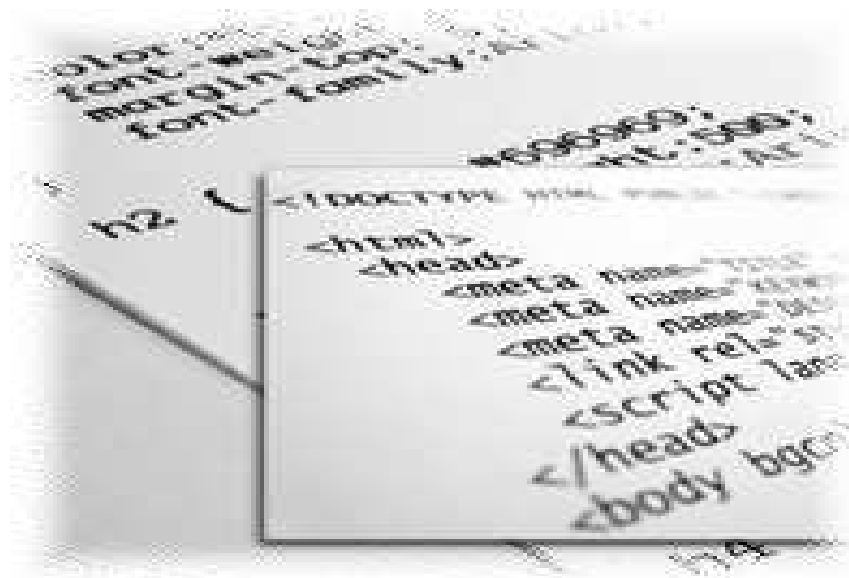




CAPITULO 4

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: Habilitación 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 del 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 producto que se mostrarán en la galeria 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.

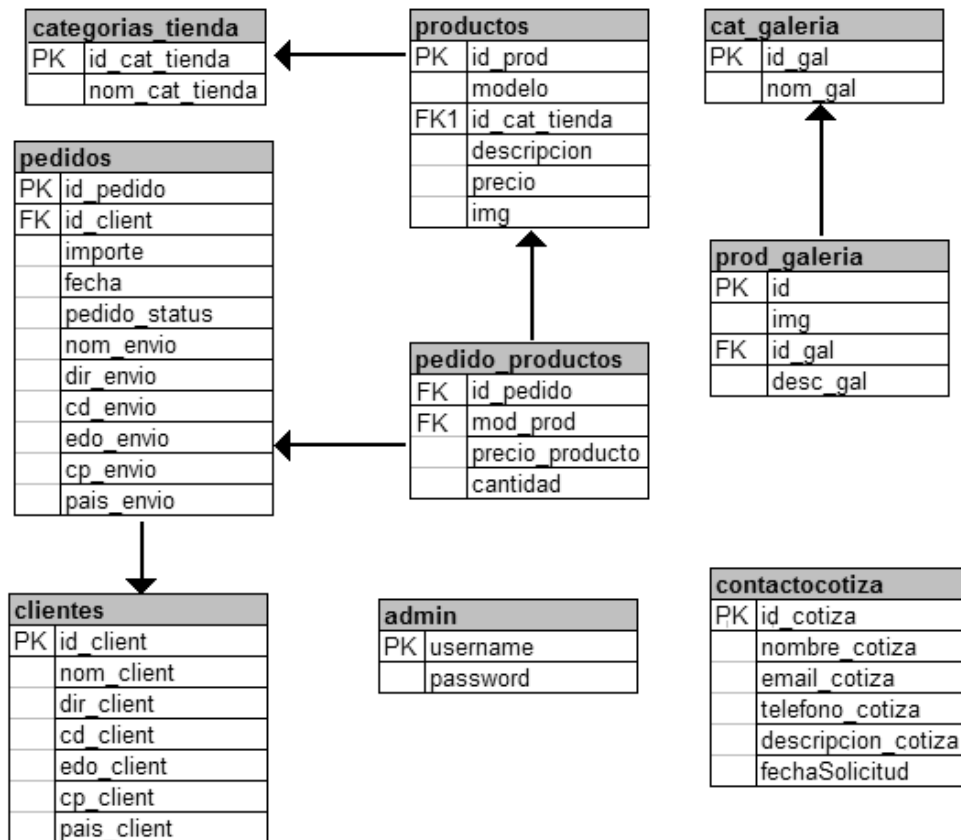


Figura. 4.2.1 Diagrama Entidad Relación de las tablas del proyecto.

- ✦ **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)

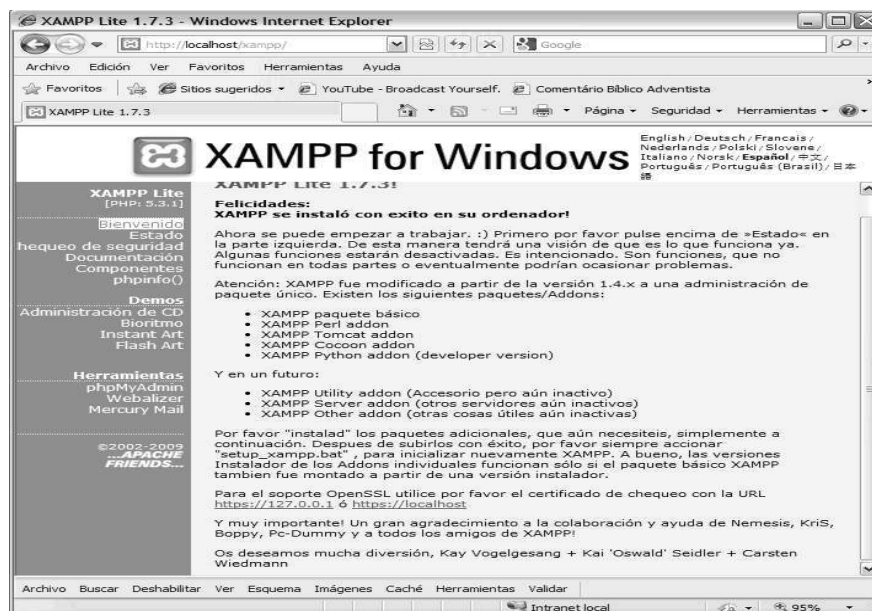


Figura 4.2.3 Consola de administración de XAMPP



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.

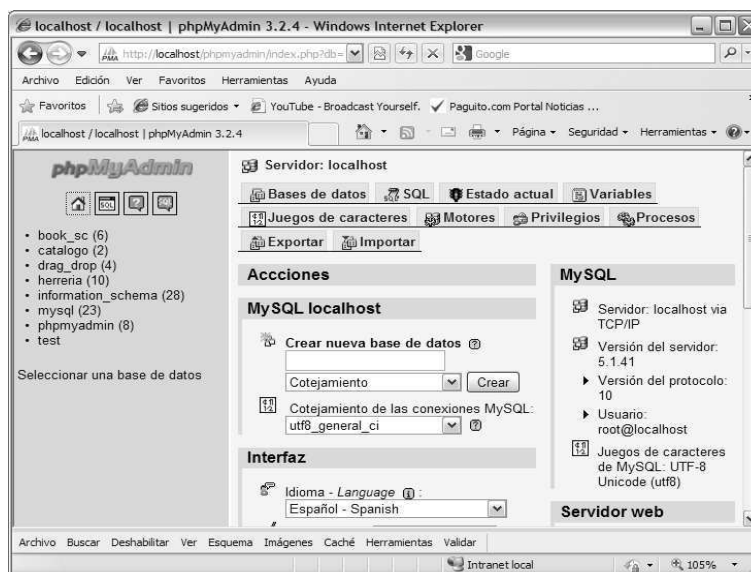


Figura 4.2.4 Página principal de phpMyadmin

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
<input type="checkbox"/>	id_cat_tienda	int(11)	NO	PRI	NULL	auto_increment
<input type="checkbox"/>	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



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 las 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: ⑦

Acción	Nombre de la clave	Tipo	Único	Empacado	Campo	Cardinalidad	Cotejamiento	Nulo	Comentario
	PRIMARY	BTREE	Sí	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).

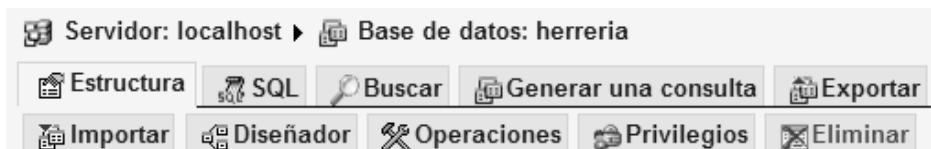


Figura 4.2.10 Menú de phpMyadmin para exportar la BD

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.

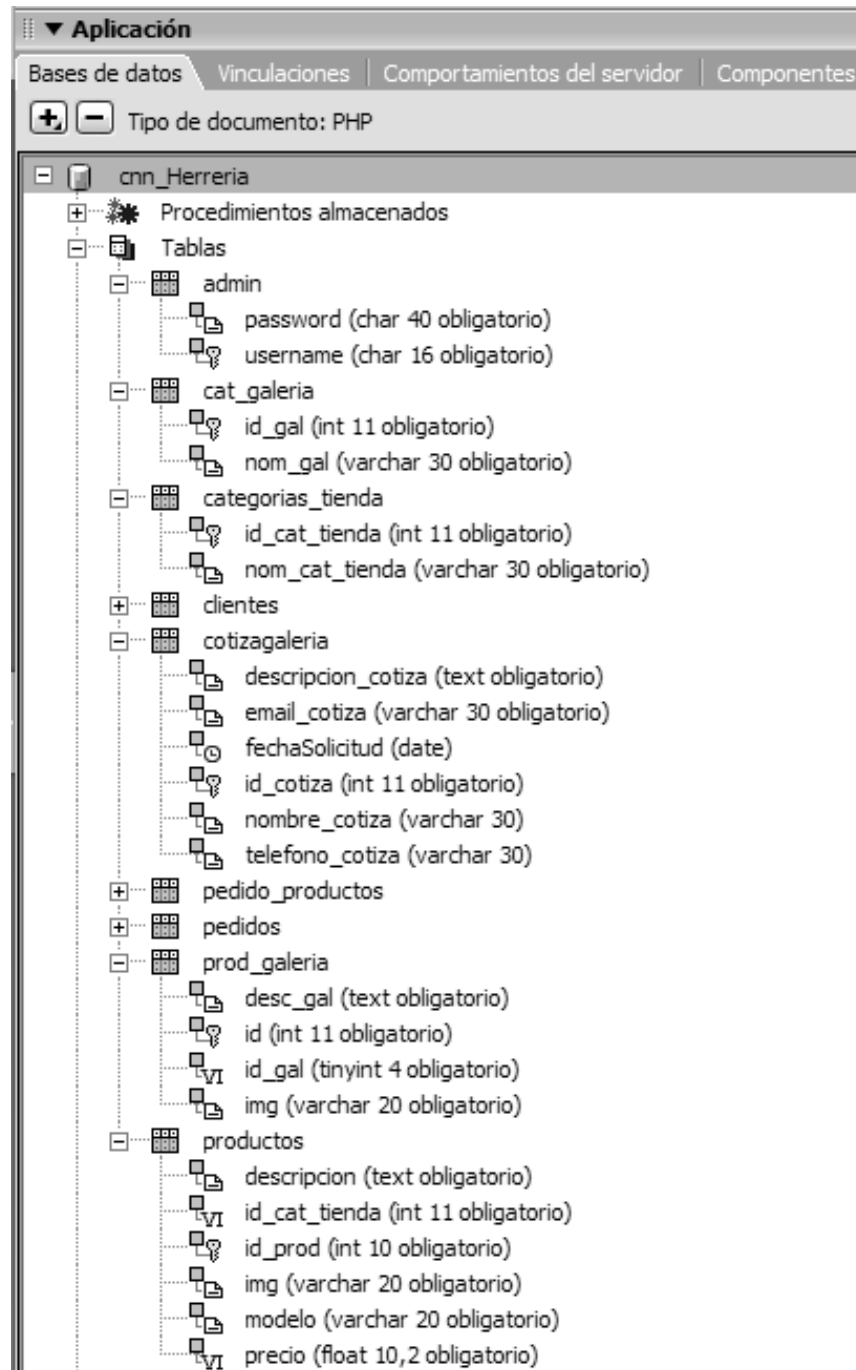


Figura 4.2.11 Árbol de tablas de la base de datos “herrería”



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).

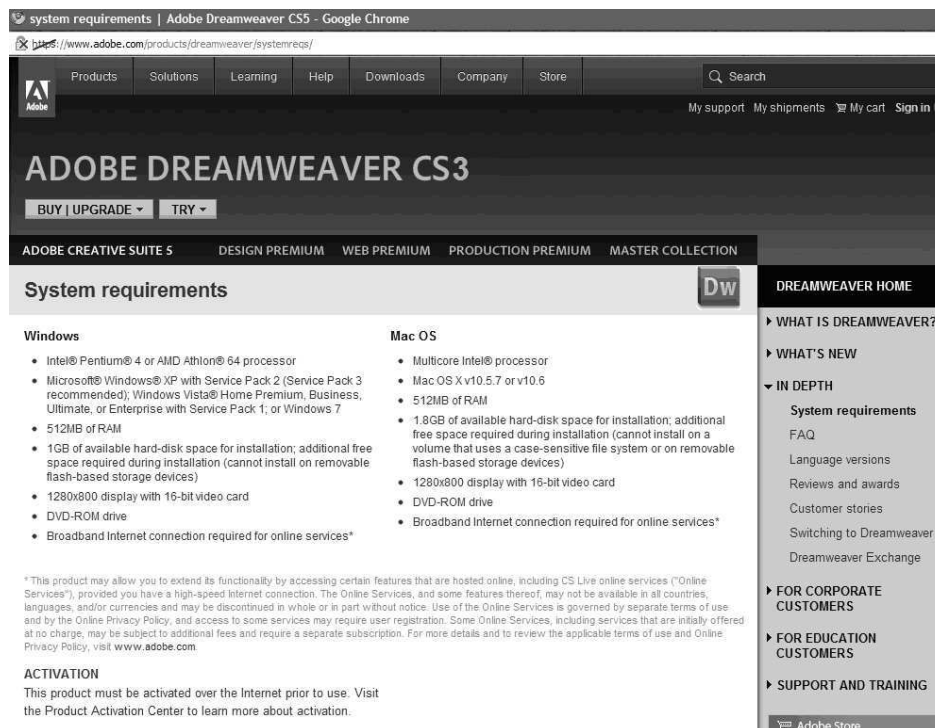


Figura 4.3.1 Página de descarga de la herramienta Dreamweaver CS3



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.

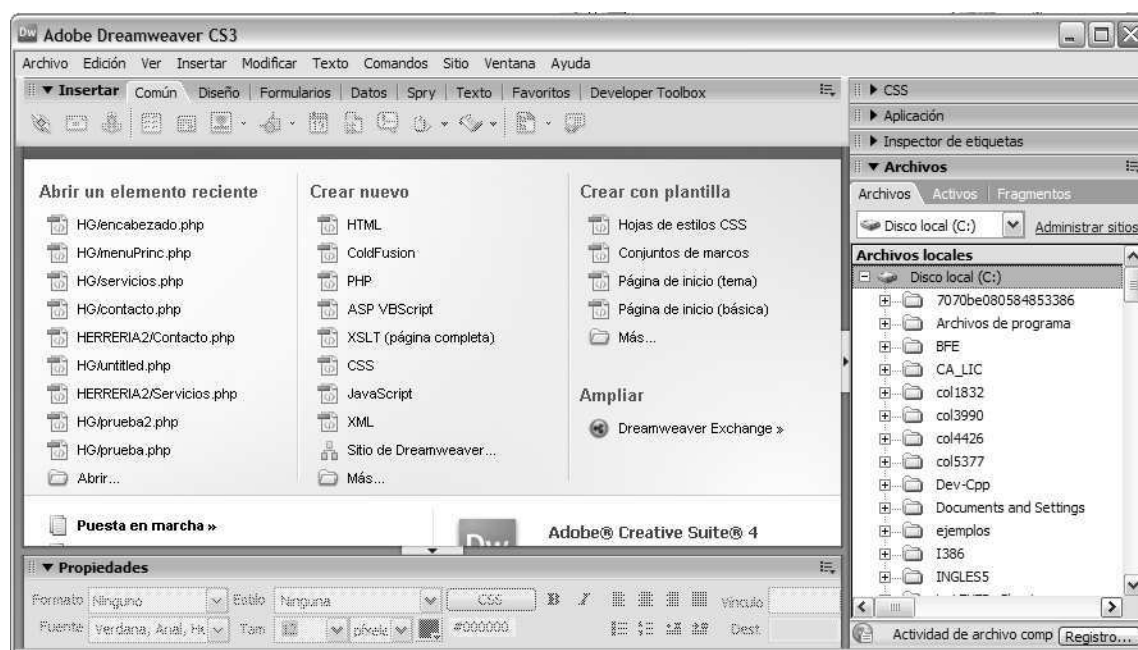


Figura 4.3.2 Vista principal de Dreamweaver CS3

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.

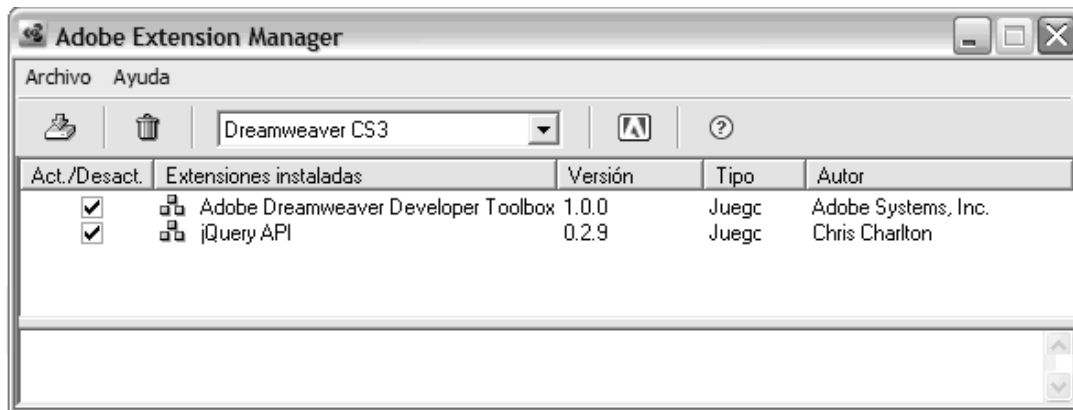


Figura 4.3.3 Administrador de extensiones de DW

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)

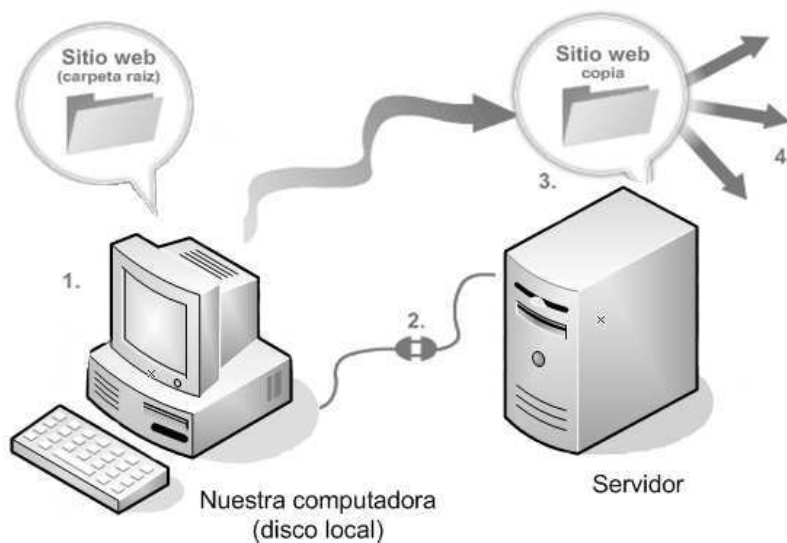


Figura 4.3.5 Método típico para la creación de sitios Web

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

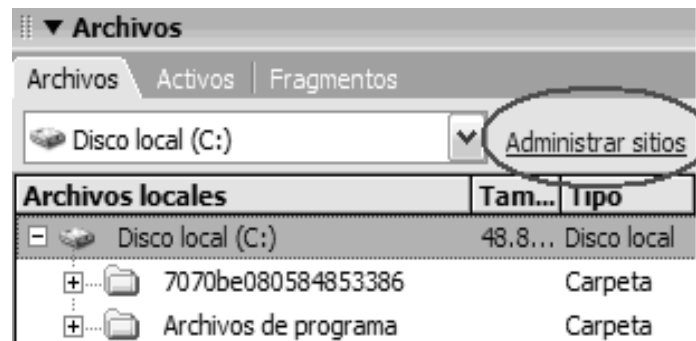


Figura 4.3.7 Ubicación del administrador de sitios

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.

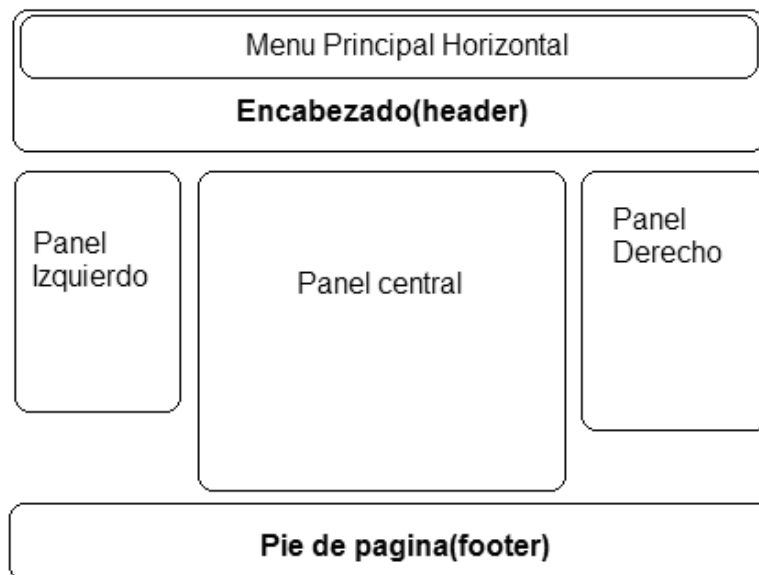


Figura 4.4.1.1 Paneles principales del sitio Web

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.

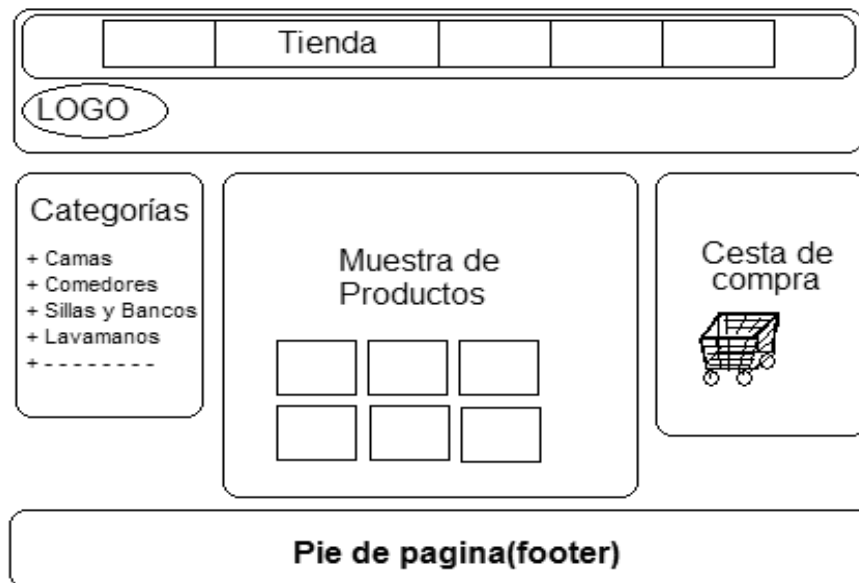


Figura 4.4.1.2 Pantalla principal para la tienda en línea



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)

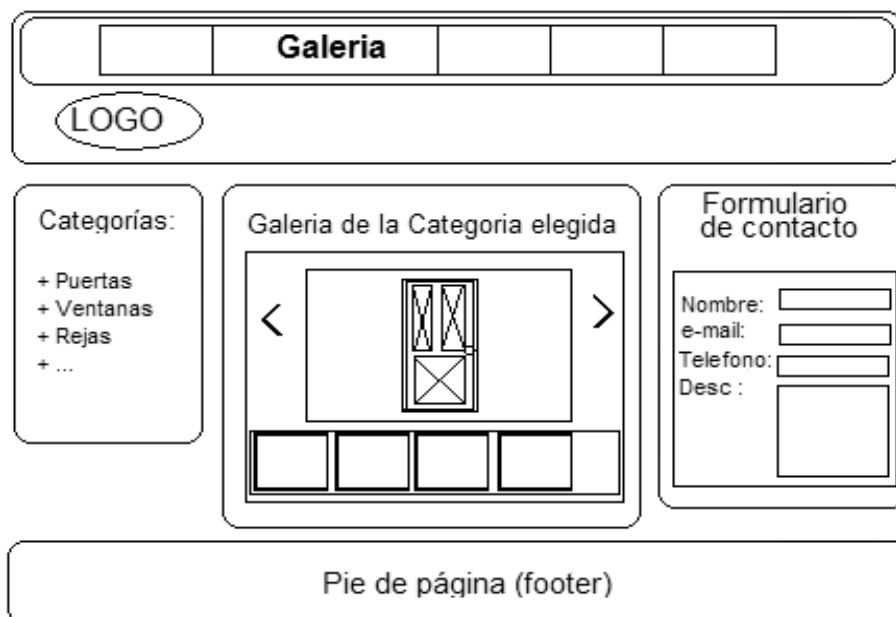


Figura 4.4.1.3 Pantalla principal para la Galería de imágenes



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).

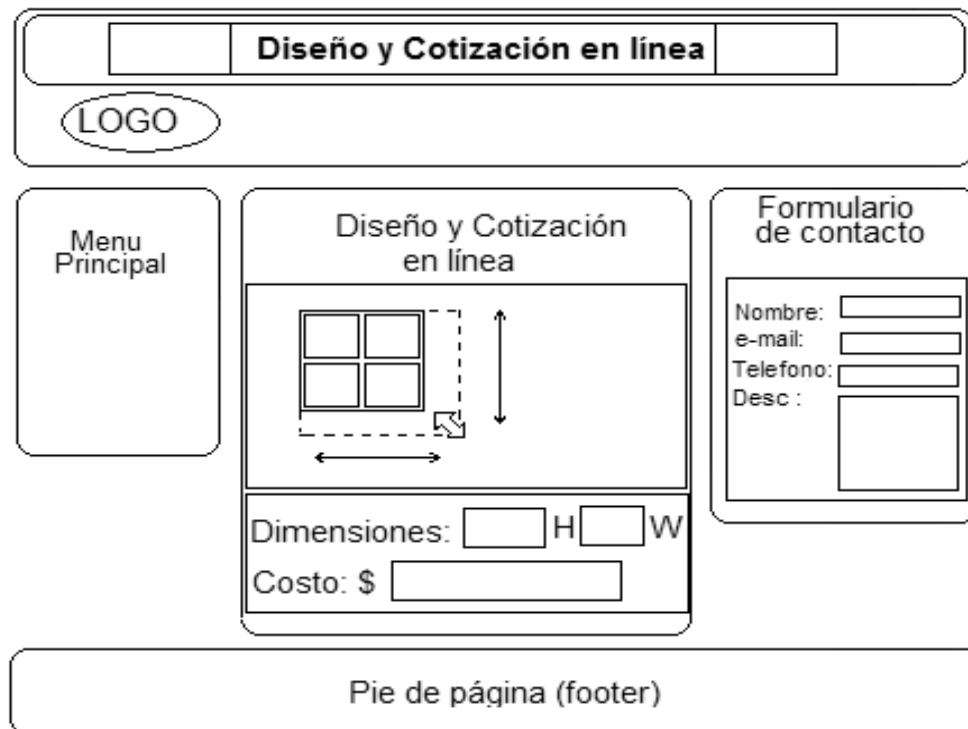


Figura 4.4.1.4 Pantalla principal para el diseño y cotización de productos en línea

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.

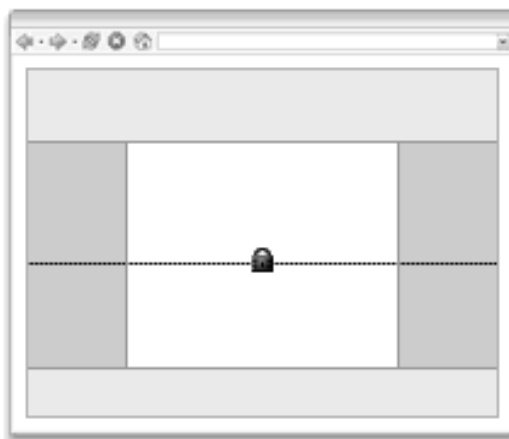


Fig. 4.4.2.1 Plantilla DW de tres columnas fijas, con encabezado y pie.

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)

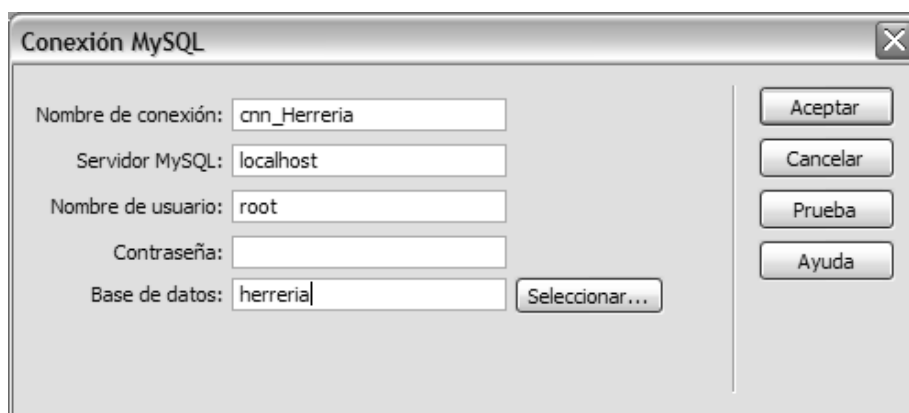


Figura 4.4.2.1.1 Establecimiento de los parámetros de conexión



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";
$database_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)

Nombre: rsCategoriasTienda

Conexión: cnn_Herreria Definir...

Tabla: categorias_tienda

Columnas: Todo Seleccionado:

id_cat_tienda
nom_cat_tienda

Filtro: Ninguno =

Parámetro URL

Ordenar: Ninguno Ascendente

Aceptar
Cancelar
Prueba
Avanzada...
QuB...
Ayuda

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).

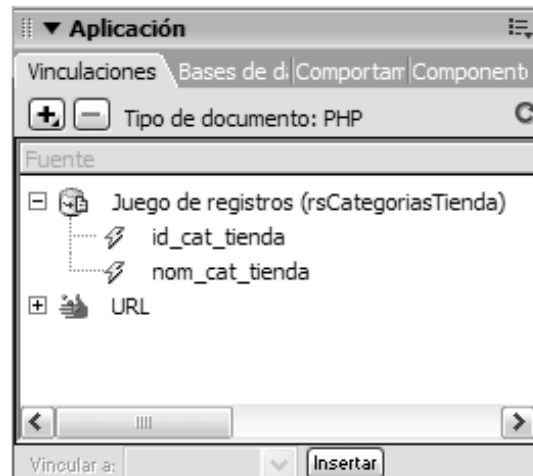


Figura 4.4.2.1.3 Árbol del juego re registros

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_rsCategori` 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).

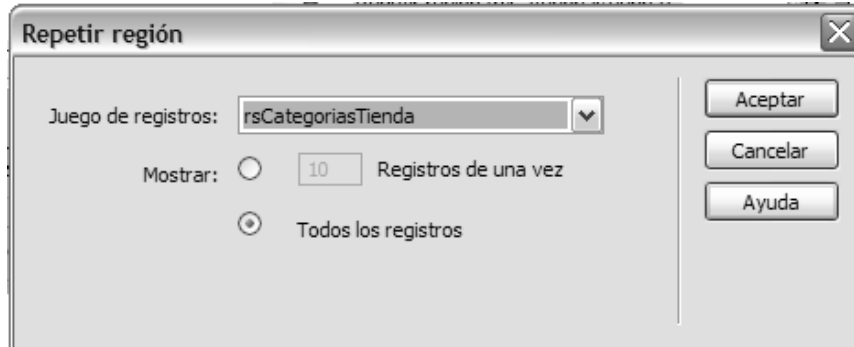


Figura 4.4.2.1.4 Región de repetición

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.

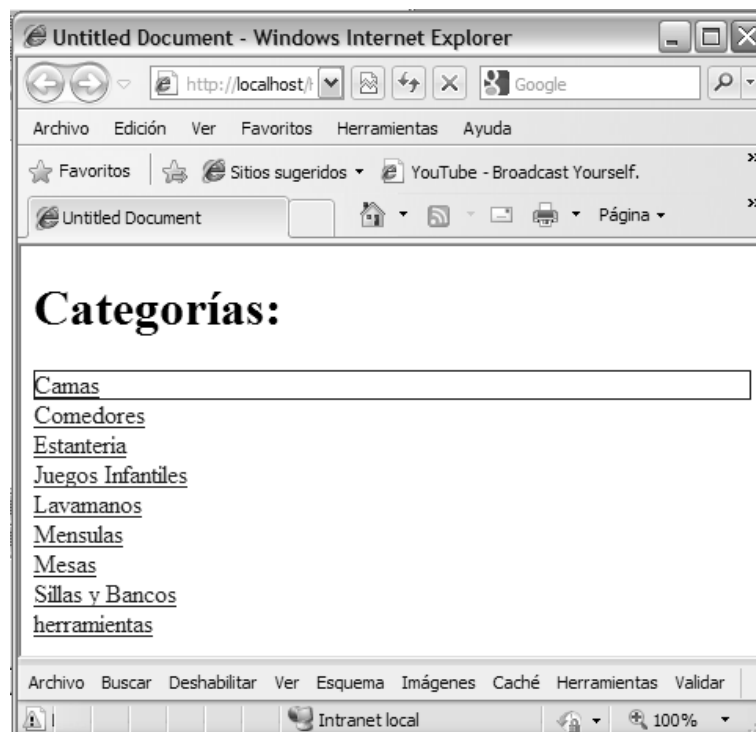


Figura 4.4.2.1.5 Vista en el navegador de la lista de categorías generada.



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.

Juego de registros

Nombre: rsCatTienda

Conexión: cnn_Herreria Definir...

Tabla: categorias_tienda

Columnas: Todo Seleccionado:

id_cat_tienda
nom_cat_tienda

Filtro: id_cat_tienda =

Parámetro URL id_cat_tienda

Ordenar: Ninguno Ascendente

Aceptar
Cancelar
Prueba
Avanzada...
QuB...
Ayuda

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).

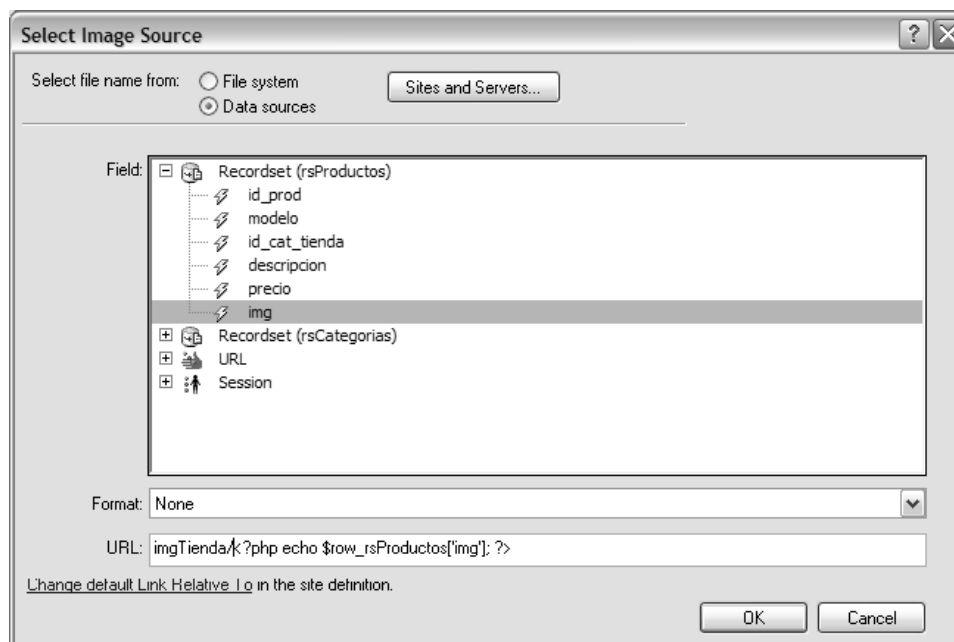


Figura 4.4.2.1.7 Selección del origen de la imagen

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 *droppable()* 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 se 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.

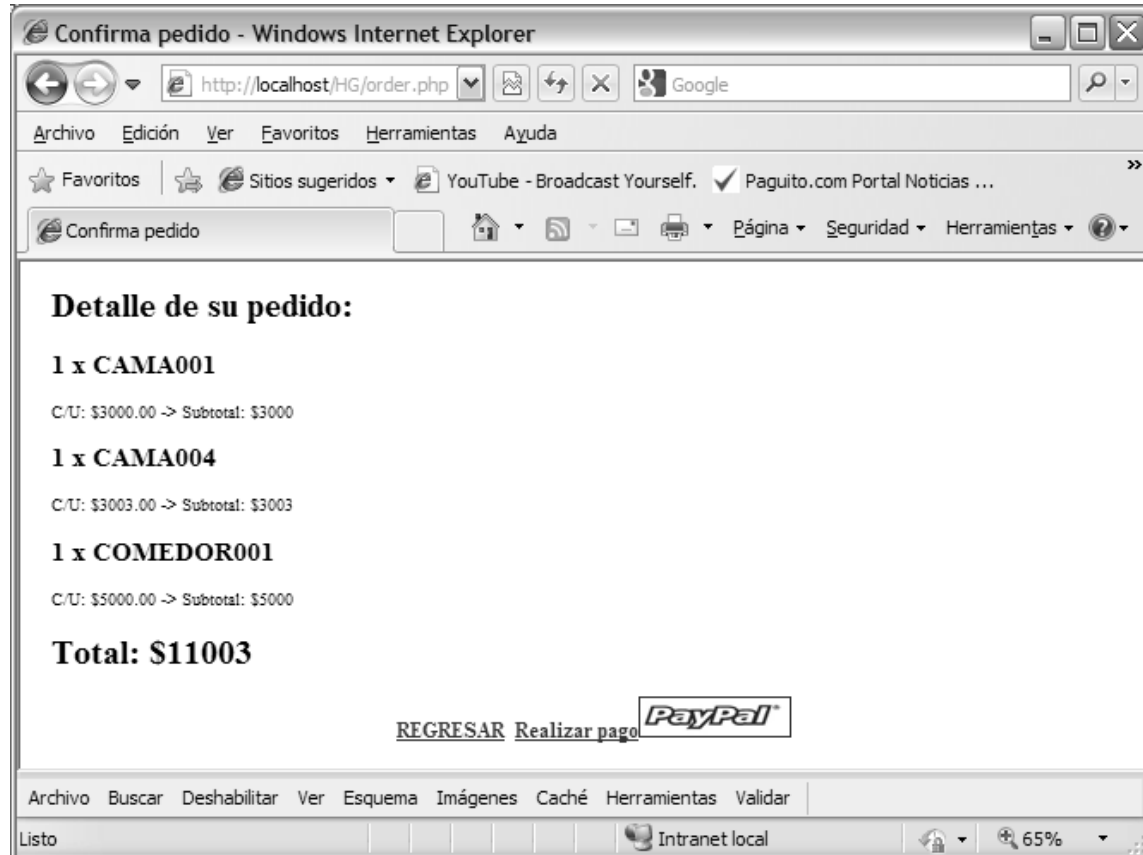


Figura 4.4.2.1.11 Vista en el navegador del resumen del pedido

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).

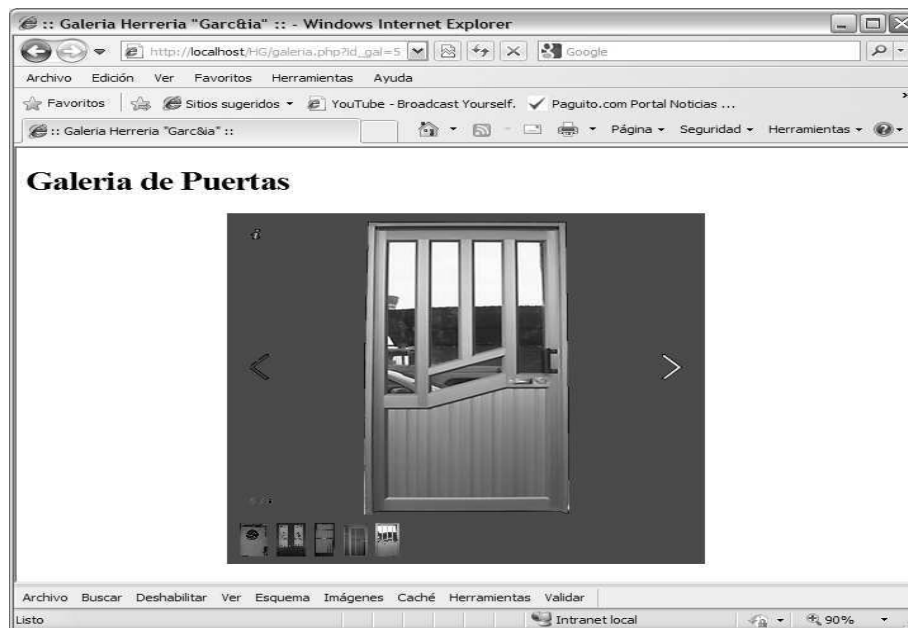


Figura 4.4.2.2 Vista en el navegador del módulo “Galería”

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.

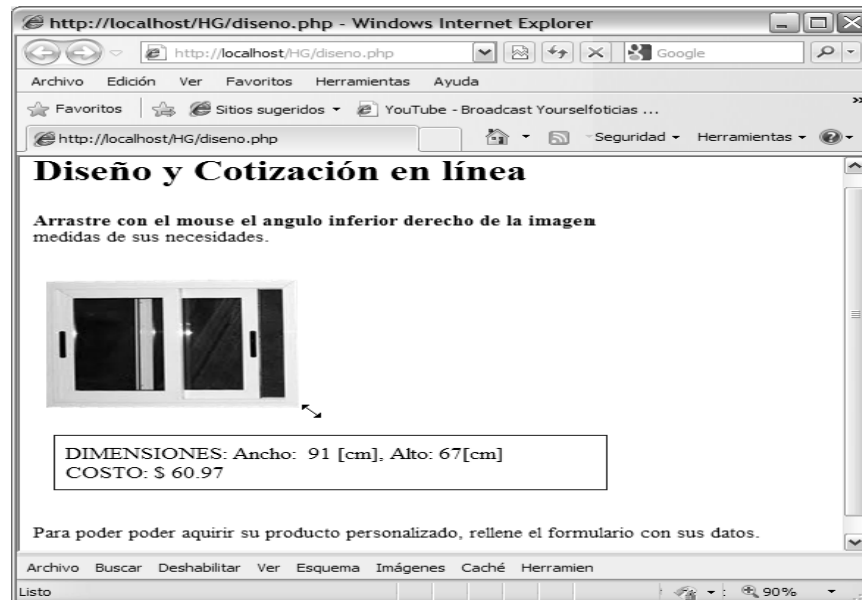


Figura 4.4.2.3.1 Vista en el navegador del módulo “Diseño y cotización”

```

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).

Acerca de nosotros



Somos una empresa formada por personal joven que ha adquirido el oficio por el legado de viejos artesanos.

Ahora, ese conocimiento se le ha añadido la tecnología de vanguardia para ponerlo a su servicio.

Ponemos a su disposición toda nuestra experiencia en el rubro de la herrería

Nuestros Servicios

En HERRERIA "GARCÍA" le ofrecemos los siguientes servicios:

- Reparación de puertas, ventanas y portones.
- Carpintería metálica.
- Traslado.
- Visita a domicilio.
- Montaje industrial.
- Construcción de todo tipo en metal.



Conoce nuestros productos contamos con más de 200 modelos de herrería fina.
Solicita un presupuesto sin compromiso. Será un placer atenderte.

Información de Contacto



Teléfono: 45344346

Cel: (044) 55 344 2323 23

e-mail: info@herreriamexicana.com

Estamos ubicados en:



Av. Camiino a Huixquilucan Estado de México

Si desea hacernos alguna consulta o solicita un presupuesto, puede completar el formulario y a la brevedad le responderemos.

Gracias

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)

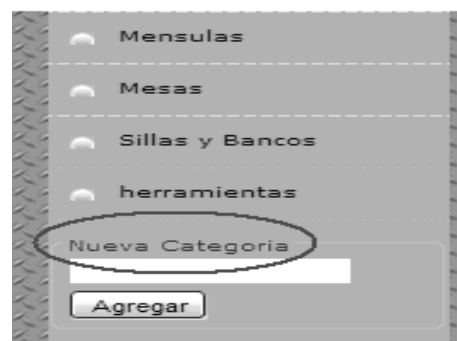


Figura 4.4.3.2 Módulo para agregar una nueva categoría.



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 cesión, esto se hará mediante el asistente que ofrece la herramienta de extensión instalada en DW (figura 4.4.3.9).

Log Out User

Log out when: Link clicked: Selection: "Cerrar sesion" Page loads

When done, go to:

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).

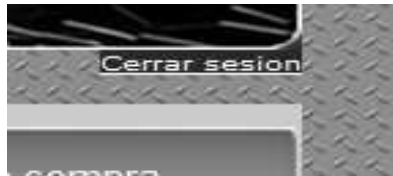


Figura 4.4.3.10 Vinculo al proceso del cierre de sesión

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.

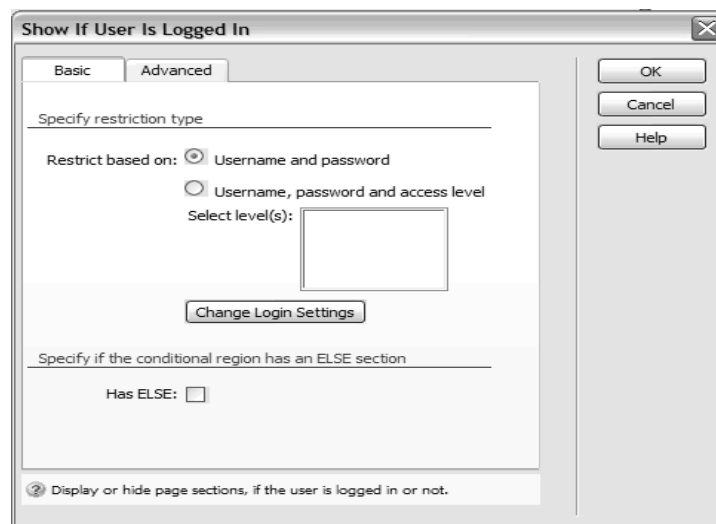


Figura 4.4.3.11 Establecimiento de las regiones visibles al administrador



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.

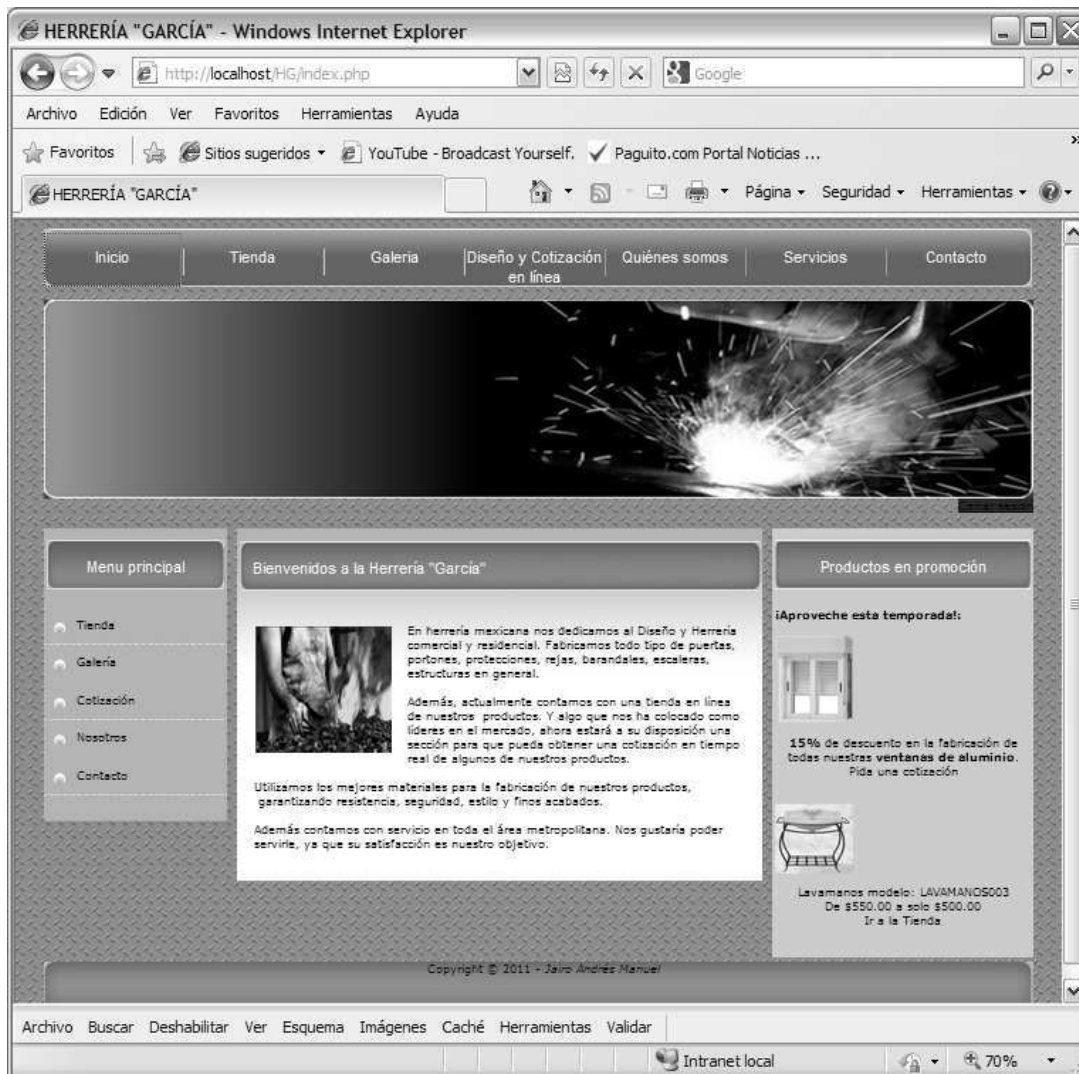


Figura 4.4.3.12 Versión final del proyecto visto en el navegador



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.



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)

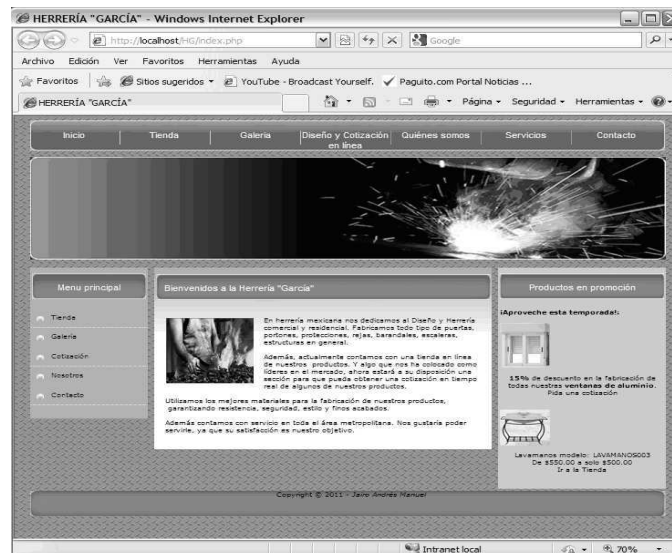


Figura 4.5.2 Sitio Web mostrado en Internet Explorer 8



✦ Opera 10.62 (Ver figura 4.5.3)

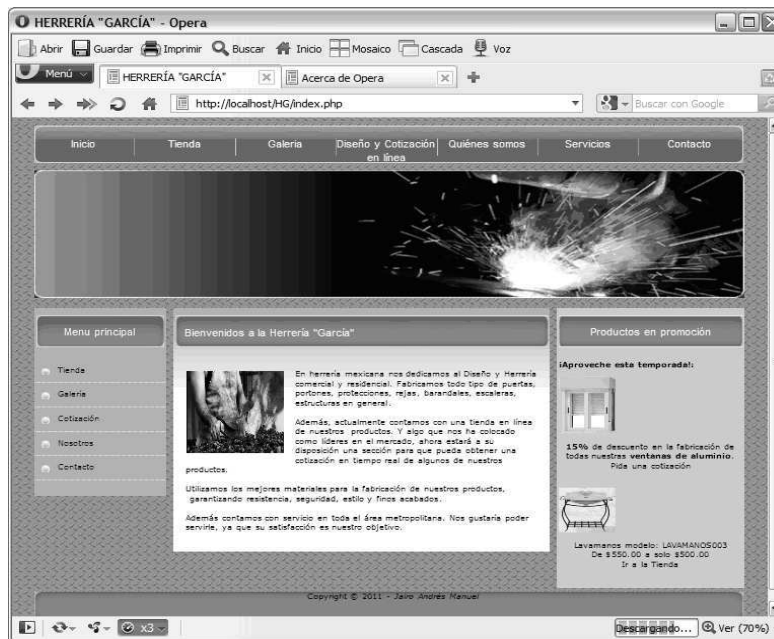


Figura 4.5.3 Sitio Web visto en Opera 10.62

✦ 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)

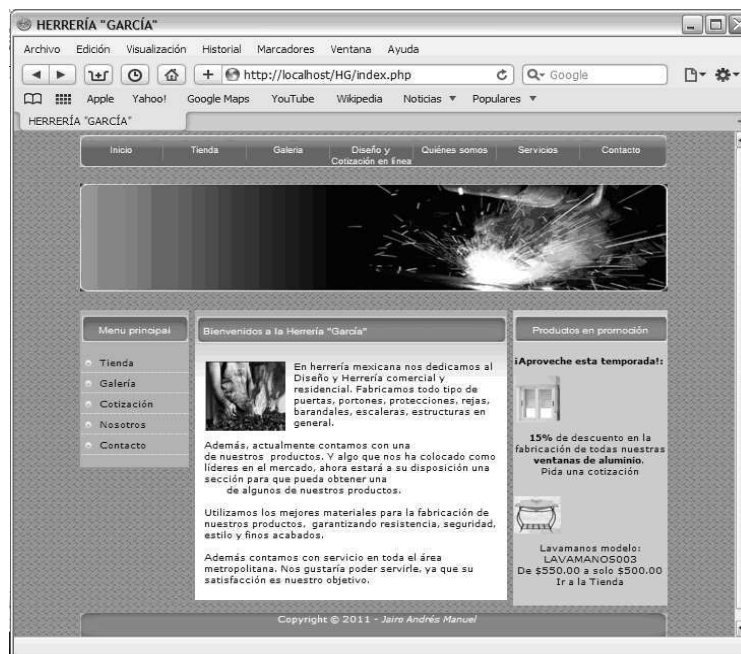


Figura 4.5.4 Sitio Web visto en Safari 5

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 de 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.



Firebug - Herreria Mexicana

Consola HTML CSS Script DOM Red

Limpiar Persistir Todo HTML CSS JS XHR Imágenes Flash Medios

URL	Estado	Dominio	Tamaño	Línea de tiempo
GET kore.js	200 OK	localhost	18.3 KB	125ms
GET panels.js	200 OK	localhost	9.1 KB	453ms
GET jquery.m	200 OK	ajax.googleapis.com	19.3 KB	391ms
GET jquery-ui	200 OK	ajax.googleapis.com	45.1 KB	81ms
GET jquery.sii	200 OK	localhost	3.4 KB	453ms
GET script.js	200 OK	localhost	2.8 KB	828ms
6 pedidos			97.9 KB (45.1 KB desde la caché)	984ms (onload: 4.41s)

Figura 4.5.5 Tráfico de datos al momento de carga de la página sin AJAX

Firebug - HERRERÍA "GARCÍA"

Consola HTML CSS Script DOM Red

Limpiar Persistir Todo HTML CSS JS XHR Imágenes Flash Medios

URL	Estado	Dominio	Tamaño	Línea de tiempo
GET jquery.m	200 OK	localhost	76.9 KB	270ms
GET jquery-ui	200 OK	ajax.googleapis.com	45.1 KB	26ms
2 pedidos			122 KB (122 KB desde la caché)	415ms (onload: 819ms)

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.