UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES "ARAGÓN"

INGENIERÍA EN COMPUTACIÓN

"Desarrollo e Implementación de Sistemas con Software Libre en LINUX"

TRABAJO ESCRITO

EN LA MODALIDAD DE SEMINARIOS Y CAPACITACIÓN PROFESIONAL QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A

RAÚL MILTON RUBIO LORENZANA

ASESOR: ING. RODOLFO VÁZQUEZ MORALES

MÉXICO, 2005

AGRADECIMIENTOS

A MIS PALDRES:

POR EL AMOR INCONDICIONAL QUE ME HAN BRINDADO, POR DARME LA OPORTUNIDAD DE SUPERARME, POR SUS NOCHES DE VELA Y LAS ALEGRÍAS QUE TANTAS ME HAN DADO...
LOS AMO.

A MIS HERMANAS:

POR SABER QUE CUENTO CON ELLAS EN LOS MOMENTOS OSCUROS Y POR CONSOLARME EN LOS TIEMPOS DIFÍCILES.

A EDGAR PEÑA:

PORQUÉ HA SIDO COMO UN HERMANO PARA MÍ, POR SU APOYO INCONDICIONAL Y SU INCALCULABLE AMISTAD.

DESARROLLO E IMPLEMENTACIÓN DE SISTEMAS CON SOFTWARE LIBRE EN LINUX

Tabla de Contenido

Introducción

Dentro de un mundo tan cambiante a lo que a tecnología se refiere, es difícil mantenerse actualizado si tomamos en cuenta que la tecnología cambia día con día, ya que cada mes aparecen nuevos productos, tanto de software como de hardware: un procesador más rápido, un mayor bus de datos, mayores velocidades de discos duros, mayor almacenaje de los mismos, etc. Tal vez lo que tarde un poco más en cambiar son los métodos de análisis para algún problema, metodologías y procedimientos.

En el área del software ocurre lo mismo, se puede comprar una licencia de un software para satisfacer determinadas necesidades, tal vez no al cien por ciento, pero si en un alto porcentaje. Pero tal vez ese software requiere una mejora o una adecuación, como la personalización del producto, todo esto se traduce en dinero, sin mencionar que una vez teniendo este producto tal vez haya que pagar la licencia anual o cada seis meses.

¿Qué inconvenientes tenemos con éste tipo de software?.

Bueno, en principio y como es muy visible, el costo es elevadísimo. No hay que dudar que hay software muy bueno y de gran eficiencia, pero si uno requiere alguna modificación simplemente no se puede hacer, Habrá que llamar a la compañía para ver, si es posible, la modificación del mismo. **Dinero...**

¿Qué hay de la seguridad?

Constantemente oímos que se realizaron ataques electrónicos por piratas o hackers a determinada empresa aprovechando las vulnerabilidades de su software. Éstas mismas empresas invierten grandes cantidades de dinero en mejorar su productos para evitar estas desgracias, lo que a su vez encarece aún más el producto además de que tenemos que pagar por una actualización, eso sí, sin mencionar que tardan mucho tiempo en desarrollar un "parche" para las vulnerabilidades de su producto. **Tiempo y seguridad...**

Esto aplica para cualquier ramo del software, llámense: *Manejadores de Bases de Datos, Sistemas operativos, herramientas administrativas, herramientas de seguridad, etc.*

Pero no todo es malo en el Software de licencia, hay que reconocer también que dan mucho soporte a sus productos y no se tienen conflictos con el hardware a la hora de la instalación. Algunas veces ellos capacitan al personal de alguna empresa que compra su producto, dan manuales de operación y tal vez hasta mochilas o gorras nos regalen...

Todo esto está muy bien si somos una empresa con gran solvencia económica y de gran capital, pero ¿qué hay si no tenemos demasiados recursos o la infraestructura suficiente para poder adquirir productos o servicios de este tipo? Si quisiéramos implementar una aplicación para web, un ejemplo muy trillado, un carrito de compras, por lo menos lo que necesitaríamos como base sería:

- Sistema Operativo.
- Servidor Web.
- Manejador de Bases de Datos.
- Plataforma para la programación.
- Lenguaje para interfaces.

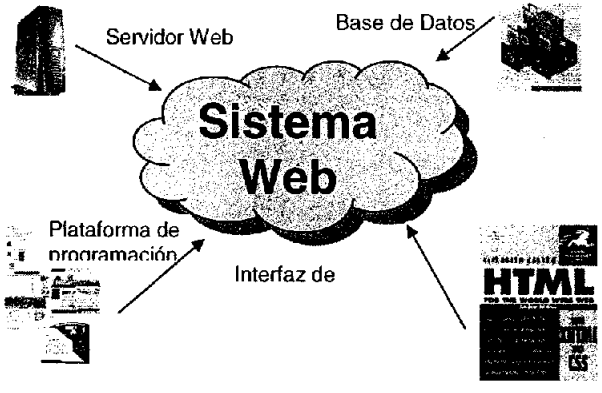


Figura 1. Esquema del Sistema Web

Tal vez nuestros proveedores podrían ser:

- *Windows NT,* de Microsoft como sistema operativo
- *IIS,* de Microsoft como servidor web
- *Oracle,* podría ser nuestro manejador de Bases de Datos
- *Java2 o .NET,* como nuestra plataforma de programación
- *HTML (*no entraría en el costo, pero si como base para desarrollo*)*

Con esto ya estamos hablando de números grandes, muy grandes y sin mencionar que algunas licencias vencen al año y hay que volver a pagarlas.

Una propuesta

Software Libre.

Así de simple y fácil es la respuesta. *Software libre.*

¿Qué por qué Software Libre?

Porqué es totalmente gratuito, bajo la licencia GPL (General Public Licence). Existe infinidad de Software Libre para desarrollo de aplicaciones tan simples o tan complejas como uno quisiera, lo que nos ahorra mucho **Dinero.**

El código fuente es gratuito, a diferencia de un software licenciado, el código fuente bajo la licencia GPL está a disposición de todo mundo, se puede descargar y modificar libremente, lo que hace que uno ajuste el software a su manera sin necesidad de invertir un solo peso.

Es seguro, pues al encontrar una vulnerabilidad o deficiencia del software, muchas personas alrededor del mundo se dan a la tarea de desarrollar dichos "parches" para ser implementado en futuras versiones o en las versiones que estén vigentes. Al igual que el software de licencia es robusto.

Para poder implementar el mismo ejemplo del carrito de compra utilizando software libre, podemos utilizar:

- *Sistema Operativo Linux* (no importa la versión)
- *Servidor Web Apache*
- *Postgresql*, servidor de base de datos
- *PHP*, como plataforma de programación
- *HTML*

Lo que nos da una inversión o gasto de **$0 PESOS**, pues todo éste software está disponible en la red para su utilización sin costo alguno. Lo que si hay que reconocer, es que no existe un soporte técnico formal sobre cada producto a diferencia del software licenciado, pero existen muchos foros en la red donde uno puede consultar en línea.

Puesto que hoy en día todas las aplicaciones, o si no todas, en su mayoría están enfocadas a Web es que es necesario aprender todo lo necesario para el desarrollo de éstas. Es por ello que me interesó este diplomado en particular.

Para lograr tal, dentro del **primer capítulo** trataré se dará una visión general de lo que es Linux y un poco de su estructura básica. En el **capítulo segundo** se abarcarán los aspectos que hay que considerar en la administración de un sistema Linux. El **capítulo tercero** mostrará algunas de las etiquetas más comunes del HTML para elaboración de interfaces de usuario de un sistema. La administración y puesta en marcha de un servidor Web, como lo es APACHE, será detallada en el **cuarto capítulo** de este trabajo. Para desarrollar aplicaciones Web en forma, el **capítulo quinto** describirá lo que es el lenguaje de programación PHP y su interacción con Bases de Datos usando como motor MySQL en el **sexto capítulo**.

La seguridad es muy importante en un sistema en línea, es fundamental y necesaria, esto será tratado en el **capítulo séptimo.** Finalmente el **capítulo octavo** ampliará lo ya visto en el capítulo quinto incluyendo la programación orientada a objetos y la interacción de PHP con el motor de base de datos POSTGRES.

Capítulo I. Sistema operativo LINUX

En 1990, Linus Torvals, un estudiante de 23 años de la Universidad de Helsinki, en Finlandia, comenzó a desarrollar un proyecto basado en el MINIX de Andrew Tenembaum. Quería llevar a cabo, sobre una computadora con procesador Intel 80386, un sistema operativo tipo UNIX que ofreciese más capacidades que el limitado MINIX[1]. Linus empezó escribiendo el núcleo del proyecto en ensamblador, y luego comenzó a añadir código en C, lo cual incrementó la velocidad de desarrollo. En octubre de 1991, anuncio la primera versión "oficial" de LINUX, la 0.02, que ya era capaz de ejecutar el SHELL bash[2] y el compilador gcc de GNU Linux es un sistema operativo multitarea y multiusuario. Esto es, se trata de un programa que interacciona entre el hardware de la máquina y nosotros, permitiéndonos, gracias a los programas, hacer tareas que de otro modo sería muy complicado (acceso a disco, escritura de archivos, etc.). Decimos *multitarea* porque es capaz de tener varios programas ejecutándose al mismo tiempo, y decimos *multiusuario*, porque es capaz de admitir a varios usuarios distintos trabajando sobre la misma máquina al mismo tiempo.

Dado que se trata de un sistema multiusuario, cuando alguien quiere acceder a la máquina, debe *identificarse*, para poder ser reconocido por esta y permitirle la entrada al sistema, si se trata de un usuario autorizado. Este proceso es el que se conoce como ***logging in***. Durante este proceso, la máquina nos preguntará nuestro nombre de usuario (*login*) y nuestra contraseña (*password*). Es el administrador del sistema (root) quien debe crearnos una cuenta en la máquina y quien nos dará los datos.

[1] Sistema operativo basando en MULTICS e inspirando en UNIX, mucho más reducido. Su nombre proviene de Mini-UNIX que sería la conjunción de ambos sistemas operativos.
[2] Acrónimo de *Bourne Again Shell*. Bourne escribió este intérprete de comandos.

1.1 Características

Dentro de este capítulo se describieron las características de manera general del sistema operativo Linux. Por mencionar algunas, tenemos:

- Multiproceso. Permite la ejecución de varias aplicaciones simultáneamente.
- Multiusuario. Distintos usuarios pueden acceder a los recursos del sistema simultáneamente aunque se trate de una instalación en una sola máquina.
- Multiplataforma. Funciona con la mayoría de plataformas del mercado: Intel 386/486/Pentium, Motorola 680, Sun Sparc, etc
- Shells[3] programables que lo convierten en el sistema más flexible que existe.
- Soporte para cualquier cantidad y tipo de dispositivos directamente en el núcleo.
- Soporte para la mayoría de sistemas de archivos.

En el mundo de Linux tenemos infinidad de "sabores" de éste potente sistema operativo y cada una de ellas tiene sus propias características, ventajas y desventajas. Tal vez sea muy difícil tomar una decisión sobre que distribución elegir a la hora de empezar a trabajar con él. La mejor decisión será aquella que cumpla con todos nuestros requerimientos, objetivos y trabajo. Algunas de las distribuciones más comunes son:

- Debian GNU/Linux
- Gentoo Linux
- LindowsOS
- Lycoris Dekstop/LX
- Knoppix
- Mandrake Linux
- Red Hat Linux
- Slackware Linux
- SuSE Linux

[3] Interfaz del sistema operativo que interactúa con el usuario y pasa los comandos para ser ejecutados

1.2 El sistema de archivos

El sistema de archivos es más o menos "la forma de escribir los datos en el disco duro". El sistema de archivos nativo de Linux es el EXT2. Ahora existen otros sistemas de archivos con journalising (si se arranca sin haber cerrado el sistema, no necesitan hacer un chequeo sino que recuperan automáticamente su último estado), los más conocidos son EXT3, ReiserFS y XFS.

La estructura de directorios que sigue Linux es parecida a la de cualquier UNIX. No tenemos una "unidad" para cada unidad física de disco o partición como en Windows, sino que todos los discos duros o de red se montan bajo un sistema de directorios en árbol, y algunos de esos directorios enlazan con estas unidades físicas de disco.

Estructura de directorios en Linux

Directorio	Descripción
/	Es la raíz del sistema de directorios. Aquí se monta la partición principal Linux EXT.
/etc	Contiene los archivos de configuración de la mayoría de los programas.
/home	Contiene los archivos personales de los usuarios.
/bin	Contiene comandos básicos y muchos programas.
/dev	Contiene archivos simbólicos que representan partes del hardware, tales como discos duros, memoria...
/mnt	Contiene subdirectorios donde se montan (se enlaza con) otras particiones de disco duro, CDROMs, etc.
/tmp	Archivos temporales o de recursos de programas.
/usr	Programas y librerías instalados con la distribución
/usr/local	Programas y librerías instalados por el administrador
/sbin	Comandos administrativos
/lib	Librerías varias y capítulos del kernel[4]
/var	Datos varios como archivos de log (registro de actividad) de programas, bases de datos, contenidos del servidor web, copias de seguridad.
/proc	Información temporal sobre los procesos del sistema

[4] Es lo que hace que un sistema operativo sea un sistema operativo. Es por decir de alguna manera "el corazón del sistema operativo"

Los nombres de archivos en Linux distinguen mayúsculas de minúsculas. En Linux los archivos no tienen por qué tener una extensión. La suelen tener como en motivo de orientación o identificación, pero no es en absoluto necesario. Linux sabe qué contiene cada archivo independientemente de cuál sea su extensión.

Los archivos y directorios ocultos en Linux comienzan su nombre por un punto (.). Hay ciertos caracteres que nunca se deberían utilizar a la hora de nombrar un archivo. Uno de ellos es el espacio, nunca se debe llamar a un archivo con un nombre que contenga un espacio. Tampoco son recomendados otros caracteres raros como signos de puntuación (a excepción del punto), acentos o la ñ

1.3 Los usuarios y los permisos de archivos

Linux, como se mencionó anteriormente, es un sistema operativo multiusuario. Cada usuario generalmente tiene su carpeta de usuario en /home/usuario. Por defecto sólo puede escribir, modificar y borrar archivos dentro de esta carpeta. Ningún otro usuario (excepto root[5]) puede acceder a los archivos que hay en este directorio, ni si quiera puede ver cuáles son. Este usuario puede leer en el resto de las carpetas que hay en el sistema de archivos excepto en la de root y las de otros usuarios. Un usuario no puede causar por este motivo daño al sistema ni cambiar su configuración de ninguna forma.

Todos y cada uno de los archivos y directorios del árbol jerárquico que monta nuestro sistema Linux tienen permisos. Estos permisos dicen, para cada usuario del sistema, si puede ejecutarlo, si puede ver su contenido o si puede borrarlo o modificarlo. Del mismo modo, cada elemento del sistema de archivos tiene un dueño. Por defecto, este dueño del elemento (tanto directorio como archivo) tiene acceso total a él y puede realizar todas las acciones posibles permitidas. El resto de usuarios pueden leer y ejecutar este elemento por defecto aunque todo esto se puede cambiar para cada uno de los elementos. Todos los archivos de las carpetas de sistema y configuración suelen tener a root como propietario. Los de

[5] Es llamado así el superusuario que tiene control total sobre el sistema. Del inglés **root** que significa "raíz".

la carpeta personal de cada usuario tienen a ese usuario como propietario, pero el resto de usuarios normales no tienen ningún permiso sobre estos elementos, y lo mismo ocurre con la carpeta de root (que se encuentra en la raíz, en /root).

1.4 Nombramiento de dispositivos y particiones

Debemos saber de qué manera nombra Linux a los discos duros que tenemos conectados a nuestra máquina y sobre todo a sus particiones. Todos los discos duros (IDE[6]) comienzan su nombre como **hd**[NT]. Un ejemplo de nombre completo de disco duro sería hda y de la primera partición de ese disco duro sería hda1.

La 'a' significa que ese disco duro está conectado al IDE1 como maestro. Si fuera esclavo tendría la 'b', y si estuviera conectado al IDE2 como maestro, la 'c', y si estuviera al IDE2 como esclavo, la 'd'. El número 1 indica que es la primera partición (primaria y no lógica) del disco duro en cuestión. La segunda geométricamente hablando (primaria) sería la 2 y así sucesivamente. La primera partición lógica de un disco duro se nombra con el número 5, independientemente de si pertenece a la primaria 1, 2, 3 ó 4. La segunda se nombraría con un 6 y así sucesivamente.

1.5 Creación y gestión de archivos y directorios

Algunos de los comandos de Linux para el manejo de archivos y directorios son:

ls: lista el contenido de directorios del sistema. Hay muchas opciones para este comando (*-a, -l, -d, -r,...*), que a su vez, se pueden combinar de muchas formas. Algunas de las que podríamos considerar más comunes son:

> **-l (*long*):** formato de salida largo, con más información que utilizando un listado normal.

---

[6] Integrated Drive Electronics. Interfaz que se utiliza para conectar un disco duro a la tarjeta madre

[NT] Abreviación de "hard disk" en inglés, que quiere decir "disco duro"

-a *(all)*: se muestran también archivos y directorios ocultos.
-R *(recursive)*: lista recursivamente los subdirectorios.

cd: con este comando podemos cambiar de directorio de trabajo. La sintaxis básica es la siguiente:

cd [nombre_directorio]

Si usamos el *shell* bash, (que es el que se instala por defecto en los sistemas Linux), simplemente tecleando cd, volvemos a nuestro directorio HOME[7]. Es muy posible que en nuestro Linux, al hacer cd para entrar en algún directorio, no lo veamos reflejado en el *prompt*[8] .

cp: sirve para copiar un archivo (origen), en otro lugar (puede ser un archivo o un directorio), indicado en destino. Su sintaxis es *cp <origen> <destino>*. Si el destino es un directorio, los archivos de origen serán copiados dentro de él. Para copiar de forma recursiva (es decir, también subdirectorios) podemos usar la opción *–r*.

mkdir: tal y como su nombre parece querer decir, crea un directorio. La sintaxis será *mkdir <nombre_directorio>*. Para crear un directorio tenemos que tener en cuenta los permisos del directorio en que nos encontremos trabajando pues, si no tenemos permiso de escritura, no podremos crear el directorio.

mv: con este comando podemos renombrar un archivo o directorio, o mover un archivo de un directorio a otro. Dependiendo del uso que hagamos, su sintaxis variará. Su sintaxis podría ser: *mv <archivo(s)> <directorio>* moverá los archivos especificados a un directorio, mientras que con *:mv <archivo1> <archivo2>* renombrará el primer archivo, asignándole el nombre indicado en *<archivo2>*.

[7] Como se mencionó anteriormente, home hace referencia la directorio de trabajo de cada usuario.

pwd: imprime en pantalla la ruta completa del directorio de trabajo donde nos encontramos actualmente. No tiene opciones, y es un comando útil para saber en todo momento en qué punto del sistema de archivos nos encontramos.

rm: elimina archivos o directorios. Sus tres opciones son *-r* (borrado recursivo, es decir, de subdirectorios), *-f* (no hace preguntas acerca de los modos de los archivos), y *-i*(interactivo, solicita confirmación antes de borrar cada archivo). Su sintaxis es muy sencilla: *rm [-r] [-f] [-i] <archivo>*. Hay que tener mucho cuidado con este comando cuando se usen comodines[9], sobre todo si no lo ejecutamos en modo interactivo.

chmod: con este comando, cambiamos los permisos de acceso del archivo o del directorio que le especifiquemos como argumento. Podemos ejecutar de dos formas básicas este comando: la primera es *chmod <modo> <archivo>*, siendo *modo* un valor numérico de tres cifras octales que describe los permisos para el archivo. La segunda forma es algo más complicada:

Chmod <usuario> +|- <permiso> <archivo>

Indicaremos, en el parámetro *usuario*, la identidad del usuario(s) cuyos permisos queremos modificar *(u-user, g-group, o-others)*; a continuación irá un + o un -, dependiendo de si añadiremos el permiso o lo revocaremos, y en permiso debemos colocar el permiso a modificar *(r-read, w-write, x-exec)*.

Permisos en notación octal

Número octal	Permiso
4000	Establece el número de identificación de usuario al ejecutarse
2000	Establece el número de identificación de grupo al ejecutarse
1000	Establece el *bit adhesivo*
0400	Lectura por parte del dueño

[9] Vamos a entender por comodines a aquellos caracteres especiales que sustituyen a caracteres o a un conjunto de caracteres en alguna palabra. Estos comodines son: **. (punto), ?(interrogación), * (asterisco).**

Número octal	Permiso
0200	Escritura por parte del dueño
0100	Ejecución por parte del dueño
0040	Lectura por parte del grupo
0020	Escritura por parte del grupo
0010	Ejecución por parte del grupo
0004	Lectura por parte de los otros
0002	Escritura por parte de los otros
0001	Ejecución por parte de los otros

touch: actualiza la fecha de modificación de un archivo, o crea un archivo vacío si el archivo pasado como parámetro no existe. Con la opción *-c* no crea este archivo vacío. Su sintaxis es *touch* [*-c*] <*archivo*>.

cat: concatena y muestra el contenido de archivos. La salida de la orden cat será por defecto la pantalla.

more: visualiza un archivo pantalla a pantalla, no de forma continua, como hace cat. Es como cat, pero con pausas, lo que nos permite leer más tranquilamente un archivo. Al final de cada pantalla nos aparecerá un mensaje indicando --More--. Si en ese momento pulsamos *ENTER*, veremos una o más líneas del archivo; si pulsamos *ESPACIO*, veremos la siguiente pantalla, si pulsamos *b* la anterior, y si pulsamos *q* saldremos de *more*. Su sintaxis es *more* <*archivo*>.

passwd: se utiliza para cambiar la clave de acceso al sistema. Cuando ejecutemos este comando, tendremos que escribir la clave dos veces, y en ambas ha de coincidir. Esto nos evita que se nos asigne una clave no deseada por culpa de un error al escribir.

man: toma como argumento el nombre del programa del que se quiere ayuda, y busca en una base de datos ayuda sobre ese comando. Si no lo encuentra, indica que no existe.

1.6 Redireccionamiento

Existe la posibilidad de desviar la salida o la entrada a un comando para poder realizar funciones complejas u obtener los datos que saldrían por la pantalla directamente a otro archivo. Lo primero se denomina canalización y lo segundo redirección. El símbolo que se utiliza para efectuar la canalización es el denominado *pipe* o símbolo de canalización (|). Este símbolo permite que se pase la salida de un comando o programa a la entrada de otro. Podemos ver como ejemplo la utilización de este símbolo es cuando se requiere listar un directorio que ocupa más de una pantalla. Se podrá entonces utilizar el comando para listar **ls** y enviar su salida a otro programa que lo muestre de a una página de pantalla por vez, por ejemplo el **more**.

```
$ ls -l | more
```

Esto servirá para realizar lo que se mencionó anteriormente ya que la salida del comando **ls** será canalizada para que sea la entrada del comando **more** y éste se encargará de mostrar los datos por pantalla. Otros dos símbolos utilizados son el "<" y el ">". Lo que hacen es redirigir tanto la salida como la entrada estándar de o hacia un archivo. Por ejemplo, supongamos que necesitamos guardar en un archivo llamado listado la salida del comando **ls**.

```
$ ls > listado
```

Con esto le indicamos al comando **ls** que redireccione la salida estándar hacia un archivo de nombre *listado*. En caso de que el archivo exista, será reemplazado con la nueva información. Para agregar contenido a un archivo, inmediatamente después del contenido que posea, se tendrá que poner el símbolo de redirección dos veces

```
$ ls >> listado
```

El uso de < es para redireccionar el contenido de un archivo hacia la entrada estándar de un comando o programa. Supongamos que necesitamos ordenar el contenido de un archivo en orden alfabético, lo que se lograra con el comando **sort**. Podremos entonces redireccionar el contenido de un archivo a la entrada del comando **sort** para que éste lo ordene.

```
$ sort < nombres
```

De esta manera saldría por pantalla la lista de nombres ordenadas alfabéticamente. También se podrá redireccionar el error estándar para que no salga en pantalla. Si quisiéramos realizar un listado de un directorio y, en caso de producirse un error, este fuese redirigido a un archivo, haremos lo siguiente:

```
$ ls /bin 2>/tmp/error.ls
```

Esta simple redirección solo tendría efecto sobre el error estándar (stderr) o como también se denomina, *descriptor de archivo nº 2*. Con esta redirección los posibles errores serían redirigidos al archivo /tmp/error.ls.

1.7 Permisos de archivos, sus dueños y grupos

Para entender mejor el concepto de permisos se tendrá que tener en cuenta que cada usuario puede pertenecer a uno o más grupos. El administrador del sistema puede agregar al usuario a otros grupos. Estos grupos son necesarios para poder establecer una política de acceso más organizada dado que en cualquier momento se podría dar a un archivo el acceso a personas de un grupo determinado. Para cada objeto (archivo) que se encuentre en el sistema, Linux guarda información administrativa en la tabla de inodos[10]. Entre los datos que se guardan en esta tabla se encuentran la fecha de creación del archivo, modificación

[10] En informática, un inodo o Index Node es un apuntador a sectores específicos de disco duro en los cuales se encuentra la información del archivo, un inodo también contiene información de permisos, propietarios y grupos a los cuales pertenece el archivo.

del archivo y la fecha en que se cambio el inodo. Pero además contiene los datos en los que se centra toda la seguridad en Linux. Estos son:

- El dueño de archivo
- El grupo del archivo
- Los bits de modo o también llamados permisos de archivo

Los permisos están divididos en tres tipos: lectura, escritura y ejecución (rwx). Estos permisos pueden estar fijados para tres clases de usuario: el propietario del archivo, el grupo al que pertenece el archivo y para todo el resto de los usuarios. El permiso de lectura permite a un usuario leer el contenido del archivo o en el caso de que el archivo sea un directorio, la posibilidad de ver el contenido del mismo. El permiso de escritura permite al usuario modificar y escribir el archivo. En el caso de un directorio permite la crear nuevos archivos en él o borrar archivos existentes. El permiso de ejecución permite al usuario ejecutar el archivo, si tiene algo para ejecutarse. Para los directorios permite al usuario cambiarse a él con el comando **cd**. Si ejecutamos el siguiente comando, obtendremos la siguiente salida:

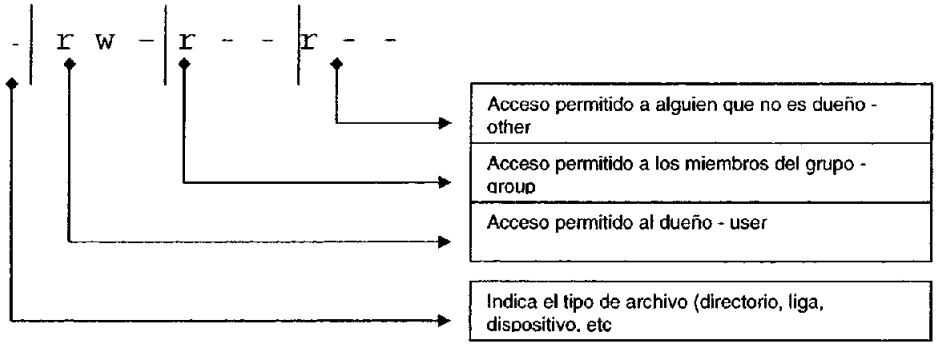
miltonr@exodus:~**$ ls -l**

```
total 824
drwxr-sr-x    2 miltonr  integra      1024 May  2 09:04 .
drwxrwsr-x    4 miltonr  integra      1024 Apr 17 21:08 ..
-rw-r--r--    1 miltonr  integra      1314 Jul  1  2004 Factorial.java
-rw-r--r--    1 miltonr  integra      2660 Jun 14  2004 PROYFINAL.zip
-rw-r--r--    1 miltonr  integra       875 Feb  1 12:11 RelacDuplic2.ctl
-rw-------    1 miltonr  integra     21504 Jan 28 14:48 archivos.doc
-rwxr-xr-x    1 miltonr  integra     11227 Feb 25 14:27 arch*
-rw-r--r--    1 miltonr  integra       976 Feb 25 14:27 arch.c
-rw-r--r--    1 miltonr  integra    411831 Jun 28  2004 bancoSource.zip
-rw-r--r--    1 miltonr  integra        93 Apr  8 08:18 contolM
-rw-------    1 miltonr  integra     66176 May 22  2004 core
-rw-r--r--    1 miltonr  integra      3442 Dec 10 17:33 datos.txt
-rw-------    1 miltonr  integra         0 Nov  3 17:05 dead.letter
```

Como se puede apreciar en este listado, también están el directorio actual, representado por un punto y el directorio padre representado por dos puntos .. . Ellos también poseen permisos y atributos que son mostrados. Para entender un

poco as más que significan los primeros 10 dígitos vamos a explicar. Se toma como ejemplo el siguiente archivo:

```
-rw-r--r--    1 miltonr  integra    411831 Jun 28  2004 bancoSource.zip
```



Tipos de archivo

Contenido	Significado
-	Archivo común
D	Directorio
C	Dispositivo de caracteres (tty o impresora)
B	Dispositivo de Bloque (usualmente disco rígido o CD-ROM)
L	Enlace simbólico
S	Socket
P	Pipe

Los siguientes 9 símbolos se toman en grupos de tres y cada grupo pertenece a una clase de permisos, y se muestran a continuación

Tipos de permisos

Permiso	Significado
r	Permiso de lectura
w	Permiso de escritura
x	Permiso de ejecución

Grupos de permisos

Columnas	Se aplica a	Significado
2,3,4	owner	Establece los permisos para el dueño del archivo
5,6,7	group	Establece los permisos para el grupo del archivo
8,9,10	other	Establece los permisos para los usuarios que no entran en las categorías anteriores

1.8 Tareas y procesos

En este punto habrá que determinar que es un proceso: Un **proceso** es un programa en ejecución y está en memoria. Además del nombre que el proceso recibe, que es el nombre del programa que esta corriendo, recibe también un número identificativo llamado PID (process ID, o ID de proceso).

Primer y segundo plano

Cualquier proceso puede estar en primer o segundo plano. Lo único a tener en cuenta es que solo un proceso estará en primer plano al mismo tiempo y es con el que estemos trabajando e interactuando en ese momento. Un proceso que este en segundo plano no recibirá ninguna señal de parte nuestra, es decir que no nos podemos comunicar con él a través. La utilidad de enviar un programa a segundo plano esta dada por el hecho de que existen tareas que no requieren de nuestro control para que se ejecuten. Para lanzar un proceso en segundo plano, tendremos que poner a continuación del comando el símbolo **&**.

1.9 LILO

LILO es un método de arranque para el sistema, un programa que reside en el sector de arranque del disco duro. Este programa se ejecuta cuando el sistema se inicia desde el disco duro, y puede arrancar automáticamente Linux desde una imagen de núcleo almacenada en el propio disco duro. LILO puede utilizarse también como una primera etapa de carga de varios sistemas operativos,

permitiendo seleccionar en tiempo de arranque qué sistema operativo (como Linux o MS-DOS, Windows, etc. La forma más simple de instalar LILO es editar el archivo de configuración, /etc/lilo.conf, y ejecutar el comando.

Hemos mencionado algunas características de Linux, planteando que es un Sistema Operativo mucho más robusto y seguro que Windows y sinceramente no es para cualquier persona o cualquier tipo de usuario. Requiere más que una simple ratón y una ventana para dar "clic´s", pues su configuración es mucho más personalizada y requiere ciertos conocimientos sobre su estructura, monta y usa un sistema de archivos mucho más complejo, provee mayores servicios y muchas opciones que otros sistemas operativos no poseen. Brinda una mayor flexibilidad de tal manera que uno puede ajustar el sistema a la forma que más nos convenga.

Así mismo incluye una gran variedad de software y herramientas tanto administrativas como para desarrollo. Su administración va más allá de lo comúnmente usado, pues no solo administra archivos, sino usuarios, permisos, grupos, administra de manera eficiente la memoria, los procesos y el uso del procesador.

En el capítulo siguiente se tratará la administración del sistema operativo Linux, en donde se mostrará que no es tan fácil o sencillo como lo es un sistema Windows, requiere algo más que unos simples "clics" para su completa y óptimas manipulación.

Capítulo II. Administración de LINUX

Linux es un SO tipo Unix y por lo tanto sus conceptos más básicos son comunes a los que incorpora cualquier sistema tipo Unix y resultan bastante distintos de otros conceptos que forman parte de la cultura de Microsoft.

La documentación de Linux es muy abundante, sin embargo muchas veces se asume una cultura general de Unix. Las generalidades más básicas de Unix muchas veces se tratan de un modo superficial y poco sistemático quizás porque son temas muy viejos.

En Linux los entornos gráficos son una opción no una obligación. El comportamiento del intérprete de comandos de Msdos y de Linux tienen alguna semejanza pero algunas cosas que estamos acostumbrados a usar en Msdos no son otra cosa que burdas imitaciones de las extraordinarias posibilidades del interprete de comandos de Linux.

2.1 Instalación

Dentro de este capítulo del Diplomado, se trabajó con la versión de Slackware 9.0. Slackware Linux no requiere de un sistema extremadamente potente para ejecutarse. Este "sabor" de Linux se puede ejecutar en equipos 386 o superiores. Los requerimientos mínimos para instalar y ejecutar Slackware son:

- Procesador 386
- 16 MB en RAM
- 5 MB de espacio libre en disco duro
- Unidad de 3.5"
- Hardware adicional, tarjeta de video para ejecutar X Windows y una tarjeta de red por si se desea tener acceso a algún tipo de red.

Es muy común en una instalación de Linux en máquinas nuevas que no reconozca la tarjeta de video. Para esto es necesario primero ver si la instalación que vamos a realizar es contiene los controladores necesarios para nuestra tarjeta de video, ya que de no ser así no podremos levantar el entorno gráfico. Regularmente al finalizar la instalación editamos el archivo que contiene la configuración de nuestro monitor llamado regularmente **X11/X11.rc** para corregir algunos problemas de resolución de nuestro monitor.

Las versiones actuales de Linux también ya incluyen un kernel que soporta discos SATA, no son todas la versiones pero si ya la mayoría. Tal vez uno de los grandes problemas o deficiencias de Linux es que no hay mucho soporte sobre tecnología nueva, los fabricantes de hardware no "abren" los códigos de sus controladores para que la comunidad del software libre pueda implementarlos con facilidad en las versiones de Linux, habrá que esperar por lo menos 6 meses para que este disponible un controlador para el dispositivo nuevo.

Antes de instalar Slackware debemos seleccionar la series de paquetes que se van a instalar. Linux Slackware es una de las distribuciones de Linux más antiguas, contiene un conjunto de paquetes que se reparten en series y comienzan desde la letra A hasta la Z. Esto trajo consigo un estándar, pues era más rápido conseguir las actualizaciones de Slackware de esta manera, ya que no era necesario descargar todo el software completo de una distribución con paquetes que no se iban a emplear, pero era mucho más rápido y fácil descargar series de paquetes individuales.

Series

A – Es la base, contiene suficiente software para levantar y ejecutar Slackware, contiene algunos editores de texto y programas de comunicaciones.

AP – Varias aplicaciones que no requieren del sistema X.

D – Herramientas de desarrollo de programas. Compiladores, depuradores, intérpretes y los manuales n.

DES – Incluye la función de **crypt()** de **libc** de GNU.

E - GNU Emacs, es tan grande que requiere su propia serie.

F - Contiene FAQ's, HOWTO's y otro tipo de documentación.

GTK – Contiene el ambiente de escritorio de GNOME, biblioteca de widgets de GTK, y el UIMP.

K – El código fuente del núcleo de Linux.

KDE - Contiene el ambiente de trabajo de escritorio de KDE.

N – Contiene programas para configuración de una red. Demonios, programas de correo, telnet, programas de lectura de noticias, etc.

T - Contiene el sistema e formato de documentos teTeX.

TCL – Contiene las herramientas y lenguaje de comandos, elt Tk, el TclX y el TkDesk.

U - Contiene paquetes de programas diseñador específicamente para trabajar solamente en sistemas de UltraSPARC.

X – Contiene la base del sistema X Windows.

XAP – Contiene aplicaciones X que no son parte de un ambiente de escritorio importante. Por ejemplo. Ghostscript y Netscape.

XD – Contiene Bibliotecas, kitde la conexión del servidor, y ayuda de PEX. Para el desarrollo de X11.

XV – Contiene juegos (una colección de juegos de BSD).

Después de arrancar la instalación, Linux detectará la mayor parte del hardware que está instalado en el equipo. El primer paso de la instalación es elegir la configuración de nuestro teclado.

Debemos contestar la pregunta y elegir la opción **"es.map"** pues el tipo de teclado español. El término mapa del teclado se refiere a la ubicación de las teclas en el mismo y dependiendo del idioma éstas tendrán un ubicación diferente.

Enseguida aparecerá una pantalla donde podemos realizar una prueba al teclado escribiendo algo solo para proba la nueva configuración del teclado, se escribe 1 si se está de acuerdo con la configuración y 2 si queremos seleccionar otro mapa.

El siguiente paso es ingresar al programa de instalación (setup) tecleando root cuando aparezca el siguiente mensaje:

```
$ Slackware login :
```

En seguida obtenemos un prompt donde podemos ejecutar una serie de comandos básicos para la instalación de Linux.

Antes de instalar los paquetes, debemos crear las particiones necesarias para la instalación. Para ello, ejecutaremos el comando **fdisk** para poder trabajar con las particiones. Éste programa difiere de la versión de MS-DOS porque es un poco mas robusto, capaz de trabajar y crear particiones tanto para Linux como otros sistemas operativos.

La forma de arrancar **fdisk** es la siguiente: `fdisk /dev/hda`

Linux tiene su propia forma de llamar a los diferentes dispositivos que se encuentran instalados en el equipo como son, discos duros, discos flexibes, unidades de CD-ROM, etc.

Los almacena dentro de un directorio especial llamado **/dev/.** Es aquí donde se encuentran todos los dispositivos, nombres de dispositivos que Linux puede manejar.

Los discos duros en especial son nombrados de acuerdo a su interfaz de conexión y al número de particiones.

En Linux es habitual tener más de una partición , el área de SWAP y el sistema de archivos Linux, esto es la mayoría de los casos. Muchas personas instalan

Windows y Linux en el mismo disco duro, en particiones independientes lo que permite flexibilidad y compatibilidad unidireccional.

Para evitar problemas de seguridad y facilitar el trabajo del administrador, se crean particiones independientes para los sistemas de archivos principales del sistema.

Una vez realizadas las particiones de nuestro sistema, para arrancar el programa de instalación tecleamos la palabra SETUP y aparecerá la pantalla de inicio de instalación. El programa de instalación nos llevará de la mano en éste proceso.

Cuando el programa de instalación detecta particiones que contienen otro sistema operativo, nos pide un punto de montaje para que sea accesible desde Linux, solo debemos indicar la ruta en donde estará montada la partición del directorio

Cuando aparezca la opción de instar lilo, éste se alojará en el MBR (Master Boot Record)

LILO (LInux LOader)

Permite tener un sistema de arranque múltiple: MS- Windows, MS-DOS, **Linux**, etc. Este programa se ubica en el sector de arranque del disco y le permite seleccionar la partición sobre la cual desea arrancar. .

El archivo de configuración LILO se encuentra generalmente en /etc/lilo.conf. Las distribuciones permiten generar un archivo automáticamente.

Se muestra un ejemplo del archivo /etc/lilo.conf en la figura 2:

Al terminar la instalación debemos reiniciar nuestro equipo, seleccionamos EXIT del menú y oprimimos las teclas. ctrl. + alt + supr. Cuando el sistema arranque nuevamente, seleccionamos Linux del menú de arranque.

```
boot = /dev/hda       # disco en el cual se encuentra el archivo LILO
  delay = 300         #  tiempo de espera, antes del lanzamiento del
arranque.
                      # permite  seleccionar la partición  deseada.

  vga = normal        # modo de pantalla a escoger.
  ramdisk = 0         # paranoia setting

  # Linux : Ultima versión :
  image = /vmlinuz    # Núcleo de  Linux 1 (ruta completa y nombre del
archivo)
    root = /dev/hda2
    append="no-hlt, aha1542=0x230"
    label = linux
    read-only

  other = /dev/hda1    # Partición 1 :
    label = dos        # texto que identifica el arranque bajo DOS
    table = /dev/hda   # disco : /dev/hda

  image = /zimage     # Núcleo Linux 2
    root = /dev/hda2  # Raiz de Linux (Partición 2 de mi disco)
    append="aha1542=0x230"
    label = old        # Texto a escribir : old
    read-only          #
```

Figura 2: /etc/lilo.conf

Alta y/o baja de Sistema

En muchos sistemas operativos, no es necesario realizar ningún procedimiento especial de desconexión. El sistema se puede apagar en cualquier momento. Linux no funciona de este modo. Sino que necesita tiempo para cerrar los procesos abiertos, guardar los datos no guardados en el disco. Para apagar el sistema Linux, necesitamos utilizar el comando **shutdown**. Este comando esta especialmente diseñado para desconectar Linux de forma segura.

Durante el proceso de apagado, **shutdown** realiza las siguientes acciones:

- Notifica a los restantes procesos y usuarios que el apagado es inminente.
- Apaga otros procesos que aún se están ejecutando
- Notifica a *root* a medida que se desconecta cada servicio.
- Si así se especifica, reinicia el sistema

Shutdown admite varias opciones de la línea de comandos, estas opciones son:

Opción	Propósito
-c	Se utiliza para cancelar un apagado que ya estaba programado
-h	Se utiliza para forzar un detención de todo el sistema tras apagarse el sistema
-k	Se utiliza para simular un apagado
-r	Se utiliza para forzar un reinicio tras apagarse el sistema
-t segundos	Se utiliza para establecer el tiempo en segundos antes de que shutdown realice su tarea

2.2 Nociones de administración

A continuación se tratarán una introducción sobre la administración del sistema.

2.2.1 Gestión de usuarios

El sistema mantiene información acerca de cada usuario, dicha información es:

nombre de usuario: El nombre de usuario es el identificador único dado a cada usuario del sistema.

clave: El sistema también almacena la clave encriptada [11]del usuario. El comando **passwd** se utiliza para poner y cambiar las claves de los usuarios.

user ID: El user ID, o UID, es un número único dado a cada usuario del sistema. El sistema normalmente mantiene la información por UID, no por nombre de usuario.

group ID: El group ID, o GID, es la identificación del grupo del usuario por defecto.

nombre completo: El nombre real o nombre completo del usuario se almacena junto con el nombre de usuario.

[11] **Encriptación:** Procedimiento de seguridad para codificar mensajes y programas, de modo que no puedan ser leídos o copiados si se desconoce la clave de encriptación

directorio inicial: El directorio inicial es el directorio en el que se coloca inicialmente al usuario en tiempo de conexión. Cada usuario debe tener su propio directorio inicial, normalmente situado bajo /home.

intérprete de comando(shell): El intérprete de comando del usuario es el intérprete de comandos que es arrancado para el usuario en tiempo de conexión.

El archivo /etc/passwd contiene la información anterior acerca de los usuarios. Cada línea del archivo contiene información acerca de un único usuario. El formato de cada línea es

```
nombre:clave_encriptada:UID:GID:nombrecompleto:dir_inicio:intérprete
```

Grupos

Cada usuario pertenece a uno o más grupos por lo tanto habrá que crearlos. La única importancia real de las relaciones de grupo es la perteneciente a los permisos de archivos, pues cada archivo tiene un "grupo propietario" y un conjunto de permisos de grupo que define de qué forma pueden acceder al archivo los usuarios del grupo.
El archivo /etc/group contiene información acerca de los grupos. El formato de cada línea es

```
nombre de grupo:clave:GID:otros miembros
```

2.2.2 Perfil y actividades del administrador

El administrador de sistemas es la persona responsable de configurar, mantener y actualizar el sistema o conjunto de sistemas que forman una red de computadoras, cuidando el funcionamiento de software, hardware y periféricos de forma que estén disponibles para ser utilizados por los usuarios.

Hay que llevar una correcta administración de los recursos del sistema (CPU, memoria y disco), proporcionar una buena atención a los usuarios, monitorear procesos del sistema, detectar fallas a tiempo, y realizar auditorias e implantación

de políticas y medidas de seguridad para con el mismo. Todo buen administrador debe conocer el hardware de su sistema, como son: **características, modelo, capacidad, etc.**.

El tener abiertos los canales de comunicación entre los usuarios y el administrador del sistema es sumamente importante, y mantenerse siempre abierto a sugerencias hace que ésta relación, administrador-usuario, hagan y seguro el sistema, pues unos orillan a otros a realizar bien su trabajo.

Dentro del perfil que un administrador, se debe considerar:

- Conocimientos del sistema
- Técnicas de programación
- Dominio de al menos un lenguaje de programación
- Funcionamiento del sistema operativo
- Técnicas de administración del sistema operativo
- Conocimientos básicos de hardware y mantenimiento de dispositivos.
- Comprensión profunda sobre redirección, tuberías, procesamiento en segundo plano, etc.
- Manejo de **vi**, pues es el común denominador entre los sistemas UNIX, LINUX.

Un sistema Linux puede o no tener levantados servicios, es pues que el administrador sabe que servicios están levantados o que elementos están corriendo en el sistema para no tener así hoyos de seguridad en servicios o puertos que no se requieren que estén activos.

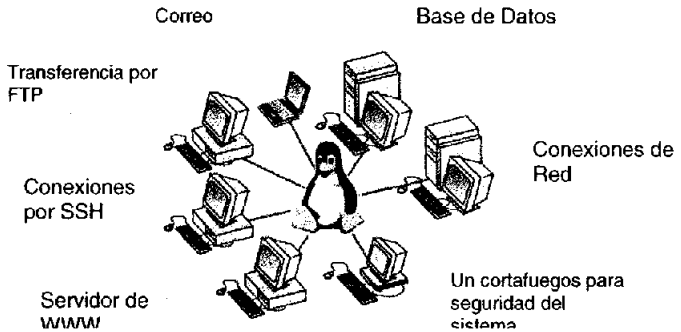


Figura 3.: Servicios WEB

También debe realizar algunas actividades muy comunes, solo por mencionar algunas:

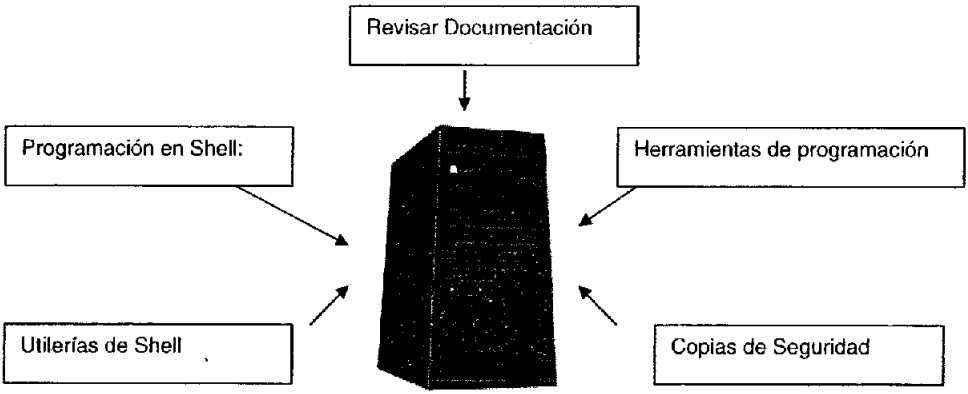


Figura 4. Servidor

Programación en Shell.

- Planear las actividades.
- Guardar copias de seguridad

Es de vital importancia conocer utilerías del sistema como: cut, sort, paste, diff, comm, tail, head, grep, egrep, compress, etc.

Un buen administrador debe tener conocimiento de algunas herramientas de programación:

- c
- shell
- awk
- perl

Algunos puntos imprescindibles de las actividades de un administrador:

- Establecer políticas
- Apertura de cuentas
- Oras de mantenimiento
- Responsabilidad de los respaldos
- Borrado de archivos temporales
- Cuotas de disco
- Seguridad del sistema

Todo el poder administrativo se otorga a root o súper usuario, root controla a los usuarios individualmente a los grupos y los archivos. Una cuenta, en un sentido más específico consta de dos elementos:

- un nombre de usuario y una contraseña
- un directorio inicial

Cuando un usuario intenta iniciar una sesión. Linux comprueba si se cumplen estos requisitos, para lo que examinan el archivo **passwd**. El archivo **passwd** se encuentra en el directorio raíz (/)

2.2.3 Usuarios

Existen varias formas de añadir usuarios:

- Utilizando herramientas gráficas
- Utilizando herramientas de línea de comando (useradd, adduser)
- Modificando el archivo /etc/passwd manualmente

Para eliminar usuarios, lo podemos hacer de las tres formas anteriores.

Grupos

Crear grupos es una labor sencilla, los grupos se almacenan en /etc/group
La estructura de /etc/group/ es parecida a la de /etc/passwd.
Para agregar un grupo, podemos:

- Utilizando herramientas gráficas.
- Utilizando herramientas de línea de comando (addgroup, groupadd).
- Modificando el archivo /etc/group manualmente.

Para eliminar usuarios, lo podemos hacer de las tres formas anteriores. Incluso para eliminar un grupo podemos usar el comando groupdel.
Muchas de las primera versiones de Linux almacenaban las contraseñas de los usuarios en **/etc/passwd** lo que no resultaba seguro, ya que ese archivos es y debe ser legible, De ahí que cualquier usuario pueda ver el contenido de **/etc/passwd.** Las contraseñas de Linux están ecnriptadas, de ésta manera

aunque **/etc/passwd** fuera público la contraseña era ilegible para cualquier usuario curioso.

La base de datos de contraseñas es el archivo shadow que se encuentra en /etc
El **/etc/shadow** es un archivo especial que almacena no solo as contraseñas de los usuarios sino también de indicadores de reglas especiales. Desde varios puntos de vista, **/etc/shadow** se asemeja a **/etc/passwd.** El archivo consta de un registro por línea y cada registro se divide en nueve campos separados por dos puntos (:).

2.3 Dispositivos y Linux

Las distribuciones de Slackware Linux incluye un kernel reciente con todos los capítulos compilados y puede detectar hardware en el momento del arranque, de forma que un administrador o un usuario en general no debe encargarse de esto. Sin embargo puede haber algunos casos excepcionales:
Si se compra hardware diseñado para Windows o que no tenga especificaciones abiertas puede ser difícil usarlo en Linux.

2.3.1 Disquetes

El controlador de Linux soporta unidades de baja, alta y extra alta densidad, puede soportar más de dos y soporta algunos formatos de capacidad extra
Aunque hay varios dispositivos para unidades de disquete (dependiendo del formato), los dispositivos `/dev/fd0`, `/dev/fd1`, etc. autodetectarán el formato. Para acceder a un disquete pueden usarse estos con **mount/umount** o con las herramientas del paquete `mtools`. Si se emplea **mount** es recomendable agregar a

```
/etc/fstab:
/dev/fd0   /floppy   auto   defaults,user,noauto    0   0
```

2.3.2 Unidades de CD-ROM y DVD

Una vez se configure la unidad de CD o DVD se podrán montar CDs o con un programa apropiado podrá escuchar CDs de audio. Los controladores para los tipos más comunes de unidades de CD y DVD también soportan CDs multisesión y algunos soportan lectura de datos digitales.
El kernel 2.2.x como la 2.4.x soportan las unidades de CD-ROM y DVD más estándares IDE/ATAPI y SCSI así como algunas unidades no tan recientes con sus propias interfaces.

2.4 Interfaces gráficas en Linux

Las interfaces gráficas para Linux y para todos los sistemas tipo Unix en general son conocidas como ambientes *X Window*. Para el caso de Linux se utiliza una implementación de *X Window* con código abierto, denominada *XFree86* que también está disponible para otros sistemas tipo Unix.

La arquitectura de un sistema *X* es del tipo cliente-servidor. El servidor se encarga de tomar la entrada de los usuarios y hacerla llegar a las aplicaciones clientes, además de recibir y redireccionar las salidas de estas aplicaciones.
La interfaz principal entre el usuario y un servidor *X* la define el gestor o manejador de ventanas (*Window Manager*). Este se encarga de definir y manipular la apariencia de las ventanas: bordes, menús, botones, barras de desplazamiento, de herramientas y de estado, entre otras múltiples funciones. Existen muchos de estos gestores con diferentes posibilidades y características. Entre ellos podemos citar:

- **FVWM** surgió a partir de TWM (*Tab Window Manager*). Incluye a FVWM, FVWM2 y FVWM95. El término VWM viene de *Virtual Window Manager* y la ``F" según el criterio de sus desarrolladores puede significar: *"Feeble, Fabulous, Famous, Fast, Foobar, Fantastic, Flexible"*

- **WindowMaker**: ofrece una gran cantidad de temas de gran belleza para configurar el *look & feel* del escritorio. La ventaja mayor de su uso es el que consume muy pocos recursos sin afectar las numerosas facilidades que debe brindar todo entorno gráfico. Es muy fácil de usar y configurar.

- **Otros**: IceWM (*Ice Window Maker*), Enlightenment, Blackbox, Sawfish, AfterStep y otros.

Los dos ejemplos clásicos de entornos de escritorio son:

- **KDE** (*K Desktop Environment*).
- **GNOME** (*GNU Network Object Environment*)

Para iniciar una sesión de trabajo en un entorno de escritorio determinado o simplemente en un gestor de ventanas, se puede invocar este directamente desde el *shell* utilizando algún comando. Por ejemplo, para el KDE se utiliza el *shell script* `startkde` y para el GNOME, `gnome-session`.. Existe además un tipo de aplicación gráfica conocida como gestor de "*displays*" (*Display Manager*) que permiten conectarse al sistema Linux directamente a través de un diálogo gráfico. Entre los gestores de "*displays*" más conocidos se encuentran: `xdm`, `kdm` y `gdm`. El primero es la base de todos los demás.

Toda aplicación cliente *X* se basa en un nombre de *display* para identificar al servidor *X* con el cual se conecta. Este nombre tienen la siguiente forma:

```
máquina:display.pantalla
```

Durante los últimos años, las tendencias indican que cada día más administradores de sistemas eligen **Linux**, por su costo reducido, sus ventajas y utilidad, mediante soluciones de Servidores para Intranet e Internet. La ventaja principal de **Linux** se encuentra en lo referido a redes. **Linux** soporta toda la familia de protocolo **TCP/IP** y todos los servicios y aplicaciones que utilizan este último.

Con **Linux** se puede realizar un acceso a Internet para brindar servicios como ssh, telnet, ftp, e-mail o cualquier otro servicio. Con **Linux** como servidor de Intranet es posible recibir información, evaluarla y distribuirla para el uso interno de la empresa o lugar de trabajo. También protege su sistema contra intrusos, mediante la utilización del Sistema Firewall de **Linux** o detectando intentos de intrusión a la red mediante sistemas como **Snort**[+].

Las características de **Linux** lo hacen un sistema con mejores y mayores prestaciones de las que brindan otras tecnologías tales como **NT** o **Novell.**

La administración no es algo tan sencillo. Por ahora dejemos aun lado el sistema operativo Linux y centrémonos en lo que necesitaremos para poder realizar una interfaz de Web, es decir, lo que el navegador presentará al usuario para que pueda interactuar con nuestro sistema Web.

[+] Snort es un sistema de detector de intrusos de red de código libre, capaz de realizar un análisis en tiempo real del tráfico de la red y de paquetes en redes IP.

Capítulo III. Introducción a HTML

El HTML, acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje informático diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, Mozilla o FireFox, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML es un lenguaje para la definición de estilos lógicos en documentos de hipertexto, siendo el medio principal para la diseminación de información en World Wide Web o por sus siglas en inglés WWW. HTML se limita a describir la estructura y el contenido de un documento, nunca el formato de la página y su apariencia, ya que éstos son muy dependientes del visualizador utilizado.

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la Web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos lo que la gente lo utilizaría en un futuro. Con el paso del tiempo se han ido incorporando modificaciones, estos son los estándares del HTML. Sirve para crear los documentos que se publican en la Web, éstos pueden contener imágenes, sonido, texto y/o archivos multimedia. O bien, podemos desarrollar lo que llamamos el FRONT-END[7] para aplicaciones Web.

3.1 Estructura básica de un documento HTML

La primera parte de una página HTML es el identificador del lenguaje. En esta sección se le indica a los navegadores qué tipo de información van a leer. Para el caso del HTML, el identificador es la etiqueta <HTML>...</HTML>. Éstas

[7] Es la interfaz de usuario que aparece en una página Web y permite a los visitantes de un sitio interactuar de manera dinámica con alguna de sus características, incluyendo bases de datos, carritos de compras, compras electrónicas y muchas características más.

deben ser la primera y última etiquetas respectivamente, que aparecen en el documento. Los documentos HTML se dividen en tres partes bien diferenciadas:

1. La Cabecera de tipo de documento. La usa el software para saber la versión de HTML que se está usando (no visible).
2. La Cabecera del documento (HEAD). Usada para dar información sobre el documento (no visible).
3. El Cuerpo del documento (BODY). Es la parte principal del documento, la parte que el usuario ve.

Dentro del área de la cabecera se incluye la etiqueta <TITLE...</TITLE> que indica al navegador el título que contendrá la página, además de que en la actualidad en ésta misma sección se introduce código javascript[12].

3.2 Etiquetas básicas

Etiquetas básica de una página Web
<HTML> Indica el inicio del documento.
<HEAD> Define el inicio de la cabecera.
<TITLE> Inicia el título del documento.
</TITLE> Fin del título del documento.
</HEAD> Define el fin de la cabecera.
<BODY> Inicio del cuerpo del documento.
</BODY> Fin del cuerpo del documento.
</HMTL> Indica el fin del documento.

Título. Es usado como el título por defecto. Para definir el título, se hará dentro de la directiva HEAD y utilizando la directiva

```
<TITLE>...</TITLE>
```

[12] JavaScript es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java.

Párrafos. La mayor parte del texto que se escribe forma parte de un párrafo de texto. Para crear un párrafo, basta con escribir el texto que lo forma dentro de la directiva

```
<P>...</P>
```

Listas. Una lista para HTML no es más que un agrupamiento o enumeración de elementos de información. HTML soporta tres tipos de listas:

1. Listas marcadas
2. Listas enumeradas
3. Listas de definiciones

Texto pre-formateado. A menudo se hace necesario que algún texto aparezca *exactamente* como lo escribimos:

- Mostrando una pantalla tal y como aparecerá
- Haciendo ``tablas''
- Que un texto contenga varios espacios o líneas en blanco consecutivas

La directiva <PRE> es usada para englobar todo aquel texto que se quiere formatear exactamente.

```
<PRE>
        TEXTO con formato, tal y como se desea ver
</PRE>
```

Rompimiento de línea. En numerosas ocasiones es necesario forzar un final de línea en el documento formateado. Para ello, es suficiente con el uso de la directiva simple
.

Enlaces (URL's). Para especificar un enlace, es necesario utilizar el atributo HREF. El atributo HREF (hypertext reference) es un enlace a otro objeto. Para crear un enlace o referencia hipertexto, se utilizará la siguiente plantilla:

```
<A HREF="pagina.html">Texto de enlace. </A>
```

Imágenes. Una de las funcionalidades más llamativas en HTML es la posibilidad de incluir imágenes dentro de un documento. Los formatos más usados son: GIF, JPEG y sus variantes (JPG,BMP,XMP,XBM), TIFF, EPS, o PCX. Para insertar una imagen en un documento HTML se utilizará la directiva simple.

```
<IMG>
<IMG src="/img/red.gif">
```

Tablas Básicas. Las tablas se especificarán siempre por filas; es decir, primero se escribirá la fila 1, después la fila 2, etc. La directiva que se utiliza para delimitar una tabla es <TABLE>...</TABLE>.

Cada fila se especifica con la directiva <TR>...</TR> y, dentro de ella, cada celda se especifica con la directiva <TD>...</TD>

3.3 Elementos avanzados

Formularios. El principio del formulario es: el usuario rellena una serie de campos y los envía a un programa de tratamiento, denominado genéricamente CGI (*Common Gateway Interface*) implementado con las distintas tecnologías que existen para eso, como son: perl, c, php. Asp, etc.

Para iniciar un formulario en HTML se utilizará la directiva <FORM> ... </FORM>. Dentro de ella especificaremos todos los campos que intervienen en el formulario.

Directiva FORM. Con esta directiva especificaremos el tratamiento que se realizará con los campos introducidos por el usuario. Para ello existen dos atributos:

METHOD. Se especifica la forma de pasar los valores de los campos al programa de tratamiento. Admite los valores **GET y POST**.

ACTION. Indicaremos el nombre del programa de tratamiento.

```
<FORM METHOD=post ACTION="cgi-bin/respuesta"> ... </FORM>
```

Directiva INPUT. Es una de las directivas que permite especificar algunas clases de campos de entrada, dependiendo de los atributos asociados:

NAME. Indicaremos el nombre asociado al campo de entrada, debe ser único dentro de un mismo formulario.

VALUE. Dependiendo del tipo o clase de campo indicará una cosa u otra:

- Campo de texto: Predefine el contenido del campo.
- Campo *submit* o *reset*: Texto que aparecerá en el botón.
- Campo *checkbox* o *radio*: Valor asociado a la opción.

TYPE. Indicaremos el tipo o clase de campo. Los valores que puede tomar son:

Texto no específico

SUBMIT. Botón que realiza el envío de la información al programa de tratamiento.

```
<FORM> <INPUT TYPE=submit VALUE="enviar"> </FORM>
```

RESET. Botón de borrado de todos los datos introducidos en el formulario.

```
<FORM> <INPUT TYPE=reset VALUE="borrar"> </FORM>
```

CHECKBOX. Es un campo multivalor ya que permite asignar uno o varios valores a una misma variable.

```
<FORM>
<INPUT      TYPE=checkbox      NAME="so"      VALUE="msdos">MS-DOS<br>
<INPUT      TYPE=checkbox      NAME="so"      VALUE="os2">OS/2<br>
<INPUT      TYPE=checkbox      NAME="so"      VALUE="unix">UNIX<br>
</FORM>
```

RADIO. Es un campo multivalor excluyente, permite escoger una y sólo una opción de un conjunto de valores.

```
<FORM>
<INPUT TYPE=radio name="sexo" value="hombre">Hombre
<INPUT TYPE=radio name="sexo" value="mujer">Mujer
</FORM>
```

PASSWORD. Es una zona de texto donde los caracteres se reemplazan por asteriscos.

```
<FORM>
<input type=password name="passwd">
</FORM>
```

HIDDEN. Permite ocultar al usuario un campo en el formulario dando un valor a la variable asociada.

```
<FORM>
<input type=hidden name="valorh">
</FORM>
```

Directiva SELECT. Se utiliza para definir listas de opciones dentro de un formulario. El atributo NAME se utiliza de la misma forma. Para especificar cada elemento de la lista u opción se utiliza la directiva **<OPTION>** indicando a continuación el texto de la opción.

Directiva TEXTAREA. Permite crear una zona de texto especificando el número de filas (atributo **ROWS**) y de columnas (atributo **COLS**) de la ventana. Si se desea especificar un texto, se hará entre la directiva de apertura y la de cierre.

```
<FORM>
<TEXTAREA NAME="opinion" ROWS=5 COLS=60>
texto:
</TEXTAREA>
</FORM>
```

Se ha comentado que los formularios se basan en la ejecución de un programa de tratamiento o *CGI* (*Common Gateway Interface*), a partir de los datos introducidos por el usuario.

El cliente manda los datos del formulario como una cadena de caracteres de pares **nombre_del_campo=valor**, separados por el caracter **&**. Si un valor contiene espacios en blanco, serán sustituidos por el caracter **+**. Si un campo contiene varios valores se repetirá el par nombre_del_campo=valor, tantas veces como valores tenga el campo.

campo1=valor1&campo2=valor2& ...

Una vez que el cliente a formado la cadena URL, la envía al servidor según el valor especificado para el atributo METHOD, el cual puede ser:

- **GET.** Cuando se utiliza el método GET, la cadena URL de datos del formulario se agrega al nombre del CGI. La separación entre el nombre del CGI y los datos de entrada se realizará con el caracter **?**. Así pues, se formará una cadena URL del siguiente aspecto:

nombre_cgi?campo1=valor1&campo2=valor2&

- **POST.** Es el método más utilizado para el tratamiento de formularios. La cadena URL de datos del formulario se envía al CGI como entrada estándar de datos y se genera la variable de entorno **CONTENT_LENGHT** que tendrá como valor el número de caracteres que forman la cadena URL. Los datos no viajan en la URL como pasaría con el método GET.

Frames. Con los paneles podemos dividir la pantalla de visualización en varias zonas donde cada una puede visualizar un documento diferente. Para ello, la estructura del documento HTML contendrá elementos diferentes a los normales. Se especifican tres nuevas directivas, FRAMESET, FRAME y NOFRAME, y un atributo en las directivas A y FORM, TARGET. En un panel HTML se define el

aspecto de las zonas de visualización y los documentos que se mostrarán en cada una de ellas.

Directiva FRAMESET. Es la directiva (apertura y cierre) que divide la zona en subzonas, bien verticalmente (atributo **COLS**), bien horizontalmente (atributo **ROWS**). Los valores de los atributos **ROWS** y **COLS** están formados por una serie de elementos separados por comas, que indicarán el tamaño de las zonas; el cual, podrá ser especificado en uno de los tres formatos siguientes:

- **num**: Indica tamaño de la zona en pixels.
- **num%**: Indica tamaño de la zona expresada en porcentaje de la zona a dividir.
- **num***: Indica tamaño de la zona expresada en proporción a las restantes. Aquí **num** es opcional.

```
<FRAMESET ROWS="10%,90%">
        ...
        <FRAMESET COLS="*,3*">
        ...
        </FRAMESET>
</FRAMESET>
```

Directiva FRAME. La directiva FRAME se utiliza para caracterizar cada una de las zonas definidas mediante la directiva FRAMESET. Acepta 6 atributos:

- SRC. Indica el URL del documento que debe mostrarse en la zona. Si este atributo no se indica, la zona estará vacía.
- NAME. Se utiliza para poner un nombre a la zona. Dicho nombre será el utilizado para dirigir cualquier enlace de hipertexto. No se pueden poner nombres que se inicien con el caracter "_".
- NORESIZE. No se le indica ningún valor; cuando aparece, no se permite la modificación de la zona con el ratón.
- SCROLLING. Indica si la zona debe o no poseer una barra de desplazamiento, o si se deja gestionar por el visualizador. Los valores posibles son, respectivamente, **yes**, **no** o **auto**.

- MARGINHEIGHT. Indicaremos, en número de pixeles, el tamaño de los márgenes superior e inferior.
- MARGINWIDTH. Indicaremos, en número de pixeles, el tamaño de los márgenes izquierda y derecha.

Con el atributo **TARGET** indicaremos a un enlace de hipertexto (directivas A o FORM) en qué zona se debe visualizar. Para ello, el atributo tomara como valor el NAME de una zona del panel u otro valor preestablecido:

- _blank. El documento se visualizará en una nueva ventana (*navegador*).
- _self. El documento se visualizará en la misma zona donde está el enlace de hipertexto. Este es el valor por defecto si no indicamos el TARGET.
- _top. Indica al visualizador que elimine todos los paneles existentes y ubique el documento en una ventana completa.
- _parent. Si el documento de nivel superior, del que contiene el enlace de hipertexto, es una definición de paneles, dicho valor provocará la visualización del nuevo documento en toda la zona de definición del documento de nivel superior.

HTML puede generar documentos Web de gran calidad, utilizando algunas o muchas de sus etiquetas, dando al documento el formato deseado, pero si lo que necesitamos es realizar toda una aplicación Web compleja, con interacción a bases de datos, compras electrónicas o servicios Web, HTML nos dará la base, es decir, el esqueleto de nuestra aplicación, pues la funcionalidad y manejo de procesos se implementará con otras herramientas que son soportadas perfectamente por HTML.

Ahora que sabemos como desarrollar nuestra interfaz se usuario debemos colocarla en un servidor Web, para ello es que en el capítulo siguiente damos una breve descripción de lo que el servidor Web Apache es.

Capítulo IV. Administración de Servidores WWW con LINUX

Una de las principales herramientas para la administración de los servicios Web es el Servidor Web Apache, basado en el originar servidor web NCSA[Ψ], al que ha sustituido prácticamente en su totalidad.

El servidor Web Apache se configura mediante un archivo de texto que contiene las directivas que le indican las instrucciones sobre como tiene que ser su comportamiento. Normalmente cada distribución de Apache trae un archivo de configuración en el que los cambios que tendremos que realizar son mínimos.

La configuración se especifica en el archivo httpd.conf que está formado por directivas de configuración y línea de comentario que comienzan por un #. Los archivos de configuración contienen una directiva por línea. Se puede usar "\" al final de una línea para indicar que una directiva continua en la próxima línea.

4.1 Módulos

Apache es un servidor modular, esto implica que en el servidor básico se incluyen únicamente las funcionalidades más básicas, otras funcionalidades se encuentran disponibles a través de módulos que pueden ser cargados por Apache. Por defecto, durante la compilación se incluye en el servidor un juego de módulos base. Si el servidor se compila para usar carga dinámica de módulos, entonces los módulos pueden ser compilados por separado, e incluidos en cualquier momento usando la directiva LoadModule. En caso contrario, Apache deberá ser recompilado para agregar o eliminar módulos. Las directivas de configuración se

[Ψ] (National Center for Supercomputing Applications). Se localiza en la Universidad de Illinois en Urbana-Champaign. Creadores de Mosaic, entre otras muchas aplicaciones para Internet.

pueden incluir de forma condicional dependiendo de la presencia de un módulo particular, poniéndolas dentro de un bloque <IfModule>.

Apache permite una administración descentralizada de la configuración, a través de archivos colocados dentro del árbol de páginas web. Los archivos especiales se llaman normalmente .htaccess, pero se puede especificar cualquier otro nombre en la directiva AccessFileName. Las directivas que se pongan dentro de los archivos .htaccess se aplicarán únicamente al directorio donde esté el archivo, y a todos sus subdirectorios. Los archivos .htaccess siguen las mismas reglas de sintaxis que los archivos principales de configuración. Como los archivos .htaccess se leen cada vez que hay una petición de páginas, los cambios en estos archivos comienzan a actuar inmediatamente.

4.2 Instalación del Servidor WWW

Para la instalación del Servidor Web se necesita contemplar los siguientes criterios de selección:

- Función del Servidor de web
- Experiencia de los administradores
- Plataforma disponible
- Numero de conexiones concurrentes
- Numero transacciones por segundo
- Costo computacional por transacción
- Proyección del crecimiento esperado.
- Soporte para la tecnología utilizada para el desarrollo.
- Análisis del retorno de Inversión.

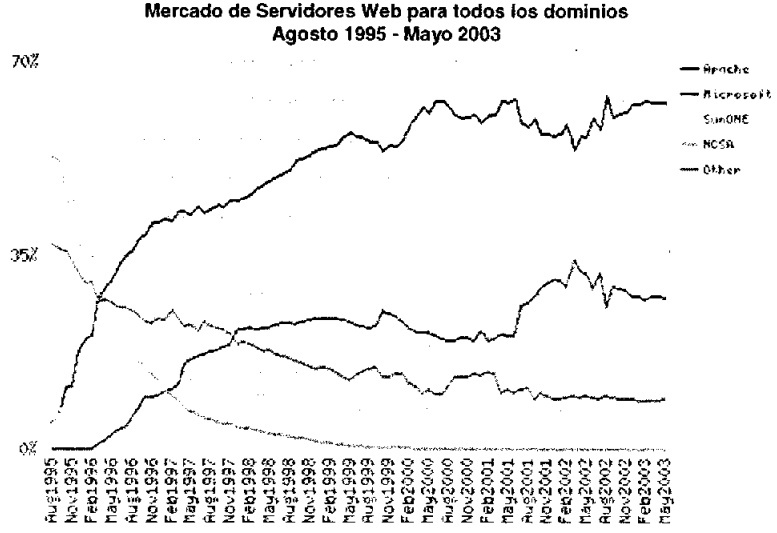
Tomando en cuenta éstos puntos habremos tomado la decisión correcta sobre nuestras necesidades.

Dentro de las ventajas que nos ofrece Apache podemos destacar:

- Robusto, Soporte de un gran numero de transacciones.
- Configurable para diferentes entornos de trabajo.
- Con 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 languages (lenguajes para interpretación de scripts) integrados como módulos (por ejemplo PHP, mod_perl)

- Incluye el código fuente del servidor
- Soporte para accesos restringidos
- Soporte para SSL[13].
- Y además es gratuito.

El uso de Apache se ha difundido enormemente, lo que nos habla de su funcionalidad y potencial, en la siguiente gráfica se muestra su crecimiento



Mayo 2003 se contabilizaron **40,444,778** sitios
Fuente :http://news.netcraft.com/
Figura 5

4.2.1 Instalación de Apache

La instalación de apache en Linux, se puede realizar de dos formas:

- RPM "La forma f"

```
$rpm  -iv apache-1.3.27.rpm
```

[13] SSL (Secure Sockets Layer) es un protocolo diseñado por la empresa Netscape Communications, que permite cifrar la conexión, incluso garantiza la autentificación. Se basa en la criptografía asimétrica y en el concepto de los certificados

– Compilación "La mejor manera"

```
tar zxvf apache-1.3.27.tgz
./configure
make
make install
```

4.2.2 Configuración del servidor

Directivas

Apache es administrado por más de 150 directivas las cuales permiten que determinadas funcionalidad pueda ser incluida. En Linux el administrador controla que directivas estarán disponibles de acuerdo a los módulos con los que se compila.

Grupos de Directivas

La configuración de apache mediante directivas es clasificada en tres grupos:

- Global Enviroment
- Main Server
- Virtual Servers (servidores virtuales)

Global Enviroment. La sección Global Enviroment administra las directivas generales de operación para apache

ServerType. Define el esquema mediante el cual opera apache, las posibles opciones son:

– standalone

– Inetd

User. Esta directiva opera para modo standalone y define el ID del usuario mediante el cual apache operara, para que esto funcione se requiere que apache se arranque como root.

Valor por Default: User #-1

Se puede usar en: Server config, virtual host

Group. Esta directiva opera para modo standalone y define el ID grupo mediante el cual apache operara, para que esto funcione se requiere que apache se arranque como root.

Valor por Default: User #-1
Se puede usar en: Server config, virtual host

Port. Esta directiva define cual es el puerto en que operara el servidor de Web, el valor por default: 80

ServerAdmin. Esta directiva define el correo electrónico del administrador del servidor web.

DocumentRoot. Esta directiva define la ruta absoluta donde se almacenarán los archivos html que se desean publicar. Valor por Default:

/usr/local/apache/htdocs /var/www/html

MinSpareServers. Esta directiva define cual es la cantidad de demonios que estarán corriendo como mínimo. Se monitorea el numero de conexiones en espera, si esta es menor a "MinSpareServers" se levantan los demonios necesarios. Valor por Default: 5

MaxSpareServers. Esta directiva define cual es la cantidad de demonios que estarán corriendo como máximo. Se monitorea el numero de conexiones en espera si esta es mayor a "MaxSpareServers" se eliminan los demonios necesarios. Valor por Default: 5

ErrorDocument . En caso de que un error o problema ocurra con Apache, este puede ser configurado para que haga una de las siguientes cosas:

1. Enviar un mensaje de error (default)
2. Enviar un mensaje personalizado

3. Redirigir a un URL local para manejar el problema o error.
4. Redirigir a un URL externo quien manejara el problema o error.

HostNameLookup. Esta directiva habilita la resolución de nombre cuando se establece una conexión.

HostNameLookup on|off
Default *file*: off

<Directory> y <DirectoryMatch>

```
<Directory dir>
...
</Directory>
```

La directiva Directory permite aplicar otras directivas a un grupo de directorios, es importante comentar que dir se refiere a directorios absolutos.

Options. La directiva **options** controla que características del servidor están disponibles para un directorio en particular.

Options [+|-] *option* [[+|-]*option*]

Puede ser colocado a **option none** en caso de que no se desee ninguna funcionalidad adicional, o tulizar las siguientes opciones:

- All. Todas las opciones se activan excepto MultiViews, (default).
- ExecCGI. La ejecución de ejecución de scripts CGIs es permitida.
- FollowSymLinks. Las ligas simbólicas son validas dentro de este directorio.

Nota: Aún cuando el servidor seguirá las ligas simbólicas este no cambiará la ruta definida por la directiva **<Directory> sections.**

Includes. Tiene las siguientes opciones:

- Includes. Permite el uso de Server-side includes.
- IncludesNOEXEC. Server-side includes son permitidos, pero **#exec commands** y **#exec CGI** son desactivados. Es también posible **#include** virtual CGI scripts desde directorios ScriptAliases.

• Indexes. Si una URL mapea a un directorio solicitado y no hay DirectoryIndex (index.html) en este directorio, entonces el servidor devolverá un listado de directorio.

4.2.3 CGI Common Gateway Interface

Los CGI´s son programas que corren en el servidor, reciben parámetros desde el cliente y su salida es enviada al navegador, permiten generar paginas dinámicas y fueron las primera alternativas para generar dinamismo a un sitio web. Sus principales elementos de configuración son:

ScriptAlias. ScriptAlias /cgi-bin /usr/www/cgi-bin
Se puede usar en: Server config, virtual host. Convierte las solicitudes vía URL al path absoluto donde residen los CGI´s.

AddHandler. Permite que determinada extensión sea relacionada a un evento en particular, en el caso de los CGI´s lo que se indica es que la extensión .cgi queda identificada como un Script.

ScriptAlias. ScriptAlias /cgi-bin /usr/www/cgi-bin

Se puede usar en: Server config, virtual host. Convierte las solicitudes vía URL al path absoluto donde residen los CGI´s.

4.2.4 Accesos Restringidos

Access Control List. A través de la lista de control de acceso podemos indicar que clientes pueden conectarse a nuestro servidor y que es lo que pueden y no ver de nuestro sitio web. Para esto hay diferentes directivas que utiliza Apache.

Order. Permite definir que reglas aplicaran primero para la denegación o acceso a un servicio de Web, tiene las siguientes opciones:

Allow from all|host|env=env-variable [host|env=env-variable]
Deny from all|host|env=env-variable [host|env=env-variable]

AuthUserFile. *AuthUserFile /usr/local/apache/htdocs/respaldos/.htpasswd*
Indica cual es el archivo que contiene la lista de usuarios y passwords para realizar la autenticación.

AuthGroupFile. *AuthGroupFile /dev/null*
Define cual es el archivo que contiene la lista de grupos y los usuarios que la conforman para realizar la autenticación.

AuthName. *AuthName "Nombre del recurso protegido"*
Define como se llamará el nombre de autorización para el recurso protegido, de forma tal, que cuando aparece la caja de dialogo solicitando password pueda ofrecer la clave adecuada.

Limit. Tiene como función definir que tipo de acceso e se tiene hacia el servidor web en determinados recursos.

<Limit GET POST>
</Limit>

Require valid-user. *require valid-user backup*
Es la directiva que permite definir que usuario tendrá acceso al recurso protegido.

4.2.5 Ejecución del Servidor

Se puede Iniciar Apache de dos formas: al poner en marcha el servidor o a través el demonio ***inetd*** cada vez que se solicite una conexión HTTP. La segunda posibilidad es, sin embargo, poco recomendable, porque siempre es mejor tener el servidor activo y listo para que se le solicite algo. El arranque está determinado por un simple script **(inetd)** que existe en la ruta de donde fue instalado apache.

Con la llamada **start** Apache se activa, de forma que se crea el proceso principal (padre) que se pondrá en acción mediante los procedimientos secundarios (hijos) para gestionar las llamadas. La llamada start es la que se solicita al arrancar el servidor.

El comando **httpd start** levantará el demonio de apache. El reinicio tiene un papel bastante importante: si tenemos un servidor utilizado exclusivamente como eso y si hacemos algunas modificaciones a la configuración de Apache con el que trabaja el servidor. Así como existe la orden "start", también existe la orden "restart", pero ésta se tiene que dar a mano a través de la orden: **httpd restart.**

La última orden es apagar que, claramente, para Apache por completo. La orden apagar se puede llevar a cabo de dos formas: la primera es automática cuando se apaga el equipo, es decir, cuando init envía un SIGTERM a todos los procedimientos activos. Por lo que se refiere a Apache, tendría que aparecer algo como: **httpd stop**

4.2.6 Registro de Accesos

Los archivos de log son simples archivos de texto en los que Apache escribirá, en general, los accesos y los errores detectados. Para un control completo en el servidor, y también para depurar scripts, es útil saber leer estos archivos.

Para indicarle a Apache donde debe guardar los archivos de registro (logs) habrá que editar la directiva *ErrorLog* que se encuentra en el archivo de configuración de Apache **httpd.conf**

Como se indica, el lugar señalado es donde Apache guardará sus logs; las especificaciones del directorio se tienen que hacer de esta forma:

```
ErrorLog /var/log/apache/error.log
```

Además de los errores, nos interesamos también en los accesos, en el archivo access.log. Para especificarlo, bastará con tener una línea parecida a:

```
CustomLog /var/log/apache/access.log
```

Es de suma importancia revisar las bitácoras que genera Apache sobre los errores o accesos que se registran y tener un constante monitoreo de los mismos, pues es

gracias a ellos que podemos encontrar accesos no deseados a nuestro sitio, ver los errores que se generaron por cualquier situación fuera de lo normal y saber qué operaciones fueron realizadas por qué usuario y desde qué equipo sobre nuestro servidor.

Además de configurar los que tienen que ser los archivos de log de Apache, podemos también establecer cómo tienen que estar compuestos, cuáles informaciones tienen que contener, lo completos que tienen que estar, etc. Si vamos recorriendo el archivo httpd.conf, notaremos una serie de líneas (bajo ErrorLog) que empiezan por:

```
# LogLevel: Control the number of messages logged to the error_log.
```

A través de este directorio, podemos decidir el nivel de profundidad de los mensajes que se escriben en los archivos error_log; tenemos distintas opciones:

- **Debug**: mensajes útiles en el debug.
- **Info**: informaciones generales.
- **Notice**: condiciones normales, pero significativas.
- **Warn**: atención (warning).
- **Error**: errores generales (del tipo "Premature end of script headers", útiles para la depuración de los script CGI).
- **Crit**: condiciones críticas.
- **Alert**: hay que tomar medidas inmediatamente.
- **Emerg**: emergencia, el sistema no se puede utilizar.

4.3 Manejo de sitios virtuales

Los sitios Web virtuales nos permiten tener varios sitios Web independientes dentro de la misma máquina sin tener que lanzar varias instancias de Apache. Así podemos optimizar recursos. Hay fundamentalmente dos formas de definir sitios virtuales:

Por IP: se identifica a cada servidor por una IP diferente.

Por nombre: el servidor web escucha peticiones en una única IP y es el nombre de la página en que define qué contenido se va a mostrar.

Ésta última forma es la más sencilla y la que más habitualmente se encuentra. También se pueden definir sitios virtuales utilizando otros puertos (por ejemplo para sitios https) y mezclando identificación por IP y por nombre. Los servidores virtuales permiten que un mismo servidor de apache pueda responder a diferentes solicitudes, con lo cual se puede mantener diferentes sitios web.

<VirtualHost nameserver>

</VirtualHost>

Ejemplos de servidores virtuales pueden ser los siguientes:

```
<VirtualHost 10.2.40.5:8097>
    ServerAdmin root@web
    ServerName administracion.hlynur.net
    DocumentRoot /home/hlynur/htdocs/Modulo_Externo
</VirtualHost>
<VirtualHost 10.2.40.5:8088>
    ServerAdmin root@web
    ServerName administracion.hlynur.net
    DocumentRoot /home/hlynur/htdocs/Modulo_Interno
</VirtualHost>
```

Por mucho el servidor Web Apache aporta muchas herramientas y servicios que un servidor de licencia podría aportar. Es seguro y su instalación y configuración es simple y rápida. Esto no quiere decir que su uso quede limitado a personas inexpertas o para tener un servidor casero. Por el contrario, su simplicidad y potencia y robustez lo hacen uno de los mejores en el mercado, y lo más importante, es que sigue siendo gratuito. Continuamente hay mejoras y actualizaciones al encontrarse una vulnerabilidad.

A diferencia de IIS, uno de sus más cercanos competidores, no requiere una licencia, no requiere demasiados recursos de hardware para su instalación y no cuesta ni un peso. No está casado con una sola plataforma, llámese Windows, Linux, Solaris, etc, pues lo mismo corre en un ambiente Windows, como en un ambiente Linux.

El lenguaje PHP corre mucho mejor en una máquina con servidor Linux, y acto seguido detallaremos lo que éste lenguaje de programación puede hacer.

Capítulo V: Introducción al lenguaje PHP

PHP, (acrónimo de 'PHP: Hypertext Preprocessor') es un lenguaje "Open Source"' interpretado de alto nivel, fue creado para desarrollar aplicaciones Internet, llegó a ser muy popular debido a que puede ser inmerso en páginas HTML y tiene una sintaxis fácil de aprender.

PHP4, con un modelo orientado a objetos bastante limitado, ofrecía lo necesario para programar usando algunas de las ventajas de los objetos. Pero el mayor cambio, sin lugar a duda, lo trae PHP5. Con una orientación a objetos bastante parecida a Java, PHP5 presenta mejoras significativas y un ambiente de programación orientada a objetos mucho más completo que permitirán que PHP proporcione un alto rendimiento a las aplicaciones Web empresariales al nivel de las plataformas J2EE y .NET.

5.1 Ventajas de PHP para programar en Web

Algunas ventajas que ofrece PHP son:

- Simplicidad. Su sintaxis está inspirada en C, ligeramente modificada para adaptarla al entorno en el que trabaja
- El nuevo modelo orientado a objetos de PHP5 trae una sintaxis muy parecida a la del lenguaje Java.
- PHP5 presenta las características necesarias para crear toda una estructura de lógica de negocios basada en la orientación a objetos.
- Hay un gran número de desarrolladores y colaboradores, que mantienen al día las actualizaciones del PHP, cualquier error que hubiese es rápidamente corregido..
- PHP es suficientemente versátil y potente como para hacer tanto aplicaciones complejas que necesiten acceder a recursos de bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario llenado por un cliente.

- Si bien es cierto que hay ciertas características avanzadas que presentan las plataformas J2EE o .NET y que PHP no las tiene, no todas las aplicaciones Internet ameritan tal grado de complejidad. PHP fácilmente puede cubrir más del 75% de las necesidades del mercado.
- Hay abundante información, manuales de PHP en más de 25 idiomas. Listas de interés, servidores de noticias, foros, tutoriales de PHP en línea y diferentes canales donde encontrar ayuda.
- Soporte a diferentes motores de bases de datos.
- Envío y recepción de correo electrónico. PHP tiene funciones que permiten enviar y recibir correos electrónicos.

PHP es un lenguaje interpretado que se introduce dentro del HTML, como JavaScript, pero con una gran diferencia, el PHP se interpreta en el servidor y JavaScript el la maquina del cliente. Mediante PHP podemos trabajar con formularios, cookies y su mayor potencia es el trabajo con Bases de Datos, aparte de numerosas características como el trabajo con sockets y distintos protocolos como el manejo de e-mails.

5.2 Sintaxis

Debemos diferenciar el código HTML del código PHP y para realizar esta diferencia tenemos que abrir el código PHP con la etiqueta <?php y cerrarlo con ?>, también existen otros métodos pero para esto hay configurar una serie de parámetros. A continuación se muestra un breve ejemplo de código PHP. Al finalizar cada instrucción se debe agregar el caracter ';' como en C.

```
<?php echo "Esto es una prueba";?>
```

Como todo lenguaje tiene la posibilidad de agregar comentarios, a continuación se muestran las formas de realizar los comentarios:

```
<?php
echo "Hola a Todos!!!"; // Esto es un comentario tipo c++
/* Comentario multilínea
```

```
para varias lineas de codigo*/
 echo "Saludos!!!"; # Comentario estilo shell de unix
?>
```

5.2.1 Variables

Las variables no tienen un tipo definido, si se quiere forzar a que una variable tenga un tipo de datos especifico se tiene que utilizar la función **settype()** que sirve para establecer el tipo de una variable. La forma de escribir una variable es mediante el signo '$' seguido del nombre de la variable, ejemplo:

```
<?php
$numero = 4;
$cadena = "hola a todos";
?>
```

5.2.2 Arrays (Arreglos)

PHP también soporta arreglos su forma de implementarlos es la siguiente:

```
<?php
$a[0] = "hola";
$a[1] = "adios";
$b["primero"] = 1;
?>
```

Aparte de arreglos escalares (0, 1, 2, ...) pueden crearse arreglos asociativos, en lugar de un número que identifique a un elemento del arreglo pueden ponerse cadenas. Otra forma de ingresar datos a un arreglo es la siguiente:

```
<?php
$c[] = 0;
$c[] = 1;
$c[] = 2;
?>
```

Asignar un valor a un arreglo sin indicar el índice hace que ese elemento se agregue al final del arreglo. También se pueden crear arreglos multidimensionales,

```
<?php
$c[1][1] = 0;
$d["uno"][2] = 1;
$f[3]["uno"][4][1] = 2;
?>
```

5.2.3 Constantes

La forma de declarar una constante es a través de la palabra clave define, ejemplo:

```
<?php
define("NOMBRE", "Milton.");
echo NOMBRE; // muestra "Milton."
?>
```

5.2.4 Operadores

Los operadores son componentes esenciales de cualquier lenguaje de programación. Con ellos podemos asignar, unir, cambiar o comparar valores de datos, cambiar el flujo del programa, etc. Los operadores son **símbolos** que representan operaciones sobre un valor.

Operador de asignación. El símbolo = permite asignar valores a variables:

```
<?php
$ciudad = "madrid";
$mi_numero = 123;
$var1 = var2 = 0;
?>
```

Operador de concatenación. Usando el símbolo del . podemos unir valores:

```
<?php
$nombre = "milton";
$apellido = "rubio";
```

```
$id = $nombre." ".$apellido; // suma los dos valores mas un espacio
?>
```

Operadores aritméticos. Los simbolos + - / * realizan las operaciones de suma, resta, división y multiplicación. El símbolo % permite hallar el resto de una división.

Operadores de incremento – decremento. Los símbolos ++ y -- aplicados a una variable, permiten incrementar o decrecer su valor. Su efecto es distinto según se empleen precediendo o siguiendo el nombre de la variable:

`++$a` Incrementa $a en uno y después devuelve $a.

`$a++` Devuelve $a y después incrementa $a en uno.

`--$a` Decrece el valor de $a en uno y después devuelve $a.

`$a--` Devuelve $a y después decrece su valor en uno.

Operadores de Comparación.

$a == $b (igualdad). Cierto si $a es igual a $b.

$a === $b (identidad). Cierto si $a es igual a $b y si son del mismo tipo (sólo PHP4)

$a != $b (desigualdad). Cierto si $a no es igual a $b.

$a < $b (menor que). Cierto si $a es estrictamente menor que $b.

$a > $b (mayor que). Cierto si $a es estrictamente mayor que $b. $a <= $b (menor o igual que). Cierto si $a es menor o igual que $b.

$a >= $b (mayor o igual que). Cierto si $a es mayor o igual que $b.

Operadores Lógicos.

$a and $b **Y**: Cierto si tanto $a como $b son ciertos.

$a or $b **O**: Cierto si $a o $b son ciertos.

$a xor $b **O exclusiva**: Cierto si $a es cierto o $b es cierto, pero no ambos a la vez.

! $a **Negación**: Cierto si $a no es cierto.

$a && $b **Y**: Cierto si tanto $a como $b son ciertos.

$a || $b **O**: Cierto si $a o $b son ciertos.

5.2.5 Funciones de usuario en PHP

Una función de usuario en PHP no es mas que una porción de código que podemos llamar en cualquier momento. De un lado, al dividir nuestro código en funciones podemos aislar y perfeccionar cada una de sus funcionalidades; de otro lado, podemos reutilizar este código, ya que una función puede ser llamada cuantas veces la necesitemos. Para definir una función debemos usar la palabra reservada **function**. Para llamar la función basta con invocar su nombre.

```
<?php
function ligas () {
            // Entre parentesis () podemos incluir valores que
            // deba usar nuestra función. El uso de ()
            // es siempre necesario.
            // El código de la función entre llaves {}
    }
ligas();      // esta linea llama la función
?>
```

Los paréntesis permiten pasar valores a las funciones y son siempre necesarios, aunque no se vaya a pasar valor alguno.

5.2.6 Funciones y parámetros

Se puede pasar a las funciones valores para que operen sobre ellos. PHP permite pasar a las funciones variables, cadenas de texto, números o arreglos. Los datos pasados a la función pueden ser *por valor* o *por referencia*. En el primer caso solo pasamos el valor del dato, no su contenedor, de forma que dicho dato (por ejemplo almacenado en una variable) conserva su valor original fuera de la función, sin verse afectado por los cambios que la función pueda ejecutar. En cambio si pasamos el valor por referencia, el dato original quedará afectado por el resultado de la función, se puede especificar por **Return.** Si la función debe retornar algún valor al script, se debe indicar por medio de la expresión **return.** El código existente en la función después de **return** *no* será ejecutado.

5.2.7 Control de flujo de programa

Todo lenguaje de programación dispone de órdenes de control de flujo, que permite al programa *tomar decisiones lógicas* según reciba unos parámetros o otros. Las posibilidades que ofrece php son:

- if/else
- if/elseif/else
- switch
- do/while
- while
- for

5.3 Herramientas elementales

Muchas son las **funciones** propuestas por PHP para el tratamiento de arreglos, La siguiente tabla muestra las funciones más comunes

Función	Descripción
array_values (mi_array)	Lista los valores contenidos en mi_array
asort(mi_array) y arsort(mi_array)	Ordena por orden alfabético directo o inverso en función de los valores
count(mi_array)	Nos da el numero de elementos de nuestro array
ksort(mi_array) y krsort(mi_array)	Ordena por orden alfabético directo o inverso en función de las claves
list ($variable1, $variable2...)=mi_array	Asigna cada una variable a cada uno de los valores del array
next(mi_array), prev(mi_array), reset(mi_array) y end(mi_array)	Nos permiten movernos por dentro del array con un puntero hacia delante, atras y al principio y al final.
each(mi_array)	Nos da el valor y la clave del elemento en el que nos encontramos y mueve al puntero al siguiente elemento.

De gran utilidad es también el bucle **foreach** que recorre de forma secuencial el arreglo de principio a fin. El siguiente ejemplo ilustra muy bien este concepto. Se quiere almacenar dentro de una misma tabla el nombre, moneda y lengua hablada en cada país. Para hacerlo podemos emplear un arreglo llamado país que vendrá definido por estas tres características (claves). Para crearlo, deberíamos escribir

una expresión del mismo tipo que la vista anteriormente en la que meteremos un arreglo dentro del otro. Este proceso de incluir una instrucción dentro de otra se llama anidar:

```
<?
$pais=array
(
"mexico" =>array
  (
  "nombre"=>"Mexico",
  "lengua"=>"Castellano",
  "moneda"=>"Peso"
  ),
"francia" =>array
  (
  "nombre"=>"Francia",
  "lengua"=>"Francés",
  "moneda"=>"Franco"
  )
);
echo $pais["mexico"]["moneda"] //Saca en pantalla: "Peso"
?>
```

Manejo de Cadenas

Una de las variables más corrientes a las que tendremos que hacer frente en la mayoría de nuestros scripts son las cadenas, que no son más que información de carácter no numérico (textos, por ejemplo).

Para asignar a una variable un contenido de este tipo, lo escribiremos entre comillas dando lugar a declaraciones de este tipo:

```
$cadena="Esta es la información de mi variable";
```

Si queremos ver en pantalla el valor de una variable o bien un mensaje cualquiera usaremos la instrucción *echo* como ya lo hemos visto anteriormente:

```
echo $cadena //sacaría "Esta es la información de mi variable"

echo "Esta es la información de mi variable" //daría el mismo resultado
```

Podemos yuxtaponer o concatenar varias cadenas poniendo para ello un punto entre ellas:

```
<?
$cadena1="Perro";
$cadena2=" muerde";
$cadena3=$cadena1.$cadena2;
echo $cadena3 //El resultado es: "Perro muerde"
?>
```

Para poder imprimir caracteres especiales como el signo de $, " , \ o numeros fraccionarios, lo que hay que hacer es escribir una diagonal invertida delante:

Carácter	Efecto en la cadena
\$	Escribe dólar en la cadena
\"	Escribe comillas en la cadena
\\	Escribe contrabarra en la cadena
\8/2	Escribe 8/2 y no 4 en la cadena

Además, existen otras utilidades de esta diagonal invertida que nos permiten introducir en nuestro documento HTML determinados eventos:

Carácter	Efecto en la cadena
\t	Introduce una tabulación en nuestro texto
\n	Cambiamos de línea
\r	Retorno de carro

Estos cambios de línea y tabulaciones tienen únicamente efecto en el código y no en el texto ejecutado por el navegador. En otras palabras, si queremos que nuestro texto ejecutado cambie de línea hemos de introducir un *echo "
"* y no

echo "\n" ya que este ultimo sólo cambia de línea en el archivo HTML creado y enviado al navegador cuando la página es ejecutada en el servidor.
Las cadenas pueden asimismo ser tratadas por medio de funciones de todo tipo.. Existen muchas posibles acciones que podemos realizar sobre ellas: Dividirlas en palabras, eliminar espacios sobrantes, localizar secuencias, remplazar caracteres especiales, etc.

5.4 Inclusión de Archivos

Las construcciones **include** y **require** son de las más conocidas en php. Con ellas se puede reutilizar porciones de código (script, o simple html) cuantas veces se quiera, siendo uno de sus usos mas sencillos y típicos el de incluir cabeceras y pies de páginas en un sistema de plantillas.

Include. La sentencia include() inserta y evalúa el archivo especificado. Se puede incluir aquí no solamente un archivo en el servidor, sino una página web remota (indicando la url). Su uso típico sería <?php include ("header.php");?> , que llama al archivo header.php y lo inserta en el propio punto del script donde hacemos la llamada.

Require. La diferencia *documentada* entre include y require consistiría en que la llamada con include podría hacerse condicionalmente. A diferencia de include(), require() siempre leerá el archivo referenciado, incluso si la línea en que está no se ejecuta nunca. Si se quiere incluir condicionalmente un archivo, se usa include(). La sentencia condicional no afecta a require(), aunque si la línea en la cual aparece el require() no se ejecuta, tampoco se ejecutará el código del archivo referido.

Include_once y require_once. Estas construcciones presentan como única diferencia que la inclusión del archivo se ejecuta una sola vez (aunque a lo largo de la ejecución del script existan otras llamadas al mismo), lo que es útil para evitar conflictos con redefiniciones de funciones o nombres de variables.

5.5 Entrada de datos a partir de formularios

Los formularios no forman parte de PHP, sino del lenguaje estándar de Internet HTML. Todo formulario comienza con la etiqueta <FORM ACTION="pagina.php" METHOD="post/get">. Con ACTION indicamos el script que va procesar la información que recogemos en el formulario, mientras que METHOD nos indica si el usuario del formulario va ha enviar datos (post) o recogerlos (get). La etiqueta <FORM> indica el final del formulario. A partir de la etiqueta <FORM> vienen los campos de entrada de datos que pueden ser algunos de los ya mencionados en el **Capítulo III: Introducción a HTML**:

Todos lo tipos de campo tienen un modificador llamado **name**, que no es otro que el nombre de la variable con la cual recogeremos los datos en el **script** indicado por el modificador ACTION de la etiqueta FORM, con **value** establecemos un valor por defecto.

5.6 Cookies

Las cookies es información almacenadas por un sitio Web en el disco duro del usuario. Esta información es almacenada en un archivo tipo texto que se guarda cuando el navegador accede al sitio Web. Su utilidad principal es la de poder identificar al navegador, una vez éste visite el sitio por segunda vez y así, en función del perfil del cliente dado en su primera visita, el sitio puede adaptarse dinámicamente a sus preferencias (idioma, colores de pantalla, formularios rellenados total o parcialmente, redirección a determinadas páginas...).

Para crear un archivo cookies, modificar o generar una nueva cookie lo podemos hacer a partir de la función SetCookie:

```
setcookie("nombre_de_la_cookie", valor, expiracion);
```

Es importante que la creación de la cookie sea previa a la apertura del documento

HTML, en otras palabras, las llamadas a la función setcookie() deben ser colocadas antes de la etiqueta HTML.

Es interesante señalar que el hecho de definir una cookie ya existente implica el borrado de la antigua. Del mismo modo, el crear una primera cookie conlleva la generación automática del archivo texto.

5.7 File Upload (carga de archivos a un servidor)

En el caso de subir archivos a un servidor Web por HTTP y a través de una página con un formulario, donde se permite seleccionar el archivo que queremos cargar de nuestro disco duro, se puede hacer con una de las funciones que incluye PHP.

Una página de envío de archivos se puede crear mediante un formulario parecido a éste:

```
<form enctype="multipart/form-data" action="_URL_" method="post">
<input type="hidden" name="MAX_FILE_SIZE" value="1000">
Send this file: <input name="userfile" type="file">
<input type="submit" value="Send File">
</form>
```

La _URL_ debe tener como destino un script PHP. El **input** oculto MAX_FILE_SIZE debe encontrarse antes del **input** de tipo "**file**" para indicar el tamaño máximo de archivo que se puede enviar en bytes.

La variable **$HTTP_POST_FILES/$_FILES** contendrá la información sobre el archivo recibido.

A continuación se describe el contenido de ***$HTTP_POST_FILES***. Se ha tomado el nombre 'userfile' para el archivo recibido tal y como se usaba en el script de ejemplo anterior:

- ***$HTTP_POST_FILES['userfile']['name']*** El nombre original del archivo en la máquina cliente.

- ***$HTTP_POST_FILES['userfile']['type']*** El tipo mime del archivo (si el navegador lo proporciona). Un ejemplo podría ser *"image/gif"*.
- ***$HTTP_POST_FILES['userfile']['size']*** El tamaño en bytes del archivo recibido.
- ***$HTTP_POST_FILES['userfile']['tmp_name']*** El nombre del archivo temporal que se utiliza para almacenar en el servidor el archivo recibido.

Por defecto, los archivos serán almacenados en el directorio temporal del servidor, a no ser que se especifique otra localización con la directiva upload_tmp_dir en `php.ini`. El directorio temporal por defecto del servidor puede ser modificado cambiando el valor de la variable de entorno *TMPDIR* en el contexto en que se ejecuta PHP.

En principio, PHP es un producto Open Source, por lo tanto podemos usarlo en proyectos comerciales si queremos, sin tener que pagar por los licenciamientos. El que sea Open Source trae además muchas otras ventajas.

Hay que tener en cuenta el costo del hardware. Para desarrollar en PHP no se requiere tener grandes capacidades de hardware, como sí lo requieren los pesados IDEs para programar en Java o .Net. Luego, en el caso de los servidores, una aplicación en PHP no requiere tantos recursos máquina como podría requerir una aplicación en Java con sus servidores de aplicaciones que podrían requerir hasta varios procesadores y varios Gigas de memoria RAM..

Ningún sistema estaría completo sin una capa de datos, para cumplir con este requisito el siguiente capítulo describirá la interacción de PHP con MySQL.

Capítulo VI. PHP con MySQL

Las bases de datos son el mejor recurso en la web para que los visitantes de un sitio en Internet puedan consultar y/o adicionar información específica en un gran listado de información.

Una de las ventajas de utilizar la web para este fin, es que no hay restricciones en el sistema operativo que se debe usar, permitiendo la conexión entre si, de las páginas Web desplegadas en un navegador que funciona en una plataforma, con servidores de bases de datos alojados en otra plataforma. Además, no hay necesidad de cambiar el formato o estructura de la información dentro de las bases de datos. Pero no todo es ventaja, la principal desventaja, es que siempre hay alguien malicioso que explota alguna vulnerabilidad de nuestro SMBD y la información que tenemos almacenada se ve implicada en riesgos y tentada a perder, dependiendo del ataca del sujeto, integridad o veracidad.

Es de vital importancia conocer las vulnerabilidades de cada sistema y así saber cómo actuar para minimizar el riesgo a la exposición de nuestros sistemas.

6.1 Manejo de Formularios como FRONT-END[14]

El lenguaje HTML consta de una serie de etiquetas que permitirán crear de forma rápida y sencilla numerosos elementos de entrada de datos como se vio anteriormente. Estos elementos, que reciben el nombre de controles, serán gráficos en la mayoría de navegadores. Un formulario no es más que un conjunto de estos controles cuya información será enviada conjuntamente cuando el usuario pulse sobre el botón de envío.

[14] Elementos de un sitio con los que tiene contacto el usuario final y que influyen en su experiencia al navegar por el sitio. Interfaz de usuario, frontal.

6.2 Instalación y configuración de la base de datos en Linux

Comparativa entre bases de datos para Linux

PostgreSQL Vs. MySQL

POSTGRESQL

- Postgres es un sistema de bases de datos de mayor nivel, al nivel Oracle, Sybase o Interbase.
- Consume bastantes recursos y carga más el sistema.
- Soporta transacciones y desde la versión 7.0, llaves foráneas (integridad referencial).
- Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.

MYSQL

- Su principal objetivo de diseño fue la VELOCIDAD.
- Consume muy pocos recursos, tanto de CPU como de memoria.
- No considera las llaves foráneas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.
- Se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".

MySql puede ser descargado gratuitamente de: http://dev.mysql.com/downloads/

6.3 Instalación y configuración

Es mucho más fácil y seguro instalar MySql a través de los códigos fuentes. Así pues que una vez teniendo la distribución que nos acomode, procederemos a la instalación de la Base de Datos:

Para nuestra instalación utilizamos la distribución binaria que se encuentra bajo Standard binary (tarball) distributions bajo el enlace:

```
Linux (Intel libc6 systems) [pc-linux-gnu-i686]
```

Antes de instalar MySQL es necesario ejecutar los siguientes comandos:

Añadir el grupo mysql en el sistema:

groupadd mysql

Si no funciona este comando se intenta:

/usr/sbin/groupadd mysql

Añadir el usuario mysql en el sistema:

useradd -g mysql mysql

Si no funciona se intenta:

/usr/sbin/useradd -g mysql mysql

Después de copiar la distribución en el equipo se ejecutan los siguientes pasos.

Descompactar el programa en el directorio /usr/local:

```
# cd /usr/local
```

Cambiar **/ruta/** por la ubicación donde se copió MySQL en el equipo:

```
# tar zxvf /ruta/mysql-3.23.34-pc-linux-gnu-i686.tar.gz
```

Crear un enlace simbólico del directorio **mysql** a la versión instalada:

```
# ln -s mysql-3.23.34-pc-linux-gnu-i686 mysql
```

Ejecutar los comandos (*scripts*) de instalación de MySQL. La ejecución de este comando creará las tablas necesarias para definir los permisos de accesos por parte de los usuarios a las bases de datos:

```
# cd mysql
# scripts/mysql_install_db
```

Asignar los permisos a los directorios:

```
# chown R root /usr/local/mysql
# chown R mysql /usr/local/mysql/data
# chgrp R mysql /usr/local/mysql
# chown R root /usr/local/mysql/bin/
```

Para iniciar MySQL cuando se reinicie el equipo:

```
# cd /usr/local/mysql/support-files/
# cp mysql.server /etc/rc.d/mysql
# cd /etc/rc.d/
# chmod 755 mysql
```

Finalmente se coloca en el archivo **/etc/rc.d/rc.local** una línea al final de este archivo así:

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

También se puede detener el servidor ejecutando:

```
# /etc/rc.d/mysql stop
```

Ahora es importante definir las claves de acceso del administrador (usuario *root*) para MySQL.

```
# bin/mysqladmin u root p password 'nuevaclave'
# bin/mysqladmin u root h 'nombreservidor' p password 'nuevaclave'
```

Con esto quedará instalado MySQL en el equipo.

6.4 Introducción a SQL

El lenguaje **SQL** está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Existen dos tipos de comandos **SQL**:

- Los DLL que permiten crear y definir nuevas bases de datos, campos e índices.
- Los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

6.4.1 Comandos DLL

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

6.4.2 Comandos DML

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

n6.4.3 Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Comando	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

6.4.5 Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

6.4.6 Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
>=	Mayor ó Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

6.5 Introducción al desarrollo de CGIs

El **CGI** (Por sus siglas en inglés **"Common Gateway Interface"**) cambio la forma de manipular información en la web. En sí, es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la

de añadir una mayor interacción a los documentos web que por medio del HTML se presentan de forma estática. El **CGI** es utilizado comúnmente para contadores, bases de datos, motores de búsqueda, formularios, generadores de e-mail automático, chats, comercio electrónico y mapas de imágenes, juegos en línea y otros.

Los programas que maneja el **CGI** pueden estar compilados en diferentes lenguajes de programación. El más popular para el desarrollo de contenidos Web es PHP, aunque también podemos mencionar: C, C++ y Java. El funcionamiento de esta tecnología es muy sencillo. Los scripts residen en el servidor, donde son llamados, ejecutados y regresan información de vuelta al usuario.

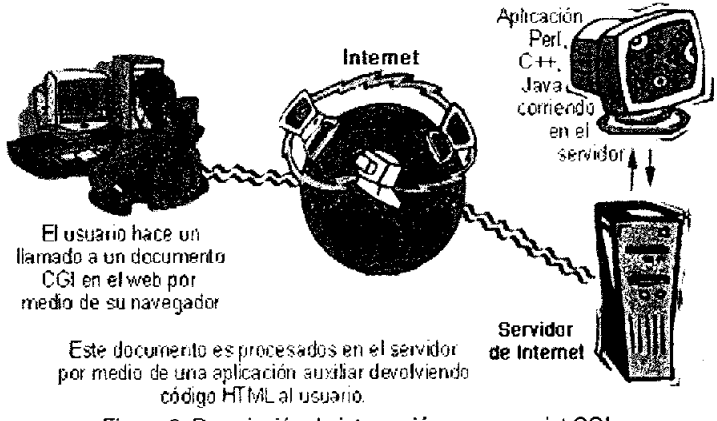


Figura 6. Descripción de interacción con un script CGI

Hay que tomar muy en cuenta que existen dos formas de enviar información; siendo estas por medio de **GET** y **POST** vistos anteriormente.

6.6 Desarrollo del back-end [15] con MySql

La relación entre PHP y bases de datos, especialmente MySQL y Postgres, es muy estrecha y beneficiosa. De hecho, cuando se habla de Web y PHP, es muy

[15] Todos los elementos que involucran la administración de operaciones de un sitio, desde tecnología hasta la integración de sistemas.

difícil que no se mencione también a una base de datos. Mientras mayor sea la cantidad de información y más alta la frecuencia de actualización de un sitio web, mayor es su valor y sus ventajas sobre otros medios.

Tal vez la mayor ventaja de PHP sobre sus competidores es la integración con los sistemas de bases de datos y el soporte nativo a las distintas bases de datos existentes, libres y comerciales. Las razones principales para usar una base de datos son:

- Evitar redundancias.
- Evitar programas complicados.
- Búsquedas.
- Seguridad.
- Arquitectura *n-tier*[16]

6.6.1 Acceso a Base de Datos

Para conectarse al servidor, usualmente necesitamos de un nombre de usuario (login) y de una contraseña (password), y si el servidor al que nos deseamos conectar está en una máquina diferente de la nuestra, también necesitamos indicar el nombre o la dirección IP de dicho servidor. Una vez que conocemos estos tres valores, podemos conectarnos de la siguiente manera:

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

Cuando ejecutamos este comando, se nos pedirá que proporcionemos también la contraseña para el nombre de usuario que estamos usando. Si la conexión al servidor MySQL se pudo establecer de manera satisfactoria, recibiremos el mensaje de bienvenida y estaremos en el prompt de mysql:

[16] Termino referido al uso de una arquitectura de ***n*** capas lógicas para un sistema.

```
shell>mysql -h localhost -u root -p

Enter password: ******

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

Your MySQL connection id is 5563 to server version: 3.23.41

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

mysql>
```

Este prompt nos indica que mysql está listo para recibir comandos. Después de que nos hemos conectado de manera satisfactoria, podemos desconectarnos en cualquier momento al escribir "quit", "exit", o presionar CONTROL+D.

La mayoría de los ejemplos siguientes asume que estamos conectados al servidor, lo cual se indica con el prompt de mysql.

6.6.2 Crear una Base de Datos

Es muy fácil crear una base de datos en MySql, solo necesitamos ejecutar el siguiente comando.

```
mysql> CREATE DATABASE miBase;

Query OK, 1 row affected (0.00 sec)
```

El sistema regresará un mensaje como el anterior indicando que la Base de Datos fue creada.

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

```
mysql> USE miBase

Database changed

mysql>
```

6.6.3 Agregar Información a la Base de Datos

Para insertar datos en la base de datos realizamos lo siguiente:

Insert

```
INSERT INTO tabla (col1, col2, col3) VALUES (val1, val2, val3);
INSERT INTO tabla VALUES (val1, val2, val3);

insert into secciones values(1, 'Primera');
insert into secciones values(2, 'Segunda');
```

6.6.4 Modificación de los datos de las Bases de Datos

Update

Se utiliza para modificar datos previamente almacenados en la base de datos.

```
UPDATE table SET campo1='valor1, campo2='valor2'
WHERE condición
update articulos set seccion=1
  where id=2;
```

6.6.5 Eliminar información de la Base de Datos

Delete

```
DELETE FROM tabla WHERE condición;
```

Es muy importante especificar la cláusula *where*, de otra forma se borrarán todos los datos almacenados en la tabla especificada.

6.6.7 Consultar la Base de Datos

Select

SELECT es el comando principal para obtener información de una base de datos. La sintaxis es muy sencilla:

```
SELECT campo1, campo2 FROM tabla WHERE condición;
```

El usar CGI's nos permite recoger información en una página y enviarla al servidor: por ejemplo, hacer una encuesta a nuestro visitante, o solicitar la introducción de un *login* y una contraseña para darle acceso a cierta parte el sitio.

Con dichas características podemos:

- Presentar contenidos de forma dinámica en una página: almacenar contenidos en una base de datos y mostrarlos de forma dinámica, en función de ciertos datos introducidos por el visitante en un formulario.
- Actualmente, el uso de los CGIs ha decaído con la llegada de nuevas tecnologías (como el lenguaje Java y los ASPs). Sin embargo, para determinados propósitos son todavía el modo más simple y práctico (ejemplos no faltan: contadores, libros de visitas, envío de datos de un formulario,...).

Éste tipo de cosas anteriormente eran imposibles sin ésta tecnología y tal vez hoy en día es casi imposible imaginar un sitio web completamente estático, sin conexiones a base de datos, etc, y cada día es mucho más fácil realizar aplicaciones Web o sitios dinámicos utilizando CGI's, pues hay mucho disponibles en la red de forma gratuita, pero también estamos de acuerdo en que uno se puede basar en ese tipo de cuestiones si va a desarrollar toda una aplicación mucho más robusta, compleja y seria.

El manejador de base de datos es lo de menos, tal vez, pues PHP brinda soporte para varios de ellos, ya será cosa del desarrollador elegir la más conveniente, como en éste caso MYSQL.

Capítulo VII. Introducción a la Seguridad en Cómputo

En términos generales, la seguridad puede entenderse como aquellas reglas técnicas y/o actividades destinadas a prevenir, proteger y resguardar lo que es considerado como susceptible de robo, pérdida o daño, ya sea de manera personal, grupal o empresarial. En este sentido, es la información el elemento principal a proteger, resguardar y recuperar dentro de las redes empresariales.

Las redes fueron diseñadas para el intercambio de información y compartir recursos y en aquel entonces, la seguridad no era un factor tomado en cuenta en el diseño de las redes, pues no se tenía una idea de las dimensiones que éstas llegarían a tener en un futuro no muy lejano.

Desgraciadamente, en el gran crecimiento de las redes, también se incluyó a individuos deshonestos que, al ser una tecnología "nueva", abusaron de éstas situaciones para introducirse a los sistemas de información. Fue hasta entonces que las personas empezaron a ver que se tenían grandes deficiencias en la seguridad de sus sistemas y, por lo tanto, se empezaron a desarrollar los primeros pasos para seguridad informática y consecuentemente la seguridad en redes.

Así pues, los datos deberían ser protegidos de algún daño durante su tratamiento normal, manipulación, transferencia y almacenamiento y los procesos habituales deben ser conservadores con los datos y no destruir información, salvo cuando ésta es su misión expresa.

En la década de los años 90 proliferan los ataques a sistemas informáticos, aparecen los virus y se toma conciencia del peligro que nos acecha como usuarios de PCs[17] y equipos conectados a Internet. Las amenazas se generalizan a finales

[17] PCs iniciales de la palabra inglesa PERSONAL COMPUTER, Ésta denominación se refiera al plural de computadora personal

de los 90. En los años 00s los acontecimientos fuerzan a que se tome en serio la seguridad informática.

Principalmente por el uso masivo de Internet, el tema de la protección de la información se ha transformado en una necesidad y con ello se populariza la terminología técnica asociada a la criptología: Cifrado, descifrado, criptoanálisis, firma digital, etc. Autoridades de Certificación, comercio electrónico, entre muchas tecnologías más que requieren operaciones seguras sobre nuestras redes.

Claramente, la amplia utilización de los sistemas informáticos ha puesto en evidencia la importancia de la seguridad informática. El objetivo principal de la seguridad informática es proteger los recursos informáticos del daño, la alteración, el robo y la pérdida. Esto incluye los equipos, los medios de almacenamiento, el software, y en general, los datos. Una efectiva estructura de seguridad informática se basa en cuatro técnicas de administración de riesgos, mostradas a continuación:

7.1 Algunos conceptos de seguridad

Para entender mejor la seguridad, es necesario que aclaremos algunos conceptos importantes para tener una visión generalizada de lo que es la seguridad informática:

Amenaza: Circunstancia o evento que puede causar daño violando la confidencialidad, integridad o disponibilidad.

Vulnerabilidades: Por vulnerabilidad entendemos la exposición latente a un riesgo. En el área de informática, existen varios riesgos tales como: ataque de virus, códigos maliciosos, gusanos, caballos de Troya[18] y hackers; no obstante, con la adopción de Internet como instrumento de comunicación y colaboración, los riesgos han evolucionado y, ahora, las empresas deben enfrentar ataques de

[18] Se llaman caballos de Troya pues utilizan el mismo concepto que la historia mítica de la guerra de Troya. Son códigos maliciosos que viene ocultos en archivos que podrían parecer inofensivos

negación de servicio y amenazas combinadas; es decir, la integración de herramientas automáticas de "hackeo", accesos no autorizados a los sistemas y capacidad de identificar y explotar las vulnerabilidades de los sistemas operativos o aplicaciones para dañar los recursos informáticos.

> También podríamos definirla como: "Ausencia de una contramedida, o debilidad de la misma, de un sistema informático que permite que sus propiedades de sistema seguro sean violadas"[φ].

Riesgo: Es e potencial para pérdida o falla de un sistema, como respuesta a las siguientes preguntas:

- ¿Qué podrá pasar (o cuál es la amenaza)?
- ¿Qué tan malo puede ser (impacto o consecuencia)?
- ¿Qué tan frecuente puede ocurrir?
- ¿Qué tanta certidumbre se tiene en las primeras 3 respuestas (grado de confianza)?

Debemos hacer notar que si no hay incertidumbre, no hay un riesgo per se.

Concepto de Seguridad: La existencia de personas ajenas a la información, pueden, incluso, formar parte del personal administrativo o de sistemas, de cualquier compañía; de acuerdo con expertos en el área, más de 70 por ciento de las Violaciones e intrusiones a los recursos informáticos se realiza por el personal interno, debido a que éste conoce los procesos, metodologías y tiene acceso a la información sensible de su empresa, es decir, a todos aquellos datos cuya pérdida puede afectar el buen funcionamiento de la organización. El resultado es la violación de los sistemas, provocando la pérdida o modificación de los datos sensibles de la organización, lo que puede representar un daño con valor de miles o millones de pesos.

[φ] Definición obtenida de "Apuntes de Introducción a la Seguridad".

Garantizar que los recursos informáticos estén disponibles para cumplir sus propósitos, es decir, que no estén dañados o alterados por circunstancias o factores externos, es una definición útil para conocer lo que implica el concepto de seguridad informática.

Dentro de la seguridad también debemos ver que existan mecanismos y políticas que nos permitan garantizar la *confidencialidad*, la *integridad* y la *disponibilidad* de los recursos de un sistema, pues en la actualidad, el activo más importante den un organización es la información.

Confidencialidad: Un sistema pose la propiedad de *confidencialidad* si, la información manipulada por éste no es disponible ni puesta en descubierto para usuarios, entidades o procesos no autorizados. La confidencialidad tiene relación con la protección de información frente a posibles accesos no autorizados, con independencia del lugar en que reside la información o la forma en que se almacena.

Disponibilidad Un sistema posee la propiedad de disponibilidad si, la información es accesible (está disponible) en el momento en que así lo deseen los usuarios, entidades o procesos autorizados La disponibilidad es la garantía de que los usuarios autorizados puedan acceder a la información y recursos cuando los necesiten. La falta de disponibilidad se manifiesta principalmente de dos formas:

La *denegación, o repudio, del servicio* debido a la falta de garantías de la prestación del mismo, tanto por parte del prestador del servicio como del solicitante o tomador (controles de identificación fehaciente, falta de prestaciones de los equipos, congestión de líneas, etc.).

Integridad: Un sistema posee la propiedad de integridad si los datos manipulados por éste no son alterados o destruido por usuarios, entidades o procesos no autorizados. La **integridad** se refiere a la protección de información, datos,

sistemas y otros activos informáticos contra cambios o alteraciones en su estructura o contenido ya sean intencionados, no autorizados o casuales.

Autentificación La autenticación se refiere a demostrar la identidad de las entidades involucradas en la transacción. Evita que alguien tome la identidad de otro. Generalmente toma dos formas:

- Autenticación dl proveedor del proveedor de bienes o servicios
- Autentificación del cliente

Un entorno operativo seguro exige la garantía de que sólo puedan acceder a una determinada información aquellos que están autorizados para ello, que la información se procese correctamente y que está disponible cuando se necesita. La autenticación pretende establecer quién eres. La autorización (o control de acceso) establece qué puedes hacer con el sistema.

No Repudiación Permite comprobar las acciones realizadas por el origen o destino de los datos

- Con pruebas de origen.
- Con pruebas de entrega.

Los mecanismos principales son los certificados, la notarización las firmas digitales. Con el esto se logra de alguna manera garantizar que alguien que hay recibido un pago no pueda negar este hecho, de tal forma que es necesario garantizar que alguien que haya efectuado un pago no pueda negar haberlo hecho.

7.2 Estructura de la seguridad informática

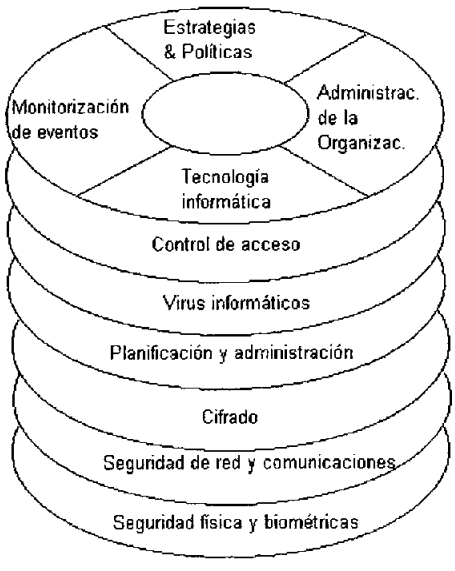


Figura 7.

- **Estrategias y políticas**: Estrategias de administración para seguridad informática y políticas, estándares, guías o directivos usados para comunicar estas estrategias a la organización.
- **Administración de la organización**: Procesos que se dirigen hacia políticas profesionales y programas de capacitación, administración de cambios y control, administración de seguridad y otras actividades necesarias.
- **Monitorización de eventos**: Procesos reactivos que permite a la administración medir correctamente la implementación de políticas e identificar en que momento las políticas necesitan cambios.
- **Tecnología informática**: Es la tecnología necesaria para proveer la apropiada protección y soporte en los distintos procesos involucrados en la organización. La seguridad informática abarca un amplio rango de estrategias y soluciones, tales como:

- *Control de acceso*: Básicamente, el papel del control de acceso es identificar la persona que desea acceder al sistema y a sus datos, y verificar la identidad de dicha persona.

- **Planificación y administración del sistema**: Planificación, organización y administración de los servicios relacionados con la informática , así como políticas y procedimientos para garantizar la seguridad de los recursos de la organización.
- **Cifrado**: La encriptación y la desencriptación de la información manipulada, de forma que sólo las personas autorizadas pueden acceder a ella.
- **Seguridad de la red y de comunicaciones**: Controlar problemas de seguridad a través de las redes y los sistemas de telecomunicaciones.
- **Seguridad física**: Otro aspecto importante de la seguridad informática es la seguridad física de sus servicios, equipos informáticos y medios de datos reales; para evitar problemas que pueden tener como resultado: Pérdida de la productividad, pérdida de ventaja competitiva y sabotajes intencionados. Algunos de los métodos de prevenir el acceso ilegal a los servicios informáticos incluyen:
  - *Claves y contraseñas para permitir el acceso a los equipos.*
  - *Uso de cerrojos y llaves.*
  - *Fichas ó tarjetas inteligentes.*
  - *Dispositivos biométricos (Identificación de huellas dactilares, lectores de huellas de manos, patrones de voz, firma / escritura digital, análisis de pulsaciones y escáner de retina, entre otros).*

7.3 Control de Acceso

Existen dos modelos para la autenticación: DAC y MAC

- Control de acceso discrecional (DAC):
  - Un usuario bien identificado (típicamente, el creador o propietario del recurso) decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema.

- Control acceso mandatario (MAC)
    - Es el sistema quién protege los recursos, todo recurso del sistema y todo usuario tienen una etiqueta de seguridad.

Nombres de los usuario Cada persona que va a tener acceso a un sistema necesita un nombre de usuario para poder autenticarse ante él.

Grupos de usuarios: Para tener una mejor administración sobre los usuarios que acceden a nuestros sistemas, se forman grupos de trabajo, a los cuales asignamos a los usuarios. De ésta forma sabemos, incluso, que tipo de usuario es, a que sistemas puede acceder, y que permisos tiene en nuestro esquema. Un usuario puede pertenecer a uno o más grupos

Super usuarios: La cuenta superusuario, normalmente llamada `root`, viene pre-configurada para facilitar la administración del sistema, y no debería ser utilizada para tareas cotidianas como enviar o recibir correo, exploración general del sistema, o programación.

7.4 Administración Básica de la Seguridad

Una política de seguridad informática es una forma de comunicarse con los usuarios, ya que las mismas establecen un canal formal de actuación del personal, en relación con los recursos y servicios informáticos de la organización.

No se puede considerar que una política de seguridad informática es una descripción técnica de mecanismos, ni una expresión legal que involucre sanciones a conductas de los empleados, es más bien una descripción de los que deseamos proteger y él por qué de ello, pues cada política de seguridad es una invitación a cada uno de sus miembros a reconocer la información como uno de sus principales activos así como, un motor de intercambio y desarrollo en el ámbito de sus negocios.

Como una política de seguridad debe orientar las decisiones que se toman en relación con la seguridad, se requiere la disposición de todos los miembros de la empresa para lograr una visión conjunta de lo que se considera importante.

Las Políticas de Seguridad Informática deben considerar principalmente los siguientes elementos:

- Alcance de las políticas, incluyendo facilidades, sistemas y personal sobre la cual aplica.
- Objetivos de la política y descripción clara de los elementos involucrados en su definición.
- Responsabilidades por cada uno de los servicios y recursos informáticos aplicado a todos los niveles de la organización.
- Requerimientos mínimos para configuración de la seguridad de los sistemas que abarca el alcance de la política.
- Definición de violaciones y sanciones por no cumplir con las políticas.
- Responsabilidades de los usuarios con respecto a la información a la que tiene acceso.
- Otro punto importante, es que las políticas de seguridad deben redactarse en un lenguaje sencillo y entendible, libre de tecnicismos y términos ambiguos que impidan una comprensión clara de las mismas, claro está sin sacrificar su precisión.

7.5 Monitoreo de Sistemas

El monitoreo se realiza constantemente sobre las entradas, los procesos y las salidas de los sistemas: Monitorear los procesos que están ejecutándose de preferencia con *scripts* [19] externos es la mejor manera de ver el estado del sistema. Evitar accesos no autorizados al sistema por medio de detectores de intrusos a nivel e host como TRIPWIRE y deshabilitar los servicios que no se

[19] Un script es un conjunto de comandos agrupados en un archivo, ordenados, para simplificar tareas.

estén usando son algunas acciones que se deben llevar a acabo para la administración del sistema.

Para facilitarnos la tarea, Linux pone a nuestra disposición algunos comandos como :

netsta –an: Muestra las conexiones punto a punto y los servicios abiertos

ps –x: Muestra los procesos que están corriendo en el sistema

top: Nos muestra el uso del cpu, y que procesos están consumiendo mas recursos

md5sum: Permite comprobar la integridad de archivos y saber si fueron o no alterados

Suite pstools: Una poderosa suite de herramientas que permiten monitorear el sistema sin usar los comandos integrados en nuestro servidor, ideal para saber si alguna puerta trasera esta ejecutándose sin nuestro consentimiento

Es incluso de vital importancia revisar las **bitácoras** de acceso al sistema, *syslog*. Lo podemos encontrar en **/var/log/syslog/**. También es altamente recomendable tener el demonio *syslog* escuchando en una computadora diferentes al servidor principal.

7.6 Procesos

En un sistema multitarea todo funciona con procesos. En un sistema multitarea no se debe usar el término programa para hablar de la ejecución del mismo. En su lugar hablaremos de proceso indicando con ello que esta arrancado y funcionando. Un programa puede dar lugar a varios procesos. Un proceso puede estar detenido pero a diferencia de un programa existe una información de estado asociada al proceso. Un programa es algo totalmente muerto. Un proceso detenido es más bien como si estuviera dormido.

7.7 PID y PPID

A cada proceso le corresponderá un número PID que le identifica totalmente. Es decir en un mismo momento es imposible que existan dos procesos con el mismo PID. Lo mismo que todos los procesos tienen un atributo PID que es el número de proceso que lo identifica en el sistema, también existe un atributo llamado PPID. Este número se corresponde con el número PID del proceso padre. Todos los procesos deben de tener un proceso que figure como padre.

7.8 Seguridad en Red

Implicaciones de la seguridad en Internet

Las principales amenazas o riesgos que enfrentan al utilizan las redes son:

1. Intercepción de las Comunicaciones: la comunicación puede ser interceptada y los datos copiados o modificados. La intercepción puede realizarse mediante el acceso físico a las líneas de las redes.

2. Acceso no Autorizado a computadoras y Redes de computadoras: el acceso no autorizado a computadoras o redes de computadoras se realiza habitualmente de forma mal intencionada para copiar, modificar o destruir datos. Técnicamente, se conoce como intrusión y adopta varias modalidades: explotación de información interna, ataques aprovechando la tendencia de la gente a utilizar contraseñas previsibles, aprovechar la tendencia de la gente a desvelar información a personas en apariencia fiables e interceptación de contraseñas.

3. Perturbación de las Redes: actualmente las redes se encuentran ampliamente digitalizadas y controladas por computadoras, pero en el pasado la razón de perturbación de la red más frecuente era una falla en el sistema que controla la red y los ataques a las redes estaban dirigidos principalmente a dichas computadoras. En la actualidad, los ataques más peligrosos se concretan a los

puntos débiles y más vulnerables de los componentes de las redes como son sistemas operativos, ruteadores, conmutadores, servidores de nombres de dominio, etc.

4. Ejecución de Programas que Modifican y Destruyen los Datos: las computadoras funcionan con programas, pero lamentablemente, los programas pueden usarse también para desactivar un computadora y para borrar o modificar los datos. Cuando esto ocurre en una computadora que forma parte de una red, los efectos de estas alteraciones pueden tener un alcance considerable.

5. Declaración Falsa: a la hora de efectuar una conexión a la red o de recibir datos, el usuario formula hipótesis sobre la identidad de la computadora que le contesta en función del contexto de la comunicación. Para la red, el mayor riesgo de ataque procede de la gente que conoce el contexto. Por tal razón, las declaraciones falsas de personas físicas o jurídicas pueden causar daños de diversos tipos. como pueden ser transmitir datos confidenciales a personas no autorizadas, rechazo de un contrato, etc.

6. Accidentes no Provocados: numerosos problemas de seguridad se deben a accidentes imprevistos o no provocados como: son tormentas, inundaciones, incendios, terremotos, interrupción del servicio por obras de construcción, defectos de programas y errores humanos o deficiencias de la gestión del usuario, el proveedor de servicio o el usuario.

7.9 Detección de Intrusos

Los sistemas computarizados y aplicaciones están en permanente evolución, por tal razón pueden surgir nuevos puntos vulnerables. A pesar de los avances en los sistemas de seguridad, los usuarios no autorizados con herramientas muy sofisticadas tienen grandes posibilidades de acceder a las redes, sistemas o sitios de las organizaciones e interrumpir sus operaciones.

Factores que Propician el Acceso de Intrusos a la Redes y Sistemas

- Los sistemas operativos y las aplicaciones nunca estarán protegidos. Incluso si se protege el sistema nuevas vulnerabilidades aparecerán en el entorno todos los días, como las que actualmente representan los teléfonos, equipos inalámbricos y dispositivos de red.
- Falta de seguridad física en algunas empresas y falta de políticas de seguridad informática
- Los empleados no siempre siguen y reconocen la importancia de las políticas de seguridad.
- Requerimientos cada vez mayores de disponibilidad de redes y acceso a ellas.

Medidas para Controlar el Acceso de Intrusos

- Utilizar un *firewall*, que no es más que un dispositivo localizado entre la computadora anfitriona y una red, con el objeto de bloquear el tráfico no deseado de la red mientras permite el cruce de otro tráfico.
- Utilización y actualización de antivirus.
- Actualizar todos los sistemas, servidores y aplicaciones, ya que los intrusos por lo general a través de agujeros conocidos de seguridad.
- Desactivar los servicios innecesarios de redes.
- Eliminar todos los programas innecesarios.
- Analizar la red en busca de servicios comunes de acceso furtivo y utilizar sistemas de detección de intrusos los cuales permiten detectar ataques que pasan inadvertidos a un firewall y avisar antes o justo después de que se produzcan, y
- Establecer la práctica de crear respaldos o backups.

7.10 Firewalls

Es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de red privada y el Internet. El firewall determina cual de los servicios

de red pueden ser accedidos dentro de esta por los que están fuera, es decir quien puede entrar para utilizar los recursos de red pertenecientes a la organización. Para que un firewall sea efectivo, todo trafico de información a través del Internet deberá pasar a través del mismo donde podrá ser inspeccionada la información. El firewall podrá únicamente autorizar el paso del trafico, y el mismo podrá ser inmune a la penetración. desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece entorno a este.

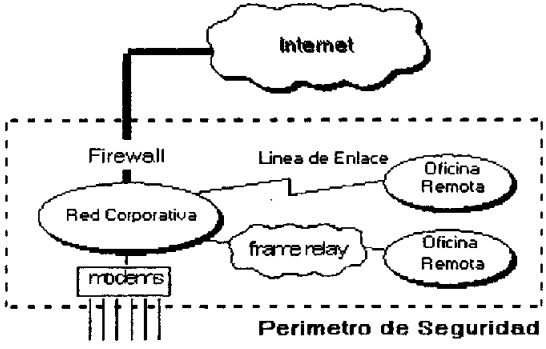


Figura 8. Esquema de un Firewall.

Beneficios

Los firewalls en Internet administran los accesos posibles del Internet a la red privada. Sin un firewall, cada uno de los servidores propios del sistema se exponen al ataque de otros servidores en el Internet. Esto significa que la seguridad en la red privada depende de la "Dureza" con que cada uno de los servidores cuenta y es únicamente seguro tanto como la seguridad en la fragilidad posible del sistema.

El firewall permite al administrador de la red definir un "embudo", manteniendo al margen los usuarios no-autorizados (tal, como., hackers, crakers, vándalos, y espías) fuera de la red, prohibiendo potencialmente la entrada o salida al vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques posibles. El firewall ofrece un punto donde la seguridad puede ser monitoreada y si aparece alguna actividad sospechosa, este generara una alarma ante la

posibilidad de que ocurra un ataque, o suceda algún problema en el transito de los datos.

En los últimos diez años, la importancia de la seguridad informática se ha puesto de manifiesto por algunos incidentes. Uno de ellos fue la del gusano de Internet, en 1988, que se extendió por decenas de miles de computadoras, como resultado de la obra de un hacker[20] llamado Robert Morris.

Claramente, la amplia utilización de los sistemas informáticos ha puesto en evidencia la importancia de la seguridad informática. El objetivo principal de la seguridad informática es proteger los recursos informáticos del daño, la alteración, el robo y la pérdida. Esto incluye los equipos, los medios de almacenamiento, el software, y en general, los datos.

Aplicar planes de contingencia y previsión son actividades que no se deben dejar pasar por alto. Así como realizar auditorias a nuestros sistemas con tal de realizar un análisis sobre nuestras vulnerabilidades y deficiencias , y de ésta forma, ejecuta acciones para el saneamiento de nuestros sistemas y equipos.

Implantar las políticas y mecanismos de seguridad en un sistema es vital en la actualidad, es incluso imprescindible. Asimismo lo es en un esquema de Base de Datos. Y como muestra, el próximo capítulo nos ilustra la interacción de PHP con el motor de Base de Datos Postgres, que es mucho más robusto y más capaz que un muy limitado MySQL.

[20] Un hacker (del inglés hack, recortar), también conocidos como sombreros blancos es el neologismo utilizado para referirse a un experto en varias o algunas ramas relacionadas con la computación y telecomunicaciones: programación, redes de comunicaciones, sistemas operativos, hardware de red/voz.

Capítulo VIII. PHP con PostgreSQL

PHP es una tecnología del lado del servidor, que funciona embebida (es decir, incrustada) dentro del código HTML de una pagina, dándole mayor dinamismo a la misma, con acceso a bases de datos, creación de foros, libros de visita, etc.

Su sintaxis es heredada de **C/Java** y posee gran cantidad de funciones que permiten realizar todas las acciones que soporta el PHP.

Esta tecnología inició como un pasatiempo de Rasmus Lerdorf pero hoy en día, gracias a ser gratuito, cuenta con miles de adeptos y actualizaciones muy constantes. Muchas empresas se han fijado en el potencial de PHP, y han dado su apoyo a ésta, como es el caso de Zend, uno de los principales contribuyentes para el lanzamiento de la versión 4 de este lenguaje.

Gracias a sus librerías para trabajar con Bases de Datos es que podemos trabajar con distintos proveedores, siendo las más comunes MySql y PostgreSQL.

8.1 Programación orientada a objetos

PHP no es un lenguaje explícitamente orientado a objetos, si bien es cierto que está preparado para aprovechar una serie de aspectos de las clases que son interesantes y recomendables en el desarrollo de aplicaciones.

8.2 Sintaxis de clases en PHP

La programación orientada a objetos, aparte de las clases, se basa en más conceptos como la herencia, las interfaces, el polimorfismo, algunos de los cuales todavía no se acaban de implementar en la versión actual de PHP. La sintaxis básica de una clase en PHP es la siguiente.

```
<?php
class nombre_clase {

var $propiedad_1;

var $propiedad_2;

var $propiedad_3;


function método_1($parametro) {

instrucciones_del_método;
}

}
?>
```

Una vez definida la clase, que es el molde del objeto, se pueden crear instancias a partir de ella. En PHP se hace de la siguiente forma.

```
<?php
$nombre_instancia = new nombre_clase($parametros);

?>
```

8.3 Objetos y clases

Cuando hablamos de POO (Programación Orientada a Objetos) los objetos casi siempre son elementos físicos, como puede ser un cliente, proveedor, etc. o elementos conceptuales que existen en el entorno software, por ejemplo un objeto encargado del mantenimiento de archivos. El objetivo es representar a éstos elementos de la vida real y a los conceptuales como unidades de software.

Una de las principales ventajas de la programación OO es el concepto de encapsulamiento, conocido también como protección de datos, mediante el cual solo se pueden modificar los datos de un objeto accediendo a través de sus métodos u operaciones (interfaz del objeto). Nunca se pueden modificar directamente desde la aplicación principal.

En algunas áreas de la programación de aplicaciones Web el uso de la programación OO está desestimada, usándose una metodología estructurada basada en funciones, esto es debido a que determinados proyectos no son lo suficientemente extensos como para aplicarles una metodología OO.

Constructor de una clase

Las clases soportan un tipo de función especial que se conoce como constructor. El constructor es llamado cuando se crea el objeto. Normalmente utiliza para inicializar tareas como: asignación de valores a determinados atributos, crear nuevos objetos necesarios para el correcto funcionamiento de el objeto, etc.

Instanciar una clase

Después de haber declarado una clase, ya podemos usarla creando un objeto que es una instancia a esa clase. Es muy importante que esto quede claro, los objetos son instancias a una clase, por lo tanto cada objeto es único.
En php creamos un objeto usando la palabra reservada **new**. También debemos indicar a que clase va a instanciar el objeto que creemos y pasarle los parámetros (si los requiere) al constructor. Una clase, tiene un puntero especial al que podemos referenciar como $this.

Usar los métodos de una clase

Los métodos de una clase se pueden llamar de la misma forma que se llaman a los atributos. Los métodos de un objeto, son llamados de la misma forma que se llaman a funciones normales, pasándoles como es el caso de método el parámetro que necesiten.

Herencia

Como su nombre indica el concepto de herencia se aplica cuando creamos una clase, que va a heredar los métodos y atributos de una ya definida, entonces la

clase que hemos creado es una subclase. Para que una clase sea subclase de otra ya creada deberemos usar la palabra reservada extends.

Sobrescribir métodos y atributos

Como se mencionó anteriormente, una subclase declara atributos nuevos y operaciones sobre una superclase. Es posible y en muchos casos útil sobrescribir las mismas operaciones o atributos declarados en la superclase. Esto se hace para dar a un atributo un valor diferente en la subclase que el que tiene en la superclase o en el caso de las operaciones para cambiar la funcionalidad de éstas.

Polimorfismo

Cualquier lenguaje de programación orientado a objetos debe soportar el polimorfismo, esto significa que clases diferentes tendrán un comportamiento distinto para la misma operación.

8.4 Trabajando Bases de Datos de PostgreSQL

PostgreSQL es un Sistema Manejador de Bases de Datos Objeto-Relacional (ORDBMS). El paradigma Orientado a Objetos difiere significativamente del Modelo Relacional, con PostgreSQL es posible utilizar ambos conceptos con el fin de construir aplicaciones mucho más poderosas.

8.4.1 Características

PostgreSQL es el gestor de bases de datos de código abierto más avanzado actualmente, ofreciendo control de concurrencia multi-versión, soportando SQL, además de una gran variedad de lenguajes de programación.

8.4.2 Tipos de Datos

PostgreSQL provee una gran variedad de tipos de datos definidos así como la opción de crear nuevos tipos definidos por el usuario.

Numéricos

Tipo Numérico	Tamaño	Rango
smallint	2 bytes	-32768 a +32767
integer	4 bytes	-2147483648 a +2147483647
bigint, int8	8 bytes	-9223372036854775808 a 9223372036854775807
real	4 bytes	6 dígitos decimales de precisión.
double precision	8 bytes	15 dígitos decimales de precision.
numeric	variable	sin límite
decimal	variable	sin límite
serial	4 bytes	1 a 2147483647
bigserial	8 bytes	1 a 9223372036854775807

Caracteres

Tipo Numérico	Rango
character(n), char(n)	Longitud fija rellena con espacios en blanco.
character varying(n), varchar(n)	Longitud Variable con límite.
text	Longitud Variable ilimitada.

Fechas

Tipo	Descripción	Tamaño
timestamp[(p)] con zona horaria	8 bytes	Ambos fecha y hora.
timestamp[(p)][sin zona horaria]	8 bytes	Ambos fecha y hora.
interval[(p)]	12 bytes	Para intervalos de tiempo.

Tipo	Descripción	Tamaño
date	4 bytes	Solo fecha.
time[(p)][sin zona horaria]	8 bytes	Hora del día.
time[(p)] con zona horaria	12 bytes	Hora del día.

Boléanos

Un booleano puede tomar 2 estados "true" o "false". El tercer estado "desconocido" es representado con el estado NULL.

8.5 Operadores y Funciones

PostgreSQL provee un gran número de funciones y operadores. Los usuarios pueden además definir sus propios operadores y funciones. Los comandos psql \df y \do muestran la lista de funciones y operadores disponibles.

Matemáticas

Operadores

+, -, *, /, %, ^, |/, ||/, !, !!, @, &, |, #, ~, <<, >>

Funciones

- abs(x)
- cbrt(dp)
- ceil(numeric)
- degrees(dp)
- exp(dp)
- floor(numeric)
- ln(dp)
- log(dp)
- log(b numeric, x numeric)
- mod(y , x)
- pi()
- pow(n dp, e dp)
- radians(dp)
- random()
- round(dp)
- round(v numeric, s integer)
- sign(numeric)
- sqrt(dp)
- trunc(dp)
- trunc(numeric, s integer)
- acos(x)
- asin(x)
- atan(x)
- atan2(x, y)
- cos(x)
- cot(x)
- sin(x)
- tan(x)

Cadenas

Operadores y Funciones

• string || string • bit_length(string) • char_length(string) o character_length(string) • lower(string) • octet_length(string) • position(substring in string) • substring(string text[from integer][for integer]) • trim([leading | trailing | both][characters] from string) • upper(string) • ascii(text) • btrim(string text, trim text) • chr(integer)	• convert(string text,[src_encoding name,]dest_encoding name) • initcap(text) • length(string) • lpad(string text, length integer [, fill text]) • ltrim(string text, trim text) • pg_client_encoding() • repeat(text, integer) • rpad(string text, length integer [, fill text]) • rtrim(string text, trim text) • strpos(string, substring) • substr(string, from [, count]) • translate(string text, from text, to text)

Otros

Funciones de Grupos de Valores

• AVG(expression) • count(*), count(expression) • max(expression) • min(expression)	• stddev(expression) • sum(expression) • variance(expression)

8.6 Creación de Bases de Datso en PostgreSQL

```
usuario@servidor:~$ createdb prueba
```

De ésta manera es como creamos una BD en Postres, obviamente hay que tener permisos para poder realizar esto, de lo contrario no será creada la Base de Datos.[⊕]

Tablas

A continuación se muestra un ejemplo:

```
=> CREATE TABLE libros(
      id integer UNIQUE,
      titulo text NOT NULL,
      autor_id integer,
      editorial_id integer,
      constraint libros_id_pkey primary key (id));
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
'libro_id_pkey' for table 'libros'
CREATE
```

Esto nos creará la tabla **libros** con los atributos especificados en el script[21]

Insertar

```
INSERT INTO table_name
       [ ( column_name [, ...] ) ]
       VALUES ( value [, ...] )

=> INSERT INTO libros (id, titulo, autor_id, editorial_id)
->        VALUES (41472, 'PostgreSQL',1212,4);
INSERT 16579 1
```

Ahora tenemos un registro en la Base de datos.

[⊕] **Nota**: No es fin de éste capítulo mostrar el uso de permisos y creación de usuarios para poder realizar éstas operaciones, se da por entendido que se es el usuario *postgres* el que efectúa las acciones. El usuario *postgres* es el administrador de todas las bases de datos.
Tampoco pretende ser un manual o curso sobre SQL estándar.

[21] Entiéndase por *script* al conjunto de comandos sql estándar

Acceso a bases de datos PosgreSQL con PHP

Una vez que tenemos una vista general de lo que es Postgres y sus cualidades, podemos entrar en materia de interacción con PHP para realizar sitios dinámicos El sistema de acceso y manipulación de bases de datos desde PHP es similar al de otros lenguajes de script: establece la conexión con la base de datos, ejecuta las sentencias de consulta o modificación y finalmente cierra la conexión.

Conexión con bases de datos PostgreSQL

Para establecer la conexión con una base de datos PostgreSQL desde PHP, se utiliza la función:

```
pg_connect("host=NombreHost dbname=BaseDatos user=Usuario
password=Contraseña")
```

cuyos parámetros, pasados como una cadena única, indican el nombre del servidor -o IP del mismo- 'NombreHost' donde se encuentra la base de datos, el nombre de la base de datos 'BaseDatos', el 'Usuario' de acceso a la base de datos, y la 'Contraseña' de acceso. En caso de éxito la función devuelve un identificador del enlace con el sistema de bases de datos.

Finalmente, para cerrar la conexión utilizamos:

```
pg_close($conexion)
```

donde debemos pasarle como parámetro el enlace con la conexión inicialmente establecida.
Ejemplo:

```
<?php
#Conectamos con PostgreSQL
$conexion = pg_connect("host=192.168.0.3
dbname=BaseDatos user=Usuario password=Contraseña") or
die ("Fallo en el establecimiento de la conexión");

# ################################### #
# Aquí insertaríamos las consultas sobre la base de
datos #
# ################################### #
#Cerramos la conexión con la base de datos
pg_close($conexion);
?>
```

Consultas sobre bases de datos PostgreSQL

Para efectuar consultas sobre una base de datos PostgreSQL, se utiliza en PHP la función:

```
pg_query($conexión $sql)
```

que toma como parámetros, el enlace con la base de datos y una cadena con la consulta SQL a ejecutar (SELECT, INSERT, DELETE, etc.). Devuelve un identificador del resultado en caso de éxito o FALSE en caso de error en la consulta.

Solo con la ejecución de la consulta sobre la base de datos, no podemos presentar el resultado de la misma. Para poder mostrar información resultante de una consulta deberemos hacer uso de funciones complementarias. Una de las posibles es:

```
pg_fetch_array($id_resultado $fila)
```

que devuelve la fila '$fila' en forma de arreglo con el resultado de la sentencia extraída identificada por el parámetro '$id_resultado'.

Ejemplo:

```
<?php
#Conectamos con PostgreSQL

$conexion = pg_connect("host=192.168.0.3 dbname=BaseDatos user=Usuario
password=Contraseña") or die ("Fallo en el establecimiento de la
conexión");

#Efectuamos la consulta SQL

$result = pg_query ($conexion, "select * from personal" )
or die("Error en la consulta SQL");

#Mostramos los resultados obtenidos

$i=0;
while( $row = pg_fetch_array ( $result,$i )) {
   echo $row [ "id" ];

   echo $row [ "nombre" ];

   $i++;

}
?>
```

De esta manera interactuamos con nuestra base de datos para obtener los resultados deseados.

PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python).

Tal vez Postgres es la mejor solución en la sustitución de Oracle para el manejo de Base de Datos. Su robustez mostrada demuestra el porque después de varios años el proyecto sigue en desarrollo. Ofrece todo lo que un manejador de base de datos comercial haría.

Todo lo anteriormente expuesto no serviría de nada si no lo ponemos en práctica, por tal motivo se expone un caso práctico con la aplicación de lo ya estudiado.

Conclusiones

Con los conocimientos aportados por el Diplomado de **"Desarrollo e Implementación de Sistema con Software Libre en LINUX"** he podido tener una visión mucho más amplia del desarrollo de sistema que se encuentran en línea, es decir, en Internet o en cualquier tipo de red. Así como también me lleva a pensar de manera diferente y a tener que considerar muchas más cosas y en mayor grado al tratar de diseñar sistema de esta índole.

Con estas nuevas herramientas mi campo de trabajo se amplia para poder aportar algo más a la sociedad mexicana.

En una empresa el desarrollo de sistemas computacionales se ha vuelto imprescindible, lo que implica asignación de recursos tanto financieros como humanos. Al implementar sistemas basados en software libre se logran reducir los costos de desarrollo del mismo ahorrando gran cantidad de dinero en licencias de software propietario obteniendo con la misma calidad.

Dado que la nueva tendencia es el desarrollo en Web, es por esto que me interesó en demasía este Diplomado; ahora puedo diseñar e implantar sistemas robustos y de gran calidad con herramientas que están al alcance de todos.

Crear sistemas Web nos brinda grandes ventajas sobre sistemas tradicionales, pues los cambios y el mantenimiento que se tengan que realizar sobre éstos no representan grandes costos.

CASO PRÁCTICO

Sistema de Servicios y Atención Ciudadana Versión 2.0

SISTEMA DE SERVICIOS Y ATENCIÓN CIUDADANA

DOCUMENTO DE VISIÓN

OBJETIVO DEL NEGOCIO

(Motivación del Proyecto)

SITUACIÓN ACTUAL.

La visión actual de la Secretaría de Finanzas es ofrecer a la ciudadanía atención a sus dudas, quejas y sugerencias con respecto al pago de las contribuciones de una manera ágil y eficiente mejorando la administración del sistema.

Proporcionar una solución ágil y personalizada al contribuyente para responder vía correo electrónico sus solicitudes de atención de manera oportuna y también llevar un mejor control de todas las solicitudes.

Con esto se busca mejorar la administración y control de todas las solicitudes de atención ofreciendo varias maneras de consultar la información y atender las solicitudes por parte de los administradores del sistema.

Actualmente el sistema proporciona solamente algunos reportes gráficos pero no reportes tabulados, además el Personal de Atención debe responder las solicitudes de atención vía correo electrónico por medio del cliente de correo de Finanzas. Con las versión 2.0 del SSAC se busca tener una mejor administración y un mayor control sobre las solicitudes de atención de los contribuyentes y sobre el sobre las respuestas proporcionadas por el Personal de Atención.

Problema.

Con respecto a las solicitudes de atención:

- El contribuyente no recibe un correo electrónico con el cual pueda corroborar que su solicitud de atención ya fue identificada por el sistema.
- El desarrollo de un nuevo sistema de administración y control de todas las solicitudes de atención y del personal.
- El sistema no cuenta con un módulo que permita responder los correos electrónicos por parte del Personal. Esto se hace a través del cliente de correo.

Con respecto a la administración del sistema:

- Falta de reportes tabulados que permitan obtener información sobre las solicitudes de atención para llevar un mejor control de las mismas.
- Inconsistencia en los reportes gráficos respecto a los folios atendidos.
- Falta de gráficas realmente informativas
- Falta de interfases gráficas cuando se requiere actualizar algún dato, tales como: errores de dedo en el correo electrónico del contribuyente, si no llega el correo electrónico al personal de atención en el momento de turnar folio, entre otras.

IMPLICACIÓN.

- Realizar el análisis y diseño de la nueva base de datos.
- Migrar los datos de la base de datos actual a la base de datos nueva
- Realizar el análisis del sistema en un documento formal
- Realzar las nuevas interfases gráficas del diseño de sistema
- Llevar a cabo el desarrollo del sistema
- Proporcionar capacitación para los administradores del sistema y para el personal de atención.
- Contar con un catálogo que permita registrar a los usuarios del sistema
- Cifrar las contraseñas de los usuarios del sistema

NECESIDAD.

Derivado del problema actual en la administración y control sobre las solicitudes de atención, es necesario llevar un control adecuado y consistente sobre las solicitudes de información y sobre el personal que atiende dichas solicitudes.

En este sentido, es necesario contar con una nueva versión del Sistema de Información "Servicios de Atención Ciudadana" que permita la atención y seguimiento de las solicitudes de atención de los contribuyentes además de controlar al Personal de Atención de una manera sencilla y eficiente.

SITUACIÓN ÓPTIMA

Contar con un sistema de Administración y Control eficiente que permita a los administradores registrar, consultar, turnar y dar seguimiento de las solicitudes de atención de los contribuyentes así como también llevar un control sobre el Personal de Atención que responde los correos electrónicos de los contribuyentes.

Declaración de la Visión

OBJETIVO DEL SISTEMA

¿Qué se va a hacer? Diseñar y Desarrollar la nueva versión del Sistema de Servicios y Atención Ciudadana denominado "SSAC" que permita al Contribuyente enviar sus dudas, comentarios, sugerencias, quejas, entre otras; así como también llevar una mayor administración y control en las solicitudes de atención y del Personal que las atiende

¿Mediante? El desarrollo de un sistema que automatice y mejore los procesos de recepción de solicitudes, turnar folios, responder folios, y dar seguimiento y control de los mismos; así como la actualización constante de los catálogo y operaciones realizadas por el administrador del sistema.

¿Para? Ofrecer a los usuarios del sistema actual (administradores y personal de atención), nuevos mecanismos de atención de solicitudes más eficientes, más sencillos y con una apariencia más presentable, además de obtención de información más útil.

¿Quiénes? Dirección General de Registro Civil, Oficina del Secretario, Procuraduría Fiscal, la Subsecretaría de Egresos, Subtesorería de Política Fiscal, La Subtesorería de Administración Tributaria, la Subtesorería de Catastro y Padrón Territorial, Contraloría

Interna, Sistema de Aguas de la Ciudad de México y la Dirección General de Informática.

ALCANCE

Se contempla que el nuevo SACC realice lo siguiente:

- Proporcionar ayuda en línea: El contribuyente podrá consultar diferentes preguntas que se realizan frecuentemente con el fin de publicar las dudas más relevantes sin necesidad de enviar una solicitud de atención.
- Envío de solicitudes Al igual que en el sistema actual, el contribuyente podrá seguir enviando sus solicitudes de atención con el fin de solicitar información que no se encuentre en la ayuda en línea.
- Catálogos del sistema: El sistema contará con un módulo en el cual se pueden tener actualizados los catálogos con los cuales funciona el sistema, tales como: personal, temas, preguntas, áreas y dependencias, entre otros.
- Turno de solicitudes de atención. Turno de las distintas solicitudes de atención hacia el personal. Estas solicitudes estarán controladas por un número de folio.
- Respuesta de solicitudes de atención: El personal podrá atender las solicitudes a través de un módulo para responder los correos electrónicos.
- Conclusión de Solicitudes Conclusión de las solicitudes de atención después de ser atendidas o canceladas
- Recordatorio de solicitudes: El SSAC generará recordatorios automáticos cuando se venza la fecha de responder una solicitud (7 días), y tendrá la opción de enviar un comentario adicional
- Cancelar solicitudes: Cancelar las solicitudes por diversos motivos tales como: comentarios del contribuyente vacíos, duplicidad de folios, contenido agresivo, entre otros.
- Dar seguimiento a las solicitudes: Seguimiento de los comentarios de los contribuyentes después de que sus solicitudes fueron atendidas.
- Consulta general de solicitudes atendidas: Consulta de solicitudes por rango de fechas o período, por número de folio, por nombre de contribuyente, por nombre del personal de atención.
- Estatus de las solicitudes. Consultar los diferentes estatus de las solicitudes tales como: cuantas se encuentran recibidas o pendientes de atender, cuantas se encuentran turnadas,

EXCLUSIONES:

- No existe liga al cliente de correo de la Secretaría de Finanzas.
- No se registrará el seguimiento de las solicitudes de atención automáticamente.

Análisis del Sistema de Servicios de Atención Ciudadana (SSAC)

El Sistema de Servicios y Atención Ciudadana es un Sistema en Web que permite el registro de solicitudes de Atención por parte de los contribuyentes de la Secretaría de Finanzas y responder sus solicitudes por parte del personal a través del correo electrónico.

En este Sistema se encuentran identificados los siguientes Actores:

1. Administrador del Sistema
2. Contribuyente
3. Personal
4. Usuario

Y los siguientes Casos de Uso:

Para el contribuyente:

1. Consultar ayuda en línea
2. Enviar solicitud

Para el administrador

3. Administrar catálogos
4. Turnar folio
5. Responder folio
6. Concluir folio
7. Enviar recordatorio
8. Dar seguimiento a folio
9. Cancelar folio

Para el personal

10. Consultar folio

Para el usuario:

11. Registrar Autenticación
12. Cambio de contraseña
13. Consultar reportes

DOCUMENTO DE CASOS DE USO

INTRODUCCIÓN

OBJETIVO DEL DOCUMENTO

El presente documento describe los Casos de Uso identificados por el equipo de diseño del Sistema de Servicios y Atención Ciudadana (en adelante referido como SSAC); sistema informático con el que se automatizará el registro y seguimiento de las solicitudes de atención de los contribuyentes enviados desde la página web de la Secretaría de Finanzas.

CASOS DE USO

Un problema frecuente durante las etapas de análisis y diseño de un sistema es la deficiente comunicación entre los usuarios finales y el equipo de desarrollo. Es por esto conveniente implementar mecanismos que sirvan de interfase entre el lenguaje propio del dominio del negocio y un lenguaje que el equipo de desarrollo pueda entender, con el objetivo de lograr una especificación que cubra los requerimientos de los usuarios finales. Una de estas herramientas son los Casos de Uso, que forman parte del Lenguaje de Modelado Unificado (UML, por sus siglas en inglés).

Podemos ver a los Casos de Uso como la colección de **situaciones** respecto al uso del sistema que se va a desarrollar. Con ellos es posible describir una secuencia de eventos, misma que es iniciada por una persona, otro sistema, una parte del hardware o por el paso del tiempo. A las entidades que inician secuencias se les conoce como **actores**. El resultado de la secuencia debe ser algo utilizable ya sea por el actor que la inició o cualquier otro. En resumen, los Casos de Uso son una descripción del sistema desde el punto de vista del usuario final.

Es importante señalar que los Casos de Uso no especifican la forma en que se han de resolver los problemas, sino cuáles problemas serán resueltos por el sistema; es decir cubre el **qué** y no el **cómo**.

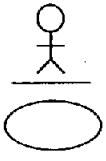
DIAGRAMAS DE CASOS DE USO

Una de las características más poderosas de los diagramas de Casos de Uso es su simplicidad, ya que están compuestos básicamente de tres símbolos:

Actores:

Relaciones:

Casos de Uso:



RELACIÓN DE REQUERIMIENTOS DEL ÁREA USUARIA

(Según Documento de Visión)

1. Proporcionar ayuda en línea: El contribuyente podrá consultar diferentes preguntas que se realizan frecuentemente con el fin de publicar las dudas más relevantes sin necesidad de enviar una solicitud de atención.
2. Envío de solicitudes Al igual que en el sistema actual, el contribuyente podrá seguir enviando sus solicitudes de atención con el fin de solicitar información que no se encuentre en la ayuda en línea.
3. Catálogos del sistema: El sistema contará con un módulo en el cual se pueden tener actualizados los catálogos con los cuales funciona el sistema, tales como: personal, temas, preguntas, áreas y dependencias, entre otros.
4. Turno de solicitudes de atención. Turno de las distintas solicitudes de atención hacia el personal. Estas solicitudes estarán controladas por un número de folio.
5. Respuesta de solicitudes de atención: El personal podrá atender las solicitudes a través de un módulo para responder los correos electrónicos.
6. Conclusión de Solicitudes Conclusión de las solicitudes de atención después de ser atendidas o canceladas
7. Recordatorio de solicitudes: El SSAC generará recordatorios automáticos cuando se venza la fecha de responder una solicitud (7 días), y tendrá la opción de enviar un comentario adicional
8. Cancelar solicitudes: Cancelar las solicitudes por diversos motivos tales como: comentarios del contribuyente vacíos, duplicidad de folios, contenido agresivo, entre otros.
9. Dar seguimiento a las solicitudes: Seguimiento de los comentarios de los contribuyentes después de que sus solicitudes fueron atendidas.
10. Consulta general de solicitudes atendidas: Consulta de solicitudes por rango de fechas o período, por número de folio, por nombre de contribuyente, por nombre del personal de atención.
11. Estatus de las solicitudes. Consultar los diferentes estatus de las solicitudes tales como: cuantas se encuentran recibidas o pendientes de atender, cuantas se encuentran turnadas,

DESCRIPCIÓN DE LOS ACTORES.

Contribuyente: Es la persona que envía sus solicitudes de servicio vía Web con el fin de recibir una respuesta.

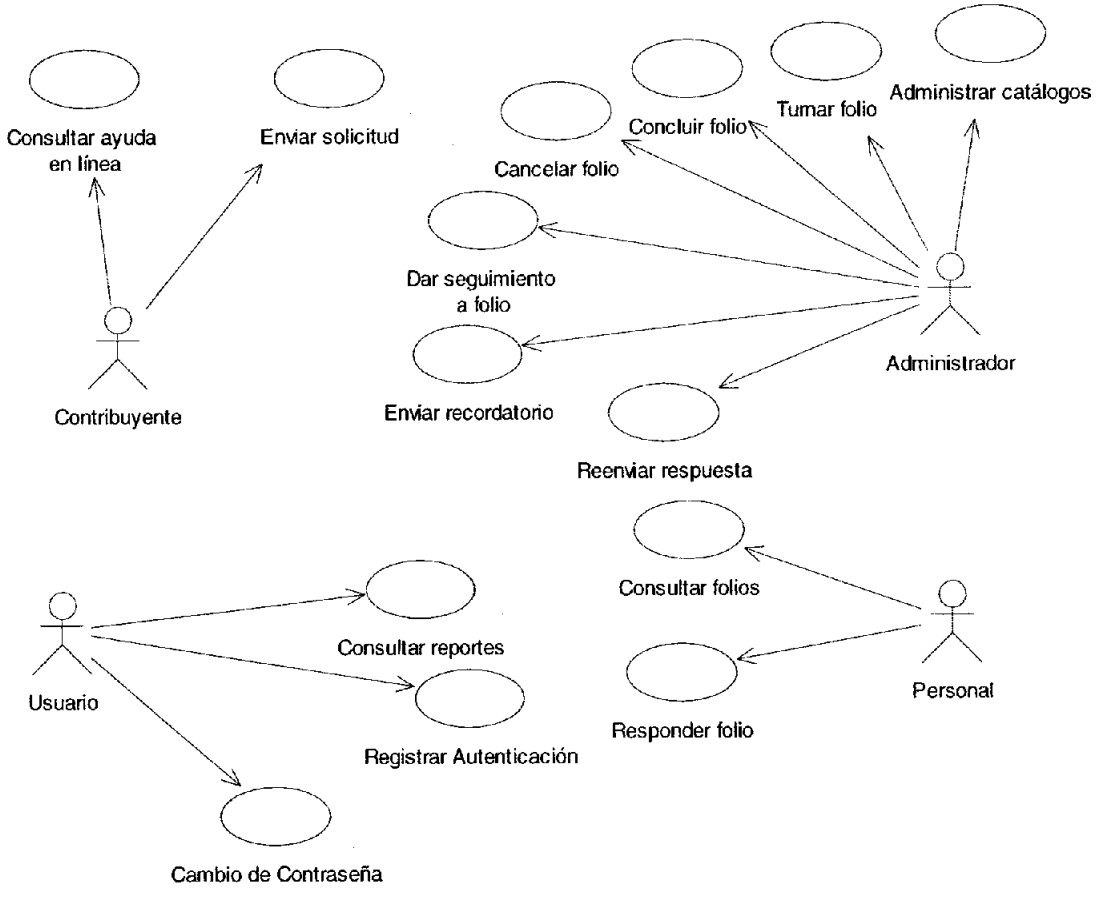
Personal: Es el funcionario al cual se le turnan los distintos folios con el fin de que pueda atenderlos vía correo electrónico.

Administrador: Es el usuario que se encarga de todo la gestión del sistema dependiendo de sus privilegios. Entre las actividades que realiza son: administrar folios, actualizar los catálogos del sistema y consultar reportes.

Existen dos tipos de administradores: parcial y total, dependiendo del grupo al que pertenecen.

Usuario: Es la persona que hace uso del sistema, dependiendo de los privilegios que tenga realizará las actividades del Personal o del Administrador.

DIAGRAMA DE CASOS DE USO



CASOS DE USO RELATIVOS A LA OPERACIÓN

CASO DE USO	1. CONSULTAR AYUDA EN LÍNEA
DESCRIPCIÓN	El Contribuyente entra al Sistema de Servicios de Atención Ciudadana y busca a través de la ayuda en línea la(s) pregunta(s) más frecuentes que han hecho otros contribuyentes con el fin de encontrar la misma pregunta que tiene en ese momento.
ACTOR QUE INICIA	Contribuyente
PRECONDICIONES	Entrar al Sistema de Servicios de Atención Ciudadana por medio del URL.
POSCONDICIONES	No existen postcondiciones.
ACTOR QUE SE BENEFICIA	Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El Contribuyente entra al Sistema de Servicios de Atención Ciudadana y busca el tema de su interés **(E1)** con el fin de visualizar las diferentes preguntas relacionadas con ese tema y que han realizado otros contribuyentes. Al encontrar la respuesta a su pregunta entonces puede decidir si necesita una atención más específica al respecto y enviar una solicitud de atención o si es suficiente con la respuesta previamente publicada.

Flujos de Excepción:

(E1): Si el contribuyente no ha elegido el tema de su interés, el sistema enviará un mensaje de error indicando que debe elegir un tema.

CASO DE USO	2. ENVIAR SOLICITUD
DESCRIPCIÓN	El Contribuyente entra al Sistema de Servicios de Atención Ciudadana y escribe sus datos personales en la pantalla, así como también la solicitud de atención que esté demandando en ese momento.
ACTOR QUE INICIA	Contribuyente
PRECONDICIONES	Entrar al Sistema de Servicios de Atención Ciudadana por medio del URL.
POSCONDICIONES	Emisión del folio referente a la solicitud de atención, el cual servirá

	para identificar de manera única cada una de las solicitudes durante todo el tiempo en el cual sea atendida.
ACTOR QUE SE BENEFICIA	Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El Contribuyente entra al Sistema de Servicios de Atención Ciudadana. El Sistema muestra una pantalla donde el Contribuyente escribe sus datos personales en la pantalla, tales como: Nombre, Apellido Paterno, Apellido Materno **(E1)**, Nombre y Tipo de Dependencia (si las tuviera), País y Estado al que pertenece, Domicilio Particular, teléfono, correo electrónico **(E2)** y su solicitud de atención **(E3)**.

Una vez enviada la solicitud de atención **(E4)**, esta se registra y emite un número de folio único para el contribuyente, el cual deberá conservar para cualquier duda o aclaración posterior **(A1)**.

Flujos Alternos:

(A1): Internamente el Sistema envía un correo electrónico al Contribuyente con copia para la lista de administración con la solicitud de atención y el número de folio emitido.

En caso de que el Contribuyente conteste el correo, se registran sus comentarios de manera histórica y el folio pasa de estado de Recepción a estado de Previo a Turno.

Flujos de Excepción:

(E1): Si el nombre completo del contribuyente está vacío o contiene números y/o caracteres especiales, el sistema envía un mensaje de error indicando que los datos son incorrectos o se encuentran vacíos.

(E2): Si el correo electrónico del contribuyente se encuentra vacío, contiene caracteres especiales como comas, o no contiene los caracteres especiales arroba (@) y punto (.), el sistema envía un mensaje de error indicando que debe ser escrito correctamente.

(E3): Si el contribuyente no captura su solicitud de atención el sistema envía un mensaje de error indicando que el comentario se encuentra vacío.

(E4); Si por alguna razón el sistema registra de manera duplicada alguna solicitud de atención, quedarán registradas tanto la solicitud original y las duplicadas, diferenciándose únicamente por el número de folio emitido.

CASO DE USO	3. ADMINISTRAR CATÁLOGOS
DESCRIPCIÓN	El administrador del Sistema elige un catálogo con el fin de realizar algunas operaciones de modificación tales como: altas, bajas, cambios y consultas.
ACTOR QUE INICIA	Administrador
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No existen postcondiciones
ACTOR QUE SE BENEFICIA	Administrador, Personal y Contribuyente.
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El administrador entra al Sistema y elige un catálogo **(E1)**. Una vez elegido, el Sistema muestra que operación es la que se va a realizar y el administrador elige la opción.

Si la opción es alta, entonces el Sistema muestra una pantalla de captura de datos. El administrador captura los datos requeridos **(E2)**, y los envía para que sean registrados.

Si la opción es baja, entonces el Sistema muestra los datos que se desean dar de baja y permite que el administrador elija si lo da de baja o elige otro registro.

Si el administrador decide dar de baja el registro, entonces el Sistema permite confirmar la decisión y si es confirmada elimina de la base de datos ese registro y muestra los datos que fueron eliminados.

Cabe señalar que para el catálogo de personal el registro no se elimina, únicamente se cambia de estatus, de tal manera que el personal pasa de estatus activo a estatus inactivo.

Para el catálogo de tema y de pregunta se tiene un campo de estatus con el fin de saber si tema o pregunta se publican o se ocultan.

Si la opción es cambio, el Sistema muestra los datos que se desean actualizar y permite que el administrador elija si lo actualiza o elige otro registro.

Si el administrador decide actualizar el registro **(E2)**, entonces el Sistema actualiza en la base de datos ese registro.

Si la opción es consulta, el Sistema permite elegir el criterio por el cual se van a consultar los registros, ya sea por fecha, nombre, cargo del personal, entro otros.

Flujos de Excepción:

(E1): Si el administrador no elige un catálogo, el Sistema envía un mensaje de error indicando que debe elegir un catálogo.

(E2): Si el administrador no captura alguno de los datos requeridos el Sistema envía un mensaje de error indicando que debe capturar todos los datos requeridos.

CASO DE USO	4. TURNAR FOLIO
DESCRIPCIÓN	El Administrador del Sistema de Atención Ciudadana entra al módulo administrativo y asigna responsable de atención al folio.
ACTOR QUE INICIA	Adminstrador
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No exisiten postcondiciones
ACTOR QUE SE BENEFICIA	Admnistrador, Personal y Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El Sistema de Atención Ciudadana (SSAC), recibe las solicitudes de información, quejas, sugerencias, demandas de servicios mediante un folio que en primer instancia fue asignado al Contribuyente.

El Administrador revisa los correos electrónicos del Contribuyente. Si alguno de los correos es incorrecto, el Sistema permite modificarlos **(E1)**.

Una vez que los datos del Contribuyente, el Administrador consulta los comentarios que hizo llegar el Contribuyente y posteriormente cataloga el tipo de solicitud (queja/denuncia, solicitud de información/comentario), y el tema que corresponde a la solicitud (predial, agua, % sobre nómina).

Una vez catalogada **(E2)** el administrador asigna responsables de atención.

El número de responsables de atención puede ser múltiple y va depender de cuantos temas diferentes contenga el comentario para la asignación de las áreas responsables de atención**(E3)**.

Se turna al responsable asignado. **(A1)** y el Sistema envía un mensaje indicando que el folio ha sido turnado **(E4)**.

En caso de que el Contribuyente conteste el correo, se registran sus comentarios de manera histórica y el folio pasa de estado de Turnado a estado de Previo a Respuesta.

Flujos Alternos:

(A1): Se envía un correo electrónico a la(s) persona(s) responsable(s) de atención, al Contribuyente para avisarle quien le brindará la atención a su comentario y, se envía otro correo a lista de Administradores del Sistema para ser avisados de la asignación del responsable.

Flujos de Excepción:

(E1): Si el correo electrónico del contribuyente se encuentra vacío, contiene caracteres especiales como comas, o no contiene los caracteres especiales arroba (@) y punto (.), el sistema envía un mensaje de error indicando que debe ser escrito correctamente.

(E2): Si no se cataloga el folio de atención el Sistema mandará un mensaje de error que indique "Debe catalogar el folio."

(E3): Si no se asigna responsable de atención el Sistema mandará un mensaje de error que indique "No se han asignado Responsables".

(E4): En caso de que el folio no se haya turnado al personal de atención correspondiente, será necesario que el administrador lo turne como si fuera una nueva solicitud de atención y se repite el flujo principal.

CASO DE USO	5. RESPONDER FOLIO
DESCRIPCIÓN	El Personal responsable de atención de folios responde por medio del Sistema de Servicios de Atención Ciudadana el folio que le fue turnado
ACTOR QUE INICIA	Adminstrador
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No exisiten postcondiciones
ACTOR QUE SE BENEFICIA	Admnistrador, Personal y Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El personal de atención entra al Sistema y revisa los folios que tiene pendientes y elige la opción de responder folio. En seguida el sistema muestra una pantalla en la que aparece el nombre y correo electrónico del Contribuyente, indicando que se le enviará un correo electrónico con copia para la lista de correo de los administradores del Sistema y que se envía desde el correo electrónico del personal de atención.

El personal de atención escribe la respuesta al Contribuyente **(E1)** en el área de texto destinada para ello y envía la respuesta al Contribuyente **(A1)**. El Sistema de manera automática genera la firma del responsable de Atención.

Flujos Alternos:

(A1): De manera alterna el Sistema envía por correo electrónico la respuesta al Contribuyente y a la lista de administración.

Flujos de Excepción:

(E1): Si el Personal de Atención omite la respuesta al Contribuyente el Sistema enviará un mensaje de error indicando que debe escribir la respuesta antes de enviar el correo.

CASO DE USO	6. CONCLUIR FOLIO
DESCRIPCIÓN	El administrador del Sistema de Atención Ciudadana (SACC), finaliza la atención de un folio.
ACTOR QUE INICIA	Adminstrador
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No exisiten postcondiciones
ACTOR QUE SE BENEFICIA	Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El administrador del Sistema solicita los folios respondidos por el Personal de Atención. El Sistema muestra los folios respondidos y el Administrador elige el que desea concluir. el Sistema muestra un conjunto de respuestas predefinidas y da la opción de escribir una diferente a las que se tienen:

El administrador decide concluir el folio **(A1),** posteriormente el Sistema envía un mensaje indicando que el folio ha sido concluido. Cabe señalar que el sistema coloca el folio en un estatus de "Conclusión Folio" ya que puede ser susceptible de seguimiento.

En caso de que se concluya por seguimiento, el folio pasa de estatus "Conclusión Folio" a "Conclusión Seguimiento".

El administrador revisa que se tenga en el correo (lista de administradores), la respuesta al "Aviso de Atención" por parte del personal de atención del Sistema de Atención Ciudadana.

Flujos Alternos:

(A1): Se manda un correo electrónico al Contribuyente avisándole que su folio a sido concluido, con copia a la lista de administradores para estar enterados de la conclusión del folio.

CASO DE USO	7. ENVIAR RECORDATORIO
DESCRIPCIÓN	El Sistema de Atención Ciudadana (SSAC) envía automáticamente los recordatorios al personal que en el plazo establecido no contestó los folios correspondientes para su atención. Los recordatorios de los folios se pueden enviar de manera manual además de que se pueden anexar comentarios al responsable de atención en caso de que se requiera.
ACTOR QUE INICIA	Administrador
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado. Que el responsable de atención no atienda los folios en el plazo establecido que requería de su atención.
POSCONDICIONES	No existen Postcondiciones.
ACTOR QUE SE BENEFICIA	Admnistrador y Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El sistema de manera automática envía los recordatorios tras cumplir con el plazo para contestar cada folio **(A1).**

Cuando se requiera mandar de manera manual el recordatorio, el administrador del Sistema selecciona de la lista de folios vencidos y se manda el recordatorio **(A1).**

Cuando el recordatorio se genere de manera manual, el Administrador tiene la opción de enviar una nota de atención al responsable. De esta manera el Sistema enviará un mensaje indicando que el folio ha sido enviado a la persona responsable para la atención como un recordatorio.

Flujos Alternos:

(A1): De manera alterna el Sistema envía por correo electrónico la respuesta al Contribuyente y a la lista de administración.

Flujos de Excepción:

(E1): Si el Personal de Atención omite la respuesta al Contribuyente el Sistema enviará un mensaje de error indicando que debe escribir la respuesta antes de enviar el correo.

CASO DE USO	8. DAR SEGUIMIENTO A FOLIO
DESCRIPCIÓN	Los administradores del Sistema de Atención Ciudadana (SACC) registran los comentarios del Contribuyente y respuestas brindadas por el personal de atención.
ACTOR QUE INICIA	Contribuyente
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado. Que el contribuyente mande agradecimientos, más comentarios, dudas, a través del correo electrónico de la lista de administradores o la del responsable de atención.
POSCONDICIONES	No existen Postcondiciones.
ACTOR QUE SE BENEFICIA	Admnistrador y Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El administrador del Sistema copia los comentarios de agradecimientos, dudas o comentarios, así como las respuestas del personal de atención (si las hay), en el módulo de administración en la sección de registrar seguimiento de un folio.

El sistema envía un mensaje indicando que se ha registrado un nuevo comentario en el folio..

El sistema cambia el estatus del folio de "Seguimiento" a "Conclusión Seguimiento"

CASO DE USO	9. CANCELAR FOLIO
DESCRIPCIÓN	Los administradores del Sistema de Atención Ciudadana (SACC) tienen la opción de cancelar un folio por algún motivo que no permita continuar con el proceso normal de la atención de la solicitud.

ACTOR QUE INICIA	Contribuyente
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado. Tener una solicitud de atención con un folio.
POSCONDICIONES	No existen Postcondiciones.
ACTOR QUE SE BENEFICIA	Admnistrador y Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El administrador del sistema consulta todos los folios nuevos que se registran por parte de los contribuyentes y entonces decide cancelar alguno de ellos por los siguientes motivos:

- Porque la solicitud de atención tiene exactamente el mismo contenido en los comentarios, nombre del contribuyente, y fecha.
- Porque los comentarios del folio se encuentran vacíos
- Porque los comentarios del folio son de contenido ofensivo o ajeno a los fines del sistema.
- Otros motivos

El administrador elige el folio a cancelar y el sistema muestra los datos completos del registro. El administrador decide cancelar el folio y el sistema solicita confirmar la cancelación. El administrador decide cancelarlo **(A1)**, y escribe los motivos por los cuales fue canceladl el folio. El Sistema envía un mensaje indicando que el folio ha sido cancelado.

Flujos Alternos:

(A1): El Sistema registra el folio de manera histórica primeramente como cancelado y posteriormente como concluido.

CASO DE USO	10. CONSULTAR FOLIO
DESCRIPCIÓN	El Personal asignado para responder los folios del Sistema de Servicios y Atención Ciudadana consulta de manera histórica sus folios y/o los folios del Personal de la misma área.
ACTOR QUE INICIA	Personal
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No existen Postcondiciones.
ACTOR QUE SE BENEFICIA	Personal y Contribuyente

REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El Personal entra al Sistema de Servicios de Atención Ciudadana y el sistema muestra una pantalla en la cual el personal puede consultar los folios que ha respondido así como también los folios que tiene pendientes de responder.

El personal puede elegir si desea consultar los folios:

- Por el estatus.
- Escribiendo el número de folio
- Folios respondidos por los responsables de atención de su misma área ordenados por meses.

El personal elige alguna de las opciones anteriores y el sistema muestra los folios bajo el criterio elegido.

CASO DE USO	11. CAMBIAR CONTRASEÑA
DESCRIPCIÓN	En este caso de uso el Personal y el administrador del Sistema asumen el rol de usuario. El usuario tiene la opción de modificar su contraseña en cualquier momento, escribiendo su contraseña actual y su contraseña nueva.
ACTOR QUE INICIA	Usuario
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	El Sistema registra la nueva contraseña la cual será con la que se identificará a usuario posteriormente.
ACTOR QUE SE BENEFICIA	Usuario
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

Cuando el usuario entra por primera vez al Sistema de Servicios de Atención Ciudadana se muestra una pantalla en la cual puede cambiar su contraseña. El usuario puede decidir cambiar su contraseña en ese momento o posteriormente. Si decide cambiarla en ese momento, entonces el sistema permite que el usuario escriba su contraseña actual **(E1)**, su contraseña nueva y confirmar su nueva contraseña **(E2).**

De esta manera cuando el usuario entra al Sistema posteriormente lo deberá hacer con su nueva contraseña.

Si el usuario decide no cambiar su contraseña, entonces cuando entre en ocasiones posteriores el sistema ya no desplegará la pantalla de cambio de contraseña, en su lugar desplegará una opción para ir a esa pantalla.

Cabe señalar que la contraseña se debe cifrar con el fin de que queden registradas de manera protegida.

Flujos de Excepción:

(E1): Si el usuario escribe incorrectamente su contraseña actual o deja el campo vacío, el sistema envía un mensaje de error indicando que debe escribir correctamente su contraseña.

CASO DE USO	12. REGISTRAR AUTENTICACIÓN
DESCRIPCIÓN	En este caso de uso el Personal y el administrador del Sistema asumen el rol de usuario. El usuario se loguea en el sistema y dependiendo del tipo de usuario y de los privilegios que tenga entra a distintos módulos.
ACTOR QUE INICIA	Usuario
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	El usuario genera una sesión en el SSAC.
ACTOR QUE SE BENEFICIA	Usuario
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El Usuario solicita entrar al Sistema. El Sistema muestra una pantalla en la cual solicita el Nombre de Usuario y la Contraseña.

El Usuario escribe su nombre de Usuario y Contraseña **(E1)**, y entonces, dependiendo de los privilegios con los que cuente, el Sistema le muestra un menú con todas las opciones que puede realizar, mostrando un mensaje de bienvenida.

Cabe señalar que si el usuario se loguea por primera ocasión, el sistema mostrará una pantalla con la opción de cambiar su contraseña. Para las siguientes ocasiones el Sistema mostrará solamente el menú.

Flujos de Excepción:

(E1): Si el usuario escribe incorrectamente su nombre de usuario y contraseña o deja los campos vacíos, el sistema envía un mensaje de error indicando que debe escribir correctamente su nombre de usuario y contraseña.

CASO DE USO	13. CONSULTAR REPORTES
DESCRIPCIÓN	En este caso de uso el Personal y el administrador del Sistema asumen el rol de usuario. El usuario se loguea en el sistema y dependiendo del tipo de usuario y de los privilegios que tenga entra a distintos módulos.
ACTOR QUE INICIA	Usuario
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No existen postcondiciones..
ACTOR QUE SE BENEFICIA	Usuario
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El usuario entra al Sistema y elige ver los reportes. Puede elegir ver reportes tabulados o reportes gráficos.

Si decide ver reportes tabulados entonces el Sistema muestra una pantalla con diferentes criterios de consulta tales como: escribir el número de folio, por Correo Electrónico del Contribuyente o por Nombre del Contribuyente.

También el administrador puede elegir un rango de fechas. Por defecto el sistema elige el rango de fechas desde el primer día del mes hasta el día actual.

El usuario puede escribir un rango de fechas distinto al de defecto **(E1)**.

Dependiendo de los privilegios con los que cuente en el Sistema, el usuario puede consultar los diferentes tipos de reportes tabulados, tales como:

- Cuantos tipos de solicitud diferentes existen por área y/o dependencia.
- Cual es la cantidad de cada uno de los estados de los estatus de los folios.
- Cuantos folios por tipo de dependencia diferentes existen.
- Cantidad de solicitudes registradas por mes.
- Cantidad de solicitudes atendidas por área.
- Cantidad de solicitudes atendidas por dependencia.
- Cantidad de solicitudes atendidas por Personal de Atención..
- Tiempo de respuesta por dependencia.

Los reportes gráficos serán los mismos que los tabulados

Flujos de Excepción:

(E1): Si el usuario elige un rango de fechas no válido, por ejemplo, si la primera es posterior a la segunda fecha, el sistema enviará un mensaje de error indicando que el rango de fechas no es válido.

CASO DE USO	14. REENVIAR RESPUESTA
DESCRIPCIÓN	El administrador reenvía el correo electrónico de la respuesta del personal de atención en caso de que no le haya llegado al contribuyente.
ACTOR QUE INICIA	Contribuyente
PRECONDICIONES	Tener un nombre de usuario y contraseña previamente asignado.
POSCONDICIONES	No existen postcondiciones..
ACTOR QUE SE BENEFICIA	Contribuyente
REQUERIMIENTOS QUE CUBRE EL CU	Ninguno.

Flujo Principal:

El Contribuyente envía un correo electrónico a la lista de administración del SSAC mencionando que no recibió la respuesta a sus comentarios o solicitudes de información por parte del Personal, por lo cual el administrador entra al sistema y realizar un seguimiento del folio, es decir, se utiliza el caso de uso Dar Seguimiento a Folio" (Ver pág. 19).

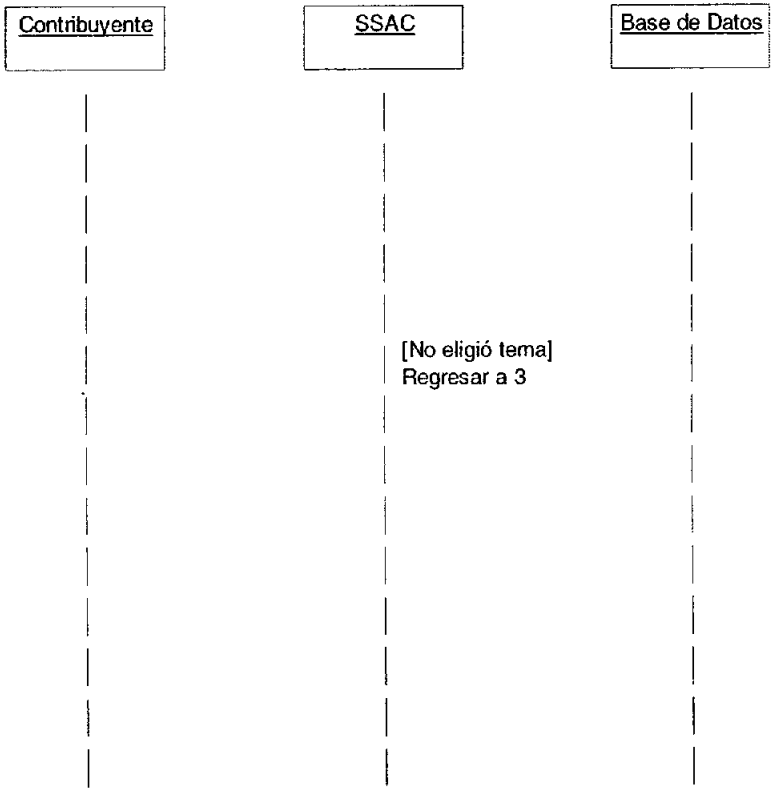
Posteriormente el administrador solicita al sistema reenviar la respuesta de atención del personal al Contribuyente. El sistema le pide el número de folio. El administrador captura el número de folio y el sistema muestra una pantalla con los datos del contribuyente y la respuesta que se le va a enviar. El administrador decide reenviar la respuesta y el sistema solicita la confirmación del reenvío. El administrador confirma el reenvío y el sistema muestra un mensaje indicando que la respuesta fue reenviada al contribuyente.

El estatus del folio pasa de "Conclusión Folio" a "Reenviar respuesta"

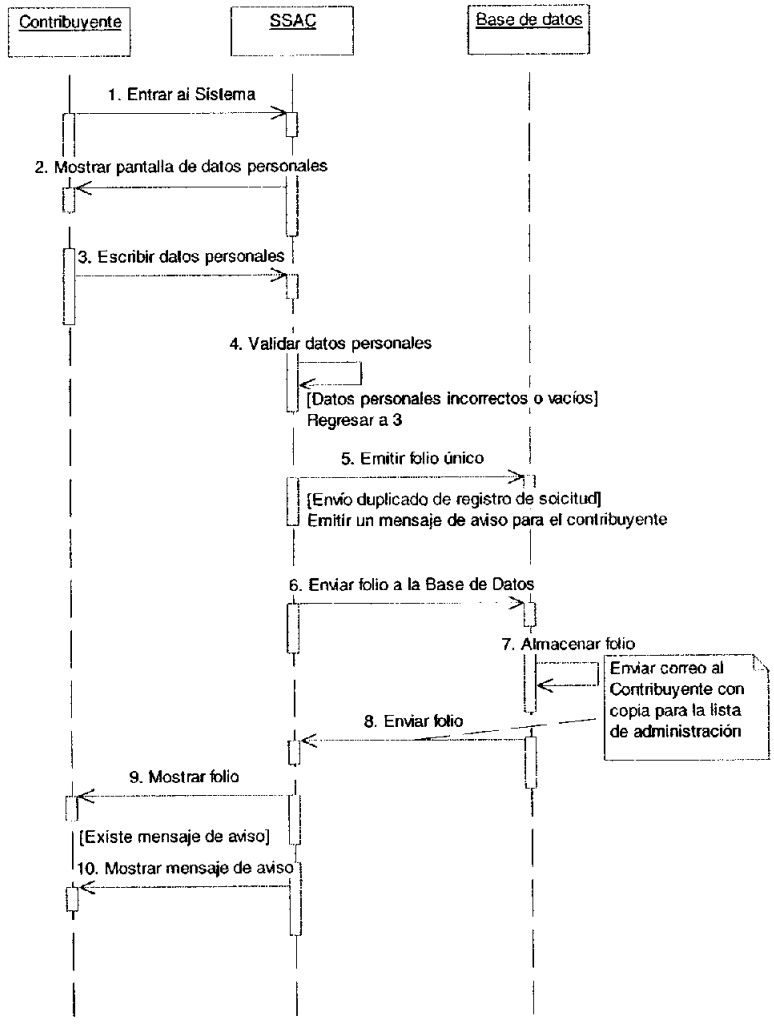
DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia especifican el orden en que las acciones se suscitan, así como lo actores involucrados para ello.

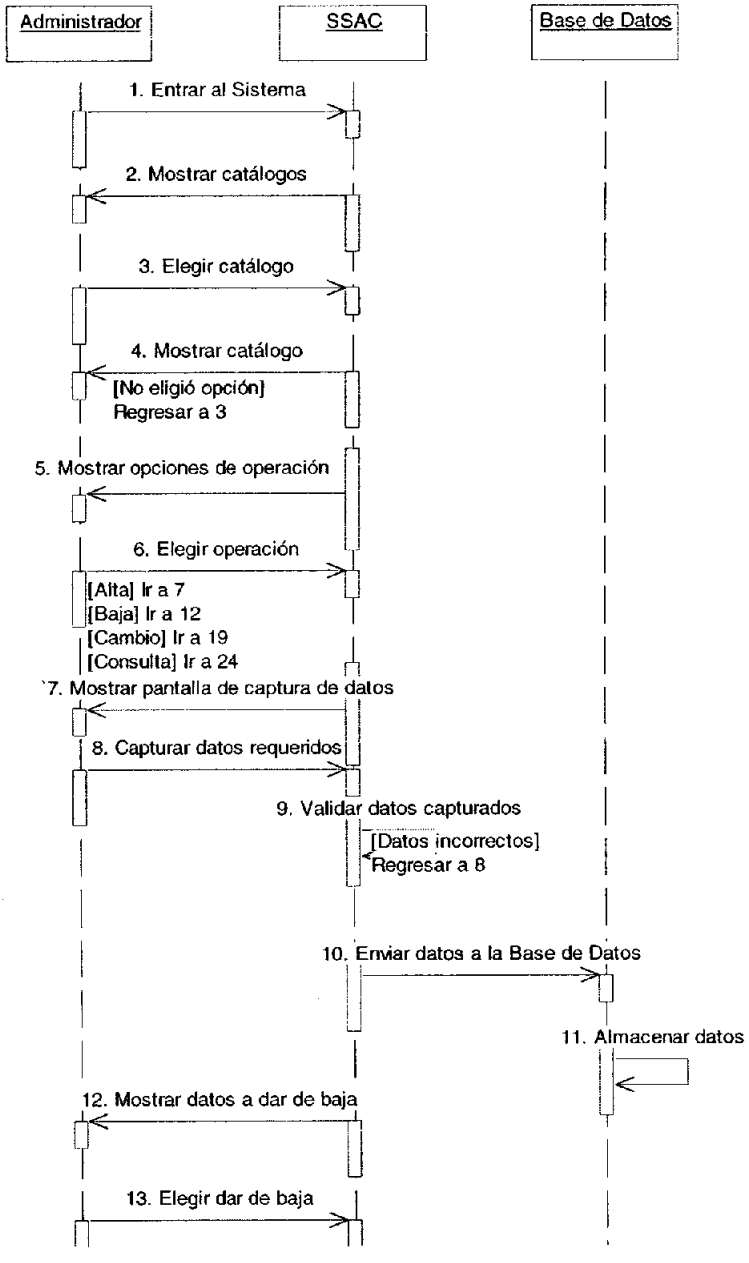
1. CONSULTAR AYUDA EN LÍNEA



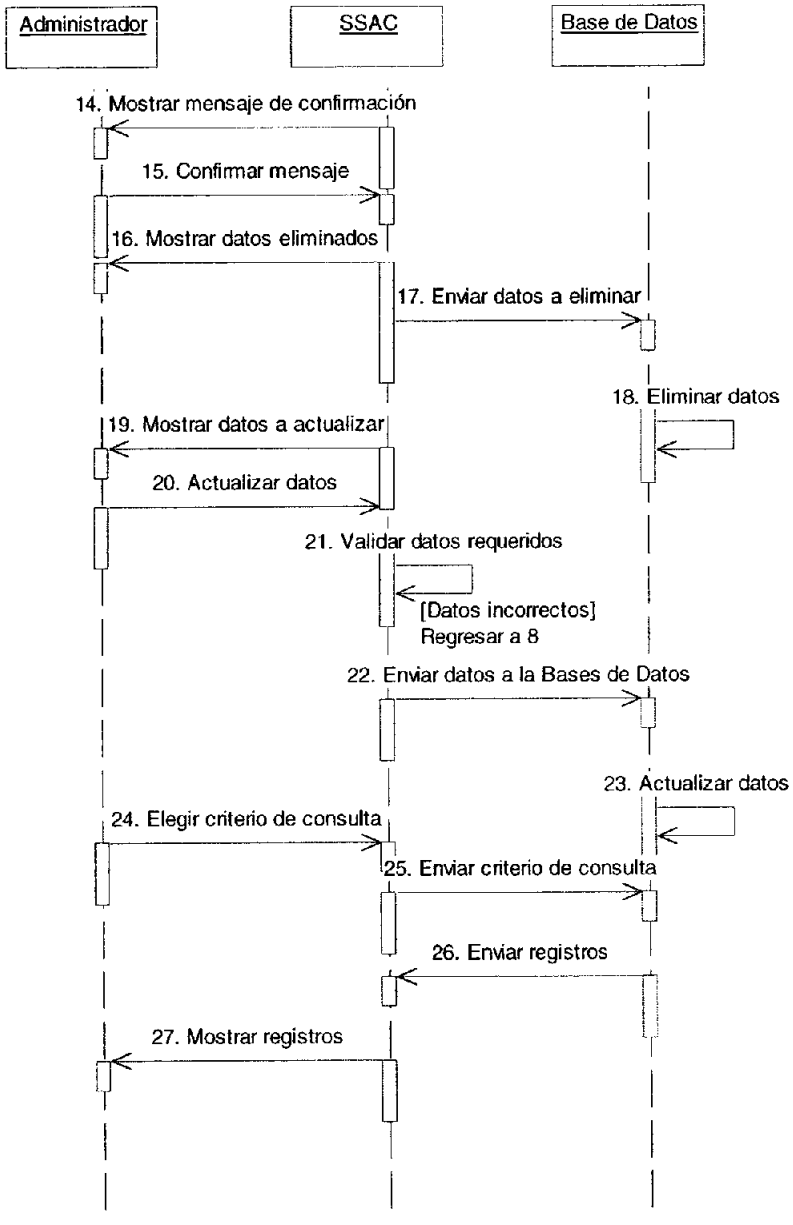
2. ENVIAR SOLICITUD



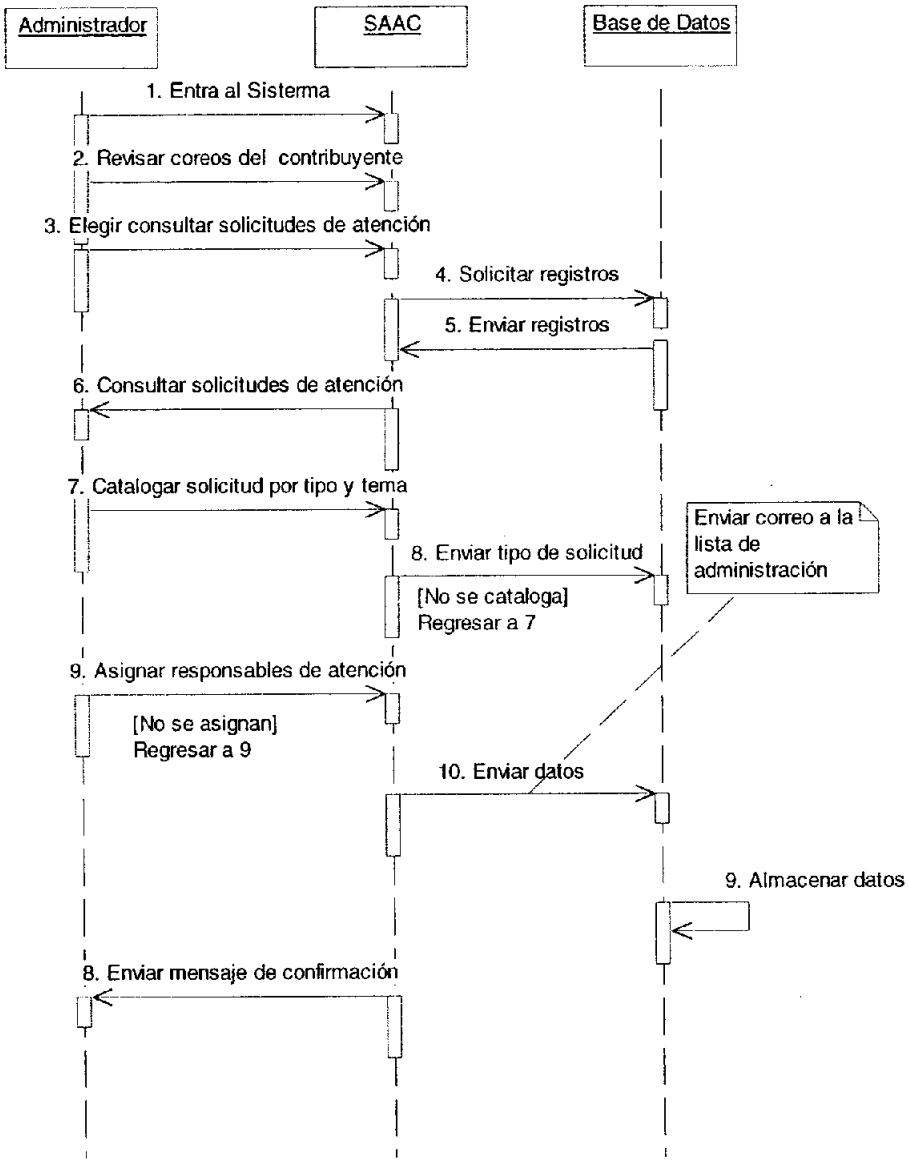
3. ADMINISTRAR CATÁLOGOS



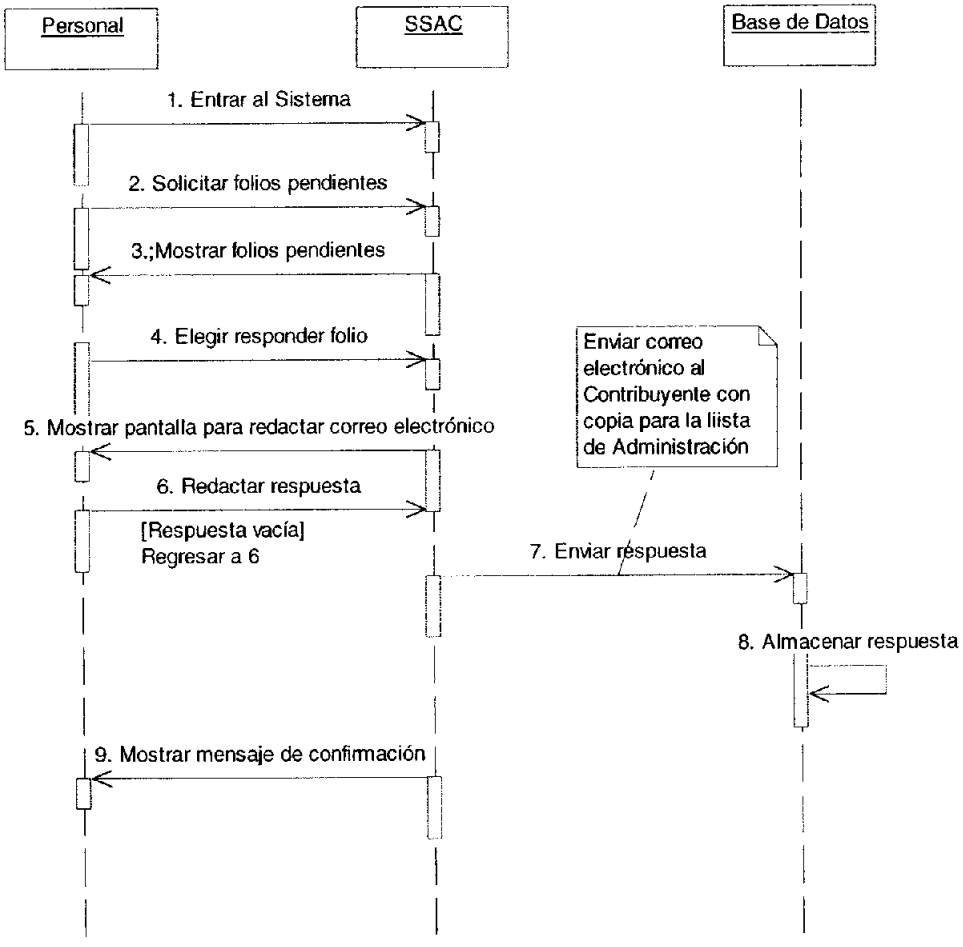
Continúa...


Administrador
SSAC
Base de Datos
14. Mostrar mensaje de confirmación
15. Confirmar mensaje
16. Mostrar datos eliminados
17. Enviar datos a eliminar
18. Eliminar datos
19. Mostrar datos a actualizar
20. Actualizar datos
21. Validar datos requeridos
[Datos incorrectos]
Regresar a 8
22. Enviar datos a la Bases de Datos
23. Actualizar datos
24. Elegir criterio de consulta
25. Enviar criterio de consulta
26. Enviar registros
27. Mostrar registros

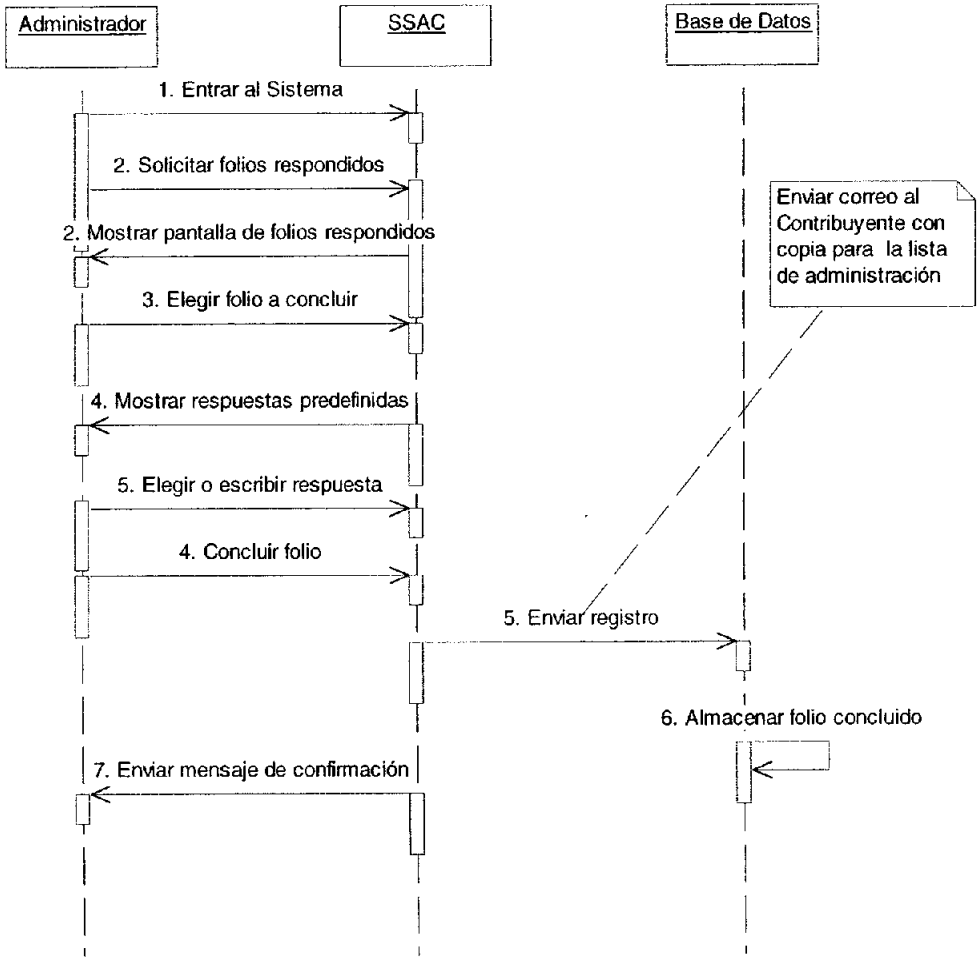
4. TURNAR FOLIO



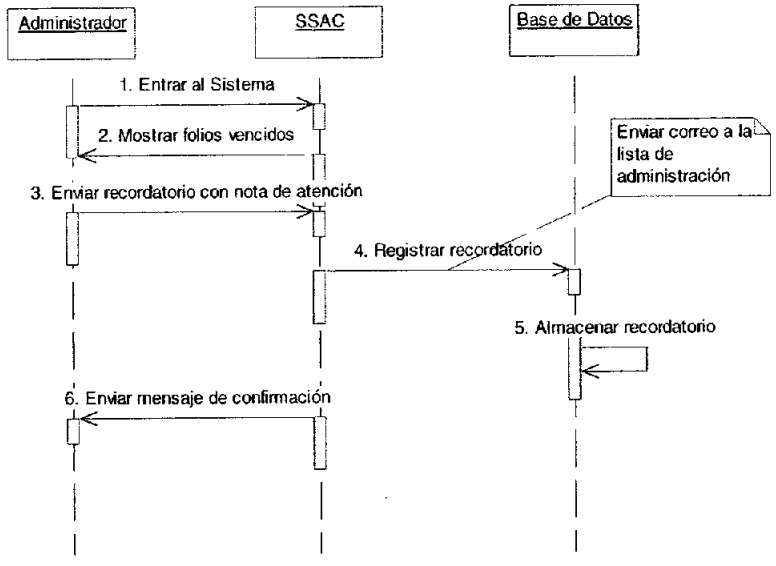
5. RESPONDER FOLIO



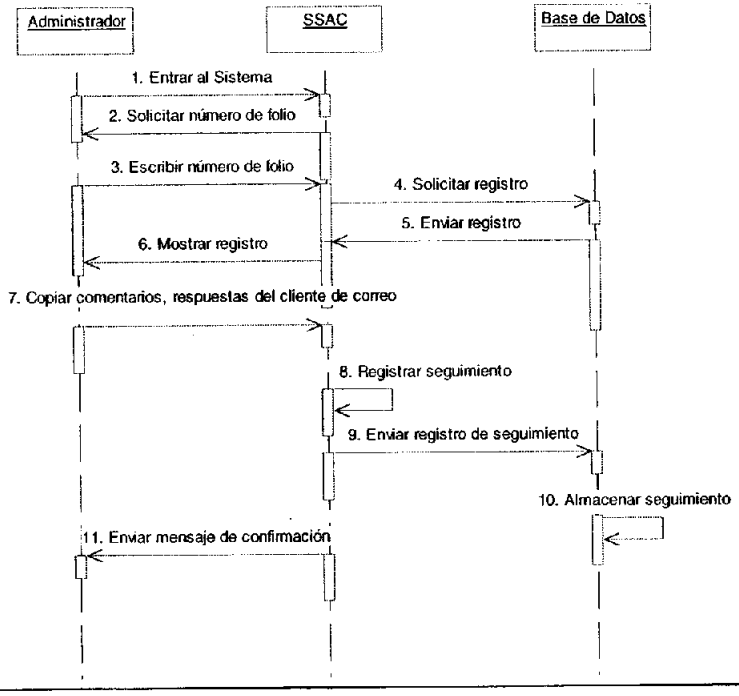
6. CONCLUIR FOLIO



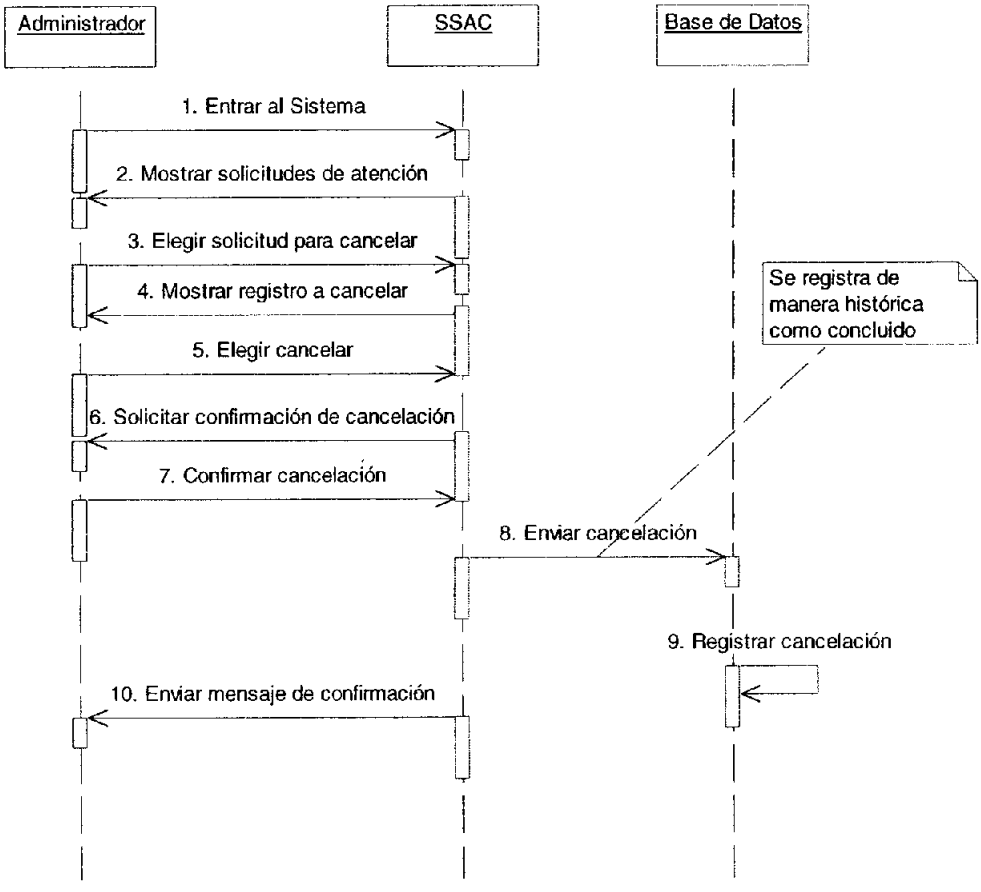
7. ENVIAR RECORDATORIO



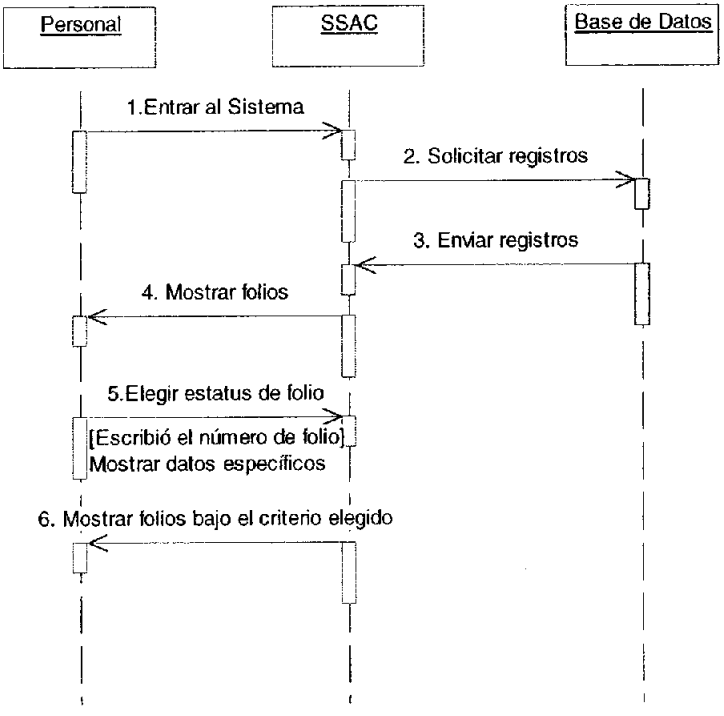
8. DAR SEGUIMIENTO A FOLIO



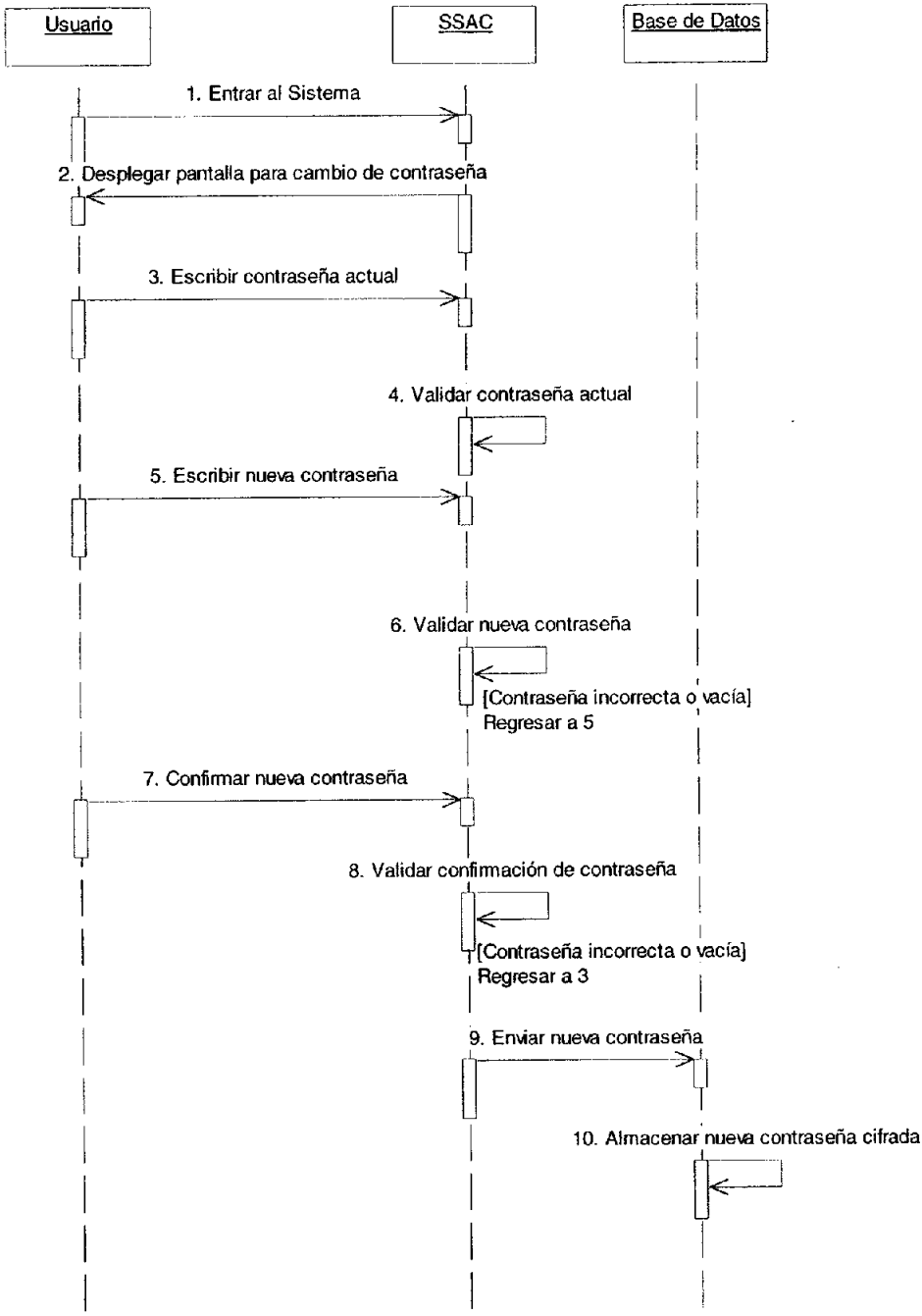
9. CANCELAR FOLIO



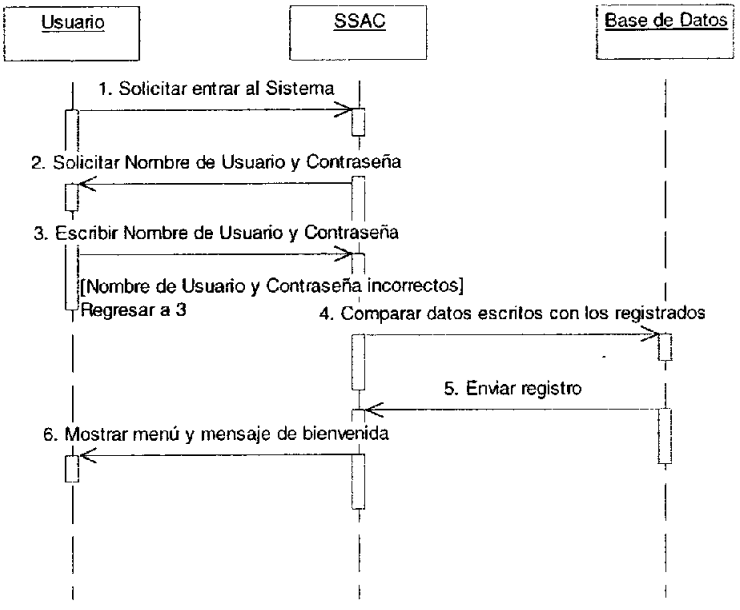
10. CONSULTAR FOLIO



11. CAMBIAR CONTRASEÑA

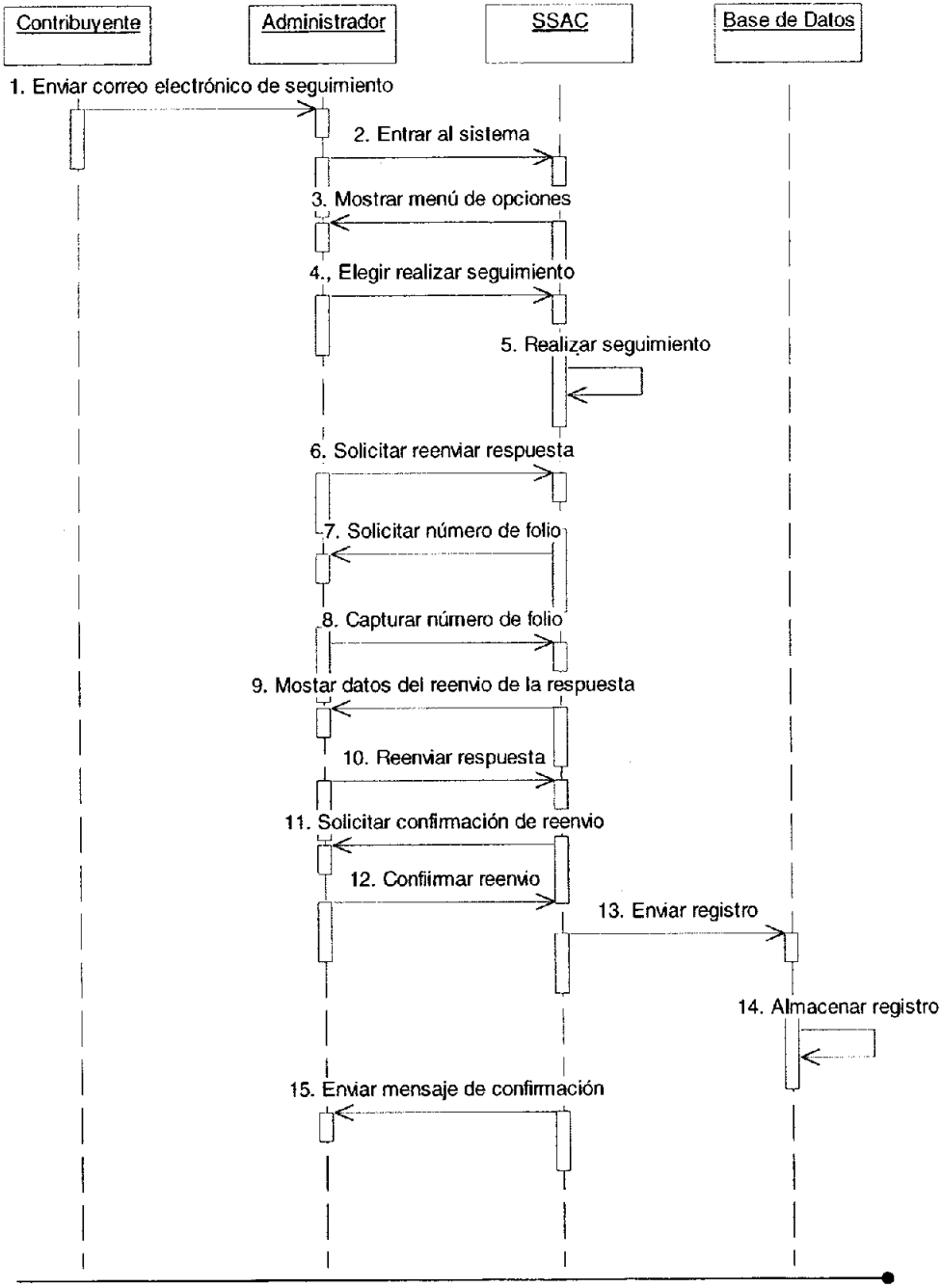


12. REGISTRAR AUTENTICACIÓN



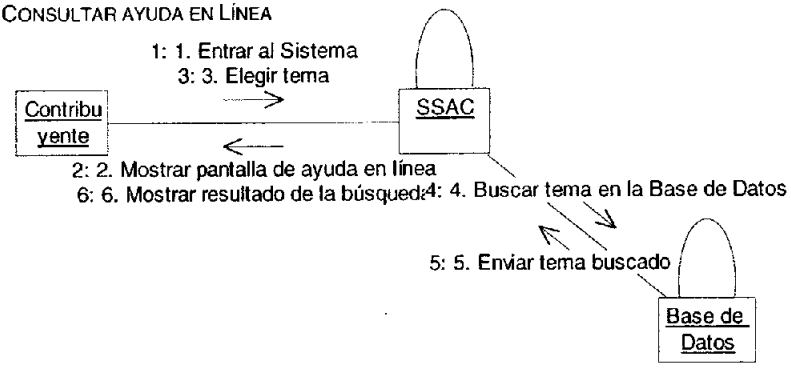
13. CONSULTAR REPORTES

14. REENVIAR RESPUESTA

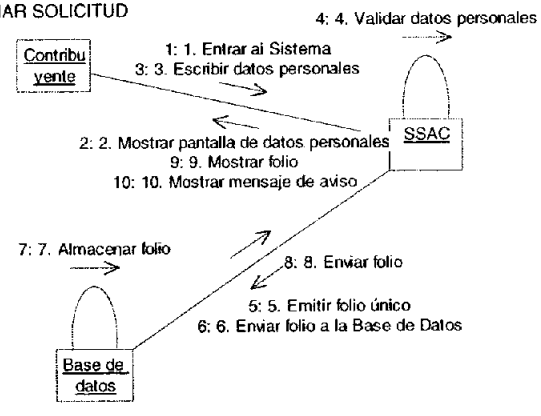


Diagramas de Colaboración

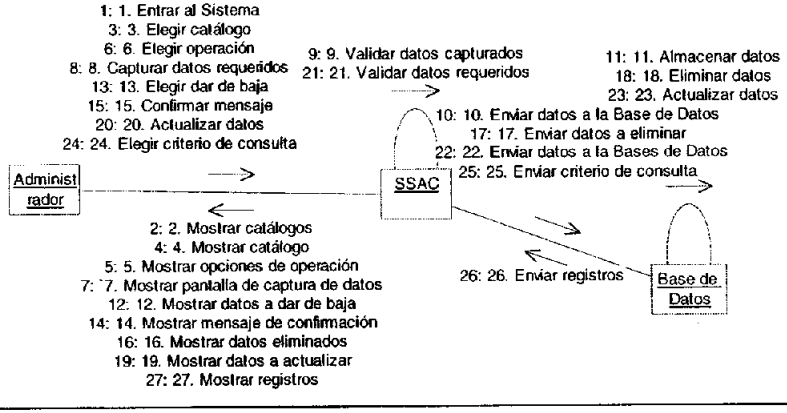
1. CONSULTAR AYUDA EN LÍNEA



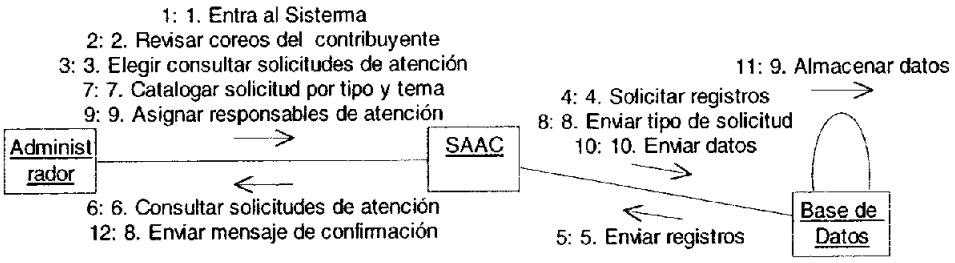
2. ENVIAR SOLICITUD



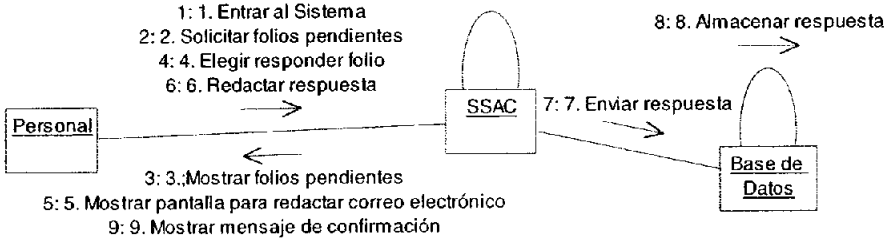
3. ADMINISTRAR CATÁLOGOS



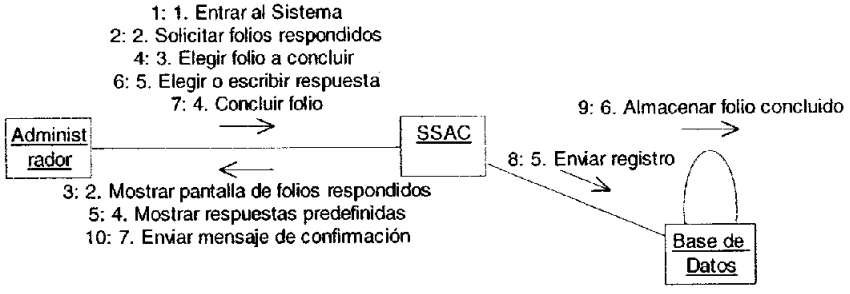
4. TURNAR FOLIO



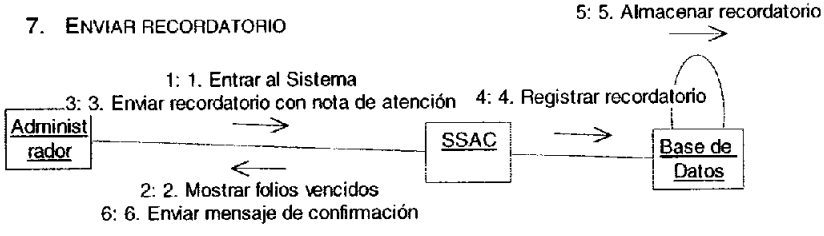
5. RESPONDER FOLIO



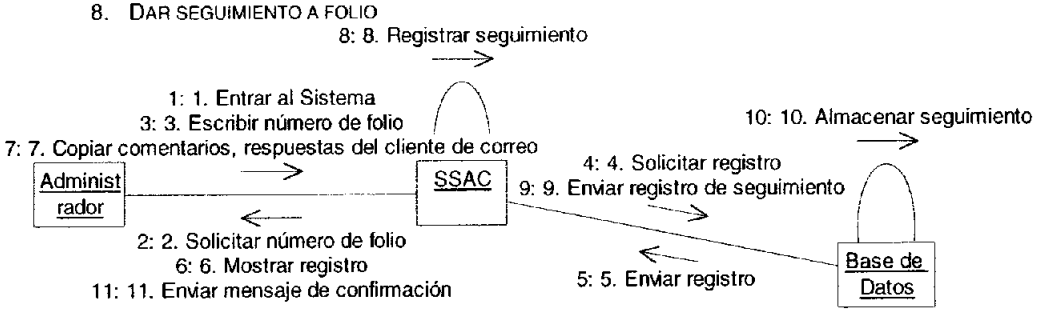
6. CONCLUIR FOLIO



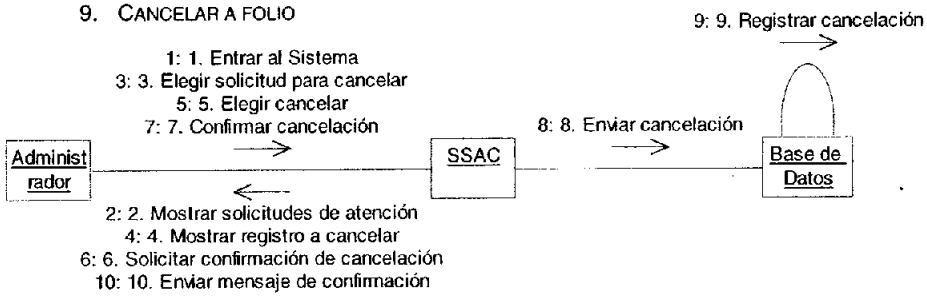
7. ENVIAR RECORDATORIO



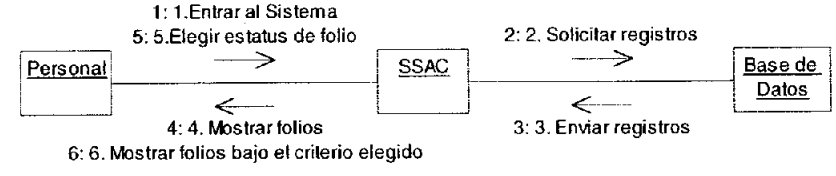
8. DAR SEGUIMIENTO A FOLIO



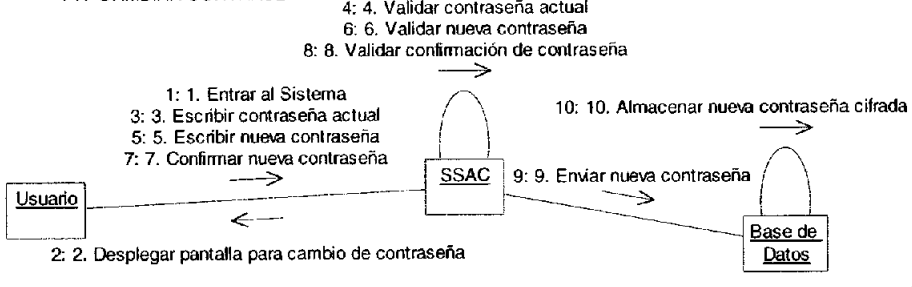
9. CANCELAR A FOLIO



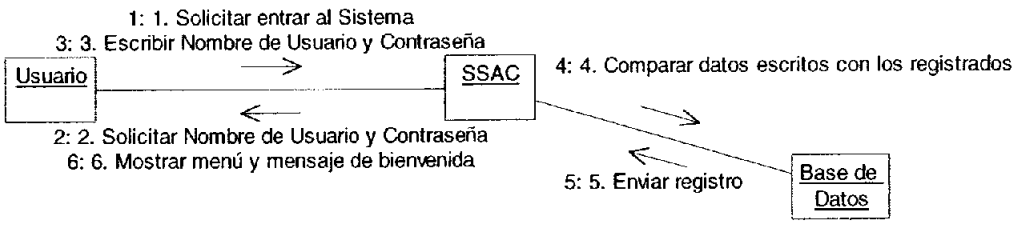
10. CONSULTAR FOLIO



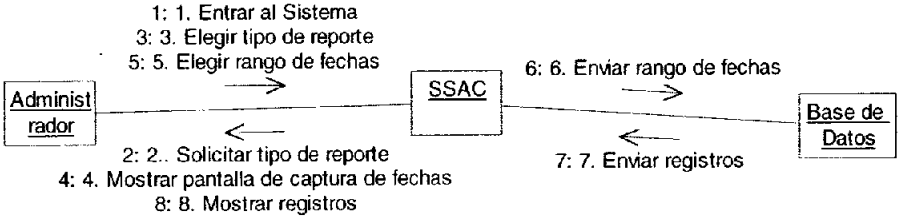
11. CAMBIAR CONTRASEÑA



12. REGISTRAR AUTENTICACIÓN



13. CONSULTAR REPORTES



14. REENVIAR RESPUESTA

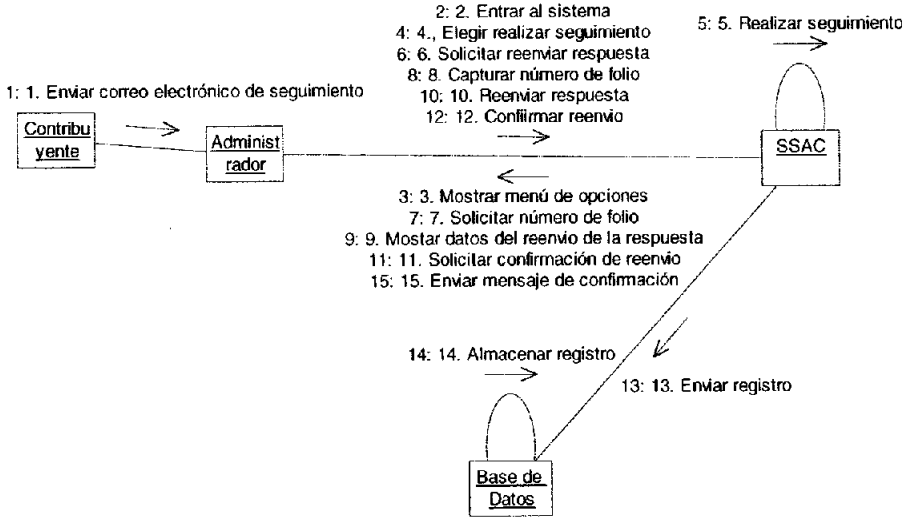
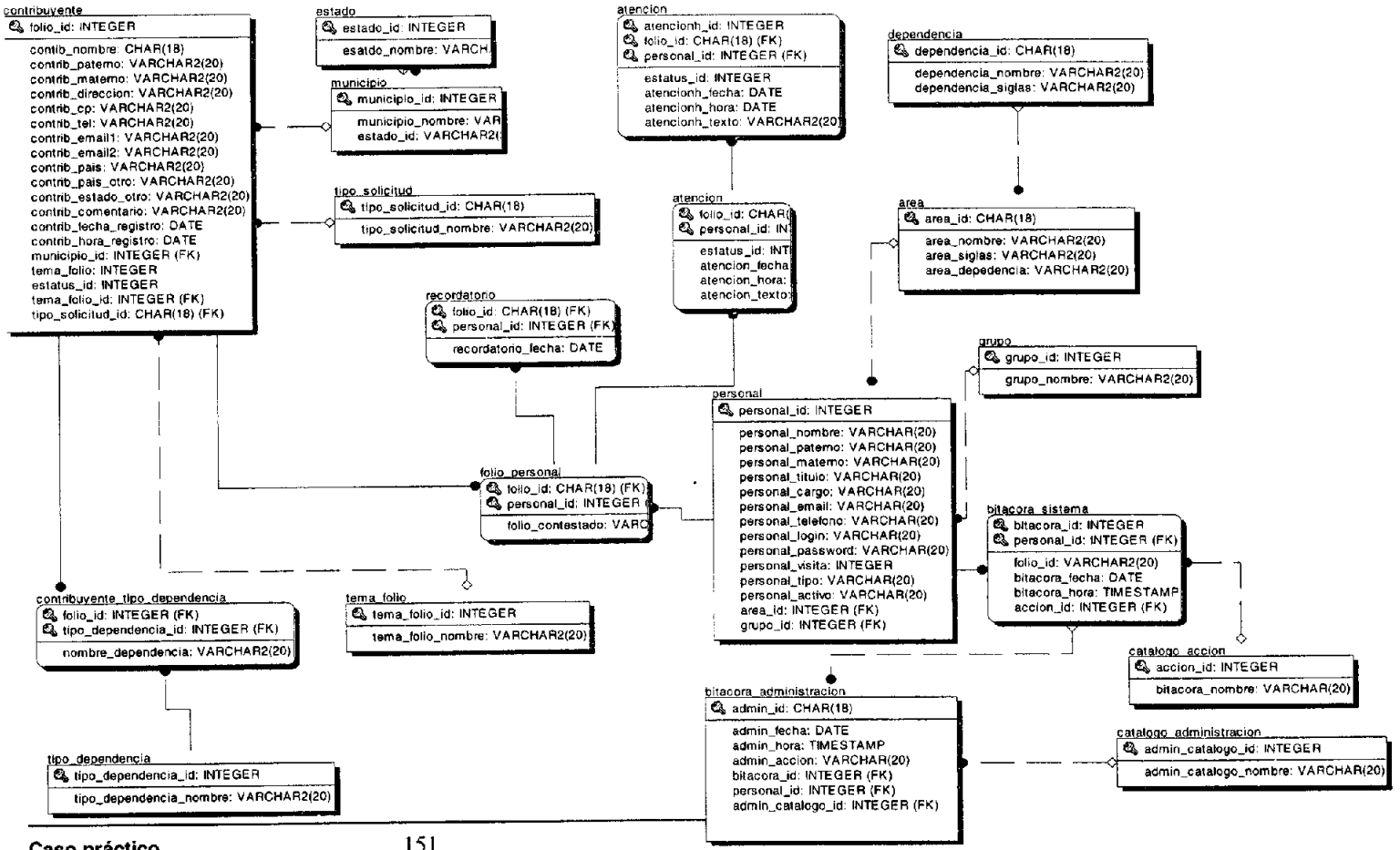


Diagrama Entidad-Relación de la Base de Datos



Diccionario de Datos

area

Registra el catálogo de áreas en las cuales se encuentra el personal

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
area_id	Clave del área en la cual se encuentra el personal que atiende el folio del Contribuyente	SI	NO	SI	INT2	1
area_nombre	Nombre del área en la cual se encuentra el personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(100)	Dirección General de Informática
area_siglas	Siglas del área en la cual se encuentra el personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(10)	DGI
dependencia_id	Clave de la dependencia en la cual se encuentra el área del personal que atiende el folio del Contribuyente	NO	SI	SI	INT2	3

atencion

Registra la última manera en la que se atendió un folio, la persona que lo atendió y en que momento fue atendido

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
folio_id	Clave del folio del Contribuyente	SI	SI	SI	VARCHAR(11)	ACF04040074
personal_id	Clave del personal que atiende el folio del Contribuyente	SI	SI	SI	INT2	3
estatus_id	Clave del Estatus del Folio	SI	SI	SI	INT2	4
atencion_fecha	Fecha en la que fue atendido un folio	NO	NO	SI	DATE	2004-04-05
atencion_hora	Hora en la que fue atendido un folio	NO	NO	SI	CHAR(8)	10:24:28
atencion_texto	Texto del correo electrónico con el que fue atendido el folio del contribuyente	NO	NO	NO	TEXT	Esperamos haberle servido con la respuesta enviada anteriormente, quedamos a sus órdenes. Reciba un cordial saludo.

atencionh

Registra el historial de todas las maneras en la que se atendió un folio, la persona que lo atendió y en que momento fue atendido

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
atencionh_id	Clave consecutiva del registro histórico de la atención	SI	NO	SI	INT2	2
folio_id	Clave del folio del Contribuyente	SI	SI	SI	VARCHAR(11)	ACF04040077
personal_id	Clave del personal que atiende el folio del Contribuyente	SI	SI	SI	INT2	4
estatus_id	Clave del estatus en el que se encuentra el folio del Contribuyente	NO	NO	SI	INT2	5
atencionh_fecha	Fecha en la que fue atendido un folio	NO	NO	SI	DATE	2004-04-05
atencionh_hora	Hora en la que fue atendido un folio	NO	NO	SI	CHAR(8)	11:20:09
atencionh_texto	Texto del correo electrónico con el que fue atendido el folio del contribuyente	NO	NO	NO	TEXT	Esperamos haberle servido con la respuesta enviada anteriormente, quedamos a sus órdenes. Reciba un cordial saludo.

contribuyente

Registra los datos del Contribuyente que envía una solicitud de Atención en el Sistema de SSAC

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
contrib_pais_otro	Nombre del País del Contribuyente	NO	NO	NO	VARCHAR(60)	Ecuador
contrib_estado_otro	Nombre del Estado del País del Contribuyente	NO	NO	NO	VARCHAR(60)	Quito
contrib_pais	Registra la clave de México	NO	NO	SI	CHAR(1)	M
contrib_email_opcional	Correo electrónico opcional del Contribuyente	NO	NO	NO	VARCHAR(100)	mexa@prodigy.net.mx
municipio_id	Clave del Municipio del Contribuyente	NO	SI	SI	INT2	1

contrib_comentario	Solicitud de atención del Contribuyente	NO	NO	SI	TEXT	Compré un auto nuevo, tengo un permiso para circular con el número de placas por un periodo de 60 días, se me venció y lo sellé por otros 60 días. Mi pregunta es: ¿Cuándo me llegarán las placas o en dado caso en dónde puedo recogerlas, en dado caso que no haya estado en mi domicilio?
contrib_hora_registro	Hora de registro de la solicitud	NO	NO	SI	CHAR(8)	17:11
contrib_fecha_registro	Fecha de registro de la solicitud	NO	NO	SI	DATE	2004-04-07
tipo_solicitud_id	Clave del Tipo de Solicitud del Contribuyente	NO	SI	NO	INT2	1
contrib_nombre	Nombre(s) del Contribuyente	NO	NO	SI	VARCHAR(30)	Alan
contrib_paterno	Apellido paterno del Contribuyente	NO	NO	SI	VARCHAR(20)	Godinez
folio_id	Clave del folio del contribuyente	SI	NO	SI	VARCHAR(11)	ACF02070009
contrib_email	Correo electrónico del Contribuyente	NO	NO	SI	VARCHAR(100)	alan@prodigy.net.mx
contrib_tel	Número telefónico del Contribuyente	NO	NO	SI	VARCHAR(40)	56-77-15-96
contrib_materno	Apellido materno del Contribuyente	NO	NO	SI	VARCHAR(20)	Castillo
contrib_cp	Código Postal del Contribuyente	NO	NO	NO	NUMBER(5)	
contrib_direccion	Calle, No. Ext., No. Int. y Colonia del Contribuyente	NO	NO	NO	VARCHAR(120)	Retorno de Rosa Zaragza No. 38 Col. CTM Culhuacan

contribuyente_tipo_dependencia

Registra el Nombre de la Dependencia a la que pertenece el Contribuyente

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
folio_id	Clave del folio del Contribuyente	SI	SI	SI	VARCHAR(11)	ACF02070016
tipo_dependencia_id	Clave del tipo de Dependencia del Contribuyente	SI	SI	SI	INT2	1
nombre_dependencia	Nombre de la Dependencia del Contribuyente	No	NO	SI	VARCHAR(100)	Instituto Politécnico Nacional

dependencia

Registra el catálogo de dependencias en las cuales se encuentra el personal

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
dependencia_id	Clave de la dependencia en la cual se encuentra el área del personal que atiende el folio del Contribuyente	SI	NO	SI	INT2	3
dependencia_nombre	Nombre de la dependencia en la cual se encuentra el área del personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(100)	Secretaría de Finanzas
dependencia_siglas	Siglas de la dependencia en la cual se encuentra el área del personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(10)	SF

estado

Registra el catálogo de las Entidades Federativas de la República Mexicana

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
estado_id	Clave de la Entidad Federativa de la República Mexicana	SI	NO	SI	INT2	1
estado_nombre	Nombre de la Entidad Federativa de la República Mexicana	NO	NO	SI	VARCHAR(30)	Aguascalientes

estatus

Registra el catálogo de los Estatus en los que se encuentra un folio durante su proceso de atención

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
estatus_id	Clave del Estatus del Folio	SI	NO	SI	INT2	1
estatus_nombre	Nombre del Estatus en el que se encuentra un folio (recepción, turnado, concluido, entre otros)	NO	NO	SI	VARCHAR(25)	Concluido

folio_personal

Registra que personal atiende que folios de los contribuyentes

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo

folio_id	Clave del folio del contribuyente	SI	SI	SI	VARCHAR(11)	ACF04040074
personal_id	Clave del personal que atiende el folio del Contribuyente	SI	SI	SI	INT2	8
folio_contestado	Indicador que permite verificar que personal a contestado que folios	NO	NO	SI	CHAR(1)	S

grupo

Registra el catálogo de grupos que puede tener un usuario del administración del sistema

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
grupo_id	Clave del grupo	SI	NO	SI	INT2	1
grupo_nombre	Nombre del grupo	NO	NO	SI	CHAR(7)	Total

municipio

Registra el catálogo de las diferentes delegaciones y municipios de los Estados de la República Mexicana

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
municipio_id	Clave de la Delegación o Municipio	SI	NO	SI	INT2	9
estado_id	Clave del Estado al que pertenece la Delegación o Municipio	NO	SI	SI	INT2	2
municipio_nombre	Nombre de la Delegación o Municipio	NO	NO	SI	VARCHAR(60)	Toluca

personal

Registra el catálogo del personal que atiende la solicitud del contribuyente

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
personal_id	Clave del personal que atiende el folio del Contribuyente	SI	NO	SI	INT2	2
personal_nombre	Nombre(s) del personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(30)	Jesús

personal_paterno	Apellido paterno del personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(30)	Tafoya
personal_materno	Apellido materno del personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(30)	González
personal_titulo	Título del personal que atiende el folio del Contribuyente (Lic., Ing., entre otros).	NO	NO	NO	VARCHAR(6)	Ing.
personal_cargo	Cargo en el cual se encuentra el personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(50)	JUD de Análisis de Sistemas
personal_email	Correo electrónico del personal que atiende el folio del Contribuyente	NO	NO	SI	VARCHAR(100)	jtafoya@finanzas.df.gob.mx
personal_telefono	Número telefónico del personal que atiende el folio del Contribuyente	NO	NO	NO	VARCHAR(100)	51-34-25-00 Ext 1568
personal_login	Nombre de usuario del personal	NO	NO	SI	VARCHAR(10)	jtafoya
personal_password	Contraseña del personal	NO	NO	SI	CHAR(32)	4534543t43655y65y6y56yu6y6yjhgr6
personal_visita	Número de veces que el personal se loguea en el sistema	NO	NO	SI	INT2	4
personal_tipo	Caracter que indica si el personal es de tipo Administrador o es de tipo Personal de Atención. puede tomar los valores A o P.					
personal_activo	Caracter que indica si el personal está activo para responder folios o no, puede tomar los valores S o N.	NO	NO	SI	CHAR(1)	S
area_id	Clave del área en la cual se encuentra el personal que atiende el folio del Contribuyente. Para el administrador, el campo deberá aparecer con valor nulo	NO	NO	NO	INT2	2
grupo_id	Clave del grupo	NO	SI	SI	INT2	1

pregunta

Registra el catálogo de preguntas más frecuentes que puede realizar el contribuyente

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
pregunta_id	Clave de la pregunta	SI	NO	SI	INT2	4
tema_id	Clave del tema	NO	SI	SI	INT2	3
pregunta_nombre	Redacción de la pregunta	NO	NO	SI	TEXT	¿Qué debo hacer para pagar un acta de nacimiento?
pregunta_texto	Respuesta a la pregunta	NO	NO	SI	TEXT	Acudir a las oficinas del Registro civil con la línea de captura correspondiente previamente pagada en el Banco de su preferencia
pregunta_estatus	Estado de la pregunta (si está publicada o no)	NO	NO	SI	CHAR(1)	S

recordatorio

Registra la fecha en la que se emite un recordatorio de atención de un folio

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
personal_id	Clave del personal que atiende el folio del Contribuyente	SI	SI	SI	INT2	1
recordatorio_fecha	Fecha de recordatorio de atención del folio	NO	NO	SI	DATE	2004-04-05
folio_id	Clave del folio del contribuyente	SI	SI	SI	VARCHAR(11)	ACF04040074

tema

Registra el catálogo de temas en los cuales puede clasificarse la pregunta del contribuyente

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
tema_id	Clave del tema	SI	NO	SI	INT2	1
tema_nombre	Nombre del tema	NO	NO	SI	VARCHAR(80)	Predial
tema_estatus	Estado del Tema (Si está publicado o no)	NO	NO	SI	CHAR(1)	N

tipo_dependencia

Registra el catálogo del Tipo de Dependencia a la que pertenece el Contribuyente

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
tipo_dependencia_id	Clave del Tipo de Dependencia del Contribuyente	SI	NO	SI	INT2	2
tipo_dependencia_nombre	Nombre de la dependencia del Contribuyente	NO	NO	SI	VARCHAR(15)	Gobierno

tipo_solicitud

Registra el catálogo del Tipo de Solicitud de Atención del Contribuyente

Nombre del Atributo	Descripción	PK	FK	NN	Tipo de Dato	Ejemplo
tipo_solicitud_id	Clave del Tipo de Solicitud del Contribuyente	SI	NO	SI	INT2	5
tipo_solicitud_nombre	Nombre del Tipo de Solicitud (Queja, sugerencia, solicitud de información, entre otras)	NO	NO	SI	VARCHAR(30)	Demanda de Servicio

Bibliografía

- **Anaya Álvarez, Carlos**; *Máxima seguridad en Internet* , Madrid : Anaya Multimedia, 1998
- ***Apuntes de "Administración de Servidores WWW con Linux"***, Diplomado "Desarrollo e Implementación de Aplicaciones Web con software libre y Linux"
- **Darren James Harkness**; *Apache essentials : install, configure, maintain,* Birmingham Friendsof, c2004
- **Dee-Ann LeBlanc**; *La Biblia de administración de sistemas Linux,* Madrid : Anaya Multimedia, c2001
- **Easttom, Hingham**; *Moving from Windows to Linux:* Chuck, Massachusetts : C. River Media, c2004
- **Facundo Arena, Héctor**; *Linux Fácil: Manual con CD-ROM.*, MP Ediciones, 2000
- **Gallego Vázquez, José Antonio**; *Desarrollo Web con PHP y MySQL*, Madrid : Anaya Multimedia, c2003
- **López Quijado, José**; *Domine HTML y DHTML* México : Alfaomega : Ra-Ma, c2003
- **Marcelo Rodao, Jesús de**; *Piratas ciberneticos : cyberwars, seguridad informatica e Internet,* Madrid : Ra-Ma, deposito legal 2001
- **Martin, Michel**; *De windows a linux : para distribuciones red hat y Suse* ; traducción Arantxa Galdós, Roberto Tato, Barcelona : Marcombo : Alfaomega, 2001
- **Matt Welsh, Matthias; Dalheimer, Kalle; Kaufman Lar,**; O'Reilly, *Linux. Guía de referencia y aprendizaje.*, Ed. Anaya Multimedia, 2000
- **Mohammed J. Kabir**; *La Biblia de servidor Apache 2,* Madrid : Anaya Multimedia, c2003
- **Morrill, Daniel** ; *Configuración de sistemas LINUX,* Madrid : Anaya Multimedia, c2003
- **Morrison, Michael**; *HTML & XML for beginners*, Redmond, Washington : Microsoft, 2001
- **Nombela, Juan José**; *Seguridad informática,* Madrid : Paraninfo : ITP, 1997
- **Wisniewski, John Robert**; *Linux & OpenVMS interoperability : tricks for old dogs, new dogs, and hot dogs with open systems* , Boston, Massachusetts: Digital, c2003
- **Ziegler, Robert**; *Guía avanzada Firewalls Linux,* traducción José Ignacio Sánchez , Eva Maria López, Madrid, Alfaomega ,2000
- http://www.htmlpoint.com/apache/01c.htm