



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
"ARAGÓN"**

**"PROYECTO DE UN SISTEMA DE CONTROL
DE BITÁCORAS CON ASP"**

**EN LA MODALIDAD DE "SEMINARIOS
Y CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
GUILLERMO PINEDA ARGUELLES**

**ASESORA:
ING. SILVIA VEGA MUYTOY**



MÉXICO

2007



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Doy Gracias a Dios por haberme guiado y haberme dado la fuerza y la voluntad de haber terminado este trabajo. Sin su ayuda no podría haber culminado esta etapa tan importante de mi vida.

Mamá Tus brazos siempre se abren cuando necesito un abrazo. Tu corazón sabe comprender cuándo necesito una amiga. Tus ojos sensibles se endurecen cuando necesito una lección. Tu fuerza y tu amor me han dirigido por la vida y me han dado las alas que necesitaba para volar.

Mamá Simona. Te agradezco todo el tiempo que me has dedicado y que has creído en mí eres mi inspiración en cada una de las cosas que hago y un ejemplo muy grande a seguir.

A mis hermanos, Juan Manuel y Arturo, Les agradezco con todo mi corazón el tiempo que estuvieron conmigo y que siempre hayan creído en mí ya que sin su ayuda nunca hubiera culminado esta etapa de la cual ustedes son los participantes más importantes.

Papá. Te agradezco que me hayas enseñado el otro lado de la vida, el de la humildad con las personas, y de lo dura que es la vida en ocasiones y de que la meta mas lejana siempre se puede alcanzar, aunque no lo creas eres la inspiración más grande, de que todo lo que se quiere se puede, te agradezco por ser así conmigo ya que sin ello no hubiera luchado por alcanzar este objetivo.

A mis abuelos Herlinda+ y Memo. Les agradezco por enseñarme los valores de la vida en especial a mi abuela Herlinda+, ya que fuiste mi inspiración mas importante de haber tomado este rumbo en mi vida, donde quiera que estes abuelita sabes que siempre estarás en mi corazón.

A mi abuelo Manuel. Gracias por haber sido mi inspiración en este arte, ya que sin esta inspiración que cosechaste en mí, no hubiera sido lo que ahora soy.

Irma. Te agradezco todos los momentos felices que hemos pasado juntos sino fuera por ti este mundo en el que vivo no sería el mismo siempre te lo he dicho eres el tesoro mas grande que me dio la vida, siempre te amare.

A toda mi Familia y Amigos. Les agradezco cada uno de los momentos que he vivido con ustedes, cada uno de ellos es un granito de felicidad que vive en mi alma gracias por haber estado ahí.

A la UNAM. Gracias por haberme dado la oportunidad de estar en tus aulas y de haber sido la fabrica de cada uno de mis sueños.

ÍNDICE

INTRODUCCIÓN.....	5
CAPITULO 1 INFORME DEL DIPLOMADO DE SOFTWARE LIBRE EN LINUX	
1.1 DEFINICION	8
1.2 HISTORIA	8
1.3 CARACTERÍSTICAS.....	9
1.4 PLATAFORMÁS Y DISTRIBUCIONES	11
1.5 REQUERIMIENTOS DE INSTALACIÓN.....	12
1.6 SISTEMA DE ARCHIVOS.	13
1.7 CONCEPTOS BÁSICOS.....	14
1.8 INSTALACIÓN DE LINUX	21
1.9 ADMINISTRACION DEL SISTEMA LINUX.....	25
1.10 ¿QUÉ ES UN SERVIDOR WEB?.....	30
1.11 ¿PARA QUÉ SIRVE LA SEGURIDAD EN COMPUTO?.....	39
1.12 ¿QUÉ ES UN LENGUAJE DE PROGRAMACIÓN?.....	50
CAPITULO 2 INFORME DE LOS CURSOS DE PROGRAMACION EN JAVA Y ASP.NET	
2.1 LENGUAJE DE PROGRAMACIÓN JAVA.....	79
2.2 ASP.NET (VISUAL BÁSIC).....	93
CAPITULO 3 INFORME DEL PROYECTO DE UN SISTEMA DE CONTROL DE BITACORAS	
3.1 PLANTEANDO LA NECESIDAD DEL SISTEMA	108
3.2 OBJETIVOS DEL SISTEMA	109
3.3 REQUERIMIENTOS DEL SISTEMA	109
3.4 SOLUCION DEL SISTEMA	110
3.5 HERRAMIENTAS DEL SISTEMA	111
3.6 INTEGRANDO HERRAMIENTAS.....	111
3.7 DESARROLLO DEL SISTEMA.....	112
3.8 PRUEBAS DE RENDIMIENTO.....	129
CONCLUSIONES.....	129
BIBLIOGRAFIA.....	132
GLOSARIO.....	134
ANEXO I.....	140
ANEXO II	146

Introducción

El desarrollo de sistemas cada vez en un método más complejo de desarrollo, a medida que se incrementan los requerimientos de ellos, su robustez y complejidad se van tornando cada vez de manera difícil para el entorno de los desarrolladores, que en la actualidad se tienen que ir enfrentando al desarrollo de sistemas de todo tipo de mayor grado de complejidad.

En la actualidad el mercado demanda desarrolladores de alto grado de competencia éstas empresas como por ejemplo, Microsoft, Sun Microsystems y Oracle que ofrecen herramientas de desarrollo bastante complejas, de las cuales nosotros como desarrolladores no podemos tener acceso por el motivo de que ellas proveen sus diferentes sistemas de desarrollo a altos costos por el licenciamiento de cada una de las herramientas desarrolladas por ellas.

En la actualidad el mundo del desarrollo se encuentra en una encrucijada en la cual la persona que desea desarrollar sistemas no se encuentra tan desamparado, éstas herramientas son las que se encuentran amparadas de los derechos de GPL, (General Public License), las herramientas que cuentan con ésta licencia permiten modificar y distribuir sin ningún costo el Software diverso, como son herramientas para desarrollo en lenguajes de alto nivel, manejadores de bases de datos, servidores Web, sistemas operativos etc. Éste tipo de software no por ser libre ofrece menos beneficios que el software propietario. Esto a desencadenado una serie de acontecimientos que a recientes fechas empresas líderes en el desarrollo como, Oracle colaboran con el movimiento del software libre de ésta manera se pone en la balanza si el desarrollo de sistemas utilizando herramientas de software libre es viable, y así dar una vuelta de 180 grados con respecto al panorama actual del desarrollo de sistemas.

La aplicación que generaremos con éstas herramientas es una aplicación de consulta de Bitácoras de Pasajeros y los vuelos que abordó cada uno de ellos, ésta herramienta es indispensable para la operación cotidiana en especial de AEROMEXICO, ya que ésta in-

formación sirve para determinar si un usuario tomo un vuelo y la atención que se le brindo al usuario desde que inicio su proceso de reservación hasta que tomo el vuelo en el aeropuerto que se solcito.

En éste caso se utilizara la herramienta ASP.NET para generar la interfaz grafica de la aplicación y unos procesos de Java para hacer la carga a la base de datos.

Por medio de ésta herramienta la visualización de la información será mucho más sencilla y servirá para tener una mejor atención de los pasajeros que toman sus vuelos y tener un seguimiento de cada uno de ellos y saber exactamente las deficiencias o los buenos comentarios de los pasajeros a ésta Aerolínea.

CAPITULO 1

INFORME DEL DIPLOMADO DE SOFTWARE LIBRE EN LINUX

1.1 DEFINICION

Linux se puede definir de una manera muy sencilla.

“Sistema Operativo de distribución publica basado en su antecesor UNIX para computadoras personales, servidores y estaciones de trabajo”.

1.2 HISTORIA

Durante la década de los 90 un estudiante de la universidad de Helsinki llamado Linus Torvalds, a los 23 años comenzó a desarrollar como hobby un proyecto basado en una de las diferentes versiones del sistema operativo UNIX ésta versión se llamaba MINIX desarrollada por Andrew Tanenbaum. Esto fue desarrollado en un sistema basado con un procesador 80386, Torvalds lo que propuso fue desarrollar un sistema operativo que fuera tan versátil como UNIX empezando por éste pequeño sistema del cual quería aprovechar su arquitectura de 32 bits en la que se desarrollaba, una de las ventajas de todos los linux desde su nacimiento es la conmutación de tareas y la eliminación de barreras de memoria.

Linus empezó escribiendo el núcleo de dicho proyecto en código ensamblador, después empezó a desarrollarlo en lenguaje C, una de las ventajas de dicho lenguaje es que combina la potencialidad de un lenguaje de alto nivel y la manipulación de sectores de memoria tal cual se pudieran manipular como si estuviéramos utilizando lenguaje ensamblador. Durante el año de 1991, en el mes de octubre, anuncio en un foro de Internet la primera versión oficial de su sistema operativo el cual llamo Linux, la primera versión en liberarse fue la 0.02, ésta fue capaz de ejecutar un Shell de Unix que en éste caso se le llamo fue BASH y su primer compilador el cual fue gcc éste podía compilar código escrito en lenguaje C. después de que Torvalds divulgara dicha carta en el foro de Internet, proponiendo a los visitantes de dicho foro que utilizaran su sistema, Linux creció de una manera desorbitada creando una nueva opción para aquellos desarrolladores que no cuentan con el

poder económico para poder llevar sus proyectos de desarrollo a cabo, ya que no contaban a con el poder económico para lograrlo. Muchos de los componentes de Linux, como los drivers, protocolos y seis salieron de otro sistema operativo llamado FreeBSD en cual fue desarrollado en la Universidad de Berkley, en la actualidad el sistema operativo ésta en continuo mantenimiento con aportaciones que desarrolladores propagan por la red, en la actualidad el sistema operativo de Torvalds sólo corresponde al 10 % por ciento de lo que actualmente son las diferentes versiones de dicho sistema operativo.

1.3 CARACTERÍSTICAS.

Como sistema operativo, Linux es un sistema muy eficiente, esto con lleva a que tiene un excelente diseño.

Características principales:

Multitarea: Linux tiene la capacidad de atender diferentes usuarios que utilizan un mismo recurso de sistema.

Multiusuario: Linux pose la característica de atender a muchos usuarios repartiendo a cada uno de ellos tiempo de procesamiento para que cada uno de ellos ejecute un proceso específicos.

Multiplataforma: Cuenta con la característica de operar en diferentes plataformas con son servidores, MainFrames, SuperComputadoras etc.

Multiprocesamiento: Cuenta con un sistema que otorga tiempos a cada uno de los programas que se encuentran corriendo y así poder atender a cada uno de ellos.

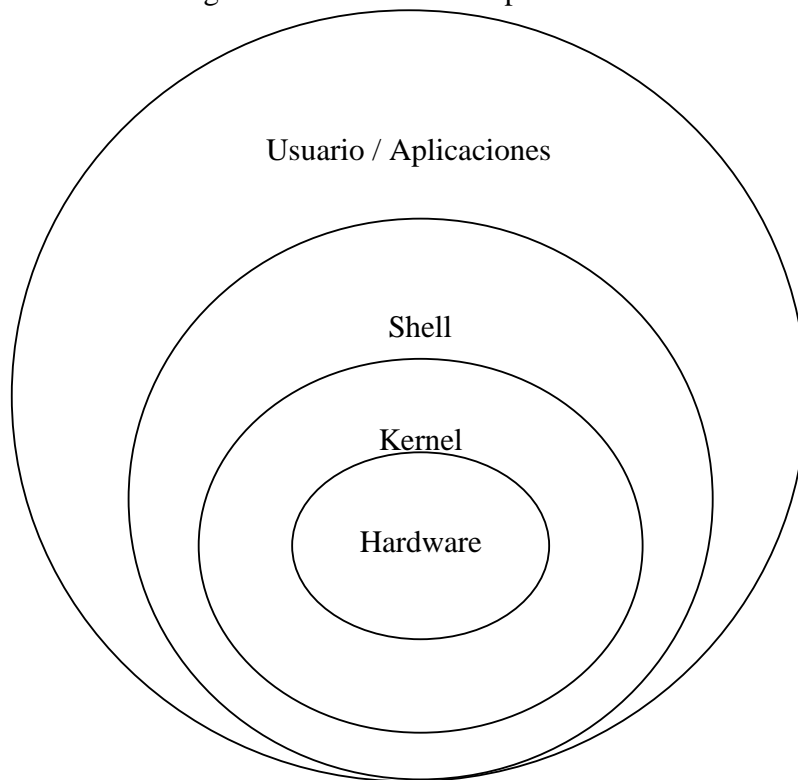
En las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer el resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el

uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente.

Todas las herramientas y programas que son utilizados en Linux contienen el código fuente con el cual fueron diseñados que en este caso es, Lenguaje C; se puede hacer comunicación hasta con 64 consolas virtuales; tiene un sistema de archivos avanzado; soporta redes tanto de TCP/IP como de otros protocolos.

Linux tiene una estructura en capas las cuales son (ver figura1.1)

Figura 1.1 Estructura en capas de Linux



La capa de hardware es donde se encuentran todos los dispositivos físicos de nuestro sistema como son: tarjeta madre, tarjeta de red, tarjeta de video, unidades ópticas, teclado, etc.

La capa de kernel es donde reside el núcleo del sistema y la interfaz que integra la comunicación entre el hardware y el shell de sistema que. Incluye entre sus funciones las operaciones más importantes de gestión del sistema operativo, como memoria, mantenimiento del sistema de archivos, asignación de tiempo del CPU a cada una de las tareas, el control del acceso mediante claves, etc.

La capa de Shell es donde se encuentra el intérprete de comandos. Éste intérprete como su nombre lo indica “interpreta” los comandos que el usuario va introduciendo al sistema. Además es utilizado en muchas ocasiones como un pequeño lenguaje de programación con las diferentes versiones de shell que existen en la actualidad como lo son BASH, BOURNE SHELL, CSHELL etc.

La capa de Usuario/ Aplicaciones alberga los programas de aplicación del usuario. En ésta capa se encuentra el usuario que se comunica con el shell, el cual a su vez se comunica con el kernel para pasarle instrucciones que se tienen que realizar y el kernel finalmente es el mediador con el hardware.

1.4 PLATAFORMÁS Y DISTRIBUCIONES

Linux fue desarrollado para el procesador i386 actualmente soporta procesadores i486 y i586 así como sus clones como lo son AMD y Cyrix. También soporta máquinas basadas en SPARC, UltraSPARC, Alpha, etc.

En la actualidad existe una gran cantidad de distribuciones de Linux: una distribución es un paquete que conforma el kernel junto con otros programas creados por diferentes compañías para formar cada uno de los sistemas operativos que presta su nombre.

Distribuciones	
<ul style="list-style-type: none"> ✓ Fedora (antes Red Hat) ✓ Mandrila (antes Mandrake) ✓ Suse ✓ Debian ✓ Knoppix ✓ Slackware ✓ Fedora (antes Red Hat) ✓ Mandrila (antes Mandrake) 	<ul style="list-style-type: none"> ✓ Suse ✓ Debian ✓ Knoppix ✓ Slackware ✓ Ubuntu ✓ Gentoo ✓ Slackware

De las siguientes distribuciones la que se manejo durante el periodo del diplomado fue la versión de Slackware, por la estabilidad del kernel, seguridad.

1.5 REQUERIMIENTOS DE INSTALACIÓN

Los requerimientos mínimos de instalación son los siguientes.

Procesador i586.

Ram 32MB.

Espacio en Disco 1GB.

Unidades extraíbles necesarias CD-ROM y/o floppy.

Si se cuenta con una unidad BOOTEABLE de CD-ROM no tendrá necesidad de utilizar la unidad de floppy para hacer la instalación del sistema. Pero si no se cuenta con la unidad de CD-ROM se necesitara una unidad de floppy para hacer una instalación vía red.

El requerimiento de espacio en disco es engañoso en el momento de la instalación. La instalación sólo requiriendo 1GB es la mínima instalación ya que para ésta instalación con todas las herramientas con las que cuenta el sistema se necesitaran alrededor de 3GB de espacio en disco duro Además de espacio adicional para archivos personales.

Para la conexión de una red se necesita una interfaz de red (tarjeta de red) o Wi-fi. Con respecto a los es necesario contar con un monitor, ratón y teclado.

1.6 SISTEMA DE ARCHIVOS.

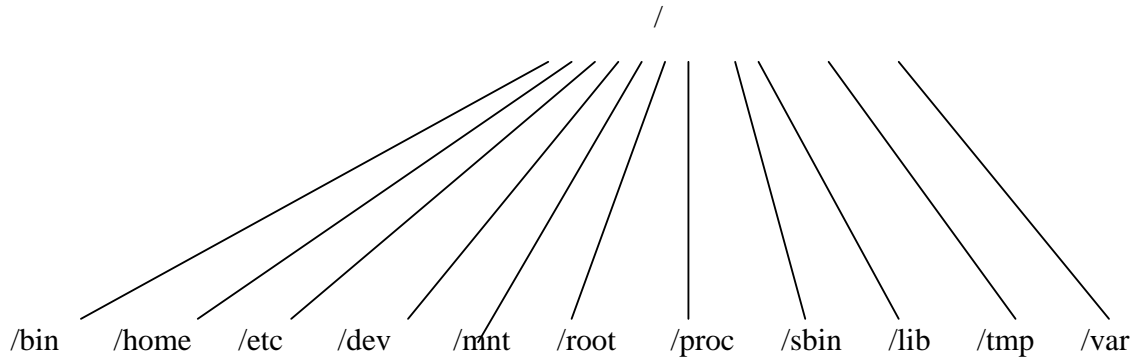
Podemos definir de forma genérica el término archivo como un conjunto de datos con un nombre asociado. Los archivos suelen residir en dispositivos de almacenamiento secundario, tales como cintas, discos duros etc.

Algunos archivos o sistemas operativos imponen a todos sus archivos una estructura determinada bien definida. En LINUX un archivo no es más que una secuencia de bytes (8bits. Algunos programas esperan encontrar estructuras de diferentes niveles, el núcleo o kernel no impone ninguna estructura sobre los archivos.

Los archivos están organizados en lo que se conoce como directorios, el sistema de archivos tiene una estructura de árbol invertido en el cual se agrupan en directorios. En el todos los archivos y directorios dependen de un único directorio denominado directorio raíz o root, el cual se representa con el símbolo de “/”, Todos deben depender del directorio raíz, (En caso de que tengamos más de un dispositivo de almacenamiento secundario), y el usuario tratara cada uno de los discos como subdirectorio que depende, directa o indirectamente de la raíz. A ésta operación se le conoce con el nombre de montaje de un subsistema de archivos.

La estructura del sistema de archivos de Linux depende directamente de la versión que éste utilizando de dicho sistema operativo a continuación veremos el sistema de archivos de la versión de Slackware de Linux versión 10.1

Figura 1.2 Árbol de Archivos de Linux



1.7 CONCEPTOS BÁSICOS.

Inicio y término de sesión

Para iniciar sesión en el equipo ya sea en entorno gráfico o desde línea de comandos, el sistema pide siempre ya sea en modo gráfico o en modo shell¹ se pide un login y password .

username: morpheus

password: (en ésta opción no aparecen caracteres durante el tecleo del password).

Para terminar sesión desde shell existen una serie de comandos para

Exit y logout

Creación de Cuentas

Lo que permite a Linux y a todos los sistemas basados en Unix, ser seguros, es que primeramente para acceder al sistema es necesario contar con una cuenta, es decir, un login y un password. Para la creación de una cuenta de usuario se realiza de la siguiente manera.

#adduser

Éste comando despliega una serie de procedimientos que el usuario debe acceder para dar de alta un usuario

Consolas Virtuales

Debido a que Linux es un sistema multiusuario, cuando se necesitan hacer algunas tareas de más, el usuario puede logearse en más de una ocasión al sistema, esto se logra a través de la serie de teclas CTRL + ALT + F(1...6), esto se puede cambiar entre consolas virtuales, Slackware proporciona 6 consolas mínimas virtuales en modo Shell, el resto de la teclas de función están reservadas para sesiones de X (sistemas de ventanas similar al de Windows). Cada sesión de entorno gráfico usa su propia consola virtual comenzando con la tecla de la función (Alt + F(7.....12)), hacia arriba.

Operaciones con archivos y directorios

En Linux al igual que Unix todo se maneja como un archivo; un directorio es un archivo que contiene archivos inclusive otros directorios.

1. Venta similar a la del sistema operativo de MS-DOS con una línea de comandos propios del sistema operativo Linux.

-Muestra contenido de archivos, `cat` `more` y `less`

Estos comandos nos permiten ver el contenido de uno o varios archivos cada uno de ellos contienen pequeñas diferencias. El comando `cat` muestra el contenido de uno o varios archivos pero de forma continua sin detenerse durante el despliegue del contenido del archivo. `More` éste comando su función es similar a la de `cat` la diferencia es que durante el despliegue del contenido del archivo puede hacer éste una serie de pausas que permiten y éste es controlado con la flecha de dirección hacia abajo y con barra espaciadora. `Less` éste comando es una versión mejorada de `less` ya que por medio de las flechas de dirección se puede controlar la visualización del comando ésta disponible sólo en Linux.

-Ayuda.

Linux cuenta con ayuda su ayuda en cada uno de los comandos que se ejecutan. Éste ayuda es el manual completo de cada uno de los comandos ya antes mencionados el cual cuenta con 8 secciones.

1. Herramientas de usuario
2. Llamadas al sistema
3. Subrutinas y funciones de bibliotecas.
4. Información de controladores de dispositivos.
5. Archivos de Configuración
6. Juegos
7. Paquetes
8. Administración de Paquetes.

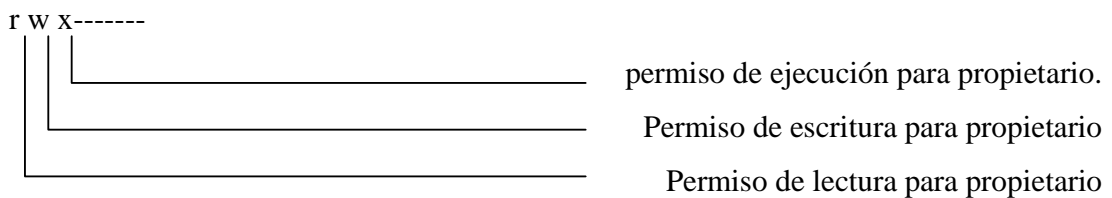
-Redireccionamiento.

En linux la entrada y salida de datos están definidas. De manera que la entrada estándar de datos es el teclado y su salida estándar es el monitor. Pero éstas formas de introducir o sacar datos del sistema se puede cambiar por medio del redireccionamiento.

Permisos de Archivos

En Linux cada aspecto del sistema es multiusuario incluyendo el sistema de archivos. El sistema almacena información como quien es el propietario de un archivo y quien puede leerlo. El sistema almacena información como quien es el propietario de dicho archivo y quien tiene permisos de lectura de dicho archivo. Se almacena información del usuario para cada archivo y directorio. Esto incluye que usuario y grupo es propietario de un archivo en particular.

Figura 1.3 Estructura de permisos de Linux



Al inicio del despliegue del directorio en el que nos encontramos con ésta serie de caracteres que nos representara el tipo de archivo y los permisos que tiene cada grupo de usuarios como lo podemos ver en la figura anterior (figura 1.3).

-Como Cambiar los permisos., Comando `chmod`.

Para poder cambiar los permisos a un archivo existe el comando `chmod`. Éste comando trabaja de dos formas una de ellas utilizando la siguiente notación:

Permisos	
<i>u</i> usuario.	r lectura
<i>g</i> grupo.	w escritura
<i>o</i> otros	x ejecución
+ agregar	
-quitar	

La otra forma es la de trabajar con el comando `chmod` es con notación octal.

Permiso	Valor Octal
Leer	4
Escribir	2
Ejecutar	1

De manera que si quiere darle al archivo los siguientes permisos: lee y ejecutar para el usuario, grupo y otros, el comando que debemos ejecutar es:

`#chmod 555 nuevo4` (en éste caso el primer numero es para el usuario el segundo es para el grupo y el tercer numero es para los otros).

Ligas

-Ligas duras y ligas suaves

Las ligas son apuntadores entre archivos. Con las ligas se pueden tener archivos en muchos lugares del sistema y ser accedidos por diferentes nombres. Existen dos tipos de ligas: duras y suaves.

Las ligas duras son nombres para un archivo en particular, sólo pueden existir en un sistema de archivos simple y sólo son removidas cuando el nombre real es removido del sistema.

La liga suave, también llamada liga simbólica, puede apuntar a un archivo fuera de su sistema de archivos. De hecho es un pequeño archivo que contiene la información que necesita. Se pueden agregar o remover ligas suaves sin afectar al archivo actual. Y por el hecho de que una liga simbólica es un archivo que contiene su propia información, puede apuntar también a un directorio.

Para la creación de una liga sólo es necesario utilizar el comando `ln` sin ninguna opción, sólo el nombre del archivo al que se va a hacer la liga y el nombre de la liga.

Procesos

Cada programa que se encuentra ejecutándose en el sistema es llamado proceso. Estos procesos son desde el sistema X hasta programas del sistema que son inicializados desde que la computadora arranca. Cada proceso corre como propiedad de un usuario en particular. Procesos que son inicializados en el tiempo de arranque usualmente corren como propiedad del usuario o nobody. Procesos que uno como usuario inicializa correrán como propios.

Cada usuario tiene control sobre los procesos que inicializa. Adicionalmente, root tiene control sobre todos los procesos del sistema incluyendo aquellos que son inicializados por otros usuarios. Los procesos pueden ser controlados y monitoreados a través de varios programas como con algunos comandos.

En ocasiones cuando un programa empieza a consumir recursos del sistema, se necesita un procedimiento que pueda solucionar dicho problema. El programa para realizar éste tipo de administración se llama **kill** (y se ejecuta con el comando del mismo nombre), puede ser usado para manipular procesos de muchas maneras. El más obvio para poder matar un proceso de muchas maneras. El uso más obvio de Kill es para terminar un proceso de manera manual. Éste comando se utiliza si un programa durante su ejecución se sale de control y por tal motivo empieza a consumir recursos del sistema o simplemente si el usuario determina parar un programa.

1.8 INSTALACIÓN DE LINUX

Antes de empezar la INSTALACIÓN de Slackware 10.1 se realizan algunos de los pasos previos. Lo primero que debemos realizar es particionar nuestro disco duro con la utilidad fdisk que viene en el disco de INSTALACIÓN de Slackware así deben de asignarse las particiones dentro de nuestro disco duro:

Requerimientos de disco duro

swap.....	256mb
/.....	3GB
/home.....	4GB
/var/spool/mail.....	2GB
/usr/local.....	5GB

La partición swap se utilice como espacio de intercambio entre la memoria RAM y el disco duro. Como una recomendación se debe crear una partición tomando en cuenta la memoria RAM total, la partición SWAP debe ser el doble de la memoria RAM, pero por otro lado no debe de ser mayor de 256MB.

/ (Partición raíz). Aquí es donde reside el sistema operativo, donde se instalan todos los componentes del sistema; desde archivos propios, como utilerías, programas o más.

La partición home se crea para que los usuarios tengan disponible éste espacio en disco para sus archivos. Esto permite que los usuarios no excedan el espacio que éste tiene asignado. El tamaño de ésta partición depende de cuantos usuarios va a tener el sistema, en

nuestro caso nosotros agregaremos 4GB por la razón de que necesita éste espacio para la alta de usuarios a nuestro sistema y los diferentes archivos que va a guardar cada uno ellos en sus diferentes cuentas de usuarios.

Para el caso de la partición `/var/spool/mail`, ésta se crea para almacenar los correos electrónicos de los usuarios; de igual manera que la partición `/home`, ésta partición depende de los usuarios que vaya a tener el sistema.

`/usr/local`, ésta partición generalmente se utiliza para que ahí residan todos los programas que se instalaran, por ésta razón el tamaño es más grande que la de las otras particiones inclusive que las del sistema.

Cada disco y dispositivo en el sistema es considerado como un gran sistema de archivos; algunas particiones, CD-ROM y floppies, son colocados en el mismo árbol. Para adjuntar éstas unidades el sistema de archivos y poder accederlos se usan los comandos `mount` y `umount`.

Después de haber hecho el proceso de partición de disco procederemos a la INSTALACIÓN del sistema. Éste se hace ejecutando el siguiente comando

#setup

Éste comando comienza con la INSTALACIÓN del sistema operativo.

Al ejecutar el comando se mostrara un menú con diferentes opciones entre las cuales están. Ayuda para la INSTALACIÓN, remapeo, del teclado, establecer una partición swap, seleccionar categorías de software para instalar, entre otras. En éste momento se establece la partición swap.

Después de haber establecido la partición swap, el programa pregunta por la partición en donde se instalara el sistema operativo. De igual manera que con la partición se detectan

automáticamente las particiones en el disco y el sistema pregunta si se desea instalar en dicha partición se procede a seleccionar la indicada.

Una vez que se termina el formateo de la partición /, el programa pregunta por los paquetes software que se desea instalar. Desde el sistema base hasta entornos para el sistema X.

Ya que se ha seleccionado los paquetes del software, aparece una nueva pantalla donde el programa pregunta por el modo en el que se desea que vaya la informado acerca de lo que se está instalando. En este caso se selecciona el modo completo (full), que requiere un poco más de 3 GB de espacio en disco.

Después de esto el programa de INSTALACIÓN pregunta si se desea instalar un kernel, si se prefiere instalarlo desde el disco de Slackware escogiendo algún kernel en especial o si se prefiere saltar este paso y dejar algún kernel que haya sido instalado.

El siguiente paso es la INSTALACIÓN del cargador de Linux (LILO). Aparece una pantalla con tres opciones.

La primera opción es la forma simple, es decir, trata de instalar LILO de forma automática, aun si encuentra otro sistema operativo como Windows. La forma experta ofrece el control sobre el proceso de INSTALACIÓN. En este caso forma se utiliza. Y finalmente la opción de saltar este paso para poder después instalar LILO con el comando liliconfig.

Por ultimo en la INSTALACIÓN del LILO, el programa pregunta por el destino donde se va a instalar el cargador. Como siempre este presenta varias opciones, la opción a seleccionar es la de MBR (Master boot Record).

Una vez que se instalo el cargador de Linux, continua la configuración del Mouse y del programa gpm que permite cortar y pegar texto en consolas virtuales usando el Mouse. A

continuación de esto el programa de INSTALACIÓN pregunta si se desea configurar la red. A lo cual se responde que sí:

Los primero parámetros que nuestro menú nos pide es el nombre del host, es decir el nombre del equipo. Además del dominio al cual va a pertenecer éste host (en éste caso al host de la UNAM). Después de esto pide la dirección IP de la tarjeta de red del host; ya sé que la dirección de loopback. En éste caso se escoge la asignación de IP por dhcp. Al escoger éste tipo de IP el programa de INSTALACIÓN pide el nombre del dhcp; si se conoce el nombre del dhcp se proporciona sino tendremos que adjudicarla de manera manual nuestra dirección IP. Con esto ya se encuentra configurada la red; se muestra una pantalla de confirmación de los parámetros que se configuraron anteriormente, si todo es se acepta la configuración con él la opción de YES.

A continuación se deben seleccionar los servicios que se requiere que se inicien al arrancar el sistema; servicios como el servidor de archivos y de impresión para redes Macintosh, Netalk, Appletalk, el servidor de DNS BIND, el servidor de impresión CUPS, el servidor de Web Apache, entre otros.

Después de los servicios anteriores, se debe de seleccionar el administrador de ventanas que va a utilizar nuestro sistema. En éste caso seleccionaremos el ambiente KDE, ya que su ambiente es más amigable y estable que los otros.

Por ultimo aparece un mensaje de advertencia que indica que no hay ningún password de usuario root detectada. En éste punto se debe de establecer un password para el usuario root. Se debe de saber que por motivos de seguridad no se debe de apuntar a ningún lado el password para root. Se debe de saber que por motivos de seguridad no debemos de proporcionar el password a ninguna persona que no sea el administrador del sistema y Además éste password debe de ser un password fuerte (fuerte significa una combinación entre números, caracteres y caracteres especiales). Una vez establecido el password del usuario root, aparece un mensaje indicando que la INSTALACIÓN se ha completado. Se debe de reiniciar el equipo para que quede por completo instalado el sistema operativo.

1.9 ADMINISTRACION DEL SISTEMA LINUX

Perfil de actividades de un administrador

El administrador del sistema debe de planear las actividades que se van a desarrollar.

La “primera regla” del un administrador de sistemas consiste en proporcionar y mantener acceso a los recursos del sistema. Al fin y al cabo todas nuestras actividades profesionales parten o se relacionan con éste objetivo. Independientemente de la plataforma informática de que se trate, todos los sistemas operativos proporcionan mecanismos para manipular recursos. Entre estos recursos se hallan los archivos, las aplicaciones, los periféricos etc.

El mantenimiento del acceso de recursos hacia un sistema trasciende en el mantenimiento del propio sistema. Si éste falla o se vuelve inaccesible por cualquier causa, el administrador de sistemas deberá restaurar prontamente el funcionamiento normal del mismo. La mayoría de la gente piensa que haciendo copias de seguridad se soluciona dicho problema. Sin embargo, nuestro planteamiento girara más bien en torno a la planificación de la recuperación, lo cual debe de entrenar hacer copias seguridad, pero también valoración de riesgos.

Para mantener un control eficaz del sistema, el Administrador debe también de establecer políticas de uso, así como políticas de administración. Dentro de las políticas, debe de establecer punto sobre, como desde donde y en que horarios pueden los usuarios ingresar al sistema, por ejemplo. Dentro de las políticas de administración se deben de establecer puntos sobre la privacidad en la información de cada usuario.

Por ultimo para tener una administración completa, es importante dentro de las actividades del Administrador del sistema mantener un canal de comunicación con los usuarios. Puede ser vía correo, o mandándolos mensajes directamente a cada usuario. Esto con el fin de darles avisos acerca de servicios, o cuando, por ejemplo, algún usuario incurre en algo no permitido.

Conceptos básicos de administración

-Uso de Lilo

Lilo (Cargador de Linux) es el arrancador del sistema operativo, es completamente configurable y puede ser fácilmente utilizado para arrancar otros sistemas operativos. Slackware viene con una utilidad configurable a través de menús llamada liloconfig. Éste programa es el primero que se ejecuta durante el proceso de arranque, pero se puede invocar después tecleando en la línea de comandos liloconfig.

-Comandos shutdown, halt y reboot.

Es sistema siempre debe de ser apagado apropiadamente. Si se apaga con el swich de energía se pueden ocasionar graves danos al sistema de archivos. Mientras el sistema ésta encendido hay archivos que están siendo usados aun no sé ésta haciendo nada.

Una de las formas apropiadas para apagar el sistema es con el programa shutdown (comando del mismo nombre). Éste comando sirve para reiniciar o apagar el sistema en un tiempo dado y puede desplegar un mensaje a todos los usuarios que se encuentren trabajando en el sistema mencionándoles que el sistema ésta en el proceso de apagar.

Manejo de usuarios y grupos

-Añadir usuarios

Como ya se menciona anteriormente, para agregar un usuario el sistema se utiliza el comando: **#adduser**

Cuando se requiere borrar un usuario se utiliza el comando: **#userdel <nombre_usuario>**

-Comandos chsh y chfn

Estos comandos sirven para cosas específicas. El comando `chsh` sirve para cambiar el shell del usuario. Cuando se ejecuta éste comando el sistema pide la password del usuario y después le pide que ingrese la ruta completa del nuevo shell.

El comando `chfn` permite cambiar la información opcional introducida durante el proceso de creación de la cuenta de usuario, como es el nombre completo, números de teléfono y número de departamento. Como el usual `root` puede cambiar ésta información de cualquier usuario con sólo indicar como argumento el nombre de usuario.

Configuración de la interfaz gráfica

Al instalar el sistema operativo, por defecto, el sistema inicia en modo texto, éste modo de arranque es especialmente útil para la administración de Linux, ya que no necesita un entorno gráfico para ello. De cualquier manera si se quiere levantar un entorno gráfico, existen un comando que lo hace llamado `startx`. Éste comando funciona siempre y cuando

la configuración de la tarjeta de video sea la adecuada; Si no se conoce a detalle el hardware que se encuentra instalado en el sistema, es difícil la configuración a punto de nuestro sistema.

Un método que se utiliza para lograr levantar la interfaz grafica fue sobrescribiendo el archivo `xorg.conf` con el archivo `xorg.conf-déb`. Aquí es donde se guarda la configuración de la nuestra interfaz grafica.

Si se desea que el sistema arranque siempre en modo gráfico, existen varias manera de configurar el sistema para ello. Una de ellas es ir al archivo `/etc/rc.d/rc.local` y agregar la siguiente línea: `/opt/kde/bin/kdm`

Otra forma es ir al archivo `inittab` y cambiar el nivel de arranque (run-level) al nivel que nosotros deseamos que para levantar la interfaz grafica es por medio del numero 4.

Administración de Paquetes

Un paquete es un conjunto de programas relacionados listos para instalarse. Cuando se baja de Internet un archivo de código fuente, uno tiene que configurarlo, compilarlo e instalarlo. Otra buena característica de utilizar los paquetes, es que son fáciles de remover y de actualizar. Linux Slackware viene con programas de administración de paquetes. Algunos de ellos son `pkgtool` e `installpkg`.

Configuración de la tarjeta de Red

Para configurar la tarjeta de red se cuenta con el comando `ifconfig` es una utilidad que permite obtener y configurar las interfaces de red del equipo. Si no se proporcionan argumentos, `ifconfig` muestra el estado de la interfaces de red del equipo. Si no se proporcionan argumentos, `ifconfig` muestra el estado de las interfaces, incluso aquellas que se encuentran desactivadas. Para configurar una interfaz se debe de utilizar la sintaxis:

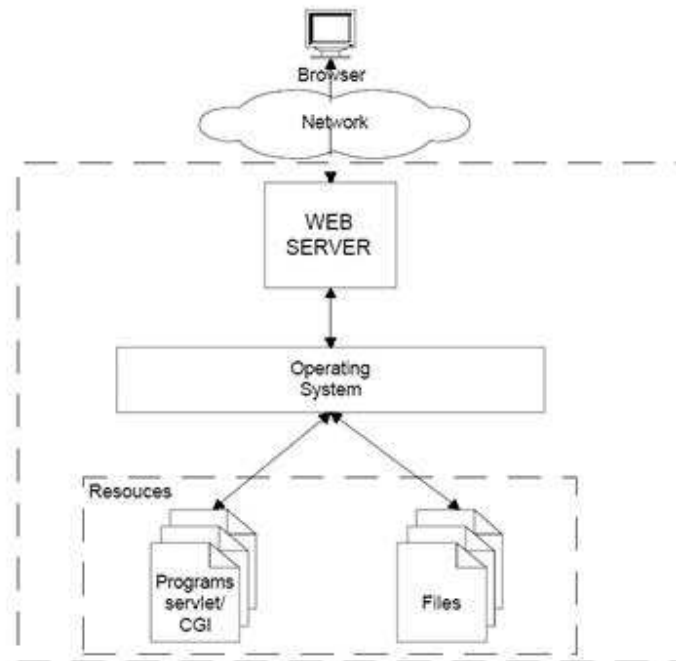
```
-ifconfig (interfaz) (familia) (dir_ip) netmásk (máscara) broadcast (dir_broadcast) up.
```

En interfaz debe proporcionarse el nombre de la interfaz de red que se desea configurar. Generalmente el nombre de interfaz se forma a partir de un nombre de manejador seguido de un número de unidad. El nombre del manejador para redes es `eth` y las unidades comienzan a enumerarse a partir de 0 hasta el número de interfaces existentes del mismo tipo menos uno (`eth0`, `eth1`, etc).

1.10 ¿QUÉ ES UN SERVIDOR WEB?

Un servidor Web es un software que se encarga de manejar peticiones HTTP. Es decir ofrece comunicación a través de dicho protocolo. Cuando el servidor Web recibe una petición HTTP, enviando de regreso una pagina HTML. Para procesar una petición, el servidor Web puede responder con una pagina estática o imagen, enviar un redireccionamiento o delegar la generación dinámica de la respuesta a algún otro programa como un script o un CGI, un JSP (Java Server Pages), un servlet, un ASP(Active Server Pages), un server-side JavaScript o alguna otra tecnología del lado del servidor. Cualquiera que sea su propósito dichos programas del lado del servidor generan una respuesta, generalmente HTML, para ser vista en un navegador Web como se puede ver en la figura 1.4.

Figura 1.4 Diagrama de Operación de un servidor Web



Servidores Web más populares

Existe un gran número de empresas y organizaciones que ofrecen Servidores Web en el mercado, cada uno de ellos con distintas características que cubren necesidades propias del usuario. Los más populares en la actualidad son:

- IIS-Microsoft
- Sun Java Web Server-Sun Microsystems
- Roxen Web Server- Open Source
- Public Domain HTTP Daemon – NCSA
- Zeus Web Server – Zeus
- Apache Web Server – Open Source

Para inclinarse por el empleo de uno de ellos se recomienda evaluar los siguientes criterios de selección.

- Función del servidor Web. Es importante establecer que función va a desempeñar el servidor con relación a la estrategia de negocios. Es decir, pudiera implantarse para que los empleados de una compañía accedan a diversas páginas, para consultar determinada información de la base de datos. O pudiera implantarse que para lo clientes de la misma compañía, a través de Internet, conozcan los productos y servicios que éste ofrece.
- Nivel de expertise que tienen los administradores. Es de vital importancia considerar que tan experto es el personal que administrara el servidor Web, en función de sus conocimientos y experiencia.

- Plataforma Disponible. Ésta situación aplica en la caso de IIS de Microsoft ya que éste sólo funciona con el sistema operativo Windows al que pertenece el sistema.
- Numero de conexiones concurrentes. Se considera la estimación del numero de conexiones concurrentes que atenderá nuestro servidor Web.
- Costo computacional por transacción. Aunado al criterio anterior se debe de considerar el costo computacional (tiempo de procesador, memoria, envío y recepción de datos) por cada transacción realizada.
- Proyección del crecimiento esperado. Para saber si el servidor cubrirá las expectativas de crecimiento en el futuro, se debe de hacer una proyección de dicho crecimiento ya que éste factor es muy importante para determinar el empleo de un servidor u otro.
- Soporte para la tecnología utilizada para el desarrollo. Ya que no todos los servidores Web tiene compatibilidad con todas las tecnologías de desarrollo, es importante considerar que el servidor que se vaya a emplear sea compatible con las tecnologías que se utilizaran para el desarrollo.
- Análisis del retorno de la inversión. Ya que el factor económico es muy importante a la hora de implantar un servidor Web, el análisis del ROI (Return On Investment) se vuelve obligatorio.

Apache el servidor Web libre

Apache es el servidor Web más utilizado en la actualidad debido a las características que ofrece.

El proyecto del servidor Web Apache es un esfuerzo de colaboración de desarrollo en el cual el objetivo es el desarrollo de un servidor HTTP o Web que pueda ser robusto, de nivel comercial, altamente funcional y de código abierto. Éste proyecto es justamente manejado por un grupo de voluntarios que se encuentran alrededor del mundo, y que utilizan Internet y la Web para comunicarse, planear y desarrollar el servidor y su documentación relacionada. Éste proyecto es parte del Apache Software Foundation.

-Características de Apache

- Robusto, soporta un gran número de transacciones.
- Configurable para diferentes entornos de trabajo.
- Ofrece un alto nivel de seguridad.
- Disponible para una gran variedad de plataformas.
- Soporte para servicio de Proxy.
- Soporte para granjas de servidores.
- Soporte para Scripting lenguajes integrados como módulos(PHP, mod_perl).
- Incluye el código fuente del servidor.
- Soporte para accesos restringidos.
- Soporte para SSL (Secure Socket Layer).
- Es libre.

-Instalación y configuración de apache

*Instalación

Apache puede obtenerse de <http://www.apache.org> de manera gratuita. Una vez que se ha bajado el archivo, se debe descomprimir con el siguiente comando.

```
#tar zxvf httpd-xx.tar
```

Al descomprimirse se genera un directorio que contiene el código fuente de la distribución que se obtuvo. Se debe ingresar a ese directorio antes de seguir con la compilación.

El siguiente paso es configurar el árbol fuente para la plataforma particular y requerimientos individuales, éste punto se puede ver en la documentación de la aplicación.

Apache viene con un set de base de módulos incluidos por defecto. Otros módulos se habilitan usando la opción `enable-module`, donde `module` es el nombre del módulo.

También en éste punto se puede configurar apache especificándole alguna característica que se desee el momento de reiniciar o iniciar nuestro servidor Web apache.

Una vez configuradas las opciones, se compila el servidor:

```
#make
```

En éste caso de ser paciente, ya que desde una configuración base se puede tardar varios minutos en compilar, y el tiempo puede variar ampliamente dependiendo del hardware y del número de módulos que se hayan habilitado.

Por ultimo viene la INSTALACIÓN, bajo la configuración establecida con –prefix:

```
#make install
```

Para comprobar que se ha instalado correctamente el servidor se inicia el comando:

```
#./apachectl star.
```

Una vez levantado el servidor, se abre un navegador Web y se escribe en la barra de direcciones localhost o la dirección de loopback 127.0.0.1: aparecerá un mensaje de bienvenida, lo cual indica que se he instalado exitosamente Apache.

*Configuración

Apache es administrado por más de doscientas directivas, las cuales permiten que determinada funcionalidad pueda ser incluida. En Linux el administrador controla que directivas estarán disponibles de acuerdo a los módulos con los que se compile apache.

En apache la configuración se realiza mediante tres grupos de directivas:

- Global Environment. La sección Global Environmet administra las directivas generales de operación para Apache.
- Main Server La sección Main Server administra las directivas del servidor principal o estándar apache.
- Virtual Host La sección Virtual Host administra las directivas donde los procesos de Apache soportan diversas direcciones IP o diversos nombres de dominio

*Cgi´s

La tecnología CGI “Common Gateway Interface” define un modelo un modelo de programación que puede ser implementado por múltiples lenguajes.

Los CGI's son programas que siguen un estándar definido, corren en el servidor, reciben parámetros desde el cliente y su salida es enviada al navegador. Fueron la primera alternativa para generar dinamismo a un sitio Web.

Existen un par de directivas para configurar la utilización de CGI's en el servidor Apache.

`ScriptAlias <alias> <path-filesystem>`

Convierte las solicitudes vía URL a la ruta absoluta desde donde residen los programas CGI. Ésta directiva se utiliza tanto en Server-Config como en VirtualHost.

`AddHandler cgi-scripts .cgi .pl`

Permite que determinada extensión sea relacionada a un evento en particular. En el caso de los CGI's lo que se indica es que las extensiones.cgi y.pl quedan identificadas como extensiones scripts. Apache interpretara el archivo en lugar de sólo enviar el contenido al cliente.

Server-Side Includes

Ésta opción activa un filtro que permite incluir etiquetas dentro del archivo HTML, las cuales son procesadas por Apache antes de ser enviadas como HTML al cliente.

Es necesario habilitar la opción + Includes en el directorio donde se encuentran los archivos incluyen etiquetas SSI.

Las directivas configuran ésta opción son:

`AddType text/html .shtml`

Relaciona los archivos con extensión.shtml como aquellos que contienen etiquetas SSI.

```
AddOutputFilter INCLUDES .html
```

Habilita el filtro para los archivos .html.

NOTA: SSI representa un riesgo y debe ser usado con cautela, para evitar un incidente de seguridad.

Las directivas presentadas son sólo algunas con las que cuenta Apache, pero estas son las más importantes para poder configurar el servidor en forma básica.

*Control de Accesos

Apache ofrece la funcionalidad al estilo “ACL”, con lo que es factible determinar las direcciones IP y los usuarios que tendrán acceso sobre recursos específicos del servidor Web. La configuración de acceso vía nombre de dominio o vía IP puede realizarse con ayuda de las siguientes directivas.

Order: Define el orden en que las directivas Allow y Deny serán implantadas.

Allow: Define los hosts que tendrán acceso a alguna área del servidor. El acceso puede ser controlado por el nombre de host, dirección IP, rango de direcciones IP o por otras características de la petición del cliente guardadas en variables de entorno.

Deny: Define la restricción del acceso al servidor basado en nombre de host, dirección IP o variables de ambiente

order allow, deny: Permite explícitamente a los clientes definidos en allow, niega a todos los demás.

order deny, allow: Niega explícitamente a los clientes definidos en deny, permite todos los demás.

*Manejo de sitios virtuales

El término Virtual Host (Host Virtual o Sitio Virtual) se refiere a la práctica de correr más de un sitio Web (como www.miempresa2.com) en una sola máquina. Los sitios virtuales pueden ser basados en dirección IP, es decir, se tienen diferentes direcciones IP para cada sitio Web, basados en nombre, es decir, se tienen múltiples nombres corriendo en cada dirección IP. El hecho de que están corriendo en un mismo servidor físico no lo hace evidente para el usuario final.

Apache fue uno de los primeros servidores en dar soporte para sitios virtuales basados en IP y más tarde dio soporte para ambos, basados en IP y basados en nombre. Ésta última variante es nombrada también basada en host (host-based) o no IP (non-IP virtual hosts).

1.11 ¿PARA QUÉ SIRVE LA SEGURIDAD EN COMPUTO?

La seguridad de un sistema de computo es la característica que permite garantizar:

- Que se opera como se espera que lo haga
- Que es ajeno a los riesgos
- Que es ajeno a toda amenaza
- Que no posee vulnerabilidades
- Que es confiable
- Que funcione sin fallos

Una amenaza es la circunstancia o evento que puede causar daño a un sistema. Frecuentemente aprovecha una vulnerabilidad. Por su parte una vulnerabilidad es la ausencia de una contramedida o debilidad de la misma. La debilidad puede originarse en el diseño, la implantación en los procedimientos para operar y administrar el sistema. En seguridad informática se denomina “hoyo”.

El riesgo es un factor que se compone de tres elementos:

- Peligro: Es la probabilidad de que se presente una amenaza.
- Grado de Exposición: Nivel de afectación (número de personas, bienes, módulos del sistema) que podrían ser afectados.
- Vulnerabilidad: Predisposición de un sistema a ser afectado por un agente perturbador.

Es la probabilidad de una vulnerabilidad sea explotada de acuerdo a un nivel de exposición y al peligro involucrado.

Riesgo= Peligro + Exposición + Vulnerabilidad

Definición: Un sistema de computo es seguro si se puede confiar en que se comportará como se espera que lo haga, que la información en el se mantendrá inalterada y accesible durante el tiempo que su dueño lo desee para los usuarios que el mismo determine.

políticas y Servicios de Seguridad

Una política de seguridad especifica las características de seguridad que una organización debe observar y proveer con el fin de salvaguardar su información.

Un servicio de seguridad es una característica que debe tener un sistema para satisfacer una política de seguridad. Estándares BS 7799 e ISO 17799 consideran que la información como un activo para su protección.

Servicios de Seguridad

➤ Confidencialidad

Sólo el propietario es capaz de descifrar la información.

➤ Autenticación

Se asegura la identidad de la persona del otro lado de la línea. Se puede realizar mediante alguno de los siguientes mecanismos.

→ "Algo que se sabe"

Primeros sistemas de autenticación se basan en claves de acceso: nombre usuario, claves de acceso, nips, password, etc.

→ "Algo que se tiene"

sistemas se basan en dispositivos de acceso como tokens, key generators, etc.

→ "Algo que se es"

Son los más recientes sistemas de autenticación y se basan en la biometría. Distinguen las características físicas únicas de cada individuo como son la huella digital, el iris, la forma del rostro, o de la palma de la mano, etc.

➤ Integridad

Se espera que la información se mantenga inalterada desde su creación

➤ Autorización

Se asegura que la información estará disponible sólo para determinados usuarios con la posibilidad de manejar diferentes niveles de acceso para cada uno de ellos.

➤ No Repudio

Se asegura que el emisor de la información no puede negar haberla enviado.

➤ Disponibilidad

Asegurar que el sistema éste disponible cuando se le requiera.

➤ Auditoria

Asegurar que se defina un registro cronológico de los eventos exitosos y fallidos, que proporcionen evidencia de la actividad del sistema.

Criptología

La seguridad en cómputo se apoya en la Criptografía que se encarga de convertir, a través de algoritmos, un texto normal y comprensible a menos que se posea un cierto conocimiento secreto. En épocas recientes la Cristología se define como la ciencia de usar las matemáticas para cifrar y descifrar información. El criptoanálisis por su parte es la ciencia de analizar y romper la comunicación segura entre 2 puntos mediante análisis del algoritmo empleado.

La criptología involucra tanto criptoanálisis como Criptografía.

Los algoritmos criptográficos pueden estar basados en Criptografía Simétrica o Criptografía Asimétrica.

Criptografía Simétrica.

Requiere que el emisor y el receptor una clave secreta “Llave”, la cual ésta utilizada para cifrar el mensaje cuando se envía y descifrar el mensaje al recibirlo. El gran inconveniente se presenta en el proceso de intercambiar de forma segura la Llave. En éste esquema de seguridad es importante que el emisor disponga de un canal seguro para realizar la entrega de la llave de inicio de la transmisión.

Criptografía Asimétrica

Resuelve el problema de intercambiar el secreto (llave) , utilizando para ello un algoritmo basado en dos llaves, tanto el emisor como el receptor utilizan un par de llaves una “pública” y otra “privada”. La privada es mantenida en secreto por un individuo, la pública ésta disponible para que otro individuo sea capaz de cifrar información importante con ella. Cuando el emisor desea enviar un mensaje lo cifra con la llave pública del receptor, siendo éste el único capaz de descifrarlo mediante el uso de una llave privada. En éste caso se consigue asegurar la confidencialidad del mensaje.

Si el emisor cifra con su llave privada, todo aquel que disponga de la llave pública puede conocer el mensaje, por lo que el objetivo en éste caso no es asegurar confidencialidad, sino que se obtiene una Firma Digital Bajo éste escenario se obtiene:

+ Autenticación, dado que sólo el propietario de la llave primaria puede leer el mensaje.

+Integridad. Considerando que el mensaje requiere llegar sin cambios para que el algoritmo de descifrado pueda operar.

+No repudio, ya que la llave privada ésta sólo en posesión del emisor, por lo que no puede negar su auditoria.

River Shamir Adleman (RSA) y Diffie – Hellman (D-H) son dos sistemas de llaves publicas.

-Message Digest

Conocidos también como Hashes, checksums son el resultado de algoritmos unidireccionales que permiten generar una “huella digital” de un mensaje. Permiten obtener una cadena de longitud fija que es una representación condensada de un mensaje. Permite garantizar la integridad de un mensaje, sin un sólo carácter del mensaje original cambia el Hash será totalmente distinto.

Ejemplos de algoritmos que generan éstas cadenas son: SHA1, SHA2, MD5.

En la actualidad existente herramientas de cifrado que emplean los algoritmos que se han mencionado como ejemplo. Algunos de éstas herramientas son:

GPG: Es un sistema de cifrado de código libre. Es un reemplazo de PGP (Pretty Good Privacy) originalmente creado por Phil R. Zimmermann. Ésta herramienta se puede obtener libremente de [http://www.gnupg.org/\(en\)/download/index.html](http://www.gnupg.org/(en)/download/index.html)

OpenSSL: El proyecto OpenSSL es un desarrollo de código abierto con el propósito de desarrollar un toolkit de SSL y TLS. Se basa en el código fuente de SSLeay, desarrollado por Eric A. Young y Tim J. Hudson. Ésta herramienta se puede descargar de <http://www.openssl.org/source/openssl0.97.tar.gz>

Seguridad en Apache

Para implantar un buen nivel de seguridad en el servidor se recomiendan los siguientes puntos:

- + Mantener en producción una versión actualizada.
- + Evitar el uso de permisos no necesarios en los directorios de apache.
- + Restringir el uso de permisos no necesarios en los directorios de apache.

- + Restringir el uso de SSI y de CGI's.
- + Revisar la frecuencia de las bitácoras.
- + Seleccionar passwords fuertes para los accesos restringidos.
- + Implementar el soporte (mod_ssl).
- + Agregar módulos orientados a incrementar el nivel de seguridad. (mod_access, mod_auth, mod_security):

mod_security

- Filtra las peticiones (request) que recibe el servidor Web.
- Analiza el contenido de las peticiones.
- Busca evitar ataques de SQL Injection.
- Permite correr Apache en un ambiente Jail.
- Analiza las peticiones e implanta técnicas anti-evasión.

Además se dispone de las directivas **Order**, **Allow** y **Deny** para permitir o negar el acceso al servidor.

SSL

SSL (Secure Socket Layer) es un protocolo diseñado por la empresa Netscape Communications, que permiten cifrar la conexión, incluso garantiza la autenticación. Se basa en la criptografía asimétrica y en el concepto de los certificados. La versión estandarizada por el IETF se conoce como TLS.

Su mayor ventaja es que funciona entre la capa de transporte y la capa de aplicación, según el modelo OSI, por esto es muy fácil usarlo para proteger los protocolos de la capa de aplicación (por ejemplo FTP, gopher, HTTP) sin tener que realizar cambios importantes en los mismos.

Objetivos de SSL.

Seguridad de la información: SSL garantiza que terceros no tengan acceso a la información mientras viaja por internet al encriptarla.

Integridad de los datos: La información recibida desde un servidor por SSL puede ser “validada” para comprobar que no ha sido alterada en la trayectoria.

Autenticidad de los Datos: Mediante algoritmos de encriptación, es posible comprobar que los datos realmente han llegado del servidor que el cliente espera. Eso evita que alguien se haga pasar por un sitio para cometer fraudes. (evitando ataques como Phishing, Man in the Middle, etc.).

Ataques

Acciones que tienen el que cualquier parte de un sistema de información automatizado, deje de funcionar de acuerdo con su propósito definido.

Tipos de ataques

Pasivos. Se refieren a intervenciones que ponen en riesgo la integridad de la información.

Activos. Incluyen alguna modificación del mensaje (información) o la creación de mensajes falsos.

Firewalls

De manera general el firewall examina el tráfico de la red tanto entrante como saliente y sobre la base de ciertos criterios determina si deja o no pasar los paquetes de información. Si detecta algo normal puede tener procedimientos a seguir o poner en aviso a un administrador.

Tipos de Firewalls.

Packet Filters (Filtrado de Paquetes a nivel TCP/IP).

Circuit Level Gateways (SOLAMENTE PERMITE EL ACCESO HASTA LA CAPA 4 DEL MODELO OSI).

Implementación de un Firewall en Linux con IPTABLES

Para implementar un firewall con Linux se debe determinar:

- El nivel de seguridad requerido.
- El tráfico en la red.
- El tráfico que va a salir en la red.

IPTables y NetFilter son las dos piezas principales de productos firewall disponibles gratuitamente en distribuciones Linux. IPTables es usado para construir reglas en tanto que NetFilter es puente entre el kernel de Linux e IPTables.

Con IPTables se puede realizar:

- Filtrado de dirección remota.
- Filtrado de dirección destino local.
- Filtrado de puerto de origen remoto y destino local.
- Filtrado del estado de la conexión TCP.
- Sondeos y exploraciones de puertos.
- Ataques DoS

IPTables se puede utilizar para robustecer la seguridad de un servidor Web Apache.

Proxy Server.

Un servidor Proxy es un intermediario entre la red interna de una institución, por ejemplo, e Internet; Puede implementar ciertos criterios de seguridad, así como también caché, lo cual significa que si el Proxy recibe una petición lo primero que hace es buscar en su caché y si encuentra el recurso que se le solicito responde a la petición.

Ataques en Web.

Entre los principales ataques sobre sistemas Web se encuentran:

- SQL Injection
- Cross Site Scripting – XSS

SQL Inyection es un ataque que aprovecha la mala validación de las entradas del usuario para inyectar instrucciones SQL que no corresponden a las realizadas de forma normal por el sistema. Eso da un acceso no permitido sobre la base de datos. El problema consiste en que se atenta con la integridad y confidencialidad de la información que reside en la base de datos de la victima.

Cross Site Scripting busca representar un contenido distinto al original a un visitante específico de un sitio. No modifica realmente el sitio, sólo lo muestra de forma distinta, es

decir, permite mostrar información falsa de un sitio. No modifica realmente el sitio, sólo lo muestra de forma distinta, es decir, permite mostrar información falsa de un sitio auténtico. Y esto atenta contra la integridad de la información que publica un sitio.

Seguridad en MySQL: Cifrado de Passwords

Con gran frecuencia los administradores olvidan las configuraciones importantes de MySQL, lo cual puede poner en riesgo la información almacenada. Un problema es cuando se la asigna un password a la cuenta de root.

```
mysql> Update mysql.user Set password=password('nuevo_password') Where user='root'  
  
mysql>flush privileges
```

MySQL dispone de soporte para cifrar la información que se almacena, ya sea password o incluso toda la base de datos.

+ La función `password` genera una cadena de 41 bytes de longitud basada en un algoritmo doble SHA-1

La función `md5` genera una cadena de 128 bits de longitud basada en el algoritmo md5

La función `sha1` genera una cadena basada en el algoritmo SHA-1.

Cuadro de Resumen

Resumen de problemas y herramientas que los pueden resolver.

Problema	Descripción	Herramienta que soluciona el problema
ACTIVIDAD DE INTRUSOS DENTRO DEL SISTEMA	Es cuando un agente externo al sistema trata de acceder a el sin los permisos necesarios para acceder a los recursos	ETHERREAL
GUSANOS	Es cuando se deja un programa que tiene la capacidad de cambiarse de lugar en las diferentes areas de un equipo de computo y de residir en el sin necesidad de que se ejecute.	FIREWALLS
VIRUS	Programas que al ejecutarse tienen el poder de autoreproducirse saturando la memoria del equipo	ANTIVIRUS
CABALLOS DE TROYA	Aplicación cliente-servidor que permitira al agente cliente acceder a un equipo servidor sin necesidad de tener los permisos necesarios para acceder a el	FIREWALLS
SQL INYECTION	Este concepto se maneja en las aplicaciones de Internet que pasan parámetros SQL por la URL de un navegador WEB dejando a este vulnerable a la aplicación	MANDAR VARIABLES POR METODO POST
PHISING	Se hace este procedimiento utilizando ingeniería social por la cual se piden numeros de tarjeta de credito o de cuentas de banco	No permitirse dejar engañar por este tipo de mensajes
CAMBIO DE CONTRASEÑAS	Es de buen uso de esta tecnica dependiendo la severidad que se tenga de la información que almacena o con la que se hacen transacciones en un equipo	Crear Politicas de cambio de contraseñas dependiendo el nivel de severidad que se tenga en una aplicación

1.12 ¿QUÉ ES UN LENGUAJE DE PROGRAMACIÓN?

Los lenguajes de programación son un conjunto de instrucciones a los cuales se les otorga una estructura lógica que indica a la computadora que realice una o varias tareas o acciones específicas.

Lenguajes de programación: Lenguaje Maquina, Ensamblador, Alto Nivel

Los programadores escriben instrucciones en diversos lenguajes de programación, algunos de los cuales comprende directamente la computadora, mientras en otros requieren pasos intermediarios de traducción. En la actualidad se utilizan cientos de lenguajes de programación los cuales se dividen en tres tipos.

1. Lenguaje Maquina.
2. Lenguaje Ensamblador.
3. Lenguaje de Alto Nivel.

Cualquier computadora puede entender de manera directa sólo su propio lenguaje de maquina. El lenguaje maquina es el lenguaje natural de una computadora en particular y ésta definido por el diseño del hardware de dicha computadora. Por lo general los lenguajes maquina consisten en cadenas de números (que finalmente se reduce a unos 1 y 0 que instruyen a las computadoras para realizar sus operaciones más elementales, una por una. Estos lenguajes son dependientes de la maquina(es decir, un lenguaje maquina en particular puede usarse solamente en un tipo de computadora). Dichos lenguajes son difíciles de comprender para los humanos.

La programación en lenguaje maquina era demasiado lenta y tediosa para la mayoría de los programadores. En vez de utilizar las cadenas de números que las computadoras podían entender directamente, los programadores empezaron a utilizar abreviaturas del in-

gles para presentar las operaciones elementales. Éstas abreviaturas formaron la base de los lenguajes ensambladores. Los programas traductores conocidos como ensambladores se desarrollaron para convertir los programas en lenguaje ensamblador a lenguaje maquina, a la velocidad de la computadora.

El uso de las computadoras se incremento rápidamente con la llegada de los lenguajes ensambladores, pero la programación en estos lenguajes aun requería de muchas instrucciones para llevar a cabo incluso las tareas más simples. Para agilizar el proceso de programación se desarrollaron los lenguajes de alto nivel, en donde podían escribirse instrucciones individuales para realizar tareas importantes. Los programas traductores, denominados compiladores, convierten lenguajes de lenguaje de alto nivel a lenguaje maquina. Los lenguajes de alto nivel permiten a los programadores escribir instrucciones que son similares al ingles común. Obviamente, los lenguajes de alto nivel son mucho más recomendables, desde el punto de vista del programador, que los lenguajes de alto nivel son mucho más recomendables, desde el punto de vista del programador, que los lenguajes maquina o ensamblador, C, C++ y Java son ejemplos de lenguajes de alto nivel y además son los más poderosos y más ampliamente utilizados.

El proceso de compilación de un programa escrito en lenguaje de alto nivel a un lenguaje maquina puede tardar un tiempo considerable. Los programas interpretes se desarrollaron para ejecutar programas en lenguaje de alto nivel directamente, sin necesidad de compilarlos en lenguaje maquina. Aunque los programas compilados son mucho más rápidos que los programas interpretados, los interpretes son populares dentro de entorno de desarrollo de programas, en los cuales los programas sé recompilan frecuentemente, a medida que agregan nuevas características y se corrigen los errores. Una vez que se desarrolla un programa, se puede producir una versión compilada para ejecutarse con la mayor eficiencia.

-El lenguaje de hipermercado (HTML)

HTML (Hyper Text Markup Language) no es un lenguaje de programación como tal, pero lo si, es un lenguaje de marcas, éstas marcas son fragmentos de texto destacado de una forma especial que permite la definición de distintas instrucciones de HTML, tanto los efectos a aplicar sobre el texto como las distintas estructuras del lenguaje. A éstas marcas se les denomina etiquetas y son la base principal del lenguaje HTML. Un documento HTML es un archivo de texto con etiquetas que variaran la forma de su presentación. El programa que sirve documentos HTML se llama servidor Web. Y la manera en que se solicitan estos documentos o también llamadas páginas es a través del browser o navegador.

Una etiqueta será un texto incluido entre los símbolos < y mayor que >. El texto incluido dentro de los símbolos será explicativo de la utilidad de la etiqueta. Existe normalmente una etiqueta de inicio y otra de fin, la de fin contendrá el mismo texto que la de inicio añadiéndole al principio una diagonal /. El efecto que define la etiqueta tendrá validez para todo lo que éste incluido entre las etiquetas de inicio y fin ya sea texto plano u otras etiquetas HTML.

El uso de mayúsculas o minúsculas en las etiquetas es indiferente, se interpretan del mismo modo en ambos casos. Pero para efectos de compatibilidad con nuevas tecnologías se utilizan minúsculas.

+Atributos de las etiquetas

Las etiquetas pueden presentar modificadores que se llaman atributos que permitirán definir posibilidades de la instrucción HTML. Estos atributos se definen en la etiqueta de inicio y consisten normalmente en el nombre del atributo y el valor que toma separados

por un signo de igual. El orden que se incluyan los atributos es indiferente, no afectando al resultado.

Cuando el valor lo toma el atributo tiene más de una palabra deberá expresarse entre comillas, en otro caso no será necesario

En éste caso la etiqueta A presentara un atributo HREF cuyo valor es <http://www.ateb.com.mx>. Igualmente una etiqueta una etiqueta podría presentar varios atributos:

```
<HR align=LEFT NOSHADE SIZE=5 WIDTH=50%>
```

En éste caso la etiqueta **<HR>** presenta cuatro atributos. El segundo atributo NOSHADE es un caso especial que no presenta valor.

+Estructura básica

Un documento HTML ésta definido por una etiqueta de apertura **<html>** y una etiqueta de cierre **</html>**. Dentro de éste se dividen dos partes fundamentales, la cabecera, delimitada por la etiqueta **<head>** y el cuerpo, delimitado por la etiqueta **<body>**. Por lo tanto la estructura de un documento HTML será:

+Espaciados y saltos de línea

En HTML no ésta permitido más de un elemento blanco (espacios, tabuladores, saltos de línea), separando cualquier elemento del texto, todos estos son convertidos a un único espacio blanco y el resto se omiten en la representación del documento. En el documento fuente se puede usar el espacio que se desee, y no deberá estar bien formateado, éste se conseguirá con las etiquetas HTML.

+Características Especiales.

Los caracteres acentuados y algunos caracteres especiales que usa el lenguaje HTML para definir sus etiquetas no se pueden incluir en un documento de manera normal, se deben de utilizar una serie de secuencias de escape que al mostrar el documento se sustituyen por el carácter deseado.

Éstas secuencias de escape todas con el símbolo ampersand (&), seguido de un texto (siempre en minúsculas) que define el carácter deseado y terminar con el símbolo punto y coma (;).

+Listas

El lenguaje HTML proporciona un modo sencillo de representar elementos en forma de lista. Dentro de una lista se puede incluir cualquiera de los elementos HTML, e igualmente una lista puede incluirse dentro de formularios, tablas etc.

+Hiperenlaces

En la utilidad básica del hipertexto, permite indicar zonas de texto o imágenes que si son seleccionados nos traslada a otros documentos HTML u otras zonas del documento actual.

Para definir un hiperenlaces se puede utilizar cualquier elemento HTML, no debe de ser texto necesariamente, pueden ser, cabeceras (<h1>), cualquiera de los estilos una imagen, etc. Un hiperenlaces igualmente podrá definirse dentro de cualquier elemento HTML: listas, párrafos de texto, tablas, formularios.

El texto del hiperenlaces aparece normalmente resaltado sobre el texto normal, en azul y subrayado, en el caso de las imágenes, si tienen definido un borde éste aparecerá

resaltado en color azul. La mayoría de los navegadores, cuando la zona por la que pasa el cursor es sensible, éste cambia de aspecto indicándolo y en la parte baja de la pantalla se indica el hiperenlaces que se activa al pulsar en esa zona. Mediante los atributos <body> es posible cambiar el color de los hiperenlaces.

+Tablas

Permite la representación de datos por filas y columnas, en forma tabular. La definición es muy flexible indicando sólo los elementos que forman cada fila y columna, calculándose de forma automática el tamaño que deben de tener las celdas. En una tabla se pueden introducir todo tipo de elementos como imágenes, enlaces, texto, listas cabeceras, formularios.

En la etiqueta inicial <**table**> se definen atributos que afectarán a toda la tabla, todos los atributos son opcionales. Todo lo que se incluya dentro de la instrucción de tabla se considera como tal, pudiendo definir las tablas anidadas (tablas dentro de tablas).

+Formularios

Los formularios son plantillas que permiten la creación de documentos HTML, con peticiones de datos. La principal utilidad de los formularios es la posibilidad de crear cuestionarios, encuestas, páginas de comentarios o cualquier documento en la que se desee una interacción por parte del usuario.

Se pueden definir distintos tipos de recuadros de dialogo, botones de selección, menús de múltiples opciones, etc. Para permitir obtener los datos de una manera más intuitiva.

Existe una etiqueta para la creación de formularios ésta es <**form**>, que tiene la siguiente estructura:


```
<form action="archivo que trata el formulario" method="POST | GET">
```

+Frames

Con las frames es posible dividir la ventana del navegador en varias subregiones (frames), permitiendo mostrar una URL, distinta en cada una de ellas. En cada frame se permite:

- Mostrar su propia URL, diferenciada del resto de frames de la pantalla, de ésta forma un hiperenlaces puede tener como destino un documento y la frame en el que se mostrara.
- Tendrán asociado un nombre, que lo distinguirán del resto de los frames de la pantalla y permitirá usarlo en los hiperenlaces.
- En el caso que se cambie el tamaño de la ventana, se podrá determinar si la frame se ajusta a éste tamaño o mantiene su tamaño intacto.

Esto permite crear nuevo tipos de documentos en los que por ejemplo se mantendrá una región estática(lista de enlaces, barra de botones, formulario) y otra zona dinámica en la que se mostrara el resultado. De ésta forma una de las principales utilidades de las frames es la creación de páginas con índice (por ejemplo un manual) o una cabecera estática, consiguiendo así una mejora de la navegación al poder acceder al índice de una manera más rápida y efectiva.

Por ultimo existe un atributo que se puede realizar en la definición de frames, éste atributo es "target" que indicara la frame de destino de un hiperenlaces, éste se muestra en la ventana del navegador sustituyendo el documento actual, con las frames se puede especificar que frame será la de destino, no siendo necesario que sea la frame del documento actual, con las frames se puede especificar que frame será la de destino, no siendo necesario

que sea la frame del documento actual. Como nombre de la frame que usará el nombre que sé que se especifico en el atributo name de la etiqueta **<frame>**.

Lenguaje de Programación PHP

-¿ Que es PHP?

PHP² (acrónimo de “PHP : Hypertext Preprocessor”) es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML (Hyper Text markup Language) y ejecutando el servidor. Para comprender esto mostraremos el siguiente ejemplo:

```
<html>
<head>
<title>Ejemplo</title>
</head>
<body>
<? php
    Echo “Hola, &iexcl; soy un script de php!”;
?>
</body>
</html>
```

Puede apreciarse que no es lo mismo que un script escrito en otro lenguaje de programación como Perl o C. En vez de escribir un programa con muchos comandos para crear una salida de HTML, escribimos el código HTML con cierto código PHP embebido (incluido) en el mismo, que producirá cierta salida (en el ejemplo, producirá un texto). El código PHP se incluye entre etiquetas especiales de comienzo y final que permiten entrar y salir del modo PHP.

Lo que distingue a PHP de tecnologías como JavaScript, la cual se ejecuta en la maquina cliente, es que el código PHP es ejecutado en el servidor. Si se tuviera un script similar al del ejemplo en un servidor, el cliente solamente recibirá el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. Aunque el desarrollo de PHP ésta concentrado en la programación de scripts en el lado del servidor, se puede utilizar para muchas otras cosas.

-¿Qué más puedo hacer PHP?

PHP puede hacer cualquier cosa que pueda hacer un script CGI, como procesar la información de los formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

- Scripts del lado del servidor: Éste es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. Él interprete PHP (CGI ó modulo), un servidor Web y un navegador. Es necesario correr el servidor Web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose, con el servidor Web.
- Scripts en la línea de comandos. Se puede crear un script PHP y correrlo sin ningún servidor Web o navegador. Solamente se necesita él interprete PHP para usarlo de ésta manera. Éste tipo de uso es ideal para scripts ejecutándose regularmente desde cron (en unix o Linux) o el planificador de

tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto.

- Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si se conoce bien PHP, y se quisieran utilizar algunas características avanzadas en programas clientes, se puede utilizar PHP- GTK para escribir dichos programas. También es posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

De modo que, con PHP se tiene la libertad de elegir el sistema operativo y el servidor. También tiene la posibilidad de usar programación procedimental o programación orientada a objetos. Aunque no todas las características estándares de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas librerías y aplicaciones grandes (incluyendo la librería PEAR) están escritas íntegramente usando programación orientada a objetos.

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

Las siguientes bases de datos están soportadas actualmente:

ADABAS D	INGRES	ORACLE (OCI7 AND OCI8)
DBASE	INTERBASE	OVRIMOS
EMPRESS	FRONTBASE	POSTGRESQL
FILEPRO (READ-ONLY)	MSQL	SOLID
HYPERWARE	DIRECT MS-SQL	SYBASE
IBM DB2	MYSQL	VELOCIS
INFORMIX	ODBC	UNIX DBM

También se cuenta con una extensión DBX de abstracción de bases de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Adicionalmente, PHP soporta ODBC (el estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte el estándar.

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

Si se usa PHP en el campo del comercio electrónico, se encontrarán muy útiles las funciones Cybercash, Cybermut, Verisign Payflow Pro y C CVS para programas de pago.

Instalación de PHP

La instalación de PHP se lleva a cabo instalando el módulo de PHP en el servidor Web Apache. Para instalar el servidor Apache con soporte para PHP se lleva a cabo los siguientes pasos:

1. Primeramente se obtiene PHP de la página <http://www.php.net>
2. Se descomprime el archivo obtenido con el comando: `# tar -zxvf php-x.x.x.tar.gz` Al descomprimir éste archivo se genera una carpeta, se ingresa a la carpeta: `# cd php-x.x.x.`
3. Después se empieza con la instalación de Apache:
`#!/configure --prefix=/usr/local/php --with-apxs2=/usr/local/apache2/bin/apxs`
`#make`
4. Una vez instalado el módulo de PHP la siguiente línea debe estar en el archivo: `/usr/local/apache2/modules.`

```
libphp4.so //modulo de php.
```

En el archivo de configuración de Apache(`httpd.conf`) debe aparecer la siguiente línea.

```
LoadModule php4_module module/libphp4.so
```

Y se debe agregar la siguiente línea en el mismo archivo:

```
AddType application/x-httpd-php .php .html
```

5. Para comprobar si la instalación fue exitosa sé checa con el siguiente script

```
<? php
    print "BIENVENIDOS A PHP"
?>
```

Se guarda éste script en donde se guardan los archivos del servidor (en éste caso /usr/local/apache2/htdocs) y se ejecuta desde el navegador de preferencia de la siguiente manera.

<http://localhost/script.php> o <http://127.0.0.1/script.php>

Sintaxis básica

Escapar del HTML

Para interpretar un archivo, php simplemente interpreta el texto del archivo hasta que encuentra uno de los caracteres especiales que delimitan el inicio del código de PHP. Él interprete ejecuta entonces todo el código que encuentra, hasta que encuentra una etiqueta de fin de código, que le dice al interprete que siga ignorando el código siguiente. Éste mecanismo permite embeber código PHP dentro de HTML: todo lo que ésta fuera de las etiquetas de PHP se deja tal como está, mientras que el resto se interpreta como código.

Hay cuatro conjuntos de etiquetas que pueden ser usadas para detonar bloques de código PHP. De éstas cuatro, sólo 2 (<?php.... ?> y <script language="php">. . .</script>) están siempre disponibles; el resto pueden ser configuradas en el archivo php.ini para ser o no aceptadas por el interprete. Mientras que el formato corto de las etiquetas (short- form tags) y el estilo ASP (ASP – style tags) pueden ser convenientes, no son portables como la versión de formato largo de etiquetas. Además, si se pretende embeber código PHP con

XML o XHTML, será obligatorio el uso del formato `<?php . . . ?>` para la compatibilidad con XML.

Tipos de datos

PHP soporta los siguientes tipos de datos

Enteros: Los Integers, o enteros, pueden tener distintos valores numéricos que se expresan con diferentes anotaciones.

Binarios de punto flotante: Este tipo de datos son los números con la coma móvil, por ejemplo, 9.876. La sintaxis para utilizarlos es bastante simple

Strings: Las cadenas pueden estar delimitadas por " o '. Si la cadena está delimitada por comillas dobles, cualquier variable incluida dentro de ella será sustituida por su valor

Arreglos: Los Arrays en PHP se pueden utilizar tanto como Arrays indexados o como Arrays asociativos. Los Arrays de una sola dirección, pueden ser tanto escalares como asociativos

Objetos: Para inicializar un objeto se utiliza el método *new* , y para acceder a cada uno de sus métodos se utiliza el operador ->

En general el tipo de dato de una variable no es decidido por el programador sino que lo decide el lenguaje en tiempo de ejecución, la instrucción `set type` puede usarse para forzar el tipo de dato de una variable en los raros casos en que esto sea necesario. Todas las variables en php se utilizan el signo "\$" precediendo el nombre de la variable.

Estructuras de Control

Las estructuras de control son sentencias que modifican el comportamiento de cierto programa, a través de evaluar una condición; en algunas estructuras (if, if...else) si la condición a evaluar es verdadera se ejecuta el bloque siguiente de instrucciones, si no es verdadera se ejecuta otro de bloque de instrucciones o simplemente no se ejecuta el primer bloque. En otras estructuras (while, for, do..while) la condición sirve para ciclar un bloque de instrucciones para que se ejecute un numero de veces deseado. La sintaxis de las estructuras de control se muestra a continuación.

Los parámetros que recibe la función pueden ser enteros, flotantes, strings, arreglos, u objetos es decir cualquiera de los tipos de datos soportado por PHP. El valor devuelto por la función también puede ser cualquier tipo de datos de PHP.

Finalmente los nombres de funciones pueden guardarse en variables e invocarse las mismas usando el nombre guardado en una variable

Esto permite guardar nombres de funciones en tablas, archivos, arreglos etc. Lo cual da lugar a ciertas maniobras interesantes.

Bases de Datos

Antes de definir el concepto de una base de datos es necesario definir la siguiente información:

Datos: Son hechos, números, letras o símbolos en bruto que por si solos no tienen ningún significado.

Información: Es la interpretación de los datos. La información consiste en conocimientos producidos como resultado de las operaciones de procesamiento de datos.

Esquema: El diseño completo de una base de datos se llama esquema de la base de datos.

Una base de datos es una colección estructurada de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. Los datos que puede almacenar una base de datos que conforman la información de algo tan simple como la de una agenda, un contador, o un libro de visitas, o tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada por una red corporativa. En éste sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

En la actualidad, y gracias al desarrollo tecnológico de campos como el cómputo y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen sistemas manejadores de bases de datos (DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Un **DBMS** consiste en una colección de datos interrelacionados y en conjunto de programas para acceder a dichos datos. El objetivo principal de un DBMS es proporcionar una manera de almacenar y recuperar la información de una base de datos de manera que sea tanto practica como eficiente.

Para definir una base de datos eficiente es necesario seguir un modelo de datos; dicho modelo es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

El modelo más utilizado hoy en día es el Modelo Entidad- Relacional. Éste modelo de datos ésta basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una (cosa) u (objeto) en el mundo real que es distinguible de otros objetos. Las entidades se describen en una base de datos mediante un conjunto de atributos. Una relación es una asociación entre varias entidades. Las entidades son representadas con rectángulos, las elipses representa los atributo, y los rombos representan relaciones.

Existe otro modelo conceptual llamado Modelo Relacional. En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas y, cada columna tiene nombre único. En éste modelo se encuentran los conceptos de grado y cardinalidad. El grado de una tabla se refiere al número de atributos de una tabla. La cardinalidad es el número de tuplas que contiene una tabla.

Un sistema de base de datos proporciona un lenguaje de definición de datos (DDL) para especificar el esquema de las bases de datos y un lenguaje de manipulación de datos (DML) para expresar consultas a las bases de datos y las modificaciones. En la practica, los lenguajes de definición y manipulación de datos no son lenguajes separados, en su

lugar simplemente forman parte de un único lenguaje de bases de datos, tal como SQL (Structured Query Language) que es actualmente un estándar en la industria.

Ciclo de vida de una base de datos.

Análisis: En la fase de análisis se entrevista a los accionistas y se examinan todos los sistemas existentes para identificar los problemas, las posibilidades y los límites. En esta fase se determinan los objetivos y el ámbito del nuevo sistema.

Diseño: La fase de diseño se encarga de la creación del diseño conceptual a partir de las necesidades determinadas previamente. También se crea un diseño lógico y físico para preparar la implementación de la base de datos.

Implementación: En esta fase se instala el sistema de administración de la base de datos (DBMS), se crea la base de datos y se cargan o importan los datos.

Pruebas: En la fase de pruebas se examinan la base de datos y se ajusta, por lo general junto a las aplicaciones asociadas.

Puesta en marcha: En esta fase la base de datos opera normalmente, produciendo información para sus usuarios.

Mantenimiento: En la fase de mantenimiento se introducen cambios en la base de datos en respuesta a las nuevas necesidades o se modifican las condiciones operativas.

La Relación entre las base de datos y los lenguajes de programación

Los bases de datos al permitir almacenar conjuntos de datos, por si mismas no representan información y tampoco los datos son meramente información. Para que los datos

puedan tener un sentido, sean útiles y tengan algún valor para la toma de decisiones en las empresas e instituciones, es necesario procesar estos datos.

El procesarlos implica extraerlos de la base de datos, pero no sólo es extraerlos, se necesitan extraer los datos que se requieran únicamente, se deben analizar y presentar en forma adecuada, de los sistemas manejadores de bases de datos, extraen los datos de las bases de datos, los analizan y aplicaciones que los utilizan, otorgando flexibilidad al desarrollo de sistemas, ya que se pueden utilizar diferentes lenguajes de programación para crear diversas aplicaciones que utilicen el mismo conjunto de datos.

El manejador de Bases de Datos de MySQL

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB, MySQL AB que es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. Una de las razones para el rápido crecimiento de popularidad de MySQL, es que trata de un producto Open Source, y por lo tanto, va de la mano con éste movimiento.

Como se mencionó anteriormente, una base de datos al ser una colección estructurada de datos, se necesita un sistema de administración de bases de datos, como MySQL para agregar, acceder, y procesar los datos almacenados en una base de datos.

Una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un sólo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hace posible combinar datos de varias tablas cuando se necesitan consultar datos. La parte de SQL de “MySQL” significa “Lenguaje Estructurado de Consultas”, y es el lenguaje más usado y estandarizado para acceder a bases de datos relacionales.

El servidor de bases de datos de MySQL es más rápido, seguro, y fácil de usar. Fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor de MySQL ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet.

MySQL, consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas Web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones posibilidades.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indización de campos de texto.
- Procedimientos almacenados
- Triggers.
- Vistas.
- Cursores.
- Transacciones Distribuidas XA.
- Esquema de Información.

Trabajando en MySql

Para conectarse al servidor, se necesita de un nombre de usuario (login) y de una contraseña (password), y si el servidor al que se desea conectar está en una maquina distinta en la cual se ésta trabajando, también se necesita indicar el nombre o la dirección IP de dicho servidor. Una vez que se conoce estos tres valores, la conexión se realiza de la siguiente manera:

```
Mysql -h NombreDelServidor -u NombreDeUsuario -p
```

Cuando se ejecuta éste comando, se pide también la contraseña para el nombre de usuario que se ésta usando.

Si la conexión al servidor MySQL se pudo establecer de manera satisfactoria, se recibe el mensaje de bienvenida e inmediatamente aparece el prompt MySQL

```
Welcome to the MySQL monitor. Commands end with \g.
Your MySQL connection id is 5563 to server version: 3.23.41

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

MySQL>
```

Éste prompt indica que MySQL está listo para recibir comandos.

Después de haber realizado la conexión de una manera satisfactoria, la desconexión se puede efectuar en cualquier momento al escribir “quit”, “exit” o presionar CONTROL-D.

Una vez que se ha hecho una conexión con el servidor se puede empezar a trabajar. Véase el siguiente ejemplo

```
MySQL SELECT VERSIÓN(), CURRENT_DATE;
+-----+-----+
| VERSIÓN() | CURRENT_DATE |
+-----+-----+
| 3.23.41   | 2002-10-01   |
+-----+-----+

1 row in set (0.03 sec)
```

Éste comando ilustra distintas cosas acerca de MySQL.

- Un comando normalmente consiste de una sentencia SQL seguida por un punto y como

- Cuando se ingresa un comando, MySQL lo manda al servidor para que lo ejecute, muestra los resultados y regresa el prompt indicando que está listo para recibir más consultas
- MySQL muestra resultados de la consulta como una tabla (filas y columnas). La primera fila contiene etiquetas para las columnas. Las filas siguientes muestran los resultados de la consulta. Normalmente las etiquetas de las columnas son los nombres de los campos de las tablas que estamos usando en alguna consulta. Si lo que se está recuperando es el valor de una expresión (como en el ejemplo anterior) las etiquetas en las columnas son la expresión en sí.
- MySQL muestra cuantas filas fueron registradas y regresadas y cuanto tiempo tardo en ejecutarse la consulta, lo cual puede dar una idea de la eficiencia del servidor, aunque estos valores pueden ser un tanto imprecisos ya que no se muestra la hora de la CPU y porque pueden verse afectados por otros factores, tales como la carga del servidor y la velocidad de comunicación de una red.
- Las palabras clave pueden ser escritas usando mayúsculas y minúsculas.

Es posible escribir más de una secuencia por línea, siempre y cuando estén separadas por punto y coma.

```
Mysql> SELECT VERSIÓN(); SELECT NOW();
```

VERSIÓN()
3.23.41

1 row in set (0.01 sec)

NOW()
2002-10-28 14:26:04

Un comando no necesita ser escrito en una sola línea, así que los comandos que requieran de varias líneas no son un problema. MySQL determinara en donde finaliza la sentencia cuando encuentre el punto y coma, no cuando encuentre el fin de la línea.

En la siguiente tabla se muestran cada uno de los prompts que se pueden obtener al ir escribiendo comandos y una breve descripción de su significado para MySQL:

PROMPT	S IGNIFICADO
Mysql>	Listo para nueva consulta
->	Esperando la línea siguiente de una consulta multitarea
'>	Esperando la siguiente línea para completar una cadena que comienza con una comilla sencilla
“>	Esperando la siguiente línea para completar una cadena que comienza con una comilla doble(“)

Como crear una Base de Datos

En MySQL se cuenta con el comando show para mostrar todas las bases de datos que se encuentran en el servidor.

Dependiendo de las bases de datos que se encuentren en el servidor son las que se van a mostrar. Las bases de datos de “MYSQL” es requerida, ya que ésta tiene la información de los privilegios de los usuarios de MYSQL. La base de datos “test” es creada durante la instalación de MYSQL con el propósito de servir como área de trabajo para los usuarios que inician con el aprendizaje de MySQL.

Al crear una base de datos no se selecciona ésta de manera automática; se debe hacerlo de manera explicita se usa la sentencia USE.

La base de datos se crea una sola vez, pero debe ser seleccionada cada vez que se inicia una sesión en el servidor de MySQL; Por esto preferentemente debe indicarse la base de datos sobre la cual se va a trabajar al momento de conectarse al servidor.

```
Shell>mysql -h nombreServidor -u NombreUsuario -p BasedeDatos
```

Crear la base de datos no conlleva mucha ciencia, pero en éste momento la base de datos ésta vacía como lo indica el comando SHOW TABLES.

+Creación de Tablas

Para crear tablas en la base de datos ventas, se utiliza la sentencia CREATE TABLES de la siguiente manera

Se crea una tabla en la base de datos que consta de tres campos, id_vendedor, nombre y apellido_paterno. Cada campo debe ser definido de un tipo de dato, como se muestra MYSQL soporta un buen numero de tipos de columnas en varias categorías, tipos numéricos, tipos de fecha y hora y tipos de cadena.

Se usa la sentencia CONSTRAINT para definir la restricción de llave primaria para el campo id_vendedor. La llave primaria de una tabla sirve para poder acceder de manera única a los datos de esa tabla.

Type=InnoDB. InnoDB es un mecanismo que permite realizar transacciones seguras en una base de datos, con capacidades de aplicar commit (realizar cambios), rollback (deshacer trabajo) y crash recovery (recuperación de datos).

Una llave foránea es un campo de una tabla, en éste caso de la tabla venta-producto, que hace referencia a un campo de otra tabla, en éste caso vendedor o producto, que se encuentra definido como llave primaria. Esto es útil para el cruce de tablas cuando se requieren extraer datos de varias tablas.

+Inserción de datos

Una vez creadas las tablas en la base de datos, se insertan datos en ellas haciendo uso de la sentencia INSERT, como se muestra a continuación.

```
INSERT INTO vendedor VALUES (1,'GUALITO', 'CALVILLO')
```

Éste comando inserta en la tabla vendedor los valores dentro de los paréntesis que corresponden en orden a los campos definidos en la tabla. De la misma forma se pueden insertar n registros dentro de la tabla, mientras no exceda el tamaño permitido de la tabla que es limitado por el S.O. donde se está corriendo el servidor MySQL.

+Selección de datos

Para extraer los datos de las tablas se usa el comando SELECT:

La estructura del comando se define de la siguiente manera:

```
SELECT nombre_campo[1], nombre_campo[2], .....nombre_campo[n]  
FROM nombre_tabla[1], nombre_tabla[2],.....nombre_tabla[n]  
WHERE condiciones
```

Se usa la sintaxis **nombre_tabla.nombre_campo** para referenciar los campos de una tabla. La sentencia WHERE hace uso de ésta referencia para establecer las condiciones de búsqueda de los datos. En el ejemplo, las condiciones se establecen de la siguiente manera.

En éste ejemplo se utiliza la función SUM(), para efectuar la suma de todos los campos de la columna cantidad. Las funciones pueden ser utilizadas como en éste caso directamente en las consultas. MySQL soporta un gran número de funciones.

```
SELECT nombre_producto, precio, SUM(cantidad) AS productos_vendidos,  
      SUM(cantidad) * precio AS total  
FROM venta_producto INNER JOIN producto ON  
      venta_producto.id_producto = producto.id_producto  
GROUP BY nombre_producto
```

En éste comando se realiza la unión de las tablas INNER JOIN. Se especifican los campos a seleccionar, en éste caso nombre_producto y precio; además se utiliza la función SUM() para sumar los campos de la columna cantidad y se le pone un nombre a la columna resultante con la opción AS productos_vendidos. Se realiza además la operación de sumar los campos de la columna cantidad y se multiplica el resultado con el operador de multiplicación * por los campos de la columna y se le asigna un nombre a la columna, total.

Después se indican las tablas de las cuales se extraen los campos, venta_producto INNER JOIN (en unión) con producto y la condición para traer los datos ON donde el campo id_producto de la tabla venta_producto sea igual al campo id_producto de la tabla producto.

+Alterar y Borrar una tabla

Si se requiere modificar la estructura de una tabla, existe la sentencia SQL ALTER. Ésta sentencia permite agregar campos a las tablas (ADD), cambiar campos, borrar llaves primarias con ayuda de la sentencia DROP, renombrar una tabla o adicionar llaves primarias.

Y se requieren borrar todos o algunos registros de una tabla se utiliza la sentencia DELETE.

El diplomado de desarrollo de sistemas de sistemas con software libre nos enseñó cosas muy importantes como las alternativas de herramientas de desarrollo con las que se cuenta, sin necesidad de utilizar una herramienta con licencia en este caso un sistema operativo no comercial que cuenta con una funcionalidad mucho más robusta que un sistema operativo Windows es Linux, cuenta con muchas características que rebasan considerablemente en rendimiento al sistema operativo de Microsoft, en cuanto a herramientas de desarrollo, en especial para la programación de herramientas Web podemos contar con muchas herramientas libres en este caso PHP satisface de buena manera las herramientas a las herramientas licenciadas como las de las tecnologías propietarias como es la tecnología .NET. diseñada por Microsoft.

En si existe un tecnología libre de cada una de las herramientas que existen en el mercado dando al desarrollador la oportunidad de decidir entre que herramientas satisface sus necesidades y en cuales tendrá que invertir mayor tiempo y soporte de la misma la, ventaja de las herramientas licenciadas es que cuentan con un soporte tecnológico más robusto que una licencia libre a la vez que las herramientas libres necesitan un proceso mayor de investigación lo que implica un mayor tiempo de horas hombre en cuanto al desarrollo, el utilizar cada una de estas herramientas quedara a consideración de las necesidades que tengan los usuarios de la aplicación y de las necesidades en cuanto costo dinero esfuerzo en e inversión de horas hombre se necesitan para desarrollar un sistema con cada una de estas herramientas ya sean libres o licenciadas.

CAPITULO 2

INFORME DE LOS CURSOS DE PROGRAMACIÓN EN JAVA Y ASP.NET

2.1 LENGUAJE DE PROGRAMACIÓN JAVA

JAVA

Java es un poderoso lenguaje de programación que fue creado como parte de un proyecto denominando Green. En Sun Microsystems para el desarrollo de aplicaciones para dispositivos electrónicos domésticos. Pero Sun se dio cuenta que el mercado de dispositivos electrónicos inteligentes para el hogar no crecía como se esperaba. Así que al popularizarse la WWW en 1993, la gente de Sun puso atención en la gran capacidad que tenía el lenguaje para agregar contenido dinámico y animaciones a páginas Web.

Finalmente en 1995 SUN anuncio formalmente Java, que genero gran interés en la comunidad de negocios, debido al auge que estaba teniendo la WWW y las características del lenguaje hacia ésta. En la actualidad Java se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar servicio de World Wide Web, para proporcionar aplicaciones para los dispositivos domésticos (teléfonos, celulares, radiolocalizadores y asistentes digitales personales [PDA's]) y muchos otros dispositivos.

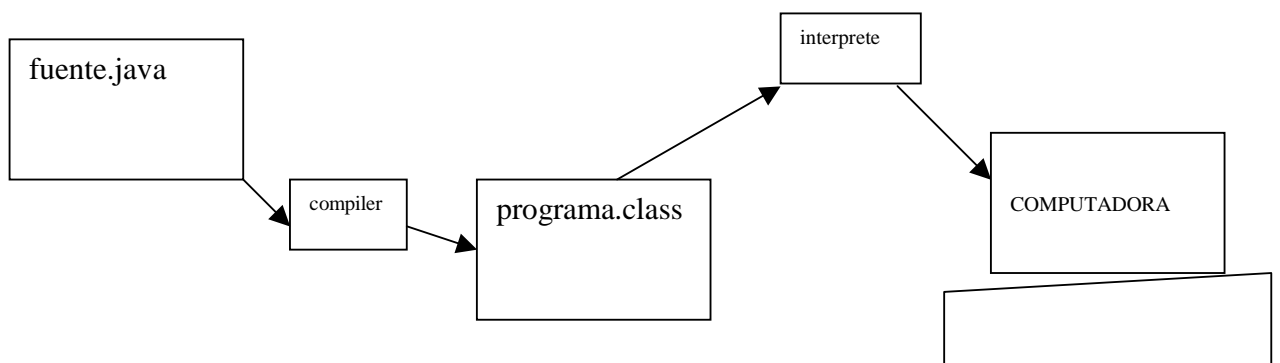
El lenguaje de programación

El lenguaje de programación Java es de alto nivel y se caracteriza por ser

- | | | |
|--------------------------|---------------|-----------|
| -Simple | -Distribuido | -Robusto |
| -De arquitectura neutral | -Multihilo | -Portable |
| -Orientado a objetos | -Interpretado | -Seguro |
| -De alto rendimiento | -Dinámico | |

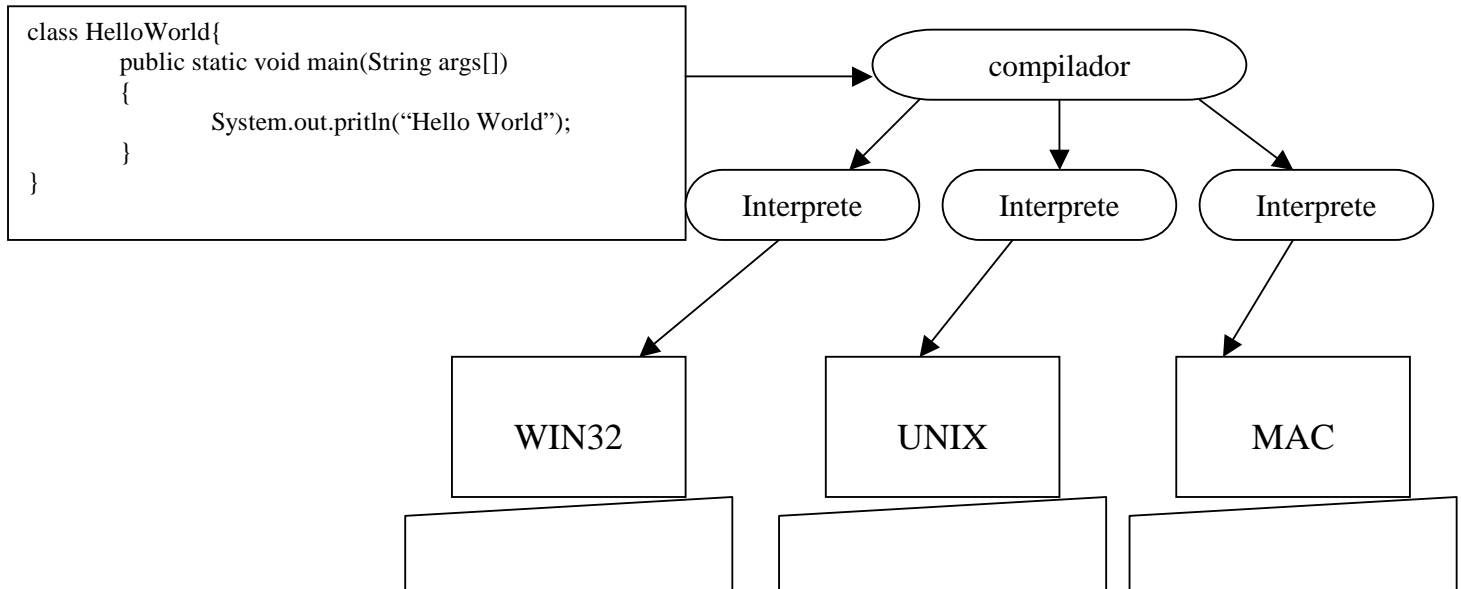
Con la mayoría de los lenguajes de programación, los programas se compilan o interpretan para poder ejecutarlos. En éste sentido Java es inusual ya que un programa es compilado e interpretado. Con el compilador primero se traslada el programa a un lenguaje intermedio llamado Java bytecodes. Que son los códigos independientes de la plataforma y que interpreta a código Java. Él interprete analiza y ejecuta cada instrucción en la computadora. La compilación ocurre una sola vez, y la interpretación ocurre cada vez que el programa es ejecuta.

Figura 2.1 Compilación de programa en Java



Los códigos de bytes de Java hacen que el lema “escribe una vez, corre en cualquier lado”, se haga posible. Se puede compilar un programa en cualquier plataforma que cuente con un compilador de Java. Los bytecodes pueden entonces ser ejecutados en cualquier implementación de la JVM. Esto quiere decir que si una computadora tiene la JVM, el mismo programa escrito en el lenguaje Java puede correr en Windows 2000, Solaris o en una Mac.

Figura 2.2 Código fuente de y equipos trabajando con maquina virtual



La plataforma Java

Una plataforma en el entorno de hardware o software en el cual corre un programa. La mayoría de las plataformas pueden ser descritas como una combinación del sistema operativo y hardware. En cambio la plataforma Java difiere del resto de las plataformas en que es una plataforma sólo de software (software-only) que corre en la cima de otras plataformas basadas en hardware (hardware – based).

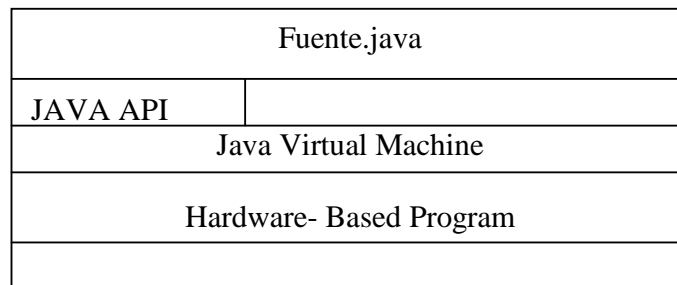
La plataforma Java tiene dos componentes:

- La maquina virtual de Java (Java virtual Machine, JVM).
- La interfase de programación de aplicaciones (Application Programming Interface, API).

La maquina virtual de Java es la base de la plataforma Java y es transportada a varias plataformas de hardware.

El API de Java es una extensa colección de componentes de software que provee una gran capacidad de utilidad, como componentes de interfaces gráficas de usuario (GUI'S). La API de Java está agrupada en bibliotecas de interfaces y clases relacionadas, éstas bibliotecas son conocidas como paquetes.

Figura 2.3 Conformación de un Programa en Java



Como la figura muestra, la API y la Máquina Virtual de Java aíslan el programa de hardware.

El código nativo, es código que después que se compila dicho código compilado corre en una plataforma de hardware específica. Como un entorno independiente de la plataforma de hardware específica. Como un entorno independiente de la plataforma, la plataforma Java puede ser un poco más lenta que el código nativo. Pero aún así, pequeños com-

piladores, interpretes bien afinados y compiladores just-in-time, pueden mejorar el desempeño y ser muy cercano al código nativo sin portabilidad.

Considere el siguiente programa

```
Public class Programa{
    Public static void main(String args[])
    {
        int sum = 0;
        for(int current=1 ; current <= 10 ; current ++){
            sum+=current;
            System.out.println("Suma =" + sum);
        }
    }
}
```

Éste programa suma los números del 1 al 10 y despliega el resultado; la salida de éste programa es:

⇒ Suma = 55

Un programa como éste hace uso de las características tradicionales del lenguaje Java, incluyendo variables operadores y sentencias de control de flujo.

Variables: Una variable es un elemento de datos que es nombrado por un identificador. Se requiere especificar un nombre y un tipo para cada variable que se use en un programa. El nombre de la variable se utiliza para referirse a los datos que la variable contiene. El tipo de variable determina que el valor se puede almacenar y que operaciones se pueden realizar con él. La sintaxis para declarar una variable es: *Tipo nombre*.

El lenguaje de programación Java tiene 2 categorías de tipos de datos: primitivos y de referencia. Una variable de tipo primitivo contiene un valor simple de tamaño y formato

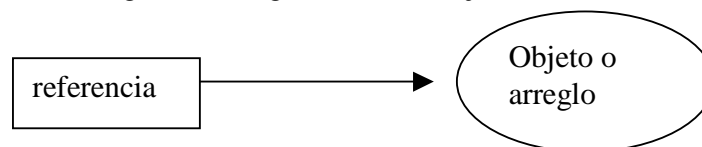
apropiado para su tipo: un número, un carácter o un valor booleano. La siguiente tabla muestra todos los tipos de datos primitivos soportados por la plataforma Java, sus tamaños y formatos y una breve descripción de cada uno.

Tabla 2.4 Tipos de Datos de Java

KEYWORD	DESCRIPTION	SIZE/FORMAT
BYTE	BYTE-LENGTH-INTEGER	8BIT
SHORT	SHORT INTEGER	16BIT
INTEGER	INTEGER	32BIT
LONG	LONG INTEGER	64BIT
FLOAT	SINGLE-PRECISION FLOATING POINT	32 BIT IEEE 754
DOUBLE	DOUBLE-PRECISION FLOATING POINT	64 BIT IEEE 754
CHAR	A SINGLE CHARACTER	16 BIT ÚNICODE CHARACTER
BOOLEANO	A BOOLEAN VALUE	TRUE O FALSE

Los arreglos, clases e interfaces son tipos de datos de referencia. El valor de una variable de tipo referencia, en contraste con el de una variable de tipo primitivo, es una referencia a (la dirección de) el valor conjunto de valores representados por la variable. Una referencia es llamada “puntero”, o una dirección de memoria en otros lenguajes de programación. Java no soporta el uso explícito de direcciones como algunos otros lenguajes. En lugar de la dirección explícita se usa sólo el nombre de la variable.

Figura2.5 Diagrama de un Objeto



NombreObjeto

Operadores: Un operador ejecuta una función sobre uno, dos o tres operandos. Los operadores que requieren un sólo operando son llamados operadores unarios. Por ejemplo, ++ es un operador unario que incrementa el valor de su operando en 1. Un operador que requiere dos operandos es un operador binario. Por ejemplo, = es un operador binario que asigna el valor del operando que se encuentra del lado derecho al operando que se encuentra del lado izquierdo. Por último, un operador ternario es aquel que requiere, para ejecutar su función de tres operandos. Java tiene un operador ternario, ?:, el cual es una abreviación de la sentencia IF-ELSE.

Los operadores en Java se dividen en varias categorías:

- Operadores aritméticos.
- Operadores relacionales y condicionales.
- Operadores de cambio y lógicos.
- Operadores de asignación.
- Otros operadores.

Expresiones, Sentencias y Bloques:

Las variables y operadores son bloques de construcción básicos en los programas. Se combinan literales, variables y operadores para formar expresiones (segmentos de código que realizan cálculos y regresan valores). Ciertas expresiones pueden estar incluidas en sentencias (unidades completas de ejecución). Agrupando un conjunto de sentencias entre {llaves}, se crean bloques de código.

-Expresiones

Una expresión es una serie de variables, operadores y llamadas métodos (construidos de acuerdo a la sintaxis de lenguaje) que se evalúan para determinar un valor.

Las expresiones realizan el trabajo de un programa. Entre otras cosas, las expresiones son usadas para computar y asignar valores a variables y para ayudar a controlar el flujo de ejecución de un programa.

Tabla 2.6 Tipos de Expresiones de Java

Expresión	Acción	Valor regresado
<code>aChar='S'</code>	asigna el valor de 'S' a la variable de tipo carácter <code>aChar</code>	El valor de <code>aChar</code> después de asignarle ('S').
<code>"The largest byte value is" + largestByte</code>	Concatena la cadena "The largest byte value is " y el valor de <code>largestByte</code> convertido a cadenzas	La cadena resultante "The largest byte value is 127"
<code>Character.isUpperCase(aChar)</code>	Llamada al método <code>isUpperCase</code>	El valor que retorna el método es: <code>true</code>

-Sentencias

Las sentencias son equivalentes a las sentencias en el lenguaje natural. Una sentencia conforma una unidad de ejecución. Los siguientes tipos de expresiones pueden ser creadas dentro de sentencias, terminando cada expresión con un (;).

-Expresión de asignación

-El uso de ++ o –

-Llama a métodos

-Creación de objetos.

Éste tipo de sentencias son llamadas sentencias de expresión.

Adicionalmente a estos tipos de sentencias de expresión, existen otros dos tipos de sentencias.

Sentencias de declaración: Son las que regulan el orden con el cual las sentencias son ejecutadas. Cuando se escribe un programa, se escriben sentencias dentro de un archivo; sin las sentencias de control de flujo, el intérprete ejecuta éstas sentencias en el orden en que aparecen dentro del archivo de izquierda a derecha de arriba abajo. De manera que se utilizan las sentencias de control de flujo para ejecutar condicionalmente las sentencias, para ejecutar repetidamente un bloque de sentencias y para cambiar de cualquier otra manera la secuencia normal de flujo de control. Por ejemplo, en el siguiente código, la sentencia `if` ejecuta condicionalmente la sentencia `System.out.println` que se encuentra dentro de los paréntesis, basándose en el valor que retorna `Character.isUpperCase(char)`:

```
char c;  
if(Character.isUpperCase(aChar))  
{  
    System.out.println("The character" + aChar + "is upper case");  
}
```

Java provee varias sentencias de control de flujo, las cuales se muestran en la siguiente tabla:

Tabla 2.7 Palabras Reservadas

Tipo de Sentencia	Palabra Clave
Ciclos	While, do-while,for
Toma de decisión	If-else,switch-case
Manejo de excepciones	Try-catch-finally, throw
Bifurcación	Break, continue, label:, return

-Bloques

Un bloque es un grupo de cero o más sentencias escritas dentro de (llaves); que puede ser usado en cualquier parte donde una sola sentencia es permitida. Las siguientes líneas de código muestran dos bloques con una sola sentencia cada uno:

```

if(Character.isUpperCase(aChar))
{
    System.out.println("The character" + aChar + "is upper case");
}
else
{
    System.out.println("The character" + aChar + "is lower case");
}
    
```

Programación orientada a objetos

Java es un lenguaje orientado a objetos, por lo cual es importante conocer los conceptos de la programación orientada a objetos. Algunos de estos conceptos son

-objeto -clase -herencia -interfase

-Objeto

Un objeto es un conjunto de software que contiene variables y métodos relacionados. Si se echa un vistazo en el mundo que nos rodea se pueden ver muchos ejemplos del mundo real: un perro, un escritorio, un set de televisión, una bicicleta, etc.

Estos objetos del mundo real comparten dos características: Tiene un estado y un comportamiento. Por ejemplo, los perros tienen un estado (nombre, color, raza) y un comportamiento (ladra, trae los juguetes y mueve la cola). Los objetos de software son modelados a partir de los objetos del mundo real, que al igual que estos tienen un estado y un comportamiento.

Un objeto de software mantiene su estado en una o más variables e implementa su comportamiento con métodos (funciones o subrutinas asociadas con el objeto).

-Clase

En el mundo real se pueden encontrar varios objetos del mismo tipo. Por ejemplo, una bicicleta una de muchas bicicletas que hay en el mundo. Cuando se usa la tecnología orientada a objetos, se dice que el objeto bicicleta es una instancia de la clase de objetos conocidos como bicicletas.

En el software orientado a objetos, es posible tener varios objetos del mismo tipo que compartan características. Así como los fabricantes de bicicletas toman ventaja de que las bicicletas comparten características, y construyen muchas bicicletas del mismo molde, así mismo se puede tomar ventaja del hecho de que los objetos del mismo tipo son similares y se pueden crear moldes para esos objetos. Un molde de software para objetos es llamado una clase. En la clase se definen las variables y los métodos comunes a todos los objetos de ese tipo.

-Herencia

En general los objetos se definen en términos de clases. Se puede saber mucho de un objeto conociendo su clase. Los sistemas orientados a objetos toman esto y lo llevan un paso más allá permitiendo a clases ser definidas en términos de otras clases. Es decir, se pueden definir subclases a partir de una clase llamada superclase. Cada subclase hereda el estado (en forma de declaraciones de variables) de la superclase y también cada subclase hereda los métodos de la superclase.

-Interfase

Una interfase es un componente (de software), el cual, objetos que no están relacionados a través de una jerarquía de clases, pueden usar para interactuar entre ellos. Un objeto puede implementar múltiples interfases.

Se utiliza una interfaz para definir un protocolo de comportamiento que puede ser implementado por cualquier clase en cualquier parte de la jerarquía de clases. Las interfases son muy útiles para:

@Capturar similitudes entre clases no relacionadas sin forzar a una relación artificial de clases

@Declarar métodos que una clase o más clases están esperando implementar

@Revelar la interfase de programación de un objeto sin revelar su clase.

Interfases Graficas de Usuario (GUI's)

Una interfaz grafica presenta un mecanismo amigable al usuario para interactuar con un programa, ya que éste último le proporciona una “apariencia visual” única. Al proporcionar distintos programas en los que los componentes de la interfaz de usuario sean consistentes e intuitivos, los usuarios pueden familiarizarse con un programa incluso antes de utilizarlo. Esto, a su vez reduce el tiempo que requieren para aprender a usar un programa y aumenta su habilidad para usar el programa de manera productiva.

Las GUI'S se crean a partir de componentes de la GUI (conocidos también como controles o widgets (accesorios de ventana). Un componente de la GUI es un objeto con el cual interactúa el usuario mediante el ratón, el teclado u otra forma de entrada. Como el reconocimiento de voz. Algunos de los componentes de la GUI son los siguientes:

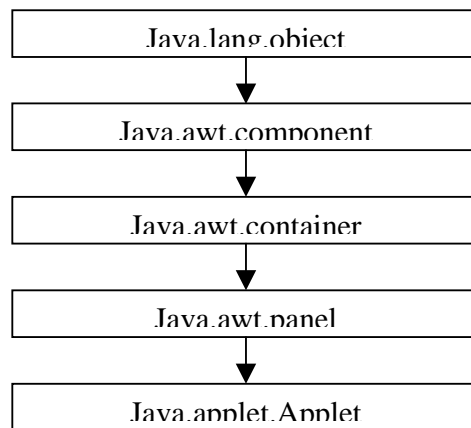
Jlabel	Áreas en donde pueden mostrarse iconos o texto no editable
JTextField	Área en la que el usuario introduce datos desde el teclado. En ésta área también puede mostrar información.
Jbutton	Área que, cuando el ratón hace clic sobre ella, desencadena un evento.
JcheckBox	Componente que puede estar o no seleccionado.
JcomboBox	Lista desplegable de elementos, de los cuales el usuario puede seleccionar uno haciendo clic en él posiblemente escribiendo dentro del cuadro.
Jlist	Área que contiene una lista de elementos, de los cuales el usuario puede seleccionar cualquiera haciendo clic en él. Pueden seleccionarse varios elementos.
Jpanel	Contenedor en el cual pueden colocarse y organizarse componentes.

Las clases crean componentes de la GUI del listado anterior, son algunos de los componentes de la GUI de SWING del paquete javax.swing. Estos componentes se hicieron estándar en Java cuando se liberó la plataforma Java 2 versión 1.2. La mayoría de los componentes de Swing, como se llama comúnmente, están escritos, se manipulan y muestran, completamente en Java (denominados (Java Foundation Clases), la bibliotecas de Java para el desarrollo de GUI'S para múltiples plataformas.

Applets

Cada applet se implementa creando una subclase de la clase Applet. La siguiente figura muestra la jerarquía de herencia de la clase Applet. Ésta jerarquía determina mucho de lo que un applet puede hacer y como lo hace.

Figura 2.8 Estructura Jerárquica de un Applet



Un applet es una clase extendida de la clase Applet que se incrusta en un documento HTML y se ejecuta en un Web Browser como Internet Explorer o Netscape. Cuando el Browser carga una pagina contiene un applet, éste se descarga en el Browser y comienza a ejecutarse. Esto permite crear programas que cualquier usuario puede ejecutar con sólo cargar la pagina Web correspondiente en su Browser.

El Browser o navegador que ejecuta un applet se conoce en términos genéricos como el contenedor de applets.

2.2 ASP.NET (VISUAL BÁSIC)

¿Que son los ASP.NET?

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por [Microsoft](#). Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de [Microsoft](#) y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

Cualquier persona que éste familiarizada con el desarrollo de aplicaciones Web sabrá que el desarrollo Web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación Web es una mezcla de varios lenguajes de etiquetas, un gran uso de scripting y plataformas de servidor. Desafortunadamente para el programador de nivel intermedio, el conocimiento y habilidades que se necesitan para desarrollar aplicaciones WEB tienen muy poco en común con las que son necesarias en el desarrollo tradicional de aplicaciones.

Requerimientos de Desarrollo

Para desarrollar aplicaciones ASP.NET, debemos contar con lo siguiente:

- .NET Framework 2.0.
- Un entorno de creación de código (Visual Studio .net, Visual Studio 2005).
- Un servidor Web como IIS (Servicios de Internet Information Server) y permisos para guardar archivos en el servidor Web.

.Net Framework

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones como podemos ver en la figura 4.9.1. Basado en ésta plataforma, Microsoft intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems.

Figura 2.9 Conformación del .Net Framework



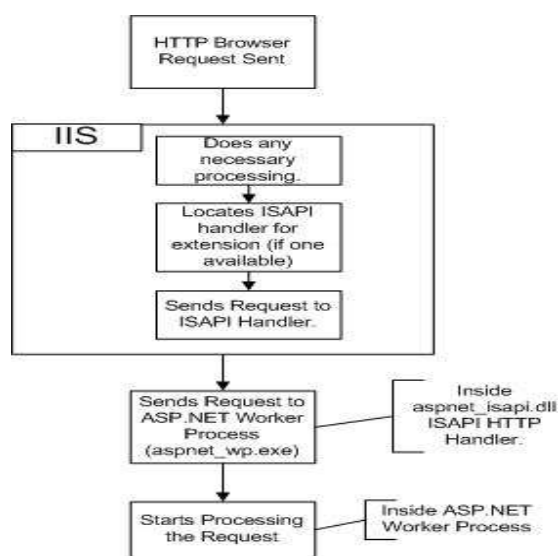
INTERNET INFORMATION SERVICES (IIS)

Es una serie de servicios para las computadoras que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Éste servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen éste servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web se basa en varios módulos como podemos ver en la figura 4.9.2 que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Figura 2.10 Estructura de IIS

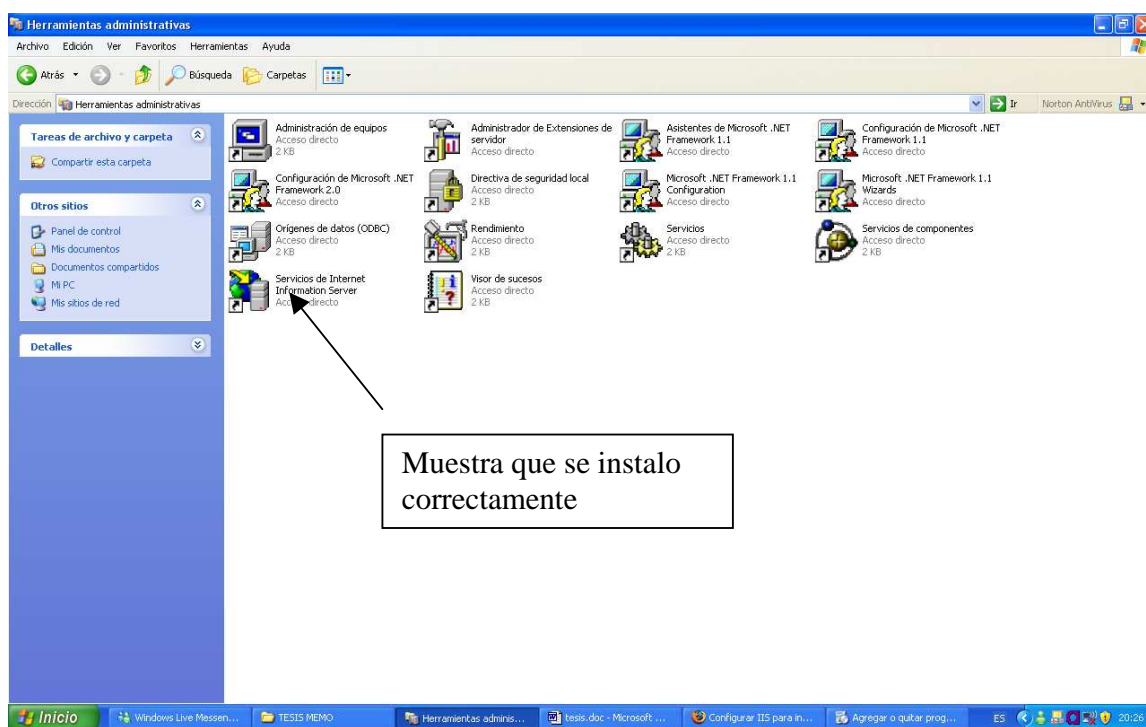


-Configuración de Servidor WEB (IIS)

Dentro de Windows uno debe dirigirse a Inicio→Panel de Control → Agregar o Quitar Software, aquí debemos seleccionar el botón que se llama Agregar o quitar programas de Windows se debe seleccionar la casilla que se llama Servicios de Internet Information Ser-

ver (IIS), una vez ahí se debe de contar con el disco de INSTALACIÓN de Windows versión Profesional y se reiniciara el equipo con el servicio de Internet arriba esto lo podemos corroborar si se instalo en la ruta Inicio → Windows → PaneldeControl → Herramientas administrativas el icono llamado como se ve en la figura 2.11.

Figura 2.11 Icono de Referencia de IIS en Windows XP

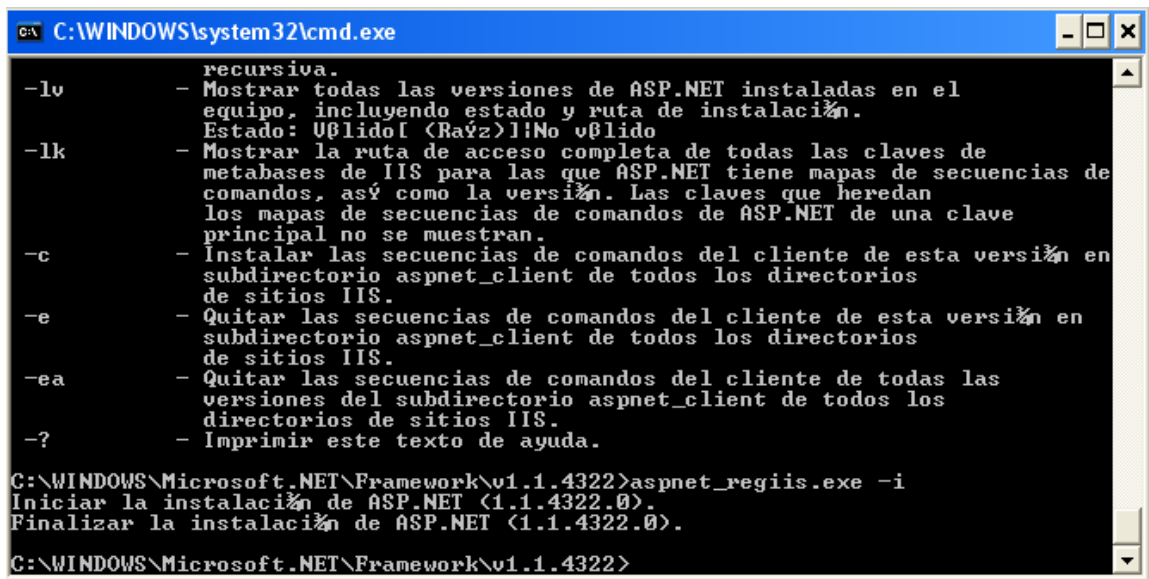


Cuando se instala el servidor Web IIS se crea un directorio llamado INETPUB y dentro de éste hay una carpeta llamada WWWROOT aquí dentro de ésta carpeta se deben de ingresar los proyectos que se hayan desarrollado en ASP, y para que estos puedan ser ejecutados por el Interprete de los ASP.

-Configuración de .NET Framework con IIS

Una vez instalado el servidor Web (IIS) se necesita instalar el Framework éste se puede conseguir en el sitio de Microsoft y se puede bajar de manera gratuita una vez de que se descargó la herramienta ésta cuenta con un sencillo instalador que realizara todo el procedimiento adecuado para la configuración del mismo en el equipo, después de la INSTALACIÓN éste pedirá reiniciar el equipo, una vez reiniciado el equipo, a Inicio → Ejecutar y ahí teclaea el comando cmd, el comando cmd devolverá una ventana de MS-DOS ahí se debe de teclear el comando cd con la siguiente ruta C:\WINDOWS\Microsoft.NET\Framework\v1.1.XXX , ahí debemos ejecutar la aplicación aspnet_regiis.exe como se ve a continuación en la figura

Figura 2.12 Alta de Servicio con Integración de ASP.NET



```
C:\WINDOWS\system32\cmd.exe
recursiva.
-lv      - Mostrar todas las versiones de ASP.NET instaladas en el
         equipo, incluyendo estado y ruta de instalación.
         Estado: v0lido| (RaVz)|!No v0lido
-lk      - Mostrar la ruta de acceso completa de todas las claves de
         metabases de IIS para las que ASP.NET tiene mapas de secuencias de
         comandos, así como la versión. Las claves que heredan
         los mapas de secuencias de comandos de ASP.NET de una clave
         principal no se muestran.
-c       - Instalar las secuencias de comandos del cliente de esta versión en
         subdirectorio aspnet_client de todos los directorios
         de sitios IIS.
-e       - Quitar las secuencias de comandos del cliente de esta versión en
         subdirectorio aspnet_client de todos los directorios
         de sitios IIS.
-ea      - Quitar las secuencias de comandos del cliente de todas las
         versiones del subdirectorio aspnet_client de todos los
         directorios de sitios IIS.
-?       - Imprimir este texto de ayuda.

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322>aspnet_regiis.exe -i
Iniciar la instalación de ASP.NET (1.1.4322.0).
Finalizar la instalación de ASP.NET (1.1.4322.0).
C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322>
```

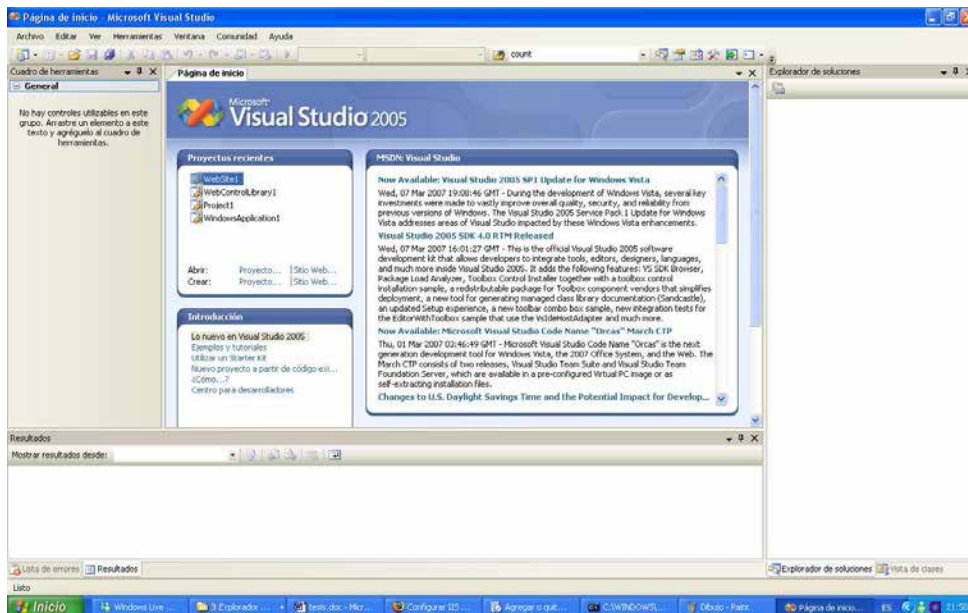
Herramientas de Desarrollo

Existen infinidad de herramientas de desarrollo de las cuales destacan 2 para poder desarrollar con ASP.NET éstas 2 herramientas son Microsoft ASP.NET WEB MATRIX y Microsoft Visual Studio .NET ó Visual Studio 2005 en éste caso usaremos la herramienta de Visual Studio 2005 ya que cuenta con herramientas de desarrollo muy sofisticadas y productivas para el desarrollo eficaz y rápido.

-Creación de un Proyecto en Visual 2005

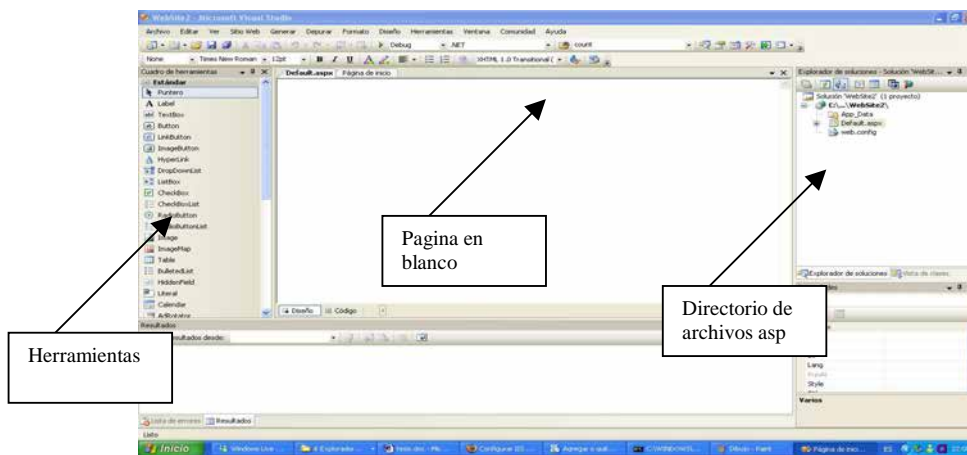
Una vez instalado el ambiente de desarrollo se debe de ingresar a Inicio → Todos los programas → Visual Studio 2005 → Visual Studio 2005, una vez ingresado ahí la aplicación mostrara la pantalla como la de la figura 4.9.5

Figura 2.13 Entorno de Desarrollo Visual Studio 2005



En ésta pantalla se debe dirigir a Archivo → Nuevo → Sitio Web, una vez ahí se debe dirigir y seleccionar Sitio WEB ASP:NET, saldrá una pantalla como la mostrada en la cual se puede ver toda la serie de herramientas de desarrollo con las que cuenta la aplicación y una pagina en blanco a la cual le agregaremos todas las herramientas que estamos viendo en éste momento.

Figura 2.12 Entorno de desarrollo de ASP.NET



Code-behind

Code behind es un una nueva metodología de programación en la cual se escribe el código fuente atrás de la vista de la forma que se ves como el código HTML éste fue diseñado intentando asimilar las nuevos conceptos de programación del modelo que estamos hablando es del modelo MVC modelo vista controlador es una nueva metodología de programación en ASP.NET que se aplica para tener diseños precisos y en la cual cada capa éste separada y no convivan íntimamente ya como otros lenguajes que existen Frameworks que tratan de simular éste tipo de diseño.

En éste caso la vista es representada con las siguiente estructura de archivo.

(nombre_de_archivo).aspx

En el caso del codebehind es representada de la siguiente manera.

(nombre_de_archivo).aspx.vb

Visual Básic. NET

Los ASP'S y en especial las tecnologías programadas bajo los entornos .NET pueden convivir con cualquier lenguaje que pertenezca a dicha tecnología alguno de estos ejemplos son C#, VB, J#. En éste caso las aplicaciones convivirán con el lenguaje VB.NET que es un lenguaje fácil de codificar y cuenta con una serie de herramientas y clases sencillas de utilizar. Esto es productivo en el momento de programar aplicaciones, también debemos saber la potencia de cada uno de los lenguajes que tiene el entorno .NET en éste caso para presentaciones pequeñas de mediano rendimiento el lenguaje Visual Básic es ideal para dichos propósitos.

Objetos de ASP.NET

Los objetos propios de los ASP.NET sirven para interactuar con el servidor Web y el navegador del cliente que ésta haciendo la petición al Web a continuación se mencionan algunos de ellos.

-Response: Éste objeto sirve para enviar información al usuario que le ésta haciendo la petición al servidor.

-Request: Éste objeto es usada para obtener información del cliente que ésta haiendo la petición al servidor

-**HttpCookie:** Éste objeto sirve para almacenar información en la memoria intermedia del navegador del cliente información que puede ser necesaria en algún momento para una nueva petición.

-**Session:** Éste objeto es para almacenar información propia del cliente que ésta utilizando la aplicación ésta información ésta disponible para todas las páginas que el usuario visite.

WebForms

En todas las aplicaciones diseñadas con ASP.NET debe de existir la etiqueta **<form>** la cual tiene también su etiqueta de término **</form>** la única característica es que ésta etiqueta contiene una propiedad llamada **runat="server"** ésta siempre debe de ser así ya que el server es el lugar donde se realizaran todas las transacciones propias de los ASP.

```
<form runat="server">  
    HTML + server controls  
</form>
```

Aquí estamos mencionando los server controls que son herramientas propias de los ASP.NET las cuales son codificadas en etiquetas propias de los ASP.NET.

WebControls

Los WebControls son etiquetas propias de los ASP.NET y que solamente son entendidas por el servidor Web que en ese momento sé éste utilizando, estos tags corren directamente en el servidor donde se encuentra instalada nuestra aplicación a continuación mostraremos un ejemplo de un Web Control.

```
<asp:Calendar id="some_id" runat="server" />
```

En éste caso éste tag lo que mostrara al instance de que sé invocado será un calendario el cual se puede seleccionar una fecha especifica y que cuenta con características específicas como podemos ver en la siguiente figura

Figura 2.13 Calendario JavaScript

≤	April 2007						≥
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	
<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	
<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	
<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	
<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	
<u>29</u>	<u>30</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	

POSTBACK

Una de las características más importantes de los ASP.NET es la facilidad se tiene para declarar controles que corren directamente en el servidor y no en la pagina del cliente, ésta es una de las características que tiene la propiedad POSTBACK ya que por medio de la entrada de datos de un usuario puede cargar dichos datos a al mismo formulario esto es por medio de la propiedad de POSTBACK y ésta información puede ser validada y cargada en la misma pagina sin necesidad de acceder a otra para que hacer la carga de datos de la URL que mando llamar.

Validation Controls

-Validacion del lado del cliente.

Una de las funcionalidades de los controles de lado del cliente es la capacidad y facilidad que tiene de manejo. Muchos desarrolladores manejan sus propias colecciones de validaciones del lado del cliente. Los controles de validación generalmente están desarrollados en JavaScript ya que es sencillo implementar las por medio de éste lenguaje del lado del cliente.

-Validacion del lado del servidor.

La validación del lado del servidor permite hacer validaciones más complejas ya que se puede interactuar directamente con datos provenientes del servidor ya que se interactúa directamente con lenguaje propio que ésta utilizando el servidor que en éste caso es el lenguaje puro de la aplicación. Se puede interactuar con mapas directamente de xml y con bases de datos o cualquier tipo de aplicación que provenga directamente del servidor.

Archivo Global.asax

Si tu agregas una nueva funcionalidad a nuestra aplicación de ASP.NET, éstas se van agregando automáticamente al archivo Global.asax. Éste archivo es usado para mantener una aplicación de ASP.NET y declarar variables globales dentro de una aplicación y manejar eventos de la misma aplicación.

Conexión a bases de datos (ADO)

ADO.NET fue introducido en la primera versión del framework desarrollado de Microsoft .NET framework, ADO fue desarrollado para mantener una aplicación conectada a una base de datos de manera sencilla y con un rendimiento superior al de las otras aplicaciones. En ésta se pueden mantener varias instancias abiertas sin que se caiga la conexión y se pueden hacer conexiones y desconexiones de datos según convenga en cada momento a la aplicación.

La clase con la cual se puede instanciar cualquier tipo de conexión a Base de datos es la clase **System.Data**.

-Como establecer una conexión

Para establecer una conexión a bases de datos se utiliza el objeto **SqlConnection** éste objeto necesita la información necesaria para leer donde se encuentra la base de datos a la cual se le va a hacer la agregación de datos o la consulta ésta información es ingresada a la clase por medio de una **Cadena de Conexión**.

Métodos

DataSource()
Database()
State()
ConnectionString()

-Como enviar consultas.

El objeto **SqlConnection** crea una conexión a la base de datos ya una vez instanciado éste objeto se pueden usar otro objeto llamado **SqlCommand** éste objeto se le puede pasar una cadena serie de sentencias propias del lenguaje SQL para ejecutar una consulta.

Métodos

CommandText()
CommandTimeout()
CommandType()
CommandText()

-Insertar, modificar y borrar datos de formularios.

Para hacer éste tipo de procedimientos en una aplicación se utilizan diferentes tipos de objetos que a continuación se mencionaran

***Data Set.**

Un objeto `DataSet` encapsula un conjunto de tablas independientemente de su procedencia y mantiene las relaciones existentes entre las tablas. El contenido de un `DataSet`

puede serializarse en formato XML. Además, se permite la modificación dinámica de los datos y metadatos del conjunto de datos representado por el objeto `DataSet`.

***Data Table**

Representa una tabla en memoria (Columns & Rows) cuyo esquema viene definido por su colección de columnas `Columns`. La integridad de los datos se conserva gracias a objetos que representan restricciones (`Constraint`) y dispone de eventos públicos que se producen al realizar operaciones sobre la tabla.

-Cierre de Conexión

Para hacer un cierre a la conexión de una base de datos una vez que se instancia una clase que haya utilizado el método de Connection String esa misma instancia se le otorga el método `close()` éste mata la instancia de la base de datos y cierra la conexión a la misma teniéndose que crear de nueva manera otra instancia para que vuelva a vivir por medio de la instancia que la manda llamar.

-Archivo Web.config

Las aplicaciones en .Net cuenta con los conocidos archivos de configuración `web.config` y `app.config`, que nos sirven para definir parte del comportamiento de nuestras aplicaciones, como por ejemplo el tipo de autenticación de usuarios a utilizar, cómo manejar los errores (redireccionarlos a una página en especial), opciones de globalización, etc.

Además de las secciones ya definidas podemos agregar opciones de configuración definidas el desarrollador para guardar valores propios de la aplicación.

El aprendizaje de estas herramientas nos proporciona, capacitación profesional eficaz ya que estas son en el momento las herramientas que más se utilizan en el mercado, Java es una herramienta muy poderosa, deciendo de lenguaje C y su principal características es la implementación de Objetos que por lo general ya están desarrollados y que cuentan con un

performance idoneo para convivir con diferentes procesos que se ejecutan como clases dentro de la maquina virtual de Java, actualmente Java cuenta con herramientas llamadas Frameworks (ambientes de trabajo). Para areas especificas como por ejemplo el Framework Struts que se implementa para desarrollo Web, Hibernate para persistencia con diferentes Bases de datos, Spring para implementar proyectos con EJB permitiendo persistencia y performance adecuando en cada una de sus aplicaciones. Utilizar estas herramientas proporciona un mejor desempeño en cuanto al performance de una aplicación de diferente ambito. Estas herramientas son eficaces para el desarrollo de herramientas de software robusto e ideal para cualquier tipo de cliente eel cual necesite realizar bastantes operaciones durante el dia por ello podemos mencionar que Java por mucho tiempo sera una herramienta en el mercado ya que al momento no hay herramiementa que superen su crecimiento. En cambio los ASP es una herramienta para desarrollos sencillos en las cuales las horas hombre sean pocas y que los desarrollos sean eficaces ASP permite la interaccion con diferentes lenguajes ya sea J#, C# y Visual Basic. En este caso el aprendizaje de esta herramienta nos proporciono la opcion de hacer buenos desarrollos en poco tiempo la diferencia seria la necesidad de los sistemas, los usuarios que se van a atender, y a partir de estos topicos podremos considerar la herramienta a utilizar.

CAPITULO III

INFORME DEL PROYECTO DE UN SISTEMA DE CONTROL DE BITACORAS

3.1 PLANTEANDO LA NECESIDAD DEL SISTEMA

Internet se desarrollo hace más de cuatro décadas y su patrocinio estuvo a cargo del Departamento de Defensa de los Estados Unidos. Ésta red fue diseñada en un principio para conectar lo sistemas de computo principales de aproximadamente una docena de Universidades y organizaciones de Investigación. Hoy en día Internet ésta siendo utilizada por cientos de millones de computadoras alrededor del mundo.

Con la introducción del World Wide Web (que permite a los usuarios localizar y ver documentos basados en multimedia, sobre casi cualquier tema a través de Internet). Internet se ha convertido explosivamente en uno de los principales mecanismos de comunicación del mundo.

Internet y la World Wide Web se encuentran sin duda, entre las creaciones más importantes de la humanidad. Años atrás la mayoría de las aplicaciones de computadora se ejecutaban en equipos que estaban conectados entre sí. Las aplicaciones en la actualidad pueden diseñarse para intercomincarse entre los cientos de millones de computadoras de todo el mundo. Internet mezcla las tecnologías de la computación y las comunicaciones. Ofrece tener información accesible de forma instantánea y conveniente en todo el mundo. Hace posible que los individuos, empresas y pequeños negocios locales obtengan una exposición mundial; con esto sé ésta cambiando la manera en que se hacen los negocios. La gente puede buscar los mejores precios de casi cualquier producto o servicio. Los investigadores pueden estar íntimamente al tanto de los descubrimientos.

En el mundo actual tanto las pequeñas, medianas y grandes empresas necesitan estar cerca de sus clientes y proveedores, en comunicación continua con ellos a través de medios eficaces y eficientes. Uno de esos medios es Internet, que a lo largo de los años de crecimiento se ha convertido en una herramienta en la cual basar las estrategias de negocio. Por esto el proyecto se enfoca un sistema Web que se encargara de ofrecer una solución flexible y eficiente para los negocios. Su función es proporcionar un esquema en el cual el sistema

de administración de bitácoras con lo cual los usuarios que necesiten consultar una historia de un vuelo o de una reservación de vuelo ésta pueda ser consultada sin ninguna restricción solamente con la restricción del tipo de usuario que éste logeado en ese momento en el sistema además de que varias personas puedan consultar al mismo tiempo una historia de vuelo o una reservación.

3.2 OBJETIVOS DEL SISTEMA

- Ofrecer una herramienta eficiente para la consulta de bitácoras para todo el personal que tenga permisos para acceder a la aplicación.
- Promover el uso de los Asp como una herramienta productiva de fácil manejo como una alternativa a soluciones pequeñas.
- Demostrar la facilidad de manejo.
- Desarrollar un sistema profesional con un máximo rendimiento.

3.3 REQUERIMIENTOS DEL SISTEMA

El sistema requiere ser implantado en un servidor Web para ser accedido por cualquier computadora a través de Internet. Se necesitara contar con una conexión de banda ancha. El equipo en el que se implementara el servidor Web tendrá las siguientes características.

- Procesador Pentium 4 a 2.0 GHZ.
- Disco duro de 500 GB de capacidad a 7200 RPM
- RAM: 2GB DDR CON ECC
- Tarjeta de Red: 10/100 Integrada a la tarjeta principal
- S.O Windows XP Profesional.

Cabe destacar que las características de éste equipo son adecuadas para pruebas durante el desarrollo y puesta en marcha. Sin embargo en éste caso de llevar el sistema a un ambiente empresarial donde el número de accesos al servidor sea mucho mayor, entonces se deberá de emplear con un equipo con mayores capacidades de almacenamiento, procesamiento y memoria principal.

El sistema deberá de contar con una Interfaz amigable para el usuario, que se desarrollara con algún lenguaje de programación VB.NET. Para almacenar los datos que provienen de los vuelos se necesitara de un manejador de base de datos.

3.4 SOLUCION DEL SISTEMA

El sistema va a estar disponible para los usuarios en un sitio Web. Los usuarios se registraran en el sistema, debiendo proporcionar un nombre de usuario (login) y contraseña (password). Esto con el fin de llevar un control de accesos al sistema. Una vez que se han registrado los usuarios pueden acceder a la parte del sistema donde podrán acceder a diferentes pantallas.

-PNR – Historia

-Búsqueda por fecha.

-Búsqueda por vuelo.

-Búsqueda por Nombre Pasajero.

-Búsqueda por Record Locutor.

-ACI – Historia

-Búsqueda por fecha.

-Búsqueda por vuelo.

-Búsqueda por Numero Pasajero.

-APIS – Historia

-Búsqueda por vuelo.

- Admón. de Usuarios
- Agregar Usuario
- Modificar Usuario
- Borrar Usuario

Aquí el usuario va a poder seleccionar cualquiera de éstas opciones y en cada una de ellas saldrá un icono de un calendario en el cual seleccionara la fecha y ahí se desplegaran los vuelos que se realizaron ese día.

Cualquier usuario del sistema puede hacer consultas de los vuelos y de la historia de cada uno de los pasajeros que realizaron el vuelo.

3.5 HERRAMIENTAS DEL SISTEMA

- Sistema Operativo Windows XP
- Servidor Web IIS
- Herramienta de Desarrollo (VISUAL STUDIO 2005)
- Manejador de Bases de Datos SQL SERVER 2000

3.6 INTEGRANDO HERRAMIENTAS

Como se ha mencionado, las herramientas que se van a emplear para desarrollar e implantar éste sistema, son herramientas de desarrollo productivo en éste caso el Sistema Operativo Windows XP permitirá acceder al sistema y manipular los recursos del hardware y las herramientas de desarrollo que se va utilizar pasando el servidor Web que permitirá acceder al sistema desde cualquier computadora a través de Internet y por el manejador de bases datos que administrará el almacenamiento de los vuelos.

Ahora bien, todas éstas herramientas han sido desarrolladas a través de los años para trabajar prácticamente juntas, de manera que la compatibilidad entre ellas es casi nata.

Aunque de cualquier forma existen distribuciones de servidores Web como Apache y MySQL que pueden llegar a ser compatibles con las aplicaciones y las herramientas de desarrollo de Visual Studio 2005, pero se ha visto que el mejor desempeño es bajo las herramientas del ambiente de Windows y el .NET Framework.

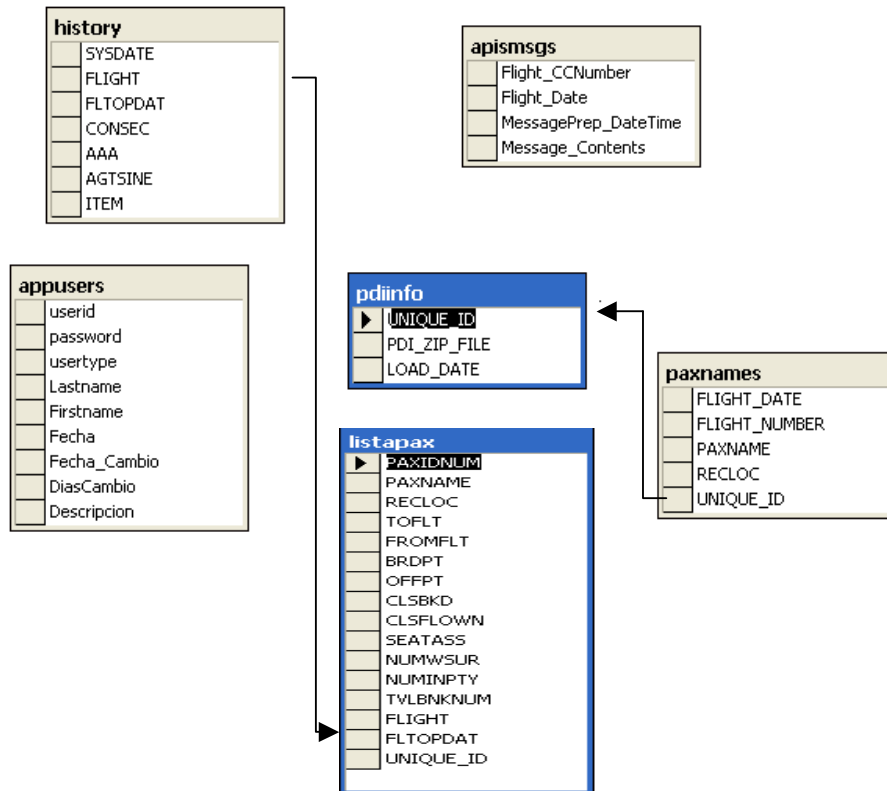
El primer paso de Integración del sistema será la instalación del sistema operativo Windows XP que va a fungir como servidor. El sistema operativo es el sistema operativo Windows XP Professional. Una vez que se halla instalado el sistema operativo se integrara con el Servidor Web IIS, el cual debe de ser instalado con soporte para aplicaciones de ASP.NET, después se instalar el servidor de Bases de Datos de SQL SERVER.

3.7 DESARROLLO DEL SISTEMA

Base de datos

La base de datos va a almacenar los datos de los vuelos como el número de la reserva, la historia del vuelo, los datos de los usuarios, y los mensajes de los vuelos interlineales (entre líneas aéreas).

Figura 3.1 Modelo físico de Base de Datos



Creación de la Base de Datos.

La base de datos se llamara PDIAM. Constará de las siguientes tablas.

-pdiinfo -appusers -paxnames -listapax -history -apismsgs

Diccionario de Datos

PAXNAMES-Table

Field Name	Datatype	Length	Key
FLIGHT_DATE	DATE		
FLIGHT_NUMBER	CHAR	4	
PAXNAME	CHAR	30	
RECLOC	CHAR	6	
UNIQUE_ID	CHAR	16	

PDIINFO-Table

Field Name	Datatype	Length	Key
UNIQUE_ID	CHAR	16	<input checked="" type="checkbox"/>
PDI_ZIP_FILE	Image		
LOAD_DATE	DATE		

HISTORY-File

Field Name	Datatype	Length	Key
SYSDATE	SMALLDATETIME	4	<input checked="" type="checkbox"/>
FLIGHT	NVARCHAR	4	<input checked="" type="checkbox"/>
FLIGHTOPDATE	SMALLDATETIME	4	
CONSEC	NVARCHAR	50	<input checked="" type="checkbox"/>
AAA	VARCHAR	3	
AGTSINE	VARCHAR	2	
HISTORYTEXT	VARCHAR	400	

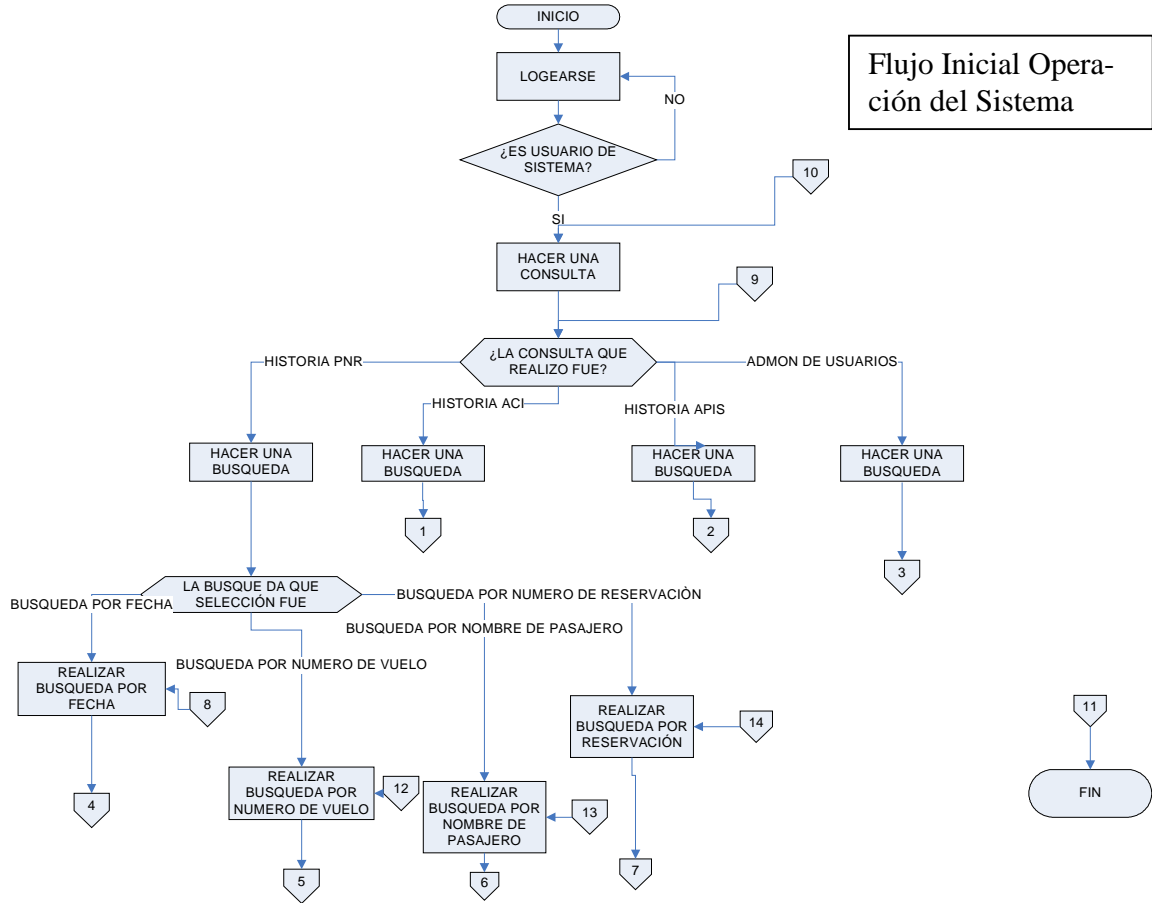
LISTAPAX - File

Field Name	Datatype	Length	Key
PAXIDNUM	VARCHAR	4	
PAXNAME	VARCHAR	30	
RECLOC	VARCHAR	6	
TO_FLIGHT	VARCHAR	6	
FROM_FLIGHT	VARCHAR	6	
BOARD_POINT	VARCHAR	3	
OFF_POINT	VARCHAR	3	
CLSSBKD	VARCHAR	2	
CLSFLOWN	VARCHAR	2	
SEATASS	VARCHAR	60	
NUMWSUR	NVARCHAR	3	
NUMPTY	NVARCHAR	3	
FQTVNUMBER	VARCHAR	18	
FLIGHT	NVARCHAR	4	
FLTOPDATE	SMALLDATETIME	4	
UNIQUE_ID	CHAR	16	

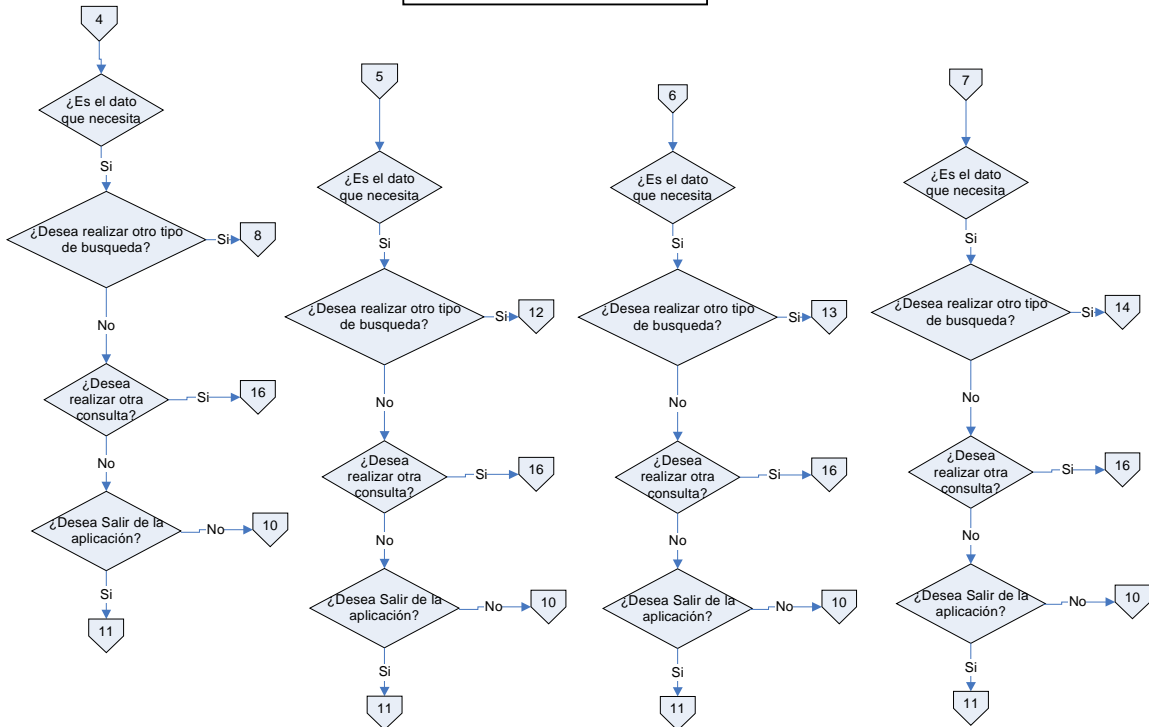
USER TYPE

Field Name	Datatype	Length	Key
USERID	DATE		
PASSWORD	CHAR	4	
USERTYPE	CHAR	30	
LASTNAME	CHAR	6	
FIRSTNAME	CHAR	16	

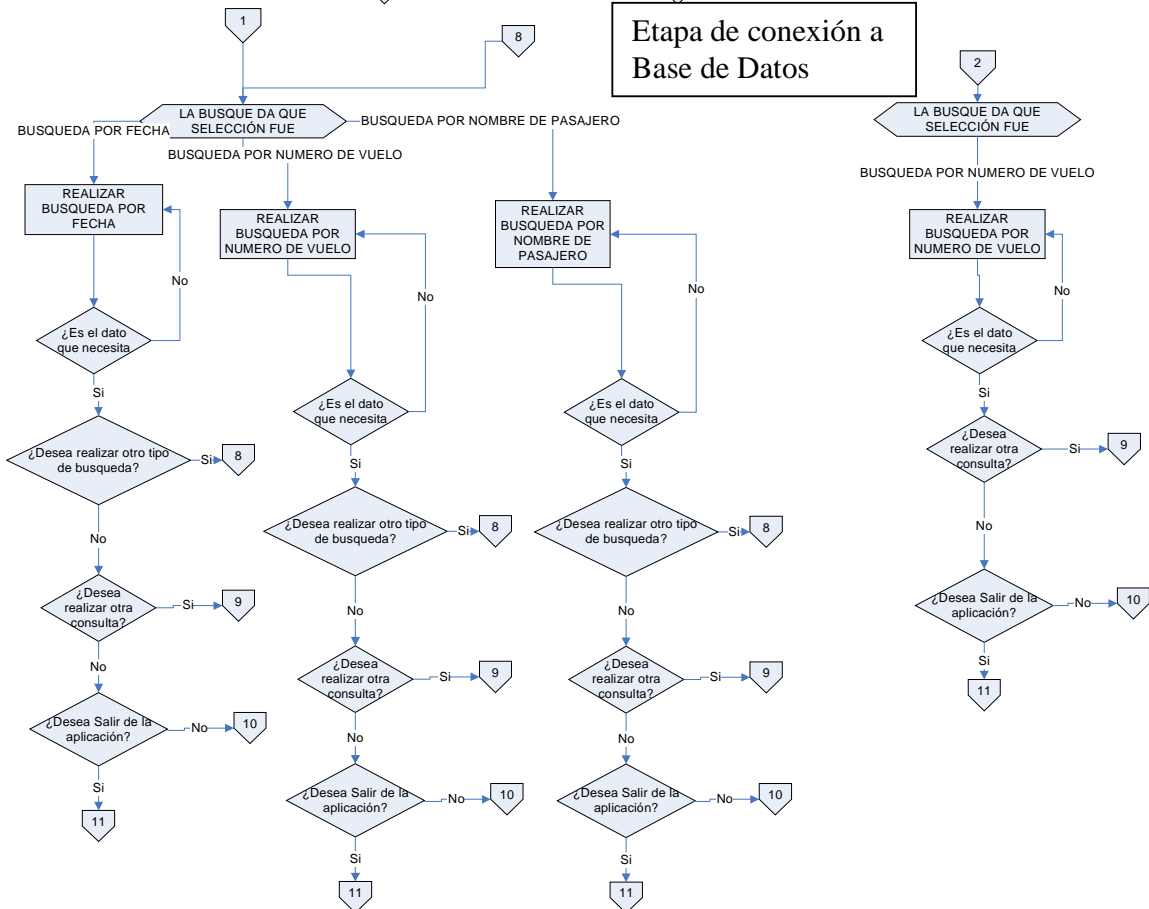
Figura 3.2 Diagrama de Flujo del Sistema PDIWEB

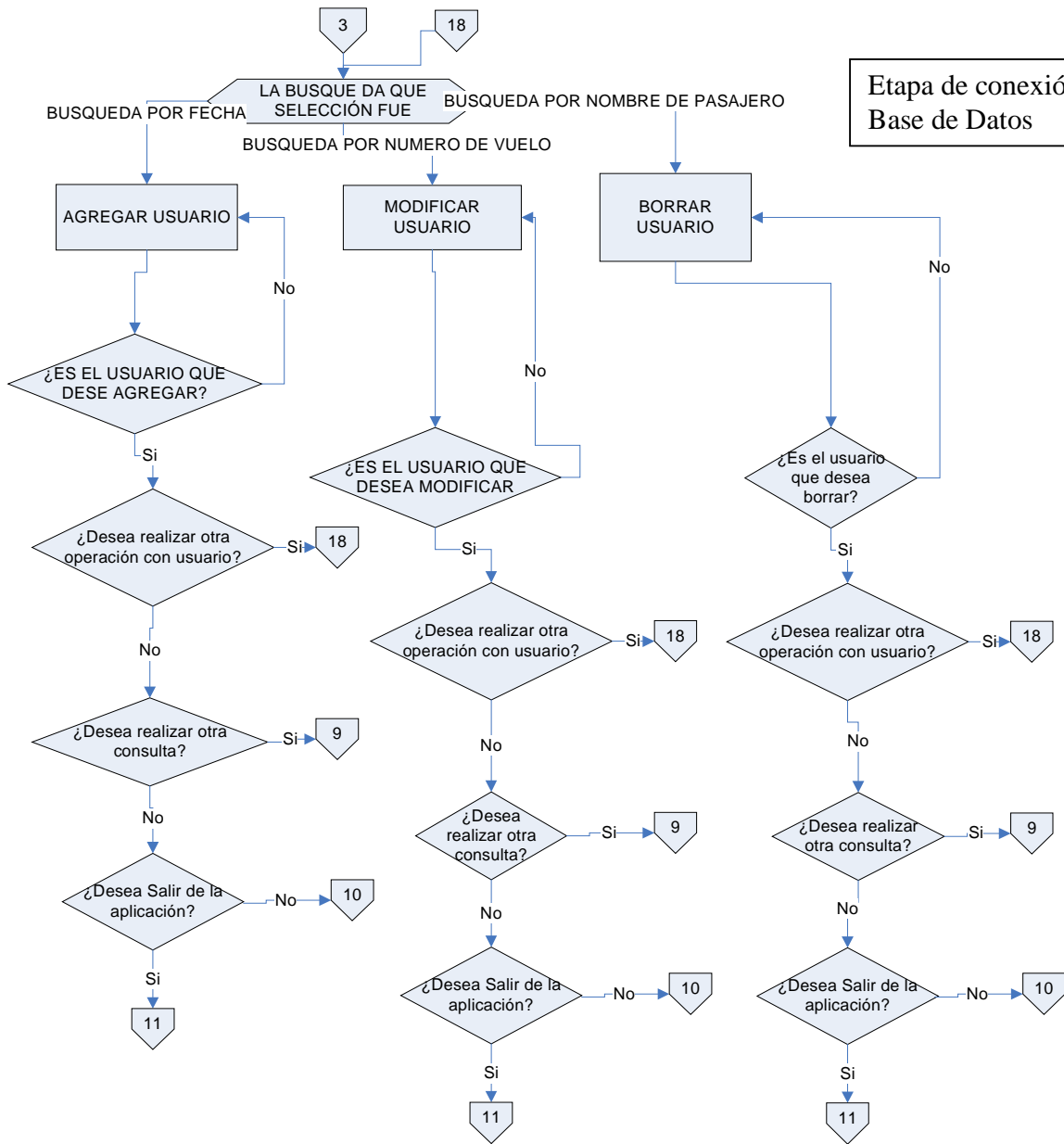


Etapa de Consultas



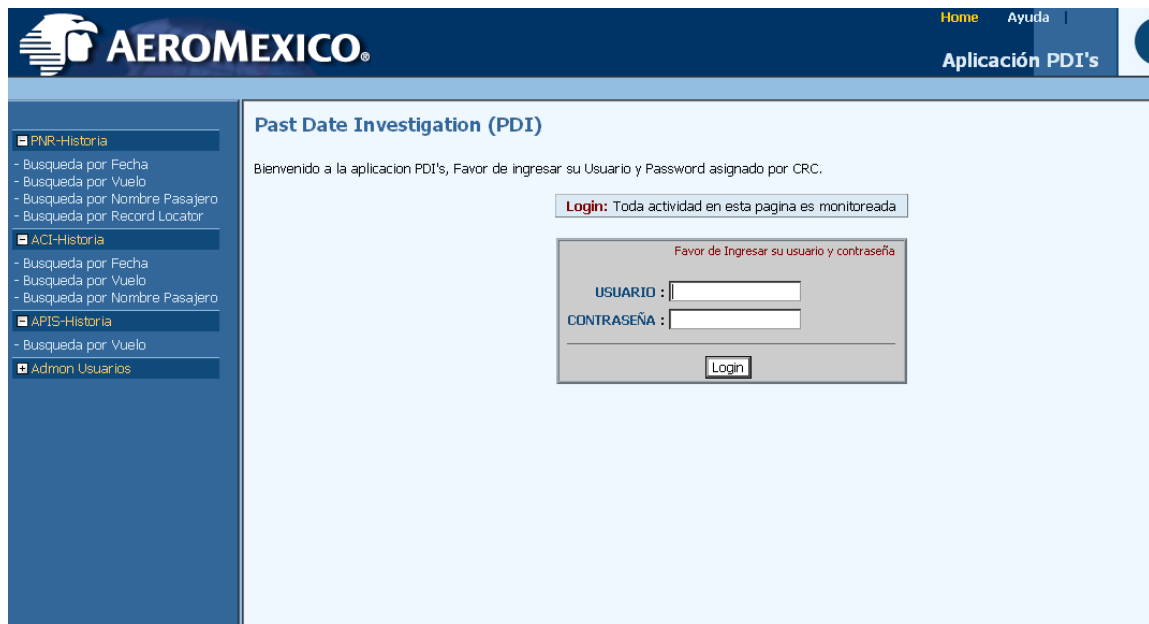
Etapa de conexión a Base de Datos





Interfaz de Usuario

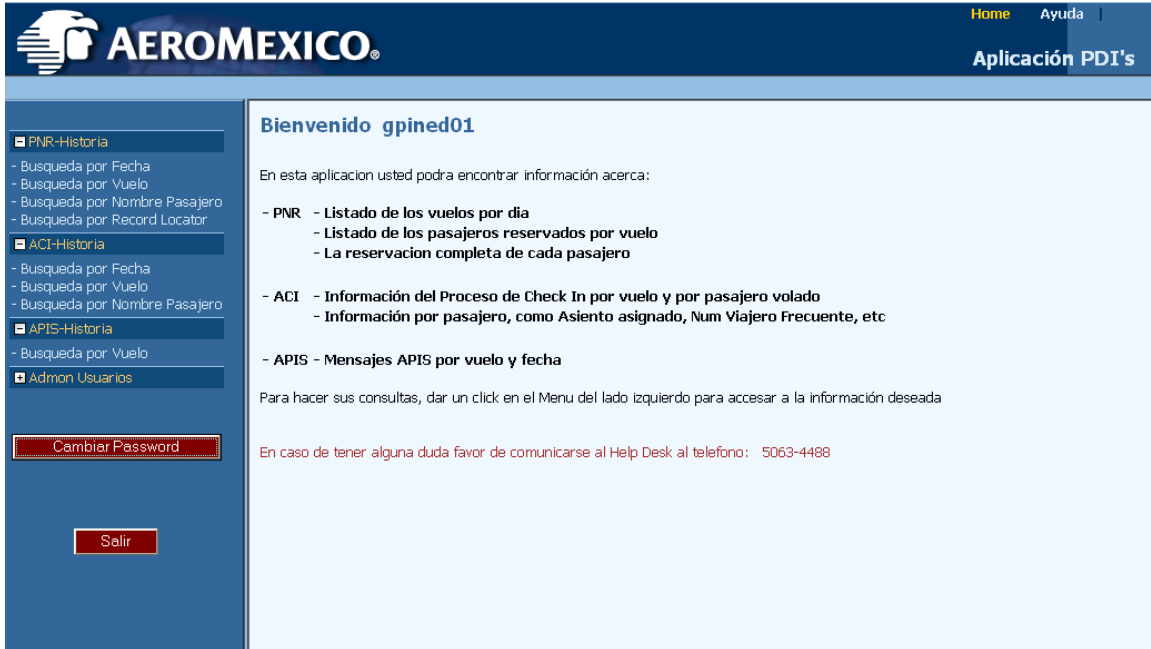
Figura 3.3 PANTALLA DE LOGIN



The screenshot displays the login page for the 'Past Date Investigation (PDI)' application. At the top, the Aeromexico logo is on the left, and navigation links for 'Home' and 'Ayuda' are on the right. Below the logo is a sidebar menu with categories: 'PNR-Historia', 'ACI-Historia', 'APIS-Historia', and 'Admon Usuarios', each with sub-options for search criteria like 'Fecha', 'Vuelo', and 'Nombre Pasajero'. The main content area is titled 'Past Date Investigation (PDI)' and contains a welcome message: 'Bienvenido a la aplicacion PDI's, Favor de ingresar su Usuario y Password asignado por CRC.' Below this is a red warning box: 'Login: Toda actividad en esta pagina es monitoreada'. The central focus is a login form with the heading 'Favor de Ingresar su usuario y contraseña'. It includes two input fields labeled 'USUARIO :' and 'CONTRASEÑA :', and a 'Login' button at the bottom.

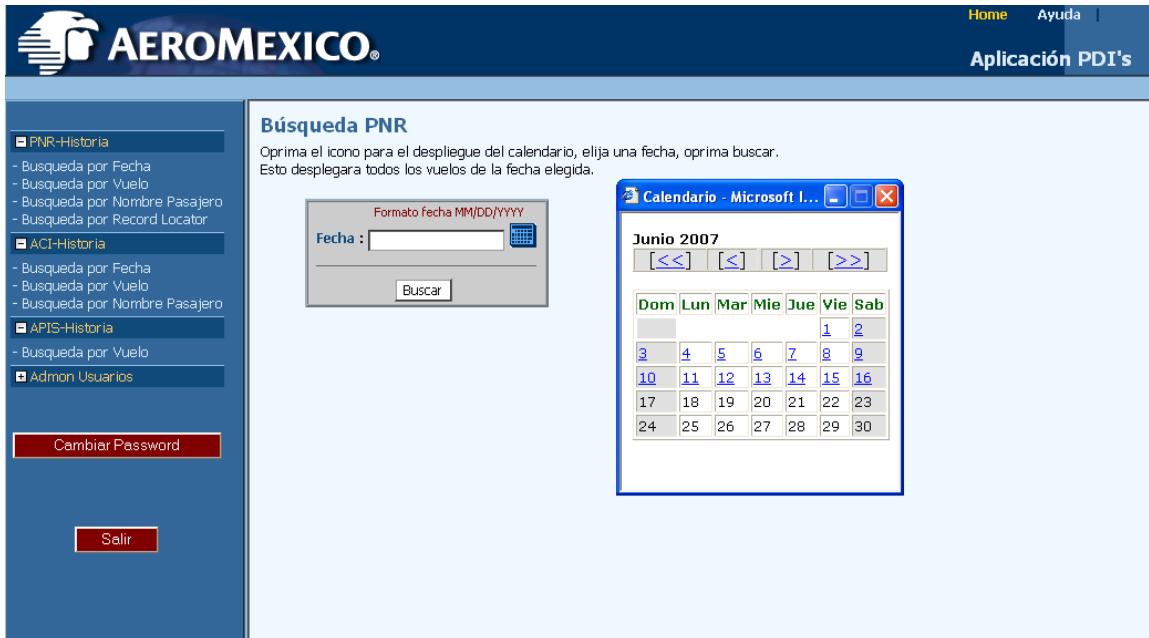
Ésta es una de las pantallas del sistema, donde los usuarios se autenticaran. En caso de que el usuario no cuente con un usuario y un password para acceder al sistema entonces el sistema regresara a ésta misma pantalla con un mensaje de que el usuario no ésta dado de alta dentro del sistema.

Figura 3.4 Pantalla de Inicio



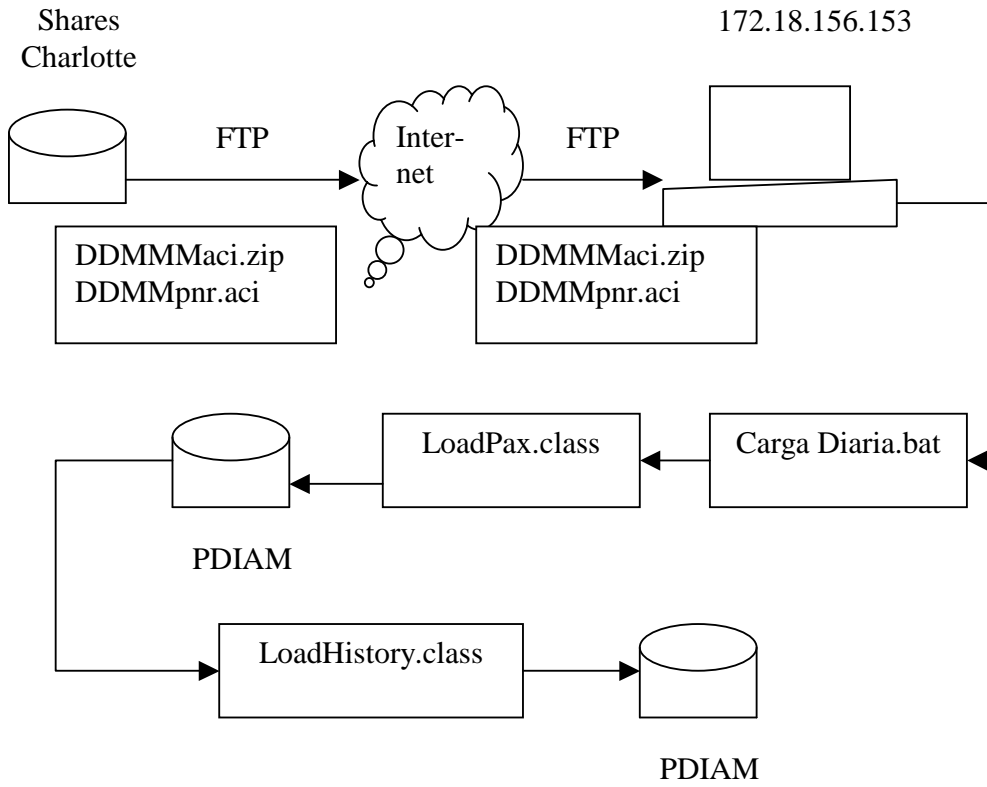
Una vez que se haya ingresado al sistema se despliegan los menús de lado izquierdo para los tipos de búsqueda que puede hacer el usuario en la aplicación.

Figura3.5 Pantalla de búsqueda de PNR por fecha



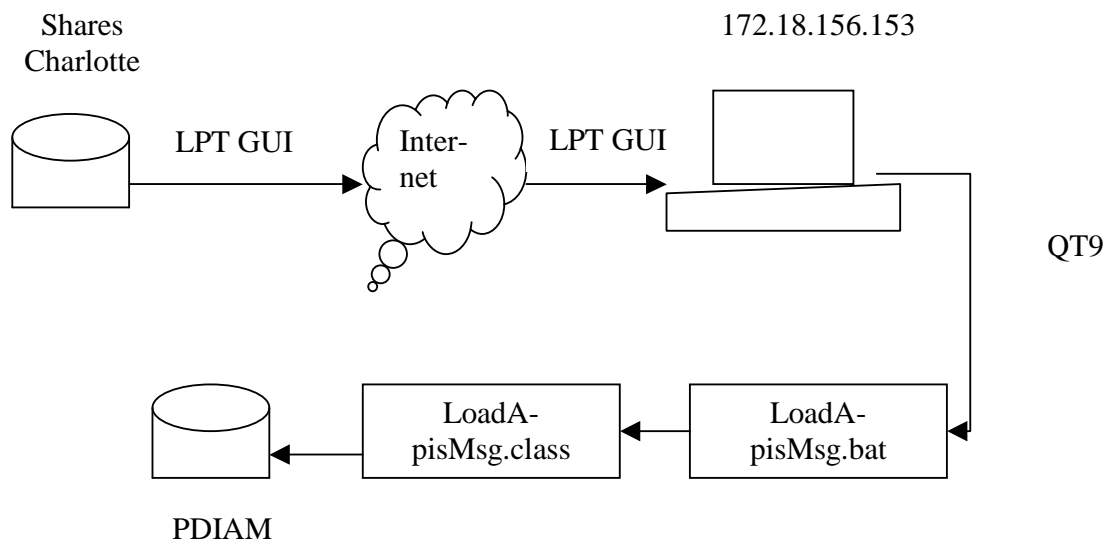
El siguiente paso es hacer el tipo de búsqueda que el usuario desea realizar y ésta se puede hacer por diferentes tópicos referentes un dato que es importante ingresar es el campo de la fecha ya que en éste campo sé los iconos que estén habilitados es la información que ésta disponible para ese día. Si ningún icono ésta habilitado es que dicho mes no cuenta con información cargada a la fecha. Para la carga de la información corren diariamente 2 Tareas que hacen el cargo de la información a nuestra base de datos de SQL a continuación veremos la manera en la cual se carga la información a nuestra aplicación en la figura

Figura 3.6 Diagrama de Operación de Shares con PDI



Ésta operación se hace diariamente desde las 3 de la mañana y el bat corre a partir de las 8:00 de la mañana para hacer la carga hacia la base de datos.

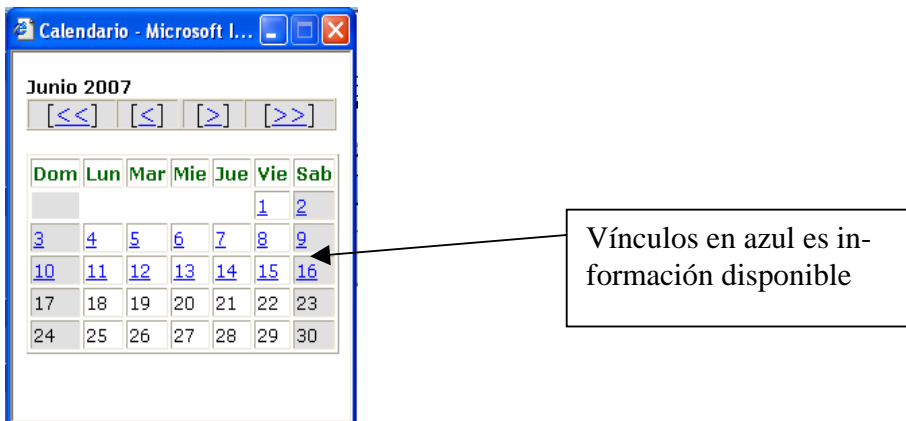
Figura 3.7 Diagrama de Operación de APIS



Aquí se hace la carga diaria ya que el archivo se está escribiendo diario y la operación termina a las 6 de la tarde todos los días.

El siguiente paso es hacer una consulta para obtener la información referente a la reservación, para adquirir dicha información necesitamos ingresar la fecha del vuelo después de ingresar la fecha obtendremos una serie de links de los vuelos disponible en la base de datos, como podemos ver en el calendario.

Figura 3.8 Calendario de PDIWEB



Una vez que el usuario ha seleccionado la fecha idónea para hacer la consulta se desplegará la información de todos los vuelos asociados a ésta fecha.

Figura 3.9 Vista de una consulta por fecha



A partir de ese momento el usuario puede seleccionar el vuelo que anda buscando y ver los pasajeros asociados a ese vuelo como se puede ver a continuación.

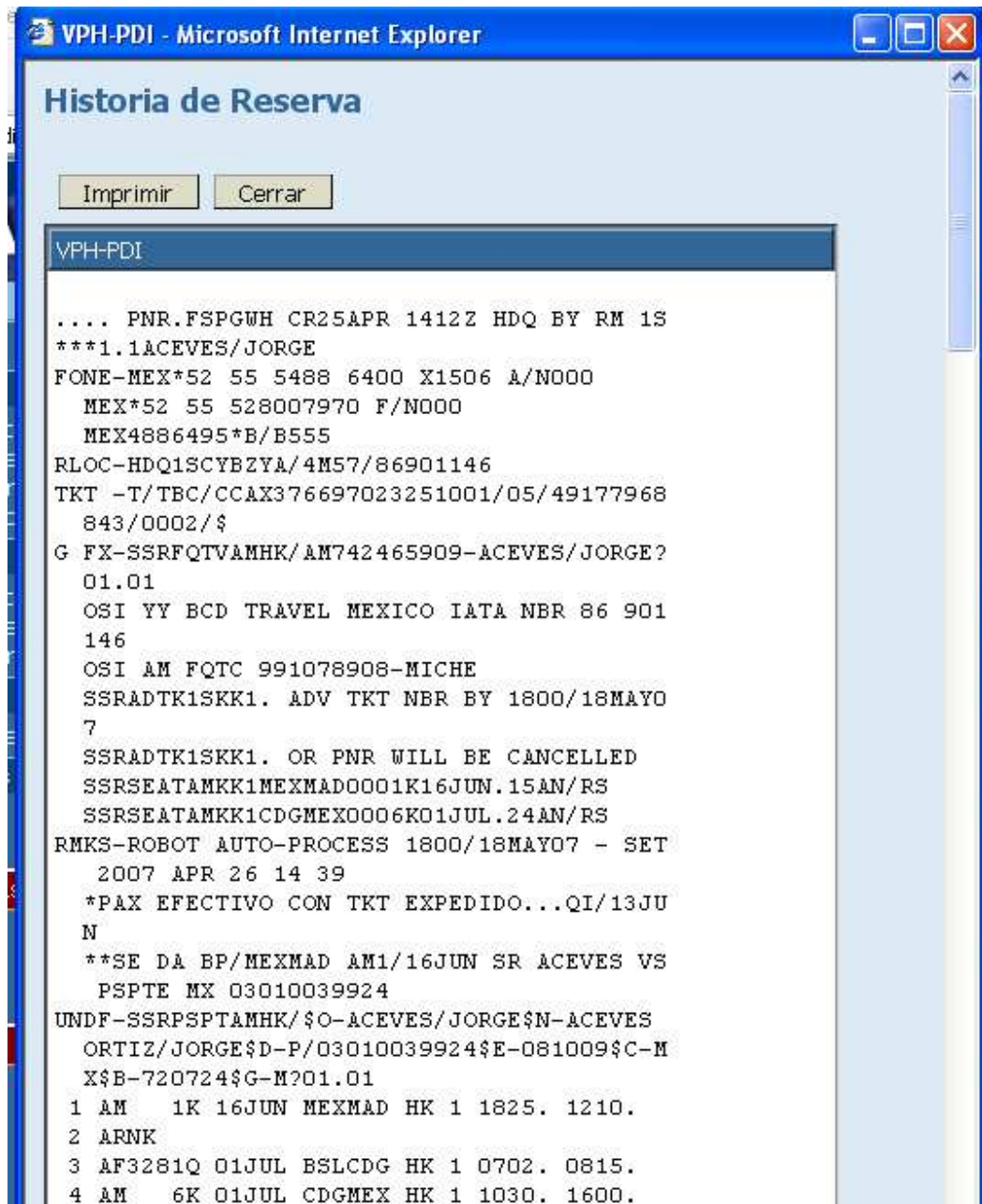
Figura 3.10 Vista de una consulta por vuelo

The screenshot shows the Aeromexico PDI application interface. At the top, there is a navigation bar with the Aeromexico logo, 'Home', 'Ayuda', and 'Aplicación PDI's'. On the left side, there is a sidebar menu with options like 'PNR-Historia', 'ACT-Historia', 'APIS-Historia', and 'Admon Usuarios'. The main content area is titled 'Lista de Pasajeros' and contains a search filter for 'vuelo : 0001 del día (mm/dd/yyyy): 06/16/2007'. Below the filter is a table of passengers.

Nombre Pasajero	Recloc
ACEVES/JORGE	FSPGWH
ACHA/JUAN/ADRIANA	CQ5GA3
ACOSTA/GEMAMRS	FYOIVI
ACOSTA/GEMAMSS	BBYB0R
AGUILAMONTIEL/MIGUELMR	GIOW3X
AGUILAR/GLORIAIRENE	FXCKA3
AGUILAR/SELADENISSE	FXCKA3
AGUIRRE/MONICAMS	GL0B0L
AHUMADA/MAEUGENIA	EX2DJT
ALARCON/MARIADOLORES	CERCUB
ALBA/ALEJANDRA	GHL1ZP
ALBA/MARIA	GHL1ZP
ALBAVOCES/JUAN	EAQVL2
ALCALA/CESARMR	FH3NE5
ALDAPA/VICTOR	FOTN2C

En ese momento se puede desplegar la información que nos interesa de cada uno de los pasajeros la cual es la historia PNR(Historia de todo el proceso de boleteado del pasajero) de cada uno de ellos al seleccionar al pasajero como se puede ver en éste caso seleccionaremos al usuario ACEVES/JORGE del cual nos desplegara su información correspondiente a partir de que el usuario inicio su proceso de cotización hasta que abordo el vuelo.

Figura 3.11 Vista de un PDI



Ésta información es interpretado por personal especializado para dar seguimiento a una aclaración por parte del pasajero y que los usuarios del sistema le otorgue la respuesta correcta con respecto a la historia que de despliega en ésta paliación.

La interfaz de usuario se apoya de la programación con ASP.NET que comprende varios archivos. Algunos de estos son **default.aspx**, **searchInput.aspx**, **contents.aspx**, **welcome.aspx** y **login.aspx**.

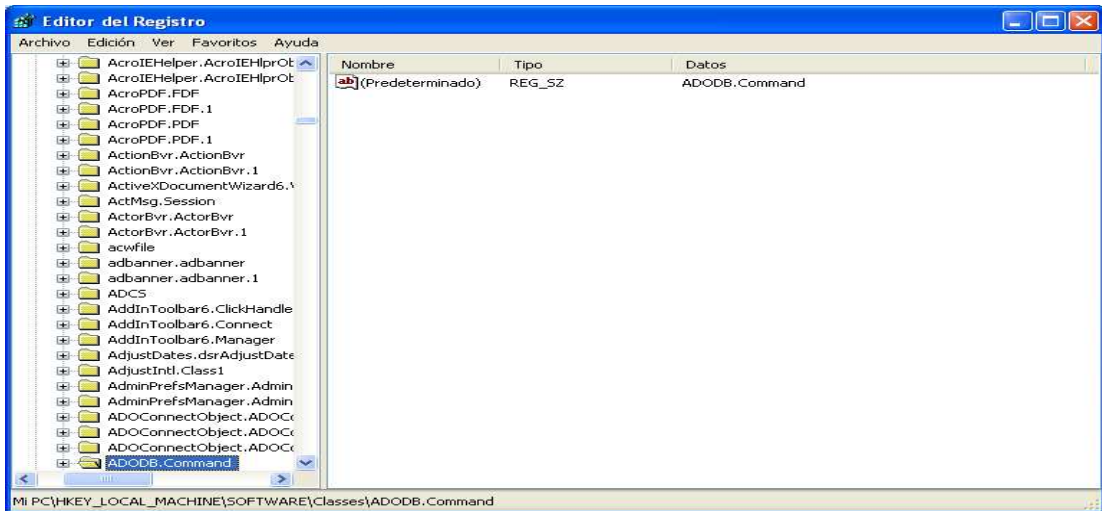
La conexión a la base de datos se realiza a través del archivo DBConnection.aspx

Figura 3.11 Código para conexión a bases de datos

```
<%  
StringConnection = "DRIVER={SQL server}; Server=172.18.156.132\PDI; Database=PDIAM; UID=pdfam; PWD=pdfam;"  
Function Connect()  
    Dim DB  
    set DB = server.createObject("ADODB.Connection")  
    DB.open StringConnection  
    Set Connect = DB  
End Function  
%>
```

Ésta depende directamente de una DLL la cual se llama ADOBD y por medio de ésta es una clase de permite hacer la escritura y consulta a la base de datos ésta debe de estar registrada en el sistema RegEdit del sistema operativo sino no funcionara la aplicación.

Figura 3.12 Registro de DLL en servidor Aplicativo



3.8 PRUEBAS DE RENDIMIENTO

Se hicieron las siguientes pruebas de rendimiento dentro de la aplicación.

Prueba	Descripción	Exitoso ó No Exitoso
Rendimiento de más de 1000 usuarios conectados al mismo tiempo.	Se hizo la prueba en la aplicación por medio de la aplicación JMeter en la cual se crean 1000 instancias hacia la aplicación la prueba era si esta puede soportar a los 1000 usuarios conectados al mismo tiempo	Exitoso
Pruebas Particulares	Se realizaron pruebas en la funcionalidad de cada una de las pantallas de la aplicación de determinandose que su funcionalidad es la adecuada para el usuario	Exitoso
Pruebas con Clientes	Se realizaron la pruebas de funcionalidad con el cliente determinandose que la aplicación funcionamiento con cliente determinando este que la aplicación queda funcionando correctamente y dispuesto a lo planeado	Exitoso
Pruebas en producción	Se deja trabajando la aplicación en producción y se monitorea durante 15 en los cuales solamente se reportaron problemas de procedimiento equivocados por parte de los usuarios se les otorgo el procedimiento correcto para acceder a las aplicaciones	Exitoso

CONCLUSIONES

Analizados los temas de los tres capítulos anteriores se puede concluir lo siguiente.

Las herramientas de software libre se encuentran a la par en características, funcionalidad uso e implementación de aquellas propietarias, esto es, por lo que se debe pagar un costo por su licencia e implementación.

Al igual que con herramientas propietarias se desarrollan sistemas robustos y eficientes, con las herramientas libres también se pueden desarrollar con la mismas características en beneficio de la seguridad, confiabilidad, escalabilidad, compatibilidad y proyección

Mientras siga habiendo un impulso continuo al movimiento de Software Libre, habrá mayor confianza en el uso de éstas tecnologías, ya que la continuidad de éste movimiento con lleva implícitamente al mejoramiento de éstas herramientas.

Es importante que en la UNAM se impartan cursos y diplomados como aquel que se basa el presente trabajo, ya que al difundirse el uso de nuevas tecnologías se asegura que la UNAM se encuentre a la vanguardia en cuanto a conocimiento, técnica e investigación.

En especial los que van ecaminados al desarrollo de aplicaciones ya que es el mercado laboral en el que vivimos se necesitan mas desarrollos de nuevas tecnologías o soporte tecnologico a herramientas que se encuentran productivas, a continuación veremos una tabla de la comparación entre diferentes entornos de desarrollo.

SOFTWARE PROPIETARIO

SOFTWARE LIBRE

Ventajas	Desventajas	Ventajas	Desventajas
<p>-Las compañías de software propietario destinan numerosos recursos para investigar en su aplicabilidad, disponen de departamentos para el control de su calidad, además de tener programadores expertos.</p> <p>- Existen muchas publicaciones que facilitan el uso de las nuevas tecnologías.</p> <p>- La seguridad que ofrece el software propietario es mucho mayor, por lo que la posibilidad de que puedan insertarte un virus es infinitamente menor que en el caso de los open source.</p>	<ul style="list-style-type: none"> - Total dependencia hacia proveedor por parte del usuario, ya que es el único que dispone del código fuente. - El proveedor no garantiza su aplicabilidad para ningún fin determinado, por lo que no se responsabiliza de los posibles defectos o daños que contenga el programa. - El único soporte que facilitan son instrucciones de configuración, uso e instalación. - Tienes que pagar un precio para adquirirlo, lo que le hace menos atractivo. - Es difícil aprender a utilizar el software propietario 	<ul style="list-style-type: none"> - Puedes hacer uso del programa con total libertad. - Podemos modificarlo y adaptarlo a nuestras necesidades. - No se necesita grandes inversiones para su distribución y promoción. - Si el producto es interesante técnicamente, aparecerán nuevos usuarios o empresas que contribuirán a su desarrollo. - Es gratuito 	<ul style="list-style-type: none"> -Lleva asociado un enorme riesgo de virus. -Una vez haber hablado un poco sobre estos dos nuevos modelos, me parece oportuno dar mi punto de vista. Yo creo que un software libre eficazmente desarrollado puede llegar a resultar mucho más atractivo que uno propietario, ya que lo interesante de este es que una aplicación se puede ir adaptando según las necesidades del usuario en cuestión.

El desarrollador debe aprender a decidir entre que herramientas puede sacar el mayor provecho en cuenta eficacia de desarrollo, inversion de horas de desarrollo y la eficacia de la herramienta a desarrollar, por ello este trabajo se basa en hacer la diferencia entre estas herramientas y las herramientas propietarias llamase Java, PHP. NET como lenguajes de desarrollo o como bases de Datos como MySQL, POSTGRESS, ORACLE o SQL SERVER y en cuanto sistemas operativos lo que sería Linux o Windows este es el proposito del trabajo dejar al desarrollador una experiencia mas y en cuanto a esa experiencia leida tener la mejor solucion a la tarea encomendada.

El diplomado en Desarrollo e Implementación de sistemas con Software Libre en Linux me apporto en especial los conocimientos necesarios y junto con el una profundización de cada tema me han permitido aplicarlos en el campo laboral, aunque sin duda hay aspectos del Diplomado que se pueden mejorar; el primordial de ellos es la actualización de los contenidos, en donde a la par que se vaya impartiendo él diplomando, se desarrollen uno o varios sistemas enfocados al manejo real de información para los negocios.

Para que los alumnos se vayan desarrollando mas en un ambiente productivo y real ya que en muchas ocasiones la salida al mundo real es una escalada bastante fuerte y difícil de escalar ya que si no se cuenta o no se tienen el conocimiento necesario para afrontar este tipo de problemas, puede dejar insatisfecho al alumno de haber estudiado una carrera como estas que requiere de mucho sacrificio personal pero que a la larga trae recompensas muy grandes para la vida cotidiana.

Bibliografía

Hicks, Alan. Slackware Linux Essentials, Second Edition.
Brentwood, CA, USA 2005 Ed, Slackware Linux Inc.

Saether Bakken, Stig. PHP Manual. PHP Documentation Group
Deitel Harvey y Deitel Paul. Java, Como programar 5a Edicion

México 2004 Ed. Pearson Education.
MySQL Referente Manual MySQL AB

The Java Tutorial, Sun Microsystems

Argerich Luis et al. Professional PHP4
New York N.Y. USA Ed. Apress

Laurie Ben and Laurie Peter, Apache: The Definitive Guide. Third Edition.
Sebastopol CA, USA 1999 Ed. O Reilly & Associates Inc

Kofler Michel. The Definitive Guide to MySQL, Second Edition
New York N.Y. USA 2004 Ed. Apress.

Converse, Tim; Park, Joyce; Morgan, Clark. PHP5 and MySQL Bible
Indianapolis, IN USA 2004 Ed. Wiley Publishing

Marco Bellinasso, ASP.NET 2.0 Website programming: Problem - Design – Solu-
tion, USA 2004 ,Ed. Wrox (“Programacion con ASP.NET”)

MacDonald Matthew, ASP.NET. Manual de referencia, México 2005 ,Ed. Mac-
GrawHill

Fuentes

- <http://www.linuxcordoba.org/node/73> Tema (“Administración Linux”)
- <http://tldp.org/index.html> Tema (“Administración Linux”)
- <http://es.wikipedia.org/wiki/MySQL> Tema (“Base de datos MySQL”)
- <http://www.linuxcordoba.org/node/73> Tema (“Administración Linux”)
- <http://www.mysql-hispano.org/page.php?id=2> Tema (“Base de datos MySQL”)
- http://www.mysql.com/products/what_is_mysql.html Tema (“Base de datos MySQL”)
- http://pisuegra.inf.ubu.es/lsi/Invest/Java/Tuto/II_7.htm Tema (“Programación Java”)
- <http://databases.about.com/cs/specificproducts/g/foreignkey.htm> Tema (“Bases de Datos”)
- http://www.programacion.com/bbdd/tutorial/mysql_basico/ Tema (“Base de datos MySQL”)
- <http://www.netcraft.com.au/geoffrey/postgresql/fkey.html> Tema (“Bases de Datos”)
- <http://mysql.conclase.net/curso/index.php?cap=011> Tema (“Base de datos MySQL”)
- <http://www.javaworld.com/javaqa/2002-08/01-qa-0823-appvserver.html> Tema (“Programación Java”)
- http://news.netcraft.com/archives/web_server_survey.html Tema (“Seguridad Informática”)
- <http://httpd.apache.org/docs/2.2/en/install.html#extract>
- <http://httpd.apache.org/docs/2.2/en/mod/directives.html#E> Tema (“Servidor Web Apache”)
- <http://httpd.apache.org/docs/2.2/en/vhosts/> Tema (“Servidor Web Apache”)
- <http://www.iec.csic.es/criptonicon/ssl.html> Tema (“Seguridad Informática”)
- http://www.ocm.gob.pe/portal_ongei/seguridad2_archivos/Lib0510/cap0203.htm Tema (“Seguridad Informática”)
- <http://www.php.net/docs.php> Tema (“Programación con PHP”)
- <http://www.elguille.info/NET/ASPNET/tutorialLogin/tutorialLogin.htm> Tema (“Programación con ASP.NET”)
- <http://www.w3schools.com/aspnet/default.asp> Tema (“Programación con ASP.NET”)

GLOSARIO

ACL (Access Control List).- Método para limitar el uso de recursos específicos a usuarios autorizados.

APACHE.- Servidor Web de libre distribución y es el más empleado actualmente en Internet para servir páginas.

APPLET.- Es un pequeño programa hecho en lenguaje Java que se descarga y se ejecuta en un navegador Web

CGI. (Common Gateway Interface). Ésta tecnología define un modelo de programación que puede ser implementada en múltiples lenguajes. Son programas que siguen un estándar definido, corren en el servicio recibiendo parámetros desde el cliente y su salida es enviada al navegador.

COOKIE.- Pequeño archivo de texto (ilegible si se abre con algún procesador de textos) que se almacena en el disco duro del visitante de una página Web a través de su navegador a petición del servidor de la página. La información que contiene éste archivo es recuperada por el servidor en posteriores visitas a la página.

CPU.- (Central Processing Unit). Es la unidad central de procesamiento y se encarga de realizar todos los cálculos de una computadora incluyendo el manejo de algunos dispositivos. Se le denomina comúnmente como “procesador”.

DBMS.- (DataBase Management System). Los sistemas manejadores de bases de datos son los encargados de almacenar y posteriormente permitir el acceso a los datos de forma rápida y estructurada.

DDL.- (Data Definition Language). Es un lenguaje que permite especificar el esquema de la base de datos, a través de tablas, vistas, procedimientos almacenados y triggers.

DML.- (Data Manipulation Language). Es un lenguaje que permite la consulta y manipulación de datos. Éste lenguaje que el DLL son proporcionados por el sistema manejador de bases de datos.

DoS.- (Denial of Services). Es un tipo de ataque de seguridad en computo cuyo objetivo es generar una alta demanda sobre un servicio valido con el fin de saturar la capacidad y propiciar la caída de los sistemas.

DSO (Dynamic Shared Objects). Es un mecanismo que provee una forma de construir parte de código de un programa en un formato especial para que pueda ser cargado en tiempo de ejecución en el espacio de direcciones de un programa ejecutable

FIREWALL.- Es un mecanismo que puede ser físico o lógico que examina el tráfico de una red, y en base a ciertos criterios permite o no el paso de los paquetes de información.

GNU.- (GNU No es Unix). Fue el proyecto iniciado por Richard Stallman en 1984 con el objetivo de crear un sistema operativo libre que fuera compatible con UNIX.

GPG.- Es un sistema de cifrado de código libre que viene a ser un reemplazo de PGP creado por Phil R. Zimmermann.

GUI.- (Graphical User Interface). Es el método para facilitar la interacción de los usuarios con un programa o aplicación.

HCL.- (Hardware Compatibility List). Lista en la que se encuentran los dispositivos de hardware que son compatibles con las diferentes distribuciones de Linux.

HTML.- (HyperText Markup Lenguaje). Lenguaje estándar para la creación de documentos o páginas Web. Éste lenguaje consta de etiquetas las cuales definen un comportamiento específico la hora de ser interpretados por el navegador Web.

HTTP.- (HyperText Transfer Protocol). Éste protocolo permite la transferencia de hipertexto a través de la Web

HTTPS.- Es la versión segura del protocolo http, utiliza un cifrado basado en SSL (ver SSL) para crear un canal cifrado apropiado para el tráfico de información sensible.

INTERNET.- Es el nombre que se le ha otorgado a la red mundial de redes que interconecta millones de computadoras entre sí.

IP, Dirección (Internet Protocol). Las direcciones IP son un conjunto de números que definen la identidad única de una computadora en una red. Su estructura se establece por cuatro octetos separados por puntos.

JAVA.- Plataforma de software desarrollada por Sun Microsystems, que en un principio se pensó para dispositivos domésticos, inteligentes, pero más tarde se observó su versatilidad para la Web.

JVM.- (Java Virtual Machine). La máquina virtual de Java es parte fundamental de la plataforma Java; se encarga de traducir los códigos de bytes (bytecodes) que se generan de la compilación de un programa a código que puede ser reconocido por cada plataforma de hardware en particular.

LILO.- (Linux Loader). Se encarga de arrancar el sistema operativo Linux e inclusive otros sistemas operativos que se encuentren en la misma maquina.

MINIX.- Sistema Operativo basado en Unix que fue escrito por el profesor Andrew Tanenbaum.

MYSQL.- Manejador libre de bases de datos que se ha popularizado por sus características incluyendo funcionalidad, compatibilidad y estabilidad.

OOP.-(Object Oriented Programming). La programación orientada a objetos es una metodología de diseño de software y un paradigma de programación que define un programa en términos de clases de objetos. Los objetos son entidades que combinan estado (datos) y comportamiento (procedimientos o métodos).

OPENSSL.-Es una implementación de código abierto de los protocolos SSL y TLS. El núcleo de librerías implantan las funciones básicas de criptografía y proveen varias funciones básicas de criptografía y proveen varias funciones útiles.

PHP.- (PHP: HyperText Preprocessor). Es un lenguaje de alto nivel embebido en páginas HTML que se ejecuta en el servidor.

PID.- (Process ID). Es un identificador que distingue a cada proceso que se ejecuta en un S.O. generalmente todos los sistemas operativos basados en UNIX utilizan este identificador.

PROXY. Es un intermediario entre una intranet e Internet; puede implantar criterios de seguridad así como de cache.

RUN-LEVEL.- El nivel de arranque es utilizado por el sistema operativo Linux para cargarse con determinadas características o bajo cierto modo. Los niveles de arranque van desde el nivel 0 hasta e 6.

SISTEMA OPERATIVO.- Conjunto de programas que se encargan de administrar los recursos de un sistema de cómputo; estos recursos incluyen hardware, software, comunicaciones, entre muchos otros.

SNIFFERS.- Herramientas para efectuar ataques pasivos, en donde un proceso “olfateada” el tráfico que se genera en una red, siendo capaz de leer toda la información que circule en el segmento de red en el que se ubique.

SQL.- (Structured Query Language). Lenguaje estructurado de consulta que se ha convertido en un estándar para la industria. Ésta formado por un conjunto de sentencias que permiten la definición y la manipulación de datos.

SSL.- (Secure Sockets Layer). Protocolo diseñado por Netscape Communications que permite cifrar una conexión, incluso garantiza la autenticación. Se basa en la criptografía aritmética y en el concepto de certificados.

SWAP.- Es un área de intercambio de datos que se establece en el disco duro con el fin de mantener en memoria principal sólo los datos más utilizados de un programa y el resto de ellos almacenarlos en ésta área.

TCP/IP.-(Transmission Control Protocol / Internet Protocol).- Es un conjunto de protocolos de red en una pila que permite la comunicación entre computadoras conectadas entre si. Es la base sobre la que trabaja Internet, enlazando millones de computadoras que utilizan diferentes sistemas operativos. Fue desarrollado por el Departamento de Defensa de los Estados Unidos.

UNIX.- Sistema Operativo desarrollado por los laboratorios Bell en los años 60, sobre éste sistema están finalmente basadas todas las distribuciones de Linux.

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

WWW.- (World Wide Web). Servicio de Internet que permite la transferencia de documentos de hipertexto y multimedia.

XWINDOWS.- Es el sistema que maneja la interfaz grafica en un sistema operativo Linux.

ANEXO I

Tipos de Datos MySQL

Tipos numéricos:

TinyInt: Es un número entero con o sin signo. Con signo el rango de valores válidos va desde -128 a 127. Sin signo, el rango de valores es de 0 a 255.

Bit o Bool. Un número entero que puede ser 0 ó 1.

SmallInt: Número entero con o sin signo. Con signo con el rango de valores que va desde -32768 a 32767.

MediumInt. Número entero con o sin signo. Con signo el rango de valores va desde -8.388.608 a 8,368,607. Sin signo el rango va desde 0 a 16777215.

Integer, Int. Numero entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483648. Sin signo el rango va desde 0 a 429,4967,295.

BigInt: Numero entero con o sin signo. Con signo el rango va desde -9,233,372,036,854,775,808 a 9,223,372,036,854,775,807. Sin signo el rango va desde 0 a 18,446,744,073,709,551,615.

Float: Número pequeño de coma flotante de precisión simple. Los valores válidos van desde $-3.402823466E+38$ a $-1.175494351E-38$, 0 y desde $2.2250738585072014E - 308$.

xReal, Double: Numero de coma flotante de precisión doble. Los valores permitidos van desde $-1.7976931348623157E+308$ a $-2.22507385850720E-308$, 0 y desde $2.22507385855072014E-308$

Decimal, Dec, Numeric. Numero de coma flotante desempaquetado. El número se almacena con una cadena.

La siguiente tabla muestra una relación del tipo de campo con el tamaño de almacenamiento que permite:

Tipo de Campo	Tamaño de Almacenamiento
TinyInt	1Byte
SmallInt	2Bytes
MediumInt	3Bytes
Int	4Bytes
Integer	4Bytes
BigInt	8Bytes
Float(X)	4 u 8 Bytes
Float	4Bytes
Double	8Bytes
Double Precision	8Bytes
Real	8Bytes
Decimal(M,D)	M+2 Bytes si D > 0, M+1 bytes si D = 0
Numeric(M,D)	M+2 Bytes si D > 0, M+1 bytes si D = 0

Tipos de fecha y hora.

A la hora de almacenar fechas, MySQL no comprueba de una manera estricta si una fecha es válida o no. Simplemente comprueba que el mes éste comprendido entre 0 y 12 y que el día éste comprendido entre 0 y 31.

Date: Tipo de fecha, almacena fecha. El rango de valores va desde 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día.

DateTime. Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de Diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día.

Timéstamp: Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo.

Tamaño	Formateo
14	AñoMesDiaHoraMinutosSegundo aaaammddhhmmss
12	AñoMesDiaHoraMinutosSegundo aammddhhmmss
8	AñoMesDia aaaammdd
4	AñoMes aamm
2	Año aa

Time: Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de “HH:MM:SS”.

Year. Almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño 2 o tamaño 4 dependiendo de que si se quiere almacenar el año con 2 o cuatro dígitos.

Tipos de cadena:

Char(n): Almacena una cadena de longitud fija. La cadena podrá contener desde 0 a 25.

VarChar(n). Almacena una cadena longitud variable. La cadena podrá contener 0 a 255 caracteres.

Diferencia de almacenamiento entre los tipos de Char y VarChar

Valor	Char(4)	Almacenamiento	VarChar(4)	Almacenamiento
''	''	4 Bytes	''	1Byte
'ab'	'ab '	4 Bytes	'ab'	3Bytes
'abcd'	'abcd'	4 Bytes	'abcd'	
'abcdefgh'	'abcd'	4Bytes	'abcd'	5Bytes

Dentro de los tipos de cadena que se pueden distinguir otros 2 subtipos, los tipo Text y los tipo BLOB (Binary Large Object).

La diferencia entre un tipo y otro es el tratamiento que reciben a la hora de realizar ordenamientos y comparaciones. Mientras que el tipo Text se ordena sin tener en cuenta las mayúsculas y las minúsculas, el tipo BLOB se ordena teniéndolas en cuenta.

Los tipos BLOB se utilizan para almacenar datos binarios como pueden ser ficheros.

TinyText y TinyBlob: Columna con una longitud maxima de 255 carcateres.

Blob y Text: Un texto con un máximo de 65535 caracterese.

MédiumBlob y MédiumText: Un texto con un máximo de 16,777,215 caracteres.

LongBlob y LongText: Un texto con un máximo de caracteres 4,294,967,295. Hay que tener en cuenta que debido a los protocolos de comunicación los paquetes pueden tener un máximo de 16 Mb.

Enum: Campo que puede tener un único valor de una lista específica. El tipo Enum acepta hasta 65535 valores distintos.

Set: Un campo que puede contener, ninguno, uno o varios valores de una lista. La lista puede tener un máximo de 64 valores.

Tipo de campo	Tamaño de Almacenamiento
Char(n)	N bytes
Varchar(n)	N + 1 bytes
TinyBlob, TinyText	Longitud + 1 bytes
Blob, Text	Logitud + 2 bytes
MediumBlob, MediumText	Longitud + 3 bytes
LongBlob, LongText	Longitud + 4 Bytes
ENUM('value1','value2')	1 ó dos bytes dependiendo del numero de valores
SET('value1','value2')	1,2,3,4 ó 8 bytes, dependiendo del numero de valores

ANEXO III

FUNCIONES SQL

**FUNCIÓNES DE CONTROL DE FLU-
JO.**

IF	Elección en función de una expresión booleana
IFNULL	Elección en función de si el valor de una expresión es NULL
NULLIF	Devuelve NULL en función del valor de una expresión

FUNCIÓNES MATEMATICAS.

ABS	Devuelve el valor absoluto.
ACOS	Devuelve el arco-coseno
ASÍN	Devuelve el arco-seno
ATAN y ATAN2	Devuelven el arco-tangente
CEILING y CEIL	Redondo hacia arriba
COS	Coseno de un angulo
COT	Cotangente de un angulo
CRC32	Calculo de Comprobación de una redundancia cíclica

DEGREES	Conversión a grados radianes
EXP	Cálculo de potencia de s
FLOOR	Redondeo hacia abajo
LN	Logaritmo natural
LOG	Logaritmo en base arbitraria
LOG10	Logaritmo en base 10
LOG2	Logaritmo en base2
MOD o %	Resto de una división entera
PI	Valor del numero π
POW O POWER	Valor de potencias
RADIANS	Conversión de radianes a grados
RAND	Valores aleatorios
ROUND	Calculo de redondeos
SIGN	Devuelve el signo
SIN	Calculo del seno de un angulo
SQRT	Calculo de la raíz cuadrada
TAN	Calculo de la tangente de un angulo
TRUNCATE	Elimina decimales

FUNCIÓNES PARA EL TRATAMIENTO DE CADENAS DE CARACTERES

ASCII	Valor de código ASCII de un carácter
BIN	Conversión a binario
BIT_LENGTH	Calculo de una longitud de cadena de bits
CHAR	Convierte de ASCII a carácter
CHAR_LENGTH Ó CHARACTER_LENGTH	Calculo de longitud de cadenas de caracteres
COMPRESS	Comprime una cadena de caracteres
CONCAT	Concatena dos cadenas de caracteres
CONCAT_WS	Concatena cadenas con separadores
CONV	Convierte números entre distintas

	bases
ELT	Elección entre varias cadenas
EXPORT_SET	Expresiones binarias como conjuntos
FIELD	Busca el índice en listas de cadenas
FIND_IN_SET	Búsqueda en listas de cadenas
HEX	Conversión de números en base hexadecimal
INSERT	Inserta una cadena en otra
INSTR	Busca una cadena en otra
LEFT	Extraer parte izquierda de una cadena
LENGTH U OCTECT_LENGTH	Calcula la longitud de una cadena en bytes

LOAD_FILE	Lee un fichero en una cadena	QUOTE	Entrecomilla una cadena
LOCATE Ó POSITION	Encontrar un posición de una cadena dentro de otra	REPEAT	Construye una cadena con una repetición de otra
LOWER O LCASE	Convierte una cadena a minúsculas	REPLACE	Busca una secuencia en una cadena y la sustituye por otra
LPAD	Añade caracteres a la izquierda de una cadena.	REVERSE	Invierte el orden de los caracteres de una cadena
LTRIM	Elimina espacios a la izquierda de una cadena	RIGHT	Devuelve la parte derecha de una cadena
MAKE_SET	Crea un conjunto a partir de una expresión binaria	RPAD	Inserta caracteres al final de una cadena
OCT	Convierte un número a Octal	RTRIM	Elimina caracteres blancos a la derecha de una cadena
ORD	Obtiene el código ASCII, incluso con caracteres multibyte	SOUNDEX	Devuelve la cadena

	“soundex” para una ca- dena concreta
SOUNDS LIKE	Compara cadenas se- gún su pro- nunciación
SPACE	Devuelve cadenas con- sistentes en espacios
SUBSTRING O MID	Extrae sub- cadenas de una cadena
SUBSTRING_INDEX	Extrae sub- cadenas en función de delimitadores
TRIM	Elimina sufi- jos y/o prefi- jos de una cadena
UCASE O UPPER	Convierte una cadena a mayúsculas
UNCOMPRESS	Descomprime una cadena comprimida mediante COMPRESS

UNCOMPRESS_LENGTH	Calcula la longitud ori- ginal de una cadena com- primida
UNHEX	Convierte una cadena que represen- ta un número hexadecimal a cadena de caracteres

**FUNCIONES DE CÓMPARACION DE
CADENAS:**

STRCMP	Compara cadenas
--------	-----------------

FUNCIÓNES PARA EL MANEJO DE FECHAS:

ADDDATE	Suma un intervalo de tiempo a una fecha
ADDTIME	Suma tiempos
CONVERT_TZ	Convierte tiempos entre distintas zonas horarias
CURDATE O CURRENTDATE	Obtener la fecha actual
CURTIME O CURRENT_TIME	Obtener la hora actual
DATE	Extrae la parte correspondiente a la fecha
DATEDIFF	Calcula la diferencia en días entre 2 fechas
DATE_ADD	aritmética de fechas, suma intervalo de tiempo
DATE_SUB	aritmética de fechas, resta un intervalo de

	tiempo
DATE_FORMAT	Formatea el valor de una fecha
DAY O DAYOF-MONTH	Obtiene el día del mes a partir de una fecha
DAYNAME	Devuelve el nombre del día de la semana
DAYOFWEEK	Devuelve el índice del día de la semana
DAYOFYEAR	Devuelve el día del año para una fecha
EXTRACT	Extrae parte de una fecha
FROM_DAYS	Obtener una fecha a partir de un número de días
FROM_UNIXTIME	Representación de fechas de UNIX en formato de cadena
GET_FORMAT	Devuelve una cadena de formato
HOUR	Extrae la hora de un valor time
LAST_DAY	Devuelve la fe-

	cha para el último día del mes de una fecha	O SYSDATE	
MAKEDATE	Calcula una fecha a partir de un año y un día del año	PERIOD_ADD	Añade meses a un periodo (año/mes)
MAKETIME	Calcula un valor de tiempo a partir de una hora, minuto y segundo	PERIOD_DIFF	Calcula la diferencia de meses entre 2 periodos (año/mes).
MICROSECOND	Extrae los microsegundos de una expresión de fecha/hora o de hora	QUARTER	Devuelve el cuarto del año para una fecha
MINUTE	Extrae el valor de minutos de una expresión	SECOND	Extrae el valor de segundos de una expresión time
MONTH	Devuelve el mes de una fecha	SEC_TO_TIME	Convierte una cantidad de segundos a horas, minutos y segundos
MONTHNAME	Devuelve el nombre de un mes para una fecha	STR_TO_DATE	Obtiene un valor DATETIME a partir de una cadena con una fecha y una cadena de formato
NOW O CURRENT_TIMESTAMP O LOCALTIME O LOCALTIMESTAMP	Devuelve la fecha y la hora actual	SUBDATE	resta un intervalo de tiempo de una fecha
		SUBTIME	resta 2 expresio-

	nes time		el año cero
TIME	Extrae la parte de la hora de una expresión fecha/hora	UNIX_TIMÉSTAMP	Devuelve un timéstamp o una fecha o una fecha en formato UNIX, segundos desde 1070
TIMEDIFF	Devuelve en tiempo entre dos expresiones de tiempo	UTC_DATE	Devuelve la fecha UTC actual
TIMÉSTAMP	Convierte una expresión de fecha en fecha/hora o suma un tiempo a una fecha	UTC_TIME	Devuelve la hora UTC actual
TIMÉSTAMPADD	Suma un intervalo de tiempo a una expresión de fecha/hora	UTC_TIMÉSTAMP	Devuelve la fecha y hora UTC actual
TIMÉSTAMPDIFF	Devuelve la diferencia entre dos expresiones de fecha.	WEEK	Calcula el número de semana para una fecha.
TIME_FORMAT	Formatea un tiempo	WEEKDAY	Devuelve el número de día de la semana para una fecha
TIME_TO_SEC	Convierte un tiempo a segundos	WEEKOFYEAR	Devuelve el número de la semana del año para una fecha
TO_DAYS	Calcula el número de días desde	YEAR	Extrae el año de una fecha
		YEARWEEK	Devuelve el año y semana de una fecha

FUNCIÓNES DE ENCRIPADO

AES_ENCRYPT Y AES_DECRYPT	Encriptar y de- senciptar datos usando el algo- ritmo oficial AES
DECODE	Desencipta una cadena usando contraseña
ENCODE	Encripta una cadena usando una contraseña
DES_DECRYPT	Desencipta una cadena usando una contraseña
DES_ENCRYPT	Encripta usando el algoritmo Tri- ple-DES
ENCRYPT(str)	Encripta str usando la llama- da del sistema Unix crypt()
MD5(String)	Calcula un checksum MD5 de 128 bits para

	la cadena string
PASSWORD u OLD_PASSWORD	Calcula una ca- dena contraseña a partir de la ca- dena en texto.
SHA o SHA1	Calcula un checksum SHA1 de 160 bits

FUNCIÓNES DE INFORMACIÓN

BENCHMARK	Ejecuta una expresión varias veces
CHARSET	Devuelve el conjunto de caracteres de una cadena
COERCIBILITY	Devuelve el valor de restricción de colección de una cadena
COLLATION	Devuelve la colección para un conjunto de caracteres de una cadena
CONNECTION_ID	Devuelve el ID de una conexión
CURRENT_USER	Devuelve el nombre de usuario y el host para la conexión actual
DATABASE	Devuelve el número de la base de datos actual

FOUND_ROWS	Calcula cuantas filas se hubiesen obtenido de una sentencia SELECT sin la cláusula LIMIT
LAST_INSERT_ID	Devuelve el último valor generado automáticamente para una columna AUTO_INCREMENT
USER O SESSION_USER O SYSTEM_USER	Devuelve el nombre de usuario y host actual de MySQL
VERSIÓN	Devuelve la versión del servidor de MySQL

FUNCIÓNES DE GRUPOS:

AVG	Devuelve el valor medio
BIT_AND	Devuelve la operación de bits AND para todos los bits de una expresión
BIT_OR	Devuelve la operación de bits OR para todos los bits de una expresión
BIT_XOR	Devuelve la operación de bits XOR para todos los bits de una expresión
COUNT	Devuelve el número de valores distintos de NULL en las

	filas recuperadas por una sentencia SELECT
COUNT_DISTINCT	Devuelve el número de valores diferentes, distintos de NULL
GROUP_CONCAT	Devuelve una cadena con la concatenación de los valores de un grupo
MIN	Devuelve el valor mínimo de una expresión
MAX	Devuelve el valor máximo de una expresión
STD O STDDEV	Devuelve la desviación estándar de una expresión
SUM	Devuelve la suma de una expresión
VARIANCE	Devuelve la varianza estándar de una expresión

FUNCIÓNES GENERALES

DEFAULT	Devuelve el valor por defecto para una columna.
FORMAT	Formatea el número según la plantilla '#,###,###,##'
GET_LOCK	Intenta obtener un bloqueo con el nombre dado
INET_ATON	Obtiene el entero equivalente a la dirección de red dada en formato de cuarteto con puntos
INET_NTOA	Obtiene la dirección en formato de cuarte-

	to con puntos dado un entero
IS_FREE_LOCK	Verifica si un nombre de bloqueo ésta libre
IS_USED_LOCK	Verifica si un nombre de bloqueo es usado
MÁSTER_POS_WAIT	Espera hasta que el esclavo alcanza la posición especificada en el diario maestro
RELEASE_LOCK	Libera un bloqueo
UUID	Devuelve un identificador único universal