“TUTORIAL DE INSTALACIÓN Y CONFIGURACIÓN DE MAPSERVER EN CENTOS 5.2 ”

Monografía previa a la obtención del título de Ingeniero en Sistemas

AUTORES:

Jorge Leonardo Coronel Rosero
Isaac Bolívar Guzmán Suárez

DIRECTOR: Ing. Chester Sellers W.

CUENCA, ECUADOR

2009
DEDICATORIA

Este trabajo monográfico lo queremos dedicar a nuestros familiares, quienes nos depositaron la confianza necesaria durante el transcurso de nuestros estudios superiores, es así, que tenemos que citar que ellos fueron la fuente de motivación que nos llevó a dar este gran paso dentro de nuestras vidas.

Como olvidar los primeros días en las aulas de este establecimiento, es algo que nunca se borrará de nuestras mentes y corazones, es por ello, que queremos dedicar a todas aquellas personas que nos supieron entender, al momento que, compartían sus conocimientos, para así podernos convertir en profesionales. También es grato dedicar a nuestros amigos quienes siempre estuvieron presentes en los buenos y malos momentos.
AGRADECIMIENTOS

Al culminar esta etapa de nuestras vidas, queremos agradecer a nuestro padre Dios, por habernos permitido seguir de pie y decirle gracias por darnos unos padres maravillosos, quienes nos guiaron por buen camino y nos han apoyado en todo lo que hemos realizado.

Es un honor terminar nuestros estudios, en la primera universidad acreditada del Ecuador, es por ello que, queremos agradecer al personal docente y administrativo de la Universidad del Azuay, quienes son los pilares fundamentales para el desarrollo de nuestra sociedad.

También queremos agradecer a nuestro director de monografía Ing. Chester Sellers W. quien confió ciegamente en nosotros, brindándonos todo su conocimiento y apoyo constante.

De igual manera quisiéramos agradecer a nuestros amigos quienes jugaron un papel muy importante dentro de nuestra vida universitaria.
INDICE DE CONTENIDOS

Dedicatoria ........................................................................................................................................... ii
Agradecimientos .................................................................................................................................... iii
Índice de Contenidos .............................................................................................................................. iv
Índice de Ilustraciones ............................................................................................................................. v
Índice de Anexos ....................................................................................................................................... viii
Resumen ................................................................................................................................................... ix
Abstract ................................................................................................................................................... x
Introducción ..............................................................................................................................................

Capítulo I. Temario Centos ..................................................................................................................... 2
Introducción ............................................................................................................................................ 3
  1.1 Requisitos de Instalación .................................................................................................................. 3
  1.2 Instalación y Configuración ................................................................................................................ 3
  1.3 Conclusión ....................................................................................................................................... 19

Capítulo II. Temario Apache .................................................................................................................. 20
Introducción ............................................................................................................................................ 21
  2.1 Previo a la Instalación ........................................................................................................................ 21
  2.2 Instalación ....................................................................................................................................... 22
  2.3 Configuración de arranque automático ............................................................................................. 26
  2.4 Pruebas de funcionamiento ............................................................................................................. 27
  2.3 Conclusión ....................................................................................................................................... 28

Capítulo III. Temario PostgreSQL ......................................................................................................... 29
Introducción ............................................................................................................................................ 30
  3.1 Previo a la Instalación ........................................................................................................................ 30
  3.2 Instalación ....................................................................................................................................... 31
  3.3 Configuración después de la instalación ............................................................................................ 34
  3.4 Pruebas inicio y funcionamiento ....................................................................................................... 35
  3.5 Configuración de arranque automático ............................................................................................. 36
  3.6 Conclusión ....................................................................................................................................... 38

Capítulo IV. Temario MySQL .................................................................................................................. 39
Introducción ............................................................................................................................................ 40
  4.1 Previo a la Instalación ........................................................................................................................ 40
  4.2 Instalación ....................................................................................................................................... 41
4.3 Configuración después de la instalación.......................................................... 44
4.4 Pruebas inicio y funcionamiento................................................................. 45
4.5 Configuración de arranque automático....................................................... 46
4.6 Conclusión...................................................................................................... 49

Capítulo V. Temario PHP.................................................................................... 50
Introducción........................................................................................................ 51
  5.1 Previo a la Instalación.................................................................................. 51
  5.2 Instalación .................................................................................................. 53
  5.3 Configuración después de la instalación.................................................... 55
  5.4 Pruebas de funcionamiento......................................................................... 59
  5.5 Conclusión.................................................................................................... 60

Capítulo VI. Temario PostGIS............................................................................. 61
Introducción........................................................................................................ 62
  6.1 Previo a la Instalación.................................................................................. 62
  6.2 Instalación .................................................................................................. 67
  6.3 Configuración después de la instalación.................................................... 70
  6.4 Pruebas de funcionamiento......................................................................... 70
  6.5 Conclusión.................................................................................................... 72

Capítulo VII. Temario MapServer....................................................................... 73
Introducción........................................................................................................ 74
  7.1 Previo a la Instalación.................................................................................. 74
  7.2 Instalación .................................................................................................. 77
  7.3 Pruebas de funcionamiento......................................................................... 80
  7.4 Conclusión.................................................................................................... 82
Recomendaciones................................................................................................. 83
Referencias........................................................................................................... 83
Bibliografía........................................................................................................... 83

INDICE DE ILUSTRACIONES
Figura 1.1 Pantalla de Instalación de CentOS...................................................... 4
Figura 1.2 Pantalla de bienvenida de CentOS, en modo texto............................. 4
Figura 1.3 Pantalla de bienvenida de CentOS5 modo gráfico............................... 5
Figura 1.4 Pantalla para elegir el idioma de instalación........................................ 5
Figura 1.5 Pantalla para elegir el idioma de la configuración del teclado............. 6
Figura 1.6 Pantalla para la verificación de la partición................................................... 6
Figura 1.7 Pantalla para elegir el modo de partición del disco duro............................... 7
Figura 1.8 Pantalla para elegir el modo de partición del disco duro............................... 7
Figura 1.9 Ventana de Aviso............................................................................................ 8
Figura 1.10 Pantalla para elegir la configuración de la red.............................................. 8
Figura 1.11 Pantalla para elegir la configuración regional............................................... 9
Figura 1.12 Digitamos la contraseña para el usuario raíz (root)...................................... 9
Figura 1.13 Paquetes de instalación por defecto............................................................ 10
Figura 1.14 Selección para instalar los “Entornos de Escritorio”.................................... 11
Figura 1.15 Selección para instalar “Aplicaciones”.......................................................... 11
Figura 1.16 Selección para instalar “Herramientas de Desarrollo”..................................... 12
Figura 1.17 Selección para instalar “Servidores”............................................................... 12
Figura 1.18 Selección para instalar “Sistema Base”........................................................... 13
Figura 1.19 Selección para instalar Almacenamiento del Cluster...................................... 13
Figura 1.20 Mensaje de registro completo para iniciar la instalación.............................. 14
Figura 1.21 Final de la Instalación.................................................................................... 14
Figura 1.22 Pantalla de bienvenida post instalación....................................................... 15
Figura 1.23 Pantalla de Cortafuegos............................................................................... 15
Figura 1.24 Pantalla de SELinux.................................................................................... 16
Figura 1.25 Pantalla de Kdump..................................................................................... 16
Figura 1.26 Configuración fecha y hora.......................................................................... 17
Figura 1.27 Pantalla de creación de usuarios.................................................................... 17
Figura 1.28 Tarjeta de Sonido......................................................................................... 18
Figura 1.29 CDs adicionales............................................................................................ 18
Figura 1.30 Pantalla de inicio de centOS.......................................................................... 19
Figura 2.1 Sitio web para descargar Apache.................................................................... 21
Figura 2.2 Archivos del directorio httpd-2.2.11............................................................... 22
Figura 2.3 Opciones de compilación de Apache............................................................... 23
Figura 2.4 Script para la compilación de Apache 2.......................................................... 24
Figura 2.5 Resultado de la compilación del Apache.......................................................... 25
Figura 2.6 Resultado del comando make de Apache....................................................... 25
Figura 2.7 Resultado del comando make install de Apache............................................ 26
Figura 2.8 Configuración de auto arranque de Apache................................ ..................... 27
Figura 2.9 Prueba de funcionamiento del Servidor Apache............................................. 28
Figura 6.1  Sitio para descargar librería Proj4................................................................. 62
Figura 6.2  Resultado de ejecución ./configure de las librerías Proj4.................................. 63
Figura 6.3  Resultado del comando make de las librerías Proj4........................................ 64
Figura 6.4  Resultado del comando make install de las librerías Proj4.............................. 64
Figura 6.5  Sitio para descargar las librerías GEOS............................................................ 65
Figura 6.6  Resultado del ./configure de las librerías GEOS............................................. 66
Figura 6.7  Resultado del comando make de llibrerías GEOS............................................ 66
Figura 6.8  Resultado del comando make install de las librerías GEOS............................ 67
Figura 6.9  Sitio Web para descargar PostGis..................................................................... 68
Figura 6.10 Ejecución del archivo de configuración de PostGis............................................. 69
Figura 6.11 Ejecución del comando make de PostGis.......................................................... 69
Figura 6.12 Ejecución del comando make install de PostGis................................................ 69
Figura 6.13 Creando funciones........................................................................................... 71
Figura 6.14 Cargando la base de datos con registros de referencia EPSG............................ 71
Figura 7.1  Contenido del Archivo compile_gdal.sh............................................................ 75
Figura 7.2  Resultado ejecución de compile_gdal.sh........................................................... 75
Figura 7.3  Resultado del comando make de las librerías gdal......................................... 76
Figura 7.4  Resultado del comando make install de las librerías gdal............................... 76
Figura 7.5  Sitio Web para descargar MapServer............................................................... 77
Figura 7.6  Contenido del archivo compile_map.sh............................................................. 78
Figura 7.7  Resultado compile_map.sh............................................................................... 79
Figura 7.8  Resultado del comando make de MapServer..................................................... 79
Figura 7.9  Librerías cgi-bin de MapServer.......................................................................... 80
Figura 7.10 Versión de MapServer..................................................................................... 80
Figura 7.11 Información de MapScript................................................................................ 82

INDICE DE ANEXOS
Anexo I  Comando Básicos de Centos.............................................................................. 84
Anexo II  PgAdmin III........................................................................................................ 110
Anexo III  PhpMyAdmin y PhpPgAdmin........................................................................... 116
Anexo IV  Prácticas con clientes ligeros para la publicación de mapas............................. 122
RESUMEN

El presente tutorial pretende ser una guía tanto para estudiantes como para profesionales que se disponen a configurar un servidor de mapas sobre la plataforma de software libre, CentOS 5.2, además de los paquetes de software necesarios para generar un servidor web con capacidad de presentar mapas, para lo cual, hemos puesto a disposición los conceptos básicos de instalación además de instrucciones precisas y detalladas para una correcta compilación de los paquetes de software necesarios para dicha instalación y configuración. Presentaremos ejemplos prácticos para una correcta configuración y manipulación de la información a ser publicada en el servidor de mapas.
This tutorial aims to be a guide for students and professionals who are going to configure a map server on CentOS 5.2 free software platform and also the required packages of software to generate a web server with the capacity of showing maps. We show the basic installation concepts, precise and detailed instructions for the compilation of the needed software for the installation and configuration. We will show practical examples for correct configuration and information handling which will be published in the map server.
INTRODUCCION

La publicación de mapas en la web está teniendo un crecimiento de tipo exponencial dentro de nuestra sociedad, siendo esta una de las alternativas el uso comercial, industrial, académico o personal.

Con la finalidad de que esta herramienta esté al alcance de todos los entornos antes mencionados, hemos visto la necesidad de utilizar Software Libre. Es por esto que hemos tomado la iniciativa de realizar este Tutorial.

Este tutorial pretende ser una guía para estudiantes, profesionales y personas en común que describe la configuración de un servidor de mapas con las últimas versiones de los paquetes de software necesarios sobre la plataforma CentOS en su versión 5.2, por lo que hemos visto la necesidad de detallar mediante ilustraciones los pasos a seguir, con la finalidad de evitar cualquier tipo de confusión o error al momento de configurarlos.

El objetivo de este documento es incentivar la utilización de esta potente herramienta de software libre, ya que podemos ver el crecimiento e importancia que las empresas de nuestro medio le están dando, además de los decretos oficiales emitidos que incentivan el uso de este tipo de software.

Esta monografía pretende ser un tutorial de instalación y prácticas con servidores de mapas bajo la plataforma de Linux. Al finalizar la instalación de todos los componentes de software y al terminar con la configuración de los mismos, hemos puesto a disposición unos ejemplos para verificar su perfecto funcionamiento.
CAPÍTULO I

TEMARIO
Introducción
Requisitos de Instalación
Instalación y Configuración
Conclusiones
INTRODUCCIÓN

El presente capítulo hace referencia a la Instalación del Sistema Operativo CentOS en la versión 5.2, explicándolo paso a paso con ilustraciones gráficas el proceso de instalación.

CentOS 5.2 es un sistema operativo de software libre o más conocido como open source, es una distribución basada en el sistema operativo Red Hat Linux Enterprise 5.

1.1 Requisitos de Instalación.

Hardware recomendado para operar:

- Memoria RAM: 64 MB (mínimo).
- Espacio en Disco Duro: 1024 MB (mínimo) - 2 GB (recomendado).

Procesador: CentOS soporta casi las mismas arquitecturas que Red Hat Enterprise Linux:

- Advanced Micro Devices AMD64(Athlon 64, etc) e Intel EM64T (64 bit).
- Las versiones 3.x y 4.x además soportaron:
  - Intel Itanium (64 bit).
  - IBM Mainframe (eServer zSeries y S/390)...
  - Alpha procesador (DEC Alpha)
  - SPARC
1.2 Instalación y Configuración.

Al introducir el *DVD*, e iniciar el computador se visualizará la siguiente pantalla:

![Figura 1.1 Pantalla de Instalación de CentOS.](image1)

En esta pantalla no indica las opciones para la instalación de *centOS*, presionamos la tecla “*enter*” y nos indicara la siguiente pantalla.

![Figura 1.2 Pantalla de bienvenida de CentOS, en modo texto.](image2)
En esta pantalla nos pregunta si deseamos comprobar el disco de instalación, si presionamos “OK” el proceso demorará alrededor de unos 30 minutos, en nuestro caso presionamos “Skip”, para evitarnos la comprobación del disco y a continuación nos mostrara la siguiente pantalla.

**Figura 1.3** Pantalla de bienvenida de CentOS5 modo gráfico.

En esta pantalla damos clic en el botón “Next” para iniciar la instalación.

**Figura 1.4** Pantalla para elegir el idioma de instalación.
En esta pantalla nos pedirá que elijamos el idioma de instalación en nuestro caso buscaremos <spanish (Español) > y damos clic en el botón “Next”, (figura 1.4) y a continuación elegiremos el idioma para la configuración del teclado (figura 1.5)

Figura 1.5 Pantalla para elegir el idioma de la configuración del teclado.

Figura 1.6 Pantalla para la verificación de la partición.
La figura anterior nos indicará una pantalla de aviso sobre la partición, en nuestro caso presionamos “Sí” porque estamos utilizando un máquina virtual, y de este modo no se va a borrar nada de nuestro disco duro, ya que este es un disco virtual.

Figura 1.7 Pantalla para elegir el modo de partición del disco duro.

Damos “clic” donde se encuentra el puntero del ratón y nos visualizará las opciones para particionar el disco duro, en nuestro caso, elegimos la que se encuentra por defecto y presionamos el botón “siguiente” (figura 1.7, figura 1.8).

Figura 1.8 Pantalla para elegir el modo de partición del disco duro.
A continuación nos presentará una ventana en la cual presionamos el botón “SI”, ya que, nosotros estamos utilizando una maquina virtual y nuestro disco duro físico no correrá ningún riesgo (figura 1.9).

![Figura 1.9 Ventana de Aviso.](image)

Paso seguido nos mostrará la pantalla para la configuración del adaptador de red, en la cual, nos permitirá ingresar la dirección de red, en nuestro caso, elegimos la opción **DHCP** para que nos asigne una dirección de red automática (figura 1.10).

![Figura 1.10 Pantalla para elegir la configuración de la red.](image)

Presionamos el botón “**siguiente**” y a continuación elegiremos la configuración regional, en nuestro caso buscaremos “**América/Guayaquil**”, que es la referencia para el país Ecuador. Podemos hacerlo navegando en el mundo o buscándolo en la lista
desplegable (figura 1.11), una vez encontrada la ubicación procedemos a presionar el botón “siguiente”.

![Figura 1.11 Pantalla para elegir la configuración regional.](image)

Ahora nos pide que ingresemos una contraseña para el usuario raíz (root), la cual tiene que tener un mínimo de 6 caracteres (figura 1.12), una vez digitada la contraseña procedemos a presionar el botón “siguiente”.

![Figura 1.12 Digitamos la contraseña para el usuario raíz (root).](image)
La siguiente pantalla nos indica cuales son los paquetes de software que se instalarán por defecto, pero nosotros elegiremos la opción “Personalizar ahora” y daremos “clic” en el botón “siguiente” (figura 1.13).

![Figura 1.13 Paquetes de instalación por defecto.](image)

En esta pantalla procederemos a seleccionar los paquetes de software que se instalarán conjuntamente con la instalación del sistema operativo, en la parte izquierda de la pantalla, existen varias opciones, las cuales hacen referencia a los paquetes de software, en nuestro caso, vamos a dejar la sección de servidores si marcar ninguna opción ya que, algunos de estos servidores los instalaremos de forma manual (figura 1.14, 1.15, 1.16, 1.17).

En la sección de “Entornos de Escritorio” marcaremos las dos opciones disponibles como se puede apreciar en la siguiente figura (figura 1.14).
Figura 1.14 Selección para instalar los “Entornos de Escritorio”.

En la sección de “Aplicaciones” hemos decidido dejar los valores por defecto ya que no vemos la necesidad de instalar todo el paquete para cumplir el propósito de este tutorial (figura 1.15).

Figura 1.15 Selección para instalar “Aplicaciones”.

En la sección de “Desarrollo” marcamos todas las opciones para no tener problemas al momento de compilar los paquetes de software de manera manual, ya que, muchos de ellos necesitan de algunas librerías que ya podemos ya tener instaladas (figura 1.16).
Figura 1.16 Selección para instalar herramientas de desarrollo.

En la sección de “Servidores” no marcamos ninguna opción ya que procederemos a instalar manualmente (figura 1.17).

Figura 1.17 Selección para instalar “Servidores”.

En la sección “Sistema Base” de igualmente marcaremos todas las opciones para no tener complicaciones al momento de compilar el servidor de mapas (MapServer), (figura 1.18).
Figura 1.18 Selección para instalar “Sistema Base”.

Por último procedemos a marcar las opciones de virtualización, agrupamiento (clustering), almacenamiento del cluster y procedemos a presionar el botón “siguiente” (figura 1.19).

Figura 1.19 Selección para instalar Almacenamiento del Cluster

En esta pantalla presionamos “Siguiente” para iniciar la instalación (figura 1.20).
Luego nos visualizará la siguiente pantalla, el siguiente paso será esperar alrededor de 30 a 45 minutos, para que se instalen todos los paquetes de software seleccionados (figura 1.21).
En la pantalla que nos muestra la figura 1.22, nos pedirá que se reinicie el computador, para finalizar la instalación. Luego de que se reinicie el computador nos presentará la pantalla de bienvenida y a continuación configuraremos algunas opciones.

**Figura 1.22** Pantalla de bienvenida post instalación.

Presionamos el botón “Adelante” y luego seleccionamos los cortafuegos (firewall), en nuestro caso dejaremos las opciones por defecto y presionaremos el botón adelante (figura 1.23).

**Figura 1.23** Pantalla de Cortafuegos.
En la pantalla de seguridad de Linux (figura 1.24), dejaremos las opciones que se encuentran por defecto en el programa.

![Pantalla de SELinux](image)

**Figura 1.24** Pantalla de SELinux

De igual manera en la pantalla de *Kdump* dejaremos las opciones por defecto, luego presionaremos el botón “*Adelante*” (figura 1.25).

![Pantalla de Kdump](image)

**Figura 1.25** Pantalla de *Kdump*.

A continuación configuraremos fecha, hora y procedemos a presionar el botón “*Adelante*” (figura 1.26).
En la siguiente pantalla, nos recomienda crear un usuario nuevo, en nuestro caso creamos el usuario “monografía” y le asignamos un password luego presionaremos el botón “Adelante” (figura 1.27).

En la siguiente pantalla se nos indica la configuración de la tarjeta de sonido, la cual nos permite probar el sonido, luego presionamos el botón “Adelante” (figura 1.28).
Para finalizar, nos presenta la opción de instalar discos adicionales (CDs adicionales), pero como disponemos de discos adicionales, presionamos finalizar (figura 1.29).

A continuación nos visualizará la pantalla para acceder a nuestro sistema operativo, en la cual nos pedirá el nombre de usuario y posteriormente el password (figura 1.30).
1.3 Conclusión

Siguiendo todos los pasos anteriormente detallados, podremos obtener una correcta instalación del sistema operativo `centOS` versión 5.2, para a continuación configurar de mejor manera un servidor de mapas. Para recordar o para tener mejores referencias sugerimos recurrir a los comandos básicos de `Linux`, dirigirse al ANEXO I.
TEMARIO

Introducción

Previo a la Instalación

Instalación

Configuración de arranque automático de Apache

Pruebas de funcionamiento

Conclusiones
INTRODUCCION

En este capítulo explicaremos de la manera más sencilla y detallada la configuración del servidor web “Apache”, bajo la plataforma de software libre CentOS. Posteriormente indicaremos la configuración de este software y para finalizar verificaremos que el servidor se encuentre funcionando.

El servidor HTTP Apache es un servidor web HTTP 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.

2.1 Previo a la Instalación.

Nos dirigimos a la página http://httpd.apache.org/download.cgi para descargarnos el paquete de fuentes, en nuestro caso instalaremos la versión 2.2.11, entonces procederemos a descargar el archivo httpd-2.2.11.tar (figura 2.1).

![Sitio web para descargar Apache.](image)

Figura 2.1 Sitio web para descargar Apache.
Una vez descargado el código fuente del servidor web procederemos a descomprimir dicho paquete, le recomendamos que lo realice en la siguiente dirección “/usr/local/”, como se observa a continuación.

```
[root@localhost ~]# tar --xzf httpd-2.2.11.tar --C /usr/local
```

### 2.2 Instalación.

Ingresamos al directorio `/usr/local/httpd-2.2.11` con el siguiente comando.

```
[root@localhost ~]# cd /usr/local/httpd-2.2.11
```

A continuación procedemos a listar los archivos que se encuentran dentro de este directorio para identificar los archivos importantes (figura 2.2).

```
[root@localhost Instaladores]# tar --xzf httpd-2.2.11.tar.gz --C /usr/local/
[root@localhost Instaladores]# cd /usr/local/httpd-2.2.11
[root@localhost httpd-2.2.11]# ls
```

**Figura 2.2** Archivos del directorio httpd-2.2.11.

El archivo “`configure`” que vemos en la figura 2.2 de color verde, crea un comando de los archivos de configuración que utiliza la aplicación. Se procede de igual manera para la construcción de los archivos binarios, los controles de todos los componentes necesarios (`software` y sistema). Para poder revisar las opciones de este paquete digitamos el siguiente comando:

```
[root@localhost httpd-2.2.11]# ./configure --help
```

Y nos devolverá una pantalla similar a la siguiente figura 2.3.
Para compilar Apache podemos hacerlo de manera directa digitando en el terminal de comando lo siguiente “./configure”. Si por lo contrario desea realizar una compilación personalizada como la que nosotros hemos realizado podemos construir un script llamado “compile.sh” el cual lo crearemos bajo el directorio “/usr/local/httpd-2.2.11/”. 

Figura 2.3 Opciones de compilación de Apache.
lo podemos realizar con el comando “vi” o con cualquier editor de textos que se encuentren instalados en nuestro Sistema, el script estará compuesto por lo siguiente.

 specified "vi" or any text editor that may be installed in our System, the script will consist of the following.

```
./configure 
--prefix=/usr/local/apache2 
--enable-deflate 
--enable-info 
--enable-mime-magic 
--enable-rewrite 
--enable-so 
--enable-speling 
--enable-ssl 
--enable-unique_id 
--enable-usertrack 
--with-mpm=prefork 
--with-apxs2
```

Figura 2.4 Script para la compilación de Apache 2.

A continuación describiremos el significado de algunos de estos comandos:

```
--prefix=/usr/local/apache2
```

Esta línea nos indica el directorio donde instalaremos el paquete de software, en este caso Apache.

A continuación procedemos a dar permisos al archivo “compile.sh”, esto lo hacemos con el siguiente comando:

```
[root@localhost httpd-2.2.11]# chmod 755 compile.sh
```

Paso seguido tendremos que ejecutar el script y lo procederemos de la siguiente manera:

```
[root@localhost httpd-2.2.11]#/compile.sh
```

Al final del proceso de ejecución del archivo “compile.sh”, que demora alrededor de unos 20 minutos obtendremos el siguiente resultado (figura 2.5).
Si no tenemos un resultado similar al de la figura 2.5 favor revisar el archivo “./compile.sh” y volver a ejecutar. A continuación digitaremos el siguiente comando y nos devolverá una pantalla similar que se puede apreciar en la figura 2.6:

```
[root@localhost httpd-2.2.11]# make
```

*make* es una herramienta de generación o automatización de código, muy usada en los sistemas operativos tipo *Unix/Linux*. Por defecto lee las instrucciones para generar el programa u otra acción del fichero *makefile*. Las instrucciones escritas en este fichero se llaman dependencias.

**Figura 2.5** Resultado de la compilación del Apache.

**Figura 2.6** Resultado del comando *make* de Apache
Ahora tenemos que instalar el código generado por el comando `make` y lo haremos escribiendo el siguiente comando.

```
[root@localhost httpd-2.2.11]# make install
```

El resultado de este comando nos devolverá una pantalla similar a la de la figura 2.6.

![Figura 2.7 Resultado del comando make install de Apache](image)

2.3 Configuración del arranque automático de Apache

Para hacer que nuestro “Apache” personalizado se inicie junto con los demás servicios del sistema operativo, debemos realizar los siguientes pasos:

- Copiar el script que controla el arranque de Apache al directorio `/etc/init.d`

  ```
  [root@localhost httpd-2.2.11]# cp /usr/local/apache2/bin/apachectl /etc/init.d/
  ```

- Después de copiar este archivo, procedemos a editarlo ya sea con el comando “vi” o con algún editor de texto.
[root@localhost httpd-2.2.11]# vi
/etc/init.d/apachectl

- A continuación deberemos agregar las líneas que se encuentran de color azul en la figura 2.8

```
#!/bin/sh
#
# chkconfig: - 85 15
# description: Apache is a Web server used to serve HTML and CGI.
# processname: httpd
# pidfile: /usr/local/apache2/logs/httpd.pid
#
# Copyright 2000 - 2005 The Apache Software Foundation or its licensors, as applicable.
```

- **Figura 2.8** Configuración de auto arranque de Apache

- Luego ejecutaremos el siguiente comando, el cual nos permitirá anclar el servicio Apache para que se ejecute automáticamente al iniciar CentOS.

  [root@localhost httpd-2.2.11]# /sbin/chkconfig --add apachectl

- Para activar el servicio deberemos realizar el siguiente comando:

  [root@localhost httpd-2.2.11]# /sbin/chkconfig apachectl on

2.4 Pruebas de funcionamiento

Para comprobar que nuestro servidor web esté perfectamente compilado procederemos a abrir un navegador web, en nuestro caso Mozilla Firefox, en donde se digitará la dirección “ip” de la máquina, el “localhost” o su dirección por defecto “127.0.0.1” en lugar de una dirección web (figura 2.8). Si se nos presente algún error por favor verificar que el servidor este ejecutándose, si el error persiste y para mayor seguridad reiniciaremos el servicio con el siguiente comando.
[root@localhost  httpd-2.2.11]# service apachectl restart

Si no se presenta ningún error deberemos apreciar la siguiente pantalla (figura 2.8)

![Image](http://example.com/image.png)

**Figura 2.9** Prueba de funcionamiento del Servidor Apache

### 2.5 Conclusión.

Durante la recopilación de información y posterior instalación del "Apache" se pudo adquirir un gran conocimiento sobre este paquete de software y si se siguen los pasos anteriormente detallados podemos asegurar que ningún usuario tendrá problemas al instalar este servidor web.
CAPITULO III

TEMARIO
Introducción
Previo a la instalación
Instalación
Configuración después de la instalación
Pruebas de inicio y funcionamiento
Configuración de arranque automático
Conclusiones
INTRODUCCIÓN.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD (Berkeley Software Distribution). La licencia BSD al contrario que la GPL (General Public License) permite el uso del código fuente en software no libre.

En este capítulo explicaremos la manera de realizar la compilación de PostgreSQL, de forma personalizada, está la realizaremos con una guía de ilustraciones que describen a continuación.

3.1 Previo a la instalación.

Nos dirigimos a la página http://www.postgresql.org/download/linux para descargarnos las fuentes, en nuestro caso instalaremos la versión 8.3.5, entonces procedemos a descargar el archivo “postgresql-8.3.5.tar” (figura 3.1).

![Figura 3.1 Sitio web para descargar PostgreSQL](image-url)

http://www.postgresql.org/download/linux
Una vez descargado el código fuente del servidor de bases de datos, procedemos a descomprimir, le recomendamos que los realice en la siguiente dirección “/usr/local/”, como se lo puede apreciar a continuación.

```
[root@localhost ~]# tar -xzf postgresql-8.3.5.tar.gz -C /usr/local/
```

### 3.2 Instalación.

El proceso de aquí en adelante es algo similar al del capítulos anterior, el siguiente paso será ubicaremos en el directorio “/usr/local/” con el siguiente comando.

```
[root@localhost ~]# cd /usr/local/postgresql-8.3.5
```

A continuación procedemos a listar los archivos que se encuentran dentro de este directorio para identificar algunos archivos importantes (figura 3.2).

![Figura 3.2 Archivos del directorio postgresql-8.3.5.](image)

Como lo explicamos en el capítulo anterior, el archivo configure crea un comando de los archivos de configuración, que utiliza la aplicación, de igual manera para la construcción de los binarios, los controles de todos los componentes necesarios (software y sistema), necesarios para la instalación del servidor de base de datos PostgreSQL.

Para compilar PostgreSQL podemos hacerlo de manera directa digitando en la terminal de comandos el siguiente “/configure”. Si por lo contrario se desea realizar una compilación personalizada como la que nosotros hemos realizado podemos construir un script llamado “compile_pg.sh” el cual lo crearémos bajo el directorio
“/usr/local/postgresql-8.3.5/” lo podemos realizar con el comando “vi” o con cualquier editor de textos que se encuentren instalados en nuestro sistema operativo, el script estará compuesto por lo siguiente.

Paso

```
LDLIBS=-lstdc++ ./configure \
--prefix=/usr/local/pgsql \
--with-perl \
--with-python \
--with-krb5 \
--with-openssl
```

**Figura 3.3** Script para la compilación del PostgreSQL.

siguiente le otorgamos permisos de ejecución al archivo “compile_pg.sh” (figura 3.4) el cual acabamos de crear, paso seguido lo ejecutaremos y nos devolverá un texto similar al de la figura 3.5.

```
[root@localhost postgresql-8.3.5]# chmod 755 compile_pg.sh
[root@localhost postgresql-8.3.5]# ls -l
total 1596
-rwxr-xr-x 1 258 258 445 abr 23 2004 aclocal.m4
-rwxr-xr-x 1 258 258 118 abr 10 04:43 configure
-rwxr-xr-x 1 258 258 775932 oct 30 21:38 compile_pg.sh
-rwxr-xr-x 1 258 258 51678 oct 30 21:38 configure.in
-rwxr-xr-x 1 258 258 1192 ene 1 2006 COPYRIGHT
-rwxr-xr-x 1 258 258 4096 abr 21 05:42 configure
-rwxr-xr-x 1 258 258 3780 feb 9 2007 GNUmakefile.in
-rwxr-xr-x 1 258 258 642565 oct 30 22:17 HISTORY
-rwxr-xr-x 1 258 258 46251 oct 30 22:17 INSTALL
-rwxr-xr-x 1 258 258 1423 ene 20 2007 Makefile
-rwxr-xr-x 1 258 258 2026 mar 5 2008 README
[root@localhost postgresql-8.3.5]# ./compile_pg.sh
```

**Figura 3.4** listado del script de configuración personalizada.
Ejecutaremos el comando **make** y **make install** para culminar la instalación a continuación se indicaran las pantallas con el texto que deberá recibir al final de ejecutarse cada comando figura 3.6, figura 3.7.

**Figura 3.5** Ejecución del archivo **compile_pg.sh**.

**Figura 3.6** Resultado de ejecutar el comando **make** de **PostgreSQL**.
3.3 Configuración después de la instalación

Si hemos seguido los pasos anteriores tendremos instalado el servidor de bases de datos PostgreSQL, pues bien, ahora nuestra base de datos necesita ser configurada con un usuario que sea su administrador, en nuestro caso usaremos el usuario postgres, este proceso lo realizaremos a continuación.

La primera línea crea un usuario de nombre postgres, en la siguiente línea creamos el directorio “data” en el cual se amanecerán las bases de datos. Como estamos creando el directorio con el usuario “root”, tendremos que cambiar de usuario ese directorio y esto lo realizamos en el comando número tres, paso seguido nos cambiamos de usuario, y en
la línea final estamos creando los archivos de las bases de datos, el resultado de esta secuencia lo indicamos en la figura 3.8.

```
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers/max_fsm_pages ... 32MB/204800
creating configuration files ... ok
creating template1 database in /usr/local/pgsql/data/base1 ... ok
initializing pg_authid ... ok
initializing dependencies ... ok
creating system views ... ok
loading system objects' descriptions ... ok
creating conversions ... ok
creating dictionaries ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the -A option the next time you run initdb.

Success. You can now start the database server using:

   /usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data
   or
   /usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start
```

Figura 3.8 Resultado de crear los directorios de las bases de datos.

3.4 Pruebas de Inicio y Funcionamiento.

Para realizar las pruebas de funcionamiento debemos primeramente iniciar el servicio de PostgreSQL, esto lo hacemos con el siguiente comando.

```
[postgres@localhost ~]$ /usr/local/pgsql/bin/pg_ctl -D
   /usr/local/pgsql/data/ -l /usr/local/pgsql/data/logfile start
```

Una vez iniciado el servidor de bases de datos procederemos a crear una base de datos, en nuestro caso la llamaremos “ejemplo”.

```
[postgres@localhost ~]$ /usr/local/pgsql/bin/createdb ejemplo
```

Para acceder a la base de datos creada anteriormente digitaremos el siguiente comando, “psql ejemplo”, para desconectarnos de la base de datos digitamos “q”.
Una vez realizado estos pasos obtendremos un resultado similar al de la figura 3.9.

Figura 3.9 Iniciando una base de datos PostgreSQL.

Y finalmente para desconectarnos del usuario postgres digitamos exit en la ventana de comandos.

3.5 Configuración de Arranque Automático.

Para lograr que PostgreSQL inicie automáticamente cuando inicia el sistema operativo tenemos que realizar un script con el texto que se puede apreciar en la tabla 3.1 y ubicarlo en el directorio “/etc/init.d/”, lo crearemos con el nombre de “postgresql”, esto lo podemos hacer con el comando “vi” o desde cualquier editor de texto:
Figura 3.10 Contenido del script para el autoarranque del PostgreSQL

Luego de haber creado el script tendremos que otorgarle los permisos necesarios para su funcionamiento.

[root@localhost postgresql-8.3.5]# chmod 700 /etc/init.d/postgresql

Luego ejecutaremos el siguiente comando, esto nos permitirá anclar el servicio PostgreSQL para que se ejecute automáticamente al iniciar CentOS.

[root@localhost postgresql-8.3.5]# /sbin/chkconfig --add postgresql

Para activar el servicio deberemos realizar lo siguiente comando.

exit 0
Después de haber realizado los pasos necesarios para compilar *PostgreSQL* de manera personalizada sobre *centOS* 5.2 y para una mayor flexibilidad al momento de manipular bases de datos, sugerimos instalar un administrador gráfico para este potente gestor de bases de datos, para ello deberemos revisar el ANEXO II.

Cabe resaltar que dentro de nuestra investigación la *Geodatabase* más potente, versátil y práctica para implementar un servidor de mapas bajo plataforma Linux fue y sigue siendo *PostgreSQL*.
CAPITULO IV

TEMARIO
Introducción
Previo a la instalación
Instalación
Configuración después de la instalación
Pruebas de funcionamiento
Configuración de arranque automático
Conclusión
INTRODUCCION

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, en este capítulo instalaremos de manera personalizada este programa, utilizaremos representaciones gráficas para ilustrar la instalación y configuración este potente gestor de bases de datos.

4.1 Previo a la instalación.

Nos dirigimos a la página http://dev.mysql.com/downloads/ para descargarnos las fuentes, en nuestro caso, instalaremos la versión 5.1.31, entonces procedemos a descargar el archivo denominado “mysql-5.1.31.tar” (figura 4.1).

![Figura 4.1 Sitio web para descargar MySQL](image)
Una vez descargado el código fuente del servidor de bases de datos MySQL procedemos a descomprimir este paquete, le recomendamos que los realice en la siguiente dirección “/usr/local/”, como se lo puede apreciar a continuación.

```
[root@localhost ~]# tar -xzf mysql-5.1.31.tar.gz -C /usr/local/
```

4.2 Instalación.

El proceso de aquí en adelante es algo similar al de los capítulos anteriores. El primer paso es ubicarnos en el directorio “/usr/local/” con el siguiente comando.

```
[root@localhost ~]# cd /usr/local/mysql-5.1.31
```

A continuación procederemos a listar los archivos que se encuentran dentro de este directorio para identificar los archivos importantes (figura 4.2).

```
[root@localhost ~]# cd /usr/local/mysql-5.1.31
[root@localhost mysql-5.1.31]# ls
install DESTMANIFEST global.h locale-source osseccore server config.inc
make
```

Figura 4.2 Archivos del directorio mysql-5.1.31.

Como lo explicamos en los capítulos anteriores el archivo “configure” crea un comando de los archivos de configuración que se utiliza la aplicación, de igual manera para la construcción de los archivos binarios y los controles de todos los componentes necesarios (software y sistema), precisos para la instalación del servidor de base de datos MySQL.

Para compilar MySQL podemos hacerlo de manera directa digitando en el terminal de comandos lo siguiente “./configure”. Si por lo contrario desea realizar una compilación
personalizada como la que nosotros hemos realizado, podemos construir un script llamado “compile_mysql.sh” el cual lo crearemos bajo el directorio “/usr/local/mysql-5.1.31/”, o también podemos utilizar el comando “vi”, otra opción es utilizar algún editor de textos que se encuentran instalados en nuestro sistema. El script estará compuesto por el siguiente texto.

```
./configure \
--prefix=/usr/local/mysql \
--localstatedir=/usr/local/mysql/data \ 
--with-ssl=openssl \
```

**Figura 4.3** Script para la compilación del MySQL.

Un paso necesario es otorgar los permisos de ejecución al archivo “compile_mysql.sh” (figura 4.4), que acabamos de crear, paso seguido lo ejecutaremos y nos devolverá un texto similar al de la figura 4.5.

**Figura 4.4** listado del script de configuración personalizada.
El siguiente paso es ejecutar los comandos `make` y `make install` para culminar la instalación. A continuación se indicaran las pantallas con el texto que deberá presentarse al final de la ejecución cada comando figura 4.6, figura 4.7.

**Figura 4.5** Resultado de realizar ./compile_mysql.sh

![Figura 4.5](image)

**Figura 4.6** Resultado de ejecutar el comando `make` de MySQL.

![Figura 4.6](image)
4.3 Configuración después de la instalación

Si hemos seguido los pasos anteriormente descritos, tendremos instalado el servidor de bases de datos MySQL, pues bien, ahora nuestra base de datos necesita ser configurada con un usuario que sea su administrador, en nuestro caso usaremos el usuario mysql, este proceso lo realizaremos a continuación.

```
[root@localhost mysql-5.1.31]# /sbin/adduser mysql
[root@localhost mysql-5.1.31]# cd /usr/local/mysql
[root@localhost mysql-5.1.31 ~]# bin/mysql_install_db --user=mysql
```

La primera línea crea un usuario de nombre mysql, en la siguiente línea nos ubicamos en el directorio que se encuentra instalado MySQL. En la línea final instalamos las tablas de permisos MySQL, una vez realizado esto tendremos un resultado similar al de la figura 3.8.
Luego de haber instalado las bases de datos, procedemos a cambiar los usuarios de los directorios y esto lo ejecutamos con los siguientes comandos:

4.4 Pruebas de Inicio y Funcionamiento.

Para iniciar el gestor de bases de datos es necesario digitar el siguiente comando.

```
[root@localhost mysql-5.1.31]# chown -R root:mysql /usr/local/mysql
[root@localhost mysql-5.1.31]# chown -R mysql:mysql /usr/local/mysql/data
```

4.4 Pruebas de Inicio y Funcionamiento.

Para iniciar el gestor de bases de datos es necesario digitar el siguiente comando.

```
[root@localhost mysql-5.1.31]# bin/mysqld_safe -user=mysql &
```
Una vez iniciado el servicio procedemos a verificar que este correctamente instalado con el siguiente comando `/usr/local/mysql/bin/mysqladmin version`. Podemos ver este proceso en la figura 4.9

|root@localhost mysql| bin/mysql_safe >user=mysql & |
|root@localhost mysql| 00:00:10 mysql_safe Starting my SQ L daemon with databases from /usr/local/mysql/data |

Figura 4.9 Consulta de la versión instalada de MySQL

Luego ingresamos al gestor de base de datos, en este caso nos conectaremos con el usuario “root”, esto lo realizaremos con el siguiente comando `/usr/local/mysql/bin/mysql –u root`, como se puede apreciar en la siguiente figura.

Figura 4.10 Conexión al gestor de bases de datos MySq l

Para cambiar la contraseña para el ingreso al gestor de base de datos MySQL procedemos a digitar lo siguiente en la ventana de comandos.

```
[root@localhost ~]# /usr/local/mysql/bin/mysqladmin -u root
```

```
password  nueva_contraseña
```

4.5 Configuración de Arranque Automático.

Para realizar que MySQL inicie automáticamente cuando inicia el sistema, debemos utilizar los siguientes comandos:
• Primero debemos conectarnos como usuario “root”

[root@localhost ~]# su - root

• Después copiamos el archivo que contiene la información para el arranque del gestor MySQL como se puede apreciar en el siguiente comando.

[root@localhost ~]# cp /usr/local/mysql-5.1.31/support-files/mysql.server /etc/rc.d/init.d/mysql

• Para hacer que el “script” se ejecute en la máquina, al momento de iniciarse el sistema es necesario crear un enlace simbólico a la misma, y esto lo haremos con los siguientes comandos.

[root@localhost ~]# cd /etc/rc.d/rc3.d
[root@localhost ~]# ln -s ../ init.d / mysql S85mysql
[root@localhost ~]# cd /etc/rc.d/rc5.d
[root@localhost ~]# ln -s ../ init.d / mysql S85mysql

• Es necesario crear los enlaces simbólicos para a detener el servicio MySQL, al momento que el sistema este en ejecución, para ello digitamos los siguiente comandos:

[root@localhost ~]# cd /etc/rc.d/rc0.d
[root@localhost ~]# ln -s ../ init.d / mysql K85mysql
[root@localhost ~]# cd /etc/rc.d/rc6.d
[root@localhost ~]# ln -s ../ init.d / mysql K85mysql

• La secuencia de comandos MySQL debe tener permisos de ejecución para que funcionen correctamente.

[root@localhost ~]# cd .. / init.d
Un paso adicional es que tendremos que reiniciar el sistema y comprobar que el servicio esté iniciándose automáticamente, o ir al menú principal de CentOS (figura 4.11).

Figura 4.11 Pantalla de ingreso a los servicios del sistema

Paso seguido se nos presentará una ventana en la que buscaremos el servicio `mysql` en la parte izquierda de la misma, a continuación nos dirigimos con la barra de scroll para identificar dicho servicio, si el servicio esta activado, esto nos indica que el servicio se inicia la cada vez que se reinicie el sistema.

Figura 4.12 Configuración de servicios
4.6 Conclusión.

Siguiendo las instrucciones anteriormente detalladas, obtendremos la correcta instalación y configuración del Gestor de Bases de Datos MySQL sobre CentOS 5.2, con lo que garantizamos su correcto funcionamiento y la total versatilidad que presenta este gestor de bases de datos.
INTRODUCCIÓN

**PHP** es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt (es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario) o GTK+ (*The GIMP Toolkit* es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (*GUI*), principalmente para los entornos gráficos *GNOME*, *XFCE* y *ROX* aunque también se puede usar en el escritorio de *Windows*, *MacOS* y otros.).

En este capítulo explicaremos una instalación personalizada de la última versión 5.2.8 del lenguaje de programación PHP, para tener una perfecta interacción con *MapServer*.

**5.1 Previo a la Instalación.**

Antes de instalar **PHP** es necesario recordar que este lenguaje conjuntamente con *MapServer* deben tener soporte para ODBC (*Object Data Base Conection*), así como conexiones DNS (*Domain Name System*), entonces procederemos a instalar los siguientes paquetes: *unixODBC* y *unixODBC-devel* esto lo haremos con el comando *Yum* de la siguiente manera (figura 5.1). Para poder utilizar el comando *Yum* debemos tener acceso a internet.

```
[root@localhost ~]# yum install unixODBC unixODBC-devel
```
Después de haber instalado los paquetes ODBC tenemos que instalar los paquetes Freetds que son librerías que nos provee acceso a los servidores MS-SQLServer y Sysbase, por medio de una implementación de protocolo tabular DataStream. Estos paquetes no están disponibles para instalarlos con la aplicación Yum, por lo que debemos descargarnos de los siguientes sitios web:

http://rpm.pbone.net/index.php3/stat/4/idpl/3965725/com/freetds-0.64-1.el5.rf.i386.rpm.html

http://rpm.pbone.net/index.php3?stat=26&dist=52&size=850298&name=freetds-devel-0.64-1.el5.rf.i386.rpm

Una vez descargados procedemos a instalarlos con el siguiente comando, revisar figura 5.2:

```
[root@localhost ~]# rpm --hiv freetds-0.64-1.el5.rf.i386.rpm

[root@localhost ~]# rpm --hiv freetds-devel-0.64-1.el5.rf.i386.rpm
```

**Figura 5.1** yum install unixODBC unixODBC-devel
5.2 Instalación.

Lo primero que tenemos que realizar es descargar las fuentes de “PHP” y esto lo haremos desde el siguiente sitio web figura (5.3):

http://www.php.net/get/php-5.2.8.tar.gz/from/a/mirror

Luego de haber descargado las fuentes, procedemos a descomprimirlas bajo el directorio “/usr/local/” con la siguiente línea de comando.

[root@localhost Instaladores]# tar jvxf php-5.2.8.tar.bz2 -C /usr/local/
Entramos en el directorio /usr/local/php-5.2.8 con el siguiente comando.

[root@localhost Instaladores]# cd /usr/local/php-5.2.8

Creamos el archivo para la compilación, en nuestro caso lo llamaremos “compile_php.sh” esto lo hacemos desde el editor “vi” o algún editor de textos. El texto del archivo será el que se muestra en la figura 5.4, una vez creado, procederemos a dar los permisos de ejecución (chmod 755 compile_php.sh), el siguiente paso es ejecutar el comando (/compile_php.sh) (figura 5.5), y para finalizar ejecutaremos los comandos make (figura 5.6) y make install (figura 5.7), con lo que se presentarás diferentes pantallas que prueban la instalación, la captura de estas pantallas se detalla a continuación.

```
//configure
   --prefix=/usr/local/apache2/php
   --with-axp2=/usr/local/apache2/bin/apxs
   --disable-cgi
   --with-config-file-path=/usr/local/apache2/php
   --with-openssl
   --with-kerberos
   --with-zlib
   --with-bz2
   --with-curl
   --enable-database
   --with-gd
   --with-psql
   --with-xe
   --with-gettext
   --with-regextest
   --with-external-gcc=/usr/
   --with-openssl=configure
   --with-mysql=/usr/local/mysql
```

Figura 5.4 Archivo compile_php.sh

```
Reading config
Writing configuration tag "CM" to libexec
Generating files
Creating conf.mak
Creating apxs
Creating php-config
Creating html/index.html
Creating html/longindex.html
Creating html/index.gz
Creating html/longindex.gz
```

Figura 5.5 Resultado de ejecutar el archivo de compile_php.sh
5.3 Configuración después de la Instalación.

Es muy importante identificar el archivo de configuración de "PHP" (php.ini), este archivo no se instala cuando realizamos la compilación, es por esto que dentro del directorio de fuentes viene un archivo (php.ini-recommended) el cual copiaremos y lo cambiaremos de nombre, esto lo hacemos con el siguiente comando. Los pasos anteriormente detallados se los realiza para que el programa de "PHP" se ejecute correctamente.
Abrimos el archivo php.ini que acabamos de copiar, y editamos lo siguiente:

Lo primero que vamos a realizar es aumentar el tiempo de procesamiento, recursos de memoria, para esto debemos buscar alrededor de la línea 300 e identificar el siguiente texto.

```ini
max_execution_time = 30 ; Maximum execution time of each script, in seconds
max_input_time = 60 ; Maximum amount of time each script may spend parsing request data
memory_limit = 128M ; Maximum amount of memory a script may consume (128MB)
```

Y configurar estas líneas de la siguiente manera.

```ini
max_execution_time = 150 ; Maximum execution time of each script, in seconds
max_input_time = 300 ; Maximum amount of time each script may spend parsing request data
memory_limit = 256M ; Maximum amount of memory a script may
```

56
Ahora editaremos este archivo para que tenga la posibilidad de desplegar de mensajes de error (warnings), etc., para esto nos ubicaremos por la línea 349 y buscaremos el siguiente texto.

```php
error_reporting = E_ALL
```

Y lo editaremos de la siguiente manera

```php
error_reporting = E_ALL & ~E_NOTICE
```

Luego nos ubicaremos por la línea 356 y buscaremos el siguiente texto.

```php
display_errors = Off
```

Y lo editaremos de la siguiente manera.

```php
display_errors = On
```

Luego no ubicaremos por la línea 453 y buscaremos el siguiente texto.

```php
register_long_arrays = Off
```

Y lo editaremos de la siguiente manera.

```php
register_long_arrays = On
```

Y por último señalaremos el directorio de las extensiones, para esto buscaremos cerca de la línea 520.

```php
extension_dir = "./"
```

Y lo editaremos de la siguiente manera
extension_dir = "/usr/local/apache2/php/ext/

Ahora procedemos a crear el directorio para las extensiones de "PHP".

[root@localhost ~]# mkdir /usr/local/apache2/php/ext/

Después de este proceso debemos editar el archivo de configuración de Apache, este archivo lo ubicamos en el directorio /usr/local/apache2/conf/httpd.conf.

Para comprobar si el módulo “PHP” está perfectamente configurado nos ubicamos cerca de la línea 53 y comprobamos que se encuentre el siguiente texto, si no, lo está tendremos que agregarlo.

    # Example:
    # LoadModule foo_module modules/mod_foo.so

    LoadModule php5_module modules/libphp5.so

    #

Ahora agregaremos lo siguiente para que identifique el “index.php” para que se abra automáticamente en un sitio, esto lo haremos ubicándonos cerca de la línea 165.

    <IfModule dir_module>
        DirectoryIndex index.php index.html
    </IfModule>

Y para que apache pueda interpretar los archivos de tipo “PHP” tendremos que agregar lo siguiente cerca de la línea 310.

    AddType application/x-httpd-php .php .phtml

    AddType application/x-httpd-php-source .phps
Para finalizar y comprobar que “PHP” esté bien instalado y configurado tendremos que reiniciar el servicio de Apache, esto lo haremos de la siguiente manera.

```
[root@localhost php-5.2.1]# service apachectl restart
```

Luego de esto se nos presentará un error, como se puede apreciar en el siguiente figura.

```
[root@localhost php-5.2.8]# /etc/init.d/apachectl start
[root@localhost php-5.2.8]#
```

**Figura 5.8** Error libphp5.so

Para reparar este archivo deberemos digitar lo siguiente en la ventana de comandos, con lo que se podrá reparar la librería y obtener un correcto funcionamiento de la misma.

```
[root@localhost ~]# chcon -t texrel_shlib_t /usr/local/apache2/modules/libphp5.so
```

### 5.4 Pruebas de Funcionamiento.

Para comprobar que “PHP” está bien instalado debemos crear un archivo con extensión “PHP” bajo el directorio en donde se crean los sitios web, dentro de Apache, (/usr/local/apache2/htdocs/), en este directorio crearemos el archivo “index.php” y dentro de este archivo escribiremos lo siguiente:

```
<?php
    phpinfo();
?>
```

Ahora abriremos un navegador de páginas web y digitaremos “localhost” y lo cual nos presentará la siguiente pantalla (figura 5.9).
Un vez instalado **PHP** podremos administrar las de bases de datos que se hayan instalados anteriormente, desde un entorno *web*, el proceso de instalación y configuración lo explicaremos en el ANEXO III.

### 5.5 Conclusión.

Este capítulo define de forma clara y precisa los pasos a seguir para una compilación personalizada de los paquetes necesarios para la instalación y configuración de **PHP**, con la finalidad de tener una interacción con diferentes soportes ya sean para bases de datos y/o servidores.

---

**Figura 5.9 Phpinfo**

![Phpinfo](image-url)
TEMARIO
Introducción
Previo a la instalación
Instalación
Configuración después de la instalación
Pruebas de funcionamiento
Conclusión
INTRODUCCIÓN

*PostGIS* es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional *PostgreSQL*, convirtiéndola en una base de datos espacial, para su utilización en Sistema de Información Geográfica. Se publica bajo la norma general GNU *(General Public License)*.

En este capítulo instalaremos de manera personalizada y sencilla, el módulo *PostGIS* para así poder manejar datos georeferenciados en el gestor de bases de datos *PostgreSQL*.

6.1 Previo a la Instalación.

Para tener una perfecta instalación del módulo *PostGIS* primero necesitamos instalar las librerías de proyecciones cartográficas *Proj4*, ya que, estas librerías se utilizan para poder realizar proyecciones, cambios de *Datum* (geometría de referencia, sea ésta en una línea o un plano) o Sistemas de Coordenadas dentro del motor de datos. Entonces tendremos que descargarnos estas librerías del sitio web [http://trac.osgeo.org/proj/](http://trac.osgeo.org/proj/) (figura 6.1).

![Figura 6.1 Sitio para descargar librería Proj4](image)
Una vez descargado el código fuente de las librerías **Proj4**, procederemos a descomprimirlo, le recomendamos que esto se realice en la siguiente dirección: 
“/usr/local/”, de la siguiente manera.

```
[root@localhost ~]# tar -xzf proj-4.6.1.tar -C /usr/local/
[root@localhost ~]# cd /usr/local/proj-4.6.1
```

Ahora ejecutaremos directamente el archivo “**configure**” (figura 6.2), ya que vamos a instalar un módulo de librerías, las cuales serán usadas por otros paquetes. Luego de ejecutar el archivo de configuración tendremos que ejecutar el comando “**make**” (figura 6.3) y para finalizar la instalación de las librerías el comando “**make install**” (figura 6.4).

```
[root@localhost proj-4.6.1]# ./configure
```

![configure](image)

**Figura 6.2** Resultado del ./configure de las librerías Proj4

Si después del ejecutar el archivo de configuración obtenemos el resultado de la figura 6.2, continuamos con la ejecución del comando “**make**”. 

63
Finalmente, si el resultado es similar al de la figura 6.3 ejecutaremos el comando “make install” para finalizar con la compilación de las librerías Proj4.

Al igual que las librerías Proj4 también existen las librerías GEOS, éstas se emplean para realizar pruebas entre geometrías tales como: touches(), contains(), intersects() y operaciones de GeoProcesamiento tales como: buffer(), geomunion(), difference(),...
etc., dentro del motor de base de datos, entonces a continuación procederemos a instalar estas librerías.

Necesitamos descargarnos el paquete de librerías GEOS del siguiente sitio web http://trac.osgeo.org/geos/ (figura 6.5).

Una vez descargado el código fuente de las librerías GEOS, procederemos a descomprimirlo, lo recomendamos que esto se lo realice en la siguiente dirección “/usr/local/”, como lo hacemos a continuación.

```
[root@localhost ~]# tar jvxfgeos-3.1.0.tar.bz2 -C /usr/local/
[root@localhost ~]# cd /usr/local/geos-3.1.0
```

Como se trata de librerías, al igual que en el caso anterior, ejecutamos directamente el archivo “configure” (figura 6.6), luego de ejecutar el archivo de configuración tendremos que ejecutar el comando “make” (figura 6.7) y para finalizar la instalación de las librerías el comando “make install” (figura 6.8).
Al finalizar la ejecución del archivo `./configure`, obtendremos el resultado de la figura 6.6, a continuación ejecutaremos el comando `make`.

```
[root@localhost geos-3.1.0]# ./configure
checking build system type... 1686-redhat-linux-gnu
checking host system type... 1686-redhat-linux-gnu
checking target system type... 1686-redhat-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
cHECKING WHETHER BUILD ENVIRONMENT IS SAME... yes
checking whether gawk... gawk
cHECKING WHETHER MAKE SETS S(MAKE)... yes
checking for gcc... gcc

config.status: creating tests/unit/Makefile
cFG.config: creating tests/tut/Makefile
cFG.config: creating tests/xmltester/Makefile
cFG.config: creating tools/Makefile
cFG.config: creating tools/geos-config
config.status: creating source.headers/config.h
config.status: creating source.headers/geos/platform.h
config.status: executing depfiles commands
Swig: false
Python: false
Ruby: false
```

Figura 6.6 Resultado del `./configure` de las librerías GEOS

Figura 6.7 Resultado del comando `make` de las librerías GEOS
Finalmente si el resultado es similar al de la figura 6.7, ejecutaremos el comando “make install” para finalizar con la compilación de las librerías GEOS.

6.2 Instalación.

Procederemos a descargar los códigos fuente de este módulo, para ello nos dirigimos a la dirección http://www.postgis.org/download/ (figura 6.9).
Luego de haber descargado el código fuente tendremos que descomprimirlo, nosotros sugerimos la siguiente ruta “/usr/local/postgresql-8.5.3/contrib/”, ya que, si bien sabemos este es un módulo de PostgreSQL, entonces lo enviaremos al directorio donde está el código fuente del mismo, esto lo hacemos con el siguiente comando.

```
[root@localhost ~]# tar -xzf postgis-1.3.5.tar.gz -C /usr/local/postgresql-8.3.5/contrib/
```

Una vez realizado esto, nos ubicamos en el directorio “/usr/local/postgresql-8.3.5/contrib/postgis-1.3.5” con el comando “cd”, para después ejecutar el archivo de configuración, este proceso lo indicaremos en la figura 6.10. Después de ejecutar el archivo de configuración procedemos a ejecutar el comando “make” (figura 6.11) y paso seguido el comando “make install” (figura 6.12).

```
[root@localhost postgis-1.3.5]# ./configure --with-postgresql=/usr/local/pgsql/bin/pg_config
```
Figura 6.10 Ejecución del archivo de configuración de PostGis

```
[root@localhost postgres-1.3.5]# ./configure --with-psql=/usr/local/pgsql/bin/pg_config
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables... o
checking whether we are using the GNU C compiler... yes

HOST_OS: linux-gnu
PGSQL: /usr/local/pgsql/bin/pg_config
GEOS: /usr/local/bin/geos-config (with C-API)
(lflags): -L/usr/local/lib
PKG: prefix=/usr/local libdir=/usr/local/lib
ICONV: I
PORTNAME: linux
PREFIX: /usr/local/pgsql
EPREFIX: $prefix
DOC: /usr/local/pgsql/doc/contrib
DATA: $datarootdir
MAN: $datarootdir/man
BIN: /usr/local/pgsql/bin
EXT: /usr/local/pgsql/lib ($$libdir)
```

Figura 6.11 Ejecución del comando `make` de PostGis

```
[root@localhost postgres-1.3.5]# make
make -C lwgeom
make[1]: se ingresa al directorio `/usr/local/postgresql-8.3.5/contrib/postgis-1.3.5/lwgeom`
 make[1]: se ingresa al directorio `/usr/local/postgresql-8.3.5/contrib/postgis-1.3.5/lwgeom`

```

```
6.3 Configuración después de la Instalación.

*PostGis* posee dos herramientas de consola llamadas *DUMPERS* (convertidores en *batch*) (*shp2pgsql* y *pgsql2shp*), estos archivos se utilizan para cargar y descargar archivos de tipo *shape* de *ERSI* hacia y desde la base de datos, estas herramientas están depositadas en el directorio “/usr/local/pgsql/bin”, después de haber terminado la compilación tenemos que agregarlas al PATH del sistema operativo para que se puedan ejecutar y utilizar desde cualquier directorio, esto lo realizamos copiándolos al directorio “/usr/bin/” con los siguientes comandos:

```
[root@localhost postgis-1.3.5]# cp /usr/local/pgsql/bin/shp2pgsql /usr/bin
[root@localhost postgis-1.3.5]# cp /usr/local/pgsql/bin/pgsql2shp /usr/bin
```

Después de esto debemos registrar en el sistema operativo las librerías generadas por *Proj4, GEOS* y *PostGIS*. Esto lo haremos editando el archivo “/etc/ld.so.conf” y añadiremos lo siguiente “/usr/local/lib”. Finalmente recargaremos las librerías del sistema con el comando “*ldconfig*”.

Por último deberemos copiar todas las librerías compartidas de *PostgreSQL* al directorio “/usr/lib” para que *MapServer* pueda tener acceso a las mismas en tiempo de ejecución.

---

**Figura 6.12** Ejecución del comando *make install* de PostGis
6.4 Pruebas de Funcionamiento.

Para realizar las pruebas, necesitamos tener creada una base de datos en PostgreSQL, en nuestro caso utilizaremos la base de datos “ejemplo” que creamos en el capítulo III. Para ello debemos cambiarnos al usuario “postgres” y ejecutar los siguientes comandos:

```
[root@localhost postgis-1.3.5]# su - postgres

[postgres@localhost postgis-1.3.5]# /usr/local/pgsql/bin/createlang plpgsql ejemplo

[postgres@localhost postgis-1.3.5]# /usr/local/pgsql/bin/psql -d ejemplo -f /usr/local/pgsql/share/lwpostgis.sql (ver figura 6.13)

[postgres@localhost postgis-1.3.5]# /usr/local/pgsql/bin/psql -d ejemplo -f /usr/local/pgsql/share/spatial_ref_sys.sql (ver figura 6.14)
```

![Figura 6.13 Creando funciones.](image)
Finalmente nos desconectamos del usuario "postgres" con el comando "exit".

6.5 Conclusión.

Este capítulo es un complemento de la base de datos PostgreSQL, ya que, instalamos librerías para el manejo de datos espaciales. Esta instalación la detallamos paso a paso presentando también ilustraciones para que no exista ningún tipo de confusión. Se recomienda poner mucho énfasis al momento de crear la base de datos con compatibilidad para datos espaciales.

**Figura 6.14** Cargando la base de datos con registros de referencia EPSG
CAPITULO VII

TEMARIO

Introducción
Previo a la instalación
Instalación
Pruebas de funcionamiento
Conclusión
INTRODUCCION

MapServer es un entorno de desarrollo en código abierto (Open Source Initiative) para la creación de aplicaciones SIG (Sistema de Información Geográfica) en Internet/Intranet, con el fin de visualizar, consultar y analizar información geográfica a través de la red, mediante la tecnología Internet Map Server (IMS).

7.1 Previo a la Instalación.

Al trabajar con mapas, usamos varios tipos de datos, ya sean estos, vectoriales o raster, para esto necesitamos instalar las librerías GDAL/ORG, las mismas que, forman una capa de abstracción de datos espaciales que acceden nativamente a los formatos GIS más utilizados. Para instalar estos datos debemos descargarnos las fuentes del siguiente sitio web ftp://ftp.remotesensing.org/gdal y continuaremos los pasos señalados en los capítulos anteriores.

- Descomprimimos las fuentes y las ubicamos en el directorio “/usr/local”.

  [root@localhost ~]# tar --xzG gdal-1.6.0.tar --C /usr/local/

- Nos ubicamos en el directorio /usr/local/gdal-1.6.0

  [root@localhost ~]# cd /usr/local/gdal-1.6.0

- Creamos el archivo de configuración con la ventana de comandos “vi” o un editor de textos cualquiera.

  [root@localhost gdal-1.6.0]# vi compile_gdal.sh

Contenido del archivo “compile_gdal.sh” (Figura 7.1).
Figura 7.1 Contenido del Archivo “compile_gdal.sh”.

- Damos permisos de ejecución al archivo “compile_gdal.sh”.

```
./configure
--with-png
--with-libtiff
--with-jpeg
--with-gif
--with-pg=/usr/local/pgsql/bin/pg_config
--with-mysql=/usr/local/mysql/bin/mysql_config
--with-geos
--with-odbc
```

Figura 7.2 Resultado del compile_gdal.sh

- Ejecutamos el archivo “compile_gdal.sh” (figura 7.2)
• Ejecutamos el comando `make` (figura 7.3) y `make install` (figura 7.4)

Para finalizar debemos recargar las librerías del sistema

```
[root@localhost gdal-1.6.0]# ldconfig
```
7.2 Instalación.

Primero procederemos a descargar los códigos fuente de este módulo, para ello nos dirigimos a la dirección http://www.postgis.org/download/ (figura 7.5).

Luego de haber descargado el código fuente tendremos que descomprimirlo, nosotros sugerimos la siguiente ruta “/usr/local/”, y volveremos a realizar los mismos pasos que se han venido haciendo en los capítulos anteriores.

- Descomprimimos las fuentes y las copiamos en el directorio “/usr/local”.

  [root@localhost ~]# tar -xzf mapserver-5.2.1.tar -C /usr/local/

- Nos ubicamos en el directorio “/usr/local/mapserver-5.2.1”.

  [root@localhost ~]# cd /usr/local/mapserver-5.2.1

Figura 7.5 Sitio Web para descargar MapServer
• Creamos el archivo de configuración con la ventana de comandos “vi” o un editor de textos cualquiera.

[root@localhost mapserver-5.2.1]# vi compile_map.sh

Contenido del archivo “compile_map” (Figura 7.6).

```bash
./configure \
  --with-proj=/usr/local \
  --with-geos=/usr/local/bin/geos-config \
  --with-ogr=/usr/local/bin/gdal-config \
  --with-gdal=/usr/local/bin/gdal-config \
  --with-postgis=/usr/local/lib/postgresql/bin/pg_config \
  --with-curl-config=/usr/bin/curl-config \
  --with-httpd=/usr/local/apache2/bin/httpd \
  --with-php=/usr/local/php-5.2.1 \
  --with-wfs \
  --with-wfs-client \
  --with-wms-client \
  --enable-debug \
  --with-curl-config=/usr/lib/curl/curl-config \
  --with-threads \
  --with-wcs \
  --with-wcs-client \
  --with-sos \# new in 4.10 \
  --with-gd \
  --with-freetype \
  --with-jpeg
```

Figura 7.6 Contenido del archivo “compile_map.sh”.

• Damos permisos de ejecución al archivo “compile_map.sh”.

[root@localhost gdal-1.6.0]# chmod 755 compile_map.sh

• Ejecutamos el archivo “compile_map.sh” (figura 7.7)
Ejecutamos el comando `make` (figura 7.8)

```
Figura 7.8 Resultado del comando `make` de `mapserver`
```
En este caso **no ejecutaremos** el comando *make install* ya que, colocaremos manualmente los ejecutables y las librerías en sus directorios respectivos, esto lo haremos con los siguientes comandos (figura 7.9).

```
[root@localhost mapserv-5.2.1]# cp mapserv /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp legend /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp scalebar /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp chp2img /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp shp2pdf /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp shptree /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp snptreest /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp shptreevis /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp sortshp /usr/local/apache2/cgi-bin/
[root@localhost mapserv-5.2.1]# cp tile4ms /usr/local/apache2/cgi-bin/
```

**Figura 7.9** Librerías cgi-bin de MapServer

7.3 Pruebas de Funcionamiento.

Después de realizar todos los pasos anteriores, comprobaremos que la instalación se haya realizado de forma correcta y lo haremos de la siguiente manera.

En la terminal de comandos digitaremos `./mapserv -v` (figura 7.10).

```
[root@localhost mapserv-5.2.1]# ./mapserv -v
MapServer version 5.2.1 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBM OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=GDAL SUPPORTS=GDAL2 SUPPORTS=GDAL3 SUPPORTS=GDALWMS SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER SUPPORTS=THREADS SUPPORTS=GEOGRAPHIC INPUT=EPSG INPUT=GAR INPUT=OGS INPUT=GEOM INPUT=SHAPEFILE
```

**Figura 7.10** Versión de Mapserver

Si podemos observar el contenido de la figura 7.8 entonces **MapServer** ha sido instalado de forma correcta, ahora procedamos a realizar la configuración, caso contrario verificar el archivo *compile_map.sh* y realizar nuevamente la compilación.

Si observamos el archivo de configuración de **MapServer** *compile_map.sh*, podremos observar la línea `--with-php=/usr/local/php-5.2.8`, con la cual le indicamos a **MapServer** que PHP se encuentra instalado en nuestro sistema, esto permite la creación
de un módulo para extender **PHP** con la **API de MapServer**, y así poder acceder a ella con toda su potencia y flexibilidad.

El módulo del que hablamos se encuentra en el directorio “/usr/local/mapserver-5.2.1/mapscript/php3” y lo tenemos que copiar al siguiente directorio “/usr/local/apache2/php/ext”.

Luego ejecutamos el siguiente comando.

```
[root@localhost ~]# cp /usr/local/mapserver-5.2.1/mapscript/php3/php_mapscript.so /usr/local/apache2/php/ext
```

Después procedemos a reiniciar el Servidor Apache con el siguiente comando.

```
[root@localhost ~]# service apachectl restart
```

Ahora editaremos el archivo “/usr/local/apache2/htdocs/index.php”.

```php
<?php
dl("php_mapscript.so");
phpinfo();
?>
```

Procedemos a consultar la información en la web (figura 7.11).
Figura 7.11 Información de MapScript

Para mayor información referirse al ANEXO IV.

7.4 Conclusión.

En este capítulo presentamos una guía de instalación y configuración personalizada del servidor de mapas en su última versión, la misma presenta ilustraciones con los resultados de los procesos realizados, de tal forma que se pueda seguir paso a paso la instalación, para evitar la probabilidad de que el usuario cometa error alguno en la instalación y configuración.
RECOMENDACIONES

A continuación enumeraremos algunas recomendaciones o sugerencias derivadas de las experiencias recolectadas en el transcurso de la elaboración de esta monografía.

- Recomendamos la utilización de software libre, para propósitos ya sean personales como laborables.

- Sugerimos utilizar este tutorial para configurar un servidor de mapas además de seguir paso a paso las instrucciones para tener éxito con la instalación y configuración de este software.

REFERENCIAS

Capítulo 1.


BIBLIOGRAFÍA

- http://drupalab.com/blog/compile-install-mysql-linux-centos
ANEXO I

Este Anexo fue extraído del documento “laboratorio_sistemas_operativos.doc” que es una guía de clases del Ingeniero Pablo Esquivel.

COMANDOS BASICOS DE CENTOS

Comando para ayuda.

`man [opciones...]` página

Muestra la documentación de un determinado comando (en realidad también de cualquier documentación, no sólo comandos).

Por ejemplo “man ls” nos dará todas las opciones del comando ls.

Cada documento se denomina “página”, las páginas están divididas por “secciones”.

A veces existen páginas en secciones distintas con el mismo nombre, para especificar la sección se usa el número de ella como opción.

Por ejemplo “man 3 printf”.

“man -a printf” mostrará todas las páginas, en sucesión, de printf en todas las secciones.

Si se quiere buscar.

“man -k printf” mostrará un listado resumido de todas las páginas donde aparezca printf en la descripción corta. El número mostrado entre paréntesis es la sección.

Se puede indicar que muestra la documentación en otros idiomas (si están instalados en el sistema).

Por ejemplo:

“man -L es ...” en castellano

“man -L en ...” en inglés
Comandos para manejo de archivos o directorios.

**ls [opciones...] [directorio/fichero ...]**

Lista el contenido del directorio, sin argumentos lista el contenido del directorio actual de trabajo. La opción más habitual es “-l” que muestra información más completa de cada directorio y fichero. La opción “-R” hace un listado recursivo en la jerarquía de directorios.

- `ls –l` Listado largo
- `ls –a` Listado de ficheros ocultos
- `ls –la` Listado largo con ficheros ocultos
- `ls –l ejemplo` Listado largo de los archivos del directorio ejemplo

**pwd**

Imprime el directorio actual de trabajo.

Ejemplo: `pwd`

Retorna: `/home`

**cd [directorio]**

Cambia de directorio. Sin argumentos lleva al directorio del usuario (HOME). Si el directorio es “..” sube un nivel.

Ejemplo: `cd /home/User`

**mkdir [opciones...] directorio**

Crea el directorio con el nombre indicado.

Ejemplo: `mkdir ejemplo`

**vi [opciones] archivo**

Es un editor de texto.

Ejemplo: `vi ejemplo`
Para editar el archivo presionamos “i”, para salir del modo INSERTAR presione ESC.

Dentro del editor los siguientes comandos sirven para:

:q  Sale del editor sin grabar.
:wq  Graba el contenido y sale del editor.
:w!  Graba así no tenga permisos de escritura
:w   Esto graba el contenido del archivo.
:q   Sale del editor sin grabar.
:234 Va a la línea 234
:u   Deshace el último cambio
:x   Borra carácter bajo el cursor.
:dd Borra la línea queda guardado.
:a   Inserta después del cursor
:1G Comienzo del archivo
G    Fin del archivo
:I   Insertar al principio de la línea
:A   Insertar al final de la línea
/:cadena  Busca la cadena
:yy Copia una línea
:P   Pega antes del cursor
:p   Pega después del cursor

rm [opcions...] ficheros
Borra ficheros. Las opciones más habituales son “-f” para forzar el borrado sin preguntar al usuario (la opción contraria es “-i”).

“-r” borra recursivamente todos los subdirectorios.

Ejemplo: `rm ejemplo`

`rmdir [opciones...] directorio...`

Borra un o varios directorios si están vacíos.

Ejemplo: `rmdir carpeta`

`mv [opciones...] fuente... destino`

Cambia el nombre de un fichero por otro o mueve una serie de ficheros y directorios a un directorio destino.

Ejemplo: `mv /home/User/ejemplo2/otro /home/User/ejemplo`

El ejemplo anterior mueve el archivo `otro` que está dentro del directorio `ejemplo2` al directorio `ejemplo`.

`cp [opciones...] fuente... destino`

Permite copiar un fichero, o varios ficheros a un directorio. Quizás la opción más usada es “-r” que permite copiar recursivamente directorios hacia otros directorios.

Ejemplo: `cp –r /home/User/ejemplo/directorio /home/User/ejemplo2`

El ejemplo anterior copia el directorio llamado “directorio” que está dentro del directorio `ejemplo` al directorio `ejemplo2`.

`mcopy [opciones...] archivo a:`

Copia archivos desde y hacia diskettes.

Ejemplo: `mcopy imagen1.jpg a:`

`cat [opciones...] [ficheros...]`

Muestra el contenido de los ficheros por la “salida estándar”. Si no se especifican ficheros, lee de la “entrada estándar”.
Ejemplo: cat prueba

**sort [opciones...] [ficheros...]**

Imprime la concatenación ordenada lexicográficamente de los ficheros o entrada estándar. Opción “-n” ordena numéricamente.

Opción “-r” en orden inverso.

“--field-separator=SEP” hace que SEP sea el separador de campos...

Ejemplo: sort -n prueba

**more [opciones...] [ficheros]**

Muestra el contenido de los ficheros o la entrada estándar página a página cada 25 líneas y espera que el usuario indique las acciones a tomar. Estas acciones se suelen indicar con una tecla, por ejemplo “<ESPACIO>” es para avanzar una página, “<ENTER>” avanza una línea. “h” da la ayuda, “?”, sirve para buscar una cadena, “q” para salir.

Ejemplo: more prueba

**ln [opciones] destino [nuevo_alias]**

Crea un enlace a un fichero, apuntará a los mismos datos que el fichero “destino”, siempre.

La opción “-s” hace que se cree un enlace simbólico, es lo que se conoce como “Acceso Directo” en Windows, o “Enlace” en la interfaz gráfica de Macintosh.

Ejemplo: mkdir /root/enlace

    ln -s /tmp /root/enlace

    rm /root/enlace (Borra enlace)

**wc [opciones...] [ficheros]**

Indica la cantidad de caracteres, palabras y líneas que tienen los ficheros. “-l” indica sólo número de líneas, “-w” palabras y “-c” los bytes, “-m” caracteres.
Ejemplo: \texttt{wc archivo.txt}

\begin{verbatim}
  1   2   6 archivo.txt
\end{verbatim}

\textbf{du [opciones…] [ficheros]}

Instrucción para ver el tamaño de archivos o carpetas.

Ejemplo: \texttt{du /var/spool/mail}

\begin{verbatim}
du –sh /var muestra el tamaño total del directorio
\end{verbatim}

\textbf{grep}

Busca cadenas dentro de archivos.

Ejemplo: \texttt{grep cadena *}

\begin{verbatim}
grep –RH cadena *
\end{verbatim}

-\texttt{R} busca en forma recursiva

-\texttt{H} muestra el nombre del archivo por cada coincidencia

También se puede utilizar para recuperar archivos de la siguiente forma.

\begin{verbatim}
grep -a -B2 -A200 "hola" /dev/hda1
\end{verbatim}

\textbf{stat}

Despliega información detallada sobre el archivo especificado como: fechas de modificación y cambio, dueño del archivo, etc.

Ejemplo: \texttt{stat archivo.txt}

\textbf{find}

Busca un archivo.

Ejemplo: \texttt{find / -name “nombre.txt” -print}

Para localizar los ficheros secundados (S para el usuario) podemos utilizar la orden

\begin{verbatim}
find / -perm -4000 -type f –print
\end{verbatim}
Mientras que para localizar los segundados (S para el grupo) podemos utilizar

```
find / -perm -2000 -type f –print
```

**tail**

Permite ver el final de un archivo, este comando es útil ya que los archivos de registros "logs" crecen constantemente. Para ver el final del archivo log podemos utilizar:

```
tail --f /var/log/messages
```

Este comando anterior despliega las últimas 15 líneas del archivo messages (el default es de 10). La --f mantiene el archivo abierto para poder observarlo conforme se agregan eventos.

**which**

Ve el path de cualquier programa o comando.

Ejemplo: `which awk`

```
/bin/awk
```

**Comandos para el manejo de usuarios y grupos.**

**useradd [opciones...] [LOGIN]**

Instrucción para crear un usuario.

Ejemplo: `useradd -c "Antonio Alonso Martinez" -d /home/aalonso -g mail -m aalonso`

**usermod [opciones...] [LOGIN]**

Modifica a un usuario.

Ejemplo: `usermod –g apache aalonso`

**userdel [opciones...] [LOGIN]**

Instrucción para borrar usuarios.

Ejemplo: `userdel –r aalonso`

**passwd**
Instrucción para cambiar de password a un usuario.

**finger**

Muestra información sobre el usuario.

Ejemplo: finger usuario

**groupadd**

Crea un nuevo grupo.

**groupdel**

Borra un grupo.

**gpasswd**

Asignación de usuarios existentes a grupos existentes.

Ejemplo: gpasswd -a usuario-que-sea grupo-que-sea

**chgrp**

Cambia el grupo al cual pertenece un archivo o directorio.

chgrp –R actúa en forma recursiva

**chown**

Cambia el usuario al cual pertenece un archivo o directorio.

chown –R actúa en forma recursiva

**history**

Lista los últimos comandos utilizados por el usuario.

En la carpeta del usuario el archivo donde se almacena es .bash_history

echo $HISTFILE variable donde se almacena el archivo que utiliza history.

**su**

Permite cambiarse de usuario sin salirse del usuario actual. Para salir del usuario se digita la palabra exit.

**sudo**
Permite ejecutar un comando como si fuera otro usuario. Ejemplo:

```bash
sudo –u root vi /etc/passwd
```

**who**

Muestra los usuarios de sistema que han iniciado una sesión.

**Comando para configurar permisos de acceso a los ficheros.**

**chmod**

Cambia los permisos de acceso de ficheros.

<table>
<thead>
<tr>
<th>Octal</th>
<th>Binario</th>
<th>Permisos</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>- - -</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>- - x</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>- w -</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>- w x</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>r - -</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>r – x</td>
</tr>
</tbody>
</table>
Ejemplo: chmod 751 texto.txt
Permisos: Usuario => 111 => r w x
Grupo => 101 => r - x
Otros => 001 => - - x

Clases de usuarios:

u => usuario propietario
g => grupo
o => otros

Cambiar permisos:

Modo absoluto
chmod 652 notas => r w - r - x - w -
Se ejecuta con los permisos del propietario del archivo.

chmod -s       Desactiva
chmod u+s      Usuario
chmod g+s      Grupo

Chmod 644 nombre_ archivo

# Hace que "nombre_ archivo" sea de lectura / escritura para el propietario, de lectura para los demás.

Comandos para el manejo del FILE SYSTEM.

Isattr

Lista atributos de file system ext3.
chattr

Modifica los atributos de file system ext3.

chattr +a fichero  a modo solo de añadir del fichero
chattr +Ss fichero  i no permite hacer cambios al fichero o borrarlo
chattr -sa fichero  -R actua en forma recursiva
                    s cuando se borra el archivo con atributo s el sistema
                    rellena con ceros el contenido del archivo
                    S hace que los cambios sobre el archivo se escriban
                    inmediatamente en el disco en lugar de esperar el sync del
                    sistema operativo.

Comandos para el manejo de procesos en el sistema.

ps

Muestra los procesos que se están ejecutando en el sistema.

Ejemplos:

  ps –aux

  Para encontrar la cantidad de procesos del sistema:

  ps -A | grep bash | wc -l (esto demostrará la cantidad de procesos).

  ps   muestra los procesos del usuario actual.

  ps –a  muestra los procesos de todos los usuarios.

  ps –A  muestra los procesos de todo el sistema incluido la de todos los usuarios.

  ps –x  muestra los procesos que no estén ligados a una tty.

  ps –l  muestra los procesos según su prioridad columna PRI el valor mas
          elevado de PRI es el que tiene menos prioridad.

kill

Elimina un proceso dándole el número de proceso.
Ejemplo: kill -9 8909

**Killall**

Elimina un proceso dándole el nombre.

Ejemplo: killall gateway

**Comandos para el manejo de puertos servicios de correo, servicios de red e internet.**

**Service**

Instrucción para arrancar, apagar o restart un servicio.

service httpd start

**Netstat**

Para ver el servicio ligado al puerto es:

Ejemplos: netstat –ltunp o sino tambien

    netstat –pel

    netstat –anp |grep 953

Usted puede encontrar la cantidad de conexiones a Apache con este comando:

netstat -nt | grep :80 | wc -l

l : muestra todos los puertos que están en modo listen.

t : muestra todos los tcp.

u : muestra todos los udp.

n : no resuelve nombres.

p : muestra el PID y el nombre asociado.

e : muestra información extendida.

a : los puertos que están esperando conexión.

netstat –i da la estadísticas de las interfaces.

netstat –ta muestra todas las conexiones activas.
Archivo donde están todos los puertos conocidos con el nombre del servicio
/etc/services.

**Ifconfig**

Configura la tarjeta de red.

Ejemplo:

ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up

ifconfig –a consulta la configuración actual.

Para cambiar dirección mac en Linux ath0=WiFi eth0=Ethernet

ifconfig eth0 down

ifconfig eth0 hw ether 00:11:22:33:44:55

ifconfig eth0 up

También se puede poner los gateways editando los siguientes archivos según las tarjetas
que tenga.

/etc/sysconfig/network-scripts/ifcfg-eth0

/etc/sysconfig/network-scripts/ifcfg-eth0:1

/etc/sysconfig/network-scripts/ifcfg-eth1

/etc/sysconfig/network-scripts/ifcfg-eth1:1

/etc/sysconfig/network-scripts/ifcfg-eth1:2

/etc/sysconfig/network-scripts/ifcfg-eth1:3

Ejemplo del contenido de un archivo ifcfg-eth0

GATEWAY=192.188.47.3

BOOTPROTO=none

TYPE=Ethernet

HWADDR=00:0D:60:EB:BF:AA

DEVICE=eth0
NETMASK=255.255.255.0
BROADCAST=192.188.47.255
IPADDR=192.188.47.2
NETWORK=192.188.47.0
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes

**Ifup**
Habilita la interfase especificada.
Ejemplo: ifup eth0

**Ifdown**
Deshabilita la interface especificada.
Ejemplo: ifdown eth0

**route**
Configura el gateway del equipo o las rutas del equipo.
Ejemplo: route muestra las rutas actuales
                      route add default gw 192.168.0.1 añade una ruta
                      route del default gw 192.168.0.1 borra una ruta

Para guiar toda la información de la red 206.171.55.16 netmask 255.255.255.240 vía la interfase eth0.

route add -net 206.171.55.16 255.255.255.240 eth0

**Ping**
Envía un paquete a un host y este le responde si esta activo y el tiempo que se demoro.
Ejemplo: ping -l 65000 127.0.0.1 -i 0
ping -l 65527  127.0.0.1

-i    Especifica cada cuántos segundos hace el ping el valor máximo es 255.


**Traceroute**


traceroute www.google.com

**nslookup**

Cuando un sitio Web no se puede visualizar, no tiene porque estar caído, puede ser que los servidores DNS que se este usando no estén funcionando correctamente para ese dominio. Se puede comprobar si un DNS resuelve bien la IP de un servidor mediante el comando llamado “nslookup“ que existe tanto en unix como en Windows.

nslookup www.google.com 157.100.1.2  el primer parámetro es el sitio Web que se quiere ver cual es la IP, el segundo parámetro es el servidor DNS a quien se le pregunta. Se puede entrar a modo interactivo digitando nslookup sin ninguna opción y allí se puede optar por preguntas mas específicas con el subcomandos set q=

Set q=a    Especifica la dirección IP un equipo.

Set q=ANY    Especifica todos los tipos de datos.

Set q=CNAME    Especifica un nombre canónico para un alias.

Set q=MX    Especifica el intercambiador de correo.

Set q=TXT    Especifica la información de texto.
Set q=ns  Especifica registros de nombres de servidores (NS).

Con el subcomando Server se especifica el servidor al cual se quiere preguntar ejemplo Server 157.100.1.2. Para salir del modo interactivo se escribe la instrucción exit.

Ejercicio para ver cómo funcionan los DNS resolviendo el dominio uazuay.edu.ec

nslookup

server c.root-servers.net.  (Pregunta al servidor raíz)

set q=ns

ec.

server dns1.nic.ec

edu.ec.

server dns2.nic.ec

uazuay.edu.ec.

server gye2.satnet.net.

set q=any

uazuay.edu.ec

Una actualización del archivo de los servidores raíz se la encuentra en ftp://ftp.internic.net/domain/named.cache este archivo se lo coloca en /var/named/named.ca

Dig

Igual que nslookup.

Host

Igual que nslookup.

mail

Envía un correo electrónico.

mail jleon@yahoo.com
Subject: Asunto

Cuerpo del mensaje

.   --> para salir se pone punto y se da un enter

Cc: copias

mail –v jleon@yahoo.com (muestra detalles de cómo resuelve el mail).

mail jleon@yahoo.com < archivo.txt envía un archivo por mail.

mail si se ejecuta solo el comando mail este lee el archivo de mails

   los comandos más usados en este entorno son:

h    lista los mail.

h60  comienza el listado de mails desde el mail 60.

60   lee el mail 60.

z    pasa a la siguiente página.

d1   borra el mail 1.

d1-10 borra los mail desde el 1 hasta el 10.

x    sale grabando los cambios.

q    sale sin grabar los cambios.

mailq

Muestra los mails encolados.

ftp

Cliente para la transferencia de archivos.

Ejemplo: ftp 162.168.0.1

Comandos más utilizados.

ascii para transferencia en modo ASCII.

binary para transferencia en modo binario.

dir  para ver el contenido de una carpeta.
get transfiere un archivo de la maquina remota a la local.

mget transfiere un varios archivos (*) de la maquina remota a la local.

mput transfiere un varios archivos (*) de la maquina local a la remota.

mkdir crea un directorio en la maquina remota.

put transfiere un archivo de la maquina local a la remota.

pwd muestra el path actual.

quit sale.

telnet

Cliente para conexión remota.

Ejemplo: telnet 162.168.0.0.1

Ssh

Cliente para conexión remota encriptada.

Ejemplo: ssh jleon@168.0.0.1

sftp

Cliente para conexión remota de ftp encriptada.

Ejemplo: sftp jleon@168.0.0.1

hostname

Muestra o cambia el nombre del equipo.

hostname para mostrar el nombre del equipo.

hostname otro nombre para cambiar el nombre del equipo.

nmap

Herramienta para exploración de red y scanner de seguridad. El archivo donde se pueden encontrar los servicios conocidos y sus puertos son /etc/services

Modo detallado

nmap –v 127.0.0.1
Lanza un sondeo de tipo SYN (envía un paquete como si fuera una conexión real y espera la respuesta) sigiloso contra cada una de las 255 máquinas en la “clase C” de la red donde está el sistema "scanme.namp.org". También intenta determinar cuál es el sistema operativo que se ejecuta en cada máquina que esté encendida (Opción O).

```
nmap -sS -O scanme.nmap.org/24
```

Ve la versión del servicio (-sV) que se está ejecutando en los puertos (-p) 22, 53, 110, 143, 4564 (22 sshd, 53 DNS, 110 pop3, 143 imap) desde la red 198.116.0 hasta la red 198.116.255 pero solo las 127 primeras direcciones ip.

```
nmap -sV -p 22,53,110,143,4564 198.116.0-255.1-127
```

Analiza la red 216.163.128.20/20 (4096 ips) sin enviar ping (-P0) para descubrir si está activo el equipo y los resultados los graba en formato xml (-oX) y también en formato txt (-oG).

```
nmap -P0 -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20
```

Para conocer el sistema operativo (-O) que se está ejecutando en 127.0.0.1
```
nmap -O 127.0.0.1
```

Para ver cuáles hosts están activos en la red 192.168.0.0 mediante ping (-sP)
```
nmap -sP 192.168.0.1-255
```

**iptraf**

Muestra en aplicación de consola la cual analiza todo el tráfico de red IP, UDP, ICMP. Permite utilizar filtros, y es muy útil para diagnóstico y depuración de errores de red.

**Tcpdump**
Herramienta para análisis de tráfico de red.

**wget**

wget es una herramienta de Software Libre que permite la descarga de contenidos desde servidores web de una forma simple. Su nombre deriva de «World Wide Web» (w), y de «obtener» (get), esto quiere decir: obtener desde WWW. Actualmente soporta descargas mediante los protocolos HTTP, HTTPS y FTP.

wget http://www.mat.univie.ac.at/~flo/linux/dsniff-2.4b1-11.i386.rpm (programa que las claves de los accesos al servidor).


**chkconfig**

chkconfig sendmail off

chkconfig --level 2345 MailScanner on

chkconfig --list sendmail


**lynx**

Navegador de Texto el gráfico es htmlview.

lynk www.google.com

**Comandos para el manejo del disco duro.**

**df**

Muestra el espacio en disco disponible.

Ejemplo: df, df –h

-h añade un letra indicativa para el tamaño.
Si no se pone ninguna opción las unidades son de 1024 bytes.

**fdisk**

Crea tabla de particiones.

fdisk –l para ver las particiones.

fdisk /dev/hda para particionar el primer disco IDE.

**mount**

Monta unidades de disco duro, diskette, cdrom.

mount /dev/hda /media/cdrom

El archivo del sistema donde están las unidades que se montan cuando se inicia el servidor es /etc/fstab

Para montar un archivo iso en la carpeta /centos para ver o copiar su contenido

mount -t iso9660 -o ro,loop=/dev/loop0 /var/CentOS-5.0-i386-bin-1of6.iso /centos

Para montar un USB

mount -t vfat /dev/sda /usb

**umount**

Desmonta unidades.

Ejemplo: umount /dev/hda

**fsck**

File system check es una herramienta que revisa el disco duro y repara la estructura de ficheros dañada. Para revisar la estructura de un disco primero hay que desmontarlo.

fsck –y /dev/hda1

fsck –y /dev/sda1

La opción –y indica a fsck que responda "sí" a todas sus preguntas sobre arreglos, reparaciones o copias de seguridad de la información.
**hdparm**

Ve el rendimiento del disco duro.

Ejemplo: 

```
hdparm -tT /dev/hda
```

```
hdparm -tT /dev/sda3
```

```
hdparm -tT /dev/sda1
```

-T para ver los tiempos de lectura del cache.

-t para ver los tiempos de lectura del disco.

**badblocks**

Descubre los sectores malos de un disco y los graba a un archivo ejemplo:

```
badblocks -v /dev/hda1 > bad luego se puede formatear el disco indicándole cuales son los sectores malos de la siguiente manera:
```

```
mkfs.vfat -F 32 -l bad /dev/hda1  para fat32
```

```
mkfs.ext3 -l bad /dev/hda1 para ext3
```

También se pueden ver los sectores malos con fsck -f /dev/hda1

**Comandos para el empaquetar o comprimir archivos.**

**tar**

El programa tar es usado para almacenar archivos y directorios en un solo archivo que por lo general tiene la extensión tar. Si utiliza ampliamente en el respaldo de archivos.

Instrucciones para empaquetar y desempaquetar.

```
tar -cvf nombre_del_archivo.tar directorio
```

```
tar –xvf nombre_del_archivo.tar
```

Si se quiere hacer con gzip para empaquetarlo y comprimirlo habría que poner:

```
tar cfvz nombre_del_archivo.tar.gz directorio
```

Ahora para desempaquetarlo y descomprimirlo se haría de la siguiente forma:

```
tar- xfvz nombre_del_archivo.tar.gz
```
Ahora para hacer lo mismo pero comprimiéndolo con bzip2 habría que poner:

`tar -jfvC nombre_del_archivo.tar.bz2 directorio`

Ahora para desempaquetarlo y descomprimirlo se pondría:

`tar -jfvx nombre_del_archivo.tar.bz2`

**gzip**

Comprime archivos.

Ejemplo: `gzip install.log` producirá un archivo llamado `install.log.gz`

**gunzip**

Desempaqueta paquetes en formato gz.

Ejemplo: `gunzip install.log.gz`

**unzip**

Desempaqueta paquetes en formato zip.

Ejemplo: `unzip install.zip`

**Comandos para el manejo de fecha y hora del sistema.**

**uptime**

Muestra la hora actual, tiempo que lleva el sistema corriendo desde el último "reboot", usuarios conectados al servidor, carga del sistema en los últimos 1,5 y 15 minutos.

**date**

Muestra o configura la fecha y hora del sistema.

`vie sep 1 11:13:24 ECT 2006`

`date 010102022005 el formato es MMDDhhmmYYYY`

**Comandos para la configuración del sistema.**

**set**

Muestra todas las variables de entorno y sus valores.

JAVA="/etc/jdsk"
export JAVA

JAVA_HOME="/usr/java/jdk1.5.0_04"

export JAVA_HOME

init

Instrucción que vuelve a leer los parámetros que se encuentran en /etc/inittab.

init 0  apaga el equipo

Niveles en Linux

Archivo /etc/inittab

#  0 – apaga el equipo
#  1 – modo monousuario
#  2 – modo multiusuario sin NFS sin red
#  3 – modo multiusuario con red
#  4 – no usado
#  5 - X11  ambiente gráfico
#  6 – reboot del equipo

ntsysv

Ambiente para manejo de los programas que se quieren cargar cuando se inicia el equipo.

Para cargar algo cuando arranque el servidor lo que se quiere que arranque se pone en el archivo /etc/rc.d/rc.local

setup

Ambiente para configuración del equipo.

Free

Ve la memoria Libre.
free –m
-m para que muestre en megas

**Top**
Monitorea el sistema.

**vmstat**
Es muy similar a top ya que es un condensado de los procesos del sistema, para que esta herramienta se vuelva dinámica se deben especificar los argumentos: vmstat -n <numero de segundos por actualización> vmstat -n 1

**uname**
Muestra información del sistema.
uname –a muestra toda la información del sistema

Linux uazuay 2.6.9-34.ELsmp #1 SMP Wed Mar 8 00:27:03 CST 2006 i686 i386
GNU/Linux.

**reset**
Reinicia el equipo.

**poweroff**
Apaga el equipo.

**Comandos para el manejo de paquetes.**

**rpm**
Package Manage originalmente llamado Red Hat Package Manager es una herramienta de administración de paquetes pensada básicamente para Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas. RPM es el formato de paquete de partida del Linux Standard Base.
rpm –e sendmail elimina el paquete sendmail.
rpm -q sendmail pregunta por el paquete sendmail.
rpm –qa lista los paquetes instalados.
rpm -ivh bindd.rpm instala el paquete bind.
rpm –Uvh bindd.rpm actualiza el paquete bind.
rpm --test -i bind.rpm realiza un test para aver si puede instalar el paquete.
rpm –ql sendmail lista el contenido del paquete sendmail.
rpm -qf /bin/ls muestra que paquete instalo el comando ls.
rpm –V sendmail verifica paquete.
i instala.
U actualiza.
e burre.
q query.
v muestra información de progreso de instalación.
h muestra información más detallada se usa con v.
V verifica un paquete.

2.15 Comandos para el manejo de parches.

Diff

Busca diferencias entre dos archivos. Se lo utiliza comúnmente para generar parches para los programas Ejemplo:

diff -Naur archivo_original archivo_cambiado > parche.diff

patch

Aplica parches generados con diff a un programa fuente. Ejemplo:

patch archivo_a_parchear parche.diff
ANEXO II

PgAdmin III es una aplicación gráfica para administrar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está desarrollada en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se puedan usar en Linux, FreeBSD, Solaris, Mac OS X y Windows.

Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I además de otras herramientas útiles dentro de las bases de datos. La conexión al servidor puede hacerse mediante conexión TCP/IP (Tranfer Control Protocol/Internet Protocol) o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad.

Instalación
Luego de haber culminado la instalación del gestor de base de datos *postgresql* procedemos a instalar la herramienta *pgAdmin*.

Lo primero que haremos es desinstalar todo lo vinculado a *pgAdmin*, esto lo haremos con el comando *yum*:

```
[root@localhost ~]# yum -y remove pgadmin3
```

Ahora procedemos a instalar las dependencias de “*pgAdmin3*”, de igual manera lo haremos con el comando *yum*:

```
[root@localhost ~]# yum install wxGTK libxslt mesa-libGLU SDL
```

Ahora descargamos e instalamos el *MSPack*.

```
[root@localhost ~]# wget ftp://ftp.pbone.net/mirror/ftp.pramberger.at/systems/linux/contrib/rhel5/i386/libmspack-0-0.2.200609-20alpha.el5.pp.i386.rpm
[root@localhost ~]# rpm -ivh libmspack-0-0.2.200609-20alpha.el5.pp.i386.rpm
```

Ahora descargamos e instalamos *wxGTK* para i386.

```
[root@localhost ~]# wget ftp://ftp.pbone.net/mirror/ftp.pramberger.at/systems/linux/contrib/rhel5/i386/wxGTK-2.8.8-1.el5.pp.i386.rpm
[root@localhost ~]# rpm -ivg wxGTK-2.8.8-1.el5.pp.i386.rpm
```

libs-8.3.5-1PGDG.rhel5.i386.rpm.html, luego procederemos a instalarlas con el siguiente comando:

```
[root@localhost ~]# rpm –ivh postgresql-libs-8.3.5-1PGDG.rhel5.i386.rpm
```

Descargamos los paquetes “rpm” correspondientes a pgAdmin3 para CentOS 5 en la siguiente página web http://rpm.pbone.net/ nosotros hemos descargado los siguientes:

pgadmin3-docs-1.8.4-2.el5.kb.i386
pgadmin3-1.8.4-2.el5.kb.i386

Y procedemos a instalarlos de la siguiente manera:

```
[root@localhost ~]# rpm –ivh pgadmin3-1.8.4-2.el5.kb.i386.rpm
[root@localhost ~]# rpm –ivh pgadmin3-docs-1.8.4-2.el5.kb.i386.rpm
```

Luego de instalar los módulos “rpm” procederemos a abrir el pgAdmin, esto lo hacemos dirigiéndonos al menú principal de CentOS. Nos digimos a “Aplicaciones” luego “programas” y ahí buscamos “pgAdmin3” damos clic y nos mostrará una ventana como la siguiente:
Ventana de la herramienta pgAdmin III

Damos doble clic en el botón conectar y nos mostrará una ventana que cargaremos de la siguiente manera:

Ventana de conexión a la base de datos
Damos clic en “Aceptar” y si nos muestra la ventana que está en la siguiente figura, la conexión fue un éxito, caso contrario, debemos iniciar el servicio de postgres que ya lo explicamos en el capítulo III\(^1\).

\[\text{Ventana del Administrador pgAdmin conectado al PostgreSQL}\]

Una vez conectados, podemos navegar sobre esta herramienta creando bases de datos, tablas, registros, backups, crear usuarios, asignar permisos, etc...

\(^1\) Capítulo 3, página 35, párrafo 3.4
**ANEXO III**

*PhpMyAdmin* es una herramienta escrita en *PHP*, con la intención de manejar la administración del gestor de la base de datos *MySQL* a través de páginas *web*, utilizando *Internet*. Actualmente puede crear y eliminar Bases de Datos. Crear, eliminar y alterar tablas. Borrar, editar y añadir campos, ejecutar cualquier sentencia *SQL*, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. Se encuentra disponible bajo la licencia *GPL*.

**Instalación y configuración:**

Lo primero que tenemos que realizar será de descargar el paquete “*phpMyAdmin*” esto lo haremos desde el sitio web de la herramienta, [http://www.phpmyadmin.net/](http://www.phpmyadmin.net/), en nuestro caso descargamos el paquete *phpMyAdmin-3.1.4-rc2-all-languages.tar*, luego se procedió a descomprimir el paquete y los haremos de la siguiente manera:

```
[root@localhost ~]# tar -xzf phpMyAdmin-3.1.4-rc2-all-languages.tar.gz -C /usr/local/apache2/htdocs/
[root@localhost ~]# mv /usr/local/apahce2/htdocs/phpMyAdmin-3.1.4-rc2-all-languages /usr/local/apache2/htdocs/phpMyAdmin
```

Una vez realizado este proceso nos ubicamos en el directorio “*/usr/local/apache2/htdocs/phpMyAdmin*” y luego copiamos y renombramos el
archivo de configuración, ya que, el paquete trae un archivo ejemplo para la
configuración.

```
[root@localhost ~]# cd /usr/local/apache2/htdocs/phpMyAdmin
[root@localhost ~]# cp config.sample.inc.php config.inc.php
```
Luego de esto modificamos el archivo “config.inc.php”, esto lo podemos realizar con
la ventana de comandos “vi” o con cualquier editor de texto.

```
[root@localhost ~]# vi config.inc.php
```
Luego Buscamos la siguiente línea:

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```
Y la editamos de la siguiente manera

```
$cfg['Servers'][$i]['auth_type'] = 'http';
```
Luego de esto reiniciamos el servicio Apache:

```
[root@localhost ~]# service apachectl restart
```
Para verificar que esté funcionando el “phpMyAdmin” abrimos un navegador de
internet y digitamos lo siguiente `localhost/phpMyAdmin` y nos mostrará lo siguiente:

Ventana para el ingreso a phpMyAdmin

En esta ventana digitaremos el usuario que controla MySQL, en nuestro caso el usuario
es “root” y procedemos a digitar la contraseña, si no accede tendremos que realizar el
siguiente paso desde una terminal de comandos:
En este caso la contraseña será **pin888**, volvemos a intentar colocando el usuario y contraseña, en este caso “**root**” y “**pin888**”, nos presentará la siguiente página web.

**Página web phpMyAdmin**

**PhpPgAdmin** es una aplicación web que provee una manera conveniente a los usuarios para crear bases de datos, tablas, modificarlas y consultar sus datos usando el lenguaje estándar **SQL**.
**PhpPgAdmin** estuvo basado en **PhpMyAdmin**, pero hoy día ya no comparte código con él; incluso provee las mismas funcionalidades y más a los usuarios del servidor de base de datos **PostgreSQL**.

### Instalación y configuración:

Lo primero que tenemos que realizar será descargarnos el paquete “**phpMyAdmin**” esto lo haremos desde el sitio web de la herramienta, [http://phppgadmin.sourceforge.net/](http://phppgadmin.sourceforge.net/) en nuestro caso descargamos el paquete **phpPgAdmin-4.2.2.tar**, después de esto procederemos a descomprimir el paquete y los haremos de la siguiente manera:

```
[root@localhost ~]# tar -xzf phpPgAdmin-4.2.2.tar.gz -C /usr/local/apache2/htdocs/
[root@localhost ~]# mv /usr/local/apache2/htdocs/phpPgAdmin-4.2.2 /usr/local/apache2/htdocs/phpPgAdmin/
```

Una vez realizado este proceso nos ubicamos en el directorio “/usr/local/apache2/htdocs/phpPgAdmin/conf/” para luego modificar el archivo “**config.inc.php**”, esto lo podemos realizar con la ventana de comandos “**vi**” o con cualquier editor de texto.

```
[root@localhost ~]# cd /usr/local/apache2/htdocs/phpPgAdmin/conf/
[root@localhost ~]# vi config.inc.php
```

Luego buscamos la siguiente línea:

```
$conf['servers'][0]['host'] = '';
```

Y la editamos de la siguiente manera

```
$conf['servers'][0]['host'] = 'localhost';
```
Luego buscamos la línea:

```plaintext
$conf['servers'][0]['pg_dump_path'] = '/usr/bin/pg_dump';
```

Y la editamos de la siguiente manera

```plaintext
$conf['servers'][0]['pg_dump_path'] = '/usr/local/pgsql/bin/pg_dump';
```

Luego buscamos la línea:

```plaintext
$conf['servers'][0]['pg_dumpall_path'] = '/usr/bin/pg_dumpall';
```

Y la editamos de la siguiente manera

```plaintext
$conf['servers'][0]['pg_dumpall_path'] = '/usr/local/pgsql/bin/pg_dumpall';
```

Y por último buscamos la línea:

```plaintext
$conf['extra_login_security'] = false;
```

Y la editamos de la siguiente manera

```plaintext
$conf['extra_login_security'] = true;
```

Para verificar que esté funcionando el “phpPgAdmin” abrimos un navegador de internet y digitamos lo siguiente `localhost/phpPgAdmin` y nos presentará lo siguiente:

![Página web phpPgAdmin](image)

Aquí daremos clic en el texto que dice **PostgreSQL** de la parte izquierda de la página, y nos mostrará lo siguiente:
Ventana de login de usuario

En esta ventana digitaremos el usuario que administra PosgreSQL en nuestro caso el usuario es “postgres” y no digitamos ninguna contraseña y nos mostrará lo siguiente

Ventana del administrador PhpPgAdmin
ANEXO IV

PRACTICAS CON CLIENTES LIGEROS PARA LA PUBLICACION DE MAPAS.

Para tener conocimiento de los pasos que se deben seguir para obtener el archivo necesario para publicar un mapa en la web pedimos revisar los capítulos 3, 4 y 5 del documento monográfico de los Ingenieros Diego Francisco Pacheco Prado y Chester Andrew Sellers Walden (ACTUALIZACIÓN Y DEPURACIÓN DE LA NUEVA CARTOGRAFÍA EXISTENTE PARA LA CUENCA DEL RÍO PAUTE”), el cual tenemos a disposición en la biblioteca de la Universidad del Azuay.

Nosotros hemos seguido los pasos basándonos en el documento anteriormente mencionado y hemos obtenido y configurado el documento “RutaIglesias12.map”, lo importante sería mencionar las opciones para la petición de datos tipo shape dentro de nuestro archivo “.map”.

Ejemplo de una petición de datos dentro de un archivo con extensión “.map” sin conexión a base de datos.

```
LAYER
   NAME 'Ciudades'
   GROUP 'Ciudades'
   DATA './shapes/Ciudades_CRP_25k_SAM56'
   PROJECTION
      "init=epsg:24877"
   END #end projection
   METADATA
      "queryable" "true"
      "ows_title" "Ciudades_CRP_25k_SAM56 selection"
      "ows_abstract" ""
      "ows_keywordlist" ""
      "wms_extent" "713269,1872 96725676,2828 733287,6854 9685483,7325"
      "wms_metadataurl_type" "TC211"
      "wms_dataurl_format" "text/html"
      "wms_dataurl_href"
```
Ejemplo de una petición de datos dentro de un archivo con extensión `.map` con conexión a una base de datos.

LAYER
   NAME 'Ciudades'
   GROUP 'Ciudades'
   CONNECTIONTYPE postgis
   CONNECTION "host=localhost dbname=iglesias user=postgres password=pin888 port=5432"
   DATA "the_geom from ciudades"
   PROJECTION
   "init=epsg:24877"
END #end projection
METADATA
   "queryable" "true"
   "ows_title" "Ciudades_CRP_25k_SAM56 selection"
   "ows_abstract" ""
   "ows_keywordlist" ""
   "wms_extent" "713269,1872 9675676,2828 733287,6854 9685483,7325"
   "wms_metadataurl_type" "TC211"
   "wms_dataurl_format" "text/html"
   "wms_dataurl_href"
"http://www.yourorganization.org/geonetwork/srv/en/metadata.show?id="
   "wms_style_default_title" "default"
"wms_format" "image/png"
"ows_srs" "EPSG:24877"
"wms_attribution_onlineresource" "http://www.yourorganization.org/"
"wms_attribution_title" "Data from Your Organization"
"wms_attribution_logourl_width" "20"
"wms_attribution_logourl_height" "20"
"wms_attribution_logourl_format" "image/jpg"
"wms_attribution_logourl_href" "http://www.yourorganization.org/geonetwork/images/very_small_logo.jpg"
END #end metadata

TYPE polygon
STATUS ON
TOLERANCE 8 #default is 3 for raster, 0 for vector
#TOLERANCEUNITS meters #default is meters, [pixels|feet|inches|kilometers|meters|miles|dd]
TEMPLATE "query.html"
CLASS
   #ANTIALIAS false
   COLOR 211 255 190
   BACKGROUNDCOLOR 211 255 190 # not sure about this one
   OUTLINECOLOR 0 0 0
END #end style
END #end layer
MSCROSS

Visualización de mapas desde el cliente ligero Mscross, utilizamos el siguiente archivo con extensión "html", el cual lo llamamos "index.html".

```
<html>
<head>
  <title>Practica Uno</title>
  <script src="js/mscross.js" type="text/javascript"></script>
  <style type="text/css">
    .Estilo1 {
      font-size: 16px;
      font-weight: bold;
    }
  </style>
</head>
<body bgcolor="#990000" bottommargin="0" topmargin="0" leftmargin="0" rightmargin="0">
  <table width="700" height="500" border="0" align="center" cellpadding="0" cellspacing="0"
    vspace="0">
    <tr><td>&nbsp;</td></tr>
    <tr><td><div style="width: 600px; height: 500px; background-color:#FFFFFF" id="map_tag"
      onmousemove="myMap1.ShowCoordinates(event)"></div>
      <font size=2>
        <div id="coordinates1" style="background-color:#FFFFCC">X:</div>
        <div id="coordinates2" style="background-color:#FFFFCC">Y:</div>
      </font></td></tr>
  </table>
</body>
</html>
```
Al momento de ejecutar se mostrara así:
OPEN LAYERS

Es muy importante mencionar que, para el uso del cliente OpenLayers necesitamos editar el archivo de configuración del Apache, para indicarle donde está ubicado el archivo con extensión “.map”, como referencia ver la siguiente figura.
Archivo de configuración del Apache

Para visualizar los mapas desde los clientes ligeros de tipo OpenLayers, utilizamos el siguiente archivo con extensión “html”, el cual lo llamamos “index.html”.

```html
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <style type="text/css">
        #map {
            width: 650px;
            height: 512px;
            border: 1px solid black;
            background-color: white;
        }
    </style>
</head>
<body>
    <link rel="stylesheet" href="/theme/default/style.css" type="text/css" />
    <link rel="stylesheet" href="style.css" type="text/css" />
    <script src="/lib/OpenLayers.js"></script>
    <script type="text/javascript">
        var lon = -79.0032;
        var lat = -2.8972;
        var zoom = -20;
        var map, layer, layer1,layer2,layer3,layer4,layer5,layer6;
        function init() {
            var map = new OpenLayers.Map('map',{projection:
                "EPSG:24877",height: 400, width:400,minResolution: 5, maxResolution: 30,units:"m",maxExtent: new OpenLayers.Bounds(714000,9674500,733140,9686500)));
            var ol_wms = new OpenLayers.Layer.WMS(
```
Al momento de ejecutar se presentará así:
Ruta de las Iglesias

Cliente OpenLayers