

Informe Técnico

Instalación de un Servidor de Servicios de Gestión de Comunidades para la Educación

Autores:

Mtra. Josefina Bárcenas López

Mtro. José Antonio Domínguez Hernández

Pas. Yhibranth Lugo Peña

PROYECTO PAPIME PE-200409 FINANCIAMIENTO: DGAPA, UNAM 2009-2011

Indice

Resumen -----	3
Introducción -----	3
1. Requerimientos tecnológicos de la plataforma tecnológica -----	4
1.1 Planificación y Especificación de Requisitos -----	4
1.2 Elección de la plataforma tecnológica: Linux -----	4
1.3 Modelo Entidad-Relación del Sistema Web Comunidades de Aprendizaje Dentizt@	7
1.4 Especificación del Servidor Web (Servidores http) -----	9
2. Instalación y configuración del sitio Web Dentizt@ -----	10
2.1 Configuración de red para el servidor -----	13
2.1.1 Instalación de Apache versión (2.2.14-1) -----	13
2.1.2 Instalación y configuración de PHP con MySQL -----	14
2.1.3 Configuración de MySQL (seguridad) -----	15
2.1.4 Iniciar los servicios automáticamente de Apache y MySQL en Linux -----	16
2.1.5 Respaldo de la base de datos -----	16
2.1.6 Habilitar la salida de la página Web a Internet por medio de iptables -----	16
Conclusiones -----	17

Resumen

En el ámbito de la educación, el desarrollo de sistemas de gestión de contenidos educativos ha sido un elemento consecuencia del uso de las Tecnologías de la Información y la Comunicación en las escuelas, sin embargo pocos de estos sistemas dan independencia a sus usuarios sobre la manera en que pueden instalar y administrar un sistema de estas características.

Las particularidades de los conocimientos técnicos para instalar un sistema Web, presentan ciertas dificultades a aquellas personas que no tienen, por lo menos, un nivel medio de conocimientos de las Tecnologías de la Información, lo que a lo largo de un proyecto educativo basado en TIC llega a dificultar la operatividad del mismo. De esta manera en este documento se ofrece una guía de instalación de un servidor Linux y su preparación para la instalación de un sistema Web, por lo que no es indispensable por el momento mencionar el código desarrollado del sistema Web que se instalará en el servidor. Aclarado esta particularidad es deseable indicar que esta tampoco es una guía de Administración del sistema operativo Linux y mucho menos un manual de apoyo para el desarrollo de *scripts* en Linux. Este informe tiene la intención de documentar las actividades que se realizaron para llevar a cabo la instalación del sistema Web Dentizt@, el cual se desarrolló con el objetivo de proporcionar un espacio de interacción y comunicación entre docentes especialmente del área de Odontología de la Facultad de Estudios Superiores Iztacala.

Introducción

El desarrollo de aplicaciones en la Internet para la colaboración y comunicación entre grupos o comunidades virtuales ha desembocado en una gran variedad de opciones informáticas en la Internet, las cuales proporcionan grandes ventajas para la comunicación entre los usuarios, incluyendo el acceso a espacios donde se puede publicar información importante para los miembros de estas comunidades o para el mismo usuario. Sin embargo estas opciones presentan limitantes tanto para la administración general del propio espacio del miembro de un grupo así como para la comunidad de aprendizaje en general.

Dado lo anterior, la presencia de las comunidades virtuales en Internet, organizadas con el propósito de colaborar y explorar nuevas formas de construir el conocimiento, cuentan con los recursos telemáticos mínimos necesarios para realizar su función en el espacio virtual. Sin embargo, varios de estos grupos terminan sólo en buenos propósitos, o en un intento de existir y subsistir ya que, en muchos de los casos, quedan en el olvido después de funcionar por un periodo corto en la red, debido a que para su funcionamiento es necesario contar con los recursos informáticos y humanos que se encarguen de realizar las actividades técnicas propias para la administración, gestión y mantenimiento del sitio donde se encuentran los recursos y/o servicios a compartir.

Actualmente, y como parte de un proyecto piloto que se desarrolla en la FES Iztacala de la UNAM, existen grupos de docentes que trabajan e interactúan a través de estos medios. Es por ello que se diseñó un sitio dirigido específicamente a estos grupos académicos que tienen la necesidad de contar con un espacio donde puedan difundir sus experiencias y materiales didácticos. El proyecto del sitio Web fue pensado, diseñado y desarrollado para cubrir las necesidades de grupos de profesores con poca experiencia en el manejo de recursos telemático, proporcionando a su vez independencia en el control y administración del mismo.

Este sistema está pensado para trabajar bajo el sistema operativo Linux, por la ventaja de ser software libre, que además de reducir costos permite hacer mejoras posteriores. También elegido por contar con un esquema de seguridad basado en un sistema de permisos de lectura, escritura y ejecución establecidos a los archivos y directorios con lo que los usuarios únicamente podrán modificar sus propios archivos a menos que el administrador le otorgue permisos adicionales.

1. Requerimientos tecnológicos de la plataforma tecnológica.

1.1. Planificación y Especificación de Requisitos

El proyecto “Comunidades de Aprendizaje con Apoyos Telemáticos” requirió del diseño de un sitio Web que permitiera mantener informados y enlazados a los grupos académicos que colaboran en el proyecto. En este sitio los usuarios deben tener la posibilidad de publicar su propio material para ser consultado por estudiantes y profesores. Éste depende de un administrador, responsable de administrar el sistema, generar, actualizar y eliminar comunidades y recursos que se publican.

El sistema se desarrolló bajo la filosofía de la programación orientada a objetos. Pensado para trabajar bajo el sistema operativo Linux, por la ventaja de ser software libre, que además de reducir costos permite el ingreso al código fuente del Sistema Operativo para poder hacer mejoras. También elegido por contar con un esquema de seguridad basado en un sistema de permisos de lectura, escritura y ejecución establecidos a los archivos y directorios con lo que los usuarios únicamente podrán modificar sus propios archivos a menos que el administrador le haya dado los permisos correspondientes.

La planificación de la solución del problema, la definición de los requisitos que el sistema presenta, así como, la construcción de prototipos se ha realizado tomando en cuenta las características y necesidades de la población que utilizará el sistema, pensando en el diseño de un sistema donde los usuarios puedan posteriormente administrar sus contenidos y actividades. (Ruíz, 2007)

1.2 Elección de la plataforma tecnológica: Linux

Cuando se trabaja con Linux se está ante un sistema operativo orientado al trabajo de redes de computadoras. Linux dispone de varios protocolos como PPP, SLIP, TCP/IP, PLIP, etc., para la transferencia de archivos entre plataforma. Tiene a su disposición multitud de aplicaciones de libre distribución que permiten navegar a través de Internet y enviar y recibir correo electrónico.

Con Linux se puede montar un servidor en nuestra propia casa sin tener que pagar las enormes cantidades de dinero que piden otros sistemas. Linux es distribuido mediante una serie de distribuciones como RedHat, Slackware, Debían, etc., las cuales se diferencian por su método de instalación y por los paquetes (software) que viene incluido.

De manera funcional la arquitectura de Linux (Figura 1) para la interacción o solicitud de un servicio por parte de un programa de usuario, sigue los siguientes pasos:

1. El usuario en su programa solicita una función de la librería libc.
2. La función de libc, hace una llamada al núcleo a través de una llamada (syscall).
3. El núcleo recibe esa llamada mediante una función (System Call).
4. El núcleo dirige esta llamada a los módulos implicados que a su vez llaman a otros hasta llegar al hardware y realizar la solicitud del usuario. El núcleo hace llegar a la aplicación del usuario el resultado de su solicitud.

En términos generales podemos decir que Linux dispone de varios tipos de sistema de archivos para poder acceder a archivos en otras plataformas. Incluye un entorno gráfico X Window (Interface gráfica estándar para máquinas UNIX). Está orientado al trabajo en red, con todo tipo de facilidades como correo electrónico por ejemplo. Posee cada vez más software de libre distribución, por lo que actualmente es preferido por la mayoría de los desarrolladores de software.

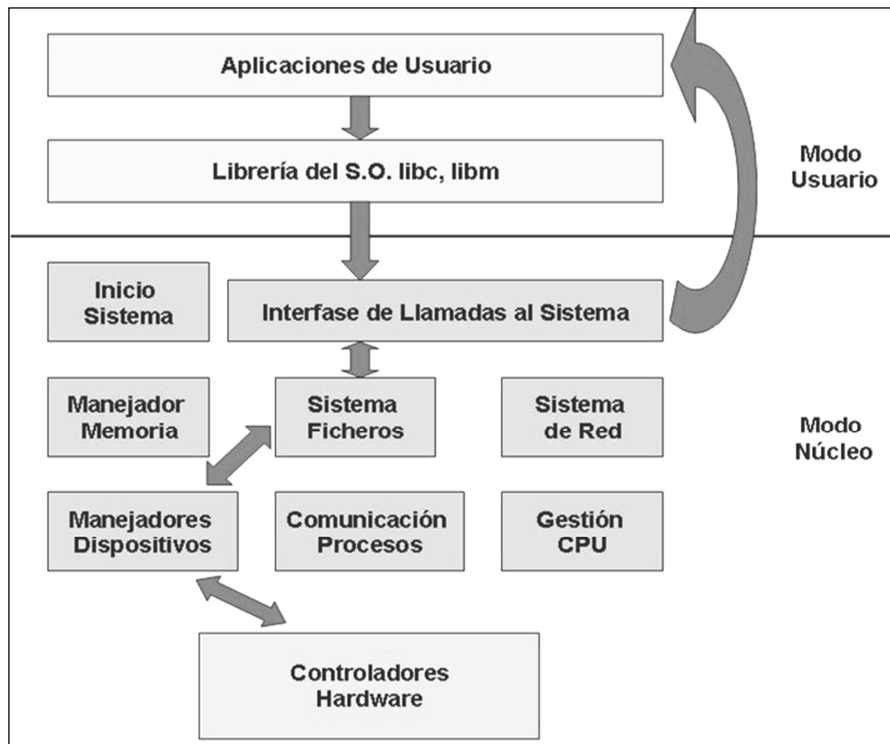


Figura 1. Arquitectura del Sistema Operativo Linux

Para llevarla a cabo se adoptó un enfoque iterativo, tomando en cada iteración un subconjunto de los requisitos y llevándolo a través del diseño de alto y bajo nivel hasta la implementación y pruebas. El sistema va creciendo incrementalmente en cada ciclo. Con esta aproximación se consigue disminuir el grado de complejidad que se trata en cada ciclo, y se tiene rápidamente funcionando una parte del sistema.

La creación de espacios virtuales en donde se intercambie información y se realice la comunicación entre los diversos integrantes del mismo (Figura 2). Es importante mencionar que la construcción de este sistema se realizó desde el punto de vista de la ingeniería del software con herramientas de seguridad que previenen los ataques informáticos de cualquier tipo, resguardando la información en cada uno de los espacios (Figura 3).

DIAGRAMA DE COLABORACIÓN

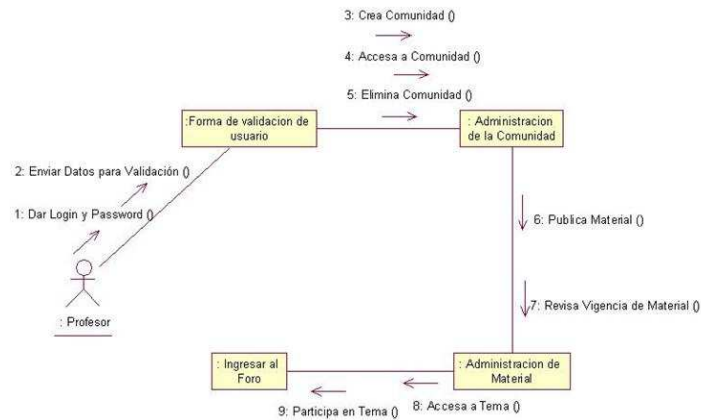


Figura 2. Entorno de trabajo colaborativo

DIAGRAMA DE CLASES

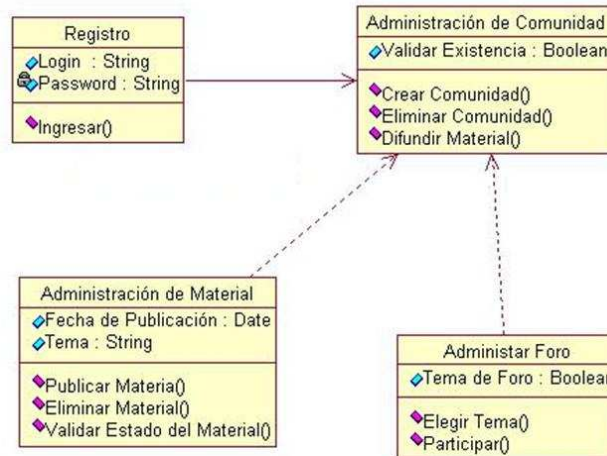


Figura 3. Autonomía del trabajo y seguridad en el entorno.

Se seleccionó el Lenguaje PHP ya que proporciona la flexibilidad de un lenguaje híbrido dónde se pueden utilizar Programación Orientada a Objetos y Programación Estructurada tradicional. PHP posee la característica de ser un lenguaje multiplataforma, lo que facilita la independencia del sistema de la plataforma en que esté en funcionamiento. Cuenta con la capacidad de conexión con el manejador de bases de datos MySQL el cual fue elegido de acorde al tamaño mediano de base de datos con que

cuenta el sistema, siendo suficiente para la construcción del mismo. Aunado a lo anterior, el costo de desarrollo con PHP, que es un software libre, representa una alternativa de fácil acceso y costo.

Asimismo, PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de Páginas Web Dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS (Servidores Web) con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (PHP).

Entre sus ventajas destacan:

- Es muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objetos, clases y herencias.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Presenta capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Sus principales desventajas son:

- Se necesita instalar un servidor Web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la Modularización.

Dificulta la organización por capas de la aplicación.

1.3 Modelo Entidad-Relación del Sistema Web Comunidades de Aprendizaje Dentizt@.

En el diseño de la base de datos de Dentizt@ se mapearon los requerimientos de información y fueron reflejados en un modelo E-R, el cual se muestra a continuación en la figura 4.

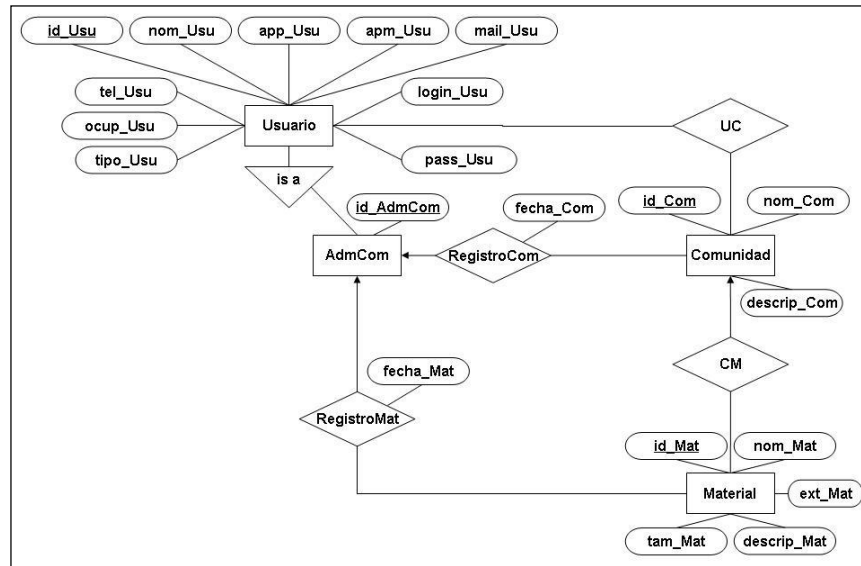


Figura 4. Modelo Entidad-Relación de la Base de Datos Dentizt@

En el diagrama se observan que para nuestro diseño existen 4 entidades las cuales se consideraron por su relevancia en el registro de la información, estas son *Usuario*, *Administrador de Comunidad*, *Comunidad* y *Material*. Dichas entidades presentan relaciones, las cuales se les ha dado el nombre de *Relación Registro de Comunidad (RegistroCom)*, *Relación Registro de Material (RegistroMat)*, *Relación Usuario-Comunidad (UC)* y *Relación Comunidad-Material (CM)*. (Larman, 2002)

En la figura 5 se muestra también el diseño de las tablas para la base de datos y la relación entre ellas de manera gráfica, lo que ayuda a comprender de otra manera lo antes mencionado.

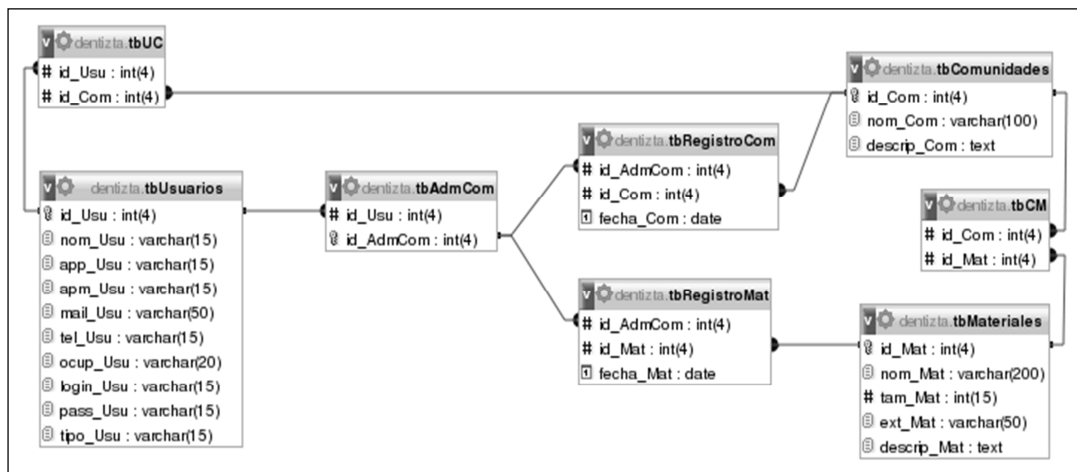


Figura 5. Diseño de las Tablas de la Base de Datos Dentizt@

La administración de bases de datos se realiza con un sistema llamado **DBMS** (Data Base Management System) –Sistema Administrador de Bases de Datos- se puede definir como: "El Conjunto de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad" (Cataldi., Lage , Pessacq, García,2003, p 185)

Las funciones principales de un DBMS son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser localizados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- El respaldo y recuperación, que consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- El control de concurrencia que controla la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- La seguridad e integridad que cuenta con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

MySQL el sistema de gestión de bases de datos elegido para este proyecto por su diseño multihilo (Permite a un programa informático realizar varias tareas a la vez) le permite soportar una gran carga de forma muy eficiente, es un gestor de bases de datos que, probablemente, es el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

1.4 Especificación del Servidor Web (Servidores http)

También conocidos como servidores de páginas Web, ya que se encargan de alojar estas últimas.

Básicamente un *servidor Web* sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP) (Flores, 2008).

El *Servidor HTTP Apache* es un software libre de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual.

El servidor Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- *Módulos Base*: Módulo con las funciones básicas del Apache
- *Módulos Multiproceso*: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- *Módulos Adicionales*: Cualquier otro módulo que le añada una funcionalidad al servidor.

A continuación se describen algunas de las funcionalidades del servidor Apache:

1. Apache puede ejecutarse en modo híbrido multiproceso-multihebra. Esto mejora la escalabilidad para muchas configuraciones aunque no para toda.
2. Tiene la infraestructura necesaria para servir distintos protocolos.
3. Apache hace automáticamente la ordenación de módulos, lo que ofrece una mayor flexibilidad.
4. Los módulos de Apache pueden escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos tal y como salen del servidor o tal y como son recibidos por el servidor.
5. Los mensajes de error que se envían a los navegadores están disponibles en diferentes idiomas, usando documentos SSI. Estos mensajes pueden personalizarse por el administrador del sitio Web.

2. Instalación y configuración del sitio Web Dentizt@

El paso inicial fue la instalación del sistema operativo bajo el cual trabajará el servidor, para lo cual se utilizó *Fedora Core 6* (Figura 6) la cual es una distribución de Linux, el cual es software libre y de código abierto. Que entre sus características cuenta con los entornos Gnome 2.16 y KDE 3.5.4, X.org 7.1 y el kernel 2.6.18.

Respecto a la seguridad, Fedora posee la arquitectura de seguridad SELinux ("Security-Enhanced Linux") la cual esta integrada en el kernel 2.6.18 usando los módulos de seguridad Linux (o Linux Security modules, LSM). Entre sus características destaca la implementación de una gran variedad de políticas de seguridad, incluyendo control de acceso obligatorio (MAC "Mandatory Access Control"), a través de los Módulos de Seguridad de Linux. (Petersen 2004)

Al ejecutar un kernel SELinux MAC se protege al sistema de aplicaciones maliciosas o dañadas que pueden perjudicar o destruir el sistema. SELinux define el acceso y los derechos de transición de cada usuario, aplicación, proceso y archivo en el sistema. SELinux gobierna la interacción de estos sujetos y objetos usando una política de seguridad que especifica cuán estricta o indulgente una instalación de Red Hat Enterprise Linux dada debería de ser.

Entre los requerimientos mínimos que se consideraron para la instalación de Fedora Core 6 se encuentran las especificaciones del procesador, el cual requería para su funcionamiento óptimo de un procesador Pentium 4 o posteriores. Respecto a la memoria RAM el mínimo a considerar son 256 MB, para el manejo de gráficos. Sobre el disco duro, se tomo en cuenta que la instalación mínima requería de 90MB y 1750MB para la instalación que incluía casi todos los paquetes. Adicionalmente tomamos en cuenta que se necesita espacio adicional para los datos del usuario, y que de acuerdo a las recomendaciones se debe reservar al menos un 5% de espacio libre para el funcionamiento adecuado del sistema.



Figura 6. Métodos de Instalación del Sistema Operativo Fedora

Fedora presenta 2 formas de instalación, el modo gráfico y el modo texto (Figura 6), siendo la primera la más recomendada, por su facilidad de uso, ya que presenta una interfaz en la que la configuración es muy sencilla y similar a la de Windows que por su popularidad es más conocida. Dentro de sus características está su modo paso a paso, que lleva de la mano a los usuarios en la instalación del sistema operativo. Dentro de sus configuraciones más relevantes, está la elección del idioma, la creación de las particiones en las que se instalará el software y la configuración de la tarjeta de red (Figuras 7 y 8). Cada una de las cuales presenta una forma predeterminada la cual es la optima, sin embargo también da opción a personalizar cada una de las configuraciones.

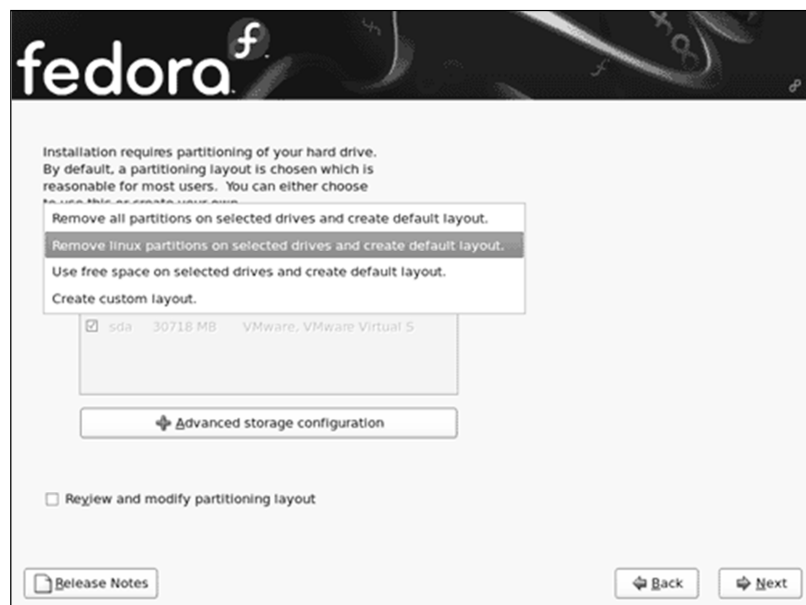


Figura 7. Creación de las particiones para la instalación de Fedora



Figura 8. Instalación y configuración de la Tarjeta de Red en Fedora

Fedora también presenta la posibilidad de seleccionar el software que queremos instalar, un ejemplo de esto son los paquetes para Desarrollo de Software y para el Servidor Web como se muestra en la Figura 9.

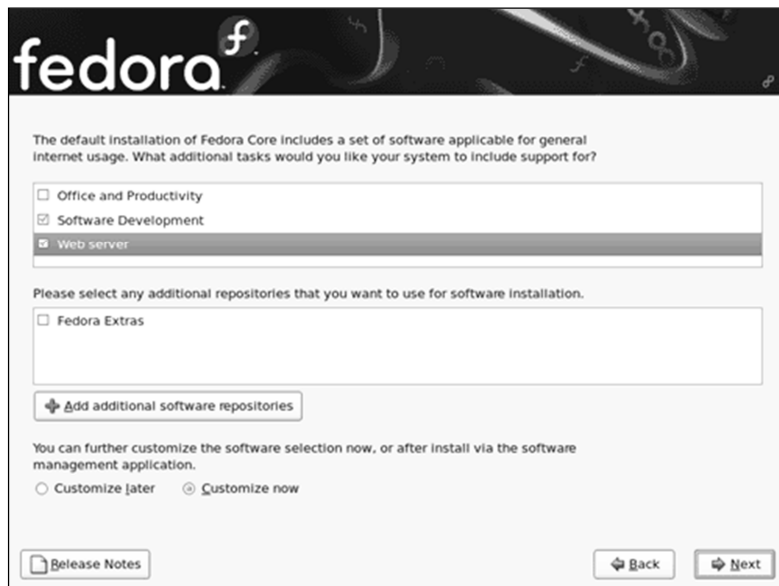


Figura 9. Instalación de paquetes para el Desarrollo Web y el Servidor Web

2.1 Configuración de red para el servidor

Con el comando “# yum update” realizamos la actualización del sistema.

Esta operación puede ser tardada dependiendo de lo que se necesite actualizar dentro del sistema además de la velocidad de conexión a Internet

2.1.1 Instalación de Apache versión (2.2.14-1)

En la consola debemos escribir la siguiente instrucción:

```
$ yum install httpd
```

De esta forma actualizamos nuestra aplicación de APACHE o realizamos una actualización del mismo.

El archivo que se tiene que modificar es httpd.conf el cual se encuentra en la siguiente ruta:
“/etc/httpd/conf/http.conf”

```
# gedit httpd.conf
```

La edición de este archivo es simple solo debemos establecer la dirección de nuestro servidor Web de esta forma:

```
Listen 132.248.36.xx:80
```

Tenemos que asignar los archivos que inicializaran como predeterminados en nuestro servidor Web asignando los siguientes parámetros:

```
DirectoryIndex index.phtml index.htm index.PHP index.PHP3 index.PHP4 index.PHP5 index.html  
index.html.var  
AddType application/x-httpd-PHP .PHP .PHP3 .PHP4 .PHP5 .phtml  
AddType application/x-httpd-PHP-source .PHPs
```

Guardamos el archivo y reiniciamos el servicio de httpd si es que esté está activo de lo contrario solo lo iniciamos.

```
Inicio “# service httpd start”
```

```
Reinicio “# service httpd restart”
```

2.1.2 Instalación y configuración de PHP con MySQL

Instalación PHP en la versión más reciente en este caso (5.3.1-1):

En la consola de Fedora escribir lo siguiente:

```
# yum install PHP PHP-devel PHP-gd PHP-imap PHP-ldap PHP-MySQL PHP-odbc PHP-pear PHP-xml  
PHP-xmlrpc PHP-eaccelerator PHP-mbstring PHP-mcrypt PHP-mhash PHP-mssql PHP-snmp PHP-soap  
PHP-tidy curl curl-devel perl-libwww-perl ImageMagick libxml2 libxml2-devel mod_fcgid PHP-cli httpd-  
devel
```

Esta instrucción instalara la versión de PHP más reciente siempre y cuando se tenga una conexión a Internet así como distintos plugins que se agregan para un mejor funcionamiento. Estos plugins ya los contiene desde una primera instalación y es posible que solo se actualicen las versiones.

La configuración de PHP es bastante sencilla y solo tenemos que modificar algunas instrucciones dentro de su archivo .ini de la siguiente manera.

```
vi /etc/PHP.ini
```

```
[...]  
cgi.fix_pathinfo = 1
```

```
# yum install MySQL
```

Aquí lo que se busca es obtener una instalación de MySQL en su versión más reciente es por ello que se tecleo la instrucción anterior.

Ahora necesitamos instalar la versión más actualizada para usar un servidor de BD con la siguiente instrucción

```
# yum install MySQL-server
```

```
[...]  
cgi.fix_pathinfo = 1
```

Cambiando el reporte de error a 1, ya que con esto obtendremos que los anuncios no se muestre

Según sea el caso se actualizará la aplicación en su versión más reciente o realizará una instalación de igual manera en su versión más reciente.

2.1.3 Configuración de MySQL (seguridad)

Para que no sea accesible la base de datos para cualquier persona debemos asignarle una contraseña y seguridad de la siguiente forma.

Primero debemos iniciar los servicios de MYSQL mediante la instrucción siguiente (dentro de la consola):

```
# service MySQLd start
```

Ahora debemos crear el usuario root para poder acceder a la BD

```
# MySQL -u root
```

De esta forma iniciamos MySQL con el usuario root y así poder asignar seguridad mediante una contraseña.

```
MySQL> SET PASSWORD FOR 'root'@'localhost' = PASSWORD ('tu_password');  
MySQL> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('tu_password');
```

Lo siguiente es comprobar que la seguridad por medio de una contraseña es correcta.

Salimos de MYSQL

```
MySQL> quit
```

Ahora comprobamos que en verdad contenga una contraseña nuestro servidor.

```
# MySQL -u root -p
```

Enter password: [teclear password]

Si la contraseña es correcta ahora podrás manipular el entorno de tu manejador de BD MySQL.

2.1.4 Iniciar los servicios automáticamente de Apache y MySQL en Linux

Para poder iniciar los servicios de APACHE (httpd) y MySQL (MySQLd) debemos de ejecutar las siguientes instrucciones desde la consola y con permisos de administrador(root) de la siguiente manera:

```
"# chkconfig --levels 235 MySQLd on" Para MySQL  
"# chkconfig --levels 235 httpd on" Para APACHE
```

2.1.5 Respaldo de la base de datos

Respaldo de la base de datos de un Servidor.

1. Entrar a la aplicación XAMPP
2. Seleccionar la BD que necesitamos respaldar
3. Exportar el código de creación de la BD en formato ".sql"

2.1.6 Habilitar la salida de la página Web a Internet por medio de iptables

Iptables es una herramienta muy potente para aceptar o negar las peticiones de los clientes para nuestro Web o incluso permitir peticiones de salida por parte de nuestra computadora.

Para poder configurar iptables es necesario hacer lo siguiente.

Elimina todas las políticas manejadas por iptables

```
# iptables -F
```

Muestra las reglas almacenadas

```
# iptables -L
```

Insertar reglas entrada

```
# iptables -A INPUT -i eth1 -p tcp -- sport 25 -j ACCEPT  
# iptables -A OUTPUT -o eth1 -p tcp -- dport 25 -j ACCEPT
```

Conclusiones

Dado que el problema primario que se observó en la versión 1.0 del sistema Web fue la ausencia de documentación, la cual fue necesaria para realizar la actualización del sitio Web Dentizt@, se consideró hacer este informe técnico con el fin mantener un seguimiento de la tecnología utilizada en el proyecto de Comunidades de Aprendizaje Telemáticas.

Uno de los resultados del seguimiento de este proyecto fue que se obtuvieron como resultado los diagramas del sistema implementado con símbolos, los cuales resultaron más sencillos de comprender que el código que se tenía de las instalaciones de Fedora, PHP y MySQL. Estos diagramas contribuyeron de manera importante al análisis sobre la funcionalidad y necesidades que se tenían para que los usuarios tuvieran un sistema integral y funcional.

El desarrollo del servidor Dentizt@ servirá para la implementación de un Sistema Web de tres capas, ya que debido a que se requiere de una independencia por parte de los grupos para la administración y la distribución de la información que se genera en los grupos, a través de un sistema que logre apoyar diversas actividades como son: la entrada, el almacenamiento y la difusión de la información, con ayuda de usuarios que autogestionen su información.

Bibliografía

Cataldi Z., Lage F., Pessacq R., García R. (2003) "Ingeniería de Software Educativo". Proceedings del V Congreso Internacional de Ingeniería Informática. Páginas 185-199. Editado por Departamento de Publicaciones de la Facultad de Ingeniería de la Universidad de Buenos Aires. Argentina

Flores J., "Reconstrucción de la arquitectura: Una actividad de la ingeniería de software." Sitio: Monografías. <http://www.monografias.com/trabajos17/reingenieria-software/reingenieria-software.shtml>. Consultado por última vez el 25 de Agosto 2008.

Larman, C. (2002) "Applying UML Patterns. 2nd. Edition", John Wiley and Sons

Petersen, R., Manual de Referencia Linux. 2^a. Mc Graw Hill: México

Ruíz J., Aguilera O. (2007) "La importancia de la Ingeniería de Software en la producción de software" Revista Holguín-Ciencias. Año, XIII, No. 2. ISSN 1027-2127. <http://www.ciencias.holguin.cu/2007/Junio/articulos/arti5.htm>. Consultada el 11 Agosto 2008