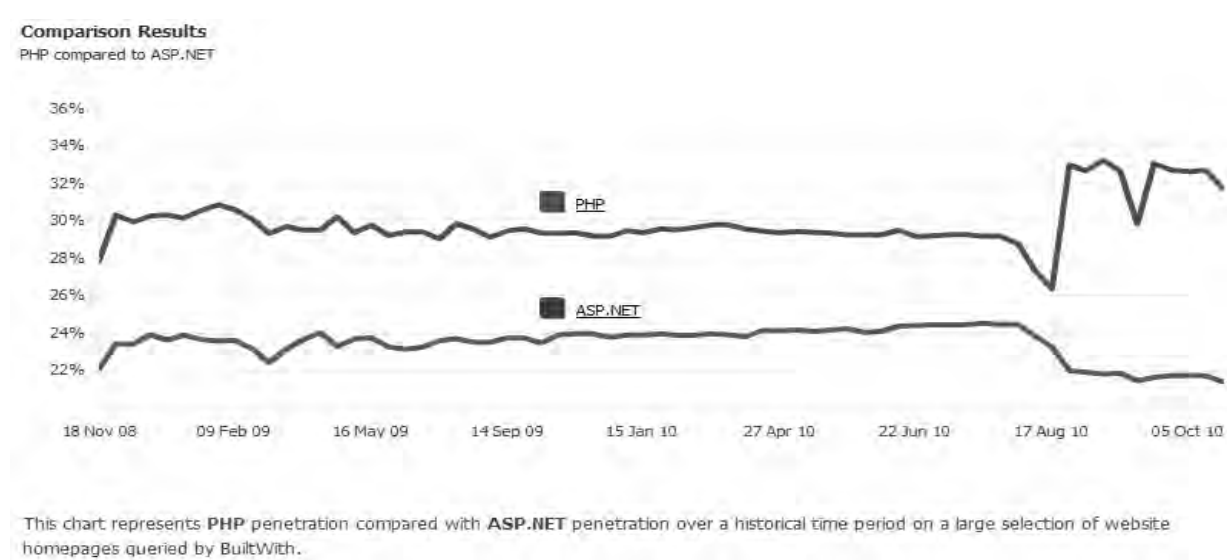


Figura 2.2.2.1 Estadística PHP vs ASP.NET (<http://trends.builtwith.com>)



Definición del lenguaje PHP

Es un lenguaje de programación interpretado, diseñado para la creación de páginas Web dinámicas y que puede ser incrustado dentro de código HTML, usado principalmente en interpretación del lado del servidor tomando el código en PHP como su entrada y creando páginas Web como salida. Actualmente puede ser utilizado desde una interfaz de línea de comandos.

Combinando con la base de datos MySQL, PHP es el lenguaje estándar a la hora de crear sitios de comercio electrónico o páginas Web dinámicas. Posee un gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, este lenguaje fue diseñado especialmente para el desarrollo de páginas Web dinámicas. Por ello, está dotado de un gran número de funciones que simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con *cookies* y sesiones, etc.

Modo de operación

Cuando el cliente hace una petición al servidor para que le envíe una página Web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos. Podemos resumir la manera en que opera PHP en la siguiente manera:

- ✦ El Navegador realiza una petición al servidor (se escribe la URL).
- ✦ Después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador.
- ✦ Por último el Navegador muestra la respuesta del servidor.



Este tipo de iteración permite algunas operaciones complejas como conexiones a bases de datos o ejecución de complejos programas. PHP además de soportar un número masivo de bases de datos, también nos ofrece una gran variedad de opciones que nos permiten desarrollar múltiples funcionalidades que van desde enviar un e-mail, subir un archivo (upload), crear una imagen en tiempo de ejecución, interactuar con diversos protocolos de comunicación, interactuar con documentos XML, autenticación, creación dinámica de documentos PDF, entre muchas otras cosas.

Características

El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

Lo que distingue a PHP del lado cliente como Javascript, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP y entonces no haya manera que los usuarios puedan saber el código fuente completamente.

Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.

Diseñado especialmente para aplicaciones Web PHP, similar al ASP de Microsoft o el JSP de Sun, puede ser incrustado dentro de código HTML el cual se ejecuta en un servidor Web, tomando como entrada el código hecho en PHP y dando como salida la creación de páginas Web. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C.



A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta del lado del servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap). Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas.

Ventajas

- ✦ Es libre y es uno de los productos más representativos y utilizados del Open Source, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✦ Gran popularidad. Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
- ✦ Gratuito. Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.
- ✦ Multiplataforma. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. PHP también soporta la mayoría de los servidores Web de hoy en día, esto incluye: Apache, Microsoft Internet Information Server, Personal Web Server, los servidores Netscape e iPlanet, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd, y muchos otros. Para la mayoría de los servidores, PHP tiene un módulo, para los otros que soporten el estándar CGI, PHP puede funcionar como un procesador CGI.
- ✦ Completamente orientado al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en una Base de Datos.
- ✦ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.



-
- ✦ Segura y confiable. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo *.ini
 - ✦ Permite aplicar técnicas de programación orientada a objetos. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
 - ✦ Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Es importante para una página Web verdaderamente dinámica, una correcta integración con base de datos. Aunque My SQL es la base de datos que mejor trabaja con PHP, puede conectarse también a PostgreSQL, Oracle, dbm, filePro, interbasem o cualquier otra base de datos compatible con ODBC (*Open Database Connectivity Standard*).
 - ✦ Enorme eficiencia. Con escaso mantenimiento y un servidor gratuito (en nuestro caso, Apache), puede soportar sin problema millones de visitas diarias.
 - ✦ Velocidad. No solo la velocidad de ejecución, la cual es importante, sino además no crea demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, esta listo para ser utilizado.
 - ✦ Estabilidad. La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho mas difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
 - ✦ Simplicidad y expansión. Es fácil de aprender, usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente asimismo se cuenta con la capacidad de



expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).

- ✦ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Biblioteca nativa de funciones sumamente amplia e incluida.

Desventajas

- ✦ El manejo de errores no es tan sofisticado como Cold Fusion o ASP. No existe IDE o Debugger, aunque una IDE puede no ser importante para la mayoría de los programadores.
- ✦ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✦ Al llamar y ejecutar PHP una función, consume más recursos (se vuelve ligeramente más lento) que ejecutando código que encuentra en línea, embebido en el script. Por eso no tiene sentido usar funciones salvo que efectivamente se vaya a utilizar esa porción de código varias veces.

2.2.3 JavaScript

Otro de los elementos importantes de una aplicación AJAX y del cual debe parte de su nombre es la “J” de JavaScript.

JavaScript fue creado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cuál fue renombrado posteriormente a LiveScript en 1995 para la versión 2.0 de su navegador, poco tiempo después Microsoft incorporaba a su navegador una implementación de ese mismo lenguaje, al que denominó JScript.



Cada nueva versión de Netscape y de Internet Explorer introducía novedades en sus particulares implementaciones de JavaScript, cada vez más incompatibles entre sí, tendencia que llegó a su fin cuando ambas empresas acordaron en Noviembre de 1996 que un organismo independiente, concretamente la institución europea de estandarizaciones ECMA, crease una especificación común del lenguaje que sería la tomada como referencia para futuras versiones de sus respectivos navegadores. Dicha especificación de referencia se denominó EcmaScript. En Junio de 1997 es adoptado este estándar y luego también por la “*Internacional Organization for Standardization*” (ISO).

Por tanto JScript no es lo mismo que JavaScript. Este último es el estándar (también se le llamó ECMAScript), mientras que el primero es propiedad de Microsoft.

La primera versión de EcmaScript surgió en 1998, seguida por actualizaciones en años posteriores, que en menor o mayor medida, han sido incorporadas en los distintos navegadores en uso actualmente. La especificación de la tercera edición de EcmaScript, fue en diciembre de 1999 y es la empleada como guía por los distintos fabricantes.

El DOM por sus siglas en inglés “Modelo de Objetos del Documento” fue diseñado para evitar incompatibilidades.

Actualmente es la fundación Mozilla la que sigue trabajando en la evolución de JavaScript con un mayor empeño.

Aunque JavaScript no es nuevo, pues existe hace más de una década, ha sido precisamente su uso en el desarrollo de aplicaciones AJAX lo que lo ha llevado a resurgir.



De acuerdo a la empresa *tiobe.com* dedicada a las estadísticas de uso de los lenguajes de programación JavaScript actualmente se encuentra dentro de los 10 lenguajes de programación más usados.

Funcionamiento

El lenguaje JavaScript es usado para realizar pequeñas aplicaciones que interactúan dentro de una página HTML. La forma en la que funciona principalmente JavaScript es de modo dinámico y para realizar interacciones con el usuario. La única forma donde puede ser ejecutado JavaScript es mediante un navegador Web, el cual interpreta y realiza según esté programado mediante los llamados eventos o instancias.

Las acciones que generalmente se suele utilizar en JavaScript:

- ✦ Efectos visuales: Donde se puede cambiar el estilo, crear movimientos a objetos y elementos HTML, etc.

- ✦ Interacción con el usuario: Donde podemos realizar instancias a consecuencia de una interacción con el usuario e incluso cambiar el contenido de la página Web.

Características:

- ✦ JavaScript es un lenguaje que lo mismo puede ser utilizado para pequeñas comprobaciones o alteraciones del contenido HTML, como también ser usado con funciones y demás estructuras complejas según requiera el sitio.

- ✦ Algunos consideran que JavaScript no es un lenguaje orientado a objetos por lo que no es necesario crear ni clases, ni métodos, etc., como en lenguajes de programación estructurada como C, Perl, etc.



- ✦ Sintaxis. Es similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún compilador adicional.
- ✦ *Lenguaje no extensible.* JavaScript en comparación con Java, no es totalmente extensible. El modelo de JavaScript está formado por un número limitado de objetos básicos, propiedades, métodos y tipos de datos, que son suficientes para desarrollar aplicaciones cliente-servidor.
- ✦ *Basado en acciones.* Es un lenguaje basado en acciones que posee menos restricciones.
- ✦ Lenguaje interpretado. Esto significa que los archivos con el código se ejecutan directamente. Ese tipo de lenguajes son más sencillos y fáciles de aprender. Además, normalmente es más fácil desarrollar, modificar y depurar programas para lenguajes interpretados porque no es necesario recompilar el código cada vez que se realiza un cambio. Los programas para lenguajes interpretados suelen requerir un mayor tiempo de ejecución que los programas compilados. En el caso de JavaScript esto no supone un problema, ya que, con los anchos de bandas actuales, la velocidad de ejecución es despreciable frente a la velocidad de transmisión. Los scripts escritos en JavaScript se compilan en formato Bytecode (como los programas Java) y evalúan mientras se está transmitiendo el script.

Actualmente existen dos tipos de JavaScript:

- ✦ El que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript.
- ✦ Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.



Frameworks y librerías

Las aplicaciones Web son cada vez más complejas, ya que incluyen efectos e interacciones que hasta hace poco tiempo eran exclusivas de las aplicaciones de escritorio. Al mismo tiempo, la programación de estas aplicaciones avanzadas se complica por varios motivos.

En primer lugar, las aplicaciones comerciales deben funcionar correctamente e igual de bien en al menos cinco navegadores diferentes: Internet Explorer 6 y 7, Firefox, Opera y Safari. En segundo lugar, el tiempo disponible para el desarrollo de cada nueva característica se reduce cada vez más por la necesidad continua de incluir novedades en las aplicaciones antes de que las incluya la competencia.

Por todo lo anterior, han surgido librerías y frameworks específicos para el desarrollo de aplicaciones con JavaScript. Utilizando estas librerías, se reduce el tiempo de desarrollo y se tiene la seguridad de que las aplicaciones funcionan igual de bien en cualquiera de los navegadores más populares.

La librería en JavaScript jQuery (fig 2.2.3.1) es una de las más sencillas y útiles que se puedan necesitar a la hora de desarrollar un sitio Web. Posee características potentes (selectores de CSS, XPath, con el poder de javascript de crear variables y funciones que interactúen con el documento) y es muy fácil de usar, a su vez es muy liviana. Además trae incorporadas funciones adicionales, como AJAX, manipulación del DOM y detección de navegadores.

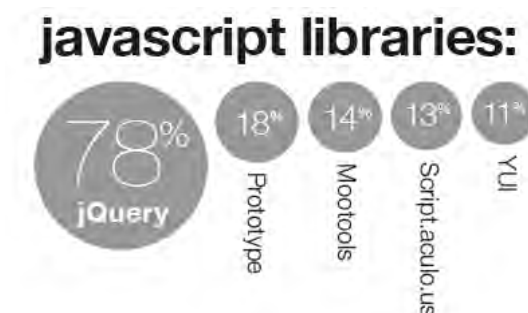


Figura 2.2.3.1 Estadísticas de uso para jQuery (<http://webdirections.org>)



Ventajas

- ✦ El código Javascript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido.
- ✦ El lenguaje de scripting es seguro y fiable porque está en claro y hay que interpretarlo, por lo que puede ser filtrado; para el mismo Javascript, la seguridad es casi total y sólo en su primera versión el CIAC (Computer Incident Advisory Committee) señaló problemas de leve entidad, entre ellos la lectura de la caché y de los sitios visitados, de la dirección e-mail y de los archivos presentes en el disco. Sin embargo, estos fallos se corrigieron ya en las versiones de Netscape sucesivas a la 2.0;
- ✦ No requiere tiempo de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.
- ✦ Javascript tiene la ventaja de ser incorporado en cualquier página Web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.
- ✦ Es un lenguaje independiente de la plataforma.
- ✦ Es simple, no hace falta tener conocimientos de programación para poder hacer un programa en JavaScript y para su desarrollo se necesita básicamente un editor de textos y un navegador Web compatible con JavaScript para sus pruebas.
- ✦ Maneja objetos dentro de nuestra página Web y sobre ese objeto se pueden definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos, etc.



- ✦ Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del ratón sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto podemos cambiar totalmente el aspecto de nuestra página al gusto del usuario, evitándonos tener en el servidor un página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.
- ✦ Los programas JavaScript tienden a ser pequeños y compactos no requieren mucha memoria ni tiempo adicional de transmisión.

Desventajas

- ✦ Los script tienen capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con Javascript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java. Se puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés *Cross Site Scripting* renombrado a XSS por su similitud con las hojas de estilo CSS).
- ✦ Una desventaja considerable también es que el código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright.
- ✦ El código del script debe descargarse completamente antes de poderse ejecutar, si los datos que un script utiliza son muchos, el tiempo que tardará en descargarse será muy largo.

2.2.4 XML

Un elemento importante más de una aplicación AJAX y del cual es la “X” de su nombre es XML.

XML (*Extensible Markup Lenguaje*, lenguaje de marcas generalizado). Es un lenguaje que se utiliza para estructurar información de manera jerárquica en un documento o en



general en cualquier archivo que contenga texto, como puede ser una tabla de datos, configuración de programas o resultado de base de datos ordenados para ser mostrado en aplicaciones AJAX.

Se ha usado en los últimos años debido a ser algo estándar, abierto y libre. Creado por el Consorcio World Wide Web, W3C (los creadores de la WWW), en colaboración de las principales compañías productoras de software. La primera propuesta XML fue en 1996 y la primera especificación apareció en 1998. Desde entonces su uso ha tenido un crecimiento acelerado, uno de las principales razones de su crecimiento ha sido la aparición de AJAX.

Ventajas de su uso

Previamente, ya existían otros lenguajes de marcas, como por ejemplo el HTML, basado en el lenguaje generalizado de marcas (SGML). El problema con el lenguaje generalizado de marcas es que, por ser muy flexible y muy general, se complica el análisis sintáctico de un documento y la especificación de la estructura. XML es más exigente que SGML en la sintaxis, lo que facilita la construcción de librerías para procesarlo (como pueden ser lectores de RSS, script de proceso de feeds, etc.).

Comparado con otros sistemas usados para crear documentos, el XML tiene la ventaja de poder ser más exigente en cuanto a la organización del documento, con lo que se obtienen documentos mejor estructurados.

Por ser posible exigir la estructura que deben tener un tipo determinado de documentos, se vuelve posible extraer información de varios documentos automáticamente, por ejemplo para crear bases de datos o listados con información sobre todos los documentos.



Conceptos básicos de XML

Los archivos XML son archivos de texto, que generalmente utiliza una codificación Unicode, pero se pueden usar otros alfabetos según convenga para la aplicación.

Existen cinco caracteres especiales en XML:

- ✦ Los símbolos menor que (<) y mayor que (>).
- ✦ Las comillas dobles (“”).
- ✦ El apóstrofe (’).
- ✦ El carácter (&)

Los símbolos mayor que y menor que se usan para delimitar las marcas que dan la estructura al documento. Hay una nomenclatura especial cuando dentro del archivo XML se necesita alguno de los cinco caracteres especiales. Para este tipo de situaciones se hace uso de las diferentes entidades estándar HTML:

- ✦ Para el símbolo menor que (<), se utiliza <
- ✦ Para el símbolo mayor que (>), se utiliza >
- ✦ Para el símbolo comillas dobles (“”), se utiliza "
- ✦ Para el símbolo apóstrofe (’), se utiliza '
- ✦ Para el símbolo ampersand (&), se utiliza &

De esta manera, se observa claramente por qué “&” es un carácter especial, se utiliza para representar entidades; una entidad es un carácter adicional que no forma parte del alfabeto que se emplea por defecto en el texto (los caracteres especiales quedan excluidos del alfabeto usado para el texto) comienza por &, seguido del nombre de la entidad e inmediatamente un punto y coma.

Una diferencia importante con SGML, y en particular HTML, es que los nombres de las marcas y de sus atributos distinguen entre mayúsculas y minúsculas; <a> y <A> serían dos marcas diferentes. Generalmente se utilizan únicamente minúsculas para los nombres de las marcas y de sus atributos. Otra diferencia sobresaliente con SGML es que en XML ninguna marca se puede dejar abierta; es decir, por cada marca, por



ejemplo `<td>`, deberá existir una marca correspondiente `</td>` que indica dónde termina el contenido de la marca.

Definición del tipo de documento (DTD)

Las posibles marcas que pueden aparecer en un documento XML y los atributos que éstas pueden tener, son definidos en un archivo de definición del tipo de documento (DTD, Document Type Definition).

Cada documento XML debe indicar al comienzo el DTD usado por medio de una marca `<!DOCTYPE>`, por ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Esta marca indica que lo que viene a continuación en el archivo es una marca "HTML" (con todas sus posibles sub-marcas), que ha sido definida en un DTD que se llama `-//W3C//DTD XHTML 1.0 Strict//EN`. La palabra clave PUBLIC procede al nombre oficial que se le ha dado al DTD respectivo; en este caso nos da alguna indicación adicional al nombre del DTD: el símbolo "-" indica que no es un DTD reconocido por alguna entidad oficial, en este caso `w3.org`, el lenguaje usado en el DTD es el inglés (EN) y la sintaxis usada es sintaxis XML. El nombre que viene entre comillas después de PUBLIC es algo arbitrario, pero como en cada sistema existe un catálogo SGML que identifica los DTD disponibles en el sistema, lo importante es usar exactamente el nombre que aparezca en el catálogo. Y para que documentos que usen el mismo DTD puedan ser transportables entre sistemas conviene usar la identificación exacta sugerida por el autor del DTD.

Después del identificador público (lo que está entre comillas después de PUBLIC) puede venir un identificador del sistema que indica el camino y nombre del archivo



donde se encuentra el DTD; en el ejemplo anterior el identificador del sistema es una URL que indica donde se puede encontrar el DTD usado.

Un archivo DTD define siempre una o más estructuras jerárquicas, con una marca principal, o padre, compuesta por otras marcas o hijos.

Un documento XML que especifique el DTD usado y siga las reglas en él definidas, se dice que es un documento XML válido.

Archivos XML. Se les suele dar un nombre determinado en .xml para identificarlos como XML. Esto es simplemente una convención para los usuarios; el estándar XML 1.0 indica que para identificar un archivo como XML es necesario que la primera línea tenga el siguiente contenido:

```
<?xml version="1.0" ?>
```

Dentro de esta marca puede ir otra información adicional. El alfabeto usado por defecto en los archivos XML es el Unicode; para documentos en español será más conveniente usar el alfabeto latin-1, lo cual se logra usando el atributo *encoding* de la etiqueta xml:

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

2.3 Características, ventajas y desventajas de MySQL

Una de las funciones de la parte servidor de una aplicación AJAX es generalmente almacenar y recuperar datos transferidos desde o hacia el cliente. Cuando la información a manipular es de tamaño considerable no es una buena opción almacenarla en archivos de XML y mantenerlos manualmente, por tanto es necesario una base de datos y su correspondiente aplicación para administrarla.

Para que pueda existir una comunicación con el servidor de bases de datos no se usará el protocolo HTTP con los métodos GET y POST sino con el lenguaje que



entiende este tipo de aplicaciones que no es más que SQL (*Structured Query Language*, Lenguaje de consultas estructurado).

Sistema de Gestión Bases de Datos

Por tanto para el desarrollo de la aplicación Web será necesario contar con un elemento dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, este elemento es el RDBMS (***Relational Database Management System***) o Sistemas de Gestión bases de datos relacionales que (SGBDR en español) se instalan en un equipo y a través de una red los clientes se comunican para trabajar sobre las bases de datos.

El RDBMS se encarga de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales.

MySQL es el RDBMS más popular entre los producidos bajo los términos del Open Source, debido a su gran estabilidad y que fue diseñado para soportar sistemas de producción de misión crítica con altas cargas de procesamiento. Además existe una versión comercial distribuida y soportada por la compañía MySQL AB.

Características

Veloz. MySQL se caracteriza por su velocidad y estabilidad, rápido, fácil de administrar y usar. Posee un sistema de direccionamiento de memoria rápido basado en *threads* (subprocesos o hilos). Tablas en memoria que son usadas como tablas temporales, logrando con esto gran velocidad de respuesta al realizar las operaciones, esto lo hace ser uno de los gestores con mayor rendimiento.

Open Source. MySQL posee licencia GNU/GPL o Comercial, es Open Source, es decir, software de fuente abierta. Fuente abierta significa que es posible para cualquier



persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Si no se desea ajustarse al GPL o se requiere introducir código MySQL en aplicaciones comerciales, se puede comprar una versión comercial licenciada.

Tipos de datos. Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos: numéricos, fecha-hora y cadenas (caracteres).

Conectividad. Los clientes pueden conectarse al servidor MySQL mediante TCP/IP o sockets de UNIX; utilizando aplicaciones hechas con C, C++, Java, Perl o PHP. Es importante mencionar que MySQL puede conectarse directamente a otros manejadores de bases de datos utilizando ODBC.

Multiplataforma. Funciona sobre múltiples plataformas incluyendo: AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, eBD, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7 y Windows Server (2000, 2003 y 2008).

Procesamiento. MySQL puede manipular el multiprocesamiento distribuido en uno o más CPU's.

Seguridad. La utilización de privilegios y contraseñas, hace de MySQL un gestor de bases de datos relacionales flexible y muy seguro, todas las claves viajan encriptadas a través de la Web. Mediante la dirección IP del cliente, MySQL puede restringir y administrar el acceso a los recursos del servidor.



Internacionalización. MySQL soporta UNICODE, admitiendo con esto el desarrollo de aplicaciones sin importar la plataforma o el lenguaje del país donde se genere. También permite el manejo interno de mensajes de error.

Almacenamiento dinámico de comandos SQL. Esto hace que MySQL tenga un mejor desempeño que los demás manejadores de bases de datos.

Asimismo soporta:

Operaciones de revisión del estado, las cuales son utilizadas para transferir una base de datos a un determinado estado después de una falla seria del sistema.

VENTAJAS

- **Escalabilidad.** Soporte a grandes bases de datos. Se usa MySQL Server con bases de datos que contienen hasta 50 millones de registros y más de 60,000 tablas. Soporta 64 índices por tabla y cada índice puede estar compuesto de 15 columnas o partes de ellas, con una longitud máxima de 1,000 bytes. Los índices pueden utilizar prefijos de columnas de tipo CHAR, VARCHAR, BLOB o TEXT.
- **Integridad.** Para revisar, optimizar y reparar tablas u otros objetos de una base de datos, MySQL posee comandos y mecanismos internos.
- **Conectividad.** Los clientes pueden conectarse con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, ó 2003). los clientes pueden usar *named pipes* (conexión de software temporal entre dos programas o comandos) para la conexión. En sistemas Unix, los clientes se pueden conectar usando ficheros socket Unix. En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción *shared-memory* (memoria compartida). Los clientes se pueden conectar a través de memoria compartida usando la opción *protocol=memory*. La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones



ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.

- **Ligero.** Para la elaboración de bases de datos no necesita de grandes requerimientos ya que puede ser ejecutado en una máquina con escasos recursos debido a su bajo consumo, sin problemas.
- **Portabilidad.** Escrito en C y en C++, Probado con un amplio rango de compiladores diferentes, Funciona en diferentes plataformas, Usa GNU Automake, Autoconf, y Libtool para portabilidad, APIs (Interfaz de Programación de Aplicaciones) disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl, Uso completo de multi-hilo mediante hilos del kernel. Pueden usarse fácilmente múltiple CPU's si están disponibles, Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si se desea añadir una interfaz SQL para una base de datos propia, Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas, El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- **Diseño Multicapas.** El diseño de MySQL Server es multicapas, con módulos independientes. Algunos de los últimos módulos se listan a continuación con una indicación de lo bien probados que están:
 - ⊕ Replicación. Hay grandes grupos de servidores usando replicación en producción, con buenos resultados. Se trabaja para mejorar características de replicación en MySQL 5.x.
 - ⊕ InnoDB tablas. El motor de almacenamiento transaccional InnoDB es usado en grandes sistemas de producción con alta carga de trabajo.



✦ BDB tablas. El código Berkeley DB es muy estable, pero todavía se está trabajando para mejorarlo con el interfaz del motor de almacenamiento transaccional BDB en MySQL Server.

- Facilidad de configuración e instalación
- Baja probabilidad de corromper datos

DESVENTAJAS

Aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características y aunque algunas de ellas ya se han incluido en las versiones más recientes no se puede asegurar la estabilidad y buen funcionamiento de estas:

- *Subconsultas*. Tal vez ésta sea una de las características que más hacía falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
- *Triggers*. Aunque éstos ya se incluyen en la última versión, no es posible hablar de una gran estabilidad ya que los *triggers* reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
- *Procedimientos almacenados*. Los procedimientos almacenados fueron implementados hasta la versión 5, la cual aún presenta algunos problemas de estabilidad.
- *Transacciones*. A partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
- *Integridad referencial*. Aunque admite la declaración de claves foráneas en la creación de tablas, internamente no las trata de forma diferente al resto de los campos.
- *Índices*. MySQL utiliza a menudo índices para realizar operaciones de clasificación rápidamente sin examinar todas las filas. Los índices agilizan la recuperación pero hace más lentas las operaciones de insertar, borrar, así como las actualizaciones de valores en las columnas indexadas, es decir, la mayor



parte de las operaciones que implican escritura porque la escritura no afecta sólo a la fila de datos, sino a menudo también a los índices. Si tiene muchos índices, los archivos pueden alcanzar el tamaño máximo de archivo antes que el de datos.

2.4 Características, ventajas y desventajas de AJAX

Breve historia de AJAX

*El término AJAX se presentó por primera vez en el artículo "AJAX: A New Approach to Web Applications (Un nuevo acercamiento a las aplicaciones Web)" publicado por el programador Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación Web que estaba apareciendo. El artículo define AJAX como un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". AJAX no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen.*

Esta forma de programación describe el proceso de utilización del objeto *XMLHttpRequest* para obtener información de un servidor Web de un modo dinámico (asíncronamente). La posibilidad de utilizar *XMLHttpRequest* no es algo novedoso, puede hacerse desde 1998, e incluso algunos navegadores como Internet Explorer 4 ofrecían los medios para utilizar AJAX desde mucho antes. Incluso antes de que existiera la mayoría de los navegadores que se usan actualmente, ya se podía utilizar JavaScript para gestionar instantáneamente las peticiones al servidor. Sin embargo el uso generalizado del concepto AJAX es relativamente reciente.

Cómo trabaja AJAX

AJAX permite una nueva manera de navegar en una página Web. La manera clásica de navegación fue superada por otra forma más interactiva, más rápida y que permite una mejor experiencia de usuario.



Modelo tradicional de navegación Web.

El modelo tradicional o clásico de “experiencia de usuario” consiste en cargar una página, pulsar un enlace o realizar una acción que lleva a otra página, esperar la respuesta y así repetidamente. El inconveniente de este modelo es que se recarga la misma página una y otra vez donde un 90% de los elementos prácticamente permanecen sin cambios, siguiendo el siguiente proceso:

- ✦ El usuario realiza una acción (ej. Enviar formulario).
- ✦ La cabecera http de la página carga otra nueva que tendremos que cargar entera para ver un resultado (ej. Información correctamente enviada)
- ✦ Si estaba trabajando en la página anterior, se tendrá que volver al principio y cargar de nuevo toda la página.

Modelo clásico de aplicación Web (síncrono)

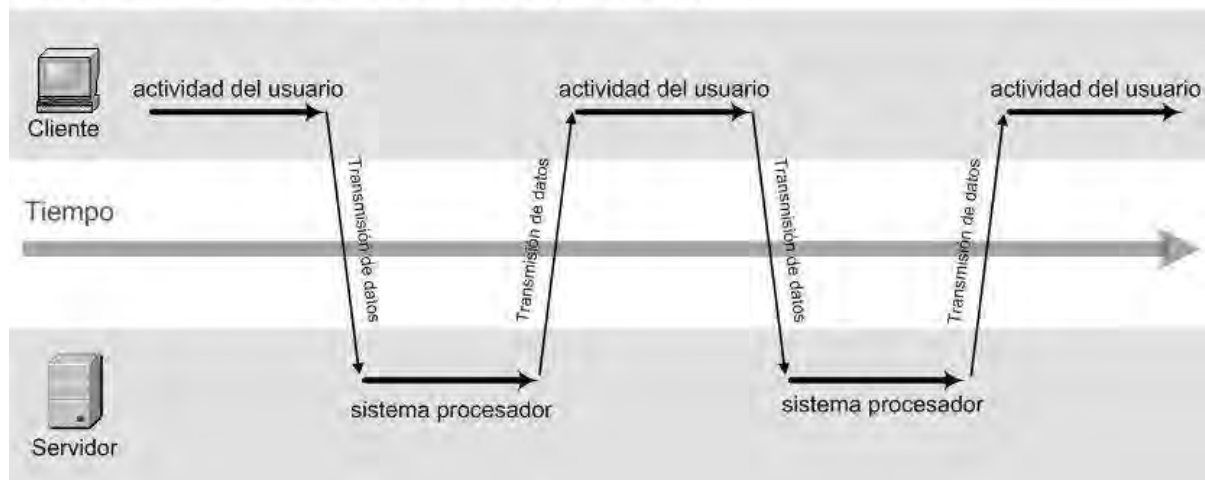


Figura 2.4.1 Modelo clásico de navegación por Internet

De la figura 2.4.1 se observa cómo la navegación del usuario se interrumpe cada vez que la página necesita conectar con el servidor para obtener nuevos datos.



Modelo AJAX. El modelo de navegación AJAX se basa en que las conexiones al servidor ocurren en segundo plano, y no necesitan cambiar de página, con AJAX se puede seguir trabajando en la página Web mientras la petición de información se gestiona en segundo plano.

Modelo Ajax de aplicaciones Web (asíncrono)

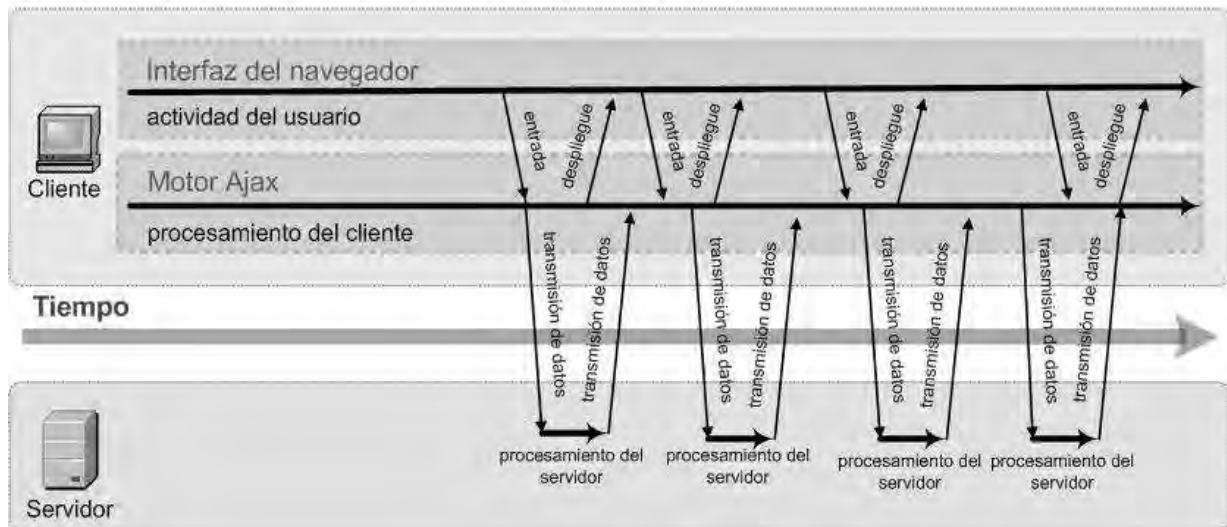


Figura 2.4.2 Modelo navegación AJAX

De la figura 2.4.2 se puede apreciar como las peticiones al servidor son paralelas al tiempo y a la experiencia del usuario, de aquí la cuestión asíncrona, es decir, no existe una espera en que una operación termine para que suceda la siguiente.

Se pueden hacer muchas llamadas y de acuerdo al tiempo de respuesta se irán ejecutando. De acuerdo al modelo clásico una página carga todas las peticiones, en cambio con el modelo AJAX cada petición es tratada de manera individual siguiendo su propio ciclo de vida.

Ciclo de vida de una petición AJAX:

- ✦ Acción: El usuario inserta un texto en un formulario o define unos parámetros de búsqueda.



- ✦ Evento: Es el disparador de la acción como pulsar un botón de formulario, pulsar una tecla, hacer clic en un enlace, etc.
- ✦ Captura: El evento llama al objeto *XMLHttpRequest* que realiza una petición al servidor.
- ✦ Servidor. El servidor ejecuta la acción con los parámetros que recibe del navegador y genera un archivo de respuesta (como lo hace en el modelo clásico de navegación).
- ✦ Filtrado. La respuesta es enviada al navegador cliente donde se procesa al llegar.
- ✦ Reacción. El navegador utiliza los datos recibidos para realizar alguna acción en la página.

Como se menciona al inicio AJAX no es una nueva tecnología sino la utilización de varias tecnologías ya existentes, de una manera coordinada. Se ejecuta en el cliente, es decir, en el navegador del usuario mientras este mantiene una comunicación asíncrona con el servidor en segundo plano.

Las tecnologías que forman AJAX son la mayoría de las tecnologías disponibles para páginas Web (Véase Figura 2.4.3):

- ✦ XHTML y CSS, para crear una presentación basada en estándares de todos los elementos de la aplicación y la información recibida por el servidor.
- ✦ DOM, para la interacción y manipulación dinámica de la presentación.
- ✦ XML, XSLT y JSON, para el intercambio y la manipulación de información, específicamente XML es el formato de los datos transmitidos del servidor al cliente (navegador) y que posteriormente serán mostrados.
- ✦ XMLHttpRequest, para el intercambio asíncrono de información.
- ✦ JavaScript, Para manejar el objeto XMLHttpRequest y para unir todas las demás tecnologías.
- ✦ Lenguaje de servidor, Genera la información útil en XML y la envía al navegador, puede ser PHP o ASP.

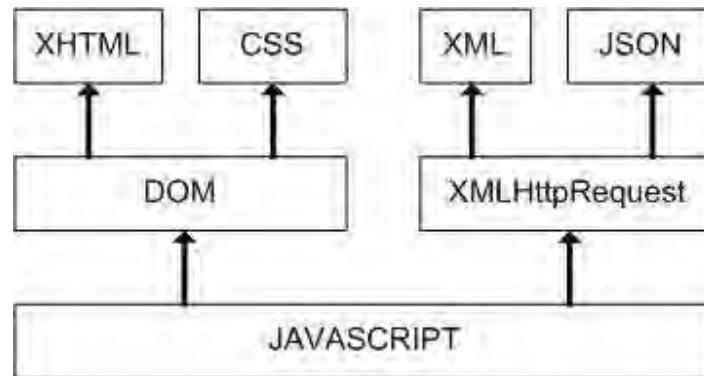


Figura 2.4.3 Tecnologías que forman AJAX

La comunicación asíncrona con el servidor, se logra mediante JavaScript al cargar las peticiones en segundo plano sin intervenir con la visualización de la página mediante el objeto *XMLHttpRequest* que está prácticamente disponible en la mayoría de los navegadores. Como AJAX está basado en estándares abiertos es una técnica válida para múltiples plataformas y sistemas operativos.

Para comprender bien el funcionamiento de AJAX es necesario primero conocer algunos conceptos básicos de las aplicaciones Web, como el procesamiento de las peticiones de los sitios Web y los tipos de respuesta que reciben del servidor.

El estándar que utilizan actualmente los navegadores para solicitar información a los navegadores Web se denomina HTTP (Hipertext Transfer Protocol). Este es el método que utilizan los navegadores Web para enviar una petición desde un sitio Web y a continuación recibir la respuesta del servidor en el que está alojado ese sitio. Al enviar una petición se incluye junto a ésta diversos encabezados que permiten al servidor saber exactamente que información debe servir y como gestionar la petición. Una vez recibida la petición, el servidor decide cómo debe contestar. Existen diversos códigos de respuesta (ver Figura 2.1.2.3). También existen varios métodos para realizar una petición, en la tabla 2.4.1 se muestran todos los métodos de petición disponibles (aunque generalmente se usan GET y POST).



Método	Descripción
GET	Obtener, Recuperar. Codifica el mensaje. Lo envía dentro de un string de consulta (query string) La información se añade a la URL del recurso del servidor simplemente solicita un recurso determinado al servidor. es la forma más frecuente de realizar una solicitud.
POST	Permite realizar la petición junto con los datos introducidos por el usuario (ideal para formularios Web). Sube datos al Servidor. Envía su petición en el cuerpo del mensaje (message body). Tampoco lo encripta.
HEAD	Es similar a GET, salvo porque la respuesta se enviará sin el cuerpo; es útil para obtener de forma rápida encabezados.
DELETE	Solicita la eliminación de un documento concreto.
TRACE	Devuelve una copia de la petición para poder controlar su progreso.
OPTIONS	Devuelve la lista completa de los métodos admitidos; es útil para averiguar qué métodos admite el servidor.
CONNECT	Es una petición de tipo Proxy que se utiliza en comunicaciones cifradas por túnel SSL.

Tabla 2.4.1 Métodos de peticiones HTTP

El funcionamiento del objeto XMLHttpRequest es el motor de AJAX para que pueda llevara cabo sus funciones en segundo plano.



El objeto XMLHttpRequest (XHR)

Antes de AJAX se usaba IFRAME para incrustar código de otra página o llamadas al servidor dentro del documento, pero con un IFRAME se carga una página Web entera.

El objeto XMLHttpRequest (Figura 2.4.4) que solucionaría este problema, al principio apareció como una creación de Microsoft.

Era un objeto ActiveX diseñado para poder usarse con Outlook. Entonces se llamó XMLHTTP, más adelante otros navegadores lo fueron incorporando, hasta que el consorcio W3C lo incluyó en sus estándares como XMLHttpRequest.

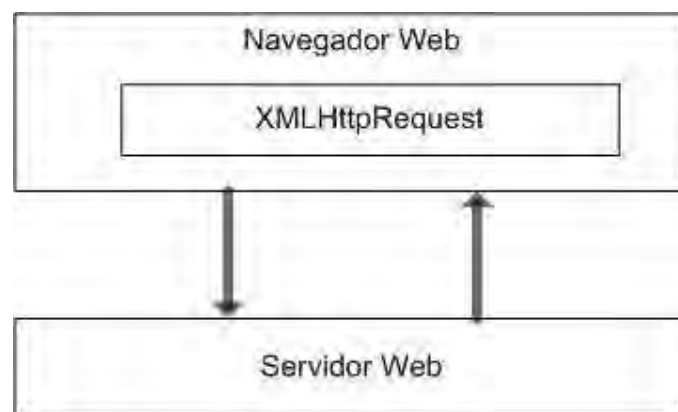


Figura 2.4.4 Objeto XMLHttpRequest

Esquema Cliente-Servidor, con instancia XMLHttpRequest

Los formatos que tienen los datos intercambiados entre el servidor Web puede ser XML o texto plano. Los datos devueltos por la llamada a XMLHttpRequest serán, por lo general obtenidos desde bases de datos en el servidor o un proceso de algún sistema Web (PHP, JSP, ASP, etc.)



Métodos del objeto XMLHttpRequest

- ✦ *setRequestHeader()*. Añade un par etiqueta/valor al encabezado en el momento de enviarlo.
- ✦ *send()*. Enviar la petición actual.
- ✦ *abort()*. Detiene la petición. Es útil cuando se quiere que una petición no dure más de un cierto periodo de tiempo, se puede aplicar para interrumpirla prematuramente.
- ✦ *getAllResponseHeader()*. Este método se puede usar para obtener la información completa sobre los encabezados http utilizados.
- ✦ *getResponseHeader* (“nombreEncabezado”). Puede utilizarse este método para obtener el contenido de un elemento concreto del encabezado. Útil para extraer los datos de la cadena, que contiene todos los encabezados. Por ejemplo, para conocer el tamaño del documento solicitado (*getResponseHeader*(“Content-Length”).
- ✦ *Open* (“método”, “URL”, “async”, “nombreUsuario”, “clave”). Es el método más importante para utilizar el objeto XMLHttpRequest. Se utiliza para establecer una conexión con un archivo concreto del servidor. No todos los argumentos de esta función son obligatorios y se usan dependiendo de cada situación.
- ✦ *setRequestHeader* (“etiqueta”, “valor”). Con este método se puede incluir en el encabezado de la petición un par etiqueta/valor. Se debe utilizar este método después de haber ejecutado el método *open()*, y siempre antes del método *send()*.
- ✦ *Send* (“contenido”). Es el método responsable de enviar realmente la petición al servidor. Si esta petición se envía de forma asíncrona, la respuesta se obtendrá de forma inmediata; en caso contrario no se obtendrá la respuesta hasta que haya sido completamente recibida.

Y en la Tabla 2.4.2 se muestra una tabla de las propiedades del objeto XMLHttpRequest.



Objeto XMLHttpRequest	
Propiedad	Descripción
onreadystatechange	Se utiliza como gestor de eventos para detectar cambios de estado.
readyState	Contiene el estado actual del objeto. 0: sin inicializar 1: cargándose 2: cargado 3: en interacción 4: completado
responseText	Contiene la respuesta en formato de cadena.
responseXML	Contiene la respuesta en formato XML.
Status	Contiene el estado de la petición en formato numérico.
statusText	Contiene el estado de la petición, pero en formato de cadena. (ej. Error 404 devolverá "Not found")
onreadystatechange	Es un gestor de eventos que nos permite ejecutar cierto bloque de código, o cierta función, cuando se produce algún cambio en el estado del avance del proceso.
readyState.	Esta propiedad indica con precisión en qué estado se encuentra actualmente la petición. Puede usarse cuando se desea ejecutar cierto bloque de código cuando la petición se encuentra en estado de carga y detenerlo cuando se alcanza el estado de completada.
responseText	Esta propiedad se devuelve una vez completada la petición.
responseXML	Es similar a la anterior pero más apropiada cuando se tiene la certeza de que la respuesta vendrá en formato XML, especialmente si se tiene pensado utilizar la funcionalidad del navegador para gestionar el formato XML.
status	Contiene el código de respuesta generado por el servidor ante la petición.
statusText	Es una variante textual de la propiedad status.

Tabla 2.4.2. Propiedades del objeto XMLHttpRequest.

La transferencia de datos entre cliente y servidor involucra una actualización de las páginas completas en cada solicitud, eso significa una gran cantidad de datos desde el



servidor y cliente y el refresco del navegador de todo el contenido que se muestra en pantalla en ese momento.

Para solucionar esto los *applets* de Java y programas Flash proporcionan una mayor interactividad sin comunicaciones continuas con el servidor y transferencias de páginas completas. La contraparte de esto es que se necesita la instalación de complementos en el navegador aparte de no ser basados en estándares y tener algunos inconvenientes relacionados con la accesibilidad.

Es aquí donde entra AJAX, es independiente del servidor, no es necesario instalar complementos en el lado del cliente, está basado en estándares y no es necesario transferir ni actualizar páginas completas.

Las interfaces que se pueden crear se comportan como elementos dinámicos que se comunican esporádicamente con el servidor para obtener porciones de datos específicos y no páginas completas.

Programación AJAX frente a los programas nativos

Ventajas

- ✦ No se necesita instalar una aplicación AJAX, solo es necesario introducir la URL en el navegador y puede comenzarse a trabajar.
- ✦ Puede accederse a la última versión de la aplicación con solo actualizarla en el servidor Web; ya no es necesario actualizaciones individuales en los clientes.
- ✦ Es independiente del sistema operativo, solo es necesario un navegador.
- ✦ La curva de aprendizaje del uso de una aplicación AJAX es bastante rápida.
- ✦ Una aplicación AJAX utiliza menos recursos, espacio en disco y memoria que un programa nativo.
- ✦ La migración de una aplicación AJAX a otras arquitecturas de hardware y software es más sencillo que el caso contrario.



Desventajas

- ✦ Las aplicaciones AJAX, en comparación con los programas nativos, tiene acceso limitado a los recursos locales de la computadora.
- ✦ Las aplicaciones AJAX no pueden acceder a bases de datos locales como lo hacen los programas nativos sino que debe transferir la información a través de redes generalmente más lentas.

Comparación frente a las aplicaciones Web clásicas

Ventajas

- ✦ Para las peticiones de los usuarios una aplicación AJAX no recarga la página Web completas sino solo las partes de la página que actúan como interfaz.
- ✦ En una petición al servidor solo se solicitarán los datos necesarios y no una nueva página completa, siendo más ágil que una aplicación Web clásica.
- ✦ No es necesario instalar complementos ni descargar programas pesados en el cliente como *applets* Java o Flash.
- ✦ La interacción del usuario con una aplicación AJAX es muy parecida a la que ocurre con un programa nativo.

Desventajas

- ✦ El navegador no muestra visual y automáticamente cuando la aplicación AJAX está esperando la recepción de una nueva página.
- ✦ No hay integración con los botones de avanzar y retroceder del navegador (aunque pueden solucionar en las aplicaciones AJAX preparando código adecuado para ello).
- ✦ Si el usuario desactiva el motor JavaScript del navegador las aplicaciones AJAX no funcionarán.
- ✦ Las aplicaciones AJAX son relativamente más complejas de desarrollar.



2.5 Metodologías y tecnologías similares a AJAX para el desarrollo de aplicaciones Web.

Aplicaciones RIA's

Las aplicaciones RIA (Rich Internet Applications, Aplicaciones de Internet enriquecidas) son aplicaciones Web de última generación con más ventajas que las tradicionales aplicaciones Web.

Estas surgen como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales ofreciendo una mejor experiencia de usuario.

Normalmente en las aplicaciones Web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces, a recargar la misma página con un mínimo cambio.

Otra de las desventajas de las tradicionales aplicaciones Web es la poca capacidad multimedia que posee, por ejemplo, para ver un vídeo es necesario usar un programa externo para su reproducción.

Las capacidades multimedia son totales gracias a que estos entornos tienen reproductores internos y no hace falta ningún reproductor del Sistema Operativo del usuario.

Existen muchas soluciones para crear RIAs, una técnica muy conocida es AJAX, sin embargo existen hoy en día más herramientas para la creación de entornos RIA. Entre estas se puede mencionar las plataformas Adobe Flash, Adobe Flex y Adobe AIR de Adobe, AJAX, OpenLaszlo, Silverlight de Microsoft, JavaFX Script de Sun Microsystems, Bindows y Javascript.



Características:

A pesar de que el desarrollo de aplicaciones multimedia para navegadores Web está mucho más limitado y es más difícil que otro tipo de aplicaciones de escritorio, los esfuerzos se justifican por varios motivos:

- ⊕ No necesitan instalación (solo es necesario mantener actualizado el navegador Web).
- ⊕ Las actualizaciones hacia nuevas versiones son automáticas.
- ⊕ Se pueden utilizar desde cualquier ordenador con una conexión a Internet sin depender del sistema operativo que este utilice.
- ⊕ Generalmente es menos probable la infección por virus, que utilizando por ejemplo programas ejecutables.
- ⊕ Más capacidad de respuesta, ya que el usuario interactúa directamente con el servidor, sin necesidad de recargar la página.
- ⊕ Ofrecen aplicaciones interactivas que no se pueden obtener utilizando solo HTML, incluyendo arrastrar y pegar, cálculos en el lado del cliente sin la necesidad de enviar la información al servidor.
- ⊕ Evita la problemática del uso de diferentes navegadores al abstraerse de ellos a través de un framework.
- ⊕ Reducen costos de desarrollo y añaden valor a la empresa.
- ⊕ Son más fáciles de mantener.
- ⊕ Tienen un alcance mayor y se pueden ampliar más fácilmente
- ⊕ El usuario interactúa en todo momento con la aplicación.
- ⊕ Reduce los procesos y tiempos de ejecución.
- ⊕ Rendimiento: puede ser mejorado dependiendo de la aplicación y de las características de la red. Aplicaciones que pueden procesar localmente en el cliente evitan viajes hacia el servidor lo que incrementa su rendimiento. Dándole más trabajo al cliente también se puede incrementar el rendimiento del servidor. Por el contrario los recursos necesarios hacen difícil que las aplicaciones puedan ejecutarse en dispositivos pequeños, móviles o embebidos



Debido al gran crecimiento de este tipo de aplicaciones, hoy en día existen, como se menciono anteriormente, una gran variedad de herramientas que disputan este mercado, entre ellas se mencionan las siguientes:

2.5.1 Silverlight

Silverlight es un producto desarrollado por Microsoft, que permite crear aplicaciones RIA.

Características

Desde un punto de vista técnico, no es más que un complemento que se instala en nuestro equipo (ocupa menos de 5MB) y hacen uso de él todos los navegadores. Actualmente, los sistemas operativos soportados oficialmente por parte de Microsoft son:

- ⊕ Windows 2000 y posteriores
- ⊕ Mac OS 10.4 y 10.5

Y para futuras versiones: Nokia S60, Windows Mobile, Mobin, etc.

Por otra parte la lista de navegadores soportados es:

- ⊕ Internet Explorer 6 y posteriores
- ⊕ Firefox
- ⊕ SeaMonkey
- ⊕ Safari
- ⊕ Opera
- ⊕ Google Chrome

El mayor potencial de Silverlight está en el entorno de desarrollo, Silverlight incluye un subconjunto de la plataforma .NET de Microsoft y con ello toda la potencia que ello implica. Como entorno preferido para el desarrollo, bajo Windows, tenemos Visual



Studio (desde la versión gratuita, la Express, hasta el entorno de gestión de ciclo de vida Team System). Además para otros entornos, Mac y Linux, tenemos MonoDevelop y Eclipse (instalando las recientes Eclipse Tools For Silverlight).

Microsoft ha intentado enfocarlo hacia aplicaciones empresariales, juegos casuales y contenido multimedia.

2.5.2 OpenLaszlo

OpenLaszlo es una plataforma open source de aplicaciones ricas de Internet (RIA), que fue fundado en el año 2000 como Laszlo Systems, pero a mediados del año 2004 fue Open Source.

La plataforma consta de dos piezas: una declarativa XML basada en un lenguaje llamado LZX que es usado para escribir aplicaciones clientes, y una aplicación servidor (y LZX Compilador) llamado el servidor presentador.

OpenLaszlo es una plataforma de código abierto para el desarrollo y distribución de RIA's. Ha sido publicado bajo la licencia Common Public License, certificada por la *Open Source Initiative*. La plataforma OpenLaszlo consiste en el lenguaje de programación LZX y el Servidor OpenLaszlo:

Ventajas

- ✦ En cuanto a Usabilidad, proporciona al desarrollador la oportunidad de implementar una aplicación que le sea muy sencilla de usar al usuario sin necesidad de formación y asistencia.
- ✦ No es necesario la carga completa de una página para hacer un pequeño cambio, se hace de manera inmediata y en el área que se desea se realiza el cambio.



Desventajas

- ✦ Openlaszlo sólo se muestra a través de Flash, por ende, afecta al rendimiento al lado del cliente – si este no cuenta con una conexión de banda ancha-, debido a que Flash originalmente fue diseñado para presentar películas o animaciones, pero no fue pensado para realizar cálculos numéricos o manipulaciones de datos.
- ✦ Lo otro es por el lenguaje a usar “LZX”, es aventurarse a programar en este lenguaje ya que implica implementar librerías, debido a que existen pocas.

2.5.3 Mozilla Prism

Mozilla Prism, anteriormente conocido como WebRunner, es un producto de Mozilla actualmente en desarrollo que integra aplicaciones Web con el entorno de escritorio, lo que permite a las aplicaciones Web ejecutarse desde el escritorio y configurarse independientemente del navegador Web por defecto.

El software se encuentra actualmente en fase beta.

Mozilla Prism fue la primera de estas aplicaciones en llegar al gran público, es la apuesta de Mozilla. Estas son las principales cosas a tener en cuenta:

- ✦ Es multiplataforma: Windows, Linux y Mac funcionan perfectamente.
- ✦ Su motor interno es el de Firefox, por tanto la compatibilidad con casi todas las páginas está garantizada. Por lo contrario, las extensiones que se tengan que usar no funcionarán de manera directa, y aunque es posible hacerlas funcionar, es algo un tanto laborioso.
- ✦ Existe en dos versiones: como extensión para Firefox o como aplicación independiente. Las dos maneras funcionan de forma similar, con las mismas opciones.
- ✦ Posibilidad de ocultar o no la barra de navegación.
- ✦ Permite elegir cualquier imagen como icono de la aplicación que se creará.



- ✦ Tiene soporte para notificaciones, como por ejemplo cuando llegue un email; también nos puede avisar en la barra de tareas/Dock/bandeja de notificación cambiando el icono para mostrar el número de elementos sin leer.
- ✦ Integración con la bandeja de notificación, a la que podemos minimizar la aplicación.

2.5.4 Adobe AIR.

Adobe AIR es una plataforma que permite crear aplicaciones que funcionen a la vez en múltiples Sistemas Operativos. Se piensa en ello como una página Web: se verá en Windows, Mac, Linux, etc. Adobe AIR es una especie de plugin que se instala en la computadora, de una manera similar al plugin de Flash del mismo Adobe.

Una vez instalado se pueden descargar aplicaciones en formato .air que se instalarán de una manera sencilla y parecida a los típicos .exe.

Estas aplicaciones .air sirven para cualquier sistema operativo que tenga AIR instalado, de tal manera que creándolas una sola vez nos sirve para multitud de equipos.

Ventajas

- ✦ Además de ser multiplataforma “por defecto”, las aplicaciones en AIR se desarrollan muy rápidamente y de manera muy sencilla. Esto se debe principalmente a que está basado en HTML/Javascript/Flash, por tanto si se tiene experiencia en el desarrollo de páginas Web resultará muy simple crear una aplicación en AIR.
- ✦ Respecto a una página Web, al ser una aplicación “nativa” no se necesitará tener un navegador abierto, se integrará en el sistema como una aplicación más y se podrá trabajar en local sin tener una conexión a Internet.



Desventajas

- ✦ Como no es una aplicación nativa de verdad, no son muy eficientes y puede tener peor estabilidad y mayores gastos de recursos que una nativa. Además, la integración no es tan buena como las nativas 100%, y aunque el aspecto suele ser espectacular no está muy a la par con el sistema en general.

2.5.5 JavaFX

JavaFX es una familia de productos y tecnologías de Sun Microsystems, anunciados en la conferencia de desarrolladores JavaOne en Mayo de 2007.

Los productos JavaFX se han pensado para ser usados en la creación de Rich Internet Applications (RIAs). Actualmente JavaFX se compone de JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX planeados.

JavaFX se ha creado para competir en el espacio que ya ocupan Flash, de Adobe, y Silverlight, de Microsoft.

Ventajas

- ✦ JavaFX es una extensión a la plataforma Java que ofrece interactividad, animación y programación compatible con AJAX, Flash de Adobe y Silverlight de Microsoft. A diferencia de éstas anteriores, utiliza los runtimes Java ya instalados en el cliente local en lugar de JavaScript.
- ✦ JavaFX permite a los creadores de contenido crear contenido rico en medios, lo que incluye el arrastrar y soltar contenido de equipos de escritorio o móviles hacia el escritorio, algo que no es posible en ninguna otra RIA.
- ✦ JavaFX se trata de un sistema de desarrollo para la Web que pretende competir con el Flash de Adobe y el Silverlight de Microsoft, con el añadido de que es Open Source.



- ✦ Millones de desarrolladores, muchos equipos con Java instalado y a eso hay que sumar que Java es libre bajo GPL.
- ✦ Multiplataforma, desde el escritorio (corriendo sobre una máquina virtual Java SE) hasta los dispositivos móviles para los que habrá un JavaFX Mobile.
- ✦ Utiliza el mismo lenguaje para la Web, para el escritorio y para la telefonía móvil.
- ✦ Tiene como objetivo que el navegador opere mejor que con JavaScript, que es lento y se implementa en formas diferentes de un navegador a otro.

Desventajas

- ✦ Con JavaFx, se agrega un lenguaje de script complejo, que necesita mantener en memoria su propia estructura semántica y sus propios objetos de plataforma, simultáneamente con los objetos gráficos.
- ✦ Al ser un lenguaje de script agrega un factor de ralentización importantísimo, por eso el desempeño no es óptimo.
- ✦ Por otro lado, las APIs de tratamiento de XML ya vienen incorporadas al JDK, generalmente están implementadas a bajo nivel, por lo que un sistema de templates es mucho mas liviano que soportar JavaFX.

Finalmente después de analizar las tecnologías similares a AJAX podemos observar la gran ventaja sobre las anteriores descritas ya que AJAX en nativo en los “navegadores Web” es el único “RIA Framework” que puede ser encontrado por los diferentes motores de búsqueda.



CAPÍTULO 3

Análisis y Planteamiento del problema





CAPÍTULO 3 ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA

3.1 Justificación de las herramientas elegidas

Después de conocer algunas de las características, ventajas y desventajas de las herramientas que se usarán, a continuación se analizarán desde el punto de vista comparativo con otras herramientas similares que permitirá justificar la elección.

Debido a que la aplicación será basada en Web, como se mencionó en la sección 2.1.2 del capítulo 2 acerca de las aplicaciones Web, es necesario justificar las herramientas elegidas para cada elemento de la aplicación como son:

- ✦ El servidor Web.
- ✦ El DBMS.
- ✦ Los lenguajes de programación tanto del lado del servidor como del lado del cliente.
- ✦ Los navegadores para soportar la aplicación.

Antes de comenzar es importante establecer que para el desarrollo de una aplicación es necesario hacerlo desde dos perspectivas diferentes: del lado del Front-End y del lado del Back-End; que básicamente serán un tipo de abstracciones que ayudarán a mantener las diferentes partes del sistema en forma separada.

A continuación se definirán ambos términos desde un contexto de desarrollo de software convencional:

- ✦ Front-End. Es la parte del software que interactúa con el o los usuarios, es decir, es una interfaz entre el usuario y el Back-End, cuya tarea es la de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas y procesarlas de una manera conforme a la especificación que el Back-End pueda usar.



Esta interfaz involucra el código más cercano al usuario, por lo general, consiste en una mezcla de HTML, Javascript, CSS, Flash, y el código de varios lenguajes del lado del servidor como PHP, ASP.Net, ASP clásico, etc.

- ✦ **Back-End.** Es la parte que procesa la entrada desde el Front-End. Aquí implica lo que son las bases de datos y los SGBD tales como Oracle, MS-SQL, MySQL, etc.

Es importante señalar que en un contexto de desarrollo Web actual, los términos se han aplicado con un enfoque de CMS (**Content Management System**, Sistema Gestor de Contenido) que se está extendiendo rápidamente.

Los CMS son aplicaciones que permite crear una estructura de soporte para la creación y administración de contenidos, por parte de los participantes, principalmente en las páginas Web que controla una o varias bases de datos donde se aloja el contenido del sitio. Es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página Web.

Con base en este último contexto el *Front-End* es la visualización del usuario navegante, es decir, la parte del sitio que interactúa directamente con el usuario. Y el *Back-End* el área de administración en donde se ingresa, edita y organiza el contenido, es la parte del administrador del sitio con sus respectivos sistemas.

Tomando en cuenta el primer enfoque de desarrollo de software, el cual se utilizará para el desarrollo del presente proyecto, se hará una comparación de las herramientas a utilizar tanto para el Back-End como el Front-End.



3.1.1 Herramientas para el Front-End

Con base en el capítulo anterior, las principales herramientas que se utilizarán para el Front-End son los lenguajes JavaScript y PHP. El lenguaje que se ejecutará en el lado del cliente es JavaScript. JavaScript no es la única alternativa para obtener las mismas funcionalidades requeridas. Su principal contendiente es VBScript. Analizaremos sus principales características así como sus principales desventajas frente a JavaScript.

- ✦ **VBScript.** Debido a la naturaleza estática de los documentos HTML, los lenguajes de programación han sido diseñados para introducir un cierto nivel de interactividad en las páginas Web. VBScript es la solución de Microsoft para el problema, es una versión reducida a escala de Visual Basic para su uso con páginas Web y otras aplicaciones que utiliza controles Microsoft Active X. VBScript es muy similar en uso y sintaxis a JavaScript, excepto que se basa en un subconjunto de la versión de Microsoft Visual al lenguaje Basic, si bien no ofrece la funcionalidad de Visual Basic, proporciona una forma fácil de aprender una herramienta poderosa que puede ser utilizado para agregar interacción con las páginas Web.
- ✦ **JavaScript.** Como se mencionó en el capítulo anterior también permite insertar código especial dentro del HTML de una página, su función es ampliar las posibilidades de HTML, JavaScript no crea programas independientes, dependen por completo del código HTML de la página.

Es un lenguaje basado en objetos sin tipo y liviano utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador Web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas. Es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos.



Además una de las ventajas fundamentales de JavaScript es que su aprendizaje y uso son muy sencillos además de que permite realizar labores complejas en una página.

Comparación de JavaScript y VBScript

Similitudes:

- ✦ Ambos lenguajes son fáciles de aprender y no requieren costosas herramientas de desarrollo.
- ✦ Ambos pueden ser utilizados para mejorar las páginas Web.
- ✦ Se ejecutan en equipos cliente y puede sustituir a los programas CGI para reducir las cargas del servidor.
- ✦ En ambos se puede ejecutar scripts maliciosos en las máquinas clientes.

Diferencias

- ✦ JavaScript es el lenguaje de scripts por defecto para los navegadores, pero VBScript debe ser especificado como el lenguaje de scripting.
- ✦ JavaScript es soportado por todos los navegadores populares, mientras que VBScript es compatible con MS Internet Explorer solamente. Por lo tanto los desarrolladores en VBScript perderían una audiencia considerable.
- ✦ Una de las cuestiones más importantes con JavaScript es que había diferentes versiones de lenguaje desde su inicio (versión 1.0). Del mismo modo, diferentes versiones de los navegadores existentes en las máquinas de los usuarios. Por lo tanto, el código escrito para una versión podría no funcionar en otro.
- ✦ JavaScript es sensible a mayúsculas, en cambio VBScript no lo es por tanto no sería propenso a errores de sintaxis.
- ✦ JavaScript se utiliza los mismos caracteres para la concatenación como lo hace para la suma (el signo +), mientras que el carácter de concatenación '&' se utiliza en VBScript. Esta es otra fuente de errores en JavaScript.

Acerca de los lenguajes de programación del lado del servidor existe una amplia gama de posibilidades: ASP.Net, ASP clásico y PHP. En la tabla 3.1.1 se muestra una tabla



comparativa de las principales características de los dos lenguajes más usados en el mercado.

Es importante resaltar que ASP es considerado como una tecnología de aplicación Web que funciona solamente para una infraestructura Microsoft y no es considerado meramente como un lenguaje de programación.

Característica	PHP	ASP
Seguridad	Instalado sobre servidores Unix o Linux, es más veloz y seguro. PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje seguro dependiendo de las necesidades.	Debido a que solo funciona bajo Windows, es más vulnerable a sufrir caídas e infecciones de virus
Rendimiento	En el caso de estar montado sobre un servidor Unix o Linux, es más rápido dado que se ejecuta en un único espacio de memoria.	Son demasiadas las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP. Consumiendo así más recursos.
Plataforma	Independiente de la plataforma y servidor (Unix, Windows, Linux)	Se trata de un sistema propietario que es usado nativamente sólo por Microsoft Internet Information Server (IIS). Esto limita su disponibilidad a servidores basados en Win32.
SopORTE	Recibe contribuciones de diversos desarrolladores	ASP es específica de Microsoft que desarrolla sus procesos internamente.

Tabla 3.1.1 Comparación entre PHP vs ASP



3.1.2 Herramientas para el Back-End

Las herramientas que se utilizarán para el Back-End son las que tienen que ver con las bases de datos y los SGBD, para el presente trabajo se utilizará un SGBD o DBMS llamado MySQL.

Existen varias alternativas DBMS que pueden emplearse para la administración de las bases de datos.

DBMS (o SGBD)

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfase entre la base de datos, el usuario y las aplicaciones que la utilizan.

Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se menciona los términos SGBD y DBMS, siendo ambos equivalentes y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, por sus siglas en inglés.

Entre los DBMS que no son gratuitos están:

- ✦ **Oracle.** Una de las primeras BD robusta que salió al mercado y se ha mantenido por lo que tiene mucha experiencia en ese campo. Multiplataforma ya que se ejecuta bien en Unix, Linux, Windows, etc.
- ✦ **DB2.** Es de IBM pero por lo investigado y visto, es bastante buena y esta para varias plataformas como Windows, Linux, Unix, etc.



- ✦ **SQL Server.** Existen quienes no le consideran una BD muy depurada, además es plataforma propietaria, es decir, puede usarse obligatoriamente sobre Windows, se puede descargar una versión de prueba por un periodo muy corto y todo se ejecuta en modo gráfico, es decir, solo se tiene que aprender a utilizar su entorno gráfico para usarlo por lo que no es la forma correcta de aprender.

Y dentro de los que son libres están:

- ✦ **MySQL.** Es una de las más usadas en sitios Web (más que todo) por que es muy liviana y rápida en sus accesos.
- ✦ **PostgreSQL.** Es un poco más robusta, permite procedimientos almacenados y otras funciones, por lo que es una BD libre bastante completa, pero es un poco lenta en su primer ingreso.
- ✦ **FireBird.** Funciones similares a PostgreSQL.

Para los propósitos del presente trabajo es necesario el ahorro en lo posible en la inversión de las herramientas a emplear, las opciones que son de paga serán descartadas y por lo tanto se procederá al análisis comparativo de las opciones gratuitas que nos ofrezcan una solución óptima.

MySQL

Este DBMS fue tratado en el capítulo anterior, por lo que solo se presentarán sus características principales.

MySQL: Su principal objetivo de diseño fue la velocidad. Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria.

Ventajas:

- ✦ Licencia dual, por una parte se puede conseguir bajo un esquema de código libre (GPL), pero también ofrece una licencia comercial.



- ✦ Funciona en diferentes plataformas.
- ⊕ Se compone de un servidor que es multihilo y multiusuario, lo cual quiere decir que nos permite llevar a cabo varios trabajos en un solo momento.

Desventajas:

- ⊕ No es intuitivo, como otros programas (ACCESS).
- ✦ No se debe considerar a este manejador para el uso con grandes bases de datos, a las que se acceda continuamente.

PostgreSQL Es otra alternativa en los DBMS de la misma modalidad de software Open Source, intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase

FireBird. Es un sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++. El proyecto se desarrolla activamente, el 18 de abril de 2008 fue liberada la versión 2.1 y el 26 de diciembre de 2009 fue liberada la versión 2.5.0 RC1.

Características de FireBird

- ✦ Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- ✦ Ejecutable pequeño, con requerimientos de hardware bajos.
- ✦ Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- ✦ Soporte de transacciones ACID y claves foráneas.
- ✦ Es medianamente escalable.
- ✦ Buena seguridad basada en usuarios/roles.



En la tabla 3.1.2 y 3.1.3 se muestra una tabla comparativa de las ventajas y desventajas de estos tres principales DBMS.

Ventajas		
PostGreSQL	MySQL	FireBird
Gran escalabilidad. Capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).	Lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento	Ofrece una gran variedad de medios para acceder a los datos (nativo/API, driver dbExpress, ODBC, OLEDB, .NET provider, driver JDBC nativo de tipo 4, módulo para Python, PHP, Perl, etc.)
Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.	Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.	Otras peculiaridades resaltables son la creación de copias de seguridad y la funcionalidad en arquitecturas de 64 bits.
Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.	Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.	Firebird administra la concurrencia en las actualizaciones en los datos, así como el manejo de las transacciones.
	Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.	Firebird ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos
	El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).	

Tabla 3.1.2. Comparación de las ventajas entre PostGreSQL, MySQL y FireBird



Desventajas		
PostGreSQL	MySQL	FireBird
Consume gran cantidad de recursos.	Carece de soporte para transacciones, rollback's y subconsultas.	Una desventaja de la BD es que ya fue descontinuada, y que solamente puede acceder a bases de datos de Firebird/InterBase 6 con el dialecto 1 de SQL.
Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.	El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí poseen esta característica.	Su uso requiere un alto conocimiento del lenguaje SQL.
Es de 2 a 3 veces más lento que MySQL.	No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.	

Tabla 3.1.3. Comparativa de las desventajas entre PostGreSQL, MySQL y FireBird

Para una elección adecuada del SGBD es importante tener en cuenta la compatibilidad y rendimiento. Como una herramienta adicional para el manejo de MySQL se usará phpMyadmin (Figura 3.1.1).

La herramienta phpMyAdmin es una herramienta escrita en PHP cuyo objetivo es manejar la administración de MySQL a través de páginas Web, utilizando Internet. Actualmente phpMyAdmin puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL.



Una herramienta similar a phpMyAdmin es phpPgAdmin, la cual ofrece funcionalidades similares pero está diseñado para PostgreSQL.



Figura 3.1.1 Interfaz de la herramienta phpMyadmin para el manejo de la administración de MySQL.

En cuanto al servidor Web, los más utilizados mundialmente son: el servidor Web Apache, IIS de Microsoft y el relativamente reciente servidor Web Cherokee.

- ✦ **Servidor Web IIS** (Internet Information Services). Es un servidor Web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.



- ✦ **Servidor http Apache.** Tiene amplia aceptación en la red, ya que es el más utilizado para la creación de sitios Web, debido a que brinda ventajas como: velocidad, seguridad, simplicidad y facilidad de desarrollo distribuido, además de ser rápido, estable y fácilmente adaptable a nuevas tecnologías y protocolos. Apache es un servidor altamente configurable ya que su arquitectura está formada por un diseño modular, además puede trabajar con la mayoría de lenguajes de programación como, PHP, JSP, etc. y otros lenguajes script.

Apache soporta HTTPS, haciendo de esta manera más seguro el servidor Web, ya que permite mantener un canal de comunicación encriptado utilizando SSL (Secure Socket Layers) con lo cual, no se permite el robo de datos ó información que se está transfiriendo entre los usuarios, puesto que, permite identificar verdaderamente al servidor y, en algunos casos, al cliente. Lo que lo hace aún mucho más seguro.

- ✦ **Servidor Web Cherokee.** El proyecto Cherokee surgió en el año 2001, por el desarrollador de software Libre Álvaro López, Cherokee es una tecnología multiplataforma de código abierto, es decir, totalmente gratuito debido a que es software libre, publicado bajo la licencia GPL (General Public License). Su principal característica es ser un servidor que posee algunas características de las cuales Apache carece, como son: mayor velocidad (cinco veces más rápido que Apache), flexibilidad (sistema de carga dinámica de módulos al igual que Apache) y la capacidad de incluirse dentro de otras aplicaciones.

Cherokee es un buen contendiente en la modalidad de software libre para Apache, sin embargo, Apache sigue presentando mejores características que Cherokee y IIS, por lo que se usará Apache como servidor Web.

Respecto al costo, al adquirir productos libres y de gran funcionalidad evitando así la compra de licencias por productos propietarios se logrará que la aplicación no resulte



costosa. A continuación se muestra una tabla comparativa de las principales características de los servidores Web mencionados. (Ver Tabla 3.1.4)

	Cherokee	Apache	IIS
Desarrollador	Cherokee Project	Apache Software Foundation	Microsoft
Sitio Web oficial	http://www.cherokee-project.com/	Apache.org	http://www.iis.net
Diseñado por	Álvaro López Ortega		Microsoft
Última versión estable	1.0.8 (info) 11 de agosto de 2010	2.2.17 19 de octubre de 2010	7.5
Género	Servidor Web	Servidor Web	Servidor Web
Escrito en	C	C	
Sistema operativo	Multiplataforma	Multiplataforma	Microsoft Windows
Licencia	Licencia Pública General de GNU	Licencia Apache	Software propietario
Idiomas	Plurilingüe1	Plurilingüe	Plurilingüe
Soporte de múltiples lenguajes de programación	SI	SI	POBRE
Módulos permanentes	SI	SI	SI
Conexiones seguras	SI	SI	SI
Mayor velocidad	SI	SI	NO
Mayor seguridad	POBRE	SI	POBRE
Soporta SSL	NO	SI	SI

Tabla 3.1.4 Comparativa de los principales servidores Web

Finalmente el conjunto de las herramientas que se utilizarán para la aplicación basada en AJAX tiene también un competidor que ha alcanzado un gran mercado, Flash. Sin embargo por ser un software propietario y por las características mostradas en la tabla de la Figura 3.1.5, se optará por una programación basada en AJAX.



Función	Flash	Ajax
Audio	Dinámicamente la carga de audio. Soporte flash integrado de audio.	Sólo se admite a través de plug-ins externos (como el Media Player).
Navegador de Integración	Flash Player plug-in. Flash se limita a un área rectangular predefinidos del navegador.	JavaScript de forma nativa con el apoyo de los navegadores modernos. Simple interacción con cualquier parte del navegador.
Problemas de compatibilidad	Puede haber variaciones menores entre las versiones de Flash.	Principales diferencias de compatibilidad entre versiones de navegadores.
CSS	Soporte limitado.	Soporte completo (según el navegador).
Dinámica generación de contenidos	Difícil. SWF es un formato de pre-compilados cerrado. (Comparación de SVG, el cual está basado en XML.)	HTML puede ser escrito usando casi cualquier tipo de tecnología de servidor. Es más fácil la generación de imágenes dinámicas.
Modelo de programación	Ofrece ActionScript 2.0, en Java, como marco sólido.	JavaScript 2.0 pero no con el apoyo de <i>cualquier</i> navegador principal. JS 1.5 no se recomienda para grandes aplicaciones de programación orientada a objetos.
De mapa de bits (mapa de bits) Gráficos	Carga estática de las imágenes de forma dinámica. Apoyo a JPG, GIF y PNG. La manipulación de mapa de bits.	Carga estática de las imágenes de forma dinámica.
Las expresiones regulares	No se admite de forma nativa por AS 2.0, pero de código abierto de soluciones están disponibles.	Soporte completo.
Integration Server	Muchas de las soluciones disponibles. Flash puede comunicarse con ASP, PHP, ASPX y muchos otros tipos de secuencias de comandos del servidor.	Limitada. Se pueden comunicar con el servidor de forma dinámica con truco IFRAME o Objeto XMLHttpRequest.
Texto	Texto API imita algunas funciones HTML.	Las potentes capacidades de diseño.
Vector Graphics	Soporte completo.	Ninguno.
Video	Carga dinámica de archivos de vídeo FLV o reproducción de videos incrustados.	Sólo se admite a través de plug-ins externos (como el Media Player). Múltiples Formatos de vídeo se pueden cargar.
XML	Soporte completo.	No se admite de forma nativa por JavaScript.

Tabla 3.1.5 Comparativa AJAX vs FLASH

Como se observa en la tabla de la figura 3.1.5 cada tecnología tiene sus ventajas y desventajas, no puede decirse que una es mejor que la otra ya que depende de la solución que está en desarrollo.



Pero hay un área donde AJAX simplemente es superior a Flash: aplicaciones pesadas de texto generadas dinámicamente y en el momento, estas dominan la Web.

Por tanto si se desea desarrollar un programa de correo electrónico, o un organizador personal en línea, o una tienda online, flash no tiene mucho sentido. Simplemente no da la misma flexibilidad para el análisis y la visualización de grandes cantidades de datos de texto. Sin necesidad de escribir demasiado código, simplemente no hay suficiente flexibilidad para reposicionar o cambiar el tamaño campos de texto.

Conclusión

Una vez que se revisaron las opciones tecnológicas para implementar el sistema se determinó que MySQL junto con Apache y PHP forman un buen equipo.

Respecto de los manejadores de base de datos, observamos que MySQL resulta la mejor opción ya que no consume demasiados recursos, es mas seguro y tiene un mejor rendimiento.

Lenguaje del lado del servidor

En cuanto a los lenguajes de programación PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más.

PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores.



Con PHP no se encuentra limitado a resultados HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash (usando libswf y Ming) sobre la marcha. También puede presentar otros resultados, como XHTML y archivos XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

Quizá la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir la interfaz vía Web para una base de datos es una tarea simple con PHP.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros. PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en Web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

Por tanto se concluyó que PHP es la mejor elección ya que no solo trabaja para su desarrollo sobre ambiente Windows. Además que MySQL tiene una mejor interacción con PHP por tanto es el mejor candidato.

En cuanto al lenguaje que se ejecuta del lado del cliente debemos considerar cual es más compatible con los navegadores de Internet más utilizados. Debido a que Visual Basic Script (VBScript) solo es compatible con Internet Explorer no resulta ser una buena alternativa ya que reduce las opciones en cuanto al funcionamiento de la aplicación, pues forzaría a los usuarios a utilizar un navegador en específico, mientras que Javascript es compatible con mas navegadores, como son Internet Explorer, Netscape, Opera, entre otros, es por ello que se utilizará JavaScript como lenguaje para ejecutarse del lado del cliente



Aunque muchos de los lenguajes pudieran funcionar bien con AJAX, PHP tiene muchas similitudes con JavaScript (que es el lenguaje en el que se basa AJAX) tanto de funcionalidad como de disposición del código y de concepto. Esto retribuye en un mejor funcionamiento con JavaScript, que lo hace más fiable.

Lenguaje del lado del cliente

JavaScript, también ha sido una tecnología abierta, especialmente porque es visible para el usuario. Existe una gran comunidad de desarrolladores de JavaScript que poseen objetivos y formas de pensar muy parecidas a los de PHP.

JavaScript es también razonablemente fiable y también comparte características con los lenguajes orientados a objetos.

Por tanto al combinar estos dos lenguajes, que se encuentran en una etapa de madurez considerable, obtenemos lo mejor de dos grandes herramientas.

3.2 Posibles Módulos de la aplicación.

A continuación se presentara una descripción de los posibles módulos del sistema a desarrollar con base en los requerimientos.

El sitio Web a desarrollar contará con varias secciones y cada una de ellas posee sus propios módulos.

- ✦ **Primer** módulo: Tienda en línea
- ✦ **Segundo** módulo: Diseño y Cotización de productos personalizados.
- ✦ **Tercer** módulo: Galería de imágenes de los productos y servicios de la PYME que no pueden venderse en línea.
- ✦ **Cuarto** módulo: Presentación de la PYME. Información de contacto



Se desarrollará cada módulo de la siguiente manera:

Primer módulo: Tienda en línea

Este primer módulo contendrá tres submódulos principales (Ver figura 3.2.1).

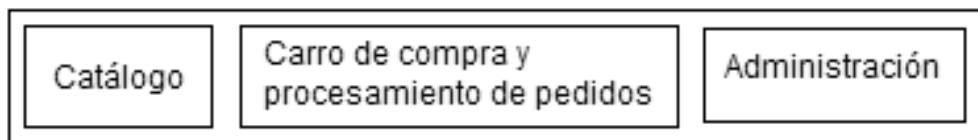


Figura 3.2.1 Módulos principales para la Tienda en línea

Además el sistema de venta en línea contará con dos tipos de vistas básicas: la vista de usuario y la vista del administrador.

✦ Vista básica de usuario:

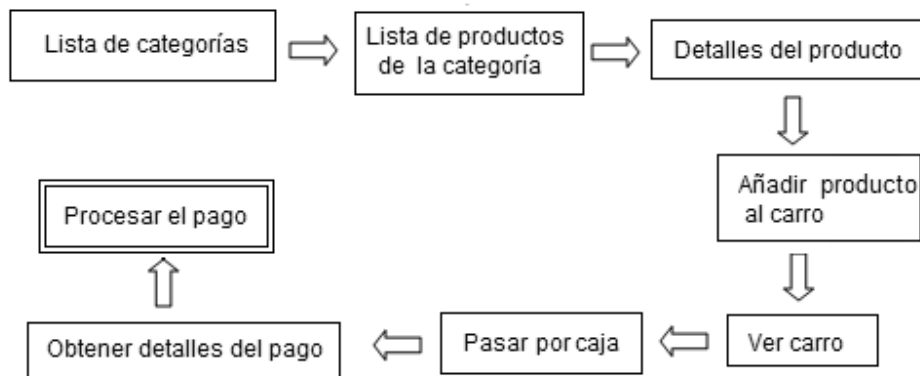


Figura 3.2.2 Diseño del flujo del sistema para la vista de usuario

Analizando la vista básica de usuario (ver figura 3.2.2), en primer lugar el cliente accede a la página principal en la que se enumeran todas las categorías de los productos del sitio, desde aquí, puede acceder a una categoría determinada y después a los detalles del producto. Con una opción visible para añadir un producto en concreto al carro.



✦ Vista básica de administrador

En el diseño del flujo del sistema para la vista de administrador, se permitirá al administrador registrar y añadir categorías y productos. (Ver figura 3.2.3)

El administrador tendrá una versión ligeramente distinta de la interfaz de usuario en el sitio, el administrador podrá seguir explorando por los productos y categorías pero en lugar de tener acceso al carro de la compra, podrá acceder a un determinado producto o categoría y modificar o eliminar dicho producto o categoría.

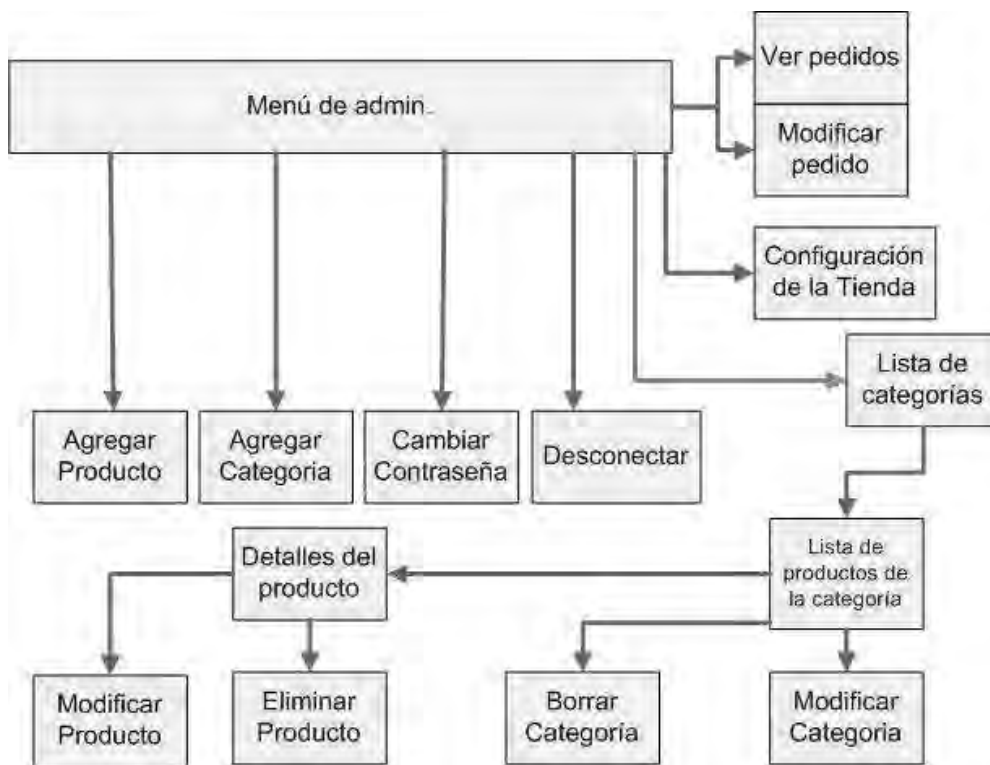


Figura 3.2.3 Flujo del sistema para la vista de administrador

Para la tienda en línea se manejan términos como carro de compra (a veces también se denomina cesta de la compra) para describir un mecanismo de compra en línea específico. Al explorar por el catálogo en línea, se podrá añadir artículos al carro de la compra. Cuando se termine, se pasa por la caja de la tienda, es decir, se adquiere los artículos del carro. Para poder implementar el carro de la compra es necesario implementar los siguientes módulos:



- ✦ Una base de datos de los productos que queremos comercializar en línea.
- ⊕ Un catálogo en línea de productos, enumerados por categoría.
- ✦ Un carro de la compra para guardar los artículos que el usuario quiere adquirir.
- ✦ Clientes. La tienda virtual crea un registro con todos sus clientes en una Base de Datos.
- ⊕ Una interfaz de administración.

Segundo módulo: Diseño y cotización de productos personalizados

Es un módulo de prueba para la empresa en la que el usuario podrá obtener una cotización en tiempo real de las medidas del producto que necesita (en este caso solo existirá un solo producto por el momento).

Por tanto solo se tendrán dos submódulos:

- ✦ Primer submódulo: Redimensionamiento visual
- ✦ Segundo submódulo: Calculo de la cotización según la redimensión de la imagen (guía de cálculo proporcionada por la empresa)

Tercer módulo: Galería de productos

Como se mencionó anteriormente como no todos los productos ofrecidos podrán venderse en la tienda en línea, por lo que es necesario contar con un mecanismo apropiado para la promoción de estos productos. Se hará con una galería de imágenes.

- ✦ Primer submódulo: Obtención de las categorías de los productos de una base de datos.
- ✦ Segundo submódulo: Presentación de las imágenes categorizadas.

Cuarto módulo: Presentación de la PYME

En esta sección se reunirá y pondrá disponible toda la información pertinente y necesaria para la presentación de la empresa, así como sus datos y formularios de contacto para posible asesoría o cotización.



3.3 Requisitos

Después de haber analizado las herramientas tecnológicas que el sistema empleará para su funcionamiento así como haber justificado su empleo, se comenzará con el proceso de conjunción de estas, de forma ordenada para los propósitos del sistema.

El primer paso en la realización de un proyecto de ingeniería es la obtención de las necesidades documentadas tanto de contenido, forma o funcionalidad de un producto a realizarse. Estas necesidades documentadas son los llamados requisitos, en muchas ocasiones se emplea la palabra requerimientos, esta es una mala traducción del inglés. La palabra *requirement* se debe traducir como requisito, en tanto que requerimiento se traduce al inglés como *request*.

De esta manera un requisito es una condición, característica o restricción que debe tener o cumplir un sistema o componente de un sistema para satisfacer un contrato, una norma, una especificación o algún otro documento formalmente solicitado.

Los requisitos establecen qué debe hacer el sistema sin detallar como se realizará. Esta información se utilizará como datos de entrada para la siguiente etapa que es la de diseño del producto.

Características de los requisitos.

Para todo requisito es necesaria la existencia de ciertas características que deben reunir para ser de calidad, estos son:

- ✦ Deben ser necesarios: Lo que pida un requisito debe ser necesario para el producto. Referenciar los aspectos importantes.
- ✦ Consistentes: Ningún requisito debe entrar en conflicto con otro requisito diferente, ni con parte de otro.



- ✦ No ambiguos: El texto debe redactarse en un lenguaje comprensible, preciso y tener una única interpretación posible.
- ✦ Completos: Los requerimientos deben contener en sí mismos toda la información necesaria, y no remitir a otras fuentes externas que los expliquen con más detalle:
 - ✦ Externamente completa si contiene todas las propiedades deseadas por el cliente.
 - ✦ Internamente completa si no existen referencias no definidas.
- ✦ Alcanzable: Un requerimiento debe ser un objetivo realista, posible de ser alcanzado con el dinero, el tiempo y los recursos disponibles.
- ✦ Deben ser correctos y sin errores.
- ✦ Deben evitar detalles de diseño.

Determinación de los Requisitos

Para poder determinar los requisitos es necesario seguir una serie de pasos para una correcta elaboración de estos.

Primeramente es necesaria la obtención de estos, la forma habitual es mediante el encuentro de interlocución con los usuarios o clientes mediante entrevistas, intercambio de opiniones, lluvia de ideas, algún cuestionario, etc., que puedan ser útiles en la redacción de un documento llamado “especificación de requisitos”. En este documento se integran además una definición de términos descriptivos para quienes se encargarán de la programación así como un esbozo de especificación previo al diseño completo. De esta manera quienes participan en la elaboración del documento son los representantes del cliente, representantes del desarrollador, diseñadores quienes usarán los requisitos como base del desarrollo, así como el equipo de pruebas quienes asegurarán que el sistema cumpla con los requisitos.

Para el presente trabajo se pudieron de igual manera obtener los requisitos mediante la interlocución con el representante directo de la PyME. Logrado en varias sesiones.



Después de la obtención de los requisitos se deben documentar para finalmente hacer una validación de estos (ver figura 3.3.1).



Figura 3.3.1 Ciclo de la determinación de Requisitos

Tipos de requisitos

En ingeniería de sistemas existen tres tipos de requisitos:

Requisitos funcionales: Estos requisitos también conocidos como de usuario o generales son los que describen la funcionalidad o los servicios que se espera proveerá el sistema entregado, la interacción entre el sistema y su ambiente. Estos requisitos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios de la aplicación. Se especifican entradas, salidas, excepciones, resultados a obtener, etc.

De esta manera este tipo de requisitos presentan lo siguiente:

- ✦ Un listado completo de las cosas que el cliente espera que haga el sistema propuesto para todas las categorías de los usuarios.



-
- ✦ Estar redactado de forma que el cliente pueda entenderlo (con claridad, conciso y congruente).
 - ✦ Es generado entre cliente y desarrollador como documento de compromiso.
 - ✦ Debe contener el propósito general del sistema con referencias a otros sistemas relacionados y un diccionario de términos y abreviaturas que puedan ser útiles.
 - ✦ Tener una descripción de antecedentes y objetivos del desarrollo.
 - ✦ Requerimientos sobre soporte, seguridad y privacidad y cualquier restricción.

Estos requisitos se obtienen mediante el planteamiento de preguntas que ayuden a determinar cuales son las características y funciones que debe cumplir el objeto para satisfacer las necesidades y las expectativas del usuario para que el producto sea utilizado, sirva y guste a quienes está dirigido, y por tanto esas preguntas deben dirigirse en el sentido de acotar con precisión las especificaciones del objeto por diseñar.

Dos variables de estos requisitos que resultan importantes también de considerar para el presente trabajo de aplicación Web son los siguientes:

- ✦ *Diseño*: Aunque el contenido de una página Web es el que genera a los usuarios de Internet la necesidad de visitarla, el diseño gráfico es el que hace que los usuarios se sientan atraídos por la misma constituyendo, en consecuencia, un elemento en el que hay que invertir un gran esfuerzo estético y funcional.

Para conseguir un diseño atractivo en la página Web del presente trabajo se realizará mediante programas específicos como Dreamweaver.

- ✦ *Usabilidad*: La Usabilidad es otra variable importante que es necesario tomar en cuenta ya que implica la facilidad con la que podrá navegarse en la página Web.

Uno de los módulos a desarrollar es una plataforma de comercio electrónico por tanto es fundamental que los usuarios de la aplicación no se pierdan entre contenidos y



productos y en consecuencia decidan no volver a la página Web. Por tanto se procurará la sencillez y la intuición en la página, para que el usuario sepa en todo momento donde se encuentra y pueda tener el control sobre la navegación que realiza.

A continuación se detallan los requisitos funcionales para el presente trabajo:

Para los administradores del sitio el sistema deberá ser capaz de:

- ✦ El sistema deberá contar con claves de acceso únicas para cada usuario autorizado que opere en el área administrativa del sistema.
- ✦ El usuario administrador podrá acceder sin problemas desde cualquier navegador, para realizar cada una de las actividades que el sistema propone.
- ✦ La base de datos solo podrá ser administrada por los usuarios autorizados mediante un sistema de autenticación basado en nombre de usuario y contraseña.
- ✦ Después de haberse autenticado correctamente como usuario con privilegios, el sistema presentará una interfaz administrativa para el sitio.
- ✦ Cada usuario con privilegios de administrador por medida de seguridad podrá cambiar su contraseña cuando lo desee.
- ✦ El administrador podrá tener acceso a una lista de módulos con sus propiedades.
- ✦ La interfaz administrativa deberá presentar al usuario las herramientas para poder añadir, modificar o eliminar algún producto o categoría de la tienda.
- ✦ Se deberá contar con un mecanismo de control de existencias en almacén.
- ✦ Podrá acceder a la información relacionada con los clientes.

Para cualquier tipo de usuario:

- ✦ Cualquier tipo de usuario podrá acceder al sitio desde cualquier navegador de Internet comercial.
- ✦ El usuario podrá tener acceso fácilmente a todas las áreas permitidas.



- ✦ Se requiere una forma de conectar la base de datos de los productos y sus categorías al explorador del usuario para que los usuarios puedan examinar los artículos por categoría.
- ✦ El usuario podrá ver una galería de imágenes de los productos en venta, así como los detalles de estos.
- ✦ Los usuarios deben poder seleccionar artículos del catálogo para su posterior adquisición.
- ✦ Se deberá dar un seguimiento de los artículos seleccionados por el usuario, mientras permanece su sesión.
- ✦ Cuando haya finalizado el usuario de realizar sus compras en línea, el sistema será capaz de presentar la suma de su pedido así como indicaciones de pago.
- ✦ El sistema contará con un apartado para los avisos al usuario respecto de los productos y de forma opcional un calendario visible.
- ✦ La página deberá contar con un apartado para la presentación formal de la PYME.
- ✦ Se requerirá una sección de contacto.
- ✦ Como un elemento opcional el sistema deberá ser capaz de presentar al usuario un mecanismo de presupuesto aproximado en tiempo real para determinado artículo que pueda ser *diseñado* por el usuario.
- ✦ El sistema será capaz de presentar al usuario un apartado para la solicitud de una cotización especial.

Requisitos no funcionales

Requisitos no funcionales o también llamados requisitos del Sistema son los que tienen como objetivo presentar y documentar las condiciones o características que debe cumplir un sistema y que tienen que ver con cuestiones como el rendimiento, la calidad, disponibilidad, el testeo, el mantenimiento, etc., en el que se especifican las asuntos del propio sistema y cómo debe realizar sus funciones.



Para el presente trabajo los requisitos no funcionales que se determinaron son los siguientes:

- ✦ Como regla comercial, todas las cantidades de los productos deberán ser siempre positivas. El sistema deberá ser capaz de evitar que el usuario intente establecer una cantidad negativa en la base de datos de información de productos, siendo rechazada esta opción por la lógica de control.
- ✦ Se deberá contar con una guía de cálculo para los presupuestos aproximados.
- ✦ La interfaz del usuario deberá realizarse en un ambiente gráfico e intuitivo para evitar inconvenientes en el uso por parte de los usuarios y para que personas con una mínima capacitación sean capaces de utilizarlo.
- ✦ El sistema debe ser capaz de manejar variables de sesión para que el administrador pueda acceder a las aplicaciones que este requiera dentro del sitio sin perder su autenticación.
- ✦ La base de datos se ubicará en un servidor central.
- ✦ El sistema deberá garantizar los servicios de seguridad de la información como son la confidencialidad, integridad y disponibilidad.
- ✦ Mediante sentencias SQL se podrá consultar la información contenida en la base de datos.
- ✦ Las ventanas de salida permitirán mostrar resultados de las transacciones realizadas.
- ✦ Debido a que el sistema posee interfaz gráfica es necesario utilizarlo en su mayor parte mediante dispositivos de entrada como son el ratón y el teclado.
- ✦ Si algún usuario intenta entrar en algún área no permitida, el sistema le anunciará el acceso denegado.
- ✦ Debe el sistema soportar y prever medios de envío de registros y archivos (fotos).
- ✦ Respaldo de datos.



Requisitos de la empresa

Requisitos de la empresa o limitaciones externas: Son condiciones que afectan de forma indirecta al producto (compatibilidades, leyes, regulaciones aplicables al producto).

Para el presente trabajo el requisito de empresa es el siguiente:

- ✦ Para aminorar costos el sistema tendrá que ser desarrollado con herramientas de distribución gratuita.

La determinación de requisitos tiene 2 propósitos:

- ✦ El acuerdo entre clientes y desarrolladores sobre qué debe ser el sistema.
- ✦ Proporcionar a los diseñadores pautas para el desarrollo

3.4 Interactividad y capacidad de respuesta en aplicaciones Web

Las aplicaciones Web basadas en una programación AJAX han demostrado obtener mejores rendimientos contra las aplicaciones Web de programación tradicional.

Se han hecho algunas pruebas para medir la eficiencia de dicha programación, tomando en cuenta el número total de Bytes transferidos y el tiempo de espera.

Haciendo consultas a una base de datos considerando ambas arquitecturas. (Ver Figura 3.3.2).

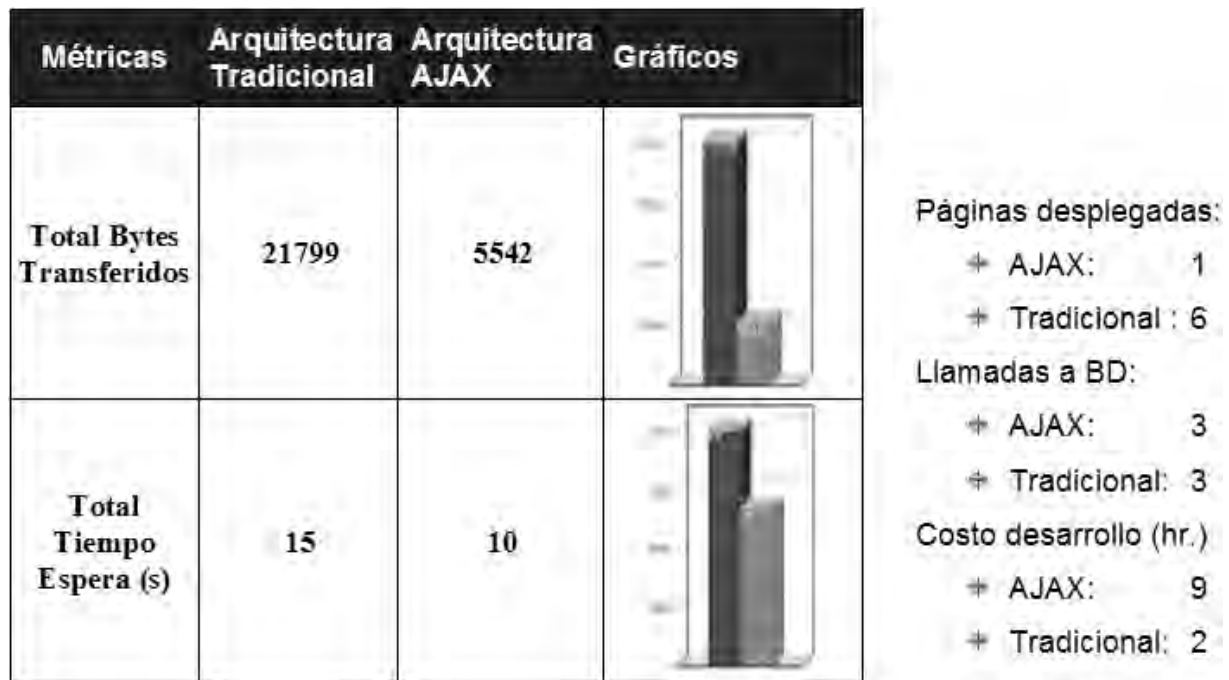


Figura 3.3.2 Capacidad de respuesta entre arquitecturas

Demostrando así que es superior en mucho la arquitectura AJAX, implicando una mayor posibilidad de brindar una mejor experiencia al usuario.

Ya que reduce tiempos de espera llevando al usuario a una mayor interactividad.

CSS se utiliza principalmente para estilizar las páginas Web y hacerlas más fácilmente comprensibles para los usuarios.

Javascript permite la accesibilidad al DOM (Document Object Model) para permitir una presentación de información dinámica y una mayor interactividad.

El DOM es un modelo de objetos estándar que hace HTML, XML y otros formatos relacionados. XMLHttpRequest es importante en las técnicas AJAX de desarrollo Web para aumentar la capacidad de respuesta de páginas Web y la interactividad. Por



ejemplo, se utiliza actualmente en los servicios de Gmail de Google y MapQuest entre otros sitios altamente interactivos.

Los principales beneficios de las aplicaciones AJAX permiten a los usuarios acceder y crear aplicaciones Web más rápido con un mejor nivel de respuesta de lo que se suele atribuir a las aplicaciones de escritorio.

Esta mayor capacidad de respuesta va acompañada de una mayor interactividad que hacen innecesario que la página Web sea vuelta a cargar cada vez que un usuario realiza una solicitud nueva de la aplicación Web.

La información aquí presentada será reafirmada en el momento de las pruebas e integración del presente trabajo



CAPITULO 4

Diseño y construcción de la aplicación





CAPÍTULO 4 DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1 Metodología a implementar

Para el presente trabajo se empleará el modelo Top-down, en el que se formulará un resumen del sistema, sin especificar detalles. Cada parte del sistema se refinará diseñando con mayor detalle hasta que la especificación completa es lo suficientemente detallada para validar el modelo. El modelo "Top-down" se diseña con frecuencia con la ayuda de "cajas negras" que hacen más fácil cumplir requerimientos aunque estas cajas negras no expliquen en detalle los componentes individuales.

El diseño Top-down fue promovido en los setenta por los investigadores de IBM Harlan Mills y Niklaus Wirth. Mills desarrolló los conceptos de programación estructurada para usos prácticos y los probó en un proyecto en 1969 para automatizar el índice de la morgue del New York Times. El éxito administrativo e ingenieril del proyecto hicieron que el enfoque top-down se esparciera por IBM y por el resto de la industria de las computadoras. Niklaus Wirth, que entre sus logros está el desarrollo del lenguaje de programación Pascal, escribió el artículo Program Development by Stepwise Refinement, que tuvo mucha influencia. Los métodos Top-down fueron favorecidos en la ingeniería de software hasta que llegó la programación orientada a objetos a finales de los 1980s.

Enfatiza además la planificación y conocimiento completo del sistema. Se entiende que la codificación no puede comenzar hasta que no se haya alcanzado un nivel de detalle suficiente, al menos en alguna parte del sistema. Esto retrasa las pruebas de las unidades funcionales del sistema hasta que gran parte del diseño se ha completado.

También consiste en efectuar una relación entre las etapas de la estructuración de forma que una etapa jerárquica y su inmediato inferior se relacionen mediante entradas y salidas de información. Podría decirse en forma práctica que consiste en una serie



de descomposiciones sucesivas del problema inicial, que recibe el refinamiento progresivo del repertorio de instrucciones que van a formar parte del programa.

La utilización de la técnica de diseño Top-Down tiene los siguientes objetivos básicos:

- ✦ Simplificación del problema y de los subprogramas de cada descomposición.
- ✦ Las diferentes partes del problema pueden ser programadas de modo independiente e incluso por diferentes personas.
- ✦ El programa final queda estructurado en forma de bloque o módulos lo que hace más sencilla su lectura y mantenimiento.

Ventajas del diseño Top-Down

- ✦ Alto nivel de abstracción: Una de las principales ventajas del diseño Top-Down es que el diseñador puede especificar el diseño en un alto nivel de abstracción sin necesidad de considerar el mismo inicialmente a nivel de componentes.
- ✦ Diseño modular: El diseño **Top-Down** ofrece como ventaja que la información se estructura en forma modular. Como el diseño se realiza a partir del sistema completo y se subdivide en módulos, permite que las subdivisiones se realicen de forma que los mismos sean funcionalmente independientes. El diseño Bottom-Up no contempla la división en partes funcionalmente independientes. Esta es la desventaja fundamental del diseño Bottom-Up. El resultado final puede aparecer confuso al no estar dividido en módulos independientes.
- ✦ Diseño jerárquico: En este entran en juego una cantidad considerable de componentes. Estos diseños deben organizarse de tal forma que resulte fácil su comprensión. Una forma de organizar el diseño es la creación de un diseño modular jerárquico. Un diseño jerárquico está constituido por niveles en donde cada uno es una especialización del nivel superior. La organización jerárquica es una consecuencia directa de aplicar la metodología Top-Down.



Planificación de sitio Web.

Se considerarán algunos aspectos útiles para la creación del sitio Web o página Web. Primeramente se hará la elección del tipo de Web que se creará de acuerdo a las necesidades antes mencionadas. Por las características de nuestro Web el sitio será comercial ya que ofreceremos al usuario la posibilidad de comprar en línea algún producto en existencia

Posteriormente se definirá la temática del sitio.

Aquí es donde se definirán los temas que se expondrán en el sitio Web, permitiendo definir términos claves de búsquedas para posteriormente realizar una metodología de posicionamiento en los buscadores más populares.

De esta manera, la temática del sitio Web a desarrollar es la promoción, venta y distribución de productos y servicios de herrería en general.

Después se plantearán los objetivos que se pretenden alcanzar con la creación del sitio Web, los objetivos de la página a construir son dar a conocer el negocio, captar clientes y generar nuevos pedidos de una forma más eficiente.

La manera en la que será construido ayudará a poder ampliar el sitio Web con futuros cambios para poderlo mejorar agregando nuevas aplicaciones, secciones, páginas, actualizaciones constantes, etc.

Después de plantear los objetivos y la escalabilidad de la Web tanto a corto como a largo plazo, se procederá a definir sobre papel el diseño de la Web, incluyendo fondos, tipos de letras, botones, formularios, links, plantillas aplicaciones, de tal manera que se obtengan todos los elementos y diseños que se desean implementar en el sitio Web.



Diseño visual y creación de la información a implementar

Aquí se define el sitio con un máximo nivel de detalle, se generan modelos del aspecto gráfico, del contenido y del funcionamiento. Estos modelos son prototipos del sitio y lo reflejan con exactitud, se trabaja con el cliente evolucionándolos hasta que quede satisfecho con todos los aspectos. Es fundamental la aprobación del cliente para seguir avanzando, pues los cambios en etapas posteriores serán más costosos.

Ahora que se conoce al detalle el sitio se corrigen los plazos y costos de las siguientes etapas.

Creación del esqueleto de la Web, tablas, encabezados, espacio para imágenes, texto, botones, etc.

Creación de las imágenes que acompañara a nuestro sitio Web, logos, cabeceras, fotografías, además del proceso concepción y materialización de la información que se va a ofrecer.

✦ Aplicaciones Web

Creación de las aplicaciones que contendrán nuestro sitio Web, como el mecanismo de venta en línea de los productos, la personalización de productos, etc., mediante programación específica y creación de bases de datos.

Se hace realidad el modelo de diseño, es decir que se elaboran y adaptan los elementos gráficos y multimediales, se codifican las páginas, los programas y scripts, se definen y preparan las bases de datos para que el sitio quede en funcionamiento.

En esta etapa hay que realizar pruebas exhaustivas para asegurar el perfecto funcionamiento del mismo, se prueba la integración con los sistemas internos del cliente, todo esto se hace primero en el ambiente de desarrollo y luego en el servidor de Internet en el que realmente funcionará.



Los administradores reciben la capacitación para desarrollar su tarea. El sitio está listo para comenzar a funcionar. Separar lógica y contenido. Gracias a la separación de la lógica y el contenido (PHP y HTML) los sitios resultan más sencillos de utilizar y de mantener a largo plazo. Es importante tener en cuenta que en proyectos con un número reducido de líneas de código puede volverse un inconveniente esta separación. Pero cuando el proyecto es mayor, esta separación es fundamental.

Esta separación de lógica y contenido puede llevarse a cabo mediante:

- ✦ Archivos para almacenar distintas partes del contenido.
- ✦ Utilizar un API de funciones o de clases con un conjunto de funciones miembro para conectar contenido dinámico o plantillas de páginas estáticas.
- ✦ Utilizar un sistema de plantillas. La principal ventaja es que si otra persona diseña nuestras plantillas, por ejemplo un diseñador gráfico, no es necesario que domine el código PHP. Pueden usarse las plantillas proporcionadas con mínimas modificaciones.
- ✦ Posicionamiento.

Una vez que se haya terminado el sitio Web lo que procede es que sea localizable en cualquier buscador, especialmente Google, que pueda aparecer en las primeras posiciones para recibir mas visitas a la Web.

Es este punto es donde se ha de implementar una metodología específica para alcanzar un posicionamiento óptimo, definiendo las palabras claves de búsqueda.

✦ Pruebas

Realización de pruebas para comprobar la Usabilidad y correcto funcionamiento del sitio, para evitar mostrar páginas con un formato diferente que pueden aportar una imagen negativa.

Puesta en Funcionamiento: Habilidad del sitio, en Internet, para que los verdaderos usuarios comiencen a servirse del mismo.



Es necesario difundirlo, mediante publicidad, papelería de la organización, por banners en sitios específicos, inclusión en buscadores, etc. Esto se hace en función de la audiencia y de los objetivos.

El sitio se pondrá en pleno funcionamiento con usuarios verdaderos accediendo y con los administradores realizando las tareas periódicas de mantenimiento como son: atención al cliente, actualización de contenidos y administración del servidor.

Implementación de un sistema de pago

En el presente trabajo se sumará el pedido del usuario y se anotarán los detalles de entrega. No se procesarán los pagos. Hay una gran variedad de sistemas de pago disponibles, cada uno con una implementación diferente.

Como mínimo, el sistema de pagos que seleccione necesitará información del usuario (por ejemplo, un número de tarjeta de crédito), información que nos identifique (para especificar en qué cuenta se va a realizar el cargo) y la suma total de la transacción.

4.2 Diseño y Construcción del Back-End

Para esta sección se diseñará primero la base de datos que se utilizará en todo el sitio Web.

En la tabla 4.2.1 se resumen todas las tablas propuestas de la Base de Datos que serán necesarias para la sección de la tienda en línea y la galería de imágenes así como una breve descripción de cada tabla.



Nombre de la tabla	Descripción
admin	Guarda las credenciales de los usuarios con privilegios de administrador
categorias_tienda	Almacena todas las categorías de los productos de la tienda en línea.
cat_galeria	Contiene todas las categorías de los productos que solo se mostrarán en la galería.
contactocotiza	Guarda los datos de los usuarios que solicitan una cotización.
productos	Almacena todos los productos que se venderán en línea.
prod_galeria	Contiene todos los productos que se mostrarán en la galería de imágenes.
clientes*	Datos de los clientes frecuentes.
pedidos*	Almacena los pedidos que se han hecho así como datos de envío.
pedido_productos*	Tabla temporal que relaciona los productos con los pedidos.

* Para los propósitos del presente no se utilizarán en el sitio Web pero si se planteará.

Tabla 4.2.1 Resumen de las tablas propuestas para el sitio

El Diagrama de Entidad Relación (DER) de las tablas a emplear se muestra en la figura 4.2.1.

A continuación se explicarán mejor cada tabla:

- ✦ **categorias_tienda.** Esta tabla almacena las categorías de los productos. Desde el DER se puede ver que la llave primaria de esta categoría es la llave foránea de la categoría productos, podría decirse que es una tabla hija.
- ✦ **cat_galeria.** De forma análoga a la tabla anterior en esta tabla se guardarán los nombres de las categorías de los productos que no se venderán en línea pero que se mostrarán en una galería de imágenes. De igual manera la llave primaria de esta tabla es la llave foránea de la tabla prod_galeria. Reafirmando la integridad referencial.

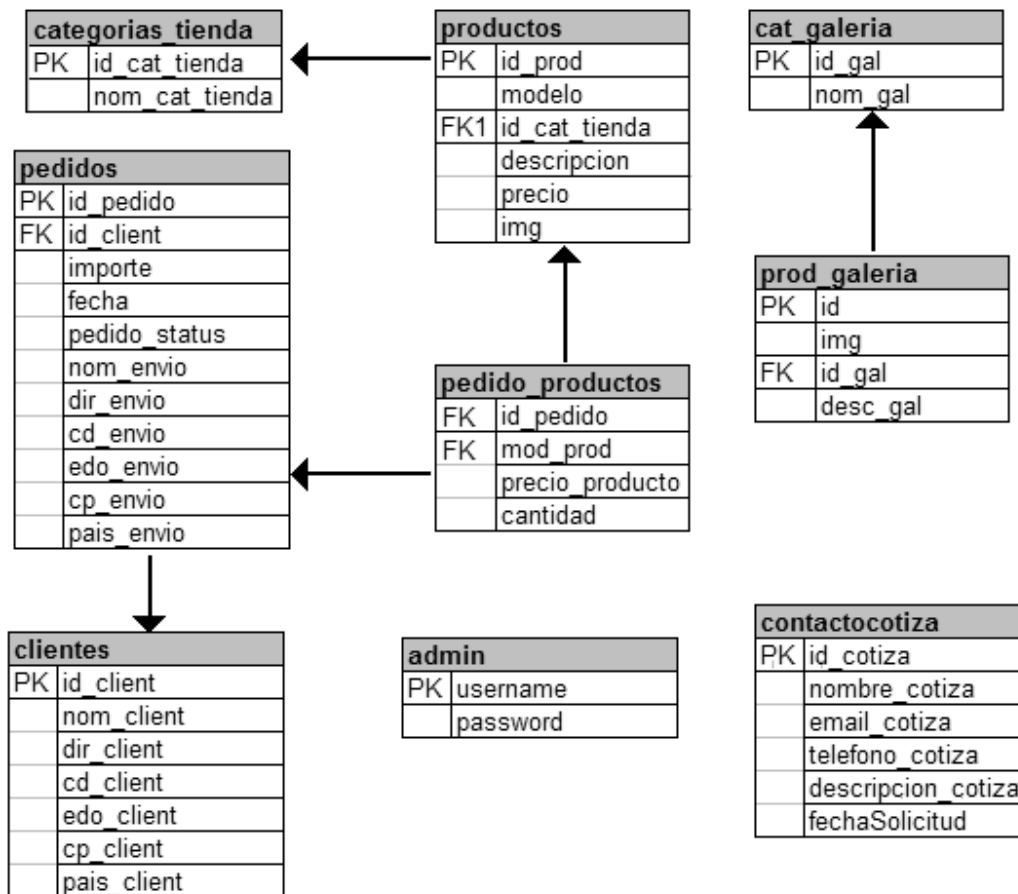


Figura. 4.2.1 Diagrama Entidad Relación de las tablas del proyecto.

✦ **productos.** En esta tabla se guardarán el modelo de los productos que se venderán en línea, la categoría a la que pertenecen, una descripción del producto, su precio asociado y el nombre del archivo de imagen del producto. El sistema será capaz de generar automáticamente las imágenes en miniatura de cualquier tamaño de imagen que el administrador guarde para ser mostradas en línea, por tanto no será necesario guardar el nombre de un archivo adicional de la imagen (thumbnail) en esta tabla.

✦ **prod_galeria.** Aquí es donde se almacenarán todos los productos que se mostrarán en la galería de imágenes, de acuerdo a su categoría. Para los fines



de la galería solo se requerirá guardar el nombre del archivo de imagen asociado al producto, la categoría a la que pertenece y una breve descripción.

- ✦ **admin.** Esta tabla contendrá las credenciales de los usuarios con derechos de administrador, como la manera de autenticar a este tipo de usuario será por nombre de usuario y contraseña, solo esos dos parámetros serán guardados.

- ✦ **contactocotiza.** Aquí irán los datos de los usuarios finales que deseen obtener una cotización de algún producto por fabricar o solicitar más información de algún producto en existencia.

Las siguientes tablas solo se plantearán y explicarán ya que para los propósitos del presente trabajo no serán necesarias en el funcionamiento del sitio, quedarán para una posterior mejora del proyecto.

- ✦ **pedidos.** Tabla que almacena los pedidos que se han hecho así como datos de envío. Esta tabla requerirá del id del cliente que ha hecho el pedido por lo que esta tabla contendrá una llave foránea que será la llave primaria de la tabla clientes. El ingreso de datos a esta tabla tendrá que ser, por norma de la empresa, posterior al proceso de pago.

- ✦ **clientes.** Datos de los usuarios que han hecho compras en el sitio.

- ✦ **pedido_productos.** Esta tabla temporal es para eliminar la relación M:M entre la tabla productos y pedidos ya que varios productos pueden estar relacionados con varios pedidos, con esta tabla se logra que solamente un producto pueda estar asociado a un pedido en particular.

Instalación de las herramientas necesarias

Para el desarrollo de la aplicación se empleará la herramienta XAMPP. A continuación se muestra el proceso de instalación:



-
- ✦ Acceder al sitio Web de *ApacheFriends*, donde se elegirá la versión para Windows y bajar la versión *zip*, aunque también podemos hacerlo en los formatos *exe* o *tar*. Mediante este modo (*zip*) no se efectuará ningún apunte en el registro de Windows y si en algún momento no se desea seguir con esta aplicación, bastará con borrar el directorio creado.

 - ⊕ Ubicarse en el la raíz del disco duro C:\ en los directorios habituales de instalación de programas al descomprimirse los archivos, automáticamente, se creará en C:/ la carpeta *xampp*

 - ✦ Ejecutar el archivo C:/xampp/xampp-control.exe, aparecerá entonces su consola de administración.

Los módulos que Apache nos permite poner en marcha o, según el caso, detener son: Apache, MySQL, Filezilla y Mercury.

En el caso de los dos primeros, podemos elegir arrancarlos como *servicio* o como *aplicación*; en este caso, dado que se trata de una instalación de pruebas, se procederá a iniciarlos como aplicación ya que, de este modo, no se sobrecargará la computadora con la ejecución de servicios innecesarios al arrancar Windows.

Para comprobar que todo funciona perfectamente se pulsará sobre el botón “Start” del módulo Apache e, inmediatamente, avisará mediante el mensaje “Running” de que el módulo se encuentra activo.

Lo mismo se hará con el módulo de MySQL que mostrará el mismo mensaje que en el caso anterior (Ver figura 4.2.2).



Figura 4.2.2 Panel de control de XAMPP

Después de lo anterior se procederá a abrir una ventana del algún navegador para acceder a su consola principal, en la barra de direcciones se introduce la dirección <http://localhost>, e inmediatamente se accederá a su consola de administración. (Ver figura 4.2.3)

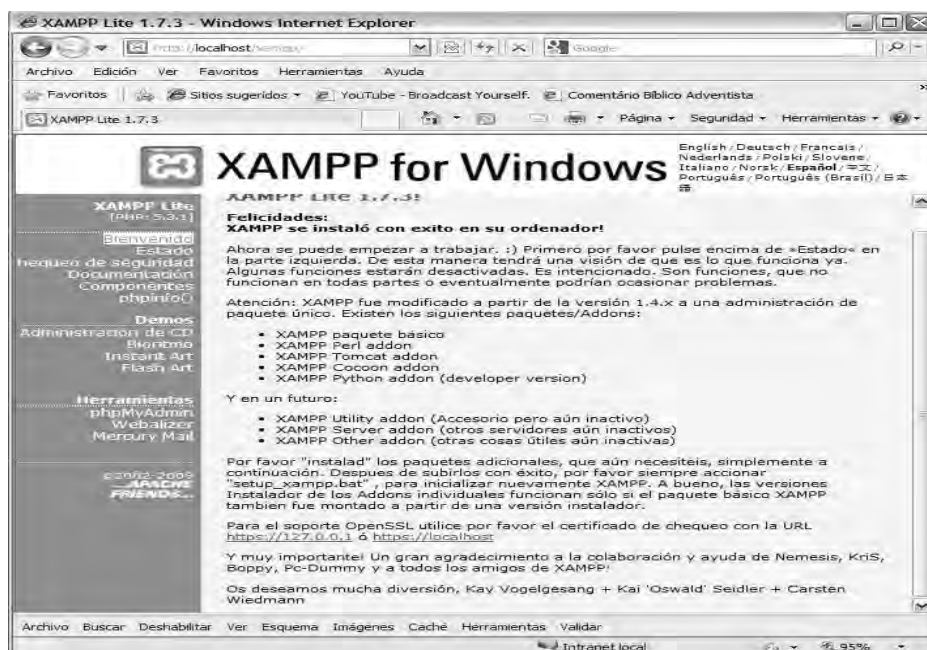


Figura 4.2.3 Consola de administración de XAMPP



CAPÍTULO 4. DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

La raíz del servidor Apache se encuentra en el directorio *C:\xampp\htdocs*. Todos los archivos que se encuentren dentro de este directorio serán procesados por Apache cuando se acceda mediante la URL especial *http://localhost*.

Desde la consola anterior, en el menú principal, se podrá acceder a la aplicación phpMyAdmin que será la que nos facilitará el uso de MySQL. Esta herramienta será de gran ayuda en el momento de realizar las tablas de la base de datos, ya que ofrece la posibilidad de usar tanto el modo comando como el modo gráfico, agilizando el proceso de las operaciones esenciales de las bases de datos como actualizar, agregar o borrar datos de las tablas de forma rápida y confiable. En la figura 4.2.4 se muestra la página principal de la herramienta phpMyAdmin.



Figura 4.2.4 Página principal de phpMyadmin

Con ayuda de esta herramienta se podrán crear las tablas necesarias para cada una de las secciones del sitio. Una ventaja importante de phpMyadmin es que por ser aplicación Web tiene la ventaja de poderse ejecutar en diferentes plataformas.



En la configuración por defecto se accede con el usuario *root* sin una contraseña, obviamente, por motivos de seguridad, esta configuración se cambiará, posteriormente se agregará una contraseña ya que no puede quedar así en la implementación final.

Como la importancia de realizar páginas dinámicas radica en las base de datos, esa será la primera tarea en llevar a cabo.

Creación de la base de datos

Es la tarea primordial ya que será en ella donde se almacenarán todas las tablas que utilizará el sitio, necesarias tanto para la tienda en línea como para las categorías y los productos respectivos. El nombre que tendrá la base de datos será “herreria” y el comando SQL que permitirá su creación es `CREATE DATABASE` seguido del nombre de la base de datos:

```
CREATE DATABASE `herreria`;
```

Posteriormente será necesaria la elaboración de cada tabla respectiva.

Creación de tablas en MySQL

Las tablas se crean con el comando `CREATE TABLE` seguido del nombre de la tabla, en la sentencia se describen los tipos de datos en cada campo, la llave primaria y Foránea si es el caso, así como si el campo soportará valores nulos.

Tablas para la tienda en línea

Tabla para las categorías. Como uno de los propósitos del sitio Web a construir es la venta de productos elaborados de herrería clasificados por categoría será necesario elaborar primeramente la tabla que guardará el nombre de las categorías. Esto se logrará con el comando antes mencionado. El nombre de la tabla para las categorías de la tienda será: `'categorias_tienda'`.



Como una opción por si se desea posteriormente cambiar el proveedor del hospedaje del sitio y se desee migrar la base de datos es posible utilizar el condicional 'IF NOT EXIST' para que se cree la tabla si es que no se ha creado previamente, quedando como sigue:

```
CREATE TABLE IF NOT EXISTS `categorias_tienda` (  
  `id_cat_tienda` int(11) NOT NULL AUTO_INCREMENT,  
  `nom_cat_tienda` varchar(30) CHARACTER SET latin1 NOT NULL,  
  PRIMARY KEY (`id_cat_tienda`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=10 ;
```

Para verificar que se haya creado correctamente puede emplearse el comando DESCRIBE categorias_tienda que mostrará la siguiente información. (Ver figura 4.2.5).

Mostrando registros 0 - 1 (2 total, La consulta tardó 0.0033 seg)

DESCRIBE categorias_tienda [Editar] [Crear código PHP]

Mostrar 30 filas empezando de 0 en modo horizontal y repetir los encabezados cada 100 celdas

+ Opciones

Field	Type	Null	Key	Default	Extra
id_cat_tienda	int(11)	NO	PR	NULL	auto_increment
nom_cat_tienda	varchar(30)	NO		NULL	

Figura 4.2.5 Resultado del comando DESCRIBE para las tabla categorias_tienda

Tabla para los productos.

De forma análoga al procedimiento anterior se creará la tabla de los productos que contendrá el modelo del producto, un campo para almacenar su clasificación, dicho campo será un llave foránea que apuntará a la tabla de las categorías, esto con el fin de evitar que el usuario administrador pueda cometer errores al registrar la categoría asociada a un nuevos producto, también la tabla contendrá un campo para la descripción del producto, otro para el precio, uno más para guardar el nombre del



CAPÍTULO 4. DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

archivo de la imagen asociada. Por tanto el código para la creación de dicha tabla es el que se muestra a continuación:

```
CREATE TABLE IF NOT EXISTS `productos` (  
  `id_prod` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `modelo` varchar(20) NOT NULL,  
  `id_cat_tienda` int(11) NOT NULL,  
  `descripcion` text NOT NULL,  
  `precio` float(10,2) NOT NULL,  
  `img` varchar(20) NOT NULL,  
  PRIMARY KEY (`id_prod`),  
  KEY `id_cat_tienda` (`id_cat_tienda`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=36 ;
```

De igual manera, para verificar que la tabla se haya creado correctamente se podrá usar el comando `DESCRIBE productos` (Ver Figura 4.2.6).

Mostrando registros 0 - 5 (6 total, La consulta tardó 0.0029 seg)

DESCRIBE productos [Editar] [Crear código PHP]

Mostrar 30 filas empezando de 0 en modo horizontal y repetir los encabezados cada 100 celdas

+ Opciones

	Field	Type	Null	Key	Default	Extra
<input type="checkbox"/>	id_prod	int(10) unsigned	NO	PRI	NULL	auto_increment
<input type="checkbox"/>	modelo	varchar(20)	NO		NULL	
<input type="checkbox"/>	id_cat_tienda	int(11)	NO	MUL	NULL	
<input type="checkbox"/>	descripcion	text	NO		NULL	
<input type="checkbox"/>	precio	float(10,2)	NO		NULL	
<input type="checkbox"/>	img	varchar(20)	NO		NULL	

Figura 4.2.6 Resultado del comando `DESCRIBE` para la tabla `productos`

Donde el campo “id_cat_tienda” por ser una llave o clave foránea se muestra como índice (Ver figura 4.2.6).



Índices: @

Acción	Nombre de la clave	Tipo	Único	Empacado	Campo	Cardinalidad	Cotejamiento	Nulo	Comentario
	PRIMARY	BTREE	Si	No	id_prod	35	A		
	id_cat_tienda	BTREE	No	No	id_cat_tienda	0	A		

Figura 4.2.6 Muestra de claves de la tabla productos.

Tablas para la galería

Como se mencionó anteriormente será necesario tener un registro de los productos que se mostrarán en la galería de imágenes.

Tabla para las categorías de la galería. Las imágenes de los productos también se encontrarán organizadas por categorías, por tanto para mostrarlas se requerirá previamente construir en la base de datos una tabla para guardar los nombres de las categorías de la galería. Usando el procedimiento descrito anteriormente el código queda como sigue:

```
CREATE TABLE IF NOT EXISTS `cat_galeria` (
  `id_gal` int(11) NOT NULL AUTO_INCREMENT,
  `nom_gal` varchar(30) NOT NULL,
  PRIMARY KEY (`id_gal`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;
```

Verificando su correcta creación con DESCRIBE cat_galeria (Ver Figura 4.2.7)

Mostrando registros 0 - 1 (2 total. La consulta tardó 0.0117 seg)

DESCRIBE cat_galeria [Editar] [Crear código PHP]

Mostrar: 30 filas empezando de 0 en modo horizontal y repetir los encabezados cada 100 celdas

+ Opciones

	Field	Type	Null	Key	Default	Extra
<input type="checkbox"/>	id_gal	int(11)	NO	PRI	NULL	auto_increment
<input type="checkbox"/>	nom_gal	varchar(30)	NO		NULL	

Figura 4.2.7 Tabla para las categorías de la glería



Tabla para los productos de la galería. Del mismo modo será necesario crear una tabla de los productos que se exhibirán en la galería, dicha tabla, a diferencia de la tienda, solamente almacenará el nombre del archivo de la imagen del producto, su categoría a la que pertenece y la descripción. Por tanto se creará dicha tabla de la siguiente manera:

```
CREATE TABLE IF NOT EXISTS `categorias_tienda` (  
  `id_cat_tienda` int(11) NOT NULL AUTO_INCREMENT,  
  `nom_cat_tienda` varchar(30) CHARACTER SET latin1 NOT NULL,  
  PRIMARY KEY (`id_cat_tienda`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=10 ;
```

De igual manera que en las anteriores tablas verificamos con `DESCRIBE prod_galeria` (ver figura 4.2.8).

Mostrando registros 0 - 3 (4 total. La consulta tardó 0.0032 seg)

DESCRIBE prod_galeria

[Editar] [Crear código PHP]

Mostrar: 30 filas empezando de 0

en modo horizontal y repetir los encabezados

cada 100 celdas

+ Opciones

	Field	Type	Null	Key	Default	Extra
<input type="checkbox"/>	id	int(11)	NO	PRI	NULL	auto_increment
<input type="checkbox"/>	img	varchar(20)	NO		NULL	
<input type="checkbox"/>	id_gal	tinyint(4)	NO		NULL	
<input type="checkbox"/>	desc_gal	text	NO		NULL	

Figura 4.2.8 Resultado del comando DESCRIBE para la tabla “prod_galeria”.



Área administrativa

En esta sección también será necesario contar con un registro de los usuarios con privilegios de administrador, por estar basado el proyecto en una autenticación del tipo username password solamente será necesario contar con una tabla que contenga un campo para el nombre de usuario y otro para el password. Usando el mismo procedimiento de las tablas anteriores queda el código SQL como sigue:

```
CREATE TABLE IF NOT EXISTS `admin` (
  `username` char(16) NOT NULL,
  `password` char(40) NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Verificando con “DESCRIBE admin.” (Ver figura 4.2.9)

Mostrando registros 0 - 1 (2 total. La consulta tardó 0.0037 seg)

DESCRIBE admin [Editar] [Crear código PHP]

Mostrar 30 filas empezando de 0 en modo horizontal y repetir los encabezados cada 100 celdas

+ Opciones

	Field	Type	Null	Key	Default	Extra
<input type="checkbox"/>	username	char (16)	NO	PRI	NULL	
<input type="checkbox"/>	password	char (40)	NO		NULL	

Figura 4.2.9 Resultado del comando DESCRIBE para la tabla “admin”

Eliminación de tablas

Si se desea borrar una tabla existente, así como sus datos, introducir el comando:

```
DROP TABLE <tabla>;
```



Consulta de la base de datos

La manipulación de datos del lenguaje de consultas estructurado, puede ser resumida en los siguientes cuatro comandos: SELECT, INSERT, UPDATE y DELETE. Existe una serie de reglas comunes de sintaxis para todas las instrucciones. A continuación se muestran algunas de estas:

- ✦ Si se utilizara más de una tabla en la cláusula FROM, todos los campos con el mismo nombre, deben ir precedidos por el nombre de la tabla en donde se vayan a utilizar.
- ✦ Las instrucciones múltiples de una cláusula WHERE deben conectarse por medio de alguna palabra clave: AND u OR.
- ✦ Todas las instrucciones SQL deben tener un verbo, una cláusula FROM o INTO y usualmente una lista de parámetros de nombres de columna.

SELECT. Esta instrucción SELECT es probablemente la más utilizada de las instrucciones SQL. Ésta devuelve únicamente los datos contenidos en la base de datos. MySQL probablemente procesa este tipo de consulta más rápido que cualquier otra base de datos. A continuación se muestra su sintaxis.

```
SELECT <columna> FROM <tabla> WHERE <condición>;
```

El nombre de columna puede ser de una o más columnas que pertenezcan a una tabla en la cláusula FROM. El orden en que se especifiquen los nombres de las columnas es con el que se mostrarán. Si se utiliza el comodín * se seleccionarán todas las columnas.

INSERT INTO. Este comando añade nuevos registros a una tabla existente. El formato es básicamente el mismo que el de la instrucción SELECT.

Ejemplo:

```
INSERT INTO `herreria`.`admin` (`username` , `password` )  
VALUES ('administrator', 'contraseña');
```



El nombre de la tabla en la que se realizará la inserción se pone después de la palabra reservada INTO. Después de la palabra clave VALUES deberá ir la lista de valores que se quieren insertar. Si se inserta un registro de esta forma, debe tener un valor para cada columna y el orden de los datos deberá concordar con el orden en que se declararon las columnas en la base de datos.

UPDATE. El comando UPDATE permite editar valores de datos existentes. Este comando es muy similar a SELECT, pero permite manipular los valores seleccionados. La sintaxis es la siguiente:

```
UPDATE <tabla> SET <columna> = <valor> WHERE <condición>;
```

Así como todas las instrucciones SQL, el verbo reservado va primero. A éste le sigue el nombre de la tabla que se manipulará. A continuación se emplea la palabra reservada SET y, por último, se coloca la lista de nombres de columna y valores que se desean establecer. Pueden existir valores múltiples después del comando SET. Cada par columna/valor debe estar separado por una coma. La cláusula WHERE designa qué registros serán actualizados.

DELETE. La instrucción DELETE es muy similar a SELECT, la única diferencia es que en lugar de seleccionar registros para verlos, borra esos registros. Debe tenerse precaución al usar esta instrucción y evitar accidentalmente borrar registros. Puede usarse primero la instrucción SELECT y así asegurarse de que solamente se borrará lo deseado.

El verbo reservado se escribe primero, enseguida la palabra reservada FROM, seguida del nombre de la tabla y después la cláusula WHERE donde se establece la condición que deben cumplir los registros a borrarse. Si no se especifica la condición se borrarán todos los registros de la tabla:

```
DELETE FROM <tabla> WHERE <condición>;
```




Respaldo de la base de datos

El respaldar la base de datos es una parte fundamental de la administración de cualquier sistema de bases de datos. Los motivos van desde cualquier incidente físico del equipo, hasta algún daño accidental por parte de algún usuario. Tan importante como el respaldo es la restauración de los datos.

En phpMyadmin es posible manejar el proceso de respaldo SQL mediante la opción "Exportar", que se encuentra en el menú general en el área checkbox de "Enviar" para generar un archivo descargable (ver figura 4.2.10).

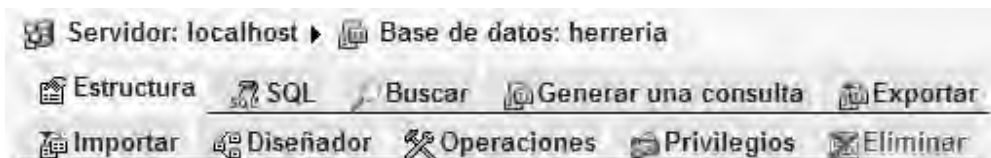


Figura 4.2.10 Menu de phpMyadmin para exportar la BD

Con esta opción el servidor exporta a un archivo de texto con extensión ".sql" que el navegador recibirá para dar la opción de "Grabar como" o "Save as" en la PC, una vez que se descargue el archivo con terminación ".sql" se podrá ver su contenido en cualquier editor de texto en caso de querer revisar el contenido de la descarga.

Finalmente después de haber creado todas las tablas es posible hacer una conexión a la base de datos "herrería" con Dreaweaver (en la siguiente sección se verá con mayor detalle) y de esta manera se podrá tener una vista general de todas las tablas involucradas, en forma de árbol.

Dicho árbol se muestra en la figura 4.2.11.

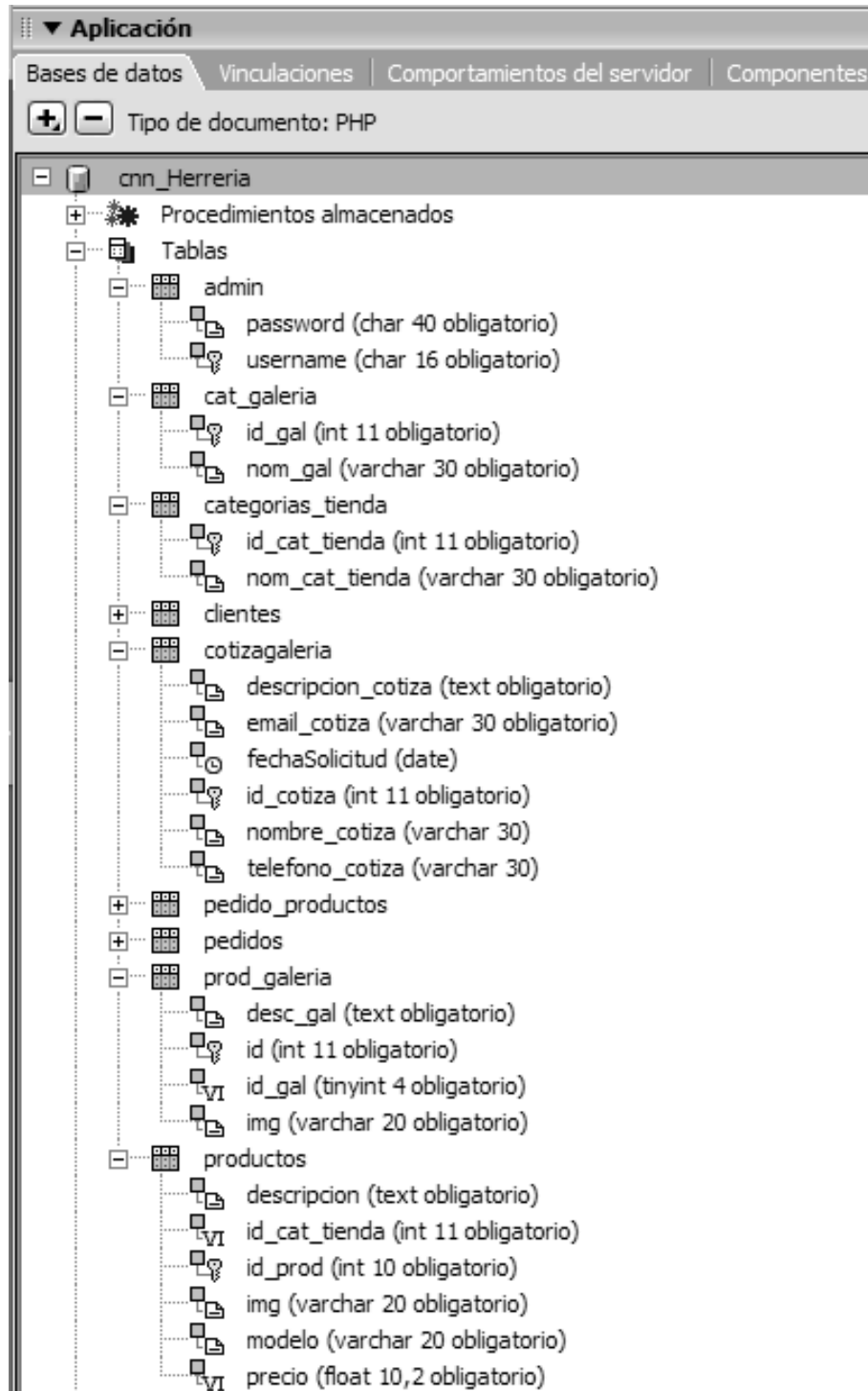


Figura 4.2.11 Árbol de tablas de la base de datos “herrería”



4.3 Instalación y configuración del sitio Web

Para comenzar con la construcción del sitio es necesario la instalación y configuración adecuada de las herramientas que se utilizaran para dicho fin.

Instalación de herramientas

Como se mencionó en capítulos anteriores se utilizará el programa Dreamweaver en su versión CS3 para la escritura de código, debido a su gran integración de herramientas y practicidad.

Por tanto se descargará una versión de prueba desde la página principal de Adobe www.adobe.com (ahora propietaria de dicha herramienta. Ver figura 4.3.1).

system requirements | Adobe Dreamweaver CS3 - Google Chrome
http://www.adobe.com/products/dreamweaver/systemreqs/

Products Solutions Learning Help Downloads Company Store Search

My support My shipments My cart Sign in

ADOBE DREAMWEAVER CS3

BUY | UPGRADE | TRY

ADOBE CREATIVE SUITE 5 DESIGN PREMIUM WEB PREMIUM PRODUCTION PREMIUM MASTER COLLECTION

System requirements

Windows

- Intel® Pentium® 4 or AMD Athlon® 64 processor
- Microsoft® Windows® XP with Service Pack 2 (Service Pack 3 recommended); Windows Vista® Home Premium, Business, Ultimate, or Enterprise with Service Pack 1; or Windows 7
- 512MB of RAM
- 1GB of available hard-disk space for installation; additional free space required during installation (cannot install on removable flash-based storage devices)
- 1280x800 display with 16-bit video card
- DVD-ROM drive
- Broadband Internet connection required for online services*

Mac OS

- Multicore Intel® processor
- Mac OS X v10.5.7 or v10.6
- 512MB of RAM
- 1.8GB of available hard-disk space for installation; additional free space required during installation (cannot install on a volume that uses a case-sensitive file system or on removable flash-based storage devices)
- 1280x800 display with 16-bit video card
- DVD-ROM drive
- Broadband Internet connection required for online services*

*This product may allow you to extend its functionality by accessing certain features that are hosted online, including CS Live online services ("Online Services"), provided you have a high-speed Internet connection. The Online Services, and some features thereof, may not be available in all countries, languages, and/or currencies and may be discontinued in whole or in part without notice. Use of the Online Services is governed by separate terms of use and by the Online Privacy Policy, and access to some services may require user registration. Some Online Services, including services that are initially offered at no charge, may be subject to additional fees and require a separate subscription. For more details and to review the applicable terms of use and Online Privacy Policy, visit www.adobe.com.

ACTIVATION
This product must be activated over the Internet prior to use. Visit the Product Activation Center to learn more about activation.

DREAMWEAVER HOME

- ▶ WHAT IS DREAMWEAVER?
- ▶ WHAT'S NEW
- ▶ IN DEPTH
 - System requirements
 - FAQ
 - Language versions
 - Reviews and awards
 - Customer stories
 - Switching to Dreamweaver
 - Dreamweaver Exchange
- ▶ FOR CORPORATE CUSTOMERS
- ▶ FOR EDUCATION CUSTOMERS
- ▶ SUPPORT AND TRAINING

Adobe Store

Figura 4.3.1 Página de descarga de la herramienta Dreamweaver CS3



Posterior a la descarga, mediante el asistente de configuración, se establecerá los parámetros básicos de instalación.

Cuando la instalación sea finalizada automáticamente se ejecutará la aplicación. En la figura 4.3.2 se muestra la vista principal de la herramienta.

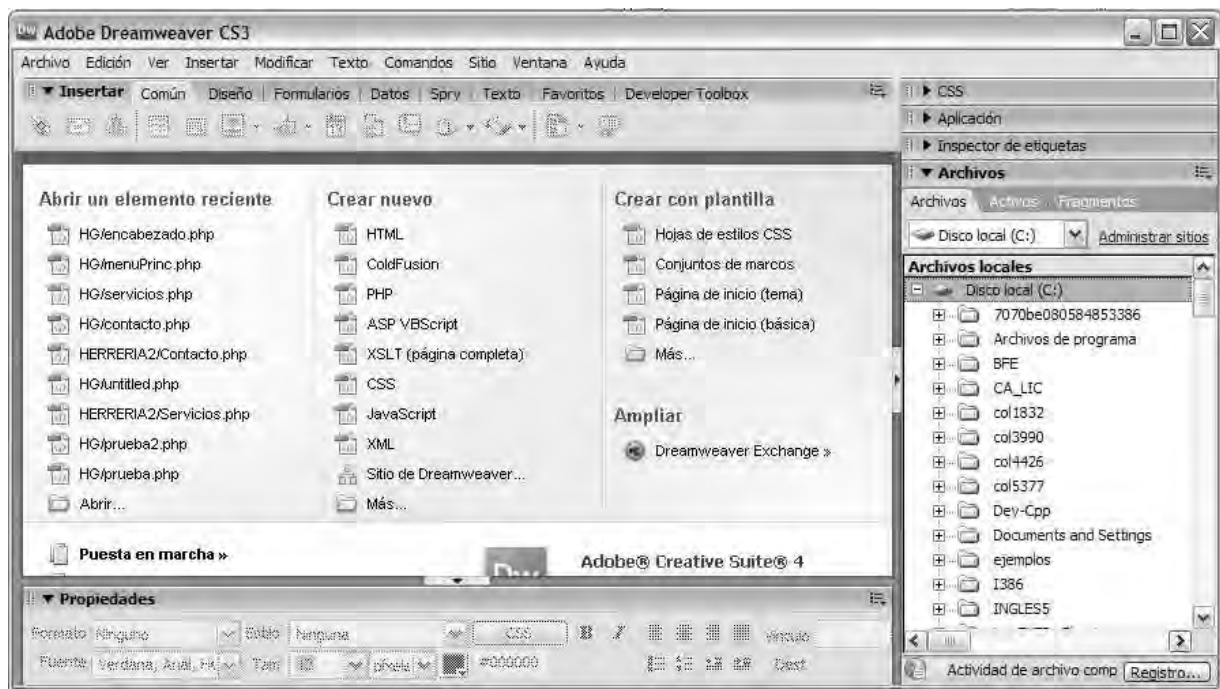


Figura 4.3.2 Vista principal de Dreamweaver CS3

Extensiones. Una de las ventajas de DW (Dreamweaver) sobre otros productos similares es que puede ampliar su funcionalidad por medio de extensiones, muchas de las cuales se puede encontrar de forma gratuita en el sitio oficial. Para instalar las extensiones es necesario el administrador de extensiones de DW, que es el que se encargará de realizar la instalación de los módulos de una forma sencilla.

Para los propósitos de la presente Tesis será necesario instalar las extensiones Developer Toolbox (para operaciones con la base de datos) y jQuery API (para la



codificación de la funcionalidad AJAX). En la figura 4.3.3 se muestra el administrador de extensiones con las extensiones ya descargadas e instaladas.

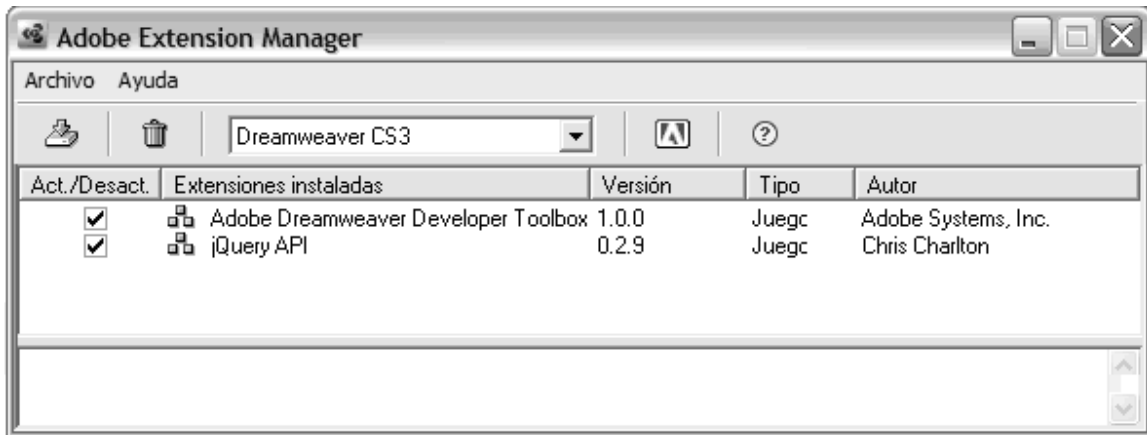


Figura 4.3.3 Administrador de extensiones de DW

Después de reiniciar el producto se muestran las herramientas adicionales en la barra de herramientas de DW (Figura 4.3.4). Solo serán visibles las herramientas de DeveloperToolbox, ya que jQueryAPI es una extensión que solo facilitará la escritura de JavaScript para el Framework jQuery por medio de listas de sugerencias de su sintaxis.



Figura 4.3.4 Herramientas para la extensión Developer Toobox.

La siguiente herramienta que se empleará será XAMPP, su instalación y utilidad se ha explicado en el apartado del Back-End, por tanto en este momento solo se iniciará dicha aplicación y se verificará que este funcionando correctamente.



Creación del sitio

Después de haber instalado adecuadamente las herramientas descritas anteriormente, se procederá a crear el sitio Web. Un sitio Web es llanamente una carpeta que contendrá los archivos para su funcionamiento.

El método generalmente más usado para crear un sitio Web utilizando DW consiste en: (1) Crear y editar páginas en el disco local, guardándolas en nuestro propio equipo. (2) Después conectarse a un servidor Web remoto donde se alojarán. (3) Cargar una copia del sitio Web y los documentos (páginas, imágenes, js, etc.) en el servidor para ponerlas a disposición pública y (4) que el servidor los “sirva”. (Ver Figura 4.3.5)

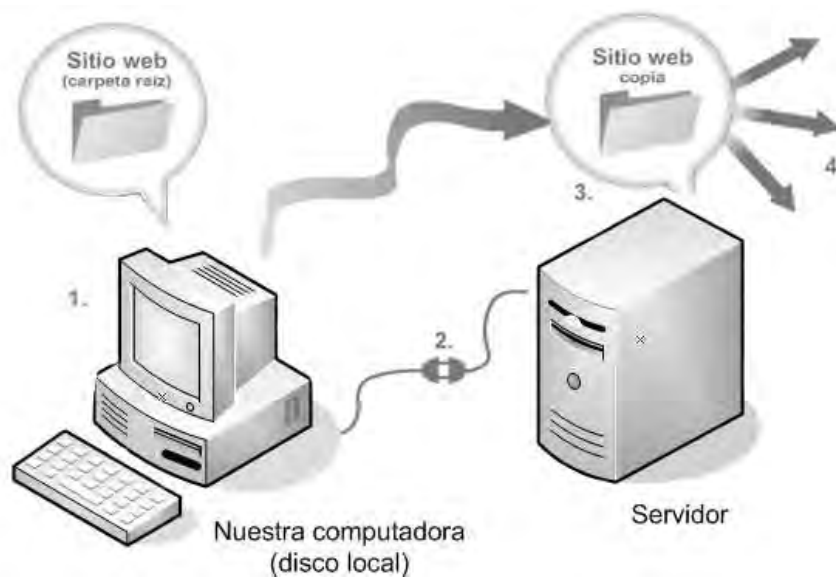


Figura 4.3.5 Método típico para la creación de sitios Web

Por tanto lo primero que se realizará es crear la carpeta del sitio que contendrá los archivos necesarios para después subirlos al servidor. Dicha carpeta se denomina carpeta raíz.



La carpeta del sitio se llamará “HG” (*Herrería García*), esta carpeta se creará en la raíz de la carpeta principal “*htdocs*” del servidor de prueba que se configuró previamente en la instalación de de XAMPP. (Ver Figura 4.3.6).

El nombre de la carpeta no necesariamente tiene que relacionarse con el nombre del sitio que se configurará posteriormente en Dreamweaver.

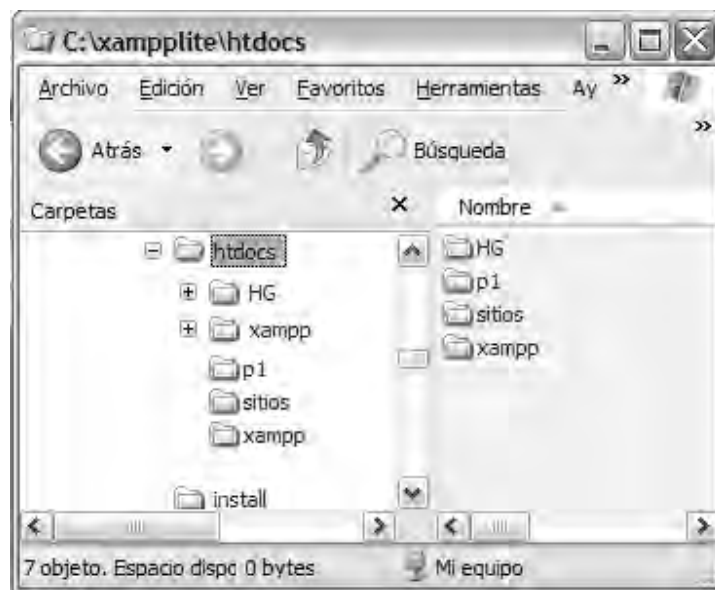


Figura 4.3.6 Creación de la carpeta del Sitio Web

Después de crear la carpeta se configurará propiamente el sitio en Dreamweaver mediante los siguientes pasos:

- ✦ Localizar la carpeta raíz que se creó previamente.
- ✦ Abrir la herramienta de administración de sitios de DW (Ver Fig. 4.3.7).
- ✦ Configurar el sitio





Figura 4.3.7 Ubicación del administrador de sitios

Configuración de datos locales. Se elige el nombre, en este caso se le llamará de la misma forma que la carpeta raíz, posteriormente se ubica la carpeta raíz local e introduce la dirección http para poder acceder a esta ubicación desde el navegador. (Ver Fig. 4.3.8).

Datos locales

Nombre del sitio:

Carpeta raíz local: 

Carpeta predeterminada de imágenes: 

Vínculos relativos a: Documento Raíz del sitio

Dirección HTTP:

Figura 4.3.8 Configuración de los datos locales

Configuración de datos remotos. Se configura el tipo de acceso por el cual se realizará el enlace al sitio, en este caso por ser servidor de prueba local será por medio de la red intranet (opción Local/red). Y se ubica la carpeta remota (en el servidor local) que contendrá al sitio (Ver Fig. 4.3.9).



Datos remotos

Acceso: Local/red

Carpeta remota: C:\xampplite\htdocs\HG\

Mantener información de sincronización

Cargar archivos en el servidor automáticamente al guardar

Permitir desproteger y proteger archivo

Figura 4.3.9 Configuración de los Datos remotos

Servidor de prueba. Se elige el tipo de servidor con base a los lenguajes y sistema de archivos que soportará. En este caso se elige la opción de PHP y MySQL con acceso por red (Ver Fig. 4.3.10).

Modelo de servidor: PHP MySQL

Acceso: Local/red

Carpeta de servidor C:\xampplite\htdocs\HG\

Figura 4.3.10 Configuración del Servidor de prueba

Esta configuración será suficiente para los propósitos del presente trabajo. Después de haber realizado el procedimientos anteriores aparecerá en el panel Archivos de DW la carpeta raíz para trabajar con los archivos del sitio (por defecto estará vacía). En otras palabras, este sitio en DW sencillamente será un acceso a la carpeta del sitio.

4.4 Diseño y construcción del front-end.

Después de haber diseñado y construido el back-end y para poder obtener su utilidad se procederá a realizar el Front-end, que será el que interactuará con todo los datos anteriormente descritos.



4.4.1 Diseño del Fron-end

Se procederá a realizar un bosquejo en papel de todo el sitio Web así como de los paneles que se desean, todos estos con base en los requisitos anteriormente planteados.

En la figura 4.4.1.1 se muestran los paneles que contendrá la página principal, sobre la cual se implementarán los módulos respectivos.

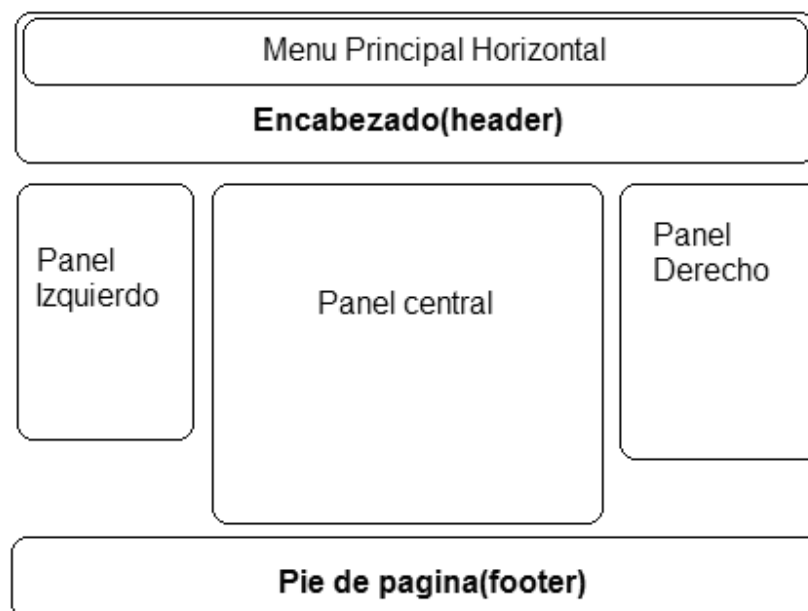


Figura 4.4.1.1 Paneles principales del sitio Web

Se contará con cinco paneles principales, cada panel mostrará de forma asíncrona las diferentes páginas o archivos PHP correspondientes que se llamarán cuando sean necesarios mediante enlaces AJAX para evitar el recargo de la pagina completa.

La página principal de la empresa contará con las siguientes secciones:

- ✦ Inicio
- ✦ Tienda en línea



- ✦ Galería
- ✦ Diseño y Cotización en línea
- ✦ Quiénes somos
- ✦ Servicios
- ✦ Contacto

A continuación se bosquejará la pantalla que se espera visualizar en cada sección del sitio. Para fines del presente trabajo solo se mostraran y explicarán los diseños de la sección de la Tienda en línea, la galería de imágenes de otros productos que no es posible vender en línea y la sección interactiva con el usuario donde se podrá obtener una cotización en tiempo real de algún producto o trabajo.

Tienda en línea

En la Figura 4.4.1.2 se muestra la pantalla que se pretende visualizar cuando se elija la tienda en línea.

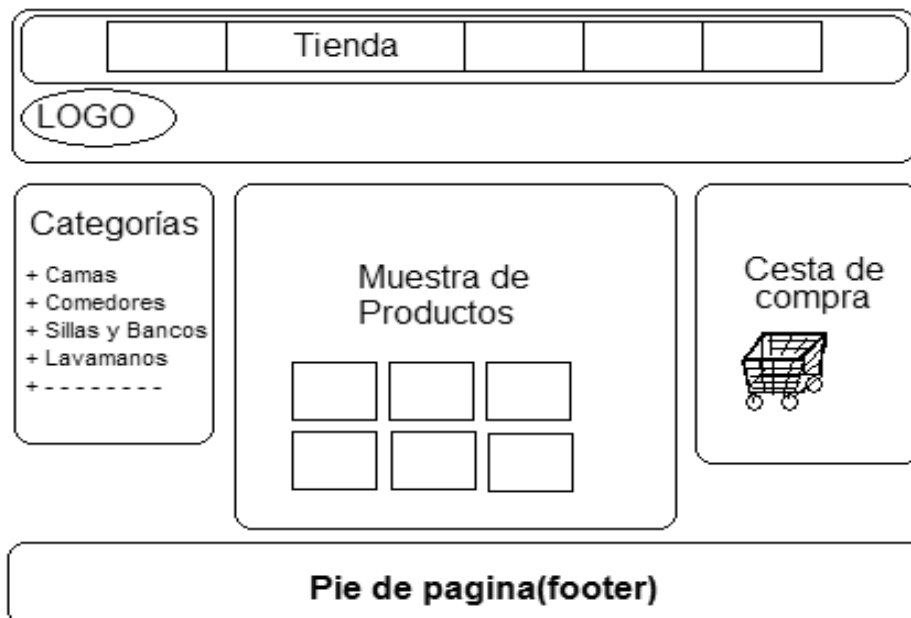


Figura 4.4.1.2 Pantalla principal para la tienda en línea



En el Panel Izquierdo se mostrará la lista de categorías de los productos en venta, cada elemento de la lista será un vínculo, que al pulsarlo desplegara las imágenes de los productos de la categoría elegida dentro del panel central, al mismo tiempo en el panel derecho aparecerá la cesta de compra o carrito de compras donde de forma visual el usuario siempre tendrá conocimiento de los artículos que va ingresando.

Galería

Existen productos o trabajos que la empresa desea se muestren a los usuarios a manera de galería de imágenes, esta galería también se mostrará por categorías, una diferencia con el mecanismo de la tienda es que en esta galería se desea visualizar las imágenes en un tamaño mayor para mejor detalle, y además se ha de agregar un formulario de contacto para que el usuario pueda solicitar alguna cotización de algún producto que visualiza en la galería. (Ver figura 4.4.1.3)

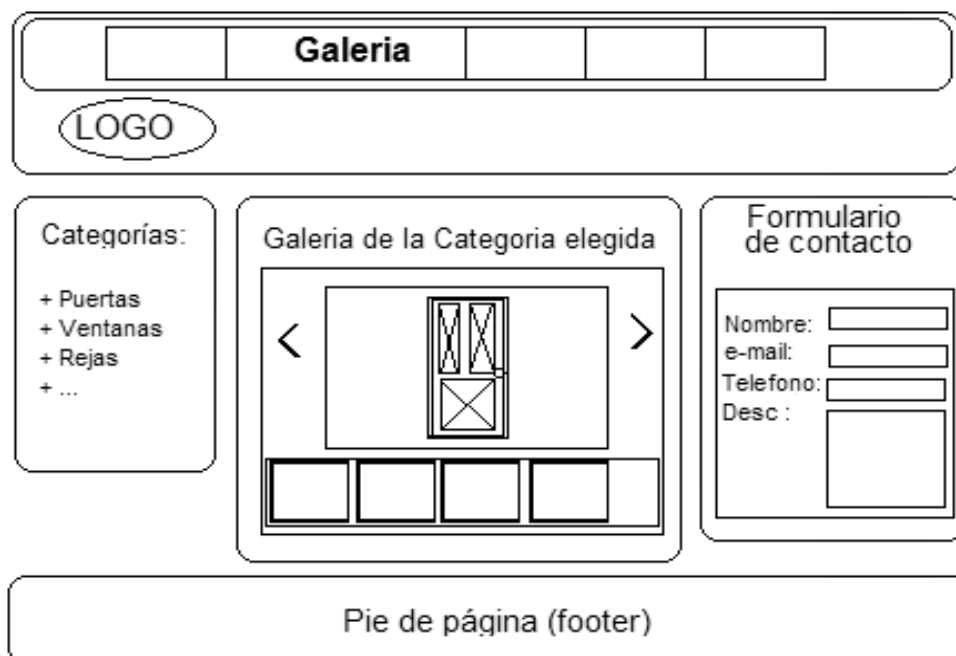


Figura 4.4.1.3 Pantalla principal para la Galería de imágenes



Diseño y cotización en línea

Este apartado es una propuesta a la necesidad de solicitud de cotización en línea en tiempo real. Para algunos productos la cotización de un producto depende del área requerida del producto final, por tanto se espera que el mecanismo permita al usuario modificar las dimensiones y por consiguiente el área de ciertos artículos, y de esta manera obtener visualmente el costo en tiempo real (Ver Figura 4.4.1.4).

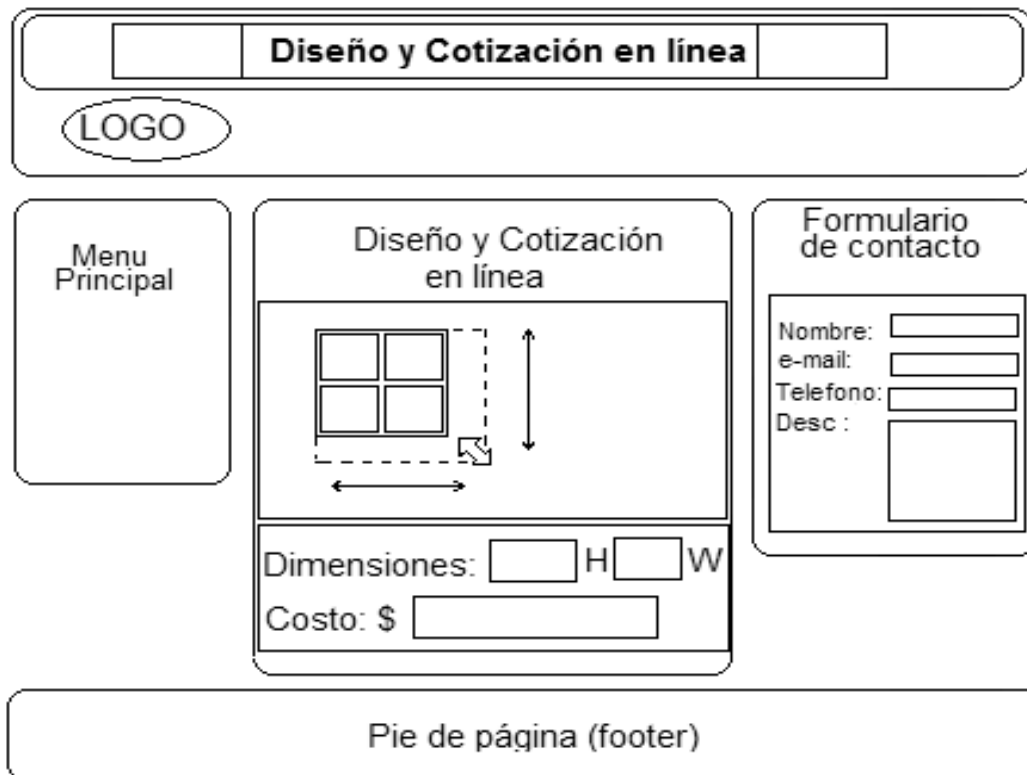


Figura 4.4.1.4 Pantalla principal para el diseño y cotización de productos en línea

Además el usuario contará con la sección de un Formulario de contacto para solicitar en cualquier momento una cotización para las necesidades particulares del cliente. Para las secciones restantes (Quiénes somos, Servicio y Contacto) solo se utilizará información de presentación de la empresa que se mostrará en el Panel Central.



4.4.2 Construcción del Front-End

✦ Maquetado

A continuación se realizará el maquetado de la página principal con base en el bosquejo que se realizó en el inicio de la sección.

Por motivos de practicidad se elegirá una plantilla entre las preconfiguradas de DW que cumpla lo más posible con los propósitos buscados en cuando a maquetación y distribución de las secciones (Ver figura 4.4.2.1). Ésta solo nos servirá como guía en cuanto a distribución (en píxeles) de una visión general de la página. Posteriormente se adecuará a las necesidades particulares.

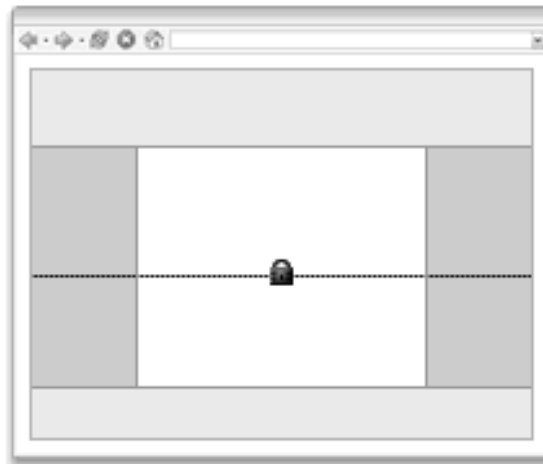


Fig. 4.4.2.1 Plantilla DW de tres columnas fijas, con encabezado y pie.

Así mismo para mantener la separación de la lógica y la presentación se trabajará a parte con un archivo de diseño (CSS) que se llamará “*estilos.css*”, que de igual manera se guardará en la carpeta raíz. Posteriormente se hará referencia a este archivo en la página principal del sitio o las páginas que requieran de este diseño.

4.4.2.1 Tienda en línea

A continuación se describirá la manera en la que se realizarán los módulos o secciones de la tienda en línea.



Cada módulo será un archivo o conjunto de archivos que se guardarán en la carpeta raíz.

✦ Generación del catálogo en línea

Para poder implementar el catálogo de los productos que se venderán en línea, primero será necesario mostrar la lista de las categorías de los productos que se desean mostrar.

Esta lista de categorías se obtendrá de la base de datos previamente creada, para lo cual será necesario establecer una conexión a la base de datos llamada “herrería”.

Conexión a la base de datos

Esta conexión a la base de datos se realizará con ayuda de las herramientas básicas de Dreamweaver.

En la sección de Bases de datos del panel Aplicación, al pulsarlo abrirá un cuadro de diálogo donde se podrán establecer los parámetros necesarios para la conexión. (Ver figura 4.4.2.1.1)



Figura 4.4.2.1.1 Establecimiento de los parámetros de conexión



Después de haber pulsado el botón aceptar generará de forma automática una carpeta con un archivo PHP con el nombre de la conexión que contendrá los parámetros necesarios para llevar a cabo dicha conexión.

El contenido del archivo se muestra en el listado 4.4.2.1.1.

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_cnn_Herreria = "localhost";
$databse_cnn_Herreria = "herreria";
$username_cnn_Herreria = "root";
$password_cnn_Herreria = "";
$cnn_Herreria = mysql_pconnect($hostname_cnn_Herreria,
$username_cnn_Herreria, $password_cnn_Herreria) or
trigger_error(mysql_error(),E_USER_ERROR);
?>
```

Listado 4.4.2.1.1 Código generado del archivo *cnn_Herreria.php*

Con base en el archivo anterior se llevarán a cabo todas las interacciones con la base de datos, por lo tanto cualquier archivo que requiera algún conjunto de registros de la base de datos deberá hacer referencia a este archivo con la siguiente línea de código:

```
<?php require_once('Connections/cnn_Herreria.php'); ?>
```

Listado de categorías

Para este módulo se creará primero un archivo llamado *MenuTienda.php* que generará el listado de todas las categorías de los productos.



Los nombres para dicha lista se obtendrán de la base de datos, específicamente de la columna `nom_cat_tienda` de la tabla `categorias_tienda` que se encuentra en la base de datos `herreria`.

Estando en el archivo `MenuTienda.php` será necesario crear un juego de registros que se llamará `rsCategoriasTienda` para poder llevar a cabo interacciones (consultas) con la base de datos de forma más práctica. Para poder generar dicho juego de registros se utilizará la herramienta “Juego de registros” la cual se accederá mediante Aplicación>Vinculaciones>Juego de registros. Tras pulsar dicha opción mostrará un cuadro de diálogo para introducir los parámetros de conexión. (Ver figura 4.4.2.1.2)



Figura 4.4.2.1.2 Parámetros del juego de registros

Después de aceptar se agregará el juego de registros en el panel vinculaciones como se muestra en la figura que permitirá utilizar los valores de los registros de la tabla.



Esta gran herramienta ofrece la posibilidad de utilizar los datos de la tabla con solo arrastrar desde este árbol gráfico los registros y soltarlos en el área de la página donde se desea aparezca (ver figura 4.4.2.1.3).



Figura 4.4.2.1.3 Árbol del juego re registros

Realmente lo que hace esta herramienta de interfaz gráfica para obtener los datos de BD es ejecutar la consulta sql: “SELECT * FROM categorias_tienda”, con la cual se obtienen todas las columnas de la tabla *categorias_tienda*. Mediante el código del listado 4.4.2.1.2.

```
mysql_select_db($database_cnn_Herreria, $cnn_Herreria);
$query_rsCategoriasTienda = "SELECT * FROM categorias_tienda";
$rsCategoriasTienda = mysql_query($query_rsCategoriasTienda,
    $cnn_Herreria) or die(mysql_error());
$row_rsCategoriasTienda =
    mysql_fetch_assoc($rsCategoriasTienda);
$totalRows_rsCategoriasTienda =
    mysql_num_rows($rsCategoriasTienda);
```

Listado 4.4.2.1.2 Código para la conexión a la BD

Que se generará en el la sección <header> de la pagina *MenuTienda.php*. Así mismo generará en la sección <body> el código siguiente:



```
<?php echo $row_rsCategoriasTienda['nom_cat_tienda']; ?>
```

Donde se observa que con el comando *echo* de php se muestra en el navegador el primer valor contenido en el arreglo `$row_rsCategoriTienda` en el que están almacenados los nombres de las categorías.

Con lo anterior se podrá obtener el nombre de la primera categoría, por tanto será necesario hacer un ciclo *do while* para obtener todos los valores de los registros.

Antes de la ejecución del ciclo *do while* se añadirán las etiquetas `<div></div>` para indicar al navegador que no muestre los elementos en forma lineal sino que los muestre en forma de listado vertical. Quedando el código como se muestra en el listado 4.4.2.1.3.

```
<?php do { ?>
<div><?php echo $row_rsCategoriasTienda['nom_cat_tienda'];
?></div>
<?php } while ($row_rsCategoriasTienda =
mysql_fetch_assoc($rsCategoriasTienda)); ?>
```

Listado 4.4.2.1.3 Código del ciclo *do while* para obtener todos los registros.

Del listado 4.1.2.1.2 se observa la operación siguiente: se muestra en pantalla cada elemento del arreglo `$row_rsCategoriasTienda` mientras existan elementos que por medio del comando `mysql_fetch_assoc` devuelva la matriz asociativa que corresponde a la fila recuperada y mueve el apuntador de datos interno hacia adelante.

La forma más rápida de obtener la operación anterior es seleccionar el registro, arrastrado y aplicarle una región de repetición, desde el panel de Aplicación (Ver Figura 4.4.2.1.4).

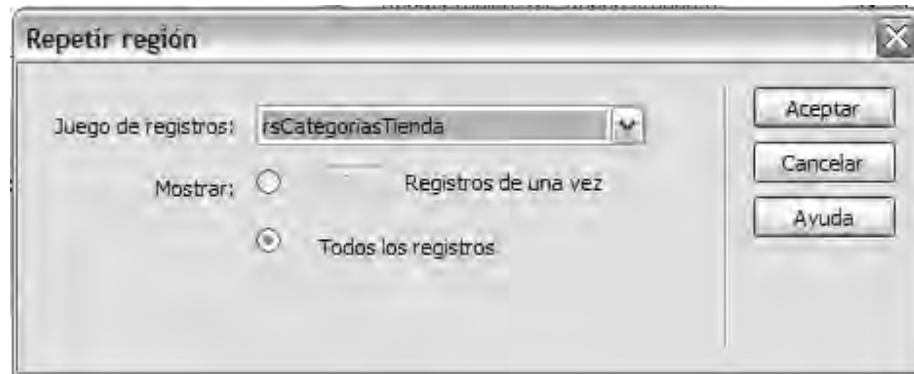


Figura 4.4.2.1.4 Región de repetición

De esta manera se ha obtenido la lista completa de todas las categorías de los productos desde la base de datos, que en el navegador se mostrara como se muestra en la Figura 4.4.2.1.5.



Figura 4.4.2.1.5 Vista en el navegador de la lista de categorías generada.



Generación del catálogo de productos

Para este módulo se creará un nuevo archivo llamado *CatalogoTienda.php*. Antes de la generación del listado de los productos se obtendrá un título dinámico, que se mostrará en la parte superior del área donde se desplegarán los productos la categoría de los productos que se hayan seleccionado de la lista de categorías del módulo anterior, se recurre a la misma operación anterior (generación de juego de registros), pero ahora aplicando un filtro para obtener el nombre de la categoría con base en su identificador único, en este caso la columna *id_cat_tienda* tendrá que ser igual al parámetro URL *id_cat_tienda*. Como se indica en la Figura 4.4.2.1.6.



Figura 4.4.2.1.6 Configuración del Juego de registros con filtro

Para obtener el desplgado de las imágenes de los productos de cada categoría se necesitará nuevamente de un juego de registros ya que se requerirán de los datos de la columna *img* de la tabla productos.



Este nuevo juego de registros de igual forma será filtrado por medio del parámetro *id_cat_tienda* que se haya seleccionado en el archivo *MenuTienda.php*.

El proceso para listar los productos de cada categoría es similar al proceso del listado de categorías. Pero ahora no será necesario mostrar una lista de texto donde se muestren los modelos de los productos de cierta categoría, sino que, de acuerdo a los requerimientos establecidos previamente, será necesario obtener las imágenes de los productos de la categoría elegida y posteriormente en un siguiente proceso se obtendrán los detalles de los productos, por tanto de manera similar se insertará una imagen donde elegiremos como origen el registro *img* del conjunto de registro previamente obtenidos, así como la ubicación física de la carpeta donde se obtendrán las imágenes (ver Figura 4.4.2.1.7).

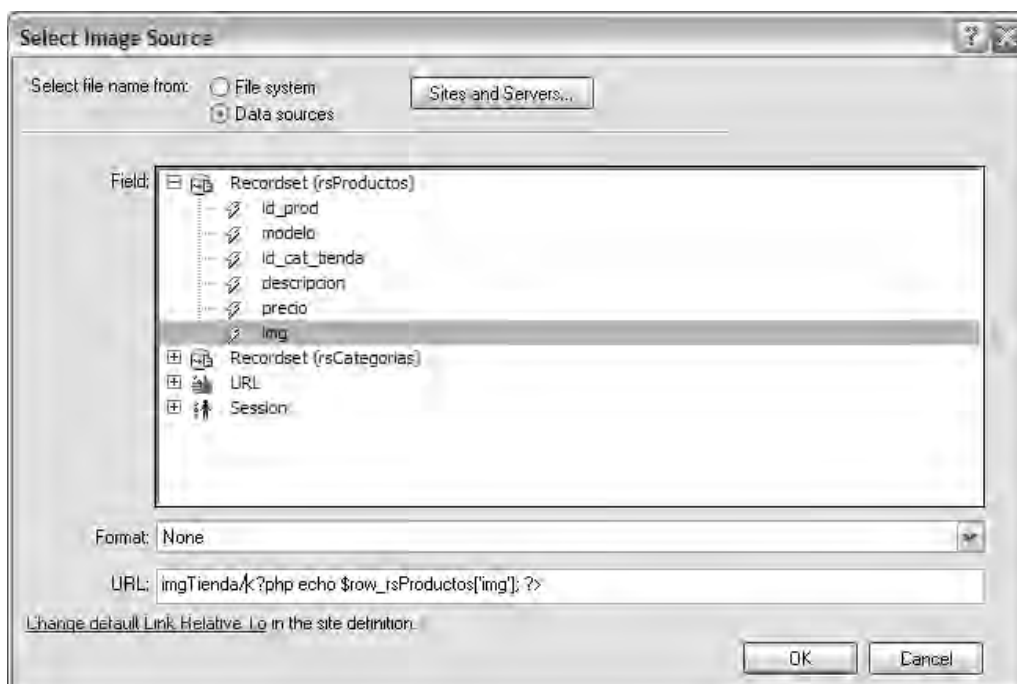


Figura 4.4.2.1.7 Selección del origen de la imagen

Posteriormente de igual forma se le aplicará una región de repetición para obtener todos los productos de la categoría deseada.



De esta manera y de forma automática se habrá generado el código del listado 4.4.2.1.4.

```
<?php do { ?>
  
<?php } while ($row_rsProductos =
mysql_fetch_assoc($rsProductos)); ?>
```

Listado 4.4.2.1.4 Código para generar el listado de imágenes

Donde se observa que el nombre de la imagen va cambiando en tanto haya elementos dentro de la columna *img* de la tabla productos de la base de datos. El enlace entre el archivo MenuTienda.php y CatalogoTienda.php se hará mediante enlaces AJAX, los cuales se explicarán posteriormete. Al visualizar el archivo CatalogoTienda.php en el navegador (ver Figura 4.4.2.1.8) requerirá el parámetro de la categoría por lo que para probar lo ingresaremos directamente en la URL (http://localhost/HG/CatalogoTienda.php?id_cat_tienda=2).



Figura 4.4.2.1.8 Vista en el navegador de las imágenes de los productos



Es así como se ha generado el catálogo de los productos listados por categoría, a continuación se describirá el proceso de la obtención de los detalles de cada producto.

Mostrar detalles de producto

Con base en los requerimientos será necesario contar con una interfaz intuitiva al usuario para la obtención de los detalles un producto.

Se propone la utilización de lo que se denomina “*tooltips*”, que no es mas que una descripción emergente, siendo una herramienta de ayuda visual que funcionará al situar el ratón sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de los detalles del producto como: el precio, la descripción y el modelo del elemento sobre el que se encuentra, proporcionando esta información adicional sin necesidad de que el usuario la solicite.

Esto se logrará con ayuda del Framework jQuery que es el conjunto de librerías de JavaScript para poder simplificar la manera de interactuar con los documentos HTML, manipular el DOM (Document Objet Model), manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX en las páginas Web. Por lo tanto se agregará las siguientes líneas de código al archivo php donde se desea dicha funcionalidad, dentro de las etiquetas <header>, para hacer referencia a estas librerías (Ver Listado 4.4.2.1.5).

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.7.2/jquery-
ui.min.js"></script>
<script type="text/javascript" src="simpletip/jquery.simpletip-
1.3.1.pack.js"></script>
```

Listado 4.4.2.1.5 Código para referenciar las librerías jQuery



Donde observamos que las dos primeras hacen referencia a las librerías que utilizará *jquery.simpletip-1.3.1.pack.js* que será la que ofrecerá la funcionalidad del *tooltip*.

Como puede apreciarse, las dos primeras están alojadas en servidores de Google, de esta manera se indica que se hará uso de dichas librerías, en línea, esta opción es una alternativa para evitar el almacenamiento de éstas en el servidor local o donde se encuentre hospedado el sitio. A continuación también se agregará la siguiente línea en la sección de la página llamada header:

```
<script type="text/javascript" src="script.js"></script>
```

Esta línea hace referencia al archivo JavaScript que será usado en el sitio y que contendrá las funciones propias necesarias para el funcionamiento de los elementos dinámicos del sitio. Es en este archivo donde se utilizarán las funciones descritas en *jquery.simpletip-1.3.1.pack.js*. La idea fundamental de la programación AJAX es el manejo del DOM para poder acceder a cada elemento de la página y operar de forma asíncrona con el servidor. En el archivo "script.js" la primera función a la que se hace referencia (Ver listado 4.4.2.1.6) es la función `$(document).ready(function () {...})` que tiene la finalidad de ejecutar las instrucciones englobadas en los corchetes cuando

```
$(document).ready(function(){
    $('.product').simpletip({
        offset:[40,0],
        content:'',
        onShow: function(){
            var param = this.getParent().find('img').attr('src');
            if($.browser.msie && $.browser.version=='6.0')
            {
                param = this.getParent().find('img').attr('style').
                    match(/src=\"([^\"]+)\"/);
                param = param[1];
            }
            this.load('ajax/tips.php', {img:param});
        }
    });
});
```

Listado 4.4.2.1.6 Código para la funcionalidad tooltip



el DOM o los elementos que se necesiten del DOM estén listos o descargados en el navegador del usuario. Por lo tanto es importante identificar cada elemento de la página con un respectivo identificador único o con el nombre de una clase. De esta manera se usará una clase llamada *product* la cual se le aplicará la funcionalidad `simpletip()`. Se cargará además una imagen gif animada que se mostrará cuando los datos estén en transferencia, para indicar al usuario que se esta llevando acabo una comunicación con el servidor.

También del listado se observan dos de las partes más importantes que son las funciones: `this.getParent().find('img')` y `this.load('ajax/tips.php', {img:param});` cuya finalidad es cargar el archivo *tips.php* con referencia al parámetro 'img', que será la imagen del producto al que se aplica.

En el listado 4.4.2.1.7 se detalla el archivo *tips.php*

```
<?php
define('INCLUDE_CHECK',1);
require "../connect.php";
if(!$POST['img']) die("No hay descripcion para este
producto!");
$img=mysql_real_escape_string(end(explode('/',$_POST['img'])));
$row=mysql_fetch_assoc(mysql_query("SELECT * FROM productos
WHERE img='". $img. "'"));
if(!$row) die("No hay descripcion para este producto!");
echo '<strong>'. $row['modelo']. '</strong>
<p class="descr">'. $row['descripcion']. '</p>
<strong>Precio: $'. $row['precio']. '</strong>
<small>Arrastrelo al carrito para comprarlo</small>';
?>
```

Listado 4.4.2.1.7 Código del archivo *tips.php*

Donde se observa que si se logra obtener el valor de la *form variable* 'img' se hará una matriz asociativa con los valores del modelo, descripción y precio obtenidos de la consulta filtrada "SELECT * FROM productos WHERE img='". \$img. "'", hecha



a la base de datos. Con esto es suficiente para que funcione la transferencia de datos necesarios para el *tooltip*, lo que restaría es darle presentación a la información, esto se logrará con la definición de los parámetros de la clase “*tooltip*” en la hoja de estilos CSS a la que está ligada la página principal. En el listado 4.4.2.1.8 se detalla la definición de la clase *.tooltip* en la hoja de estilos.

```
.tooltip{  
  position: absolute;  
  top: 0;  
  left: 0;  
  z-index: 3;  
  display: none;  
  background-color:#666666;  
  border:1px solid #666666;  
  color:#fcfcfc;  
  padding:10px;  
  -moz-border-radius:12px;  
  -khtml-border-radius: 12px;  
  -webkit-border-radius: 12px;  
  border-radius:12px;  
}
```

Listado 4.4.2.1.8 Código CSS para la clase *.tooltip*

De esta manera es como se pone en funcionamiento el *tooltip* con los detalles del producto que será accionado al situar el ratón sobre alguna imagen del producto (Ver figura 4.4.2.1.9).



Figura 4.4.2.1.9 *Tooltip* en funcionamiento



✚ Carro de compra

Esta sección será un tanto más compleja ya que no se optará por un proceso de carro de compra tradicional en el que hay refrescos de página para agregar artículos al carro de compra, sino que al pasar de una sección a otra será manejado con procesos AJAX para brindar una mayor usabilidad de la tienda online de forma más clara y visual para el usuario.

Se usará la funcionalidad **drag and drop** (arrastrar y soltar) para arrastrar aquellos artículos que se desean comprar a la “cesta de la compra”.

El usuario además de interactuar en la Web de forma novedosa y vistosa, podrá ver en todo momento el resultado de sus acciones, pudiendo modificar o consultar sus compras sin necesidad de volver a cargar la página o de abandonarla. Esto tiene la ventaja de mantener actualizado el precio total de sus compras, ya sea cuando añade o cuando elimina un producto.

A continuación se describe de forma general la metodología empleada para el proceso de compra. Con base en el procedimiento descrito anteriormente para el tooltip, en ese mismo archivo scrip.js se agregarán otras funciones que permitirán obtener las funcionalidades deseadas.

Funcionalidad Drag. Para hacer el efecto de *drag* o arrastre de los artículos se recurrirá a la función *draggable()* del framework mencionado anteriormente (ver listado 4.4.2.1.9), esta se aplicará con sus respectivos parámetros de configuración, al elemento *img* que se encuentra dentro de los *<div>* que contienen la clase *product*.

```
$(".product img").draggable({
    containment: 'document',
    opacity: 0.6,
    revert: 'invalid',
    helper: 'clone',
    zIndex: 100
});
```

Listado 4.4.2.1.9 Código para el efecto *drag*



Por tanto será necesario incluir esta clase al código que obtuvimos anteriormente para desplegar las imágenes de la categoría seleccionada (ver listado 4.4.2.1.10).

```
<?php do { ?>
    <div class="product"></div<?php } while
($row_rsProductos = mysql_fetch_assoc($rsProductos));
?>
```

Listado 4.4.2.1.10 Código de la generación de las imágenes en que se aplica el efecto

Funcionalidad Drop De forma análoga se hará para la funcionalidad del *drop* (soltar). La función asociada es *droppable()* la cual del mismo modo se aplicará al área donde deseamos sea “soltada” la imagen del producto. El área que se habrá de establecer para ser empleada dicha funcionalidad estará establecida por las etiquetas `<div>` en las que se aplicará la clase `content drop-here`, dentro de esa sección `div` se encontrará la imagen de un carro de compras tradicional para indicar al usuario donde se encuentra el área *droppable*. Esta imagen del carro de compras se encontrará en el archivo `cesta.php` que se configurará posteriormente para ser mostrado mientras se despliega el catálogo de productos, de tal manera que el usuario pueda tenerlo siempre visible mientras hace las elecciones de sus productos.

En el listado 4.4.2.1.11 se muestra el código empleado para dicha funcionalidad.

```
$( "div.content.drop-here" ).droppable({
  drop:
    function(e, ui)
    {
      var param = $(ui.draggable).attr('src');
      if ($.browser.msie && $.browser.version=='6.0')
      {
        param =
          $(ui.draggable).attr('style').match(/src=\"([^\"]+)\\"/);
      };
      param = param[1];
    }
    addlist(param);
  });
```

Listado 4.4.2.1.11 Código para el efecto drop



Añadir artículo y mostrar contenido de la cesta

Del listado 4.1.2.1.11 en la ultima parte se hace referencia a la función *addlist()* que está definida posteriormente, es decir cuando se lleve acabo el evento de “soltar” se disparará la llamada a esta función, por motivos de espacio no se incluirá en el presente trabajo sino en los archivos adjuntos.

Lo importante a destacar de la función *addlist()* es el código del listado 4.4.2.1.12

```
for(var i=0; i<purchased.length;i++)
{
    if(purchased[i].id_prod==msg.id_prod)
    {
        check=true;
        cnt=purchased[i].cnt;
        break;
    }
    if(!cnt)
        $('#item-list').append(msg.txt);
    if(!check)
    {
        purchased.push({id_prod:msg.id_prod,cnt:1,precio:msg.precio});
    }
    else
    {
        if(cnt>=3) return false;
        purchased[i].cnt++;
        $('#'+msg.id_prod+'_cnt').val(purchased[i].cnt);
        totalprice+=msg.precio;
        update_total();
    }
}
```

Listado 4.4.2.1.12 Código para añadir un artículo y mostrar el contenido de la cesta.

donde se encuentra el mecanismo que hace que cuando se suelte la imagen sobre el elemento *draggable()* se active el ciclo *for* para que se incremente en uno la cantidad de los artículos que se van comprando así mismo se irá actualizando el valor del precio total, todo esto prácticamente de forma automática, mostrándolo en el elemento *#item.list* con la función *append()*.



De igual manera su definirá la función *remove()* y *change()* para darle al usuario la opción de eliminar algún artículo del carro de compra del cual se haya arrepentido en comprar, o cambiar de forma manual el número de productos que esta adquiriendo, actualizando con la función *update_total()* el valor del precio total, el número y tipo de los productos dentro del carro. En el listado 4.4.2.1.13 se muestra la implementación de dichas funciones.

```
function remove(id_prod)
{
    var i=findpos(id_prod);
    totalprice-=purchased[i].precio*purchased[i].cnt;
    purchased[i].cnt = 0;
    $('#table_'+id_prod).remove();
    update_total();
}

function change(id_prod)
{
    var i=findpos(id_prod);

totalprice+=(parseInt($('#'+id_prod+'_cnt').val())-
purchased[i].cnt)*purchased[i].precio;
purchased[i].cnt=parseInt($('#'+id_prod+'_cnt').val())
;
update_total();
}

function update_total()
{
    if(totalprice)
    {
        $('#total').html('total: $'+totalprice);
        $('a.button').css('display','block');
    }
    else
    {
        $('#total').html('');
        $('a.button').hide();
    }
}
```

Listado 4.4.2.1.13 Código para las funciones básicas de la compra.



Es importante destacar que cuando la variable `totalprice` contiene algún valor, se muestra el botón para ofrecer al usuario pasar a la caja.

```
$('#a.button').css('display','block');
```

La función `addlist()` anteriormente mostrada devuelve el número y tipo de productos así como el costo total de la compra, estos valores son usados por el archivo `addtocart.php` para mostrar al usuario dicha información.

Aplicando posteriormente hojas de estilo CSS se podrá ver la cesta como se muestra en la figura 4.4.2.1.10.

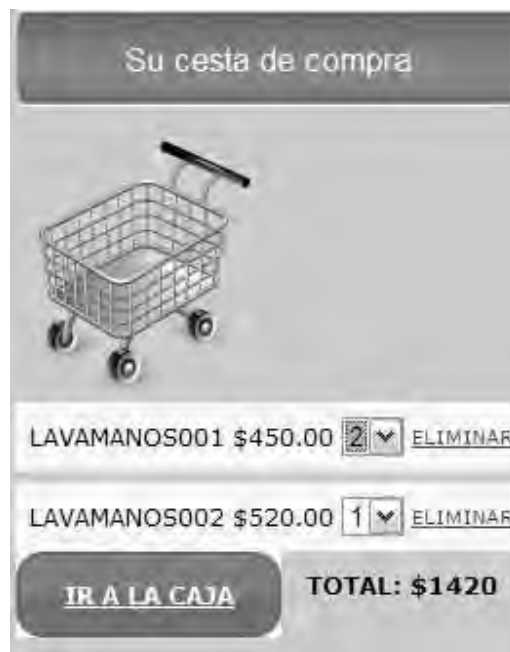


Figura 4.4.2.1.10 Cesta de compra en ejecución

Imprimir un resumen

Al evento clic sobre el botón “*ir a la caja*” los valores guardados temporalmente serán enviados al archivo `order.php` para mostrar al usuario un resumen final de su compra confirmando su pedido, para posteriormente realizar el pago.



El archivo *order.php* lo que hace esencialmente es presentar los valores del código JavaScript. El resultado de este archivo se muestra en la figura 4.4.2.1.11.

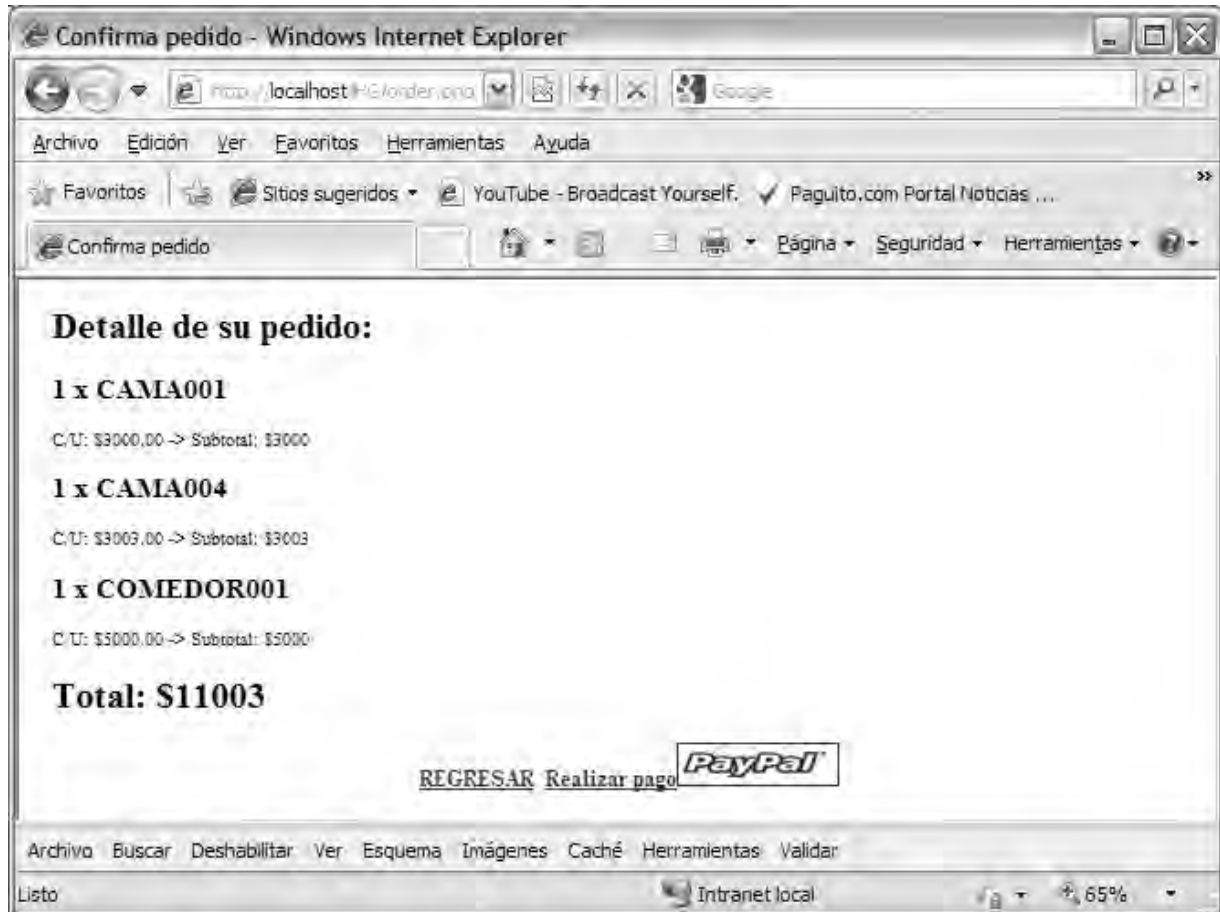


Figura 4.4.2.1.11 Vista en el navegador del resumen del pedido

Debido a los propósitos del presente trabajo no se implementará el proceso de pago. Existen una variedad de sistemas de pago debidamente autorizados para las transacciones monetarias. Cada una ofrece soporte para la implementación de su sistema en los sitios Web que opten por su servicio.

4.4.2.2 Galería de imágenes

Otro de los objetivos que se pretenden alcanzar es la promoción de los artículos y los tipos de servicios que ofrece la PYME por tanto no será suficiente con mostrar los



productos de la tienda en línea, ya que no todos los productos pueden venderse por Internet, es por eso que existirá una galería de imágenes en la que se mostrará esta clase de productos y servicios que se ofrece.

Para crear una galería de imágenes se recurrirá a un proceso análogo a la generación del catálogo para la tienda en línea, es decir, se contará con un menú de categorías cuyos valores se obtendrán de la base de datos, para los productos de la galería y la galería propiamente donde se ha de mostrar los productos de la categoría elegida.

Las imágenes de los productos se guardarán en una carpeta distinta llamada *imgGaleria*. Para reducir el tiempo de desarrollo se recurrirá a la reutilización de código ya que tomaría más tiempo diseñar y construir una galería de imágenes que sea conveniente partiendo de cero, además de salir de los límites del objetivo del presente trabajo, ya que implicaría la construcción del carrusel de imágenes, las funciones de los efectos visuales, los controles, la compatibilidad entre navegadores, la generación de *thumbnails* o vistas en miniatura de las imágenes para el carrusel; es importante destacar que la galería será capaz de generar las vistas en miniatura en el momento de ejecución y sin la necesidad de almacenar tres versiones de la misma imagen en el servidor (la original, la vista en miniatura, y la que se muestra en la página Web al momento de seleccionar su vista en miniatura), debido a estas y más funcionalidades es que se reutilizará el código gratuito de una galería basada en jQuery que se descargará libremente de Internet.

Como primera instancia y después de descargarla se hará referencia a las librerías que necesitará la galería jQuery, dicha referencia se hará dentro de la sección <header> del archivo *galeria.php* (Ver listado 4.4.2.2.1).

```
<script src="http://ajax.googleapis.com/ajax/libs/
jquery/1/jquery.min.js"></script>
<script src="galleria/src/galleria.js"></script>
```

Listado 4.4.2.2.1 Código que se agrega en el header



La principal funcionalidad se encuentra en el archivo *galleria.js* donde se encuentran las funcionalidades antes mencionadas. Posteriormente se agrega un elemento `<div>` con un identificador único que será donde se mostrará la galería de imágenes, dentro dichas etiquetas se declararán las imágenes, (mediante el proceso análogo a la Tienda en línea; se generarán de forma dinámica). Después se insertará el código `<script>` que ejecutará la galería propiamente. (Ver listado 4.4.2.2.2)

```
<h1>Galeria de <?php echo
$row_rsGaleriaProductos['nom_gal']; ?></h1>
<div class="content">
  <div id="galleria">
    <?php do { ?>
      " />
      <?php } while ($row_rsProdGal =
mysql_fetch_assoc($rsProdGal)); ?>
    </div>
  </div>
  <script>
    // Carga el tema de la galeria

Galleria.loadTheme('galleria/src/themes/classic/galleria.classic
.js');
    // Inicializa la galería
    $('#galleria').galleria();
  </script>
```

Listado 4.4.2.2.2 Código para la galería

También se agrega un título dinámico que muestre el nombre de los productos de la categoría que se muestra (ver figura 4.4.2.2.2).



Figura 4.4.2.2 Vista en el navegador del módulo “Galería”

Así mismo se presenta en el panel derecho de la página principal un formulario para que este siempre visible un formulario mientras la galería se muestra, en donde el usuario pueda obtener ayuda o asesoría adicional por parte del personal capacitado y así ponerse en contacto con la PYME.

4.4.2.3 Diseño y cotización en línea

Una de las tendencias de negocio para el mercado actual que se visualiza como el futuro de las empresas, no solo es la venta por Internet de los productos, sino además el ofrecimiento al cliente de poder personalizar sus propios productos. Donde tendrán la libertad de elegir sus propios diseños, medidas, etc.

Con referencia a la sección de requisitos que debe cubrir el sitio del presente trabajo, se requiere que si el cliente no encuentra lo que busca tenga la opción de hacer su propio diseño o redimensionamiento visual de algún producto para su posterior



elaboración, indicando algunas características del producto adicionales para ser enviadas al departamento de diseño y posterior elaboración; previo pago.

Además de esto se dispondrá de un mecanismo que ofrezca instantáneamente la cotización en línea al mismo tiempo que redimensiona el producto.

Ese costo será proporcionado por medio de un factor, que se ha de establecer por la PYME con base en las características propias del producto.

Para obtener dicho mecanismo se recurrirá a un proceso similar al anterior, es decir, se reutilizará código del Framework jQuery; ahora usando la función *resizable()* para poder redimensionar la imagen en el browser con solo arrastrar y soltar una esquina de la misma.

Al mismo tiempo se ejecuta en segundo plano otra función para conocer las dimensiones finales que el usuario está modificando y simultáneamente mostrando en pantalla tanto las medidas proporcionales como el costo proporcional al área que se obtiene, esto es posible gracias al tipo de requisitos que se solicitan, ya que para este tipo de trabajo de herrería la PYME cotiza respecto al tamaño del área, material y complejidad de diseño.

Para fines del presente trabajo, solamente se presenta dicha funcionalidad para un tipo en especial de producto (ver figura 4.4.2.1.13). El código que permitirá dicha funcionalidad se presenta en el listado 4.4.2.3.1.

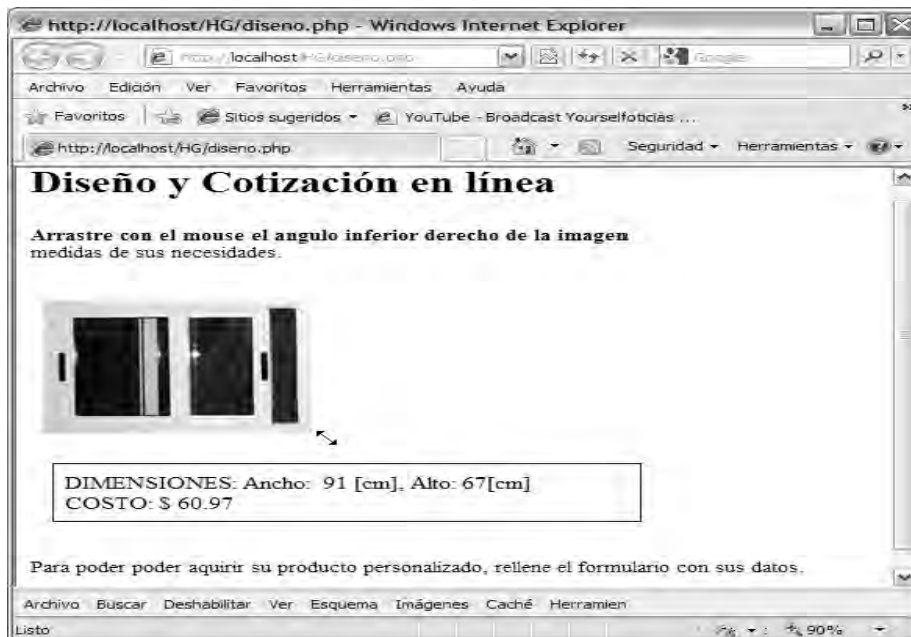


Figura 4.4.2.3.1 Vista en el navegador del módulo “Diseño y cotización”

```

if(window.ixedit){ixedit.deployed = true};
if(window.jQuery){jQuery(function(){
    (function(){ var target = jQuery('img#resizable');
    target.resizable({containment:'parent',handles:'se'});
    })();
    (function(){
    jQuery('img#resizable').bind('resize', function(event,
    ui){var target = jQuery('div#showinfo');
    target.text('DIMENSIONES: Ancho: ' +
    ($('#resizable').width())/2 + ' [cm], Alto: '+
    ($('#resizable').height())/2+'[cm]          COSTO: $
    '+(($('#resizable').height())/2
    )*(($('#resizable').width())/2)/100));});});
    });});
    
```

Listado 4.4.2.3.1 Código propuesto para la cotización en tiempo real.

Es así como a manera de versión de prueba para la PYME se ha implementado dicha funcionalidad. Posteriormente se podría implementar que el usuario pueda elegir entre un mayor número de diseños.

4.4.2.4 Información de la PYME

Para las secciones restantes que son: quienes somos, servicios y contacto, solo se mostrará información de presentación de la empresa que se mostrará en el Panel Central. Cada sección se diseñara en un archivo individual.

Se agregarán imágenes, texto y vínculos a los módulos para que el usuario pueda contactar o conocer más de la empresa. Posteriormente se aplicarán hojas de estilo CSS. (Ver figura 4.4.2.4.1).

The screenshot displays three main sections of the website:

- Acerca de nosotros:** Features a black and white photo of a person working. Text describes the company as being formed by young staff who have acquired the craft from experienced artisans, and notes that modern technology is now added to their services. It concludes by stating that all their experience in the metal trade is at the user's disposal.
- Nuestros Servicios:** Lists services offered by HERRERIA "GARCÍA":
 - Reparación de puertas, ventanas y portones.
 - Carpintería metálica.
 - Traslado.
 - Visita a domicilio.
 - Montaje industrial.
 - Construcción de todo tipo en metal.Below the list is an image of a metal cabinet and a call to action: "Conoce nuestros productos contamos con más de 200 modelos de herrería fina. Solicita un presupuesto sin compromiso. Será un placer atenderte."
- Información de Contacto:** Includes a photo of a woman on a headset. Contact details are: Teléfono: 45344346, Cel: (044) 55 344 2323 23, and e-mail: info@herreriamexicana.com. It also provides the location: "Estamos ubicados en: Av. Cemiino a Huixquilucan Estado de México". A map shows the location in Mexico. A note says: "Si desea hacernos alguna consulta o solicita un presupuesto, puede completar el formulario y a la brevedad le responderemos." The word "Gracias" is at the bottom.

Figura 4.4.2.4.1 Secciones informativas de la PYME



Vínculos AJAX

Finalmente y después de haber hecho cada modulo de las secciones del sitio Web en archivos individuales php, estos se integrarán por medio de vínculos AJAX, que será la esencia del presente trabajo.

Usando nuevamente la librería jQuery para cargar un página de forma asíncrona usando el objeto XMLHttpRequest y usando la propiedad load se escribirá el código del listado 4.4.2.4.1.

```
<script>
$(function() {
    $('a[href=HerreriaPrinc.php]').click(function() {
        $('#PanelCentro').load('HerreriaPrinc.php');
        $('#PanelIzq').load('menuPrinc.php');
        $('#PanelDer').load('anuncios.php');
        return false;
    });
});
</script>

<a href="HerreriaPrinc.php" class="but">Inicio</a>
```

Listado 4.4.2.4.1 Sección de código para los vínculos AJAX

Donde se observa que tras ejecutar el evento *click* con el Mouse sobre el vínculo se ejecutará la carga de un archivo php dentro de un elemento <div> que posee un identificador único. Este script se adaptará en cada parte de la página donde se necesite una vinculación AJAX para cargar peticiones en segundo plano sin recargar la página completa. Una característica de esta funcionalidad es que al ejecutarse las peticiones al servidor en segundo plano, el usuario no tendrá conocimiento de esta actividad a menos que se le indique de forma visual, por lo que se agregará una funcionalidad para mostrar al usuario cuando se esta llevando a cabo su petición o comunicación con el servidor y cuando esta ha terminado. En el listado 4.4.2.4.2 se muestra el código que se ejecutará cuando se pulse sobre el vínculo que llevará a cabo la funcionalidad antes mencionada.



```

$(function() {
    $('a[href=PresentacionTienda.php]').click(function() {
        $('#PanelCentro').load('PresentacionTienda.php');
        $("#cargando").css("display","inline");
        $('#PanelIzq').load("MenuTienda.php", function(){
            $("#cargando").css("display","none");});
        $('#PanelDer').load('Cesta.php');
        return false;
    });
});
<span id="cargando" style="display:none;"> Cargando...</span>
    
```

Listado 4.4.2.4.2 Código para mostrar un mensaje de carga de una página

De donde se observa que se hará uso de las propiedades CSS para mostrar el mensaje dentro de un elemento que podrá ubicarse en cualquier lugar de la página principal, y ocultarlo cuando finalice la carga del archivo. Después de lo anterior será necesario establecer las reglas de la disposición de los elementos en la página principal, es decir, la manera que se mostrarán los archivos de los distintos módulos en los paneles de la página principal. Por ejemplo, en el caso que se desee mostrar el carrito de compras, éste solo se mostrará cuando en el panel central se visualice el catálogo de los productos, o se mostrarán las promociones especiales cuando se visualice la página principal, etc. Esta administración de módulos se muestra en la tabla 4.4.2.4.1.

Cuando en el panel	Se cargue el archivo	Cargar en el panel Izquierdo	Cargar en el panel Derecho
Central	HerreriaPrinc.php	menuPrinc.php	anuncios.php
Central	PresentacionTienda.php	MenuTienda.php	Cesta.php
Central	PresentacionGaleria.php	MenuGaleria.php	FormContacto.php
Central	Diseno.php	menuPrinc.php	FormContacto.php
Central	nosotros.php	menuPrinc.php	anuncios.php
Central	contacto.php	menuPrinc.php	anuncios.php
Central	servicios.php	menuPrinc.php	anuncios.php

Tabla 4.4.2.4.1 Administración de módulos en pantalla.



Los vínculos a los principales módulos se harán desde un menú horizontal que se cargará en el encabezado de la página principal.

4.4.3 Interfaz de administración

Una modificación habitual que puede realizarse en la tienda en línea es cambiar el precio de un producto (por ejemplo para ofertas especiales o rebajas), y agregar artículos a la galería de imágenes por lo que se diseñará una interfaz de administración para los usuarios administradores del sitio Web, que no requieran de conocimientos de programación para poder dirigirlo que permitirá añadir, eliminar y modificar productos y categorías almacenados en la base de datos, que son los módulos básicos, ya que se pueden administrar gran cantidad de aspectos del sitio, que estarían fuera del alcance del presente trabajo.

La interfaz de administrador será una versión ligeramente distinta de la página que es visible para cualquier tipo de usuario, solamente que se agregarán las funcionalidades propias del administrador y serán visibles solo después de haberse registrado previamente con las credenciales autorizadas.

Inicio de sesión

Todas las cuentas de usuario se almacenan en la tabla **admin** que contendrá la información básica como el nombre de usuario y password.

Proceso del inicio de sesión:

- ✦ El administrador introduce su nombre de usuario y su contraseña.(Ver figura 4.4.3.1)
- ✦ La secuencia de comandos verifica si la combinación de nombre de usuario y contraseña exista en la base de datos.
- ✦ En caso que la combinación exista en la base de datos, podrá ir a la página principal de administración.
- ✦ Si no es así se mostrará un error.



Inicio de sesión

Nombre de usuario :

Contraseña :

Aceptar

Figura 4.4.3.1 Formulario de registro para el administrador

Para implementar la interfaz es necesario que la tabla *admin* de la base de datos no esté vacía, de lo contrario se agregarán valores para realizar las pruebas. También ha de tomarse en cuenta la manera en que se guardarán las contraseñas, por motivos de seguridad deben encriptarse con algoritmos como MD5, del cual puede configurarse con las herramientas proporcionadas por la extensión de Dreamweaver. Por tanto se agregarán los módulos para dichas tareas sobre la misma interfaz de usuario que se ha desarrollado, posteriormente se ocultarán mediante la configuración de variables de sesión para que dichos módulos puedan ser visibles exclusivamente los usuarios que se hayan autenticado con privilegios de administrador. Es así como se agregará un módulo para agregar una nueva categoría, justo de bajo del desplegado de las categorías existentes para la tienda. (Ver figura 4.4.3.2)

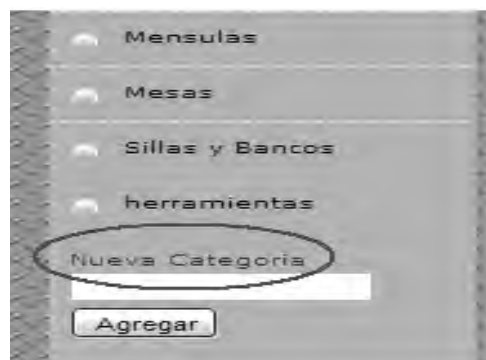


Figura 4.4.3.2 Módulo para agregar una nueva categoría.



De igual manera se agregará en el Estado de Presentación de la tienda un módulo para agregar un nuevo artículo a la tienda. (Ver figura 4.4.3.3)



Figura 4.4.3.3 Vínculo al módulo de agregar producto

Cuando se agrega un nuevo artículo se solicitan los datos del nuevo artículo como modelo, precio y descripción. También se pide que se ubique al artículo nuevo dentro de una de las categorías existentes, para evitar errores en el momento de introducir el nombre de la categoría a la que pertenece cierto artículo se ha configurado de tal manera que se muestre esa opción como un menú desplegable cuyos valores son obtenidos de la base de datos, para que el usuario solo tenga que desplegar el menú y elegir una de las categorías que se muestran.

También se ha configurado de tal manera que el administrador del sitio pueda asociar una imagen al producto cargándola desde cualquier ubicación del sistema local hasta una carpeta en el servidor, establecida para tal fin, sin tener que redimensionar la imagen para obtener las imágenes *thumbnails* o vistas en miniatura, con otros tipos de programas externos. Además se ha dispuesto que el nombre del archivo de imagen sea leído y almacenado directamente en el campo correspondiente de la base de datos para su posterior manipulación. En la figura 4.4.3.4 se muestra el formulario para el agregado de un producto nuevo. El campo de Modelo enviará el nombre del modelo del producto como información tipo texto, el campo de la categoría se mostrará como menú pero enviará la información al servidor como dato numérico, ya que el menú estará



asociado al identificador único de las categorías, el campo del precio enviará el dato como valor decimal con punto flotante y finalmente el campo Imagen se configurará para que envíe un archivo (Ver figura 4.4.3.4).

Figura 4.4.3.4 Módulo para agregar un nuevo producto

Además de lo anterior se agregará un módulo para poder modificar o actualizar el nombre de una categoría en el momento que se muestran los artículos asociados a dicha categoría y una sección para poder editar la información de los artículos mediante vínculos que mostrarán la información en el panel central. (Ver figura 4.4.3.5)

Figura 4.4.3.5 Módulos para editar el nombre de la categoría y los productos.



Cada vínculo llevará a la información del producto que anteriormente se ha registrado y que de la misma manera se mostrará en el panel central de la página principal mediante vínculos AJAX. (Ver figura 4.4.3.6).

Figura 4.4.3.6 Módulo para edición de la información del producto

Finalmente se harán las configuraciones de registro desde el panel de control de la extensión instalada en Dreamweaver donde se definirán las opciones y acciones para el sitio como son las tablas de donde se obtendrán los usuarios con privilegios. (Ver figura 4.4.3.7)

Figura 4.4.3.7 Configuración del inicio de sesión de un usuario administrador



También será necesario configurar las variables de sesión que permitirán al usuario conservar sus datos de autenticación para accesos subsecuentes a la misma aplicación Web en tanto no cierre su sesión o haya caducado. Las sesiones son específicas para los usuarios administradores. Por tanto cada usuario administrador tiene una sesión separada y tiene acceso a un conjunto separado de variables de sesión.

Posterior a la configuración se creará un nuevo archivo llamado *login.php*, que será el que contendrá la forma de registro para autenticarse como administrador. (Figura 4.4.3.8).

Nombre de Usuario : *	<input type="text"/>
Password: *	<input type="password"/>
Recordarme?:	<input type="checkbox"/>
<input type="button" value="Login"/>	

Figura 4.4.3.8 Formulario de registro para la autenticación

Así mismo se ha de establecer un vínculo donde el usuario administrador pueda cerrar su sesión, esto se hará mediante el asistente que ofrece la herramienta de extensión instalada en DW (figura 4.4.3.9).

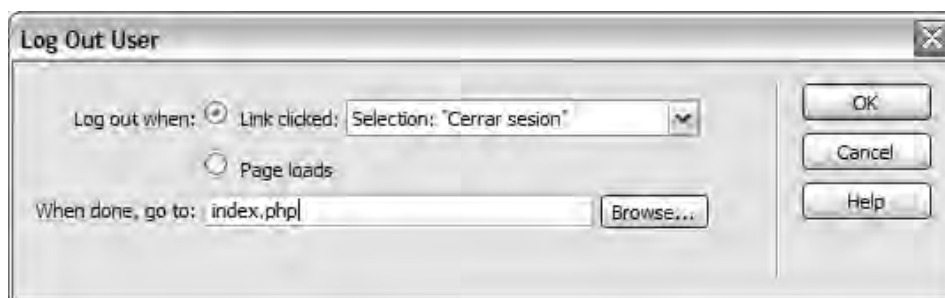


Figura 4.4.3.9 Asistente para el vínculo del cierre de sesión



Específicamente esta parte se establecerá en la página principal del sitio Web en la parte superior derecha del encabezado de la página principal para que pueda estar siempre visible al usuario administrador cuando navegue por cualquier sección del sitio. (Ver figura 4.4.3.10).

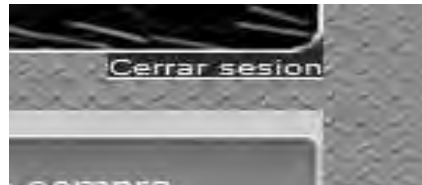


Figura 4.4.3.10 Vinculo al proceso del cierre de sesión

Finalmente y después de haber construido todos los módulos del administrador se han de determinar las regiones (módulos) que se han de mostrar al autenticarse el Administrador. Desde comportamientos del servidor en la extensión *Developer Toolbox*, en la sección de regiones condicionadas se elige la opción de mostrar si el usuario se ha registrado (ver figura 4.4.3.11). Así se evitará que un usuario no autorizado pueda entrar a los módulos de configuración del sitio. Además de eso el archivo para iniciar sesión deberá estar dentro de una carpeta o área restringida en el servidor.

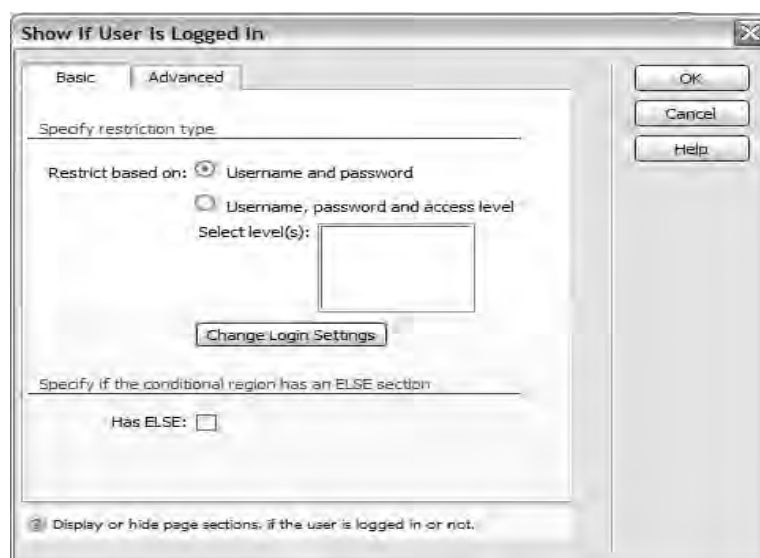


Figura 4.4.3.11 Establecimiento de las regiones visibles al administrador

Después de haber implementado todas las funcionalidades solicitadas y hacer pruebas de funcionamiento se aplicará la estética de presentación mediante las hojas de estilos a la versión final. Estas hojas de estilos CSS se guardarán en documentos aparte pero dentro de la misma carpeta raíz. Al visualizar la versión final de la página en el navegador quedará como se muestra en la figura 4.4.3.12.



Figura 4.4.3.12 Versión final del proyecto visto en el navegador



4.5 Pruebas e integración

La comprobación y la revisión del código es otro aspecto básico de la ingeniería de software que se suele pasar por alto en programación Web. Es muy habitual tratar de ejecutar el sistema con dos o tres pruebas, y decir que funciona correctamente, esto es un error muy común. Será necesario comprobar que se ha revisado y probado exhaustivamente distintos casos antes de preparar una producción final del proyecto.

Existen dos enfoques para reducir los errores del código:

- ⊕ Adoptar una práctica de revisión de código. Esto puede lograrse eficientemente mediante un equipo de programadores que revisarán el código. Gracias a esto se pueden encontrar: errores que se han pasado por alto, comprobaciones sobre las que no se ha pensado, optimización, mejoras de seguridad, componentes existentes que se pueden utilizar con el objetivo de mejorar un fragmento de código existente, funcionalidad adicional.
- ⊕ Localizar personas que prueben las aplicaciones Web y que representen a los usuarios finales del producto.

El segundo enfoque es sin duda el más importante ya que la principal diferencia entre aplicaciones Web y aplicaciones de escritorio es que cualquiera debe poder utilizar una aplicación Web. Teniendo esto en mente no se debe asumir que los usuarios estarán familiarizados con la informática. Las aplicaciones Web se deben documentar y explicar por sí mismas. Se debe pensar en cómo utilizarán los usuarios la aplicación.

Para el presente trabajo se han hecho algunas pruebas tanto de navegación como de desempeño en diferentes navegadores también se han elaborado otras métricas para verificar que el tráfico de datos sea eficiente. Para esta primera parte de la prueba se verificó que el sitio Web pueda mostrarse de la misma forma en la mayoría de los navegadores existentes.

CAPÍTULO 4. DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

Los navegadores donde fue probado el sitio fueron: FIREFOX 3.6.8, Internet Explorer 8, Opera 10.62, Google Chrome 9 y Safari 5.

✦ FIREFOX 3.6.8 (Ver Figura 4.5.1)



Figura 4.5.1 Sitio Web visto en Firefox 3.6.8

✦ Internet Explorer 8 (Ver Figura 4.5.2)

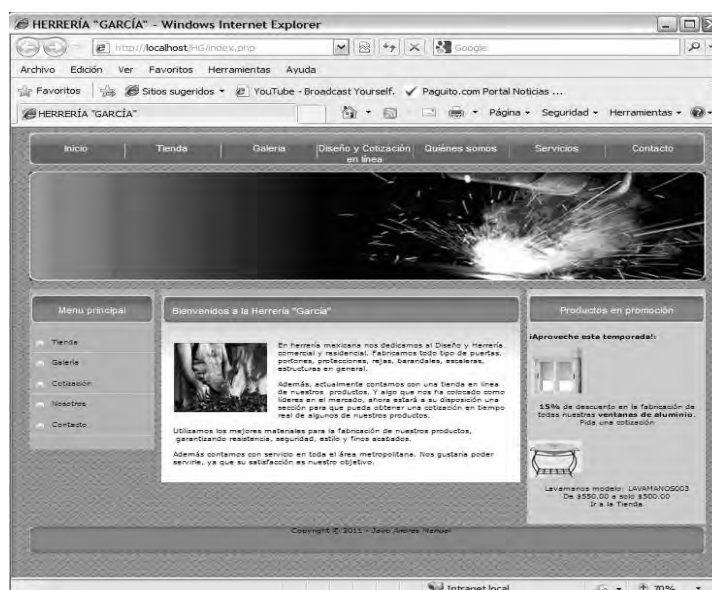


Figura 4.5.2 Sitio Web mostrado en Internet Explorer 8

➤ Opera 10.62 (Ver figura 4.5.3)



Figura 4.5.3 Sitio Web visto en Opera 10.62

➤ Google Chrome 9 (Ver figura 4.5.4)



Figura 4.5.4 Sitio Web visto en Google Chrome 9



✦ Safari 5 (Ver figura 4.5.5)



Figura 4.5.4 Sitio Web visto en Safari 5

La siguiente prueba se realizó con la ayuda de la herramienta gratuita Firebug que es un complemento para el desarrollo Web que está diseñado para el navegador FireFox.

Firebug es sin duda una de las herramientas más completas para el desarrollo Web ya que permite acceder a las propiedades DOM de los elementos y a las propiedades CSS, también permite observar las peticiones y las respuestas que se mandan en las cabeceras, los tiempos de carga de los elementos, y algunas funcionalidades más. Es por eso que se ha optado por esta herramienta para esta sección.

Como una de las intensiones del presente trabajo es la demostración de las ventajas de la programación AJAX se ha desarrollado dos versiones del sitio Web; una es copia de la otra pero en lugar de utilizar vínculos AJAX se utilizó vínculos HTML tradicionales, esto con el fin de obtener valores comparativos del tráfico de datos. Los resultados de estas mediciones se muestran en la figura 4.5.5 y 4.5.6.

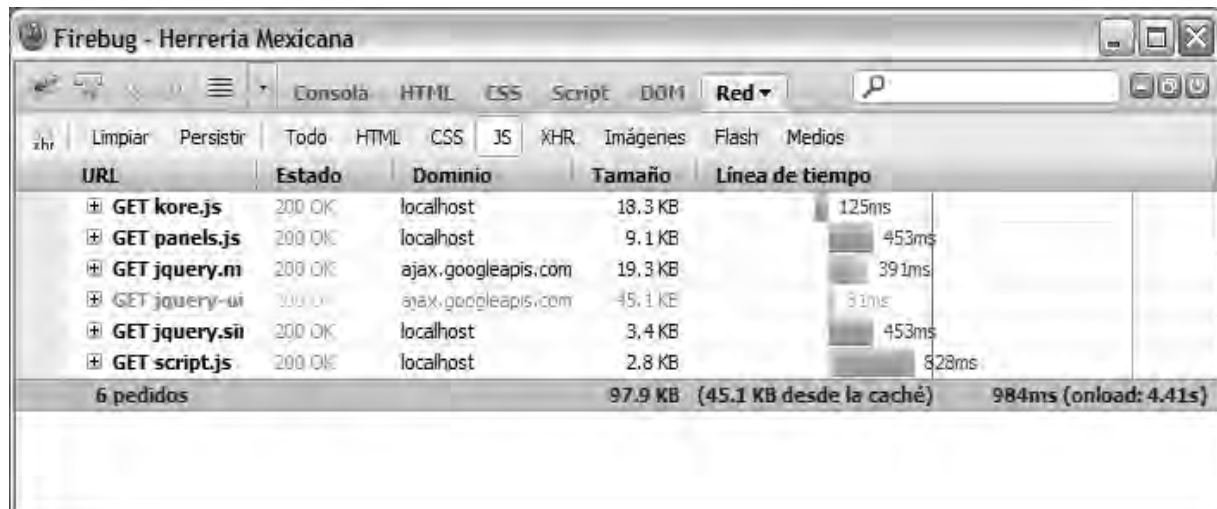


Figura 4.5.5 Tráfico de datos al momento de carga de la página sin AJAX



Figura 4.5.6 Tráfico de datos al momento de carga de la página con AJAX

Como puede observarse de las figuras anteriores una diferencia notable entre los sitios Web desarrollados con programación AJAX y los desarrollados con la programación tradicional es que la programación AJAX reduce significativamente el tráfico de datos en la red y por tanto el tiempo de carga.



Conclusiones

- ✦ Se han aplicado las bases teóricas para el desarrollo de Aplicaciones Web.
- ✦ Las ventajas de una página Web Dinámica es la interacción con el usuario recogiendo la información que realmente busca y necesita
- ✦ Cada vez se vuelve más importante la presencia de las pequeñas y medianas empresas en Internet.
- ✦ AJAX separa la parte correspondiente a la interacción con el usuario de una aplicación, de la interacción con su servidor, permitiendo que ambas procedan en forma asincrónica y en paralelo.
- ✦ AJAX realiza llamadas asíncronas al servidor para intercambiar pequeñas cantidades de datos con cada llamada.
- ✦ La programación AJAX en el desarrollo Web reduce significativamente el tráfico de red y reduce los tiempos de carga.
- ✦ El uso de AJAX promueve una mejor experiencia de usuario, ya que ofrece funcionalidades muy intuitivas.
- ✦ Aunque el uso de las tecnologías AJAX en las aplicaciones Web puede mejorar en forma considerable el rendimiento, la programación AJAX es compleja y tal vez propensa a errores. Los diseñadores de páginas requieren conocer tanto los lenguajes de secuencias de comandos como los lenguajes de marcado.
- ✦ Las bibliotecas AJAX facilitan el proceso de aprovechar los beneficios de AJAX en las aplicaciones Web, sin tener que escribir AJAX "puro".



BIBLIOGRAFÍA

- ✦ CSS, DHTML y AJAX, Xavier Colomé Fornón. ANAYA Multimedia.
- ✦ AJAX, Francisco Charte Ojeda. ANAYA Multimedia
- ✦ Desarrollo Web con PHP y MySQL, Luke Welling-Laura Thomson. ANAYA Multimedia.
- ✦ Introducción a AJAX con PHP, Lee Babin. ANAYA Multimedia.
- ✦ Programación Web con CGI 4ª. Edición, DEITHEL. PEARSON-Prentice Hall

Referencias electrónicas:

- ✦ <http://economia-de-globalizacion.blogspot.com/2006/04/ventajas-de-internet-para-las-empresas.html>
- ✦ <http://www.infomipyme.com/Docs/GT/Offline/administracion/areafuncional.htm>
- ✦ <http://www.amipci.org.mx>
- ✦ http://www.profeco.gob.mx/ecomercio/ecomercio_tendencias.asp
- ✦ <http://www.Wikipedia.org>
- ✦ <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- ✦ <http://dev.mysql.com/>
- ✦ <http://www.alegsa.com.ar/Dic/apache.php>
- ✦ <http://trends.builtwith.com/compare/PHP/ASP.NET>
- ✦ <http://www.maestrosdelweb.com>
- ✦ <http://www.milkzoft.com/blog/que-son-las-aplicaciones-ricas-de-internet/#top>
- ✦ http://www.emprende.org/component/option,com_rd_glossary/task,showcat/catid,69/Itemid,67/
- ✦ <http://stackoverflow.com/questions/636689/difference-between-frontend-backend-and-middleware-in-web-development>
- ✦ <http://www.galeon.com/zuloaga/Doc/AnalisisRequer.pdf>
- ✦ <http://dspace.ups.edu.ec/bitstream/123456789/522/6/CAPITULO4.pdf>
(comparativas Web Servers)
- ✦ <http://www.nexodigital.com.ar/metodologia.php>



-
- ✦ http://www.webandmacros.com/Diseno_web_metodologia.htm
 - ✦ <http://www.slideshare.net/Yaraher/metodologia-para-creacin-de-aplicaciones-web>.
 - ✦ <http://www.adrformacion.com/cursos/dreamcs3/leccion2/tutorial2.html>
 - ✦ <http://www.cristalab.com/tutoriales/ajax-en-jquery-c226l/>
 - ✦ <http://www.desarrolloweb.com/articulos/uso-ajax-jquery.html>
 - ✦ <http://www.lawebera.es/de0/primerospasos.php>
 - ✦ <http://www.slideshare.net/tractes/elementos-bsicos-de-una-tienda-electrnica>
 - ✦ <http://www.maestrosdelweb.com/editorial/tecnicas-ajax-mejorado-usabilidad-sitio-ejemplos-implementarlas/>