



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

“DIPLOMADO DE DESARROLLO E IMPLEMENTACION
DE SISTEMAS CON SOFTWARE LIBRE
EN LINUX”

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A:
AXEL MURILLO ZARCO

ASESOR:
ING. JOSE MANUEL QUINTERO CERVANTES

MÉXICO

ENERO 2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatorias

A mis padres:

Ernesto Murillo Hernández

Maria Esther Zarco Fernández

Por estar siempre apoyándome en todo lo que he emprendido, por que me han enseñado a perseverar para alcanzar las metas que me propongo. Por todas aquellas llamadas de atención recibidas que me han hecho aprender de mis errores. Por enseñarme a apreciar la vida e inculcarme valores que me permitirán llevar una vida honesta y respetable. Agradezco todo el amor y confianza que me han brindado, les dedico éste logro con el compromiso de seguir esforzándome a cada momento.

Los quiero mucho papás.

A mis hermanas:

Viridiana Astrid Murillo Zarco

Talia Murillo Zarco

Por el apoyo que he recibido de ellas que me permitió llegar hasta éste momento y que de igual forma tengo la seguridad que ellas también lo lograrán. Por los buenos y malos momentos que hemos pasado juntos, los cuáles nos han permitido fortalecer nuestra relación. Deseo se forjen como universitarias de excelencia y realicen todos sus sueños.

Las quiero mucho hermanitas.

A mis maestros:

Por las enseñanzas que nos han brindado dentro del salón de clases, pero además por las lecciones de vida que nos han brindado al contarnos sus experiencias profesionales. Por su compromiso con los alumnos y por inculcarnos el amor y el respeto a nuestra profesión y a la universidad.

A mis amigos y compañeros de la universidad:

Por compartir grandes momentos durante el tiempo que estudiamos juntos, así como también fuera de las aulas, como en reuniones, excursiones, etc. Por aprender de manera conjunta conocimientos y compartir experiencias. Por la confianza que me brindaron y que me tendieron su mano en los momentos difíciles.

A la Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón:

Por enseñarnos la importancia de comprometernos con nuestra Universidad y con México. Por darnos las herramientas para desenvolvemos en el mundo laboral y desarrollarnos como personas útiles a la sociedad.

Índice

Introducción	1
Objetivo	2
Capítulo I.	
Informe de actividades y contenidos del diplomado	4
Módulo I.	
Informe de actividades y contenidos del diplomado	3
Módulo II.	
Administración de Linux	6
Módulo III.	
Editores para la creación de Páginas Web	10
Módulo IV.	
Administración de Servidores WWW con Linux	13
Módulo V.	
Programación con PHP	16

Módulo VI.

Interacción de WWW con Base de Datos 20

Módulo VII.

Introducción a la Seguridad en Cómputo 23

Módulo VIII.

Desarrollo de aplicaciones Web con PHP y PostgreSQL 29

Capítulo II. Aplicaciones desarrolladas 35

Capítulo III. Reflexiones y comentarios. 43

INTRODUCCIÓN

El software libre desde su nacimiento ha tenido un crecimiento constante, y en estos últimos años ha crecido considerablemente. Muchos proyectos de software abierto han madurado hasta el punto de ser una alternativa viable contra proyectos que son desarrollados por grandes empresas. El software abierto tiene la virtud de que el código fuente de los proyectos que trabajen bajo ésta modalidad, están a disposición de cualquier programador o usuario, que estén interesados en entender como trabaja el software, así mismo están en la posibilidad de mejorarlo o adaptarlo a sus propias necesidades. Gracias a ésta forma de trabajo, es posible localizar fallos o problemas de seguridad en los programas de una forma bastante rápida.

La mayoría de los programas de software abierto están amparados bajo la licencia GPL, que explicada de una manera general permite copiar, modificar, y distribuir libremente el software, con la única condición que dichas modificaciones sean publicadas a toda la comunidad, con el fin de establecer una mejora continua en los proyectos.

Una de las principales ventajas del software libre es que no hay que realizar pago alguno por su uso. De manera que aquel dinero destinado para licencias puede utilizarse en otros rubros de la empresa. En el sector Gobierno, dado el tamaño del mismo, el ahorro es bastante considerable, y el ahorro podría destinarse en obras sociales, mejora de salarios, etc. Varias naciones Europeas han decidido optar por soluciones basadas en Software abierto para su uso en todas sus dependencias, no sólo por cuestión monetaria, si no por cuestión de seguridad. Esto es por que, al contar con el código fuente se conoce perfectamente la forma en la que trabaja y como se mencionó anteriormente se pueden corregir detalles en seguridad y estabilidad del software, cosa que no sucede con el software cerrado.

El movimiento del software tiene varios años que inició, pero su principal avance se dio a partir de la creación de Linux, y su posterior ingreso a este movimiento. A partir de este momento se incrementaron el número de proyectos. Estos proyectos van desde el desarrollo de sistemas operativos, manejadores de base de datos, lenguajes de programación, paquetes de ofimática, diseño gráfico, software multimedia, etc. Y muchos de estos proyectos pueden ejecutarse en distintas plataformas tales como Linux, Solaris, Windows, MacOS X.

Como puede apreciarse el software libre es una propuesta viable para el desarrollo de sistemas que ofrezcan seguridad y estabilidad en su uso, además de poder ajustarse a las necesidades de cada organización.

El presente trabajo está compuesto por el Capítulo I, donde se hace una breve descripción del contenido de cada uno de los módulos, en el Capítulo II se hace una explicación de los proyectos desarrollados en el transcurso del diplomado, y para finalizar el trabajo se incluye una sección de Reflexiones y Comentarios sobre el diplomado.

Objetivo

Presentar diversas herramientas en software libre y estándares abiertos para el desarrollo de sistemas de cómputo, que permitan el manejo seguro de la información. El diplomado abarca desde el aprendizaje del sistema operativo Linux, técnicas de administración de Linux, el manejo del lenguaje HTML para la creación de las paginas Web, la puesta en marcha de un servidor de páginas Web, el aprendizaje del lenguaje PHP para dar dinamismo a las páginas en base a la información que se reciba, el manejo de base de datos con MySQL y PostgreSQL según el volumen de información a tratar y técnicas básicas de seguridad.

Capítulo I. Informe de actividades y contenidos del diplomado

En esta parte se presenta un resumen donde se describe el contenido de cada módulo en el transcurso del diplomado.

Módulo I. Sistema Operativo Linux

El objetivo de éste curso, fue la de aprender las bases para comenzar a manejar el Sistema Operativo Linux. Las razones principales por las cuales se consideró la utilización de Linux como S.O. de trabajo fue por que ofrece diversas características tales como estabilidad, seguridad, excelente forma de administración de recursos, además de que cuenta con la capacidad de poder moldear el sistema según las necesidades particulares. Además Linux tiene la ventaja de ser libre, ya que, el código del kernel, se puede copiar, distribuir y modificar sin ninguna restricción, permitiendo tener un sistema fiable cuyo núcleo está en constante actualización.

Linux cuenta con diversos “sabores” o distribuciones, cada una con características añadidas y propias según las necesidades de los usuarios. Algunas distribuciones de Linux son “libres de costo” y “abiertas”, es decir, no se debe realizar algún pago para su descarga, instalación y uso, y es “abierta”, puesto que el código de todo los programas incluidos, puede ser visto y modificado por cualquier persona. Éstas distribuciones son desarrolladas y mantenidas por la comunidad internacional, que se dedica a revisar el código del Kernel para encontrar errores o “bugs”, fallas de seguridad, y posteriormente corregir dichos problemas, así como la creación de controladores para distintos dispositivos, etc.. Hay muchas distribuciones que con el paso del tiempo se han consolidado y ofrecen un excelente producto a la altura de cualquier S.O. de paga. Ejemplo de estas distribuciones son Debian, Slackware, Gentoo, etc.

También existen versiones de Linux que son de paga, versiones que son desarrolladas y mantenidas por una compañía que a su vez tiene un equipo de desarrollo, dedicado a programar aplicaciones especiales, asistentes de configuración, etc., para esa versión de Linux. Éstas distribuciones cuentan con ciertos agregados como el reconocimiento de mas hardware (que en varias ocasiones es cerrado), herramientas de configuración mas accesibles para el usuario. Algunos ejemplos de estas distribuciones son Lindows (ahora Linspire), Red Hat, SuSe.

Linux inició su desarrollo durante el año de 1991, dicho proyecto fue elaborado por el finlandés Linus Torvals, Linux comenzó como un hobby. Su creación se basó en el código de otro sistema operativo conocido como Minix

desarrollado por el profesor Andrew Tenenbaum, de la Universidad de Vrije, en Ámsterdam. A continuación se cita algunos de los primeros mensajes publicados en algunos foros por Linus Torvalds anunciando el nacimiento de Linux, aunque ni el propio autor sabría como se llamaría.

```
From: torvalds@klaava.helsinki.fi (linus Benedict Torvalds)
Newsgroups: com.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: 1991Aug25.205708.9441@klaava.helsinki.fi
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Hello everybody out there using minix
I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu)
for 386(486) AT clones.....
```

Hola a todos allá afuera usando minix.

```
Estoy haciendo un sistema operativo libre (es solo un hobby, no será grande ni profesional
como GNU) pero funciona en 386(486) AT clones, lo he estado cocinando desde abril y está
quedando listo. Quisiera alguna retroalimentación de las cosas que a ustedes les gustan y no
les gustan de minix, ya que mi sistema operativo se parece (la misma disposición física,
debido a razones prácticas, entre otras cosas).
Ya he portado bash (1.08) y gcc (1.40) y parece que las cosas funcionan. Esto implica que
tendré algo usable en unos meses, y quiero saber que es lo que a la gente le gustaría que
tenga. Cualquier sugerencia es bienvenida, aunque no prometo implementarla :-)
```

Esta es una traducción al español de los mensajes originales que Linus mandó al grupo de noticias de minix de Usenet en el 91 y en este correo Linus evidencia su intención de hacer este sistema operativo algo simple y sencillo, basado únicamente en la plataforma i386. Originalmente, Linux porta el código el compilador de GNU (gcc 1.40) y el Bourne-Again Shell (bash 1.08), con éxito. Aún para esta época el kernel se limitaba a sólo soportar discos ATA.

El proyecto interesó a muchas personas, que quisieron colaborar en él desarrollando drivers, programas, etc. Gracias al trabajo en equipo la versión 1.0 de Linux fue liberada hasta el 14 de marzo de 1994. A partir de esta primera versión estable de Linux se le comenzó agregar más aplicaciones y utilidades con la finalidad de hacerlo más útil, en ese momento el proyecto GNU se adhiere al desarrollo de Linux, con el objetivo de hacerlo un sistema operativo libre.

Con todo lo anterior, Linux comenzó a crecer a una gran velocidad, ganando el gusto de muchos desarrolladores y usuarios en todo el mundo. Con la existencia de un kernel estable, comenzaron a desarrollarse distintas distribuciones, pero siempre respetando la esencia del kernel. De ésta forma sin importar la distribución siempre se podrá trabajar de la misma forma.

Poco a poco Linux comenzó a ganar terreno en el mercado de los servidores, y actualmente posee un buen porcentaje, sosteniendo servicios de gran importancia como correo electrónico, páginas web, bases de datos, entre otros. Razón por la cuál es un participante serio ante las opciones de código cerrado tales como Microsoft, Solaris, etc.

Las razones por la cuál se optó por Linux Slackware como plataforma de trabajo durante el diplomado fueron su estabilidad, la seguridad que ofrece, el

hecho de ser una de las distribuciones con más tiempo de existencia. Además Slackware no requiere de un sistema poderoso para ejecutarse, puede instalarse en computadoras que cuenten como sistema mínimo procesador 386 con 16 MB en RAM y 50 MB en disco duro.

Durante el desarrollo de éste módulo y con un sistema instalado previamente se nos enseñaron diversos aspectos básicos en cuanto a la administración de un sistema, las responsabilidades que se tienen, así como el perfil que debe tener un administrador. Dichas responsabilidades radican en el mantenimiento del equipo, configuración del mismo, instalación y actualización del software, administración de los recursos, creación de cuentas, monitoreo del sistema, detección de fallas, creación de copias de respaldo, establecimiento de políticas, entre otras.

Para poder llegar a estos puntos fue necesario aprender los comandos básicos para moverse en éste sistema operativo. Se aprendió el uso de comandos como ls, cp, mv, su, cat, more, less, find, file, cut, ping, mail, por mencionar algunos. Posteriormente se profundizó en el estudio de comandos más elaborados como grep, vi, y comandos especializados en la gestión de usuarios y grupos como adduser, deluser, chown, chgrp.

Como una segunda etapa, una vez que se dominaron estos comandos se procedió al aprendizaje de programación Shell, el objetivo es crear pequeños scripts que permitan la automatización de diversas tareas en el sistema. Estos scripts pueden ser tan complejos según la tarea que se quiera realizar. Los scripts usan los comandos existentes en el sistema, así como instrucciones propias como for, if, foreach, while do, read, que permiten al programador darle a los scripts más funcionalidad y control sobre lo que se pretenda hacer.

Los conocimientos adquiridos en este módulo fueron muy importantes para introducirnos al Sistema Operativo Linux, de tal manera que al término del curso se cuentan con las capacidades y los mecanismos necesarios para instalarlo, y administrarlo de forma básica, así como la posibilidad de generar scripts para realizar automatizar algunas sencillas tareas.

Módulo II. Administración de Linux

Habiendo estudiado las bases, durante este curso se aprendieron elementos más detallados del sistema operativo Linux. Uno de los puntos con los que se inició el curso fue el estudio más a fondo sobre las responsabilidades que tiene un administrador de un sistema Linux. El administrador es la persona responsable de configurar, mantener y actualizar el sistema, así como cuidar el funcionamiento del software, hardware y periféricos de forma que estén disponibles para los usuarios.

Al ser la administración una cuestión muy importante y delicada, es muy recomendable que el trabajo se divida en varios administradores, según el tamaño del sistema. El administrador tiene además la responsabilidad de dar mantenimiento a las cuenta de los usuarios, es decir, es el encargado de darlas de alta o de baja, tiene la tarea de capacitarlos y asesorarlos e inculcarles normas y conductas de trabajo en el sistema. El administrador tendrá que crear políticas de seguridad y trabajo para lograr un ambiente seguro, eficiente y confiable.

Asimismo un administrador debe contar con ciertos conocimientos básicos para desempeñar de mejor manera esta tarea, debe tener conocimientos sobre técnicas de programación, dominio de algún lenguaje de programación, conocimiento del funcionamiento del sistema operativo, así como conocimientos sobre hardware y mantenimiento de dispositivos.

Se estudiaron cuestiones como la numeración del Kernel, se hizo énfasis en éste tema, ya que, como administradores de un sistema, la actualización en algún momento será necesaria, y necesitará evaluar que versión del kernel será la más adecuada sin perjudicar el rendimiento.

Antes de realizar la instalación, se estudiaron brevemente las ventajas que ofrece Slackware, su sistema de paquetes, el tipo de instalación según los servicios que se vayan a ofrecer, así como algunos aspectos a cuidar durante el proceso de instalación, tales como el tamaño de las particiones, así como algunas características de los sistemas de archivos con el que se tendrán que formatear.

El proceso de instalación de Slackware presenta muchas ventajas, ya que trabaja en entorno texto, evitando consumir recursos innecesarios, además cuenta con un software de instalación intuitivo y sencillo. Durante la instalación es muy importante definir las particiones que tendrá nuestro disco duro para alojar el sistema de archivos de Linux, es muy importante analizar la función que desempeñará Slackware, para definir el tamaño de las particiones. Una vez definidas las particiones se prosigue con la ejecución del programa de instalación.

No se sugiere instalar Linux en una sola partición. Es por ello, que en el programa de instalación será muy importante definir la distribución del sistema de archivos de Linux en las particiones recién creadas, con esto se busca mayor protección ante la falla de alguna de las particiones, además permite una mejor gestión de las copias de seguridad, mayor rapidez, etc. Durante este proceso se les dará formato a cada una de ellas, así como el sistema de archivos a utilizar (ext3 y reiserfs).

Una vez definidas las particiones, se prosigue a la elección de los paquetes a instalar. Hay varias opciones según las necesidades del usuario, desde instalación completa, personalizada y experta. Una vez copiados los archivos en el disco duro, se creará un disco de arranque para usarse en caso de algún problema.

Posteriormente habrá que realizar diferentes configuraciones, tales como el gestor de arranque LILO (Linux Loader), el Mouse, configuraciones de la red, los servicios que queremos que se ejecuten al iniciar el sistema (apache, cups, inetd, sambas, etc), el escritorio de trabajo (KDE, Gnome, xfce etc.), y algo muy importante, la definición del password de root. Para éste último punto se sugiere el uso de passwords fuertes, es decir, combinar mayúsculas y minúsculas caracteres como \$, %, &, etc. Al finalizar será necesario reiniciar la computadora.

Habiendo completado estos pasos contamos con una copia de Linux Slackware instalada en nuestra computadora.

```
Command (m for help): n
Command action
  l logical (5 or over)
  p primary partition (1-4)
1
First cylinder (2542-4865, default 2542):
Using default value 2542
Last cylinder or +size or +sizeM or +sizeK (2542-4865, default 4865):
Using default value 4865

Command (m for help): p

Disk /dev/hda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1  *           1         1044     8385898+   7   HPFS/NTFS
/dev/hda2                1045        1169     1004062+   83   Linux
/dev/hda3                1170        4865    29683120   5   Extended
/dev/hda5                1170        1294     1004032    82   Linux swap
/dev/hda6                1295        2291     8008371    83   Linux
/dev/hda7                2292        2541     2008093+   83   Linux
/dev/hda8                2542        4865    18667498+   83   Linux

Command (m for help):
```

Fig. 1 Creación de particiones con fdisk



Fig. 2 Menú de instalación de Slackware

```

DONE ADDING LINUX PARTITIONS TO /etc/fstab

Adding this information to your /etc/fstab:

/dev/sda1  /          ext2    defaults    1  1
/dev/sda6  /usr       ext2    defaults    1  2
/dev/sda7  /opt       ext2    defaults    1  2
/dev/sda8  /home     ext2    defaults    1  2

(1882)

```

Fig. 3 Asignación de particiones



Fig. 4 Tipo de Instalación

Una vez que Slackware fue instalado, se aprendió la estructura de archivos de un sistema Linux, se estudió a detalle la función de cada directorio, ubicando donde se guardan las configuraciones, donde residen los dispositivos, donde se instalan los programas para los usuarios, donde se ubican los programas para el superusuario, cuál es el directorio para los usuarios del sistema.

A continuación, se comenzó a estudiar la administración de usuarios, se practicaron diversas formas de crear y eliminar usuarios en el sistema. Primero se usaron los comandos estándares de Linux, después se crearon y borraron cuentas manualmente, modificando los archivos `/etc/passwd` y `/etc/groups` y generando sus directorios correspondientes en `/home`.

Aunado a la creación de permisos también se estudió la asignación de permisos dentro del sistema, con el fin de limitar el acceso a ciertas áreas sensibles del sistema, se busca proteger la información y configuraciones de la computadora contra daños accidentales o intencionales. Todas las configuraciones en el hardware o software sólo pueden ser modificadas por el superusuario o usuario root o por aquellos usuarios que root autorice.

Se trabajaron con más detalles los principales archivos de configuración del sistema, como `/etc/inittab`, archivo donde se configuran los modos de acceso al sistema, se trabajó con `/etc/rc.d/rc.local` para la ejecución de programas durante el arranque de Linux, así mismo se configuraron dispositivos de entrada como el Mouse y el teclado de forma manual trabajando sobre el archivo `/etc/X11/xorg.conf`. Para la identificación del hardware instalado, se ocupó el comando `lsmod` que muestra una lista del mismo, también se usó el comando `lspci` y `lsusb` para los dispositivos pci y usb respectivamente. Aunque no fue posible practicarlo durante la clase, se estudió la forma de agregar nuevo hardware al sistema, y se vio el procedimiento a seguir para instalar los módulos (drivers en el ambiente Windows) en el caso que el dispositivo no fuera reconocido.

Se hicieron diversas prácticas relacionadas al entorno de redes, donde se configuró la tarjeta de red para habilitar la computadora en una red local. Se utilizaron las herramientas de configuración de red que vienen con Slackware, a sí mismo se configuró el entorno de red de forma manual utilizando los comandos `ifconfig`, `route`, y trabajando sobre los archivos `/etc/hosts`, `/etc/resolv.conf`. Además se trabajó con la asignación de IP's virtuales sobre el mismo dispositivo de red usando el comando `ifconfig`.

A pesar que durante la instalación de Slackware se estudió la configuración de LILO, se retomó su estudio pero sin la utilización del asistente que viene en esta distribución. Para la modificación de LILO se utilizó el editor de textos vi y se modificó el archivo `/etc/lilo.conf`. Además con LILO, se aprendió a proteger

mediante una contraseña aquellos sistemas operativos que no se desean estén disponibles para cualquier persona con acceso al equipo de cómputo.

Durante éste curso se aprendió a montar distintas particiones, discos duros u otros dispositivos de lectura o almacenamiento que no fueron agregados desde la instalación. Para realizar esto fue necesario trabajar con los archivos `/etc/fstab` y `/etc/mstab`, haciendo los ajustes necesarios es posible agregar al sistema particiones y discos duros con sistemas de archivos FAT, NTFS o unidades de CD-ROM, DVD-ROM, quemadores y hasta memorias flash-usb. Gracias a esto se podrán realizar copias de respaldo a otro medio o para incrementar la capacidad del sistema en caso de que se agote el espacio en el disco duro, mediante la conexión de otro dispositivo.

Para mejorar la eficiencia del sistema y automatizar algunas tareas que por el tiempo y la hora no es posible realizar, se estudió el comando `cron`. Éste comando es sencillo y muy poderoso, por que mediante la modificación del archivo `/etc/crontab`, es posible definirle una serie de actividades a realizar a una hora específica y con una frecuencia programada. De ésta forma se podrán programar tareas a una hora de poco uso (generalmente las madrugadas), éstas tareas pueden ser creación de respaldos del sistema, limpieza de archivos innecesarios o la realización de ciertos monitoreos.

Éste curso fue muy importante ya que se aprendieron las bases para configurar nuestro sistema y adaptarlo según se necesite. Gracias a que se practicaron las configuraciones de forma manual, es posible llevarlas a cabo en cualquier distribución de Linux. Con los conocimientos adquiridos en éste módulo es posible administrar un sistema Linux de forma básica.

Módulo III. Editores para la creación de Páginas Web

Éste módulo estuvo dedicado al aprendizaje del lenguaje HTML (Hypertext MarkupLanguage) para la creación de páginas Web. Al iniciar se estudió un poco sobre la historia de éste lenguaje. Primero en 1986 se publicó la norma ISO 8879 que presenta el Standard General Markup Language, el cual da origen al HTML. En 1991, el lenguaje HTML fue desarrollado por Tim Bernes cuando laboraba en el CERN(Centro Europeo de Investigaciones Nucleares), con el propósito de publicar de forma clara y al alcance de todos los físicos los resultados de las investigaciones que se llevan a cabo en ese centro de investigación.

Posteriormente se fueron haciendo mejoras al lenguaje y se crearon los primeros navegadores basados en línea de texto. A dos años de la creación del HTML surge el primer navegador gráfico disponible para computadoras personales conocido como Mosaic desarrollado por Centro Nacional para Aplicaciones de Supercomputadoras. Un año antes se había lanzado el primer visor Web llamado Viola, pero sólo disponible para sistemas UNIX y ambiente gráfico X11. El lanzamiento de Mosaic significó el acercamiento del Internet a la mayoría de la población.

Poco a poco al lenguaje se le fueron agregando más características que lo hicieron más poderoso. Así mismo el desarrollo de más navegadores Web se hizo presente, después de Mosaic, uno de los navegadores más poderosos y populares fue Netscape, posteriormente Microsoft sacó su propio navegador. Actualmente, el Internet Explorer de Microsoft se ha consolidado como uno de los navegadores más populares en el mercado. Han salido a la luz otros navegadores más compactos y seguros, estos son Mozilla, Mozilla Firefox, Opera, estos navegadores además tienen versiones para distintas plataformas como Windows, Linux, MacOS, Unix.

Analizando la tendencia actual del crecimiento de Internet, es comprensible que la mayoría de los sistemas que se realicen tendrán un mayor impacto si se desarrollan en entorno Web, ahí la importancia de aprender el lenguaje HTML. Las ventajas de crear un sistema que trabaje en entorno Web, es que se puede acceder a través de cualquier navegador de Internet, sin necesidad de instalar algún software cliente, siendo además de plataforma independiente, es decir, no importa el sistema operativo desde el cuál se acceda.

El lenguaje HTML es un lenguaje conformado por etiquetas o marcas, las cuales indican al navegador el inicio o fin de algo que se desea desplegar, como puede ser texto, una imagen, un sonido, una tabla, alguna liga. Algunas etiquetas tienen la capacidad de alterar el contenido de lo que se desea desplegar, ya sea cambiando el tipo de letra, el tamaño, el color, modificando el ancho o alto de una

imagen, entre otras características. Es muy importante que abrir y cerrar dichas etiquetas, para que el navegador pueda interpretar correctamente el contenido. Las etiquetas no son sensitivas, de manera que pueden escribirse en mayúsculas o minúsculas sin distinción.

Lo primero que se estudió fueron las etiquetas básicas para la creación de la estructura de una página de Internet. `<html></html>` le indican al navegador que el contenido del documento es Hipertexto; `<head></head>` en esta sección se escribe el título del documento, además de contener otro tipo de información necesaria para los navegadores; `<body></body>` es el cuerpo de la página Web, es toda la información que el usuario ve desplegado en pantalla.

Posteriormente se trabajó con etiquetas para encabezados, que permite dar formato a los títulos en el documento. Se trabajó con el formato de texto, al generar párrafos, cambiando su alineación, modificando el color, tamaño y tipo de letra. Modificar las características del texto, permite resaltar información que se considere importante para el visitante de la página, para ello, se cuentan con etiquetas que permiten realzar un texto mediante el efecto de texto cursivo, en negritas, y subrayado. Es posible además alterar el color de fondo de la página, buscando un equilibrio con toda la información desplegada.

En HTML existen etiquetas especializadas en la creación de listas, con el fin de enumerar una serie de elementos. Estas etiquetas son muy útiles para la publicación de temarios, de glosarios, etc. Cuentan con diversos modos de numeración como letras, números, números romanos, y otros elementos llamados viñetas (útiles cuando los elementos listados no tienen algún orden) como son puntos, guiones y cuadrados.

Uno de los puntos fuertes del hipertexto, es que no es un texto común, cuenta con diversos elementos que lo hacen más interesante. Una liga o hipervínculo es un apuntador a algún otro tipo de dato almacenado en la misma página, en otra página diferente, en otro sitio Web, puede ser también alguna imagen, sonido o video.

Con HTML se pueden integrar imágenes al documento, es posible alterarlas en su tamaño, y en su ubicación dentro de la página. También una imagen puede ser usada como un hipervínculo, incluso que despliegue algún texto informativo con solo pasar el mouse por encima de la imagen. Igualmente una imagen puede ser usada como decoración de fondo de la página.

Uno de los temas más interesantes y que se estudió a detalle fue la creación de tablas en HTML. Esta funcionalidad se agregó en la versión 3.0 de éste lenguaje en el año de 1995. Las tablas son un elemento poderoso ya que permiten el despliegue de información de una forma más clara. Es factible cambiar la tabla o algunas de sus celdas modificando el color, el tamaño, su alineación, el

tipo de borde. Además las tablas se convirtieron en una importante herramienta para el diseño de páginas Web. Una página completa puede ser diseñada tomando como base una tabla enorme en la que dentro de cada celda se podrá mostrar distinta información según su contenido, incluyendo imágenes, vínculos, u otros elementos. Muchos portales en el Internet trabajan de ésta forma.

Junto con las tablas, los formularios son un elemento imprescindible en el desarrollo de un sitio Web. Esta funcionalidad fue introducida en la versión 2.0 en el año de 1994. Gracias a ellos se hizo posible el enviar información a través del Internet. Para la creación de formularios se cuenta con muchos elementos para ingresar información, entre ellos se encuentran los campos de texto, *input radio* que permite escoger una entre varias opciones, *checkbox* que permite escoger varias opciones, los menús desplegables, *textarea* que es un área de texto más amplia para ingresar información de forma más abierta. Los formularios se utilizan en formas de registros, foros, encuestas, entre muchas funciones más.

Otro elemento con el que se trabajó fue el uso de *frames*, estas etiquetas permiten dividir la pantalla en diferentes zonas, las cuales son independientes entre sí, el uso de éste procedimiento genera páginas con una mayor presentación, permitiendo colocar distintos contenidos en cada zona. Los frames se componen de una página principal, la cual llamará a otras páginas. Además a cada frame se le puede definir el porcentaje que ocupará en la pantalla.

Este curso fue muy importante porque son las bases para el diseño de una página Web, el buen aprendizaje del lenguaje HTML permite que proyectos más grandes puedan ser desarrollados. Al final del curso, se adquirieron los conocimientos para armar un página de Internet junto con formularios para recabar información según el estilo de página que se esté creando.

Módulo IV. Administración de Servidores WWW con Linux

Para la publicación de un sitio Web en Internet, es necesario contar con un servidor que nos provea de ese servicio. El servidor Web es un programa especializado que se encarga de ofrecer comunicación mediante el protocolo HTTP. HTTP es la abreviación de Hypertext Transfer Protocol. A través de éste protocolo los navegadores de Internet intercambian información con los servidores de páginas, para poder obtener la información y desplegarla en las pantallas.

Éste protocolo, de manera muy general trabaja de la siguiente forma: El cliente inicia una conexión con el servidor, si es aceptada el servidor regresa una señal ACK (acknowledgment) que indica que una sesión ha sido abierta, el cliente por su parte manda una señal GET solicitando información, finalmente el servidor manda una señal "response message" donde se envía el recurso solicitado y se cierra la sesión. Para ver de forma práctica este proceso se hicieron prácticas con el comando telnet hacia algunos puertos para observar la forma en la que los servidores abren y cierran una sesión.

Existen muchos servidores Web en el mercado, de distintas compañías y para diversas plataformas. Algunos de estos programas son de paga, pero también existen versiones OpenSource. Algunos de los servidores más difundidos son: Apache, AOLServer, Roxen Challenger, StrongHold, Personal Web Server, ISS (Internet Information Server), Tomcat – Yakarta. Cada uno tiene sus ventajas, sus fortalezas y dependerá de las necesidades del administrador escoger la mejor opción.

El motivo por el cuál se optó por el uso de Apache como servidor de páginas Web fue, aparte de ser un producto Open source y sin costo, por la estabilidad que ofrece, su robustez, la facilidad de configuración para diferentes entornos de trabajo, ofrece un alto nivel de seguridad, está disponible para muchas plataformas (Linux, Unix, MacOs, Solaris, Windows), soporta distintos módulos de lenguajes de programación (php, perl), tiene soporte para SSL, entre otras características. Desde su introducción al mercado Apache ha ganado terreno en el mercado de servidores Web, actualmente es el más usado según NetCraft.

Para la instalación de Apache se eligió hacerlo por el método de compilación, con el fin de poder personalizarlo según se necesite, de esta forma fue necesario bajar el código fuente de www.apache.org, definir algunas configuraciones (./configure –opciones) e iniciar su compilación (make) y posteriormente su instalación (make install).

Una vez que se ha instalado el servidor Apache, lo siguiente es configurarlo. Apache cuenta con más de 200 directivas que le agregan distintas

funcionalidades. Algunas de éstas directivas estarán disponibles de acuerdo a los módulos con los que Apache fue compilado. Las configuraciones de Apache son clasificadas en tres grupos: Global Environment, Main Server y Virtual Servers.

En Global Environment, se administrarán las directivas generales de operación para Apache. En esta sección se definirán cuestiones como el usuario y grupo por default con el cual Apache funcionará, se especifica el puerto por donde se accederá al servidor, la ruta por default donde estarán colocados los archivos html que se publicarán, se precisa el lugar donde están ubicados los directorios de configuraciones y bitácoras.

Una funcionalidad que tiene Apache es la capacidad de adaptarse a un número de consultas mayor según se necesite. Para probar ésta funcionalidad, se escribió un pequeño script usando el comando *ab*. Éste comando permite hacer pruebas de rendimiento sobre algún sitio Web. Con éste script se pudo observar la forma en la que se comporta Apache según se incrementen o disminuyan las consultas realizadas.

Cuando el Servidor Apache es levantado, a su vez levanta *procesos hijos* los cuales se encargarán de atender las solicitudes de páginas. Cuando el número de solicitudes aumenta, Apache incrementa el número de procesos hijo para darles salida. Esta funcionalidad se especifica con las directivas *MinSpareServers* (Procesos hijos mínimos en espera), *MaxSpareServers* (Procesos hijos máximos en espera), *StartServers* (Procesos que se iniciaran al arranque de Apache), *MaxClients* (Número máximo de procesos que podrán ser levantados como máximo.)

Este servidor Web cuenta también con un código de errores al presentarse algún problema cuando se pretende solicitar algún recurso, es posible personalizar la presentación de estos mensajes. Apache permite la personalización de las bitácoras de error que genera, desde el nombre del archivo, hasta el formato en la que se quiere que se almacene en la bitácora.

Se practicó además con la directiva `<Directory></Directory>`, con la que se pueden definir propiedades específicas sobre un directorio donde existen documentos a publicar. Aplicando bien algunas opciones en esta directiva es posible agregar un poco de seguridad sobre nuestro sitio Web.

Uno de los puntos más poderosos de Apache es su capacidad de generar Servidores Virtuales `<VirtualHost></VirtualHost>`. Los servidores virtuales permiten que en un mismo servidor, Apache pueda responder a distintas solicitudes, de ésta manera es posible mantener varios sitios Web con diferentes nombres y direcciones IP. Dentro de ésta directiva se agregan distintas opciones, como la ubicación de los documentos de cada sitio Web, el nombre del servidor, la dirección IP, entre otros.

Para complementar la seguridad de nuestro sitio Web, Apache nos brinda la funcionalidad de Control de Acceso. De ésta manera podemos definir las direcciones IP's y los usuarios que tendrán acceso sobre los recursos del servidor Web. Esta funcionalidad puede agregarse dentro de la Directiva <Directory>.

Apache cuenta con la capacidad de trabajar con módulos, los módulos están encargados de agregar funcionalidad adicional incrementando las directivas disponibles. Existen dos tipos de módulos: los módulos estáticos, que se agregan al compilarse apache; los módulos dinámicos, que se agregan sin necesidad de recompilar apache, para ello es necesario compilarlo con capacidad para soportar módulos.

Entre los temas tratados durante el módulo se hizo énfasis en la importancia de que exista un administrador especializado en el Servidor Web, dependiendo por supuesto del tamaño del mismo. Se habló de la necesidad de crear un sistema de respaldo de nuestro sitio ante cualquier eventualidad, ya que quedar fuera de línea equivale a pérdidas económicas importantes, además de la pérdida de confianza de los clientes. Se vieron de forma muy general algunos sistemas de respaldo que hay en el mercado, generalmente enfocados al ámbito empresarial.

Al final del módulo se estudió un poco de programación CGI (Common Gateway Interface) como una introducción a la creación de sitios Web dinámicos. Los CGI's pueden ser programados en distintos lenguajes de programación. Son programas que se ejecutan en el servidor con parámetros que son dados desde una máquina cliente a través de un navegador de Internet. Permiten dar dinamismo a los sitios Web, así como guardar la información en una base de datos. Para habilitar la ejecución de éstos scripts es necesario trabajar con algunas directivas de Apache, así como indicar la ubicación de los mismos.

Para la evaluación de este curso fue necesario instalar desde cero un servidor Apache. Para ello fue necesario bajar el código fuente de la página oficial, posteriormente hubo que compilarlo e instalarlo en base a unas configuraciones dadas, y finalmente se trabajó sobre el archivo de configuración de Apache para habilitarle algunas funciones, entre ellas levantar Hosts Virtuales y que tuviera la capacidad de ejecutar scripts CGI's.

Este módulo fue uno de los más importantes dentro del diplomado, ya que para que un sitio Web funcione perfectamente, se necesita de un buen servidor que garantice la estabilidad y la permanencia en línea del sitio. Con los conocimientos adquiridos se adquirió la capacidad de instalar un servidor, entender su funcionamiento, y de optimizarlo según se requiera.

Módulo V. Programación con PHP

Actualmente debido a la gran cantidad de información que hay en el Internet los sitios Web deben que tener la capacidad de cambiar para mostrar toda la información que se va generando. Esto no se puede lograr con el HTML clásico, se necesita de un lenguaje que permita construir páginas dinámicas con la información que se requiere.

Esto se puede lograr con varios lenguajes de programación como son: php, asp, jsp, perl, entre otros. De estos lenguajes el que ha presentado un avance importante ha sido PHP. Un gran porcentaje de sitios están desarrollados en este lenguaje, muchos CMS (Sistemas de Administración de Contenido) también fueron escritos en PHP.

Los motivos por el cual PHP fue seleccionado como lenguaje base para el desarrollo de sitios Web fue su sencillez, debido a que presenta muchas semejanzas en su sintaxis con respecto a otros lenguajes como C, Perl y Java. PHP se ejecuta sobre el servidor, de esta forma las páginas son armadas antes de ser enviadas al navegador que la solicitó. Tiene una gran cantidad de funciones que permiten conectarse de forma sencilla a distintos tipos de base de datos.

PHP fue desarrollado originalmente en 1994 por Rasmus Lerdorf como un simple conjunto de scripts escritos en Perl para llevar un control de quien visitaba su página donde tenía alojado su currículum. Más adelante decidió agregarle más funcionalidad escribiéndolo en C, lo que le permitió conectarse a base de datos y desarrollar pequeñas aplicaciones de Web dinámicas. Estas primeras versiones no eran públicas, pero Rasmus decidió liberar el código para que pudiera ser mejorado por la comunidad. El nombre inicial de este trabajo era PHP/FI (Personal Home Page Tools / Form Interpreter).

Una vez liberado se le agregaron más características, desarrollándose principalmente con lenguaje C, la versión 2.0 de PHP/FI fue liberada en Noviembre de 1997. El número de sitios en ese tiempo que utilizaban este lenguaje llegaba a casi 50,000 sitios. Pero poco tiempo después la versión 2.0 fue sustituida por la versión beta de PHP3.

Durante ese año se comenzó el desarrollo de PHP3, fue reescrito en su totalidad por Andi Gutmans y Zeev Zuraski. Ellos junto con Rasmus Lerdorf trabajaron en conjunto y anunciaron a PHP3 como el sucesor oficial de PHP/FI 2.0. Esta versión ofrecía una sólida infraestructura para distintas bases de datos, protocolos y API's. Una de las características introducidas fue el soporte de sintaxis orientado a objetos. Esta versión fue liberada bajo el nombre de PHP con el nuevo significado de "PHP: Hypertext Preprocessor" en Junio de 1998.

A partir del invierno de 1998, se comenzó a trabajar en la mejora de la versión 3. Se generó el motor Zend (Compuesto por los apellidos de los autores, Zeev y Andi), lo que permitió mejorar la ejecución de aplicaciones complejas, mejorar la modularidad del código de PHP, además de nuevas características adicionales, entre las que destacan sesiones HTTP, buffers de salida, formas más seguras para el control de las entradas de usuarios. Esta versión fue liberada en Mayo del 2000, y hasta la fecha se calcula que un 20% de dominios de Internet están utilizando PHP. En la versión 5 se plantean mejoras sustanciales al lenguaje así como la inclusión de la versión 2.0 del motor Zend.

Al iniciar el estudio de PHP, se estudiaron algunas semejanzas y diferencias entre PHP y ASP. Son lenguajes ligeramente parecidos en potencia y sintaxis; PHP está más orientado hacia sistemas Unix, Linux, pero se han desarrollado versiones para Windows, por su parte ASP, sólo puede encontrarse para ambientes Windows. Algunas funciones especializadas para ASP, deben ser compradas a empresas, en cambio, en PHP las nuevas funciones que se incorporan al lenguaje es gracias al trabajo desinteresado de otros programadores alrededor del mundo.

Para programar en PHP, no hace falta nada especial, basta con cualquier editor de texto plano, ejemplos de éstos puede ser el Bloc de Notas (Windows), kwrite (editor de KDE), vi o pico si se trabaja en entorno texto. En el mercado también existen herramientas más profesionales que tienen soporte para PHP y otros lenguajes de programación, que incluso permiten ir desplegando la página para mostrar como va a quedar, usando la tecnología WYSIWYG (What You See Is What You Get).

PHP, es un lenguaje que puede ir embebido dentro del código HTML, pero a su vez, dentro del código PHP puede imprimirse código HTML. Para separar dichos códigos hay que hacer uso de las siguientes etiquetas: `<? ?>`. Todo lo que esté dentro de estas marcas será interpretado como código PHP, el resto será interpretado como código HTML. Otro punto importante en la programación de estos scripts, es que al final de cada instrucción habrá que escribir un `;`, característica heredada del lenguaje C.

Como en todo lenguaje de programación, se estudiaron el uso de las variables, los distintos tipos que soporta (int, double, decimal, octal, hexadecimal, char, string), pero con la diferencia que al hacer referencia a una variable en PHP habrá que hacerlo anteponiendo el signo `$` antes del nombre de la variable. No hay necesidad de declarar las variables al inicio del programa, pero si se sugiere llevar un control personal de las mismas, las variables se autoconfiguran según el valor que se les asigne. Otra ventaja que ofrece es que permite realizar operaciones entre distintos tipos de datos. Las variables en PHP son "case sensitive", por lo que hay que tener cuidado al escribirlas.

PHP, soporta operaciones aritméticas como suma, resta, multiplicación, división y módulo, así mismo tiene la capacidad de realizar comparaciones como igual, mayor que, menor que, etc. Permite operadores de comparación lógicos como AND, OR y !. Además permite el incremento o decremento de una unidad de valor de una variable, ejemplo: ++\$variable, \$variable++, --\$variable, \$variable--.

Al igual que otros lenguajes de programación, PHP soporta la repetición de acciones sucesivas o la elección de alguna opción según si se presentan ciertas condiciones definidas por el programador. Para la definición de estos bloques de decisión contamos con la instrucción if, que dada cierta condición se ejecuta cierto bloque de instrucciones, de no cumplirse dicha condición se activa otro bloque de instrucciones encabezados por la instrucción else.

Para el trabajo de ciclos se cuentan con varias instrucciones. La instrucción while ejecutará un bloque de instrucciones hasta que cierta condición deje de cumplirse. La instrucción do / while, primero ejecutará una vez las instrucciones y dependiendo si se cumple la condición continuará con el ciclo. For es una estructura de control, donde se define una condición de inicio, una condición de final y el incremento de una variable, se ejecutará un bloque de instrucciones durante la duración de ese ciclo. La instrucción foreach recorrerá los valores de un arreglo.

Una vez asimilado estos temas, se estudió la creación de funciones, con finalidad de automatizar ciertas tareas dados algunos valores de entrada. Es posible crear una librería que contengan todas las funciones que se van creando con el fin de sistematizar el trabajo al momento de desarrollar aplicaciones. Se sugiere que conforme se vaya creando una librería se clasifiquen según su función, por ejemplo, librería con funciones para conexión a base de datos.

Uno de los elementos más poderosos que posee PHP es el manejo de arrays o arreglos, que permite desde arreglos unidimensionales hasta arreglos multidimensionales, además hay un gran número de funciones destinadas al manejo de arreglos. Permite además el uso de arreglos asociativos, es decir, en vez de utilizar números, se usan palabras.

Otra de las fortalezas de PHP es su excelente manejo de cadenas. Para asignar una cadena a una variable solo hay que escribirlo entre comillas. La concatenación de varios valores se hace a través del uso de un punto. PHP así mismo, posee muchas funciones dedicadas al tratamiento de cadenas.

Además PHP ofrece soporte para el uso expresiones regulares, esto permite crear aplicaciones más poderosas, ya sea en el momento de validar entradas de usuario, al realizar búsqueda de cadenas en alguna base de datos, o para validar la estructura de alguna cadena de texto si es necesario.

PHP también permite el manejo de archivos de texto y binarios, con el fin de almacenar o extraer información de ellos. Además PHP puede leer archivos que no necesariamente están en la máquina local, éstos pueden ser otras páginas de Internet, u otro archivo que esté compartido a través de la red. Hay otras instrucciones que permiten renombrar los archivos, obtener un duplicado o borrarlos del sistema.

Se estudió el uso de las variables de ambiente, ya que en el desarrollo de formularios es importante enviar la información a otra página para que sean procesados, y mucha de ésta información viaja a través de las variables de entorno del sistema. Por cuestiones de seguridad se aconseja el método de envío POST, para evitar que el usuario vea la información que se envía, así como el nombre de las variables. Para ello es muy importante el uso de las variables de entorno y el uso de instrucciones como isset para validar la información que se recibe.

Se trabajó además con algunos pequeños programas para el aprendizaje sobre el uso de las cookies, que permiten guardar distintas informaciones acerca del usuario que visita un sitio Web. Se puede guardar información tal como las veces que el usuario ha entrado al sitio, el tipo de navegador que utiliza, o algunas configuraciones que el usuario ha ajustado para personalizar la página. Se sugirió ser cuidadosos con el uso de cookies, debido a que por motivos de seguridad, algunos navegadores rechazan la creación de cookies por lo que, si nuestro sistema depende altamente de ellas, podría no funcionar adecuadamente.

Otro aspecto interesante que se estudió fue el de las sesiones, la sesión inicia en el momento en que el usuario entra al sistema, recorre el sitio y termina en el momento que lo abandona. El uso de sesiones es muy útil, debido a que permite el paso de valores de una página a otra, esta característica es muy utilizada en los sistemas de compras en línea donde se necesita almacenar la lista de productos que se van comprando mientras se está en la página. PHP también ofrece diversas funciones que permite el envío de correos electrónicos desde el sitio Web. Esta característica se logra a través de la instrucción mail.

El módulo cerró con el estudio de programación orientada a objetos en PHP. Con este tipo de programación se busca crear aplicaciones mejor estructuradas, y con mejor capacidad de actualización si ésta fuera necesaria. Ya que es más fácil optimizar y mejorar el código de una clase, sin necesidad de modificar todo el sistema.

Módulo VI. Interacción de WWW con Base de Datos

Todo sitio de Internet dinámico requiere forzosamente contar con un sistema de almacenamiento de datos que de forma fiable conserve la información y la tenga disponible en el momento que se requiera. Actualmente hay una gran gama de productos en el mercado como MS-SQL, Oracle, PostgreSQL, MySQL, DB2, por mencionar algunas, cada una diseñada para diferentes tareas y ambientes de trabajo.

Para fines de este diplomado se estudiaron dos manejadores de base de datos Open Source: MySQL y PostgreSQL. Ambos son proyectos libres y compatibles con PHP, cada uno tiene distintas capacidades y formas de administrarse.

Actualmente MySQL es un proyecto dirigido por la empresa MySQL AB, una empresa de reciente creación. En un principio utilizaban msql para conectarse a sus tablas usando sus propias rutinas de acceso a bajo nivel, pero llegaron a la conclusión de que msql no era lo suficientemente flexible para sus necesidades. Se decidió mejorarlo por su cuenta, y ahora es uno de los primeros en el ámbito de las bases de datos de código abierto.

Antes de trabajar con MySQL, primero se descargó el código fuente de su página oficial, posteriormente se descomprimió y se instaló siguiendo las instrucciones que acompañan a MySQL. Fue necesario crear al usuario mysql, el cual tiene permisos sobre las bases de datos, es decir, puede crear, borrar y modificarlas. Posteriormente se tiene que inicializar la base de datos y ejecutar el demonio encargado de poner en marcha el servicio de MySQL.

Una vez que la base de datos ha sido inicializada, hay que hacer algunas modificaciones a la base de datos del sistema "mysql". Para ello hay que entrar al entorno de MySQL, y en el prompt teclear "use mysql", para indicar que se trabajará con esta base de datos del sistema. A la tabla "users" habrá que agregar al usuario nobody, con el fin de que nuestras aplicaciones Web tengan acceso a la base de datos, hay que recordar que el usuario por default de Apache es "nobody". Si no se hace esta modificación las aplicaciones no funcionarían. Una vez hecha esta modificación habrá que reiniciar el demonio de MySQL.

Para el manejo de las bases de datos se trabajaron con diversas instrucciones, algunas de ellas son: create, crea una base de datos; use, cambia la base de datos con la que se trabaja; drop, elimina una base de datos. Estas instrucciones funcionan trabajando dentro de mysql. Existen a su vez otros comandos externos que permiten obtener información sobre la base de datos:

mysqlshow, lista las bases de datos en el sistema, mysql + nombre de BD + tabla, lista información de una tabla en especial.

Posteriormente se estudiaron las instrucciones para el manejo de la información en las tablas de nuestra base de datos: create table nombretabla: crea una tabla; insert into nombretabla: inserta un valor en la tabla; select información from nombretabla: hace una búsqueda en la tabla; update, actualiza información de una tabla; alter table nombretabla add/modify, altera las características de un campo dentro de la tabla. Con estas sencillas instrucciones es posible desde una simple aplicación Web consultar, agregar, modificar información a la base de datos.

Al igual que en los lenguajes de programación, es muy importante definir los tipo de variables que se pretenden utilizar en la base de datos, cada tipo de datos ocupa un tamaño definido, así que antes de crear la tabla se sugiere crear la estructura de la BD en papel con los campos y tipos de información que se almacenarán. Una mala elección podría hacer que ocupe más espacio en disco duro de lo esperado. MySQL soporta valores de tipo entero, char, varchar, date, flota, por mencionar algunos.

Para trabajar con una base datos primero hay que conectarse a ella, una forma es desde el propio entorno de trabajo de MySQL usando la siguiente sintaxis:

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

El otro método es a través de alguna aplicación Web, en el caso de que estuviera desarrollada en PHP, se usaría la siguiente sintaxis:

```
mysql_connect('localhost', 'mysql_user', 'mysql_password')
```

Es importante hacer notar que por motivos de seguridad se sugiere ponerle una contraseña a la base de datos.

Una vez establecida la conexión con la base de datos es posible realizar cualquier acción que se necesite, dependiendo obviamente de los permisos que tengamos sobre esa base de datos. Para un mejor entendimiento de los conceptos teóricos se trabajó con ejemplos ya escritos de una pequeña aplicación Web, donde existía la necesidad de insertar, borrar y modificar la información.

Dentro del curso un punto en el que se hizo énfasis cuando se trabaja con base de datos, es que la información es muy importante, por lo que es necesario tener un plan de respaldos de la base de datos. Es necesario hacer un plan que contemple la regularidad con la que se tienen que hacer dichos respaldo, el medio que se usará y la persona responsable de hacerlos.

Para la creación de los respaldos de una base de datos se utiliza la instrucción mysqldump:

```
mysqldump -opt basededatos > respaldo.sql
```

Si se necesita hacer un respaldo de todas las bases de datos entonces que usar lo siguiente:

```
mysqldump -all-databases >respaldo.sql
```

Y para restaurar el respaldo, primero hay que crear una base de datos vacía::

```
mysql > create database basededatos  
# mysql diplmado < respaldo.sql
```

Para hacer esto hay que contar con los permisos necesarios, por parte del administrador del servidor.

Gracias al estudio de MySQL fue posible generar aplicaciones cada vez más reales, ya que se ajusta al esquema de muchos sitios Web, donde es casi impensable crear sitios dinámicos sin la existencia de una base de datos que guarde la información que se va generando y que permita consultarla en cualquier momento.

Módulo VII. Introducción a la Seguridad en Cómputo

Actualmente la información se ha convertido en un bien invaluable para todas las empresas, no importando su ubicación o su forma física. La información puede presentarse en forma de documentos, video, audio, imágenes, o bases de datos. Hoy por hoy mucha información fluye a través del Internet para diversos fines como pueden ser transacciones bancarias, inscripciones escolares en línea, suscripciones para algunos servicios, etc.

Lamentablemente el robo de información para su uso en cuestiones ilícitas ha crecido de manera alarmante. Por obvias razones los sitios Web se han convertido en blanco para ser atacados y extraer algún tipo de información. Es por ello, que al desarrollar aplicaciones Web, hay que diseñarlas bajo algunas normas que incluyan esquemas de seguridad, el objetivo de este módulo fue el de aprender técnicas que se deben considerar para reforzar la seguridad no solo sobre el sitio, sino sobre el servidor donde estará funcionando.

Lo primero que se estudió fue la definición de seguridad: “Un sistema de cómputo es seguro si se puede confiar en que se comportará como se espera que lo haga, y que la información en él se mantendrá inalterada y accesible durante el tiempo que su dueño lo desee para los usuarios que él mismo decida”.

De esta definición salen otros conceptos importantes como lo son:

- Confidencialidad: Sólo el propietario es capaz de descifrar la información
- Autenticación: Se asegura la identidad de la persona
- Integridad: Se asegura que la información no ha cambiado durante la transmisión
- Autorización: La información sólo está disponible para determinados usuarios, con diferentes niveles para cada uno
- No repudio: Se asegura que el emisor de la información no puede negar haberla enviado.

Estos conceptos forman parte del estándar ISO 7498-2 que cubre las comunicaciones seguras entre sistemas abiertos. Pero aparte de estos servicios de seguridad, hay dos conceptos más que hay que considerar.

Disponibilidad: Asegurar que el sistema esté disponible cuando se le requiera.

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

Se analizó la importancia de identificar lo que es una amenaza para un sistema, así como la de identificar las vulnerabilidades que en determinado

momento un sistema puede presentar para evitar que algún programa malicioso o un visitante no deseado haga uso de ella y no se comprometa información vital. Al diseñar un sistema hay que considerar una lista de riesgos potenciales que el sistema pueda presentar, y hacerse una lista de lo que podría ocurrir si dichos riesgos llegaran a ocurrir.

Un ataque a un sistema puede llevarse a cabo de muchas formas, puede ser un ataque directo de parte del agresor al sistema, o tratarse de un ataque llevado en varios pasos y de manera sistemática, un ataque puede llevar como objetivo la destrucción, modificación o retraso en los servicios. Puede realizarse un ataque a través de diversas herramientas tales como virus, Caballo de Troya, Ingeniería Social, Bombas lógicas, etc. También se pueden dar ataques de forma indirecta, es decir, el atacante se apodera de otras máquinas mediante algún programa malicioso y estas quedan en su poder, posteriormente mediante una señal las máquinas esclavas realizan un ataque al servidor.

Para contar con un sistema de seguridad en el área de trabajo es necesario definir un conjunto de políticas y mecanismos que garanticen los conceptos importantes de un sistema seguro (confidencialidad, integridad, etc), teniendo en cuenta que para la organización el elemento más importante es la información.

Hay que diseñar políticas de seguridad donde se definan las características del sistema de seguridad que se implantará, identificando las amenazas contra las que hay que protegerse y los mecanismos de protección. De la misma forma hay que diseñar mecanismos de seguridad, que serán en primera instancia la herramienta básica para asegurar la integridad del sistema. Dichos mecanismos de seguridad se dividen en tres pasos: Prevención, Detección y Recuperación.

En Prevención se utilizarán sistemas de software y hardware para la identificación de usuarios autorizados a ciertos recursos definidos por el administrador, también se diseñan controles administrativos, incluye además mecanismos como encriptación en las comunicaciones, utilización de contraseñas duras, métodos de autenticación mediante tarjetas electrónicos, mecanismos biométricos, etc. La Detección estará enfocada a la búsqueda de anomalías en el sistema, y evitar en lo posible el daño en la información, identificar a los posibles culpables, y encontrar huecos de seguridad en el sistema, esto se logra principalmente mediante analizadores de tráfico. Y la Recuperación, que entrará en marcha justo después de que un ataque fue realizado, donde se hará una restauración de la información, así como un análisis de las vulnerabilidades explotadas y su correspondiente corrección, además de un seguimiento a las bitácoras del sistema para examinar el comportamiento del ataque.

La organización debe contar con un plan de contingencia que le permita trabajar ante cualquier eventualidad, ya sea por un ataque o algún desastre natural. Este plan debe incluir mecanismos de respaldos, fortalezas y debilidades

del sistema. Con un plan de contingencia se evita la pérdida de clientes, pérdida de ingresos, pérdida de confianza, ente otras.

El Internet se ha convertido en una gran autopista de la información, donde diariamente ocurren miles de transacciones, e información muy importante viaja a través de esta red sin ninguna protección, quedando a merced de individuos nada éticos que lo único que buscan es un beneficio personal a través del fraude. Debido a la inseguridad que impera en la redes, se hizo necesario buscar un mecanismo para proteger la información, encontrando una solución en la criptografía.

Primero, la criptología, es la ciencia que estudia los aspectos y contenidos de información en condiciones de secrecía. Se compone del griego *criptos* – oculto y *logos* – tratado. Así, la criptología se divide en Criptografía y Criptoanálisis. La criptografía es un conjunto de técnicas que alteran la información sin alterar el contenido, convirtiéndola en un conjunto de símbolos sin sentido para aquellos que no posean los métodos para descifrarla, en pocas palabras es el arte de construir códigos secretos. Por su parte el criptoanálisis se encarga de recuperar información que ha sido tratada con alguna técnica criptográfica, sin conocer previamente el método utilizado.

El uso de la criptografía es tan antiguo como el deseo de ocultar la información a ojos no deseados. Su uso data desde tiempo de los griegos, el envío de mensajes se hacía a través de mensajeros que corrían grandes distancias para hacerlos llegar, en el camino se exponían a ser atrapados por el enemigo y por consecuencia el contenido del mensaje. Así que se ideó una técnica, sobre un trozo de madera con unas marcas bien definidas, se enrollaba una cinta de papel, y se escribía el mensaje, al desenrollarlo solo quedaban símbolos incomprensibles. Cuando el receptor recibía este papel, lo enrollaba usando como base las mismas marcas, y el mensaje quedaba expuesto. A esta técnica le siguieron muchas más como el Cifrado Cesar, Cifrado de Vigenére. También se desarrollaron máquinas criptográficas parecidas a una máquina de escribir, pero con capacidades para ocultar un mensaje.

Con la introducción de la computadora las técnicas de criptografía comenzaron a hacerse más complejas, y se diseñaron diversas técnicas aprovechando su alto poder de procesamiento. La mayoría de estas técnicas criptográficas trabajan con el uso de “llaves” que permiten cifrar y descifrar la información. Existen dos métodos de cifrados: Simétrico, la llave que cifra y descifra es la misma, por lo tanto emisor y receptor tienen que ponerse de acuerdo para el intercambio de la llave; Asimétrico, la llave que encripta es diferente a la de desencriptación, la primera es pública, y la segunda sólo la conoce el receptor. De esta forma han surgido diferentes técnicas de encriptación. Algunos ejemplos son: Diffie-Helman, DES (Data Encryption Standard), AES (Advanced Encryption Standard), RSA (Rivest, Shamir y Adleman apellidos de los autores).

Para mejorar la seguridad en el entorno informático se ha creado lo que se conoce como “huella digital” con el fin de establecer mecanismos de autenticación fiables sobre los mensajes u archivos que viajan sobre la red. Para ello se utilizan las funciones hash, las cuales mapean el mensaje o archivo, así como su tamaño, y dan como resultado un valor numérico que sirve como autenticador, cualquier modificación a un archivo por mínima que sea, dará un valor hash completamente diferente, y es imposible que dos mensajes obtengan el mismo valor hash. La función hash más conocida es MD5.

Al igual que existen herramientas para autenticar mensajes y archivos, también existen formas para autenticar sitios Web. Para ello se ha generado los certificados digitales, éstos certificados legitiman que una llave pública y su correspondiente llave privada pertenecen a un individuo u organización, probando ser quien dice ser. Estos certificados permiten abrir un canal seguro de comunicación entre un cliente y un servidor encriptando la información que entre ellos se intercambie. Los certificados digitales son expedidos por una Autoridad Certificadora (CA), ellas generan y administran los certificados y se cercioran de la identidad del individuo u organización así como su llave pública.

Para el entendimiento de este procedimiento se practicó con el software OpenSSL (Open Secure Socket Layer). SSL es una propuesta de estándar para encriptado y autenticación en el Web, es utilizado para encriptar transacciones en protocolos a nivel de aplicación como http, FTP, etc. Se basa en un esquema de llave pública para el intercambio de llaves de sesión. Esto dificulta al atacante el comprometer una sesión, además de que la información pasa incomprensible por la red.

Para su uso se necesitó descargar los archivos fuente de su página oficial (www.openssl.org), posteriormente se configuró, se compiló e instaló. Con openssl es posible cifrar mensajes con diversas técnicas, así mismo es posible obtener huellas digitales de archivos, así como generar llaves de distintas longitudes, además permite la generación de certificados digitales. Para ello es necesario primero generar una llave privada con el mecanismo que se desee, y posteriormente se genera un “Certificate Signing Request”(CSR) el cual es un conjunto de valores que se envía a una autoridad certificadora para que la firme y se reconozca como un servicio seguro, pero también con openssl es posible autocertificarse, y ser la propia autoridad certificadora.

Otro tema que se estudió fue el de la esteganografía, que tiene como fin ocultar o camuflar información usando distintas técnicas. Algunas formas para esconder información es usando imágenes sustituyendo algunos bits poco importantes por bits del documento a esconder. La ventaja de usar esteganografía es que puede pasar inadvertido, en cambio un mensaje encriptado es más fácil de ser detectado.

Con respecto a la administración de un sistema se estudiaron algunas recomendaciones para crear un entorno de trabajo seguro. Como administrador es importante inculcarle a los usuarios políticas de seguridad sobre sus cuentas de trabajo. Una de ellas es la creación de una contraseña segura, para ellos se recomienda no usar el login como contraseña, usar contraseñas largas, que no contenga información personal, que incluya mayúsculas y minúsculas u otros caracteres especiales y que sea fácil de recordar. Es importante indicar que la contraseña es intransferible, se debe cambiar de forma periódica y no escribirla.

El administrador tiene que realizar un monitoreo constante del sistema buscando anomalías en el sistema, analizar el tráfico de la red con el auxilio de diversas herramientas, mantener el sistema actualizado, entre otras tareas. Debe delimitar los permisos de acceso para cada usuario, definir a que documentos tienen derecho de ver, modificar o borrar. Para una mejor administración sobre éstos tópicos se sugiere el uso de Listas de Control de Acceso(ACL). Las listas de control de acceso permiten restringir ciertos archivos o directorios a una clase o un grupo en particular, dependiendo el conjunto de entradas definido en una política de seguridad.

En el monitoreo básico de un sistema hay que revisar el rendimiento del equipo, el tráfico de red, las bitácoras del sistema. Para el rendimiento puede usarse el comando top, y ps y verificar que ningún programa consuma de manera sospechosa los recursos. Para el entorno de red existe netstat que muestran las conexiones establecidas en el equipo, también existen analizadores de tráfico como ethereal o nessus que nos muestra de forma detallada los paquetes que circulan en la red, y así detectar conductas extrañas que pudieran comprometer la seguridad. Existen además otros programas que buscan vulnerabilidades en los sistemas como Nikto, Sara, Saint, y nos alerta por donde podría ser atacado el equipo.

Se estudió con un poco mas a detalle los diferentes ataques que existen. Estos pueden clasificarse en dos tipos: *Pasivos*, donde el atacante sólo observa las vulnerabilidades del sistema usando algún equipo de la misma red; *Activos*, el atacante una vez que analizó los huecos de seguridad realiza el ataque de forma directa modificando o extrayendo alguna información o comprometiendo la estabilidad de sistema con el auxilio de algunas herramientas como pueden ser virus, troyanos, etc.

Entre los diferentes tipos de ataques el más temido y el que causa más daño a un sistema es el ataque de Negación de Servicios o Denial of Service(DoS), cuyo principal objetivo es negar el servicio que el sistema esté prestando, ya sea saturando el ancho de banda, por un consumo excesivo de recursos del equipo, por paquetes de información mal formados o ataques a los

DNS y enrutamiento. Como consecuencia de éstos ataques vienen cuantiosas pérdidas económicas por estar fuera de línea.

Otros tipos de ataques que se pueden listar son: *Secuestro de sesiones*, significa tomar el control de una comunicación simulando ser un usuario válido; *Sniffers*, se observa el tráfico de la red en busca de alguna vulnerabilidad o información sensible; *Virus, Gusanos, Backdors, Bombas lógicas*, programas que tienen por objetivo copiarse a sí mismos, saturar los recursos del sistema, abrir puertas del sistema para un atacante; *Ingeniería Social*, sin necesidad de hardware o software, el atacante usando trucos psicológicos, puede obtener información útil para entrar a un sistema, se vale de la inocencia de la gente principalmente, aunque también puede aplicarse al momento de hurgar literalmente en la basura de alguna compañía.

Algunas medidas que se estudiaron como medio de prevención contra este tipo de ataques fue el uso de filtro de paquetes, con el objetivo de determinar si se trata de un paquete válido de información y permitir o no su acceso. Existe hardware y software con capacidad para el filtro de paquetes, muchos firewalls incluyen desde hace tiempo ésta característica. En Linux existen dos herramientas para realizar ésta tarea: NetFilter e IPTables. A través de estos programas los administradores tienen que especificar reglas que los protocolos deben seguir. Si algún paquete cumple con alguna de las reglas definidas se ejecuta una acción.

Uno de los últimos temas que se trataron fue el de Sistemas de Detección de Intrusos, o Intrusion Detection System (IDS), es un software diseñado para reconocer los patrones de un comportamiento no deseado en el sistema, tienen la capacidad de registrar intentos de ataque, ataques en progreso y detección de vulnerabilidades. Permiten además la identificación de riesgos y como de cómo manejarlos. Aunque actualmente están muy avanzados todavía carecen de velocidad para reaccionar ante un ataque, y en ocasiones provocan falsas alarmas, y ante patrones de ataques nuevos pueden quedar vulnerables. Algunos ejemplos de IDS son Snort, EMERALD, ISS Real Secure, TRIPWIRE, CISCO NetRanger.

En el módulo se trabajó brevemente con Snort, una herramienta compacta y muy poderosa, además de ser de fuente abierta. Snort puede ser utilizado como analizador de tráfico y sniffer, entre otras funciones, la información que recauda durante su tiempo de ejecución es almacenada en una base de datos, y ésta a su vez es utilizada por otros programas especializados para realizar diversos análisis. Durante la práctica sólo se usó en su modalidad de sniffer, y se pudo observar en pantalla como la información viajaba de forma transparente, se hizo un acceso a un sitio FTP, y datos como el login y contraseña pudieron ser vistos sin ninguna dificultad.

En las últimas prácticas de este módulo se instaló el Sistema de Administración de Contenido (Content Management System CMS) de fuente abierta conocido como Mambo Server en su versión 4.50. Esto con el fin de simular métodos con los que se puede vulnerar un sistema. Para ello se accedió a páginas dedicadas al software de detección de vulnerabilidades, como pueden ser packetstormsecurity.org o freshmeat.net, bastó con ingresar el nombre y versión del Mambo Server y apareció una lista de vulnerabilidades conocidas junto con sus exploits. Se descargó un exploit que afectaba la versión instalada del Mambo Server, y éste sólo trabajó a medias, pero funcionó como una demostración de la forma en la que los atacantes intentan ingresar a un sistema. También se trabajó con Nikto, y después de un análisis al sistema se nos mostró aquellos puntos por donde nuestro sistema era vulnerable y la forma en la que podía ser atacado.

Este curso de Introducción a la seguridad, pese a su corta duración, brindó un amplio panorama de lo que es el mundo de la seguridad informática, invitando a investigar un poco más sobre él. Además proporcionó varias herramientas con las cuales se puede armar un ambiente de trabajo lo más seguro posible. Se trabajaron con técnicas para la detección de huecos de seguridad parecidas a las que usan los atacantes, pero con la finalidad de cerrar las vulnerabilidades. Esto se complementa además con la puesta en práctica de políticas de seguridad en el área de trabajo tanto para los usuarios del sistema como para el mismo administrador.

Módulo VIII. Desarrollo de aplicaciones Web con PHP y PostgreSQL

Ahora en este módulo toca el turno a PostgreSQL, este manejador de base de datos presenta ventajas sobre MySQL. Aunque puede ser más lento en algunas transacciones con respecto a MySQL, tiene mayor robustez y mejor respuesta ante una gran cantidad de demanda de información. Es capaz de adaptarse al número de procesadores y memoria para trabajar siempre de forma óptima. Tiene la capacidad de comprobar la integridad referencial (asegurarse que las relaciones entre registros de tablas relacionadas son válidas), así como también la de almacenar procedimientos en la propia base de datos.

En el inicio de este módulo se hizo un repaso de programación orientada a objetos en PHP, con el fin de utilizarlo en el desarrollo de aplicaciones Web. Se explicaron los conceptos de atributos y métodos. Además se estudió el uso de la herencia en la POO, gracias a ésta, es posible hacer que una clase herede todo el comportamiento y los atributos a otra clase, todas sus variables y funciones. Este tipo de programación permite que el mantenimiento de un programa sea más fácil, puesto que sólo se modifica lo que se necesita.

Para mejorar el desarrollo de un sitio Web, se estudió el uso de templates, los templates son plantillas de diseño, que permiten organizar la información al momento de desplegarla en el monitor, separa la parte de desarrollo de la parte de presentación. Existen muchos templates en el mercado, algunos son de paga, otros son Open Source, y cada uno ofrece diferentes ventajas. Muchos Sistemas de Administración de Contenido trabajan mediante templates, de manera que la información siempre bajo un mismo esquema de diseño. Algunos templates son escritos en PHP, otros en Perl, y algunos más en Python. Los templates pueden ser llamados como una librería, o pueden agregarse como un módulo al intérprete del lenguaje.

Los templates trabajan como marcas dentro de un documento HTML, de manera que sólo hay que agregar la información que se desea donde se desea, por ejemplo:

```
<body> {CONTENIDO} </body>
```

El comportamiento de esto puede variar según el template que se utilice, y existen algunas funciones más elaboradas, para mejorar la presentación. Cuando una página es solicitada, la página pasa primero por el lenguaje de programación que entiende el template, para rellenar esas etiquetas con la información requerida.

En este módulo se trabajó con NokTemplate, las ventajas que presenta son las siguientes: puede utilizarse como una librería en los programas PHP, no hay necesidad de instalar algo extra en el servidor o en el lenguaje PHP, además de que las instrucciones son en español, y el consumo de recursos es mínimo.

Linux Slackware no cuenta con una versión preinstalada de PostgreSQL, así que para trabajar con esta base de datos, se descargó la última versión de la página oficial, y se instaló de acuerdo a la documentación que acompaña al archivo con el código fuente. Al igual que con MySQL, hay que crear un usuario postgres, que será el encargado de administrar las bases de datos, y por obvias razones habrá que crearle un password seguro. Todas estas recomendaciones también se hacen en la documentación de instalación.

Una vez instalado PostgreSQL, hay que inicializar las bases de datos y el lugar donde se ubicarán. Posteriormente habrá que iniciar al demonio encargado de la base de datos.

```
postmaster -i -p p 5432 - D /usr/local/pgsql/data &
```

PostgreSQL también tiene un entorno de trabajo con su propio prompt donde se ejecutan comandos destinados a realizar una acción sobre las bases de datos. Para conectarse a una base de datos a través del cliente psql se utiliza la siguiente sintaxis:

```
user@local$ psql -h host - p 5432 basededatos
```

Si la base se encuentra en la máquina local y se cuenta con los permisos entonces se escribe:

```
user@local$ psql basededatos
```

El entorno de trabajo de PostgreSQL cuenta con varias instrucciones internas, éstas son algunas básicas:

- \q Termina una sesión de psql
- \? Ayuda para los comandos del entorno
- \h Ayuda para los comandos SQL
- \d Muestra información de las tablas de la base de datos
- \i Procesa un script de instrucciones SQL

PostgreSQL al igual que otros sistemas de bases de datos, permite la creación de usuarios y grupos. Las tablas de usuarios y de grupos existen como objetos globales en la base de datos, de manera que no están adscritas a ninguna base de datos en particular. La información de los usuarios está almacenada en

una tabla del sistema llamada pg_shadow. Dicha tabla sólo puede ser vista por el superusuario.

Para la creación de usuarios existen dos formas: a través del comando CREATE USER, en el entorno de trabajo de PostgreSQL, donde sólo basta escribir el nombre del usuario, aunque es posible agregar más datos como el password o el tiempo de validez para el usuario. La otra forma para crear un usuario es desde fuera del entorno de PostgreSQL, usando el script createuser. Para modificar alguna característica de los usuarios existe la función ALTER USER seguida de las opciones a modificar. Para la eliminación de un usuario se utiliza la función SQL DROP USER, también es posible eliminar un usuario mediante el script dropuser, desde el shell de Linux. Es importante hacer notar que un usuario no puede ser eliminado si existen bases de datos de su propiedad.

Los grupos existen como un mecanismo para simplificar el establecimiento de permisos para los usuarios. Así en vez de asignar permisos por cada usuario, se crean grupos con accesos a ciertos objetos de la base de datos, y posteriormente los usuarios son añadidos a estas categorías de grupos. Para la creación de grupos se utiliza la función CREATE GROUP nombregrupo acompañado de otras opciones, y para eliminarlo se utiliza la función DROP GROUP nombre grupo. Para agregar a un usuario dentro de un grupo se utiliza la siguiente sentencia ALTER GROUP nombregrupo ADD USER usuario, y para eliminarlo del grupo se escribe ALTER GROUP nombregrupo DROP USER usuario. Es posible cambiar privilegios a los usuarios o grupos mediante las funciones GRANT y REVOKE, la primera permitirá el acceso a objetos de la base de datos, mientras la segunda revocará dicho privilegios.

Para la modificación de la información se usan los comandos INSERT INTO, SELECT, DELETE, UPDATE, al igual que en MySQL. Con estos comandos es posible realizar operaciones básicas de búsqueda, así como introducir alguna información, modificarla o borrarla dada una condición previa. Para hacer uso de estos comandos es posible hacerlo desde el entorno de trabajo de PostgreSQL o haciendo una llamada desde la aplicación Web.

PHP cuenta con una serie de instrucciones para el manejo de base da datos con PostgreSQL, para conectarse a una base de datos PostgreSQL con PHP se utiliza la siguiente sintaxis:

```
Link$=pg_connect("host=localhost port=5432 dbname=test user=postgres password=postgres");
```

Los datos pueden variar según la configuración de cada sistema. Si la base conecta satisfactoriamente la consulta a la base se realiza de la siguiente forma:

```
$result=pg_exec($link, "SELECT * FROM tabla");
```

Para la creación de una base de datos se necesita trabajar desde el shell de Linux, y utilizar el comando `createdb nombreBD`, el nombre de la base de datos tiene que empezar con un carácter alfabético, y su longitud está limitada a 31 caracteres. Para la eliminación de una base de datos tiene que ejecutarse el comando `dropdb nombreBD`, esto elimina físicamente todos los archivos asociados con la base de datos y no puede ser recuperada, ya que tener precaución al usar este comando. Igualmente puede ser eliminada una base de datos desde una sesión PostgreSQL usando `DROP DATABASE nombreBD`.

Uno de las tareas más importantes al trabajar con una base de datos es la de mantener la información segura en el sistema, de manera que la realización de una copia de seguridad es una buena política de seguridad. PostgreSQL gestiona sus propios archivos de sistema, de manera que no se sugiere sacar un respaldo directo de sus archivos, puesto que éstos podrían sufrir algún tipo de corrupción al restaurarse. Es por eso que proporciona dos utilidades para realizar copias de seguridad de su sistema: `pg_dump` que funciona para el respaldo individual de una base de datos y `pg_dumpall` para el respaldo de todas las bases de una sola vez.

```
# pg_dump nombreBD > respaldo.dump           Para generar el respaldo
# createdb nombreDB                          Base de datos vacía
# cat respaldo.dump | psql nombreBD           Para restaurarla
```

Se sugiere primero crear una base vacía en la que se vaciará toda la información del respaldo. Esto mismo procedimiento se sigue para el respaldo de todas las bases de datos. Se sugiere el uso del comando `gzip`, para comprimir el archivo de respaldo y reducir su tamaño, cuando se trate de bases de datos muy grandes.

```
# pg_dumpall | gzip > respaldo.dump.gz
```

Otra de las características importantes que trae consigo PostgreSQL es la posibilidad de escribir funciones propias para automatizar y personalizar algunas tareas sobre la base de datos. Para ello se utiliza el lenguaje PL/PgSQL, el cuál es un lenguaje procedural. El lenguaje procedural amplía las capacidades del estándar SQL y permite definir secuencias de control, de flujo y de toma de decisiones.

Este último módulo fue muy enriquecedor por el aprendizaje de distintas herramientas de desarrollo, que permiten crear aplicaciones más profesionales y cercanas a la realidad. Además el conocer otro sistema de base datos permite tener más opciones al decidir sobre cuál será la mejor opción dependiendo las características del sistema a desarrollar.

Capítulo II. Aplicaciones desarrolladas

Durante el transcurso del diplomado se desarrollaron diversos proyectos para la evaluación de los cursos. La finalidad de dichos proyectos fue poner en práctica los conocimientos adquiridos para resolución de un problema lo más cercano a la realidad. La complejidad de los proyectos iba en aumento con respecto a los lenguajes de programación y los manejadores de bases de datos que se fueron aprendiendo. Al final, al conjuntar todo esto permitió desarrollar aplicaciones más funcionales.

Portal HTML (“Editores para la creación de Páginas Web”)

Dentro de este módulo, los conocimientos adquiridos fueron suficientes para la creación de un pequeño sitio de Internet, dicho sitio debería cubrir todos los temas vistos durante el curso.

Se utilizaron etiquetas para la modificación estética del texto, así mismo se trabajaron con imágenes, se usaron frames para organizar la información en pantalla, y se diseñó un pequeño formulario que aunque no era funcional, permitió entender la forma en que trabaja, ya que el diseño de formularios es una pieza clave para el desarrollo de los sistemas.

El proyecto fue alojado en el servidor *atena* del centro Mascarones. A continuación se presentan algunas capturas.



Fig. 5 Pagina principal, uso de frames, formato de texto, y colocación de imágenes.



Fig. 6 Creación de formularios



Fig. 7 Uso de iframes.



Fig. 8 Uso de hipervínculos.

Agenda Web (“Programación con PHP”)

Para la evaluación se desarrolló una agenda en la que se pudieran agregar, borrar o modificar datos de las personas registradas, además debía contar con un sistema de autenticación para evitar que cualquier persona tuviera acceso a esta agenda, además de generar una bitácora de las personas que entraron. El sistema contó con la posibilidad de escoger desde la primera página el idioma con el que se deseaba trabajar. Puesto que en éste módulo no se trabajó con manejadores de base de datos, la información se guardó en archivos de texto. Para la autenticación se trabajó con sesiones de manera que solo el usuario que se hubiese autenticado podría utilizar el sistema.



Fig. 9 Pantalla de Inicio



Fig. 10 Pantalla de ingreso



Fig. 11 Ingreso de datos



Fig. 12 Despliegue de los datos

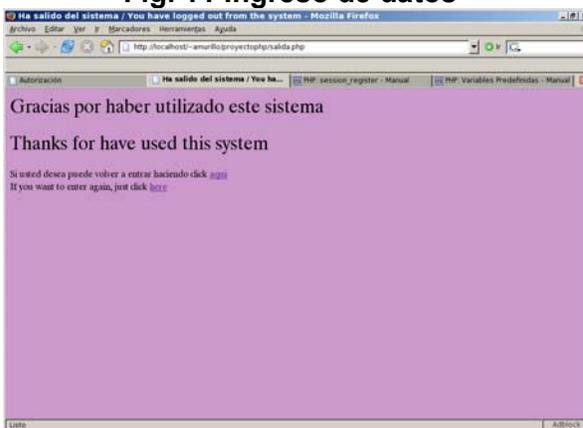


Fig. 13 Salida del sistema

Sistema de control para un hospital ("Interacción de WWW con Base de Datos MySQL")

Para la evaluación de este módulo se desarrolló una aplicación capaz de trabajar con 2 tablas y un campo que los relacionara. El tema del proyecto fue abierto, así que se desarrolló un pequeño sistema para un hospital, donde una de las tablas fue para almacenar información personal del paciente, mientras que la otra tabla fue para almacenar información referente a las consultas de cada uno de los pacientes. Las tablas se relacionaban por medio de un número identificador, dicho número es único para cada paciente y por lo tanto en el registro de las consultas se añadía el ID correspondiente. Éste fue un proyecto muy cercano a un problema real, el uso de MySQL permitió manejar la información de forma sencilla y confiable.



Fig. 14 Pantalla Principal



Fig. 15 Búsqueda de pacientes

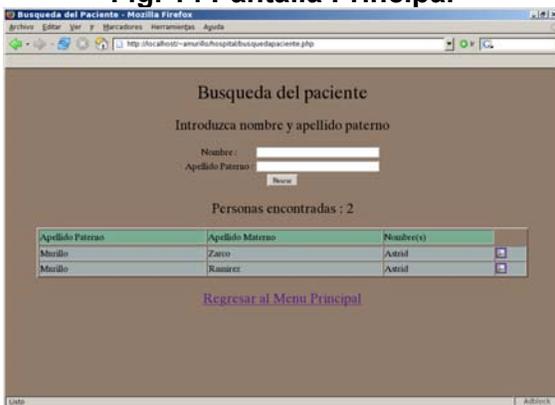


Fig. 16 Impresión de resultados



Fig. 17 Datos del paciente y sus consultas



Fig. 18 Reporte de una consulta

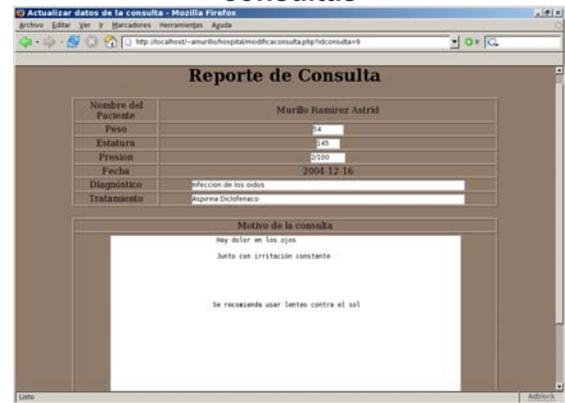


Fig. 19 Modificación de la consulta



Fig. 20 Modificación de los datos del paciente

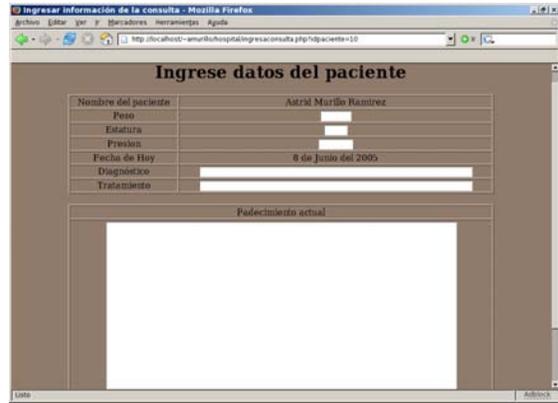


Fig. 21 Ingreso de datos para una nueva consulta



Fig. 22 Ingreso de datos para un paciente nuevo

PortalPC (“Desarrollo de aplicaciones Web con PHP y PostgreSQL”)

Para la evaluación de este módulo se desarrolló un pequeño portal, para su diseño fue necesario usar plantillas con el fin de organizar mejor la información que se va a desplegar.

La idea de utilizar los templates es para separar la parte de diseño de la parte de desarrollo, de manera que no importando cuanto pueda cambiar la estructura interna del sitio la información debe desplegarse de la misma forma. Los templates facilitaron el trabajo del diseño de la página, dándole una mejor presentación.

Este portal almacena en una base de datos todas las noticias y sitios de interés, para ello se utilizaron dos tablas respectivamente. Como manejador de base de datos se escogió a PostgreSQL por ser robusto y confiable, además de ser de los más adecuados para éste tipo de aplicaciones.

Uno de los puntos más interesantes fue el desarrollo de un módulo para realizar búsquedas dentro de los artículos, y posteriormente desplegarlos en un breve reporte, dicho reporte lleva al artículo completo con sólo dar un click sobre la liga asociada.

El Portal se dividió en varias zonas con información, una sección destinada a un listado con los últimos diez artículos publicados, otra sección con recomendaciones para visitar otros sitios Web, en la parte superior se colocó la sección de búsqueda, y la parte central funciona para desplegar los resultados de las búsquedas y los artículos.

Este proyecto es de los más cercanos a un problema real, donde es necesario conformar una página en base a la información que se tiene almacenada, y darle un formato adecuado para que sea atractivo para los usuarios. Además este tipo de estilo de portales es muy usado en diversos sitios de Internet.

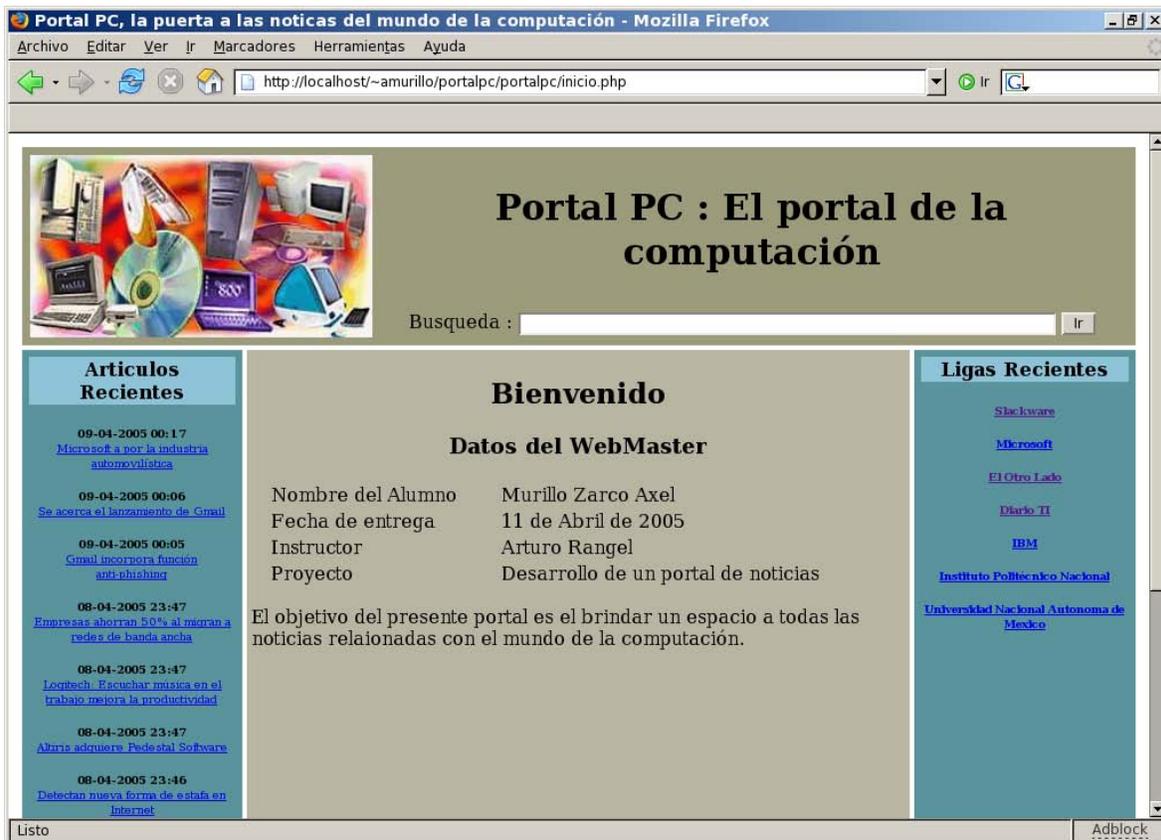


Fig. 23 Pantalla Principal



Fig. 24 Presentación de un artículo

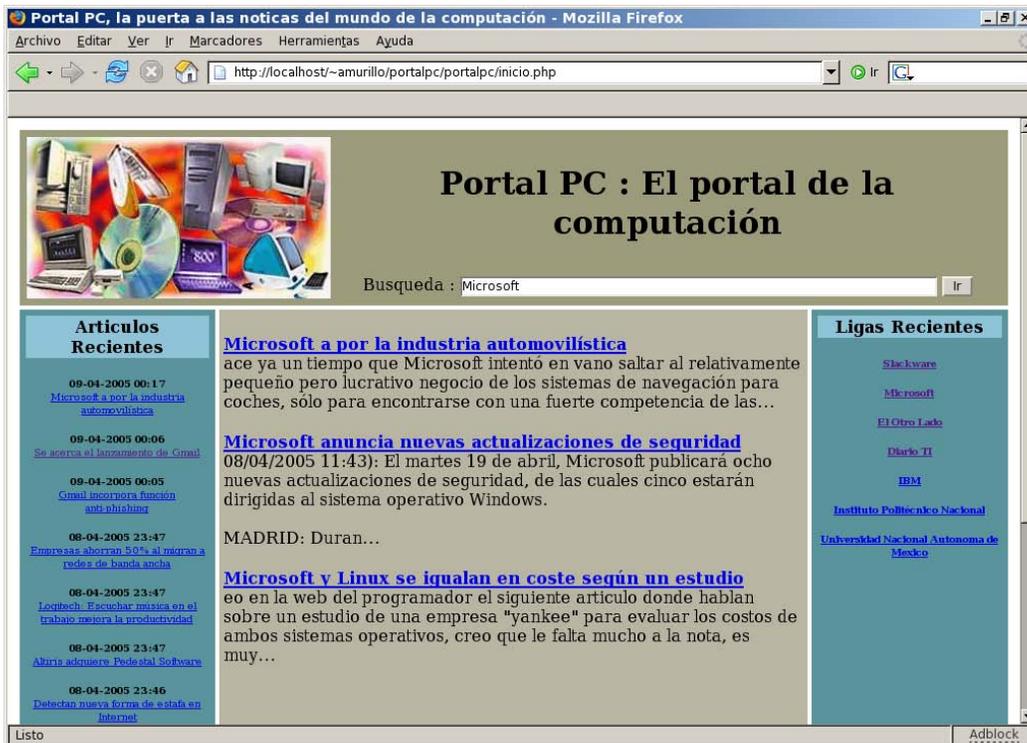


Fig. 25 Resultado de la búsqueda

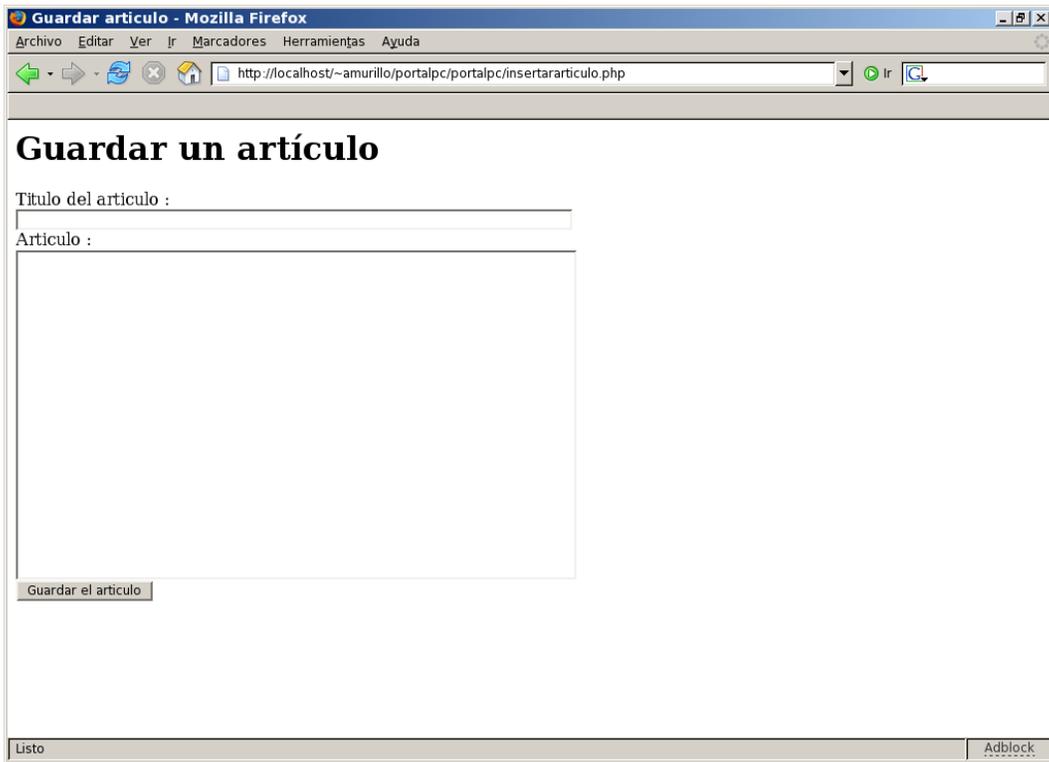


Fig. 26 Almacenar un artículo en la base de datos

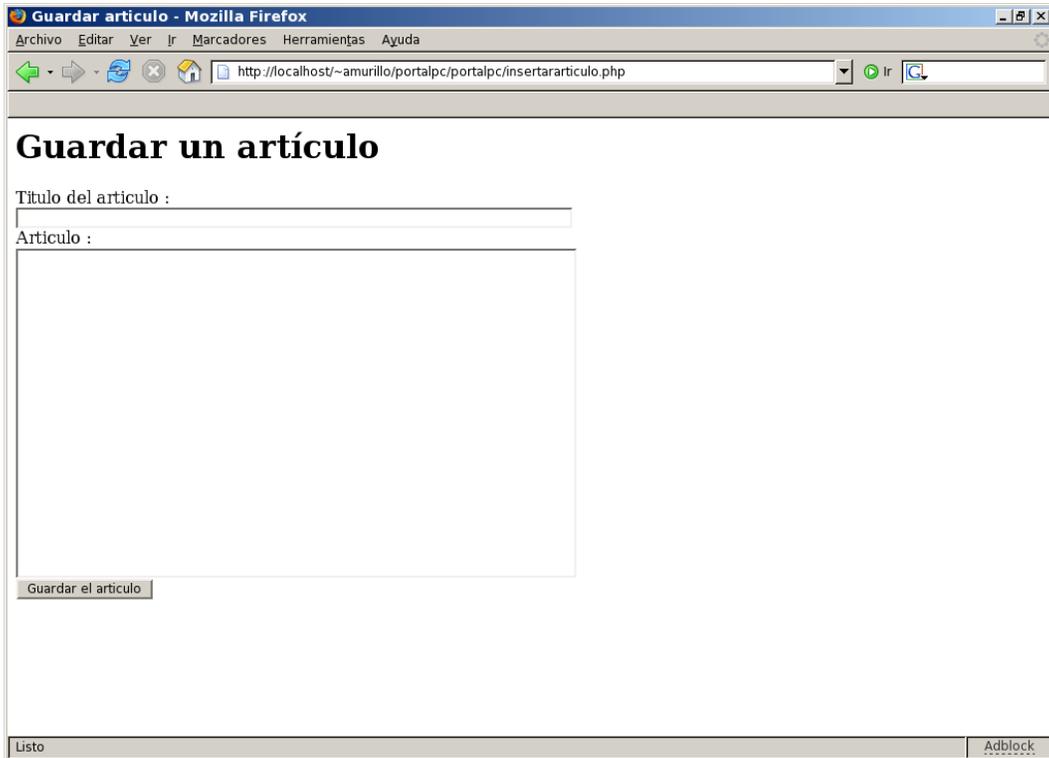


Fig. 27 Almacenar una liga en la base de datos.

Capítulo III. Reflexiones y comentarios.

El interés que me llevó a tomar este diplomado fue el crecimiento que actualmente está teniendo el software libre. El mundo de la computación es tan amplio que nos obliga a mantenernos actualizados. El software libre o abierto me ha parecido una buena opción para desarrollar aplicaciones, debido a que se han creado diversos lenguajes de programación y aplicaciones para cada necesidad. Uno como Ingeniero en Computación debe conocer una gran cantidad de herramientas para generar soluciones a la medida de los clientes.

A lo largo del Diplomado se aprendieron diversas herramientas para el desarrollo de sistemas para el manejo de la información. Conforme se avanzaba en cada módulo, los conocimientos fueron conectándose poco a poco, para ir generando varios proyectos funcionales. Pese a la corta duración de algunos cursos, los conocimientos fueron los suficientes para generar aplicaciones sencillas, que en algunos casos se complementaron con investigación en libros e Internet.

Linux es un sistema operativo en constante crecimiento que ha demostrado ser lo bastante robusto, confiable y seguro para usarse en entorno de trabajos donde la información es un factor importante. En el mercado de servidores tiene ocupada una buena parte del mercado sosteniendo importantes servicios tales como correo electrónico, buscadores, servidores de páginas, etc. Su aprendizaje es muy fácil, además con el uso de comandos es posible hacer muchas tareas sin necesidad de un entorno gráfico. Existe mucha documentación en Internet, de manera que ante algún problema la solución puede ser hallada con relativa facilidad.

En el diplomado la versión con la que se trabajó fue Linux Slackware, una de las distribuciones más antiguas que existen y una de las más apegadas a los estándares de Linux. La instalación es muy rápida y sencilla, salvo unos pocos detalles técnicos que hay que conocer, pero en general es muy accesible. Es una distribución muy ligera, y puede instalarse desde equipos muy viejos hasta en las más modernas computadoras basadas en procesadores x86. En lo personal es una muy buena distribución, y es moldeable a las necesidades de cada administrador. Su sistema de administración de paquetes permite llevar un buen control de lo que se tiene instalado en el sistema.

Tal vez algunos puntos de los cuales adolece Linux para su entrada al mercado casero de computadoras es que los escritorios de trabajo todavía carecen de cierta facilidad de uso para el usuario común. Desde mi punto de vista KDE ofrece varias facilidades para el usuario, y sigue mejorando, aunque en las últimas versiones no ha avanzado mucho. Un buen ejemplo de ello es MacOS X,

cuyo núcleo está basado en FreeBSD, lo que le permite instalar diversas aplicaciones como Apache, PHP, algunos manejadores de base de datos, etc. Y su escritorio de trabajo es amigable e intuitivo para el usuario, haciendo uso de vistosos efectos, además de contar con diversos asistentes para la configuración del sistema.

Los conocimientos adquiridos en el módulo de Administración de Linux fueron de gran ayuda para entender la forma en la que hay que administrar un sistema de éstas características. Se estudiaron las responsabilidades que se tienen al ser administrador y las tareas más comunes a realizar. Fue un módulo muy interesante donde se aprendió a personalizar el sistema a nuestras necesidades.

Con el lenguaje HTML, se diseñaron páginas Web sencillas, donde se trabajó con aspectos estéticos como los tipos de fuentes, los colores, manejo de imágenes, uso de frames, y la creación de formularios muy importantes para el ingreso de información al sistema. Con el aprendizaje del lenguaje de programación PHP, se le dio dinamismo a las páginas en función de la información que se ingresaba, además de hacer operaciones con dicha información.

Y con el uso de manejadores de base de datos como MySQL y PostgreSQL, combinado con PHP, se logró desarrollar sistemas básicos para almacenar y modificar la información, además de realizar búsquedas en la base de datos para desplegarla en pantalla. Los manejadores utilizados están a la altura de otros manejadores de paga. Estas tres herramientas combinadas permiten desarrollar poderosos sistemas para el manejo de la información. Los proyectos desarrollados se acercaron a problemas que se presentan en la realidad.

Linux junto con todos los proyectos de software libre son un buen conjunto para la creación de soluciones para las empresas e instituciones gubernamentales. El software libre ha demostrado ser una opción viable para la creación de aplicaciones confiables y seguras al mismo nivel e incluso mejor que muchas opciones comerciales que existen en el mercado.

También es posible crear combinaciones exitosas entre aplicaciones de software libre y software cerrado. Lo importante es encontrar un buen equilibrio entre ambas. Muchos proyectos de software abierto están disponibles para plataformas como Windows. De esta manera es posible ir migrando sistemas de una plataforma a otra.

El diplomado me proporcionó herramientas muy útiles para la creación de sistemas basados en Web, y me ha abierto otros campos de acción en el ámbito laboral. Aprendí a sacarle más provecho al sistema operativo Linux que se ha vuelto otra alternativa para el desarrollo de aplicaciones. Actualmente estoy trabajando ya en algunos proyectos donde estoy utilizando los conocimientos

adquiridos en el diplomado, y conforme se avanza en el desarrollo, aparecen nuevas ideas que van enriqueciendo lo ya aprendido.

Los conocimientos adquiridos durante el diplomado me han dado herramientas que me están permitiendo participar en algunos proyectos de desarrollo de sistemas, uno de los cuales fue para una agencia de viajes, y otro en la modificación de un sitio dedicado a estudios referentes al Quijote de la Mancha ubicado en Guanajuato.

En el proyecto de la agencia de viajes se desarrolló un panel de control donde el usuario puede dar de alta diversos paquetes de viajes, con información del itinerario, los hoteles incluidos, los costos, así como modificar o eliminar dichos paquetes, etc., esta información se almacena en una base de datos, de ésta manera, al tener acceso al sitio todas las páginas se van construyendo dinámicamente.

En cuanto al sitio para el centro de estudios referentes al Quijote, contaban previamente con un sistema de catálogo de imágenes, y la parte que se trabajó fue destinada a adaptarle un nuevo diseño más acorde con la temática del sitio. Por lo que fue necesario analizar solo algunas porciones del código que hacían referencia al diseño de la página, para adaptarlo al diseño deseado.

Al participar en estos proyectos he podido constatar la importancia que ha cobrado el software libre para el desarrollo de aplicaciones dentro de las empresas, por su flexibilidad en la modificación del sistema que permite llegar a los objetivos que el cliente busca.

Para el desarrollo de estos proyectos fue posible trabajarlos desde casa en una computadora con Linux Slackware instalado, junto con el servidor Web Apache, lenguaje PHP y base de datos MySQL. De ésta forma fue posible evaluar el proyecto de una forma más rápida antes de subirla al servidor. Durante el tiempo de desarrollo se presentaron distintas situaciones que fueron solucionados con investigación en el Internet.

Mis objetivos al haber cursado este diplomado son incrementar mis conocimientos en el desarrollo de sistemas con nuevas herramientas basadas en software libre con todas las ventajas que implican. Además entre mis objetivos a futuro están el de buscar estudiar una maestría en el extranjero, de preferencia Japón, y el proyecto que pretendo desarrollar es un sistema que tenga que ver con enseñanza del idioma japonés, dicho sistema estaría basado en Web. Por lo que éste diplomado resultó ser de mucha utilidad para fines de desarrollo de dicho sistema.

Después de éstas experiencias el desarrollar en aplicaciones de software libre se me ha hecho muy interesante y muy moldeable según las necesidades del

cliente. El entorno de desarrollo aunque podría parecer complicado no lo es, ya que existen editores de código muy buenos dentro de Linux como puede ser Quanta o Kwrite.

Posiblemente algunos puntos que reforzaría en el Diplomado sería en la parte de enseñanza relativo al diseño de una base de datos y aumentar el tiempo asignado para el aprendizaje del lenguaje PHP. En el futuro espero seguir en el desarrollo de sistemas principalmente basados en Linux, pero sin descartar el uso de otras opciones. Al fin y al cabo nuestro trabajo es ofrecer un excelente servicio con las mejores herramientas para el cliente, donde el único límite es sólo la propia imaginación del desarrollador.