UNIVERSIDAD DEL AZUAY

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

Monografía Previa La Obtención
Del Título De Ingeniero De Sistemas

Tema: Construcción de un Sitio Web para la cotización de artículos de ferretería, utilizando Linux Fedora, MySQL, Php y Cgi.

Autores:

Leonardo Agila A.
Ralph Ayala V.

Directores:

Ing. Fernando Balarezo.
Ing. Pablo Esquivel

Cuenca, Febrero 22 de 2005

DEDICATORIA

El presente trabajo le dedico a mi hijo, a mi esposa, a mis padres y hermanos.

Leonardo

DEDICATORIA

Dedico esta monografía a mis padres, a mi hermana, quienes con su esfuerzo y amor han logrado ser de gran apoyo para mi persona permitiéndome culminar con éxito mis estudios.

Ralph

AGRADECIMIENTO

Agradezco a mis padres por el sacrificio de darme la oportunidad de estudiar fuera del país.

Leonardo

AGRADECIMIENTO

Agradezco a Dios, por permitirme culminar mis estudios, a mis padres por su arduo esfuerzo y a quienes me supieron apoyaron a lo largo de mis estudios.

Ralph

RESPONSABILIDAD

Las ideas y opiniones vertidas en esta monografía son de exclusiva responsabilidad de los autores.

Leonardo Agila A.

Ralph Ayala V

JUSTIFICACIÓN

Debido al alto costo que se tiene tanto en la implementación de Servidores Web como en la construcción de Sitios Web para las empresas, hemos visto la necesidad de buscar una solución que disminuya los costos. Permitiéndonos así la implementación de un Servidor Web Linux Fedora y la construcción de un sitio Web utilizando MySQL, Php, dando lugar a que las empresas tengan un ahorro considerable y una gran funcionalidad.

OBJETIVOS

1.1. Abaratar costos de implementación de un Servidor Web y la construcción de un Sitio Web.

1.2. Construir un Sitio Web seguro, confiable, de acceso rápido, y flexible al desarrollo tecnológico.

INDICE

CAPITULO 1. LINUX 1

1.1. Introducción 1
1.2. Instalación 3
1.2.1. Chequeo de la integridad del medio. 4
1.2.2. Configuración del Idioma, Teclado, Mouse, Monitor 4
1.2.3. Configuración del tipo de Instalación. Particiones. 5
1.2.4. Particiones. 5
1.2.5. Configuración Boot - Loador. 6
1.2.6. Configuración de la Red. 6
1.2.7. Firewall 8
1.2.8. Ingreso de la clave de Súper Usuario. 8
1.2.9. Inicio de instalación de los rpm`s. 9
1.3. Configuración de red. 9
1.4. DNS (Domain Name Service) 9
1.4.1. Definición 9
1.4.2. Configuración del servidor DNS 10
1.4.2.1.Configurar el BIND 10
1.4.2.2. Crear una Zona 11
1.4.2.3. Habilitar la Zona 13
1.4.2.4. Probar el servidor DNS 14
1.4.2.5. Iniciar el Servidor DNS al iniciar la máquina 14
1.5. Apache 15
1.5.1. Instalación 15
1.6. Enlaces Simbólicos 17
1.7. Permisos CHMOD 17
1.8. Virtual Host 19
1.8.1. Introducción 19
1.8.2. Configuración 20

CAPITULO 2. HTML 21

2.1. Introducción 21
2.2. Estructura 21
2.3. Conceptos básicos 22
2.3.1. Formatos 22
2.3.2. Formato de letra. 23
2.3.3. Imágenes 23
2.3.4. Enlaces 24
2.3.5. Tablas 24
2.3.6. Formas 25

CAPITULO 3. PHP 28

3.1. Introducción 28
3.2. Instalación de PHP en Linux con Apache 29
3.3. Conceptos Básicos. 31
3.4. Variables y Constantes 33
3.4.1. Variables 33
3.4.2. Constantes 34
3.5. Expresiones y operadores 35
3.6. Sentencias 36
3.6.1. IF 36
3.6.2. Switch 36
3.6.3. While. 37
3.6.4. FOR 37
3.7. Conexiones a bases de datos 37
3.8. Funciones de PHP 38
3.9. Archivos y Directorios 41
3.9.1 Archivos 41
3.9.2 Directorios 44

3.10 Creación y Tratamiento de Imágenes 44

CAPITULO 4. MySQL 47

4.1. Introducción 47
4.2. Instalación y configuración 48
4.3. Creación de Base de Datos y Tablas 49
4.4. Insertar, modificar y eliminar registros de tablas 53
4.5. Tipos de datos 54
4.6. Consultas 56
4.7. Funciones PHP de acceso a MySQL 56
4.8. Conectar a MySQL desde PHP 61
4.9. Mostrar los datos de una consulta 62
4.10. Seguridad en MySql 63

CAPITULO 5. CGI 67

5.1 Descripción 67
5.2 Riesgos 69
5.2.1 Peligros 69
5.2.2 Vulnerabilidad 69
5.3 SSI (Server-side incluyes) 70
5.3.1 ¿Que son? 70
5.3.2 Peligros 71
5.3.3 Soluciones 71
5.3.4 ¿Que debemos Hacer? 72
5.3.5 Lenguaje a utilizar 73
5.3.6 Cómo enviar los datos 73

CAPITULO 6. DESARROLLO DEL SITIO WEB 75

5.1. Modelo Conceptual 75
5.2. Modelo Físico 76
5.3. Diccionario de Datos. 77
5.4. Creando la Bases de Datos en MySQL. 80
5.5. Crear la página Web utilizando HTML y PHP 81
6.5.1. Implementación de la aplicación modo cliente 81
6.5.2. Implementación de la aplicación modo administrador 85

CONTENIDO

CAPITULO 1. LINUX

1.9. Introducción

Linux es una de las tantas variantes de Unix. Se trata de un sistema operativo de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la universidad finlandesa de Helsinki. A medida que avanzaba en su desarrollo, Linux fue dejando el código fuente de las sucesivas versiones del kernel y utilidades de Linux a disponibilidad de los usuarios de Internet. Este fue sin duda un gran acierto, ya que hizo posible que una multitud de desarrolladores de todo el mundo se familiarizaran con el código, lo cual en primera instancia significó un gran aporte de sugerencias, evolucionado luego hacia un espectacular ejemplo de desarrollo distribuido de software, muchos desarrolladores independientes, desde todo el planeta tomaron a su cargo la producción de software para Linux, ya sea escribiéndolo desde cero o portándolo desde otras plataformas Unix. Esta modalidad de desarrollo continúa aún hoy y ha permitido a Linux alcanzar un alto nivel de desarrollo y madurez, así también como un amplio grado de aceptación.

Actualmente, Linux posee todas las características que pueden encontrar en cualquier sistema Unix moderno.

Las utilidades que nos ofrece son programas especializados, tales como editores, compiladores y programas de comunicaciones, que realizan operaciones de computación estándar. Incluso se puede crear sus propias utilidades.

Linux contiene un gran número de utilidades. Algunas efectúan operaciones sencillas, otras son programas complejos con sus propios

juegos de órdenes. Para empezar, muchas utilidades de pueden clasificar en tres amplias categorías: editores, filtros y programas de comunicaciones. También hay utilidades que efectúan operaciones con archivos y administración de programas.

En líneas generales podemos decir que se dispone de varios tipos de sistema de archivos para poder acceder a archivos en otras plataformas. Incluye un entorno gráfico X Windows (Interfaz gráfico estándar para máquinas UNIX), que nada tiene que envidiar a los modernos y caros entornos comerciales. Está orientado al trabajo en red, con todo tipo de facilidades como correo electrónico por ejemplo. Posee cada vez más software de libre distribución, que desarrollan miles de personas a lo largo y ancho del planeta. Linux es ya el sistema operativo preferido por la mayoría de los informáticos.

Un ejemplo de la popularidad que ha alcanzado es sistema y la confianza que se puede depositar en él es que incluso la NASA ha encomendado misiones espaciales de control de experimentos a la seguridad y eficacia de Linux.

Por lo tanto, la gran popularidad de Linux incluye los siguientes puntos:

Multitarea:

La multitarea no consiste en hacer que el procesador realice más de un trabajo al mismo tiempo, lo único que realiza es presentar las tareas de forma intercalada para que se ejecuten varias simultáneamente. Por lo tanto en Linux es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.

Multiusuario:

Linux permite a varios usuarios acceder al mismo tiempo a través de terminales, y que distribuya los recursos disponibles entre todos. Así mismo, el sistema permite la posibilidad de que más de un usuario pudiera trabajar con la misma versión de un mismo programa al mismo tiempo, y actualizar inmediatamente cualquier cambio que se produjese en la base de datos, quedando reflejado para todos.

Multiplataforma:

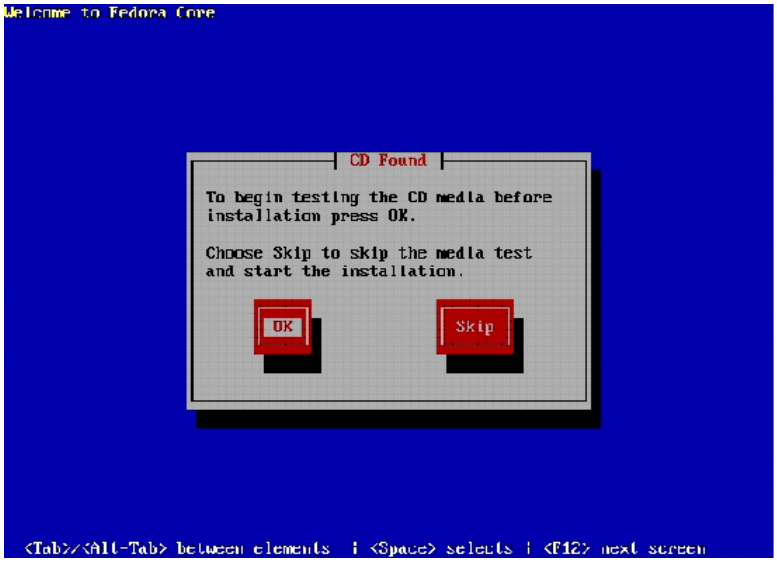
Es decir que puede correr en muchas CPU distintas (Intel, AMD, Motorola, Sun, Sparc, etc.)

1.10. Instalación

Para la instalación seguiremos la siguiente secuencia con el objetivo de presentar todos los pasos realizados en la presente monografía.

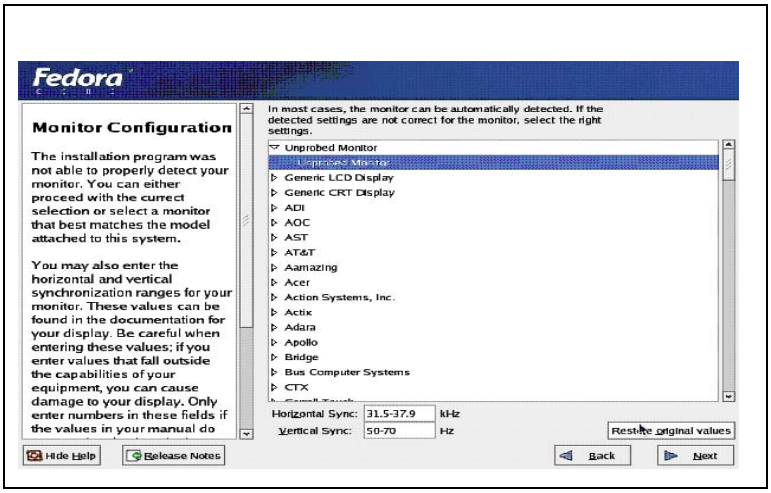
1.10.1. Chequeo de la integridad del medio.

Es necesario realizar un chequeo a los Cd's de instalación para evitarnos contrariedades en el transcurso de la instalación, para lo cual Fedora posee una herramienta que nos facilita dicho chequeo.



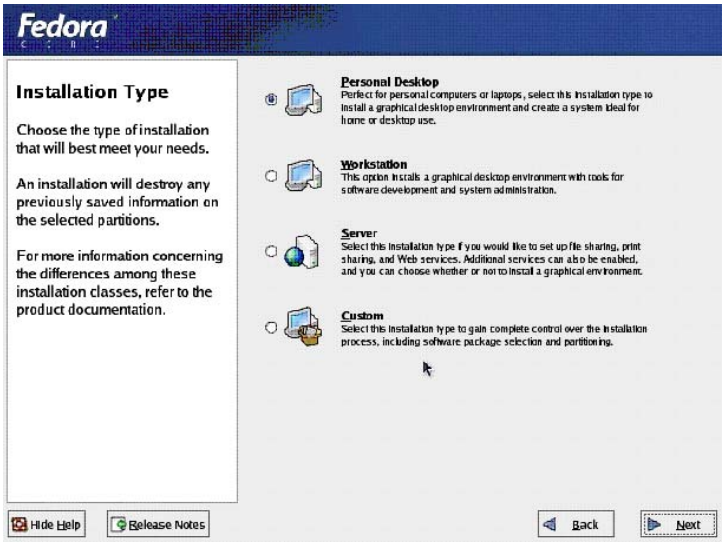
1.10.2. Configuración del Idioma, Teclado, Mouse, Monitor

Linux nos ofrece la facilidad de configurar el idioma, por lo general es elegido por los usuarios pero les recomendamos que al configurar el teclado, mouse, monitor, se opte por aceptar lo que Linux Fedora elige por default.



1.10.3. Configuración del tipo de Instalación.

Para nuestro propósito seleccionaremos la opción personalizada, con el fin de instalar los paquetes requeridos.

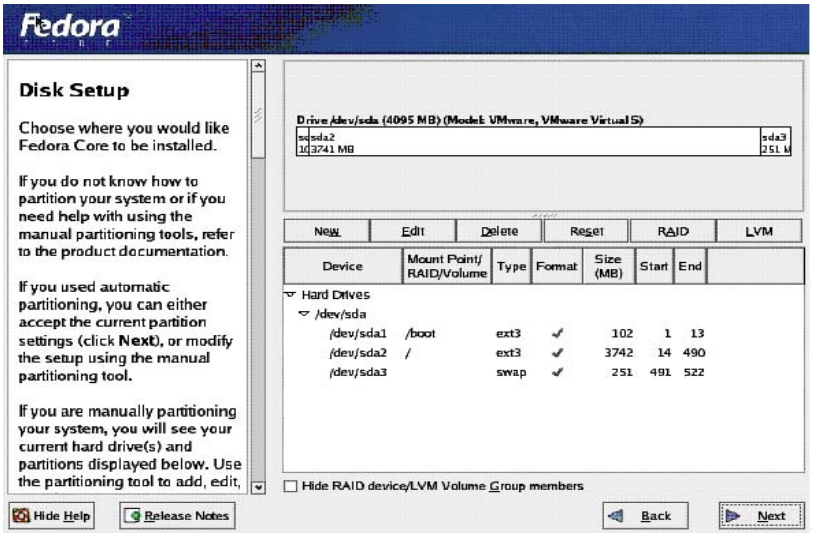


1.10.4. Particiones.

En las particiones se nos da a elegir una automática y manual, hemos procedido a realizarla manualmente para obtener el mejor rendimiento al servidor.

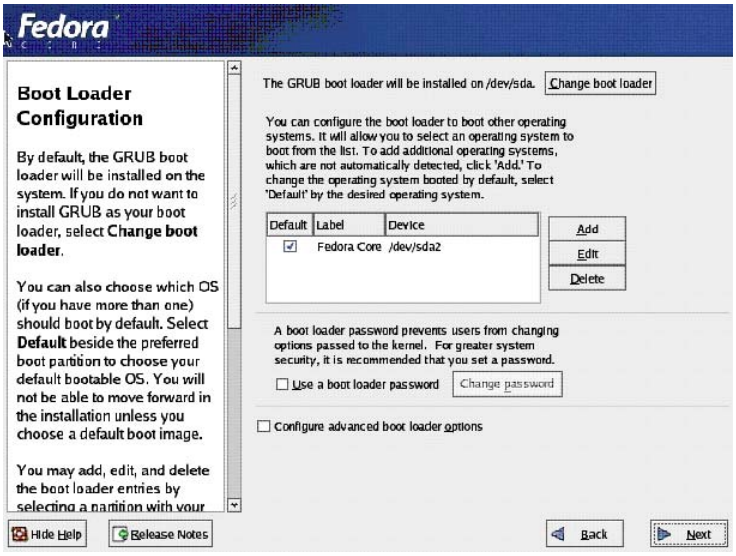
- Se selecciona la partición en blanco para crear el directorio raíz (/), se ingresa el tamaño por ejemplo si el espacio en disco es de 10 Gb entonces se digita 8000.

- Luego se crea la partición SWAP, que es el doble de memoria RAM por ejemplo, si se tiene 256 poner 512.



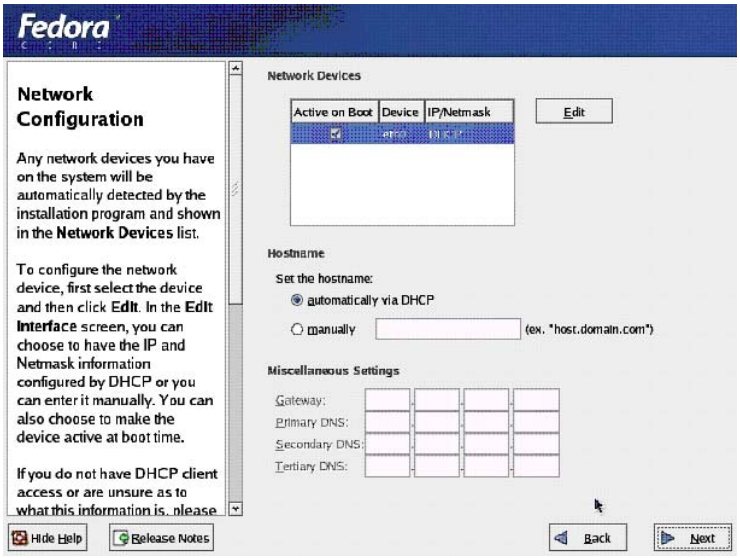
1.10.5. Configuración Boot - Loador.

Se recomienda elegir el gestor de arranque GRUB para que Linux se pueda utilizar en modo grafico.

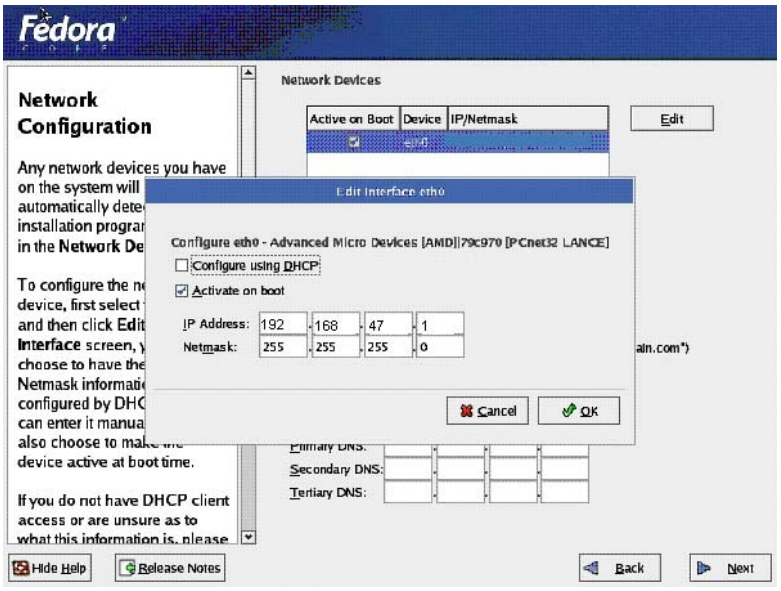


1.10.6. Configuración de la Red.

Linux nos permite realizarla al momento de la instalación y una vez instalado, nosotros tomaremos la primera opción.



Pulsamos Editar para continuar.



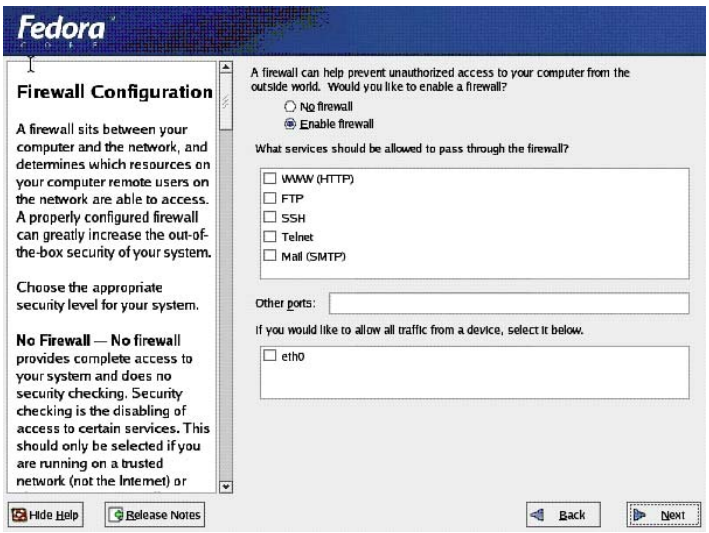
Nombre de Host: PC1

IP: 192.168.47.1

Mascara de 255.255.255.0.

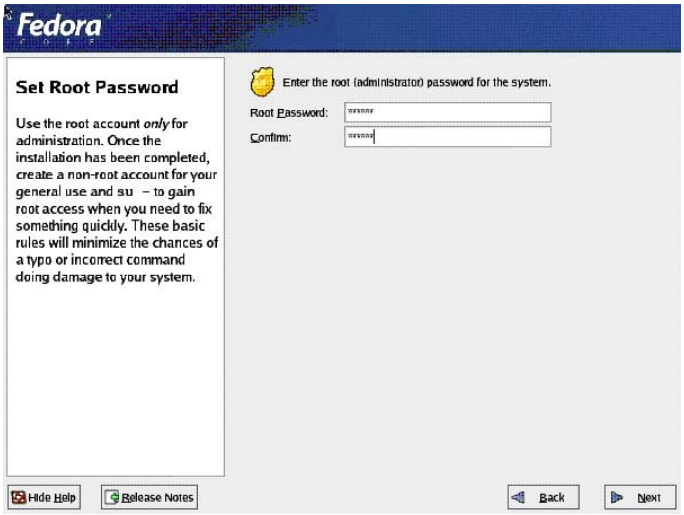
1.10.7. Firewall.

Para efectos de la monografía se va a omitir las configuraciones de Firewall, seleccione Enable Firewall, pero se recomienda configurarlo para propósitos comerciales.



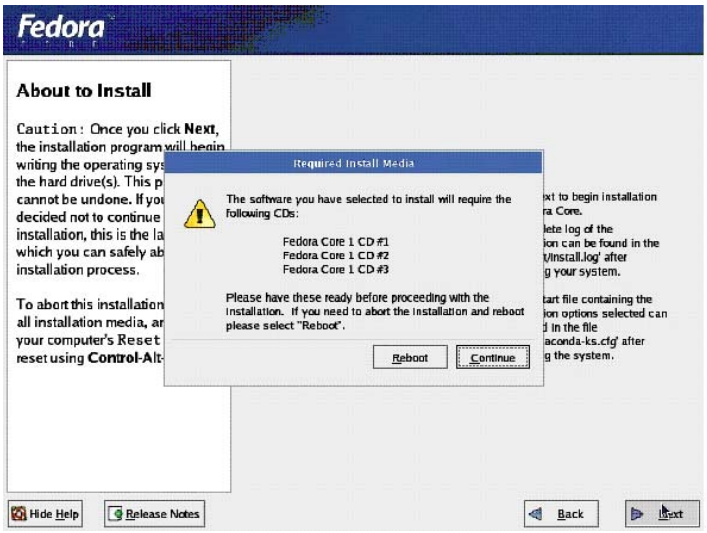
1.10.8. Ingreso de la clave de Súper Usuario.

Se necesita ingresar una clave de administrador para poder hacer configuraciones como root.



1.10.9. Inicio de instalación de los rpm`s.

Con esto estamos a punto de empezar la instalación de Linux una vez que pulsemos continuar no habrá marcha atrás.



1.11. Configuración de red.

Como mencionamos anteriormente existe la posibilidad de configurar la red en Linux una vez instalado:

- En modo gráfico pulsamos comenzar / configuración del sistema / red, esto nos mostrara una ventana en donde ingresamos el IP, mascara como lo hemos ya realizado en el momento de la instalación.

1.12. DNS (Domain Name Service)

1.12.1. Definición

El Dns (Domain Name Service) es un sistema de nombres que permite traducir de nombre de dominio a dirección IP y vice-versa. Aunque Internet solo funciona en base a direcciones IP, el DNS permite que usemos nombres de dominio que son bastantes más simples de recordar.

El sistema de nombres de dominios en Internet es un sistema distribuido, Jerárquico, replicado y tolerante a fallas. La solución a este problema es basada en un árbol que define la jerarquía entre los dominios y los subdominios. En un nombre de dominio, la jerarquía se lee de derecha a izquierda, por ejemplo, en www.ferreteria.com, el dominio más alto es com. Para que exista una raíz del árbol, se puede ver como si existiera un punto al final del nombre www.ferreteria.com, y todos los dominios están bajo esa raíz, también llamada "punto". Cada componente del dominio incluida la raíz tiene un servidor primario y varios servidores secundarios. Todos estos servidores tienen la misma autoridad para responder por ese dominio, pero el primario es el único con derecho para hacer modificaciones en él. Por ello, el primario tiene la copia maestra y los secundarios copian la información desde él.

1.12.2. Configuración del servidor DNS

1.12.2.1. Configurar el BIND

Bind es el servidor de DNS mas conocido en la actualidad, este soporta múltiples plataformas. BIND ha evolucionado a medida que el DNS lo ha hecho. Es muy importante tener siempre BIND bien actualizado, sobretodo en servidores Linux, pues en este servicio aparecen frecuentemente vulnerabilidades importantes

La configuración básica de BIND se encuentra en el fichero /etc/named.conf. Este fichero fundamentalmente describe las zonas de los dominios que controlará BIND. Para cada una de estas zonas se indica, entre otras características, el nombre del fichero

que almacena sus datos. Es en estos ficheros donde se colocan todos los records del DNS. Por defecto se agrupan en el directorio /var/named. Hay que tener en cuenta que la mayoría de la gente no necesita tener BIND instalado en su sistema. De hecho solo los servidores lo necesitan realmente.

Una vez instalado BIND, el daemon encargado de brindar el servicio es named. Este se puede manipular utilizando el script /etc/rc.d/init.d/named. A continuación veremos un ejemplo:

```
# Service named start
 Starting named:                    [  OK  ]
# Service named stop
Shutting down named:                [  OK  ]
 # Service named reload
 Reload initiated.
```

1.12.2.2. Crear una Zona

Antes de la creación de una zona hemos creído conveniente dar ha conocer los tipos de records que utilizaremos a lo largo de estas configuraciones:

SOA (Start Of Authority)

Este record es el que indica que el servidor de nombres está autorizado para la zona correspondiente. Se ubica al comienzo de cada fichero db y contiene información acerca del funcionamiento del DNS.

A (Address)

Es uno de los records más importante pues permite definir la correspondencia entre nombres y direcciones IP.

CNAME (Canonical Name)

Se utiliza para definir alias para los nombres ya definidos.

NS (Name Server)

Permite indicar cuales son los servidores de nombres de dominio de la zona correspondiente.

PTR (Domain name Pointer)

Junto a los address records es uno de los más importantes pues define las correspondencias inversas: de direcciones IP a nombres.

MX (Mail Exchanger)

Permite indicar para un nombre de dominio determinado quien es el Host que se encarga de manipular la mensajería dirigida a este.

Para crear la zona ingresamos al directorio cd/var/ named/, inmediatamente creamos un archivo llamado zone.com.ec con el comando vi:

cd / var/ named/vi zone.com.ec

Dentro del archivo completamos la información que necesitaremos:

```
@       IN      SOA   pc1.com.ec.   root.pc1.com.ec(

                                        2005011001; serial
                                        10800; refresh
                                        3600; retry
                                        604800; expire
                                        86400); ttl

                                IN      NS      pc1.com.ec.
                pc1.com.ec.     IN      A       192.168.47.1
        www.ferreteria.com.ec.  IN  CNAME   pc1.com.ec.
```

Es importante tener en claro que después del nombre del equipo (pc1.com.ec) agregar un punto al final teniendo como resultado **pc1.com.ec.,** debido a que el Host rellena lo faltante con información de la zona, es decir, si se tiene URL como **www.pc1.com.ec** el equipo agregara **.com.ec** quedando duplicado **www.pc1.com.ec.com.ec**

1.12.2.3. Habilitar la Zona

Para habilitar la zona necesitamos editar el archivo named.cof que se encuentra en el directorio /etc. Luego procedemos a aumentar información debajo de zone “localhost” en el archivo vi/etc/named.conf.

```
zone “com.ec”   in {
                        type master;
                        file”zone.com.ec”;
                        alow_update{none; };
                        }
```

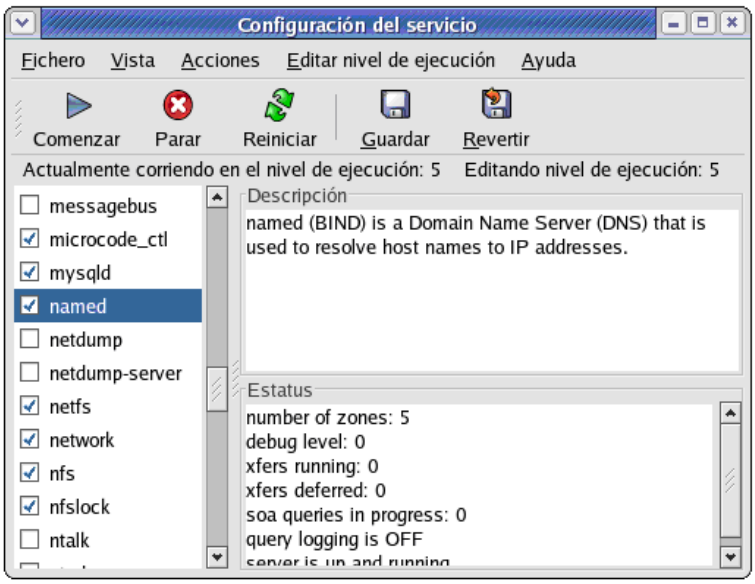
1.12.2.4. Probar el servidor DNS

- Reiniciar el servidor digitando en un Terminal service named restart.

- Digitamos el comando **dig pc1.com.ec** para que el Host devuelva el IP del servidor DNS activo, este resultado es producto de que el sistema se fija en el archivo /etc/resolv.conf.

1.12.2.5. Iniciar el Servidor DNS al iniciar la máquina

Para evitarnos la molestosa tarea de levantar el servidor cada vez que se encienda la maquina podemos iniciar todos los demonios que necesitemos mediante configuración grafica en: comenzar/configuración del sistema/configuración de servidores/servicios, se abre una ventana donde están todos los servicios disponibles, haciendo un click en cada uno de ellos lo seleccionamos, pulsamos iniciar servicio y luego guardamos los cambios.



1.13. Apache

Apache es el servidor Web que es utilizado por su fácil instalación, manejo, confiabilidad, gran fiabilidad y extensibilidad lo que lo convierte en una herramienta potente y configurable para PHP, Mysql, que utilizaremos en el desarrollo de nuestra monografía.

La implementación de un servidor Web con Apache nos va ha permitir explotar las avanzadas características, que lo han hecho líder en su campo. Aquí conoceremos todos los aspectos de la configuración del servidor Apache de Linux.

1.13.1. Instalación

Lo primero que debemos hacer es conseguirnos los paquetes necesarios, y que mejor para ello que dirigirnos a las páginas Web en donde podemos descargarlos:

- Apache: www.apache.org

- apache-1.3.x.tar.gz

Para poder realizar el proceso de instalación hay que tener acceso como root a la máquina Linux y seguir los siguientes pasos:

- Lo primero que debemos crear un directorio de instalación, aunque lo normal sería que lo hiciéramos en /usr/local, /urs/src, o bien en /opt. Como hay que escoger uno, se recomienda el primero, /usr/local, aunque el proceso sería el mismo si nos declináramos por cualquier otro. Supongamos que ya nos hemos conseguido los paquetes y los tenemos en el directorio /root/install, lo primero que hacemos es descomprimirlos:

```
cd /usr/local
tar zxvf /root/install/apache-1.3.x.tar.gz
```

- Creamos enlaces sencillos a código fuente:

```
ln -s /usr/local/apache-1.3.x  /usr/local/apache
```

- Preparamos la fuentes par la compilación de Apache:

```
cd /usr/loca/apache
./configure --prefix=/usr/local/apache
```

- Compilamos Apache:

```
cd /usr/local/apache
./configure --prefix=/usr/local/apache \
```

make
make install

- Ahora ya sólo nos queda arrancar el servidor, pero primero copiamos el script de arranque en /etc/rc.d/init.d

cp/usr/local/apache/bin/apachecte /etc/rc.d/init.d/apache
/etc/rc.d/init.d/apache start

1.14. Enlaces Simbólicos

Los enlaces le permiten dar a un único fichero múltiples nombres. Los ficheros son identificados por el sistema por su número de inodo. Un inodo es un enlace el cual es el único identificador del fichero para el sistema de ficheros.

Un directorio es una lista de números de inodo con sus correspondientes nombres de fichero. Cada nombre de fichero en un directorio es un enlace a un inodo particular.

Nosotros utilizaremos los Enlaces Blandos para la instalación de Apache, Mysql, Php con las siguientes instrucciones:

ln -s /usr/local/apache-1.3.x /usr/local/apache
ln -s /usr/local/mysql-3.22.x /usr/local/mysql
ln -s /usr/local/php-3.0.x /usr/local/php

1.15. Permisos CHMOD

Debido a que Linux es Multiusuario, este comando nos proporciona varios permisos para proteger los ficheros del usuario. Los permisos están divididos en tres tipos:

- Permisos de lectura y representado por la letra "r".
- Permisos de escritura y representado por la letra "w".
- Permisos de ejecución y representado por la letra "x".

A su vez estos permisos pueden ser fijados para tres clases de usuarios:

- Propietario del fichero.
- Grupo al que pertenece el fichero.
- El resto de usuarios.

Para definir permisos utilizaremos el sistema llamado: "sistema octal", en este sistema los números representan permisos. Por ejemplo: 0001, 0100, 0400, 1000. El problema de este tipo de sistema de asignar permisos es que conlleva que los permisos adquiridos se expresan en un número final bastante amplio, pero para facilitar las cosas, podemos usar el sistema tan solo usando 3 valores que a continuación veremos:

0 = sin permisos.
1 = ejecución.
2 = escritura.
3 = escritura y ejecución.
4 = lectura.
5 = lectura y ejecución.
6 = lectura y escritura.
7 = lectura, escritura y ejecución.

Así por ejemplo, podríamos poner **chmod 755 fichero,** este querría decir que para el propietario del archivo o fichero tiene permisos de escritura lectura y ejecución, para el grupo tendría de lectura y ejecución.

1.16. Virtual Host

1.16.1. Introducción

Existen dos tipos de Hosts Virtuales que a continuación describiremos, conceptualizando un poco el host virtual basado en nombres que es el que utilizaremos.

El hosting virtual basado en IPs usa la dirección IP de la conexión para determinar qué host virtual es el que tiene que servir. Por lo tanto, necesitará tener diferentes direcciones IP para cada host. Si usa hosting virtual basado en nombres, el servidor atiende al nombre de host que especifica el cliente en las cabeceras de HTTP. Usando esta técnica, una sola dirección IP puede ser compartida por muchos sitios Web diferentes.

El hosting virtual basado en nombres es normalmente más sencillo, porque solo necesita configurar su servidor de DNS para que localice la dirección IP correcta y entonces configurar Apache para que reconozca los diferentes nombres de host. Usando hosting virtual basado en nombres también se reduce la demanda de direcciones IP, que empieza a ser un bien escaso. Por lo tanto, debe usar hosting virtual basado en nombres a no ser que haya alguna razón especial por la cual tenga que elegir usar hosting virtual basado en direcciones IP. Algunas de estas razones pueden ser:

- Algunos clientes antiguos no son compatibles con el hosting virtual basado en nombres. Para que el hosting virtual basado en nombres funcione, el cliente debe enviar la cabecera de Host HTTP. Esto es necesario para la versión de HTTP/1.1, y está

implementado como extensión en casi todos los navegadores actuales.

- El hosting virtual basado en nombres no se puede usar junto con SSL por la naturaleza del protocolo SSL.

- Algunos sistemas operativos y algunos elementos de red tienen implementadas técnicas de gestión de ancho de banda que no pueden diferenciar entre hosts a no ser que no estén en diferentes direcciones IP.

1.16.2. Configuración

Para la configuración del Virtual Host se edita el archivo:

vi/etc/httpd/conf/httpd.conf

Para agregar al final del archivo la siguiente información:

```
<virtualhost *:80>
        servername pc1.com.ec
        documentroot /var/www/html/monografía/configura
</virtualhost>
```

CAPITULO 2. HTML

2.4. Introducción

Como ya sabemos Html es muy fácil de usar, no se necesita de un gran programador para utilizar Html, aquí aplicaremos las sentencias y funciones mas utilizadas para la programación de una pagina Html, con el fin de que se queden en claro sus funcionalidades y sacar el mejor provecho a cada una para una implementación rápida y optima de una pagina Web.

Si bien un texto ilustrado con imágenes o gráficos, que es aun el caso más normal en las páginas que se encuentran en Internet, pueden hacerse fácilmente, hay que citar aparte los ámbitos de competencia que debe tener un autor para mantener un servidor Web. Por ello no es bueno limitarse.

2.5. Estructura

La estructura básica de un documento HTML queda de la forma siguiente:

```
<html>
<head>
<title>Título</title>
</head>
<body>
Texto del documento, menciones a gráficos, enlaces, etc.
</body>
</html>
```

2.6. Conceptos básicos

2.6.1. Formatos

Existen dos tipos de formato para los caracteres:

- Formatos físicos
- Formatos lógicos.

FORMATOS FÍSICOS

Son los que ordenan como se va a presentar el texto. Son formatos físicos por ejemplo la negrita, el subrayado.

<B> </B>	Negrita
	M Subíndice
<I> </I>	Cursiva

En las tres primeras frases está muy clara la utilización de las etiquetas: Se abre inmediatamente antes de la frase que quieres que tenga el formato indicado en la etiqueta y se cierra inmediatamente después. Pero vamos a observar detenidamente la frase "Se pueden usar varios a la vez". Lo que aquí vemos es que se pueden anidar este tipo de etiquetas. A la hora de anidar las etiquetas es muy importante tener en cuenta el orden de apertura y de cierre de ellas, así que si abro primero la negrita, luego la cursiva y por último el subrayado, deberé tener en cuenta que debo cerrar primero la última que abrí, esto es, el subrayado, luego cerraré la cursiva y por último la negrita.

FORMATOS LÓGICOS

Son los que dan una cierta calidad al texto y se apreciarán dependiendo del navegador. Aquí tenemos una lista de algunos de ellos.

<CITE> ... </CITE> Texto con formato cita
<DFN> ... </DFN> Texto con formato definición
<CODE> ... </CODE> Texto con formato código
<KBD> ... </KBD> Texto con formato teclado
<SAMP> ... </SAMP> Texto con formato ejemplo
<VAR> ... </VAR> Texto con formato variable .

2.3.2. Formato de letra.

Aquí veremos como cambiar el color el tamaño y el tipo de letra, para dar un mejor aspecto a nuestros documentos. Para ello usaremos la etiqueta <FONT>.

Hasta ahora las etiquetas que hemos visto no llevaban ningún dato adicional pero como veremos muchas etiquetas llevan esta serie de datos que se llaman atributos. Los atributos irán dentro de la etiqueta y separados con un espacio. Por ejemplo:

<FONT **Size=5 color="#F0F0F0"**>Palabra</FONT>

2.3.3. Imágenes

Insertar imágenes en nuestros documentos web es muy importante ya que tiene mucho que ver con la llamativa que tengamos para el visitador y para la apariencia de la misma. Las imágenes que se utilizan generalmente en este tipo de documentos Web son archivos de imagen comprimidos para que la carga de la página sea más rápida, nosotros utilizaremos los archivos GIF, PNG y los JPG por ser los más utilizados.

Para insertar una imagen debemos utilizar el comando image source.

```
<IMG src="imagen.gif">
```

2.3.4. Enlaces

Los enlaces son una de las partes más importantes y prácticas dentro de los documentos Web ya que permiten entrelazar diferentes documentos entre si que a su vez pueden estar entrelazados con otros, etc. Hay diferentes tipos de enlaces: enlaces a otra página, enlaces a una URL, marcadores y enlaces a una dirección de correo.

Vamos a enlazar a continuación dos documentos Web dentro de un mismo directorio. Para realizar este vínculo entre páginas utilizaremos el siguiente código:

```
<A href="control.html">Ir a otra página</A>
```

2.3.5. Tablas

Las tablas son muy útiles a la hora de ordenar y estructurar nuestros documentos Web. Su estructura:

```
<table>
  <tr>
      <td>
      </td>
  </tr>
</table>
```

Para definir una anchura y altura a la tabla se le dan valores a los atributos width y height.

<TABLE border=3 widht=350 height=150>

2.3.6. Formas

Las formas, como todo lo demás en HTML se definen con un pequeño conjunto de etiquetas. Estas etiquetas simplemente definen los elementos de la forma, como campos de entrada o botones. Las formas comienzan con la etiqueta <form> y terminan con la etiqueta </form>. En la forma, puedes poner además el código HTML que desees, pero puedes usar también estas etiquetas para definir campos de entrada:

<input> define un campo de entrada de texto, casillas de verificación, botones de selección, o botones simples.

<select> define menús desplegables y cajas de selección.

La etiqueta <form> tiene un atributo action, que es el URL del script PHP a quien enviar los datos de la forma, y un atributo method, que es el método HTTP usado para enviar los datos de la forma (cualquiera de Get o Post funcionarán coordinando con tu script PHP). Así, una etiqueta <form> típica es:

<form action=" "method=post>

Usa la etiqueta <input> para crear la mayoría de los campos de una forma, así como para enviar y reestablecer botones. Tiene un gran conjunto de atributos dependiendo del tipo de atributo, el cual puede ser cualquiera de:

- text-- un campo normal de entrada (defecto)
- password-- idéntico a text, pero la entrada del usuario no se despliega
- checkbox-- una casilla de verificación (para simples valores sí/no)
- radio-- un botón de selección (para escojer una de varias opciones)
- submit-- un botón que envía los datos de la forma, cuando la entrada de datos del usuario está lista
- reset-- un botón que reestablece todos los campos de una forma a sus valores iniciales
- image-- como submit, pero muestra una imágen como botón
- hidden-- te permite definir pares extra de nombre-valor que se envían al script CGI pero no se despliegan.

Ejemplo:

<input type=hidden name="totalsofar" value="1290.65">

Usa la etiqueta contenedora <select> para crear menús desplegables y listas desplazables. Entre <select> y </select> puedes tener solamente etiquetas <option> y su texto, el cual define items en la lista.

La etiqueta <select> tiene un atributo name como todos los campos de entrada. Por Ejemplo:

```
<select name="favecolor">
        <option>verde
        <option>aguamarina
        <option selected>esmeralda
        <option>turquesa
        <option>agua
        <option value="verde2">verde
        <option value=" verde3">verde
</select>
```

CAPITULO 3. PHP

3.10. Introducción

Todos los que nos hemos enfrentado con el diseño de páginas Web hemos echado de menos un poco más de dinamismo en ellas. Representar una página repleta de gráficos y nada más, deja de ser suficiente para ciertas aplicaciones en Internet. Estas aplicaciones requieren de cierta interactividad con el usuario, y han sido muchas las tecnologías aplicadas a este fin (formularios, CGI, etc.).

Ahora está disponible PHP, Profesional Home Pages, la solución para la construcción de Webs con independencia de la Base de Datos y del servidor Web, válido para cualquier plataforma. El objetivo es conseguir la integración de nuestras paginas HTML con aplicaciones que corran en el servidor como procesos integrados en el mismo, y no como un proceso separado, como ocurría con los CGIs. Igualmente interesa que dichas aplicaciones sean totalmente independientes del navegador (lo que no ocurría con otros lenguajes basados en scripts, como JavaScript o VisualBasic Script), independientes de la plataforma y de la Base de Datos.

PHP, está más orientado a conexiones entre páginas Web y servidores donde se almacenan toda clase de Bases de Datos.

Posee un mecanismo de seguridad que permite que varios usuarios estén corriendo scripts PHP sobre el mismo servidor. Este mecanismo está basado en un esquema de permisos de ficheros, permitiendo el acceso a aquellos ficheros que son apropiados por el mismo identificador de usuario que el del script que está intentando acceder a ese fichero, o bien cuando el fichero está en el directorio que es propiedad del mismo identificador de usuario que el del script que está intentando acceder.

PHP es un lenguaje de programación soportado por HTML. La sintaxis está heredada de C, Java y Perl. Este lenguaje está orientado para los constructores de páginas Webs, permitiéndoles crear páginas dinámicamente generadas de forma rápida por lo cual la hemos elegido para la realización de nuestra monografía.

3.11. Instalación de PHP en Linux con Apache

Antes de empezar a instalar paquetes es importante mencionar que si hemos instalado las opciones mínimas de Linux, entonces necesitaremos instalar todos los paquetes necesarios para convertir a Linux en un servidor Web, en caso contrario si la instalación fue la completa no es necesario de ninguna instalación.

INSTALACIÓN DE PAQUETES.

Primero procedemos a descargar los paquetes de instalación de PHP4.x en la dirección http://www.php.net/downloads.php, obtenido este paquete se lo descomprime en un directorio temporal:

- /tmp (para iniciar la instalación).

A continuación se verifica los módulos de http_code.c y mod_so.c ejecutando el comando httpd – l donde aparecerán los comandos antes descritos.

Ingresamos al directorio /tmp y se ejecuta el siguiente comando:

```
-./configure--with-mysql--with-
apxs2=/usr/local/apache2/bin/apxs
```

Donde el parámetro --with-mysql indica que debe ser compilado con soporte para la bases de datos mysql.

El parámetro –with-apxs2=/usr/local/apache2/bin/apxs indica el directorio ejecutable apxs utilizado para compilar módulos.

Luego se procede a ejecutar los siguientes comandos con el fin de compilar e instalar el PHP:

- make;
- make install.

Al ejecutar este último, el modulo php se copiara en el directorio:

- /usr/local/apache2/modules.

Finalmente se debe generar el archivo de configuración de PHP que es php.ini que se encuentra alojado en:

- /usr/local/lib.

También se debe hacer unas modificaciones en el archivo principal de httpd.conf de apache, verifique los siguientes renglones en el archivo:

- Loadmodule php4_module modules/libphp4.so.

Con esto le indicamos al servidor que cargue el modulo PHP. Agregando la siguiente configuración le decimos al servidor que todo documento con extensión (.php) se procesado por PHP.

- Addtype application/x-httpd-php .php.

Para probar que la instalación de PHP sea correcta se realiza lo siguiente:

- Se reinicia el servidor apache con el comando apachectl restart para que se cargue el modulo php4.

- Coloque el siguiente renglón en el archivo index.php:

<? phpinfo() ?>

Se mueve este archivo donde se encuentre el directorio de sus páginas HTML, el cual estará en:

- /usr/local/apache2/htdocs/.

Finalmente procedemos a visitar en el navegador utilizando el la dirección ya definida en el Servername; por ejemplo http://ferreteria.com.ec/index.php, al visitar esta pagina obtendremos los parámetros de configuración de PHP, así probaremos que la instalación de PHP4 este correcta.

3.12. Conceptos Básicos.

¿Qué es php?

PHP es un preprocesados de hipertextos, que para podernos ilustrar de mejor manera seguiremos el ejemplo:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php echo "Hola, Esto es un Script PHP";?>
```

```
  </body>
</html>
```

Esto es muy parecido a cualquier otro Script escrito en Perl o C. El código de PHP está incluido en tags especiales "<?,?>".

Lo que hace diferente a PHP es que el código que se deba ejecutar se ejecuta siempre en el servidor.

Así, al ejecutar el script anterior, el cliente recibirá sólo los resultados de la ejecución por lo que es imposible para el cliente acceder al código que generó la página.

¿Qué se puede hacer con PHP?
En el nivel más básico PHP es equiparable a un CGI cualquiera. La mayor fuerza de PHP es que está preparado para soportar accesos a muchos tipos de bases de datos como:

- Adabas D
- dBase
- Empress
- FiclePro
- informix
- InterBase
- Solid
- Sybase
- Velocis
- Unix dbm
- mSQL
- MySQL
- Oracle
- PosgreSQL

Para nuestro caso utilizaremos la bases de datos Mysql y Protocolo HTTP.

3.13. Variables y Constantes

3.13.1. Variables

Los conceptos a tener en cuenta en PHP con las variables son los siguientes:

- Interger: por ejemplo 1.
- Double: por ejemplo 1.3.
- String: por ejemplo “hola mundo”.
- Arrays: para guardar varios valores en una misma variable.

Parta la creación e inicialización de variables se debe comenzar con $: por ejemplo $contador.

Para la creación e inicialización de arrays se debe poner $arreglo[] en donde lo que va dentro de los corchetes indica el índice, recordemos que se empieza desde cero y puede llegar a los n elementos.

VARIABLES EXTERNAS A PHP.

Cuando trabajamos en Internet nos encontramos que debemos usar variables externas.

Existen métodos para pasar los valores de las variables al servidor como:

- Get
- Post.

Cuando utilizamos el método Get, todos los valores son pasados en una URL mostrándolos por ejemplo:

http://www.ferreteria.com/navegar.php**?nombre=Jose.**

Si en una dirección Web ponemos el símbolo **?** estamos indicando que a partir de ahí comenzaremos a pasar parámetros.

La principal diferencia entre el método Get y Post radica en la forma de enviar los datos a la pagina. Mientras que en el método Get envía los datos usando la dirección URL, el método Post los envía por la entrada estándar STDIO y no son visibles a través de la URL.

El método con el cual queremos pasar los valores se indica mediante el atributo method en la etiqueta From:

<from method=”Post” action=”navegar.php”>
< from method=”Get” action=”navegar.php”>.

3.13.2. Constantes.

Una constante en un valor que una vez definido se mantiene sin cambiar. Estas nos sirven para definir y hacer referencia a elementos que siempre tendrán el mismo valor, todas las constantes se declaran usando la función denominada define(), que se define de la siguiente forma:

- define(“nombre de la constante, valor”) .

La función define() devuelve un valor Trae en caso de éxito o False si ocurre un error.

3.14. Expresiones y operadores

EXPRESIONES:

- > mayor a
- >= mayor o igual a
- = = igual a
- < menor a
- <= menor o igual a
- ! = distinto.

OPERADORES MATEMÁTICOS:

- + Suma
- - Resta
- * multiplicación
- / división
- % modulo(calcula el resto de una división)

OPERADORES DE ASIGNACIÓN:

Otro operador para asignar es el igual (=), significa que asignamos el operador de la izquierda el valor del operador de la derecha.

OPERADORES PARA CONCATENAR:

El operador para concatenar es el punto(.) por ejemplo si queremos unir dos variables:

```
$nombre=”José”;
$apellido=”López”;
Echo($nombre.“ ”.$apellido);
```

Aquí obtendremos José López sin la necesidad de hacer echo a las dos variables por separado. Describiremos algunas de las funciones más utilizadas:

- substr() .- Extrae una porción del text de la cadena.

  substr(cadena, desde, cuantos_caracteres)

- strlen().- Devuelve el tamaño de la cadena de texto.

- Split().- Separa la cadena de caracteres, pasándole como parámetro el carácter que va a actuar de separador:

  $matriz=split (“;”,$cadena)

3.15. Sentencias.

Las sentencias condicionales son aquellas que ante el establecimiento de una condición, producirán distintas alternativas de ejecución, dependiendo si la condición se evalúa por verdadero o falso.

3.15.1. IF.

```
If(condición)
        {
         Líneas de código si se cumple
        }
Else
        {
        Líneas de código si no cumple

        }.
```

3.15.2. Switch.

```
Switch($variable)
        {
```

```
Case1:
Línea de condición;
Break;
Case2:
Linea de condición;
Break;
}
```

3.15.3. While.

```
While (condición)
{
    ……
}
```

3.15.4. For.

```
For($i; $i < nfilas; $i++)
{
Linea de condiciones
}
```

3.16. Conexiones a bases de datos

Las conexiones persistentes son enlaces SQL que no se cierran cuando la ejecución del script termina. El comportamiento de estas conexiones es el siguiente.

Cuando se invoca una conexión de este tipo, PHP comprueba si existe una conexión de este mismo tipo o por el contrario, se trata de una nueva conexión. En el caso de que exista, se procede a su uso, y en el caso de que no exista, la conexión se crea. Dos conexiones se consideran iguales cuando están realizadas sobre el mismo servidor, con el mismo usuario y la misma contraseña.

3.17. Funciones de PHP

```
function foo($arg1, $arg2, ..., $argN)
 {
  echo "Función ejemplo"
  return $value;
 }
```

Un ejemplo puede ser:

```
function hacerCafe($tipo="capuchino")
 {
  return "he hecho un café $tipo\n";
 }
```

En la llamada a esta función se obtendrá una frase u otra según se llame:

```
echo hacerCafe();


echo hacerCafe("expreso");
```

VALORES DEVUELTOS

A diferencia de C, PHP puede devolver cualquier número de valores, sólo hará falta recibir estos argumentos de la forma adecuada. Ejemplo:

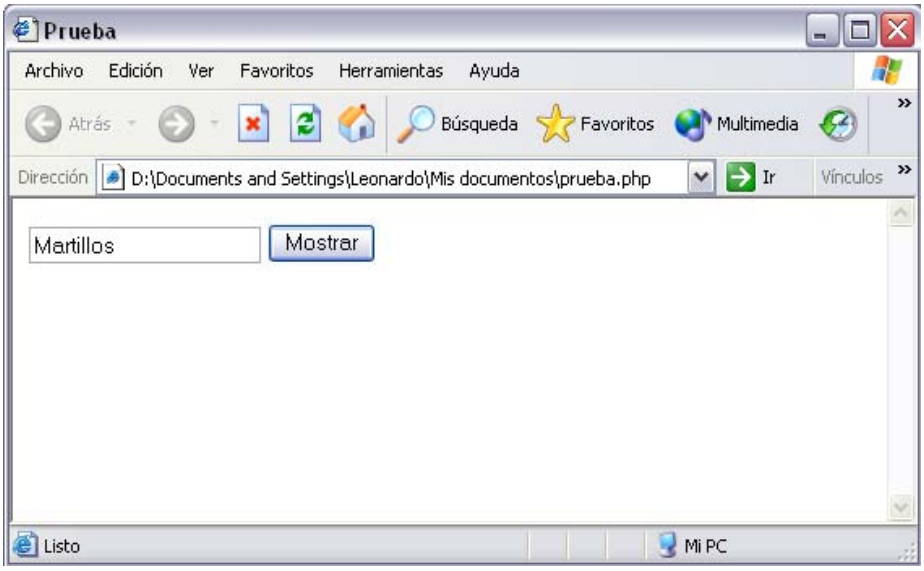
```
function numeros()
 {
  return array(0,1,2);
 }
list ($cero, $uno, $dos) = numeros();
```

Veamos un ejemplo para probar lo antes aprendido:

Creamos una página de prueba llamada prueba.php, donde digitamos el siguiente código PHP que es necesario para asignar los valores a cada variable y poder hacer uso de su contenido, al igual que mostrarlas con un echo.

```
<HTML>
<TITLE> Prueba </TITLE>
<BODY>
<? php
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
 $theValue = (! get_magic_quotes_gpc ())? addslashes ($theValue):
$theValue;
  switch ($theType) {
    case "text":
      $theValue = ($theValue! = "")? "'" . $theValue . "'": "NULL";
      break;
    case "long":
    case "int":
      $theValue = ($theValue! = "")? intval ($theValue): "NULL";
      break;
    case "double":
      $theValue = ($theValue! = "") ? "'" . doubleval ($theValue). "'":
"NULL";
      break;
    case "date":
      $theValue = ($theValue! = "") ? "'" . $theValue . "'" : "NULL";
      break;
    case "defined":
      $theValue = ($theValue! = "") ? $theDefinedValue :
$theNotDefinedValue;
      break;
  }
  return $theValue;
```

```
}
$editFormAction = $_SERVER ['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING'])) {
  $editFormAction. = "?" . htmlentities ($_SERVER ['QUERY_STRING']);
}
If ($_POST [‘boton’] = =”Mostrar”) {
        echo “Descripcion:”
        echo $_POST [‘Descripcion’];
}
?>
<form method="POST" action="<?  echo editFormAction;? >">
        <input type="text" name="Descripcion" >
        <input type="submit" name=”boton” value="Mostrar">
</form>
</BODY>
</HTML>
```



El código PHP toma del formulario los <input> y asigna a variables para luego poderlas mostrar, obteniendo una salida como:

Descripción: Martillos

3.18. Archivos y Directorios.

3.18.1. Archivos.

La importancia de utilizar archivos radica en que al cerrar el servidor se pierden los datos, se los puede utilizar mediante formularios. Para utilizar archivos se deben aplicar tres opciones básicas:

- Abrir el archivo o crear lo si no existe.
- Modificar el contenido.
- Cerrar el archivo.

Abrir Archivos.

Para abrir un archivo se utiliza:

- fopen(nombre_archivo, modo)

En donde nombre_archivo es simplemente el nombre del archivo que se desea abrir o crear.

Modo corresponde a la forma de apertura del archivo y pueden ser:

- **a** abre el archivo para agregar información al final de este y en caso de no estar creado lo crea.
- **a+** abre el archivo para agregar información y leerlo. Si no existe el archivo lo crea.
- **r** abre el archivo de modo de lectura solamente.
- **r+** abre al archivo para leer y escribir, al texto lo colocara al principio del archivo.
- **w** abre el archivo para escribir solamente, borra todo lo existente anteriormente.
- **w+** abre el archivo para escribir y leer solamente, borra todo el contenido del archivo, si este no existe lo crea.

Cerrar Archivo.

Para cerrar el archivo se utiliza la función

- fclose().

A continuación un ejemplo utilizando fopen() y fclose().

```
<?
        // se define el archivo que voy a utilizar
        $archivo=”archivo.txt”;
        //abro el archivo y creo un apuntador fp
        $fp= fopen($archivo,r);
        //se puede ejecutar algunas operaciones.
        //cierro el archivo utilizando el parámetro fp
        fclose($fp);
?>
```

Escribir un Archivo.

Para guardar datos en un archivo se utilizan las siguientes funciones:

fputs().

Esta función tambien conocida como alias de fwrite() nos permite escribir en un archivo, este recibe tres parámetros de los cuales dos son obligatorios y el tercero opcional:

- Puntero del archivo
- Texto que deseamos escribir en le archivo
- El largo de la cadena, si lo omitimos, la cadena entera será escrita

Ejemplo:

```
<?
//Defino el archivo que voy a utilizar
$archivo= “archivo.txt”;
//Defino el texto que voy a agregar
$texto= “texto a agregar al archivo”;
//opcional
$cantidad = 10;
```

```
if ($fp = fopen ($archivo,a))
{
        //escribo el texto en el archivo
        fputs ($fp, $texto, $cantidad);

}
fclose($fp);
?>
```

rewind().

Mueve la posición del puntero al inicio del archivo, recibe como parámetro a $fp al archivo que hemos abierto, si existe un error devuelve un cero.

fseek().

Esta funciona nos sitúa en una posición especifica del archivo, necesita de dos parámetros, el puntero al archivo y el numero de carácter donde queremos posesionarnos dentro del mismo.

ftell().

Esta funciona nos indica en que posición esta el puntero del archivo.

Otras Funciones.

- Copy() para copiar un archivo.
- Rename() cambiar el monbre a un archivo.
- Unlink() para borara un archivo
- File_exists() para saber si existe o no un archivo que le pasamos como parámetro.
- Filetime() devuelve la ultima fecha de acceso al archivo.

- Filesize() nos permite saber el tamaño del archivo que pasamos como parámetro.

3.18.2. Directorios.

Opendir(): Permite ingresar en un directorio para realizar operaciones como listar todos los archivos de un directorio.

Readdir(): Permite listar todos los elementos de un directorio.

Chdir(): Permite cambiarnos de directorio.

Rewinddir(): Para volver al inicio de un directorio.

Closedir(): Para liberar los recursos que utilizamos.

Getcwd(): Permite saber cual es el directorio activo en ese momento.

Mkdir(): Permite crear un directorio.

Rmdir(): Permite eliminar un directorio.

3.19. Creación y Tratamiento de Imágenes.

Imagecreate, Permite crear una nueva imagen, devolviendo un identificador a la imagen creada.

ImageColorAllocate, permite reservar un color para una imagen.
Este ejemplo podría ser invocado desde cualquier página con la línea.

Imagejpeg, permite mostrar una imagen en el navegador o crear un archivo con los datos del identificador de la imagen.

Imagedestroy, Permite destruir una imagen y liberar la memoria asociada a la imagen.

Imagesx. Permite calcular y obtener los datos acerca del ancho de una imagen.

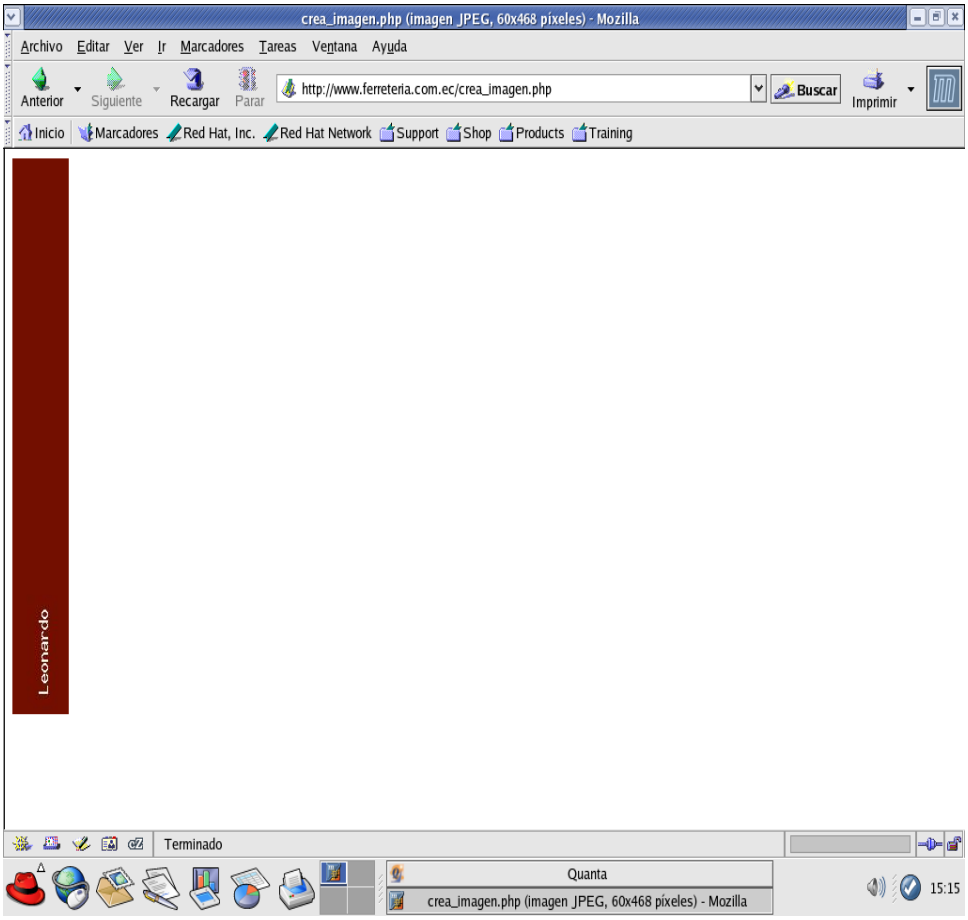
Imagesy. Permite calcular y obtener los datos acerca del alto de una imagen.

Imagecreatejpeg. Permite crear una nueva imagen a partir de un archivo en formato jpg existente.

A continuación veremos un ejemplo:

```
<?
$link=mysql_connect("localhost","root");
mysql_select_db("bd_monografia",$link);
$result = mysql_query("SELECT par_valor FROM parametro where par_codigo='1'", $link);
while ($row = mysql_fetch_row($result)){
$texto = $row[0];
}
$imagen=imagecreate(60,468);
header("Content-type: image/jpeg");
imageColorAllocate($imagen, 115,15,0);
$blanco=imageColorAllocate($imagen, 255,255,255);
imagestringup($imagen, 5,25,450,$texto,$blanco);
imageJPEG($imagen,"",75);
ImageDestroy($imagen);
?>
```

Obteniendo el siguiente resultado.



CAPITULO 4. MySQL

4.11. Introducción

Desde antes de la era de las computadoras la gente usaba bases de datos. Antes del surgimiento de las computadoras una base de datos podría ver sido una agenda que contuviera los números telefónicos de gente importante que usted conociera o podría ser un archivero con información de los registros del personal de la compañía. Actualmente las bases de datos están basadas en computadoras y se pueden encintrar virtualmente en cualquier parte. Desde las bases de datos de escritorio de su colección de discos hasta las bases de datos en la Web de grandes corporaciones, las bases de datos existen en todas las formas y tamaños.

Debido a todo esto, la industria de bases de datos ha crecido tan rápido y tan extensamente como el resto de la industria de la computación.

Hasta hace poco, las bases de datos más poderosas tenían un costo muy elevado. Podían proveer todas las herramientas y funcionalidad para manejar una empresa pero a un precio muy alto. De ahí que muchas compañías preferían utilizar bases de datos mas económicas sacrificando la funcionalidad. Además, Internet ha difundido una nueva necesidad para las bases de datos que pueden ser accedidas a través de Web, por lo cual se ha visto en la necesidad de lanzar un producto que posea las funcionalidades optimas y por su puesto su costo no sea elevado al contrario sea gratis.

Mysql es parte de esta solución. Este fue desarrollado por Tcx en 1996, creándolo por la necesidad de una base de datos relacional que pudiera manejar grandes cantidades de datos en equipos relativamente baratos. Siendo así la más rápida del mercado con un gran desempeño.

4.12. Instalación y configuración

Para la instalación requerimos del código binario, es decir un archivo comprimido llamado mysql -3.23.32 - pc - linux – gnu – i686.tar.gz que se lo puede obtener de Internet, es recomendable que se descomprima el archivo en /usr/local, ya que todos los valores predeterminados apuntan hacia esta dirección, para modificar el directorio /usr, debe tener privilegios de administrador, se recomienda que instale Mysql como root, pues parece tener menos problemas de permisos de esta manera para desempacar el archivo teclee lo siguiente:

```
cd / usr /local
tar –xvzf mysql-3.23.32 - pc - linux – gnu – i686.tar.gz
ln –s mysql-3.23.32 - pc - linux – gnu – i686 mysql
```

Una vez descomprimido y creado un enlace simbólico para el acceso rápido a la instalación:

```
cd /usr/local/mysql
./configure --witout-debug --prefix=/usr/local/mysql
make
make install
cp /usr/local/support-files/mysql.server /etc/rc.d/init.d/mysql
chmod 755 /etc/rc.d/init.d/mysql
```

Creamos la base de datos del sistema MySQL.

```
/usr/local/mysql/bin/mysql_install_db
```

Arrancamos el servidor MySQL

```
/etc/rc.d/init.d/mysql start/etc/rc.d/init.d/mysql start
```

En la base de datos mysql es donde se guardaran todos los permisos y restricciones a los datos de nuestras bases de datos. La principal herramienta de MySQL es mysqladmin, la cuál como parece indicar su nombre es la encargada de la administración.

MySQL crea por defecto al usuario root con todos los permisos posibles habilitados, podemos utilizar este usuario como administrador o crear otro, por ejemplo mysqladmin. Como el usuario root lo crea sin clave de acceso, lo primero que debemos hacer es asignarle una:

```
mysqladmin -u root password "    "
```

A partir de ahora cualquier operación que hagamos como root deberemos especificar la clave si es que se tiene claro esta. Hay que destacar que entre el modificador -p y la clave no debe haber espacios.

```
mysqladmin -u root –p miclave
```

Finalmente para acceder a MySql, digitamos en el Terminal mysql, luego ingresamos contraseña si la creo.

4.13. Creación de Base de Datos y Tablas

Para crear bases de datos se utilizan los comandos:

```
mysqladmin -u root  create mibasededatos
```

Para borrarla:

```
mysqladmin -u root  drop mibasededatos
```

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

```
mysql> SHOW DATABASES;
        +----------+
        | Database |
        +----------+
        | mysql    |
        | test     |
        +----------+
2 rows in set (0.00 sec)
```

Acceder a una base de datos ya existente:

```
mysql> USE test
Database changed
mysql>
```

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

CREAR UNA BASE DE DATOS

```
mysql> CREATE DATABASE bd_monografia;
Query OK, 1 row affected (0.00 sec)
mysql> USE bd_monografia
Database changed
```

Bajo el sistema operativo Linux, los nombres de las bases de datos son sensibles al uso de mayúsculas y minúsculas, por lo tanto debemos de tener cuidado de escribir correctamente el nombre de

la base de datos. Esto es cierto también para los nombres de las tablas.

Al crear una base de datos no se selecciona ésta de manera automática; debemos hacerlo de manera explícita, por ello usamos el comando USE en el ejemplo anterior.

La base de datos se crea sólo una vez, pero nosotros debemos seleccionarla cada vez que iniciamos una sesión con mysql. Por ello es recomendable que se indique la base de datos sobre la que vamos a trabajar al momento de invocar al monitor de MySQL.

CREANDO UNA TABLA

Crear la base de datos es la parte más fácil, pero en este momento la base de datos está vacía, como lo indica el comando

SHOW TABLES:

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

Usaremos la sentencia CREATE TABLE para indicar como estarán conformados los registros.

```
mysql>  create table articulo (
  art_codigo          CHAR (10)                  not null,
  art_descripcion     CHAR (50)                  not null,
  uni_codigo          INT                        not null,
  cat_codigo          INT                                not null,
  constraint PK_ARTICULO primary key (art_codigo));
```

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

```
mysql> SHOW TABLES FROM bd_monografia;
+---------------------------------+
| Tables_in_bd_monografia |
+---------------------------------+
| articulo            |
+--------------------+
1 row in set (0.00 sec)
```

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

```
mysql> DESCRIBE articulo;
```

Field	Type	key	Null
art_codigo	char(10)	YES	no NULL
art_descripcion	char(50)		no NULL
uni_codigo	INT		no NULL
cat_codigo	INT		no NULL

4 rows in set (0.01 sec)

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

MODIFICACIÓN DE LA ESTRUCTURA DE UNA TABLA

Cambia los atributos de una columna que ya existe:

```
ALTER TABLE [nombre_de_la_tabla] CHANGE [nombre_columna]
[nombre_columna opciones_de _columna];
```

ALTER TABLE clientes CHANGE nombre nombreapellido varchar (10);

COLOCAR UNA COLUMNA AL FINAL DE LA TABLA

ALTER TABLE [nombre_de_la_tabla] ADD [columna_nueva opciones];

ALTER TABLE clientes ADD cedula varchar (10);

CREAR UNA COLUMNA AL INICIO DE UNA TABLA

ALTER TABLE [nombre_de_la_tabla] ADD [columna_nueva opciones] FIRST;

ALTER TABLE clientes ADD cedula varchar (10) FIRST;

COLOCARLA DESPUÉS DE UNA COLUMNA DADA

ALTER TABLE [nombre_de_la_tabla] ADD [columna_nueva opciones] AFTER [nombre_de_columna];

ALTER TABLE clientes ADD cedula varchar (10) AFTER nombre;

PARA ELIMINARLA COLUMNAS

ALTER TABLE [nombre_de_la_tabla] DROP [nombre_columna];

ALTER TABLE clientes DROP cedula;

4.14. Insertar, modificar y eliminar registros de tablas

INSERTAR

Mysql > INSERT INTO clientes VALUES ('Pablo','30','600',NULL);

Nota: La comilla es obligatorio para el ingreso de datos

Otra forma de ingresar datos es:

```
mysql> insert   into clientes
-> (nombre)
 -> values
 -> ('pablo');
```

MODIFICAR

```
mysql> UPDATE clientes SET sueldo="500" WHERE nombre="pablo";
mysql> UPDATE clientes set sueldo="600" where edad=30;
```

BORRAR

```
mysql> DELETE from clientes;
```

Borrar una Tabla y su estructura

```
mysql> drop table clientes;
```

4.15. Tipos de datos

Describiremos los tipos de datos más comunes y utilizados en esta monografía:

VARCHAR: Tiene un número variable de caracteres el número que se pone es el número máximo de caracteres que puede tener este número va de 1 a 255

CHAR: Tiene un número fijo de caracteres va de 1 a 255

DATE: Tipo fecha (YYYY-MM-DD)-(‘1000-01-01' a '9999-12-31')

TIME: Tipo hora (HH:MM:SS) – (-838:59:59' a '838:59:59)

DATETIME: Tipo fecha y hora (YYYY-MM-DD HH:MM:SS)

YEAR (2 o 4): Tipo año (1970 a 2069)

INTEGER (INT):Tipo numérico entero (-2147483648 a 2147483647)

FLOAT (M, D): Número real de coma flotante M es el número y D los decimales(-3.402823466E+38 a -1.175494351E-38, 0, y 1.175494351E-38 a 3.402823466E+38.)

DOUBLE (M, D): Número real de doble precisión M es el número y D los decimales (- 1.7976931348623157E+308 a -2.2250738585072014E-308, 0, y 2.2250738585072014E-308 a 1.7976931348623157E+308)

BLOB: Para grandes textos la longitud máxima es 65535 con este tipo las búsquedas de texto son sensibles a las mayúsculas.

TEXT: Para grandes textos la longitud máxima es 65535 con este tipo las búsquedas de texto NO son sensibles a las mayúsculas.

OPCIONES PARA LOS CAMPOS

Not Null:El valor no puede ser nulo en el campo

Auto_Increment: Automáticamente incrementa el número del registro anterior

Primary Key: El primary key es un campo que MySQL usa como índice. Este índice puede hacer cosas como:

- Hallar rápidamente filas que acierten una cláusula WHERE.
- Regresar filas de una tabla desde otras tablas cuando se realizan uniones.
- Esto definitivamente te va a ayudar a agilizar sus peticiones también.

4.16. Consultas

Para los datos de una tabla se utiliza la sentencia:

```
SELECT [campos que se quiere mostrar]
FROM [nombre de la tabla de la cual se extraen datos]
WHERE [condiciones para extraer datos]
```

Ejemplo:

```
mysql> SELECT * FROM clientes;
+--------+------+--------+------+
| nombre | edad | sueldo | memo |
+--------+------+--------+------+
| pablo  |   30 | 600.00 | NULL |
| pablo  | NULL |   NULL | NULL |
+--------+------+--------+------+
2 rows in set (0.00 sec)
```

CONSULTA CON CONDICIONES:

```
SELECT * FROM clientes WHERE nombre="pablo" and sueldo="500";
```

4.17. Funciones PHP de acceso a MySQL

A continuación están las funciones descritas de PHP de acceso a MySql que nosotros utilizaremos a lo largo del desarrollo del proyecto, es necesario mencionar que no son todas, si Ud. necesita mas información pude recurrir a los manuales que se describen en la bibliografía.

mysql_close

int mysql_close (int [link_identifier]);

Devuelve: TRUE si se ha cerrado correctamente, FALSE en caso de error.

mysql_close cierra la conexión a la base de datos MySQL asociada al idenficador de conexión especificado. Si no se es especifica un identificador de conexión, se asume la de la última conexión abierta.

Note que esta función no es normalmente necesaria en conexiones no-persistentes (abiertas con mysl_connect) ya que éste se cerrará automáticamente al final de la ejecución del script o página. La función mysql_close no cierra una conexión persistente.

mysql_connect

int mysql_connect (string [hostname] , string [username] , string [password]);

Devuelve: un identificador de conexión, o FALSE en caso de error.

mysql_connect establece una conexión a un servidor de MySQL. Todos los argumentos son optativos, y si no se especifican, los valores por defecto son (' el localhost', nombre del usuario del usuario que posee el proceso del servidor, la contraseña vacía). La cadena hostname también puede incluir un número del puerto, "hostname: port".

En caso de realizar una segunda llamada a mysql_connect con los mismos argumentos, no se establecerá ninguna nueva conexión, sino se devolverá el identificador de conexión de la ya existente.

La conexión al servidor se cerrará en cuanto la ejecución del script acabe, a menos que la cerremos antes con la función mysql_close.

mysql_db_query

int mysql_db_query (string database, string query, int link_identifier);

Devuelve: un identificador de conexión, o FALSE en caso de error.

Ejecuta una consulta en una base de datos. Si el identificador no se especifica, la función intenta encontrar una conexión abierta con el servidor. Si no encuentra una conexión, intentará crear una (similar a mysql_connect ( ) sin argumentos).

mysql_error

string mysql_error ();

Devuelve el texto asociado al error producido en la última operación realizada por la base de datos.

mysql_fetch_array

array mysql_fetch_array(int result);

Devuelve un array con la información correspondiente al resultado de una consulta especificado por su identificador o 'false' si ya no hay más filas.

Es una versión extendida de mysql_fetch_row ( ). Además de almacenar los datos a través de índices numéricos del array, también lo hace a través de índices asociativos, utilizando los nombres de los campos como claves.

mysql_fetch_row

array mysql_fetch_row(int result);

Devuelve: una tabla o FALSE si hay error.

Devuelve un tabla con los valores de los campos de la fila actual de la consulta, la que especificar el indicador (result), y mueve el puntero interno que marca la fila actual a la siguiente fila, si no hay mas filas devuelve FALSE. El índice de la tabla comienza en 0.

mysql_field_type

string mysql_field_type (string result, int field_offset);

Devuelve el tipo del campo del indice especificado.

mysql_insert_id

int mysql_insert_id (void);

Esta función devuelve el ID (identificador) generado para los campos autonuméricos (AUTO_INCREMENTED). El ID devuelto es el correspondiente al de la última operación INSERT.

mysql_num_rows

int mysql_num_rows (string result);

Devuelve el número de filas del resultado de una consulta.

mysql_query

int mysql_query (string query, int [link_identifier]);

Ejecuta una consulta a la base de datos activa en el servidor asociado al identificador de conexión. Si no se especifica, se utiliza la última conexión abierta. Si no hay conexiones abiertas la función intenta establecer una.

Esta función devuelve TRUE o FALSE para indicar si las operaciones UPDATE, INSERT o DELETE han tenido éxito. Para la operación SELECT devuelve un nuevo identificador de resultado.

mysql_result
int mysql_result (int result, int row, mixed field);

Devuelve el contenido de la celda de un resultado. El argumento 'field' puede ser un índice o el nombre del campo correspondiente o el nombre del campo de la forma: tabla.campo. Si la columna tiene un alias ('select foo as bar from...') se utiliza el alias en lugar del nombre de la columna.

En lugar de esta función es preferible usar mysql_fetch_row(), mysql_fetch_array(), and mysql_fetch_object(), con la que obtendremos mejor rendimiento.

mysql_select_db
int mysql_select_db (string database_name, int [link_identifier]);

Devuelve: true on success, false on error

Establece la base de datos activa en el servidor. Si no se especifica identificador de conexión se utiliza la última conexión abierta. Si no hay conexiones anexión abierta. Si no hay conexiones activas, la función intenta establecer una. A partir de la llamada a

mysql_select_db las llamadas a mysql_query( ) actúan sobre la nueva base de datos activa.

4.18. Conectar a MySQL desde PHP

Se seguirán una serie de pasos para la conexión de mysql desde php.

Conexión a MySQL

```
<html>
<body>
<?php
    $link = mysql_connect ("localhost", "nobody");
    mysql_select_db ("mydb", $link);
    $result = mysql_query ("SELECT * FROM agenda", $link);
    echo "Nombre: ".mysql_result ($result, 0, "nombre")."<br>";
    echo "Dirección: ".mysql_result ($result, 0, "direccion")."<br>";
    echo "Teléfono:".mysql_result ($result, 0, "telefono")."<br>";
    echo "E-Mail:".mysql_result ($result, 0, "email")."<br>";
?>
</body>
</html>
```

En la primera línea del script nos encontramos con la función mysql_connect (), que abre una conexión con el servidor MySQL en el Host especificado (en este caso la misma máquina en la que está alojada el servidor MySQL, localhost). También debemos especificar un usuario (nobody, root, etc.), y si fuera necesario un password para el usuario indicado (mysql_connect("localhost", "root", "clave_del_root")). Si la conexión ha tenido éxito, la función mysql_connect () devuelve un identificar de dicha

conexión (un número) que es almacenado en la variable $link, sino ha tenido éxito, devuelve 0 (FALSE).

Con mysql_select_db () PHP le dice al servidor que en la conexión $link nos queremos conectar a la base de datos mydb. Podríamos establecer distintas conexiones a la BD en diferentes servidores, pero nos conformaremos con una.

La siguiente función mysql_query (), es la que hace el trabajo duro, usando el identificador de la conexión ($link), envía una instrucción SQL al servidor MySQL para que éste la procese. El resultado de ésta operación es almacenado en la variable $result.

Finalmente, mysql_result () es usado para mostrar los valores de los campos devueltos por la consulta ($result). En este ejemplo mostramos los valores del registro 0, que es el primer registro 0, que es el primer registro, y mostramos el valor de los campos especificados

4.19. Mostrar los datos de una consulta

Ahora que ya sabemos conectar con el servidor de BD, veremos como mostrar los datos por pantalla.

```
<html>
<body>
<?php
$link = mysql_connect ("localhost", "nobody"odigo">$link = mysql_connect ("localhost", "nobody");
mysql_select_db ("mydb", $link);
$result = mysql_query ("SELECT nombre, email FROM agenda", $link);
echo "<table border = '1'> \n";
echo "<tr> \n";
echo "<td><b>Nombre</b></td> \n";
echo "<td><b>E-Mail</b></td> \n";
echo "</tr> \n";
while ($row = mysql_fetch_row ($result)) {
echo "<tr> \n";
```

```
echo "<td>$row [0] </td> \n";
echo "<td>$row [1] </td> \n";
echo "</tr> \n";
}
echo "</table> \n";
?>
</body>
</html>
```

En este script hemos introducido dos novedades, la más obvia es la sentencia de control while (), se ejecuta mientras la condición sea verdadera. En esta ocasión while () evalúa la función mysql_fetch_row (), que devuelve un array con el contenido del registro actual (que se almacena en $row) y avanza una posición en la lista de registros devueltos en la consulta SQL.

La función mysql_fetch_row () tiene un pequeño problema, es que el array que devuelve sólo admite referencias numéricas a los campos obtenidos de la consulta. El primer campo referenciado es el 0, el segundo el 1 y así sucesivamente.

4.20. Seguridad en MySql

El sistema de permisos MySQL se guarda en una base de datos llamada mysql, la cuál se componen de cinco tablas: host, user, db, tables_priv, colums_priv.

La tabla user contiene información sobre los usuarios, desde que máquinas pueden acceder a nuestro servidor MySQL, su clave y de sus diferentes permisos.

La tabla host nos informa sobre que máquinas podrán acceder a nuestro sistema, así como a las bases de datos que tendrán acceso y sus diferentes permisos.

Finalmente, las tablas db, tables_priv, columns_priv nos proveen de un control individual de las bases de datos, tablas y columnas (campos).

A continuación mostramos el contenido de cada tabla:

Tabla **user**

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
User	char(16)	
Password	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Reload_priv	enum('N','Y')	N
Shutdown_priv	enum('N','Y')	N
Process_priv	enum('N','Y')	N
File_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla **host**

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N

Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla **db**

CAMPO	**TIPO**	**POR DEFECTO**
Host	char(60)	
Db	char(32)	
User	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

He aquí una breve descripción de los diferentes permisos:

- **Select_priv:** Permite utilizar la sentencia SELECT
- **Insert_priv:** Permite utilizar la sentencia INSERT
- **Update_priv:** Permite utilizar la sentencia UPDATE
- **Delete_priv:** Permite utilizar la sentencia DELETE
- **Create_priv:** Permite utilizar la sentencia CREATE o crear bases de datos.
- **Drop_priv:** Permite utilizar la sentencia DROP o eliminar bases de datos.

- **Reload_priv:** Permite recargar el sistema mediante mysqladmin reload.
- **Shutdown_priv:** Permite parar el servidor mediante mysqladminPermite parar el servidor mediante mysqladmin shutdown
- **Process_priv:** Permite manejar procesos del servidor
- **File_priv:** Permite leer y escribir ficheros usando comando como SELECT INTO OUTFILE y LOAD DATA INFILE
- **Grant_priv:** Permite otorgar permisos a otros usuarios
  Index_priv: Permite crear o borrar índices.

- **Alter_priv:** Permite utilizar la sentencia ALTER TABLE

Si dejamos en blanco los campos user, host o db, haremos referencia a cualquier usario, servidor o base de datos. Conseguiremos el mismo efecto poniendo el símbolo % en el campo.

CAPITULO 5. CGI

5.4 Descripción

La interfaz de pasarela común (Common Gateway Interface, CGI) es un protocolo genérico que permite extender las capacidades de HTTP. Los programas en CGI añaden funcionalidad al servidor Web, funcionalidad que podría abrir agujeros de seguridad en el servidor, ya que una aplicación en CGI mal diseñada podría permitir acceso total o parcial al servidor.

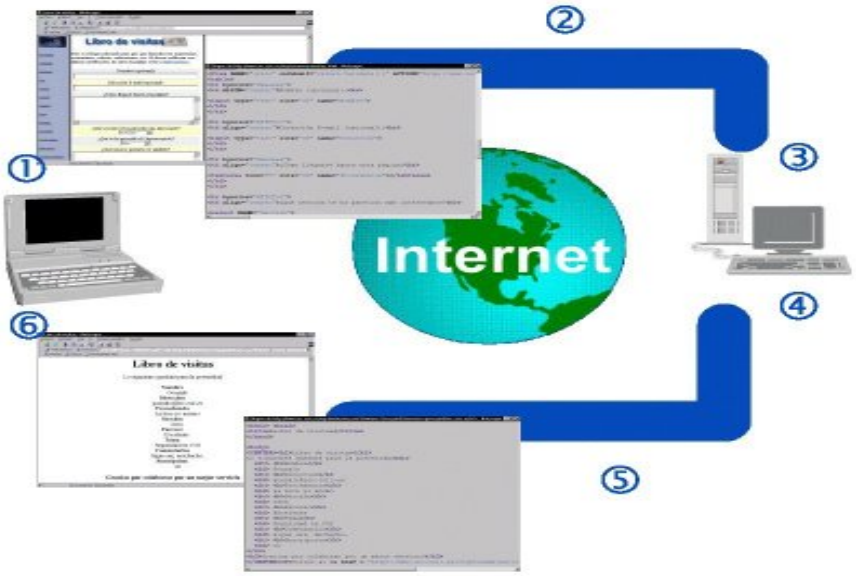
Los primeros se consideran escritos en algún lenguaje compilado como C, mientras que los segundos son los escritos en un lenguaje interpretado como PHP. Más adelante discutiremos las ventajas e inconvenientes desde el punto de vista de la seguridad que plantea cada una de las dos formas de escribir las aplicaciones en CGI (programas o guiones).

En general, es necesaria la presencia de dos elementos, una página Web en formato HTML con un formulario donde el usuario introduce sus datos, y un programa CGI en el servidor, que recibe y procesa los datos del usuario.

A continuación se muestra desde una perspectiva de alto nivel cómo funciona el mecanismo de entrega y procesamiento de datos en un programa CGI.

Mecanismo de entrega y procesamiento de datos

En la figura he representado de manera simplificada el proceso que tiene lugar desde que al usuario se le presenta el formulario de entrada de datos hasta que obtiene la página de resultado de procesar esos datos:



Al usuario se le presenta a través de su navegador una página en formato HTML que contiene un formulario, codificado mediante las etiquetas <FORM> y </FORM>. Una vez que el usuario ha rellenado los campos del formulario, pulsa el botón de enviar, que invoca al programa CGI en el servidor. Los datos del formulario se envían al servidor utilizando bien el método POST o el método GET.

Los datos llegan hasta el servidor a través de la Internet, donde la aplicación CGI invocada los procesa. Una vez convenientemente procesados, el programa CGI debe ejecutar alguna acción o generar una respuesta, generalmente en la forma de una página en formato HTML, que será enviada de vuelta al navegador.

Se añaden las cabeceras HTTP necesarias y se envía de vuelta a través de Internet el archivo HTML con la respuesta a los datos del formulario convenientemente presentada.

El navegador Web recibe el archivo HTML y lo visualiza en pantalla. Es común que la página contenga alguna información relacionada con el resultado del procesamiento de los datos, junto con algún enlace para regresar a la página del formulario, en el paso 1 de la figura.

Es importante resaltar que no es imprescindible la presencia de un formulario para invocar desde el navegador al CGI del servidor. Muchas veces, basta con pinchar en un enlace para llamar al CGI y también es posible llamarle directamente desde la ventana de URL, introduciendo el URL completo del programa CGI.

5.5 Riesgos

5.5.1 Peligros

Cuando los usuarios envían un formulario o invocan un CGI de alguna otra forma, en definitiva se les está permitiendo ejecutar remotamente un programa en el servidor. Es más, puesto que la mayoría de CGI 's aceptan datos de la entrada de usuario (bien después de rellenar un formulario o directamente desde la línea de URL), en esencia se les brinda a los usuarios la oportunidad de controlar cómo se ejecutará el CGI, de manera que podrían intentar la introducción de una serie de parámetros inesperados hábilmente manipulados para que el CGI funcionase maliciosamente.

5.5.2 Vulnerabilidad

El punto vulnerable de la programación en CGI, es decir, la amenaza que representan para la seguridad del servidor, es doble:

Por un lado, la posibilidad de que el CGI sea engañado por la entrada del usuario para que ejecute comandos imprevistos, pudiendo llegar a causar graves daños en el servidor; por otra parte la posibilidad de revelar innecesariamente información acerca del servidor, que permitirá al atacante conocer mejor la configuración del sistema y estar así mejor equipado para buscar posibles agujeros.

5.6 SSI (Server-side includes)

Los server-side includes (SSI) son directivas que se pueden incrustar en una página HTML, de manera que presentará como parte de la página el contenido de un fichero o la salida de un comando. Aunque los SSI pueden ser muy útiles y dotan de gran flexibilidad a una página Web, también pueden resultar computacionalmente costosos, pueden impedir la portabilidad de las páginas Web y tal vez más importante, pueden llegar a abrir agujeros de seguridad, ya que el autor de la página HTML decide qué programas se ejecutarán y con qué argumentos. En el caso peor, se podría llegar a ejecutar cualquier comando y así producir daños irreparables o revelar información. Por este motivo, a no ser que se tenga una buena razón para usarlos, lo más conveniente es deshabilitarlos.

5.6.1 ¿Que son?

Las directivas de Server-Side Includes (SSI) constituyen poderosas herramientas para aumentar la productividad del programador, ya que permiten al servidor modificar automáticamente el documento solicitado antes de ser enviado al navegador del cliente. Su propósito principal es insertar dentro de las páginas Web el contenido de ficheros de texto, información

dinámica sobre ficheros (como por ejemplo su tamaño) o la salida resultante de la ejecución de ciertos comandos del sistema ya preseleccionados o libres. Los Server-Side Includes pueden contener a su vez otros Server-Side Includes, de forma que se puede llegar a introducir una cantidad enorme de texto en una página con la inclusión de un único comando.

Los Server-Side Includes insertan el texto en el fichero exactamente en el mismo sitio donde aparecen. En otras palabras, se reemplazan a sí mismos en el instante en que se sirven las páginas que los contienen.

5.6.2 Peligros

En la práctica, los Server-Side Includes permiten toda clase de trucos cuando se combinan con CGI's que modifican páginas Web, como en el escenario del libro de visitas: los visitantes a una página disponen de un "libro" en el que pueden dejar un texto, que en principio podría incluir etiquetas en HTML y también, si la entrada no se filtra, includes como los siguientes:

<!--#exec cmd="rm –rf /;cat /etc/passwd" -->

<!--#include file="archivo_secreto" -->

5.6.3 Soluciones

Existen varias soluciones para paliar estas amenazas.

La más drástica pasa por deshabilitar completamente los SSI del servidor.

Otra solución menos drástica consiste en deshabilitar específicamente la ejecución de comandos con exec.

Otra posibilidad es utilizar como archivos con SSI sólo los que posean una extensión determinada, como por ejemplo .shtml. De esta forma, mientras las páginas posean extensión .html no existe riesgo de ataque mediante fallos en CGI, ya que se limitan a los archivos con extensión .shtml.

En el caso del libro de visitas, para que el ataque tenga éxito, el libro debería tener extensión .shtml, lo cual resulta muy improbable.

Por último, se pueden mantener los SSI y velar en los programas CGI por que no se produzcan entradas indeseadas, filtrando caracteres como ;, <, >, ¡, -, #, @, ‘, `, /, etc.

5.6.4 ¿Qué debemos hacer?

Visto cuáles son los riesgos a que atenerse, los objetivos que nos guiarán a la hora de escribir CGI's seguros se resumen en los siguientes:

- Los programas solamente deben ejecutar aquellas acciones para las que ha sido concebido.

- No se debe revelar más información al cliente que la que deliberadamente se desee suministrar.

- No se debe confiar en la información procedente de los datos introducidos por el cliente.

- Se debe minimizar el daño potencial al sistema en su conjunto si se produce un ataque con éxito.

5.6.5 Lenguaje a utilizar

A menudo se plantea la discusión de qué lenguaje resulta el más adecuado a la hora de escribir CGI's seguros. En una primera aproximación, puede considerarse a los lenguajes compilados como más seguros por las siguientes razones:
Impiden el acceso remoto al código fuente del CGI, lo cual limita las posibilidades de un hacker de conocer cómo funciona internamente el programa.

Los lenguajes interpretados son más sencillos de entender y rápidos de probar (ya que no necesitan pasar por todo el proceso de compilación y enlazado), y además su manejo de cadenas es mucho más sofisticado, lo que los hace menos propensos a errores con búferes de texto.

5.6.6 Cómo enviar los datos

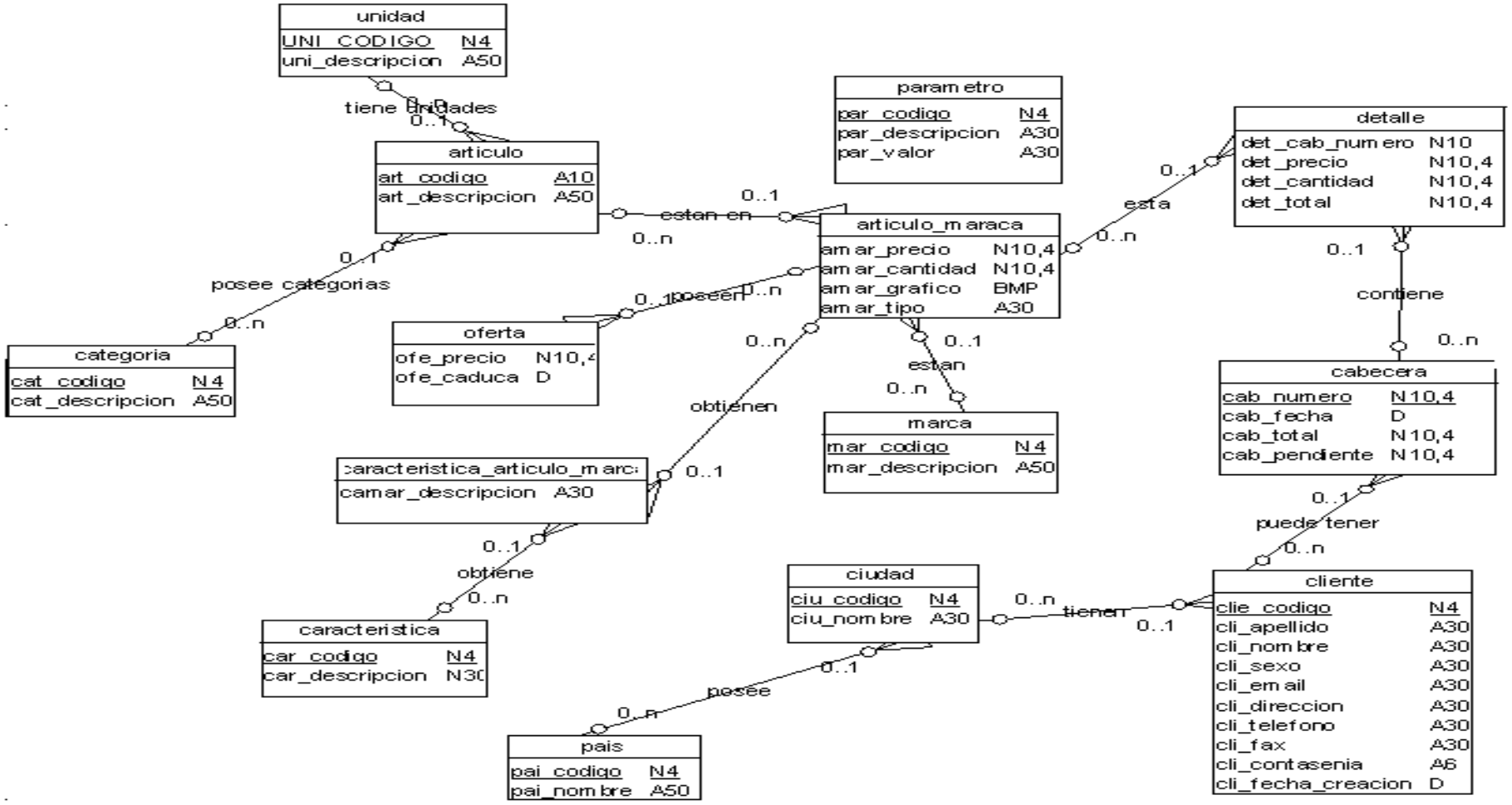
GET y POST son dos métodos empleados para enviar los datos de los formularios desde el navegador al servidor Web, especificados mediante la directiva METHOD. La principal diferencia entre POST y GET es que el CGI recibirá los datos enviados con POST leyendo la entrada estándar, mientras que los enviados con GET se recibirán por líneas de comandos y la variable de entorno QUERY_STRING. Desde un punto de vista puramente práctico, debido a que muchos sistemas operativos ponen límite a la longitud

de la línea de comandos, suele ser mejor usar POST, reservando GET para formularios con pocos datos.

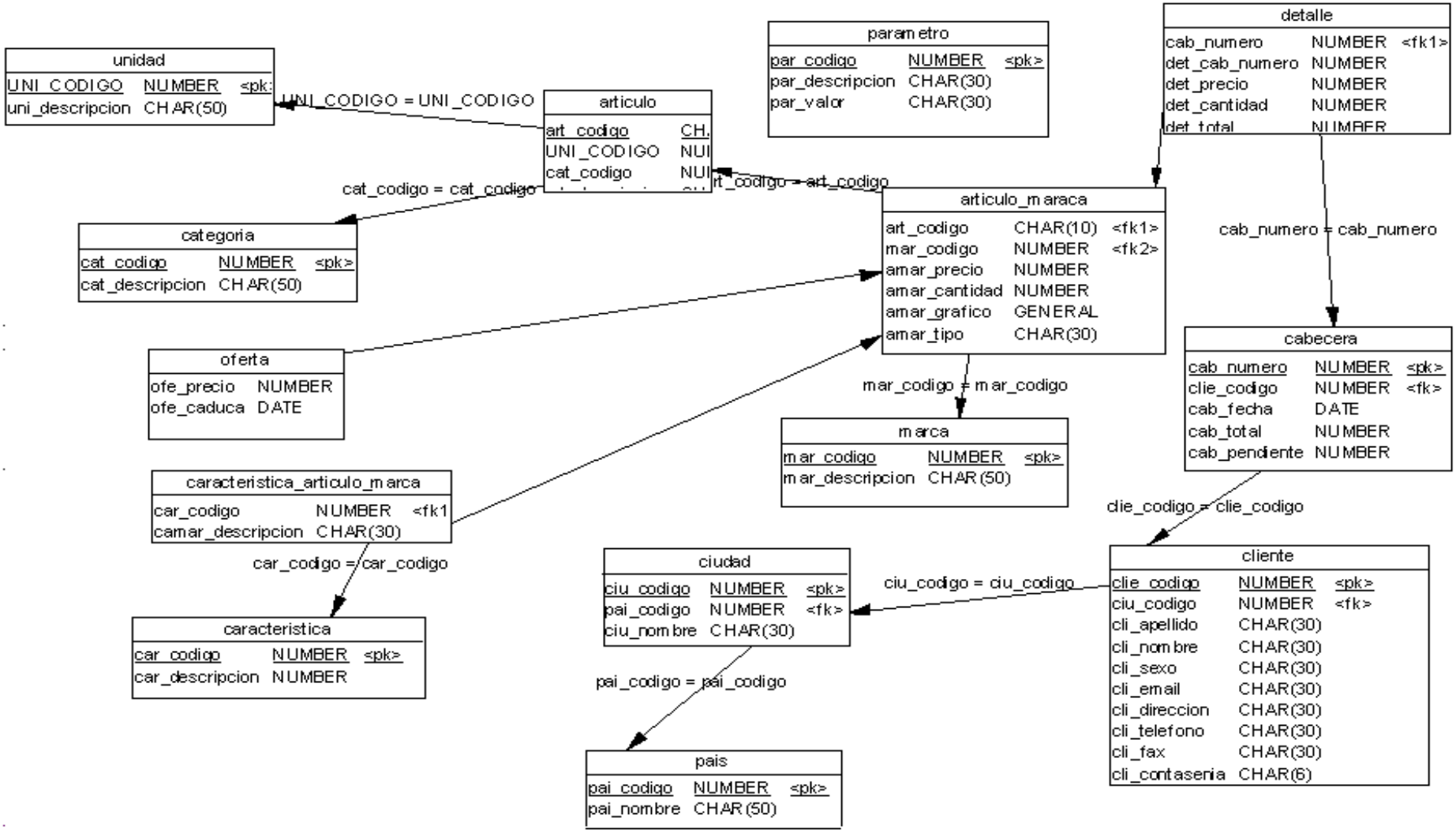
Desde el punto de vista de la seguridad la verdad es que da igual uno u otro, si bien las peticiones enviadas mediante GET quedan registradas en los ficheros de registro, con todos sus argumentos, por lo que si se enviase información confidencial sin cifrar, estos datos quedarían en el fichero log, donde a lo mejor alguien no autorizado podría husmear posteriormente.

CAPITULO 6. DESARROLLO DEL SITIO WEB

6.1 Modelo Conceptual



6.2 Modelo Físico



6.3 Diccionario de Datos.

Tabla	Atributo	Descripción	Tipo	Tamaño	Pk	Fk	Referencia
Unidad	uni_codigo	Codigo de la unidad	int	4	x		
Unidad	uni_descripcion	Descripción de unidad	char	50			
Categoría	cat_codigo	Codigo de categoria	int	4	x		
Categoría	cat_descripcion	Descripción	char	50			
Marca	mar_codigo	Codigo de marca	int	4	x		
Marca	mar_descripcion	Nombre de marca	char	50			
Pais	pai_codigo	Codigo de pais	int	4	x		
Pais	pai_nombre	Nombre de pais	char	30			
Caracteristica	car_codigo	Codigo de caracteristica	int	4	x		
Caracteristica	car_descripcion	Descripcionde caracteristica	char	30			
Ciudad	ciu_codigo	Codigo de ciudad	int	4	x		
Ciudad	ciu_nombre	Nombre de ciudad	char	30			
Ciudad	pai_codigo	Codigo de pais	int	4		x	Pais
Articulo	art_codigo	Codigo de articulo	char	10	x		
Articulo	art_descripcion	Descripción de articulo	char	50			
Articulo	uni_codigo	Codigo unidad	int	4		x	unidad
Articulo	cat_codigo	Codigo de categoria	int	4		x	categoria

Tabla	Atributo	Descripción	Tipo	Tamaño	Pk	Fk	Referencia
Articulo_marca	art_codigo	Código articulo	char	10		x	articulo
Articulo_marca	mar_codigo	Código marca	int	4		x	marca
Articulo_marca	amar_precio	Precio articulo	double	10,4			
Articulo_marca	amar_cantidad	Cantidad articulo	double	10,4			
Articulo_marca	amar_grafico	Imagen del artículo	midiumBlob				
Articulo_marca	amar_tipo	Tipo de imagen	char	30			
Cliente	cli_codigo	Código cliente	int	4	x		
Cliente	cli_apellido	Apellido cliente	char	30			
Cliente	cli_nombre	Nombre cliente	char	30			
Cliente	cli_sexo	Sexo cliente	char	1			
Cliente	cli_email	Email cliente	char	30			
Cliente	cli_direccion	Dirección cliente	char	50			
Cliente	cli_fecha_creacion	fecha de creación	date				
Cliente	cli_telefono	Teléfono cliente	char	30			
Cliente	ciu_codigo	Código ciudad	int	4		x	cuidad
Cliente	cli_contrasenia	Contraseña cliente	char	6			
Oferta	art_codigo	Código oferta	char	10		x	articulo
Oferta	mar_codigo	Código marca	int	4		x	marca
Oferta	ofe_precio	Precio oferta	double	10,4			
Oferta	ofe_caduca	Caducidad de la oferta	date				
Cabecera	cab_numero	Numero cabecera	double	10	x		
Cabecera	cab_fecha	Fecha cabecera	date				
Cabecera	cli_codigo	Código cliente	double	10		x	cliente

Tabla	Atributo	Descripción	Tipo	Tamaño	Pk	Fx	Referencia
Cabecera	Cab_total	Total cabecera	double	10,4			
Cabecera	Cab_pendiente	Cabecera Pendiente	double	10,4			
caracteristica_articulo_marca	art_codigo	Codigo articulo	char	10		x	Articulo
caracteristica_articulo_marca	Mar_codigo	Codigo marca	int	4		x	Marca
caracteristica_articulo_marca	camar_descripcion	Caracteristica de articulo	char	30			
caracteristica_articulo_marca	car_codigo	Codigo caracteristica	int	4		x	Caracteristica
Detalle	det_cab_numero	Numero de detalle	double	10			
Detalle	Cab_numero	Numero cabecera	double	10		x	Cabecera
Detalle	art_codigo	Codigo articulo	char	10		x	Articulo
Detalle	Mar_codigo	Codigo marca	int	4		x	Marca
Detalle	det_precio	Precio detalle	double	10,4			
Detalle	det_cantidad	Cantidad de detalle	double	10,4			
Detalle	det_total	Total de detalle	double	10,4			
Parametro	Par_codigo	Código de Parametro	int	4	x		
Parametro	Par_descripcion	Descripción de atributo	char	30			
Parametro	Par_valor	Contenido del registro	char	30			

6.4 Creando la Bases de Datos en MySQL.

En le disco adjunto a esta monografía se encuentra el archivo bd_monografia.sql, el cual contiene la información para crear las tablas de nuestra bases de datos. Para esto:

Copiamos el archivo a cualquier directorio en Linux, por ejemplo:

/var/www/html/monografía/diseño

Ingresamos a mysql en un terminal digitando Mysql

Una vez dentro de mysql digitamos:

create batasbases bd_monografia;

Para poder utilizar la base de datos creada utilizamos:

use bd_monografia;

Es hora de ejecutar el archivo sql para crear nuestras tablas con la siguiente instrucción:

sourse /var/www/html/monografía/diseño/bd_monografia.sql

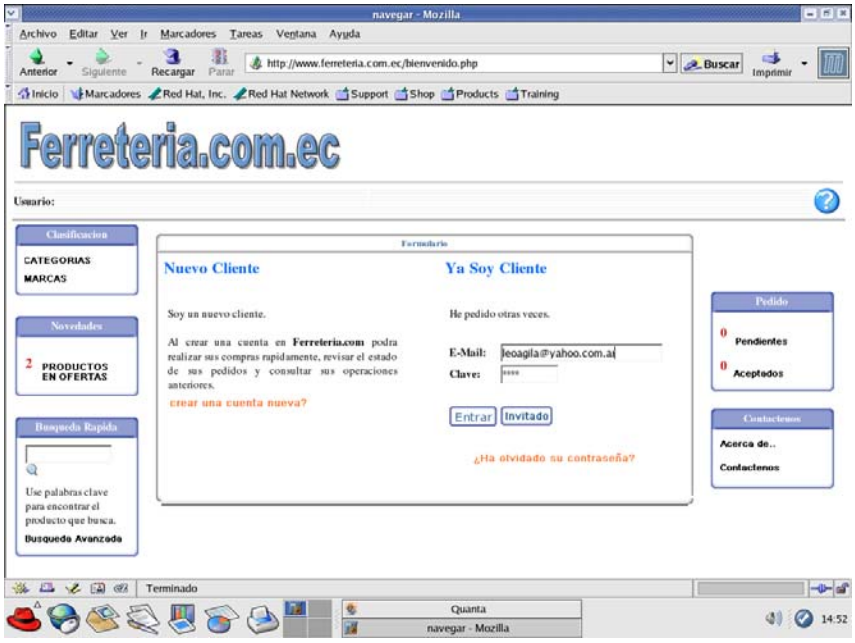
Para poder ver las tablas creadas digitamos:

show tables from bd_monografia.

6.5 Crear la página Web utilizando HTML y PHP.

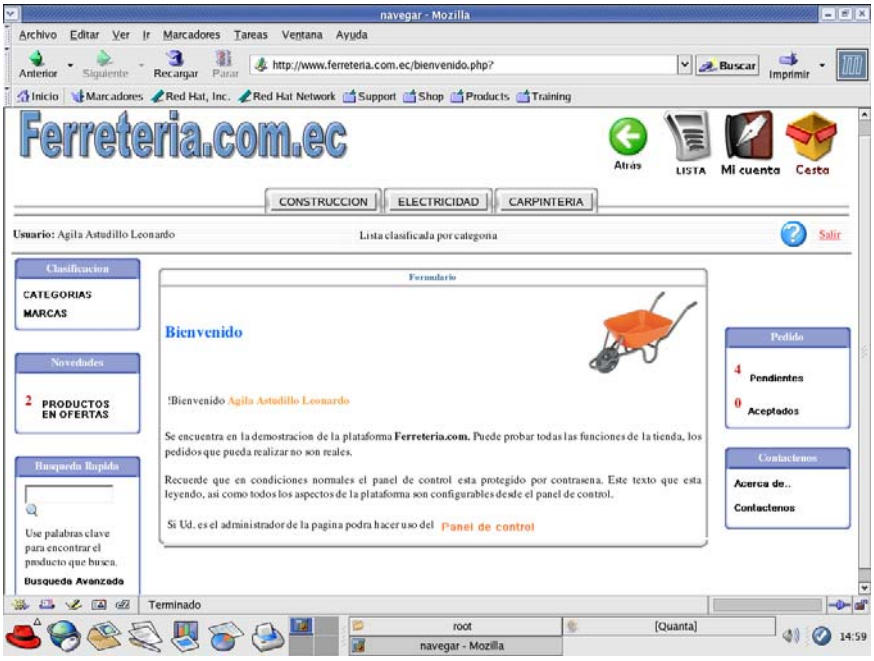
En le disco adjunto a esta monografía se encuentra la carpeta de configura dentro de monografía en la cual se encuentra todo el código fuente de las paginas que están cargadas en el servidor. A continuación se mostrara la funcionalidad de la aplicación en modo cliente.

6.5.1. Implementación de la aplicación modo cliente.

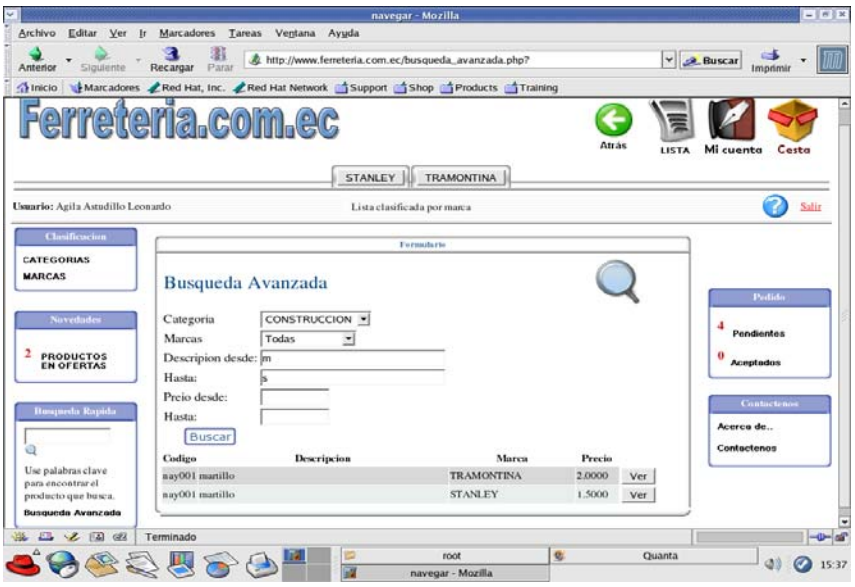


En esta pantalla se especifica si usted es un cliente nuevo podrá navegar como invitado pero con la limitación de no poder hacer un pedido. Si ya es cliente de Ferretería.com.ec, usted podrá navegar sin ninguna restricción.

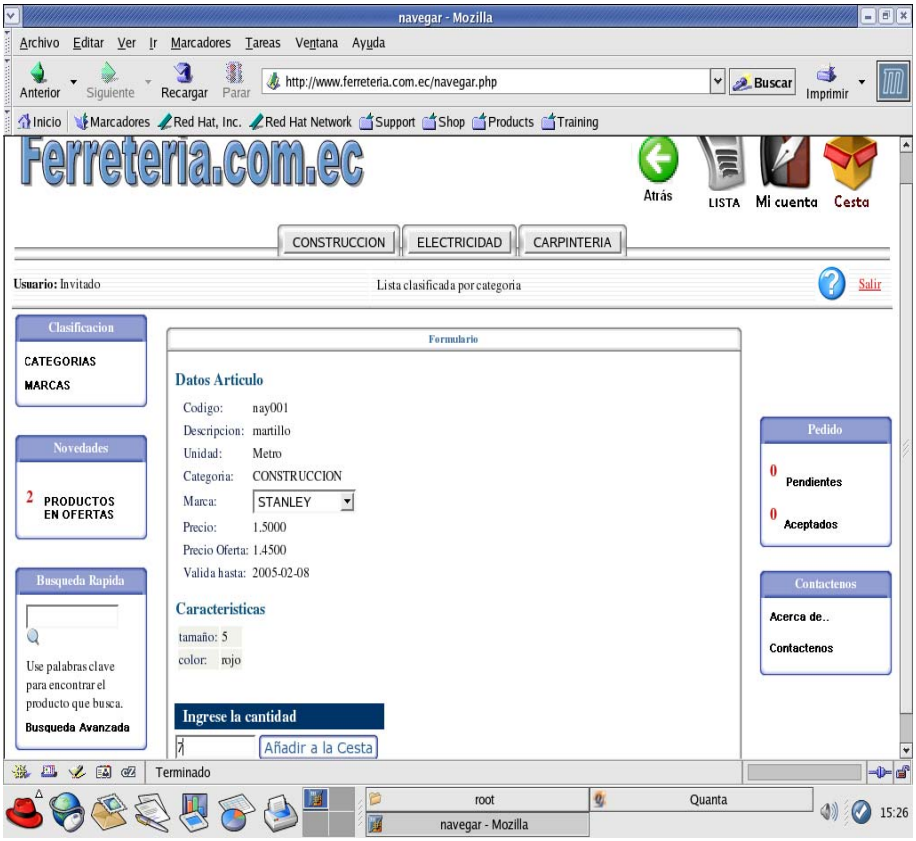
La aplicación le ofrece un menú a la izquierda en donde usted podrá clasificar los artículos de acuerdo a sus categorías y marcas, al seleccionar cualquiera de estas opciones aparecerá un menú en la parte superior de la pagina con todas las alternativas de la clasificación seleccionada. En el menú de la derecha encontraremos los pedidos pendientes junto con los que ya han sido revisados por el administrador de la empresa.



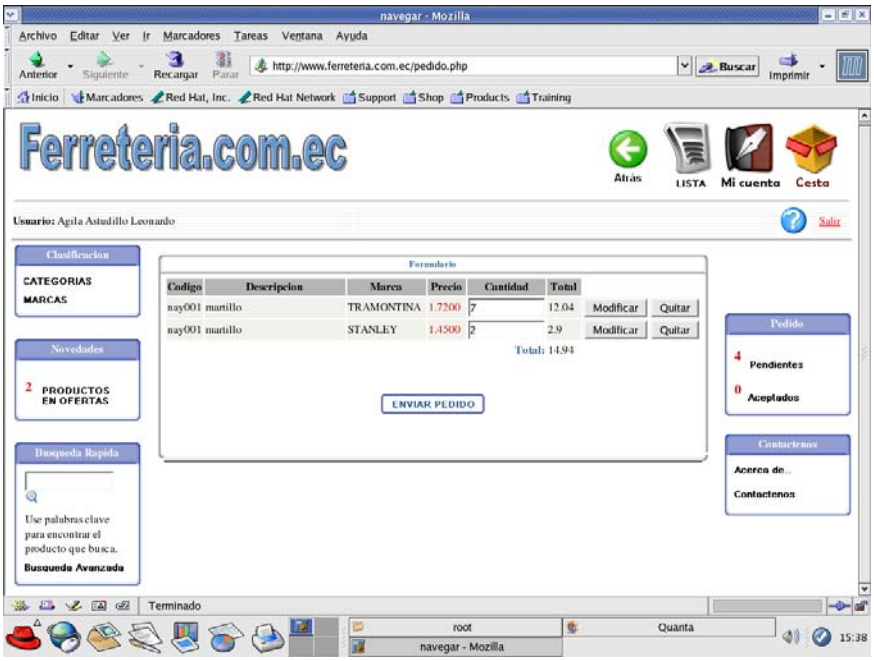
La búsqueda avanzada es una herramienta incorporada para la fácil selección de los productos, a continuación se muestra una búsqueda clasificada por categoría desde la descripción del producto que empieza con M hasta los que empiezan con S, para más información consulte la ayuda que viene en la propia búsqueda.



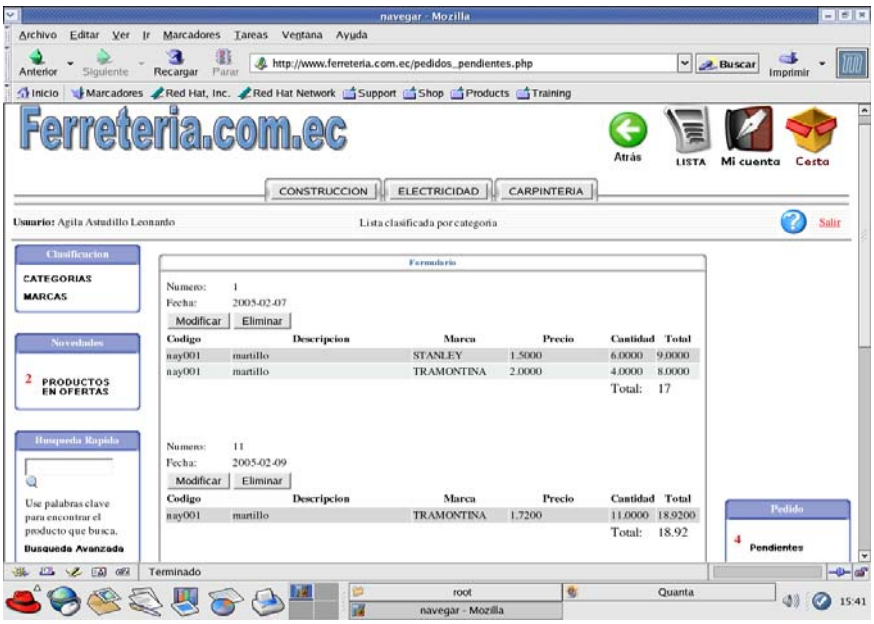
Una vez encontrado el producto pulsamos Ver para poder ingresar la cantidad que usted necesite, en esta pantalla estarán todos los datos del producto incluyendo las características y ofertas disponibles. Usted tiene la posibilidad de añadir a la cesta de cotización el artículo con solo pulsar Añadir a la Cesta, proceso que podrá ser repetido cuantos productos desee seleccionar.



Para revisar los productos seleccionados se encuentra en el menú superior Cesta, esta se presentara en una forma de detalle mostrándonos el total del pedido. Si la cotización le es conveniente podrá convertirla en pedido con solo pulsar enviar pedido.



Si desea revisar los pedidos pendientes y aceptados diríjase al menú de la derecha, donde podrá ver los pedidos pendientes y aceptados, solo pudiendo modificar o eliminar los pendientes.



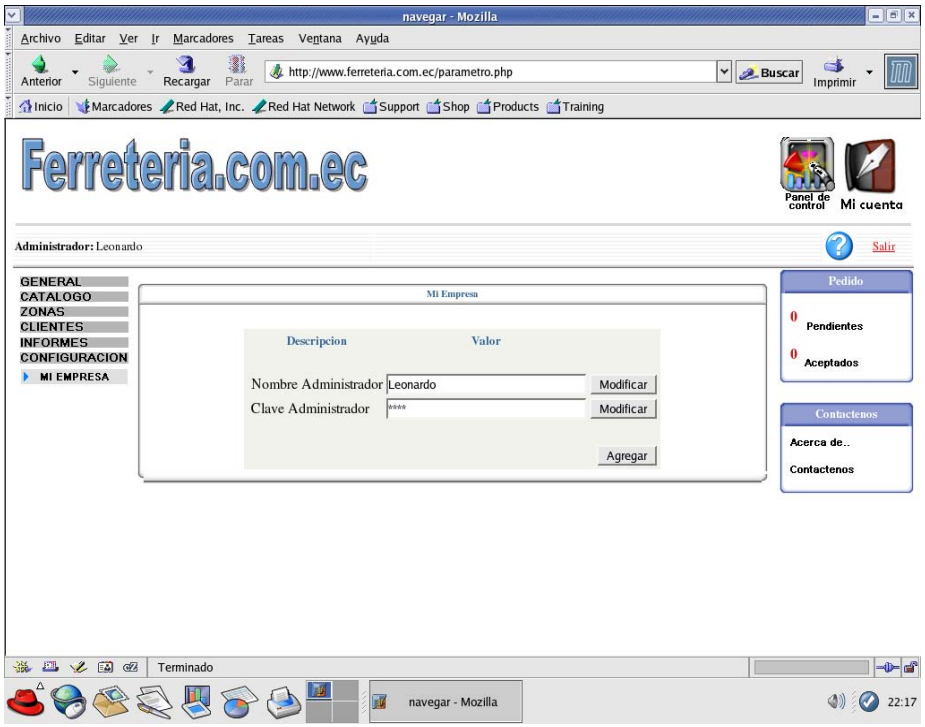
Para el control de administración de la Web implementamos las mismas funcionalidades que las antes mencionadas, respaldándose con una ayuda de mayor detalle.

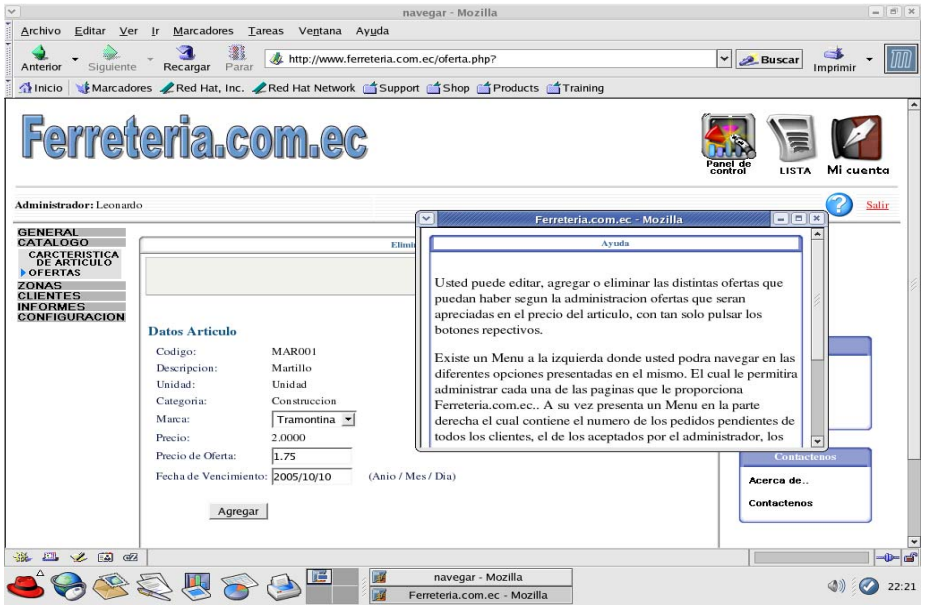
6.5.2. Implementación de la aplicación modo administrador.

La sección de configuración de la aplicación es de exclusiva responsabilidad del administrador, este tendrá una clave de acceso única, la misma que se deberá ingresar en la siguiente forma:

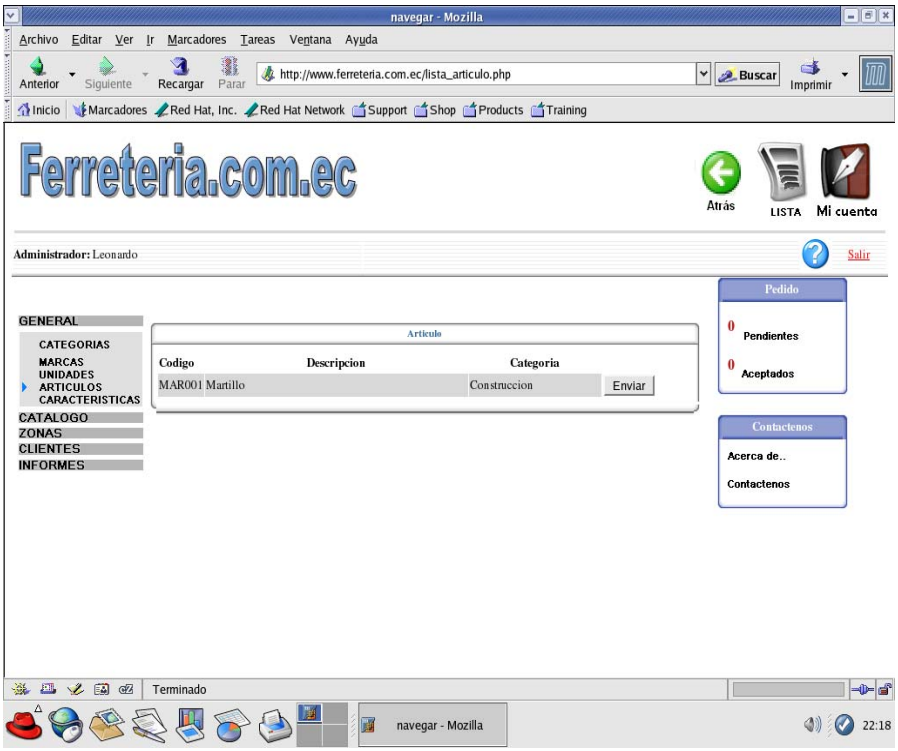
Si la calve es la correcta, la aplicación ofrecerá todas las alternativas disponibles de configuración.

El menú presentado está clasificado de tal manera que el usuario navegue sin mayor dificultad, una de las configuraciones mas importantes es la de Mi Empresa, en donde consta el nombre del administrador y su clave. Si el usuario esta perdido en sus acciones, la aplicación lo respaldará con tan solo hacer click en la imagen de ayuda.

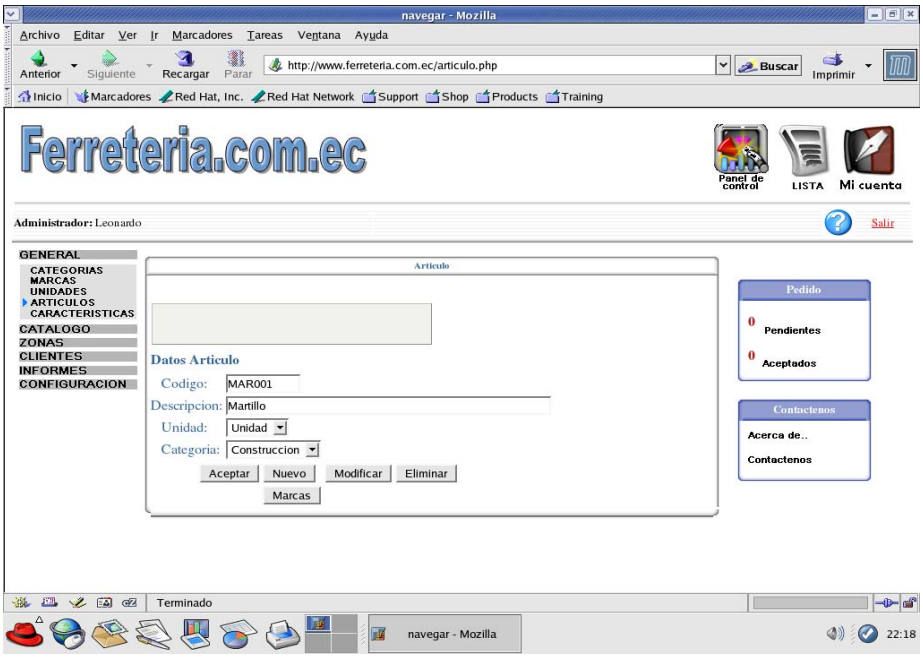


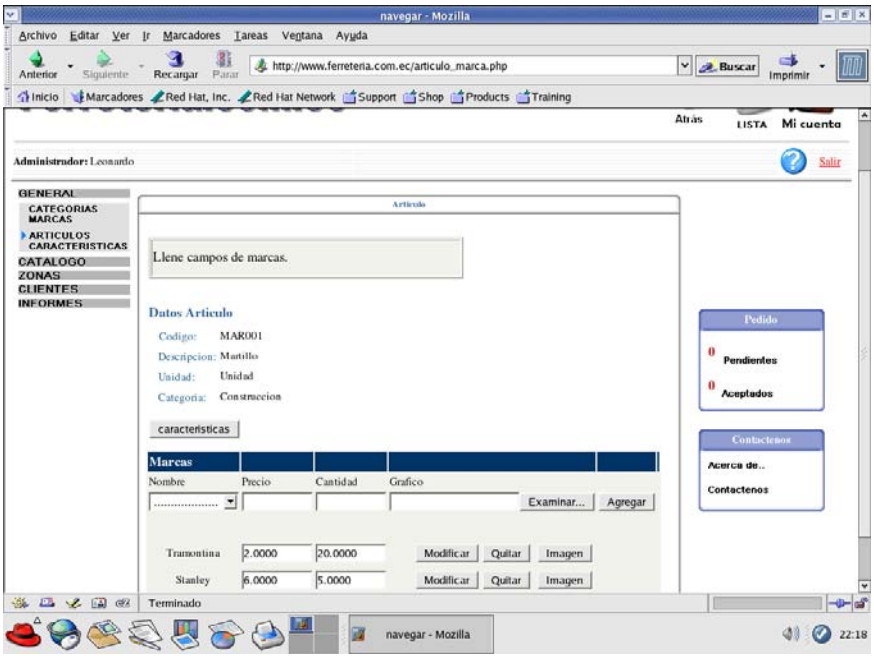


Para moverse con libertad, tiene a su disposición una lista general de los artículos creados para revisar todos sus atributos.



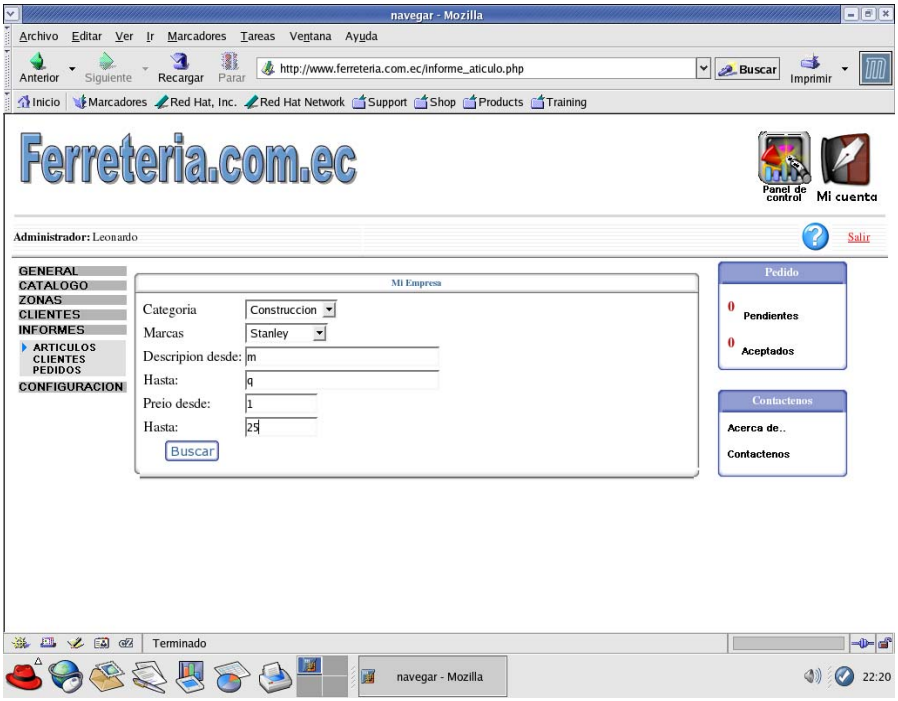
Luego de haber seleccionado el artículo este se enviara a una forma en donde haciendo click en los diferentes botones, editares y presentaremos sus atributos, como por ejemplo el botón Marcas que presentara la siguiente página:





El proceso se repite para ver tanto las características como las ofertas del artículo seleccionado.

Una de las herramientas a disposición del administrador es la generación de informes, en el cual la lógica de generación es la mima que la de la búsqueda avanzada en la sección de modo cliente antes mencionado.



CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- En la sección de cómo instalar paquetes demostramos que es posible montar un Servidor Web con recursos de distribución gratuita, lo cual significa una reducción considerable del costo de implementación.

- Linux Fedora ha demostrado ser un Sistema Operativo confiable, seguro y de fácil configuración, apto para acogerse a tecnologías independientes y en constante desarrollo, demostrándose en la sección de instalación de paquetes de diferentes versiones.

- Muchos de los productos de casas comerciales como Microsoft y otras que desarrollan software con licencias crean una dependencia a su beneficio, para que el usuario este constantemente actualizando sus productos con un costo adicional al de su licencia, siendo compatible solo con su sistema operativo y dependiente a las distintas versiones. Es por eso que al utilizar MySql y PHP evitamos crear dependencias sin dejar a un lado la eficiencia y seguridad

- Este proyecto nos a proporcionado incrementar nuestros conocimientos en nuevas herramientas de diseño y programación Web, permitiéndonos así comparar que solución es la mejor y mas barata para las solicitudes del usuario.

RECOMENDACIONES

- Al momento de instalar el sistema operativo Linux Fedora, es aconsejable seleccionar que se instalen todos los paquetes disponibles, para evitarnos conflictos en el futuro.

- Cuando configuremos los archivos en Linux, crear una copia del archivo a modificar por si hacemos algo mal, y así poder volver a la configuración estándar, puesto que casi todos los cambios que se hacen el proyecto son manuales.

- Antes de empezar las pruebas revisar que todos los demonios de Linux estén funcionando y corriendo correctamente.

- Para el análisis y diseño de la base de datos, utilizar Power Designer una excelente herramienta que nos permite crear el modelo conceptual fácilmente y generar automáticamente el modelo físico y el script SQL para ejecutarlo dentro del MySql.

- Para el desarrollo de código Html utilizar cualquier herramienta que conozca como por ejemplo Dreamweaver. Por otra parte para el desarrollo de código PHP utilizar una muy buena herramienta que viene incluida en el Linux llamada Quanta Plus, esta nos ayuda a identificar el inicio y final de las sentencias facilitándonos el trabajo de programación, así también diferenciando entre código Html y PHP.

BIBLIOGRAFÍA

- Elizabeth Castro, PERL y CGI, Segunda Edición.
- Mark Maslakowski, APRENDIENDO MySQL, Primera Edición.
- Martín Ramos Monso, PROGRAMACIÓN PHP, Primera Edición.

Sitios de Consulta en Internet:

- www.linux.org
- http://www.webestilo.com/php/php00.phtml
- http://www.ok.cl/cgi/chap0/
- http://www.jmarshall.com/easy/cgi/spanish/
- http://geneura.ugr.es/~maribel/php/
- http://www.programacion.net/tutorial/php/
- http://www.iec.csic.es/criptonomicon/cgi/
- http://www.maestrosdelweb.com/editorial/cgiintro/
- http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/
- http://www.desarrolloweb.com/index.php