



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“MANUAL DE INSTALACIÓN PARA IMPLEMENTAR UN
SERVIDOR DE DESARROLLO SOBRE PLATAFORMA
OPENSOURCE”**

**P R E S E N T A:
IVÁN SEBASTIÁN ORTEGA HUIXTLACATL**

**TRABAJO MONOGRÁFICO
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

ASESOR: ING. SILVIA VEGA MUYTOY

**NEZAHUALCÓYOTL, ESTADO DE MÉXICO
2010**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

A mi madre, por la enseñanza que me dejó, de nunca rendirme ante nada, no importando que tan grande sea el obstáculo, y por la meta que dejó de que todos sus hijos se convirtieran en profesionistas la cual hoy cumpla al ser el último en titularse como ingeniero.

A mi padre, por el ejemplo de fortaleza que me ha dado, y por la confianza que depositó en mí, al dejar que trazara yo solo mi camino, el cual forje siguiendo sus pasos, para llegar a convertirme en un hombre de bien ante la sociedad.

A mi hermana Sandra, por su apoyo en todos los sentidos, que aunque no era su responsabilidad, me cuidó y vio por mí como si se tratara de uno de sus hijos.

A mis hermanos, por sus consejos y apoyo moral que me dieron.

A mi amigo de la universidad Alejandro Alva, por haber compartido su amistad a lo largo de los 5 años de la carrera. Durante los cuales compartimos muchos momentos de estrés pero también de alegría.

Índice

Introducción.....	1
Capítulo 1 Instalación de librerías y programas base necesarios.....	4
1.1 Librerías y aplicaciones de compilación básicas.....	4
1.2 Herramientas y bibliotecas complementarias de compilación de código fuente.....	8
1.3 Herramientas de configuración, monitoreo y comunicación remota.....	26
Capítulo 2 Manejadores de Bases de Datos.....	28
2.1 MySQL.....	28
2.2 POSTGRES.....	32
Capítulo 3 Servidores Web.....	38
3.1 CHEROKEE.....	38
3.2 APACHE.....	43
Capítulo 4 Lenguajes de programación para aplicaciones Desktop y Web.....	51
4.1 JAVA.....	51
4.2 TOMCAT.....	57
4.3 PHP.....	64
4.4 RUBY.....	77
Capítulo 5 Aplicaciones utilitarias para desarrollo.....	83
5.1 Eclipse como IDE de desarrollo.....	83
5.2 SUBVERSION.....	86
5.3 DBDesigner 4.....	89
5.4 Webmin.....	92
5.5 Terminator.....	96
Conclusiones.....	99
Glosario de comandos.....	101
Referencias	104

Introducción

El presente trabajo monográfico tiene como objetivo mostrar el proceso de implementación de un servidor de desarrollo utilizando sólo software bajo licencia OpenSource. La selección de los programas que serán instalados a lo largo de este manual y los flujos de instalación de los mismos, se Realizar en base a la experiencia obtenida durante mi estancia como becario en el plan de becas que se ofrece en la Subdirección de Sistemas de la UNAM y la estancia como apoyo en proyecto en la Subdirección de Servicios Web de la UNAM. Siendo la recopilación renovada durante el transcurso de la actividad laboral hasta llegar a la versión descrita en este manual. Existe más de una forma de realizar la instalación de los programas, la que se describe en este documento es la optima para tener el control sobre la instalación, configuración y posterior mantenimiento del equipo para ser actualizado de acuerdo a las necesidades requeridas en cierto momento.

Se eligió el sistema operativo GNU/Linux en su distribución Debian por su facilidad de instalación y simplicidad de uso, ya que no es complicado manejarlo pero aún así nos permite tener el control de todas las instalaciones, y no solicita que se realicen sólo mediante la utilización de asistentes de instalación (Wizard). Otra de sus ventajas es que otras distribuciones de GNU/Linux se basan en Debian por lo tanto si se requiere realizar la implementación de este manual en alguna de ellas se podrá realizar sin problema alguno. Esto porque la instalación se Realizar compilando los códigos fuentes de los programas y la instalación de las dependencias y librerías se Realizar mediante el uso de repositorios, los cuales existen para cualquier distribución diferente a Debian que se requiera.

El flujo de instalación parte con la premisa de que se cuenta con los conocimientos básicos sobre el uso y administración de Debian por parte del usuario. Debido a que las instrucciones que se tienen que ejecutar no son complejas y el flujo de instalación ha sido ejecutado más de una docena de veces en diferentes equipos. Los requerimientos básicos con los que debe contar el equipo de cómputo donde se implementara el servidor son los siguientes: Sistema operativo GNU/Linux en su distribución Debian 3.0 o superior, procesador Pentium III a 533 mhz., espacio en disco de 7 GB para la instalación del servidor, 512 MB en Ram. Los requerimientos óptimos para el mejor rendimiento durante el desarrollo de aplicaciones son: procesador a 2 Ghz , 15 GB de espacio en disco duro y 2 GB de memoria Ram.

En el primer capítulo se lleva a cabo la instalación de las librerías y dependencias necesarias para la compilación de los programas, lo cual permitirá no sólo la implementación de los programas propuestos, sino que estas dependencias base son necesarias en la gran mayoría de las aplicaciones desarrolladas sobre la plataforma GNU/Linux. Igualmente se instalan utilidades de configuración, monitoreo y comunicación.

En el segundo capítulo se Realizar la instalación de los manejadores de bases de datos más populares dentro de la comunidad de desarrolladores de software libre, MySQL presenta una buena opción para el desarrollo de aplicaciones web que requieren altos volúmenes de transacciones de consultas, un ejemplo de su utilización es el sitio web Facebook, por su parte Postgres ofrece una infraestructura más robusta para aplicaciones que demandan un mayor volumen de transacciones de consulta y modificaciones de los datos contenidos.

El Capítulo tres esta enfocado a la instalación de los servidores web, que por la tendencia en el desarrollo de nuevas tecnologías de información, están tomando un papel muy importante para el desarrollo de aplicaciones, ya que la tendencia es migrar de programas de escritorio a aplicaciones web y servicios web. El servidor Cherokee ofrece facilidad de instalación y configuración sin tener sacrificios en su desempeño y el servidor Apache ofrece una mejor infraestructura y robustez además de permitir una mayor posibilidad de configuración de seguridad.

El cuarto capítulo se refiere a la instalación de las aplicaciones e intérpretes necesarios para la utilización de los lenguajes de programación, el lenguaje JAVA y Ruby permite el desarrollo de aplicaciones web y de escritorio, por su parte PHP es el lenguaje más utilizado actualmente para el desarrollo de sitios web. El compilador del lenguaje C y C++ es instalado en el primer capítulo, esto debido a que es utilizado para la compilación y posterior instalación de los programas de este manual. Cabe recordar que el sistema operativo GNU/Linux está construido sobre estos dos últimos lenguajes, lo cual nos recuerda que aunque no son ya tan populares en el desarrollo de sistemas, esto no significa que no sean poderosos y muy completos.

En el quinto y último capítulo se propone la instalación de algunas aplicaciones que serán de gran utilidad para los desarrolladores principiantes y avanzados. Eclipse es un potente entorno de desarrollo integrado, el cual tiene la principal característica de poder ser utilizado para desarrollar aplicaciones en la gran mayoría de los lenguajes de programación existentes a través de las extensiones(plug-in) que se encuentran desarrolladas para él. Subversion es un controlador de versiones muy utilizado dentro

de la comunidad de desarrolladores de software libre que facilita el trabajo del manejo de las versiones de los archivos de un grupo de trabajo. DBDesigner4 es una aplicación que permite el diseño y modelado de bases de datos y se puede utilizar como interfaz gráfica para manejar las transacciones de MySQL. Webmin es una aplicación web que permite la administración del equipo a través de un navegador local o remotamente, esto facilita la administración del equipo a distancia. Terminator es un emulador de terminales cuya característica es poder abrir más de una terminal en la misma ventana.

Al terminar de ejecutar los flujos de instalación propuestos se tendrá un equipo con las aplicaciones necesarias para comenzar el trabajo de desarrollo de aplicaciones, el equipo esta listo para funcionar como servidor de un grupo de desarrolladores o puede ser utilizado como equipo de trabajo de un desarrollador. El principal objetivo por el cual inicie la recopilación de la información y se fue creando el manual, es para solucionar la necesidad que existe de realizar la instalación y mantenimiento de un equipo que cumple la función de servidor de desarrollo durante la vida laboral de un desarrollador promedio o que comienza su carrera en el desarrollo de software, encontrando en este manual las bases necesarias para iniciar sus labores de programador y permitiendo tener los conocimientos bases para posteriores configuraciones.

Capítulo 1 Instalación de librerías y programas base necesarios

Para realizar la instalación de los programas que se usaran en el servidor de desarrollo a través de la compilación de los archivos fuente, se requiere la instalación previa de paquetes de programas de compilación y librerías que ofrecen soporte durante la compilación de los programas, también se instalaran algunos paquetes que son requeridos para la compilación y funcionamiento de otros programas como por ejemplo *freetype* que es referenciado en la instalación del servidor web *Apache*. En caso de no ser instalados o no contar con los programas ya instalados, se producirán errores durante la ejecución de las instrucciones de instalación de los programas que se realizara en los siguientes capítulos.

1.1 Librerías y aplicaciones de compilación básicas

BYACC

Programa que Realizar la tarea de generación de analizadores sintácticos, que son la parte del compilador que Verificar la sintaxis y proporciona estructura a la entrada que se le esta enviando al compilador.

Tipo de licencia: dominio público.

Proceso de instalación

Dentro de una consola de comandos (Imagen 1.1) ejecuta la instrucción como usuario root.

```
server:~# apt-get install byacc
```



Imagen 1.1. Sintaxis de instalación de paquetes mediante el comando apt-get.

FLEX

Programa que Realizar la tarea de generación de escáneres que son programas que reconocen patrones léxicos en el texto generado por BYACC, el resultado de su operación es un archivo que contiene reglas en forma de pares de expresiones

regulares y código C, el archivo esta compilado y vinculado a la biblioteca –LFL para producir un ejecutable el cual al ser lanzado a ejecución Verificar la ocurrencia de las expresiones regulares contenidas y cada vez que se encuentra una coincidencia se Ejecutar el código C correspondiente.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install flex
```

LIBREADLINE5 y LIBREADLINE5-DEV

El paquete libreadline5 es una biblioteca de GNU la cual ayuda en la consistencia de interfaces de usuario, a los programas discretos que necesitan proporcionar una interfaz en línea de comandos. En esta librería también se encuentra incluida la biblioteca *history* de GNU la cual proporciona una interfaz de usuario consistente para rellamar a líneas ya escritas. El paquete libreadline5-dev contiene la documentación de readline en formato info, es utilizado con fines de desarrollo.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install libreadline5 libreadline5-dev
```

BISON

Programa generador de filtros de propósito general que convierte una descripción gramatical de contexto libre en un programa escrito en C para filtrar la gramática generada por BYACC.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install bison
```

G++

Es el compilador para el lenguaje de programación C++ utilizado en GNU, es utilizado en todas las distribuciones de GNU/Linux existentes.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install g++
```

LIBSTDC++5

Paquete que contiene una biblioteca de tiempo de ejecución adicional para programas realizados en el lenguaje C++ y compilados en el compilador G++ de GNU.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install libstdc++5
```

LIBSSL-DEV

Paquete que contiene librerías de desarrollo, archivos de cabecera y manuales de implementación, las librerías son utilizadas en programas como apache con modulo SSL.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install libssl-dev
```

LIBGTK2.0-DEV

Paquete que contiene las bibliotecas estáticas y los archivos cabecera necesarias para el desarrollo de aplicaciones GTK, que son un conjunto de herramientas multiplataforma para crear interfaces gráficas de usuario.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install libgtk2.0-dev
```

AUTOMAKE

Programa que sirve para generar automáticamente archivos del tipo *Makefile.in* a partir de archivos *Makefile.am*, los archivos *Makefile.am* son una serie de macros del comando *make*, los archivos *Makefile.in* generados cumplen con los estándares de Makefile de GNU.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install automake
```

AUTOCONF

Es un conjunto de macros que producen scripts de comandos de shell para configurar automáticamente paquetes de software a partir de su código fuente, los scripts generados pueden adaptarse a diferentes tipos de paquetes de sistemas tipo UNIX sin necesidad de intervención manual extra por parte del usuario.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install autoconf
```

PKG-CONFIG

Sistema para gestionar las opciones de compilación y enlace de las bibliotecas de los nuevos programas instalados a partir de su código fuente. Este sistema funciona en conjunto de *automake* y *autoconf*.

Tipo de licencia: OpenSource

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario root.

```
server:~# apt-get install pkg-config
```

1.2 Herramientas y bibliotecas complementarias de compilación de código fuente

ZLIB

Biblioteca de compresión de datos, provee una implementación del algoritmo DEFLATE que es usado en el programa de compresión gzip, inicialmente fue orientada para trabajar en conjunto con la biblioteca *LIBPNG*, a través de esta biblioteca se puede realizar la escritura y lectura de archivos comprimidos.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *zlib-1.2.3.tar.gz*

URL de descarga: <http://zlib.net/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root (Imagen 1.2), en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente de la biblioteca a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/fuentes/zlib-1.2.3.tar.gz /usr/local/src/
```

Ingresar a la ruta /usr/local/src.

```
usuario@server:~$ cd /usr/local/src
```



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$ cp /home/usuario/fuentes/zlib-1.2.3.tar.gz /usr/local/src/
usuario@server:~$ cd /usr/local/src
usuario@server:~/usr/local/src$ tar -zxvf zlib-1.2.3.tar.gz
usuario@server:~/usr/local/src$ cd zlib-1.2.3
usuario@server:~/usr/local/src/zlib-1.2.3$ ./configure
usuario@server:~/usr/local/src/zlib-1.2.3$ make test
usuario@server:~/usr/local/src/zlib-1.2.3$ make
usuario@server:~/usr/local/src/zlib-1.2.3$ su root
server:~/usr/local/src/zlib-1.2.3# make install
server:~/usr/local/src/zlib-1.2.3# exit
server:~/usr/local/src/zlib-1.2.3$ cd ..
```

Imagen 1.2. Ejecución del flujo de instalación de paquete vía compilación de código fuente.

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf zlib-1.2.3.tar.gz
```

Ingresa a la carpeta generada.

```
usuario@server: /usr/local/src$ cd zlib-1.2.3
```

Ejecutar el comando *configure*.

```
usuario@server:~/usr/local/src/zlib-1.2.3$ ./configure
```

Realizar una prueba de la configuración de instalación, Ejecutar el comando *make test*.

```
usuario@server:~/usr/local/src/zlib-1.2.3$ make test
```

Verificar que la prueba fue satisfactoria, el programa retornara el siguiente mensaje:

```
*** zlib test OK ***
```

Ejecutar el comando *make*.

```
usuario@server:~/usr/local/src/zlib-1.2.3$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:~/usr/local/src/zlib-1.2.3$ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/zlib-1.2.3# make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/zlib-1.2.3# exit
```

Regresar a la carpeta *src*.

```
server:/usr/local/src/zlib-1.2.3$ cd ..
```

GETTEXT

Paquete que ofrece a los programadores, traductores y usuarios, un conjunto integrado de herramientas y documentación que proporcionan un marco de ayuda para producir mensajes en distintos lenguajes a otros paquetes o programas del proyecto GNU. GETTEXT esta diseñado para minimizar el impacto de la internacionalización del código fuente de los programas.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *gettext-0.17.tar.gz*

URL de descarga:

<http://www.gnu.org/software/gettext/gettext.html#TOCdownloading>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/gettext-0.17.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf gettext-0.17.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd gettext-0.17
```

Ejecutar el comando *configure*.

```
usuario@server:/usr/local/src/gettext-0.17$ ./configure
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/gettext-0.17$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
server:/usr/local/src/gettext-0.17$ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/gettext-0.17# make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/gettext-0.17# exit
```

Regresar a la carpeta *src*.

```
usuario@server:/usr/local/src/gettext-0.17$ cd ..
```

GLIB

Proporciona el sistema de objetos básicos usados en el entorno gráfico GNOME, la implementación del bucle principal y un gran conjunto de cadenas y estructuras de datos comunes necesarios para la construcción de aplicaciones y bibliotecas escritas en lenguaje C. Su principal cualidad es poder ser usado como plataforma independiente que permite que el código generado pueda utilizarse en diferentes sistemas operativos.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *glib-2.21.6.tar.gz*

URL de descarga: <http://ftp.gnome.org/pub/gnome/sources/glib/2.21/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta `/usr/local/src`.

```
usuario@server:~$ cp /home/usuario/Fuentes/glib-2.21.6.tar.gz  
/usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf glib-2.21.6.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd glib-2.21.6
```

Ejecutar el comando `configure` agregando el parámetro `prefix` al cual se le asignara la ruta donde se desea que se ubique la instalación.

```
usuario@server:/usr/local/src/glib-2.21.6$ ./configure --  
prefix=/usr/local/glib2216
```

Ejecutar el comando `make`.

```
usuario@server:/usr/local/src/glib-2.21.6$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario `root`.

```
usuario@server:/usr/local/src/glib-2.21.6$ su root
```

Ejecutar el comando `make install` para aplicar la instalación.

```
server:/usr/local/src/glib-2.21.6# make install
```

Copiar el archivo `glib-2.0.pc` en la ruta `/usr/lib/pkgconfig/`.

```
server:/usr/local/src/glib-2.21.6# cp glib-2.0.pc /usr/lib/pkgconfig/
```

Copiar el archivo `glib-2.0.pc` en la ruta `/usr/local/lib/pkgconfig/`.

```
server:/usr/local/src/glib-2.21.6# cp glib-2.0.pc /usr/local/lib/pkgconfig/
```

Ejecutar el comando `pkg-config --cflags glib-2.0 -I/usr/local/include/glib-2.0 -I/usr/local/lib/glib-2.0/include` para agregar la nueva librería a `pkgconfig`.

```
server:/usr/local/src/glib-2.21.6# pkg-config --cflags glib-2.0 -I  
/usr/local/include/glib-2.0 -I/usr/local/lib/glib-2.0/include
```

Agregar las siguientes variables al archivo `/etc/profile` mediante el editor de texto VI.

```
server:/usr/local/src/glib-2.21.6# vi /etc/profile
PKG_CONFIG_PATH=/usr/local/gtk216/glib2216/lib/pkgconfig
LD_LIBRARY_PATH=/usrlocal/gtk216/glib2216/lib
export PKG_CONFIG_PATH LD_LIBRARY_PATH
```

Cargar al cache del sistema el archivo `/etc/profile` para que estén disponibles las nuevas variables.

```
server:/usr/local/src/glib-2.21.6# source /etc/profile
```

Ejecutar el comando `ldconfig -v` para crear los links y el cache de la nueva librería instalada.

```
server:/usr/local/src/glib-2.21.6# ldconfig -v
```

Cerrar la sesión de `root` al terminar la ejecución del comando anterior.

```
server:/usr/local/src/glib-2.21.6# exit
```

Regresar a la carpeta `src`.

```
server:/usr/local/src/glib-2.21.6$ cd ..
```

TIFF

Es un formato de archivo orientado al manejo de imágenes integrando el etiquetado de la imagen el cual permite describir el formato y la compresión de las imágenes almacenadas. Permite la compresión de imágenes sin que estas pierdan su calidad original. Existen etiquetas que especifican el formato interno de almacenamiento de la imagen lo que permite a muchas aplicaciones optimizar los tiempos de Cargar o leer únicamente la zona de interés de una imagen grande.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: `tiff-3.8.2.tar.gz`

URL de descarga: <http://dl.maptools.org/dl/libtiff/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta `/usr/local/src`.

```
usuario@server:~$ cp /home/usuario/Fuentes/tiff-3.8.2.tar.gz /usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf tiff-3.8.2.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd tiff-3.8.2
```

Ejecutar el comando `configure`.

```
usuario@server:/usr/local/src/tiff-3.8.2 $./configure
```

Ejecutar el comando `make`.

```
usuario@server:/usr/local/src/tiff-3.8.2 $ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario `root`.

```
usuario@server:/usr/local/src/tiff-3.8.2 $ su root
```

Ejecutar el comando `make install` para aplicar la instalación.

```
server:/usr/local/src/tiff-3.8.2 # make install
```

Cerrar la sesión de `root` al terminar la ejecución del comando anterior.

```
server:/usr/local/src/tiff-3.8.2 # exit
```

Regresar a la carpeta `src`.

```
usuario@server:/usr/local/src/tiff-3.8.2 $ cd ..
```

JASPER

Es un conjunto de herramientas de software para el manejo de imágenes, proporciona la interfaz necesaria para la representación de imágenes y facilita la

manipulación de las imágenes, además de la importación y exportación entre numerosos formatos de imágenes, esto permite eliminar la necesidad de identificar forzosamente el formato de las imágenes que se requiere utilizar.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *jasper-1.900.1.zip*

URL de descarga: <http://www.ece.uvic.ca/~mdadams/jasper/-download>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/jasper-1.900.1.zip  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ unzip jasper-1.900.1.zip
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd jasper-1.900.1
```

Ejecutar el comando *configure*.

```
usuario@server:/usr/local/src/jasper-1.900.1 $./configure
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/jasper-1.900.1 $ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/jasper-1.900.1 $ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/jasper-1.900.1 # make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/jasper-1.900.1# exit
```

Regresar a la carpeta *src*.

```
usuario@server:/usr/local/src/jasper-1.900.1 $ cd ..
```

LIBPCAP

Es una interfaz de una aplicación de programación independiente para capturar paquetes a nivel de red, es utilizado como parte de programas cuya función es la captura de los paquetes que viajan por toda la red y para transmitir los paquetes en la capa de enlace de una red. Algunas de sus aplicaciones son: la recolección de estadísticas, monitoreo de la seguridad, la depuración de la red.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *libpcap-1.0.0.tar.gz*

URL de descarga: <http://www.tcpdump.org/release/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/libpcap-1.0.0.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf libpcap-1.0.0.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd libpcap-1.0.0
```

*Manual de instalación para implementar un servidor de desarrollo sobre plataforma OpenSource
Instalación de librerías y programas base necesarios.*

Ejecutar el comando *configure*.

```
usuario@server:/usr/local/src/libpcap-1.0.0 $ ./configure
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/libpcap-1.0.0 $ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/libpcap-1.0.0 $ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/libpcap-1.0.0 # make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/libpcap-1.0.0# exit
```

Regresar a la carpeta *src*.

```
usuario@server:/usr/local/src/libpcap-1.0.0 $ cd ..
```

OPENSSL

Es un conjunto de herramientas de administración y librerías orientadas a tareas de criptografía, las funciones criptográficas que contiene son utilizadas por otros paquetes como OpenSSH y navegadores web para el acceso al protocolo HTTPS. Otra de sus aplicaciones son: Implementar los protocolos SSL(Secure Sockets Layer), TLS(Transport Layer Security) y manejar certificados digitales para ser aplicados a servidores como Apache. Este paquete es indispensable si se requiere implementar niveles de seguridad en equipos con software libre basado en GNU/Linux.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *openssl-0.9.8k.tar.gz*

URL de descarga: <http://www.openssl.org/source/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

*Manual de instalación para implementar un servidor de desarrollo sobre plataforma OpenSource
Instalación de librerías y programas base necesarios.*

Copiar el archivo fuente del paquete a la ruta `/usr/local/src`.

```
usuario@server:~$ cp /home/usuario/Fuentes/openssl-0.9.8k.tar.gz  
/usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf openssl-0.9.8k.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd openssl-0.9.8k
```

Ejecutar el comando `configure`.

```
usuario@server:/usr/local/src/openssl-0.9.8k $./configure
```

Ejecutar el comando `make`.

```
usuario@server:/usr/local/src/openssl-0.9.8k$ make
```

Ejecutar el comando `make test`.

```
usuario@server:/usr/local/src/openssl-0.9.8k$ make test
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario `root`.

```
usuario@server:/usr/local/src/openssl-0.9.8k $ su root
```

Ejecutar el comando `make install` para aplicar la instalación.

```
server:/usr/local/src/openssl-0.9.8k # make install
```

Cerrar la sesión de `root` al terminar la ejecución del comando anterior.

```
server:/usr/local/src/openssl-0.9.8k # exit
```

Regresar a la carpeta `src`.

```
usuario@server:/usr/local/src/openssl-0.9.8k $ cd ..
```

LIBXML2

Es la biblioteca de provee la interfaz necesaria para el análisis de documentos XML, originalmente fue desarrollada para el proyecto GNOME pero puede ser utilizada en cualquier otro ambiente gráfico que funcione sobre GNU/Linux. El código de libxml es altamente portátil, ya que depende de las bibliotecas estándar ANSI C solamente, Libxml implementa un número de lenguajes standards relacionados, desde el típico XML, espacio de nombres en XML, XML Base, RFC 2396, Xpath, Xpointer, HTML4, Xinclude y catálogos SGML y XML.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *libxml2-2.7.3.tar.gz*

URL de descarga: <http://xmlsoft.org/sources/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/libxml2-2.7.3.tar.gz  
/usr/local/src/
```

Ingresa a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf libxml2-2.7.3.tar.gz
```

Ingresa a la carpeta generada.

```
usuario@server: /usr/local/src$ cd libxml2-2.7.3
```

Ejecutar el comando *configure*.

```
usuario@server:/usr/local/src/libxml2-2.7.3$ ./configure
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/libxml2-2.7.3$ make
```

*Manual de instalación para implementar un servidor de desarrollo sobre plataforma OpenSource
Instalación de librerías y programas base necesarios.*

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/libxml2-2.7.3 $ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/libxml2-2.7.3 # make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/libxml2-2.7.3 # exit
```

Regresar a la carpeta *src*.

```
usuario@server:/usr/local/src/libxml2-2.7.3 $ cd ..
```

LIBPNG

Es la biblioteca oficial para el manejo de imágenes de formato PNG, es multiplataforma y depende de *zlib* para su funcionamiento correcto y seguridad. PNG es un formato de gráficos basado en el algoritmo de compresión sin pérdida para mapas de bits, permite almacenar imágenes con una mayor profundidad de contraste y otros importantes datos, este formato es el más popular para manejo de imágenes gráficas y es soportado por todos los navegadores.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *libpng-1.2.35.tar.gz*

URL de descarga: <ftp://ftp.simplesystems.org/pub/libpng/png/src>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/libpng-1.2.35.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf libpng-1.2.35.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd libpng-1.2.35
```

Ejecutar el comando *configure*.

```
usuario@server:/usr/local/src/libpng-1.2.35$ ./configure
```

Ejecutar el comando *make* indicando las variables de ubicación de la instalación (prefix), la ubicación donde se ubicaran las librerías y el nombre del script de instalación.

```
usuario@server:/usr/local/src/libpng-1.2.35$ make prefix=/usr  
ZLIBINC=/usr/include ZLIBLIB=/usr/lib -f scripts/makefile.linux
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/libpng-1.2.35 $ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/libpng-1.2.35 # make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/libpng-1.2.35 # exit
```

Regresar a la carpeta *src*.

```
usuario@server:/usr/local/src/libpng-1.2.35 $ cd ..
```

JPEG

Es la biblioteca oficial para el manejo de imágenes de formato JPEG, su función es la compresión de imágenes utilizando el algoritmo de compresión con pérdida, lo que significa que al descomprimir la imagen no se obtendrá la imagen que originalmente se comprimió. Su principal característica es que es muy flexible al elegir el grado de compresión de la imagen según se requiera.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *jpegsrc.v6b.tar.gz*

URL de descarga:

<http://dir.filewatcher.com/d/GNU/Other/jpegsrc.v6b.tar.gz.613261.html>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/jpegsrc.v6b.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf jpegsrc.v6b.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd jpeg-6b
```

Ejecutar el comando *configure* activando las opciones de *shared* y *static*.

```
usuario@server:/usr/local/src/jpeg-6b$ ./configure --enable-shared --enable-  
static
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/jpeg-6b$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/jpeg-6b $ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/jpeg-6b # make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/jpeg-6b # exit
```

Regresar a la carpeta `src`.

```
usuario@server:/usr/local/src/jpeg-6b $ cd ..
```

FREETYPE

Es una biblioteca que se encuentra escrita en lenguaje C que implementa un motor de fuentes, es usado principalmente para transformar las imágenes vectoriales de las fuentes a mapas de bits. Esencialmente permite el acceso a los archivos de las fuentes de texto, las transformaciones de mayor nivel como distribución del texto, coloración o efectos especiales se encuentran fuera de su alcance. Soporta formatos de fuente escalables como por ejemplo las fuentes TrueType, Type1, CID-Type 1, Multiple Masters Type 1 OpenType/CFF, CFF, SFNT, X11 PCF, CEF, BDF, PFR, Type42, y Windows FNT/FON. Su arquitectura es modular, por lo que permite agregar módulos de fuente independientes.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: `freetype-2.3.9.tar.gz`

URL de descarga: <http://download.savannah.gnu.org/releases/freetype/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

Copiar el archivo fuente del paquete a la ruta `/usr/local/src`.

```
usuario@server:~$ cp /home/usuario/Fuentes/freetype-2.3.9.tar.gz  
/usr/local/src/
```

Ingresa a la ruta `/usr/local/src`.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$ tar -zxvf freetype-2.3.9.tar.gz
```

Ingresa a la carpeta generada.

```
usuario@server:/usr/local/src$ cd freetype-2.3.9
```

Ejecutar el comando *configure*.

```
usuario@server:/usr/local/src/freetype-2.3.9$ ./configure
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/freetype-2.3.9$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/freetype-2.3.9 $ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/freetype-2.3.9 # make install
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/freetype-2.3.9 # exit
```

Regresar a la carpeta *src*.

```
usuario@usuario@server:/usr/local/src/freetype-2.3.9 $ cd ..
```

GD

Es una biblioteca que es utilizada para la creación dinámica de imágenes, entre los formatos de imágenes que puede generar se encuentran; PNG, JPEG y GIF entre otros. Es comúnmente utilizado para generar tablas, gráficos, imágenes en miniatura entre otras tareas, todo de forma dinámica. No está limitado a ser utilizado sólo para el desarrollo de aplicaciones web. Soporta múltiples lenguajes de programación como: C, PHP, Perl, Python, OCaml, Tcl, Lua, Pascal, GNU Octave, REXX y Ruby.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *gd-2.0.35.tar.gz*

URL de descarga: <http://www.libgd.org/Downloads - Download the latest .282.0.35.29>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.*

*Manual de instalación para implementar un servidor de desarrollo sobre plataforma OpenSource
Instalación de librerías y programas base necesarios.*

Copiar el archivo fuente del paquete a la ruta `/usr/local/src`.

```
usuario@server:~$ cp /home/usuario/Fuentes/gd-2.0.35.tar.gz /usr/local/src/
```

Ingresa a la ruta `/usr/local/src`.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf gd-2.0.35.tar.gz
```

Ingresa a la carpeta generada.

```
usuario@server: /usr/local/src$ cd gd-2.0.35
```

Ejecutar el comando `configure` indicando la ubicación de la instalación.

```
usuario@server:/usr/local/src/gd-2.0.35$ ./configure --prefix=/usr/local/gd
```

Ejecutar el comando `make`.

```
usuario@server:/usr/local/src/gd-2.0.35$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario `root`.

```
usuario@server:/usr/local/src/gd-2.0.35 $ su root
```

Ejecutar el comando `make install` para aplicar la instalación.

```
server:/usr/local/src/gd-2.0.35 # make install
```

Al terminar la instalación verificar que las siguientes líneas se muestren en la consola de comandos.

```
#----->  
Support for PNG library:      yes  
Support for JPEG library:    yes  
Support for Freetype 2.x library: yes  
Support for Fontconfig library: yes  
Support for Xpm library:     no  
Support for pthreads:       yes  
#----->
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/gd-2.0.35 # exit
```

Regresar a la carpeta *src*.

```
usuario@server:/usr/local/src/gd-2.0.35 $ cd ..
```

1.3 Herramientas de configuración, monitoreo y comunicación remota

SYSV-RC-CONF

Herramienta de configuración que proporciona una interfaz gráfica para línea de comandos, que permite gestionar los enlaces simbólicos que se encuentran en los niveles de arranque del sistema operativo, a diferencia de otras herramientas que permiten realizar la misma función que *sysv-rc-conf*, esta herramienta permite editar los enlaces simbólicos de los diferentes niveles de ejecución.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario *root*.

```
server:~# apt-get install sysv-rc-conf
```

SSH

Paquete que facilita la instalación del cliente para comunicación remota OpenSSH y el servidor OpenSSH, estas herramientas permiten realizar la conexión remota al equipo de forma similar al comando *telnet*, sólo que la información que envía y recibe *ssh* viaja de manera ilegible usando técnicas de cifrado.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se ejecuta la instrucción como usuario *root*.

```
server:~# apt-get install ssh
```

NMAP

Programa que permite la exploración de la red y auditar la seguridad, podemos realizar escaneos utilizando ping's para verificar si una máquina está activa. Su uso principal es para verificar que puertos se encuentran activos en un determinado equipo y que servicios están escuchando en ellos.

Tipo de licencia: OpenSource.

Proceso de instalación

Dentro de la consola de comandos se Ejecutar la instrucción como usuario root.

```
server:~# apt-get install nmap
```

Al terminar la instalación de los programas anteriores, el sistema estará preparado para realizar la compilación e instalación de los programas que serán utilizados en el servidor de desarrollo, el flujo de la instalación que se documento se realizo de tal forma que se eviten los conflictos de dependencias entre los paquetes. Si se utilizan versiones diferentes a las indicadas se pueden presentar problemas de dependencias, por lo tanto tendrían que resolverse esos conflictos instalando las versiones más actualizadas si esto sucediera.

Capítulo 2 Manejadores de Bases de Datos

Los manejadores de bases de datos se han convertido en una pieza muy importante al momento de desarrollar aplicaciones tanto para web como para aplicaciones de escritorio, ya que proporcionan una gran capacidad de manejo de información a través de las transacciones. Existen diversas opciones de manejadores de bases de datos para elegir, en este manual se propone el uso de dos opciones muy populares en el mundo del desarrollo de software OpenSource, dependiendo de las características de la aplicación a desarrollar, se puede elegir entre cualquiera de estos dos manejadores que proporcionan las herramientas y funciones necesarias para implementar un robusto sistema.

2.1 MySQL

Es un gestor de bases de datos relacional, multihilo y multiusuario, se ofrece bajo el esquema GNU GPL, pero si se requiere implementar en productos privativos se necesita adquirir la licencia correspondiente. Pude usarse con distintos lenguajes de programación como: JAVA, C, C++, Perl, Pitón, Ruby entre otros, el lenguaje que la ha hecho más popular es PHP. MySQL es muy utilizada en aplicaciones Web como Drupal o phpBB, trabaja sobre plataformas Windows y Linux, es altamente recomendable utilizar MySQL en aplicaciones que demanden muchas transacciones de lectura de datos, y no se recomienda utilizarla en aplicaciones que demanden alta concurrencia de modificaciones de datos.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *mysql-5.1.33.tar.gz*

URL de descarga: <ftp://ftp.uni-hannover.de/pub/mirror/bsd/FreeBSD/ports/distfiles/mysql-5.1.33.tar.gz>

Indicaciones previas: *La ejecución de los comandos se realizara dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 2.1 se puede apreciar el flujo de comandos de instalación.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
usuario@server:~$ cp /home/usuario/Fuentes/mysql-5.1.33.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf mysql-5.1.33.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd mysql-5.1.33
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf mysql-5.1.33.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd mysql-5.1.33
```

Iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/mysql-5.1.33$ su root
```

Ejecutar el comando *groupadd* para agregar el grupo *mysql*.

```
server:/usr/local/src/mysql-5.1.33#groupadd mysql
```

Ejecutar el comando *useradd* para crear el usuario *mysql* y agregarlo al grupo *mysql*.

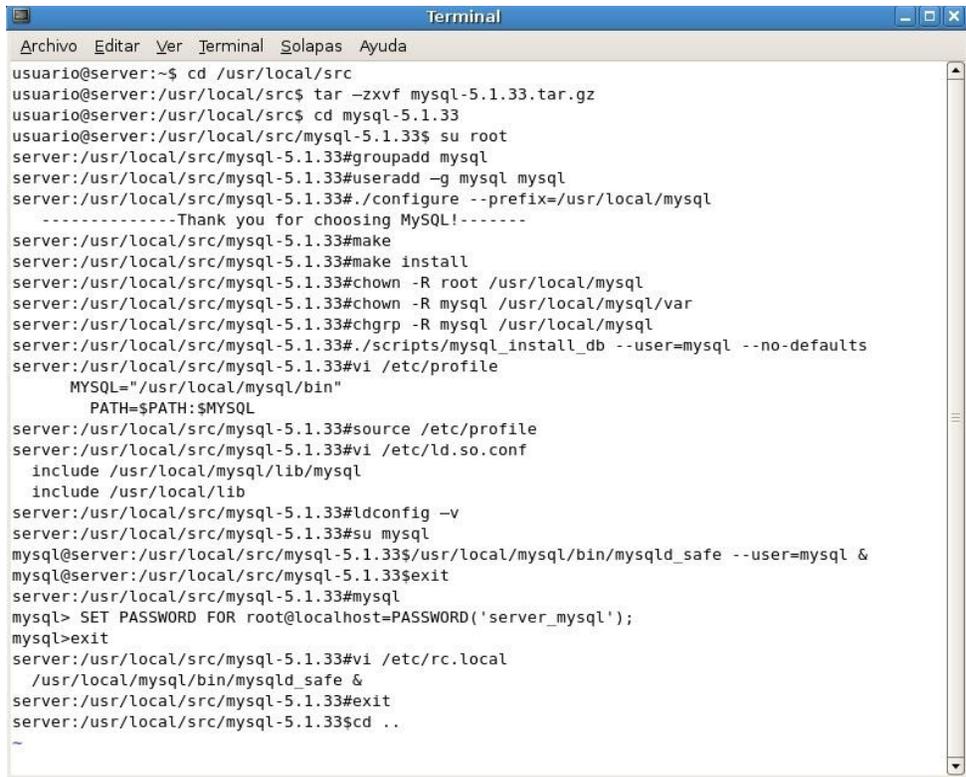
```
server:/usr/local/src/mysql-5.1.33#useradd -g mysql mysql
```

Ejecutar el comando *configure* indicando la ubicación de la instalación de *mysql*.

```
server:/usr/local/src/mysql-5.1.33#./configure --prefix=/usr/local/mysql
```

Para comprobar que la configuración es exitosa, Verificar que se muestre la siguiente leyenda al finalizar la ejecución del comando anterior.

```
-----Thank you for choosing MySQL!-----
```



```
usuario@server:~$ cd /usr/local/src
usuario@server:/usr/local/src$ tar -zxvf mysql-5.1.33.tar.gz
usuario@server:/usr/local/src$ cd mysql-5.1.33
usuario@server:/usr/local/src/mysql-5.1.33$ su root
server:/usr/local/src/mysql-5.1.33#groupadd mysql
server:/usr/local/src/mysql-5.1.33#useradd -g mysql mysql
server:/usr/local/src/mysql-5.1.33#./configure --prefix=/usr/local/mysql
-----Thank you for choosing MySQL!-----
server:/usr/local/src/mysql-5.1.33#make
server:/usr/local/src/mysql-5.1.33#make install
server:/usr/local/src/mysql-5.1.33#chown -R root /usr/local/mysql
server:/usr/local/src/mysql-5.1.33#chown -R mysql /usr/local/mysql/var
server:/usr/local/src/mysql-5.1.33#chgrp -R mysql /usr/local/mysql
server:/usr/local/src/mysql-5.1.33#./scripts/mysql_install_db --user=mysql --no-defaults
server:/usr/local/src/mysql-5.1.33#vi /etc/profile
MYSQL="/usr/local/mysql/bin"
PATH=$PATH:$MYSQL
server:/usr/local/src/mysql-5.1.33#source /etc/profile
server:/usr/local/src/mysql-5.1.33#vi /etc/ld.so.conf
include /usr/local/mysql/lib/mysql
include /usr/local/lib
server:/usr/local/src/mysql-5.1.33#ldconfig -v
server:/usr/local/src/mysql-5.1.33#su mysql
mysql@server:/usr/local/src/mysql-5.1.33$/usr/local/mysql/bin/mysqld_safe --user=mysql &
mysql@server:/usr/local/src/mysql-5.1.33$exit
server:/usr/local/src/mysql-5.1.33#mysql
mysql> SET PASSWORD FOR root@localhost=PASSWORD('server_mysql');
mysql>exit
server:/usr/local/src/mysql-5.1.33#vi /etc/rc.local
/usr/local/mysql/bin/mysqld_safe &
server:/usr/local/src/mysql-5.1.33#exit
server:/usr/local/src/mysql-5.1.33$cd ..
```

Imagen 2.1. Ejecución del flujo de instalación de MySQL vía compilación de código fuente.

Ejecutar el comando *make*.

```
server:/usr/local/src/mysql-5.1.33# make
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/mysql-5.1.33# make install
```

Asignar al usuario *root* como propietario de la carpeta */usr/local/mysql* y sus subcarpetas.

```
server:/usr/local/src/mysql-5.1.33# chown -R root /usr/local/mysql
```

Asignar al usuario *mysql* como propietario de la carpeta */usr/local/mysql/var* y sus subcarpetas.

```
server:/usr/local/src/mysql-5.1.33# chown -R mysql /usr/local/mysql/var
```

Asignar la carpeta */usr/local/mysql* y sus subcarpetas al grupo *mysql*.

```
server:/usr/local/src/mysql-5.1.33# chgrp -R mysql /usr/local/mysql
```

Ejecutar el script de instalación de la base de datos *mysql_install_db*, indicando el usuario que dará soporte a *mysql*.

```
server:/usr/local/src/mysql-5.1.33# ./scripts/mysql_install_db --  
user=mysql --no-defaults
```

Utilizar el editor de textos de preferencia para agregar al archivo *profile* la variable *MYSQL* que contiene la ruta de la ubicación de los binarios de *mysql*.

```
server:/usr/local/src/mysql-5.1.33# vi /etc/profile
```

```
MYSQL="/usr/local/mysql/bin"
```

```
PATH=$PATH:$MYSQL
```

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando *source*.

```
server:/usr/local/src/mysql-5.1.33# source /etc/profile
```

Utilizar el editor de textos de preferencia para agregar al archivo *ld.so.conf* la ruta de la ubicación de las librerías de *mysql*.

```
server:/usr/local/src/mysql-5.1.33# vi /etc/ld.so.conf
```

```
include /usr/local/mysql/lib/mysql
```

```
include /usr/local/lib
```

Ejecutar el comando *ldconfig* para crear los vínculos y cache necesarios en tiempo de ejecución las rutas de las librerías agregadas en el paso anterior.

```
server:/usr/local/src/mysql-5.1.33# ldconfig -v
```

Iniciar sesión como usuario *mysql*.

```
server:/usr/local/src/mysql-5.1.33# su mysql
```

Ejecutar el comando *mysqld_safe* para levantar el servicio de *mysql*.

```
mysql@server:/usr/local/src/mysql-5.1.33$  
/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

Cerrar la sesión de *mysql*.

```
mysql@server:/usr/local/src/mysql-5.1.33$exit
```

Ingresar a la línea de comandos de *mysql*.

```
server:/usr/local/src/mysql-5.1.33#mysql
```

Ejecutar el comando de *mysql* para asignarle contraseña de acceso al usuario *root* en *mysql*.

```
mysql> SET PASSWORD FOR root@localhost=PASSWORD('server_mysql');
```

Cerrar la sesión de línea de comandos de *mysql*.

```
mysql>exit
```

Utilizar el editor de textos de preferencia para agregar al archivo *rc.local* el comando que iniciara el servicio *mysql* al cargar el sistema.

```
server:/usr/local/src/mysql-5.1.33# vi /etc/rc.local  
/usr/local/mysql/bin/mysqld_safe &
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/mysql-5.1.33# exit
```

Regresar a la carpeta *src*.

```
server:/usr/local/src/mysql-5.1.33$ cd ..
```

2.2 POSTGRES

Es un gestor de bases de datos relacional orientada a objetos, es orientado a la comunidad de desarrolladores y organizaciones comerciales, las cuales son las encargadas de su desarrollo. Permite la alta concurrencia, tiene soporte para diferentes tipos nativos de datos, tienen soporte para triggers que proporcionan una amplia funcionalidad a través de un sistema de activación de *triggers* y procedimientos almacenados para la gestión y mantenimiento de la base de datos. Es compatible con el almacenamiento de objetos binarios incluyendo

imágenes, sonidos y videos. También cuenta con sistema RollBack, transacciones anidadas con RollBack, replicación asíncrona y es altamente escalable tanto en la cantidad de datos que puede manejar como en el número de usuarios que en concurrencia que puede soportar. Algunos de los lenguajes que soporta son: C, C++, JAVA, Perl, PHP, Pitón, Ruby entre otros.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *postgresql-8.4.0.tar.gz*

URL de descarga: <http://wwwmaster.postgresql.org/download/mirrors-ftp/source/v8.4.0/postgresql-8.4.0.tar.gz>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 2.2 se puede apreciar el flujo de comandos de instalación.*

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
server:~$ cp /home/usuario/Fuentes/postgresql-8.4.0.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$ tar -zxvf postgresql-8.4.0.tar.gz
```

Copiar el archivo fuente del paquete a la ruta */usr/local/src*.

```
server:~$ cp /home/usuario/Fuentes/postgresql-8.4.0.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

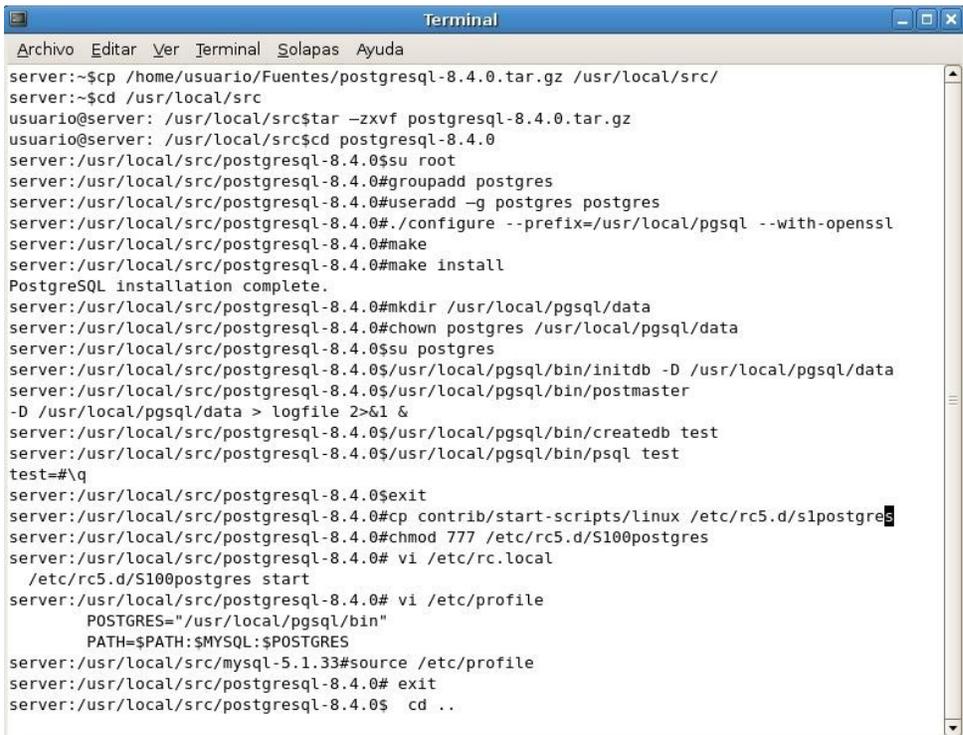
```
usuario@server: /usr/local/src$ tar -zxvf postgresql-8.4.0.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server: /usr/local/src$ cd postgresql-8.4.0
```

Iniciar sesión en la consola de comandos como usuario *root*.

```
server:/usr/local/src/postgresql-8.4.0$ su root
```



```
server:~$cp /home/usuario/Fuentes/postgresql-8.4.0.tar.gz /usr/local/src/
server:~$cd /usr/local/src
usuario@server: /usr/local/src$tar -zxvf postgresql-8.4.0.tar.gz
usuario@server: /usr/local/src$cd postgresql-8.4.0
server:/usr/local/src/postgresql-8.4.0$su root
server:/usr/local/src/postgresql-8.4.0#groupadd postgres
server:/usr/local/src/postgresql-8.4.0#useradd -g postgres postgres
server:/usr/local/src/postgresql-8.4.0#./configure --prefix=/usr/local/pgsql --with-openssl
server:/usr/local/src/postgresql-8.4.0#make
server:/usr/local/src/postgresql-8.4.0#make install
PostgreSQL installation complete.
server:/usr/local/src/postgresql-8.4.0#mkdir /usr/local/pgsql/data
server:/usr/local/src/postgresql-8.4.0#chown postgres /usr/local/pgsql/data
server:/usr/local/src/postgresql-8.4.0$$su postgres
server:/usr/local/src/postgresql-8.4.0$/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
server:/usr/local/src/postgresql-8.4.0$/usr/local/pgsql/bin/postmaster
-D /usr/local/pgsql/data > logfile 2>&1 &
server:/usr/local/src/postgresql-8.4.0$/usr/local/pgsql/bin/createdb test
server:/usr/local/src/postgresql-8.4.0$/usr/local/pgsql/bin/psql test
test=#\q
server:/usr/local/src/postgresql-8.4.0$exit
server:/usr/local/src/postgresql-8.4.0#cp contrib/start-scripts/linux /etc/rc5.d/s1postgre
server:/usr/local/src/postgresql-8.4.0#chmod 777 /etc/rc5.d/S100postgres
server:/usr/local/src/postgresql-8.4.0# vi /etc/rc.local
/etc/rc5.d/S100postgres start
server:/usr/local/src/postgresql-8.4.0# vi /etc/profile
POSTGRES="/usr/local/pgsql/bin"
PATH=$PATH:$MYSQL:$POSTGRES
server:/usr/local/src/mysql-5.1.33#source /etc/profile
server:/usr/local/src/postgresql-8.4.0# exit
server:/usr/local/src/postgresql-8.4.0$ cd ..
```

Imagen 2.2. Ejecución del flujo de instalación de Postgres vía compilación de código fuente.

Ejecutar el comando *useradd* para crear el usuario *postgres* y agregarlo al grupo *postgres*.

```
server:/usr/local/src/postgresql-8.4.0#useradd -g postgres postgres
```

Ejecutar el comando *configure* indicando la ubicación de la instalación de *postgres* y activando la conectividad con *openssl*.

```
server:/usr/local/src/postgresql-8.4.0#./configure --prefix=/usr/local/pgsql
--with-openssl
```

Ejecutar el comando *make*.

```
server:/usr/local/src/postgresql-8.4.0# make
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/postgresql-8.4.0# make install
```

Para comprobar que la instalación es exitosa, verificar que se muestre la siguiente leyenda al finalizar la ejecución del comando anterior.

```
PostgreSQL installation complete.
```

Crear la carpeta *data* bajo la ruta */usr/local/pgsql*.

```
server:/usr/local/src/postgresql-8.4.0#mkdir /usr/local/pgsql/data
```

Asignar al usuario *postgres* como propietario de la carpeta */usr/local/pgsql/data*.

```
server:/usr/local/src/postgresql-8.4.0# chown postgres  
/usr/local/pgsql/data
```

Iniciar sesión con el usuario *postgres*.

```
server:/usr/local/src/postgresql-8.4.0$ su postgres
```

Ejecutar el comando *initdb* para inicializar la configuración de la base de datos.

```
server:/usr/local/src/postgresql-8.4.0$ /usr/local/pgsql/bin/initdb -D  
/usr/local/pgsql/data
```

Ejecutar el comando *postmaster* y envíalo a segundo plano para iniciar la base de datos.

```
server:/usr/local/src/postgresql-8.4.0$/usr/local/pgsql/bin/postmaster  
-D /usr/local/pgsql/data > logfile 2>&1 &
```

Ejecutar el comando */usr/local/pgsql/bin/createdb test* para crear una base de datos de prueba.

```
server:/usr/local/src/postgresql-8.4.0$ /usr/local/pgsql/bin/createdb test
```

Ejecutar el comando `/usr/local/pgsql/bin/psql test` para acceder a la línea de comandos de *postgres* e ingresar a la base de datos *test*.

```
server:/usr/local/src/postgresql-8.4.0$ /usr/local/pgsql/bin/psql test
```

Dentro de la línea de comandos de *postgres* ejecutar el comando `\q` para salir.

```
test=#\q
```

Cerrar la sesión de *postgres*.

```
server:/usr/local/src/postgresql-8.4.0$ exit
```

Copiar el archivo *linux* que se encuentra bajo la ruta *contrib/start-scripts/* a la ruta */etc/rc5.d/* con el nombre *S100postgres*.

```
server:/usr/local/src/postgresql-8.4.0#
```

```
cp contrib/start-scripts/linux /etc/rc5.d/S100postgres
```

Cambia los permisos del archivo *S100postgres* y dar acceso total.

```
server:/usr/local/src/postgresql-8.4.0# chmod 777  
/etc/rc5.d/S100postgres
```

Utilizar el editor de textos de preferencia para agregar al archivo *rc.local* el comando que iniciara el servicio *postgres* al cargar el sistema.

```
server:/usr/local/src/postgresql-8.4.0# vi /etc/rc.local  
  
/etc/rc5.d/S100postgres start
```

Utilizar el editor de textos para agregar al archivo *profile* la variable de ambiente *POSTGRES* y agregarla al *PATH* del sistema.

```
server:/usr/local/src/postgresql-8.4.0# vi /etc/profile  
  
POSTGRES="/usr/local/pgsql/bin"
```

```
PATH=$PATH:$MYSQL:$POSTGRES
```

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando *source*.

```
server:/usr/local/src/mysql-5.1.33#source /etc/profile
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/postgresql-8.4.0# exit
```

Regresar a la carpeta *src*.

```
server:/usr/local/src/postgresql-8.4.0$ cd ..
```

Al tener instalados los manejadores de bases de datos MySQL y Postgres en el servidor de desarrollo, permite facilitar la elección sobre cuál manejador usar para la aplicación que necesite desarrollar, esto debido a que si se desarrollan clases especializadas que homogenicen la conexión a la base de datos, se podrá simplemente con sólo cambiar algunos parámetros cambiar el manejador de base de datos que se requiera utilizar, sin que esta operación afecte el código escrito previamente. Las opciones de manejadores de bases de datos propuestos son ampliamente utilizados en el desarrollo web actualmente, y aunque no son tan robustas como algunas opciones comerciales, si cumplen con todos los requerimientos necesarios.

Capítulo 3 Servidores Web

El desarrollo de las nuevas tecnologías de información se han estado enfocando al desarrollo de aplicaciones web, que permitan el acceso a las mismas desde cualquier equipo de cómputo e incluso equipos móviles, la tarea de establecer la comunicación entre los diferentes dispositivos y las aplicaciones la realiza el servidor web. El tener instalado un servidor web en el equipo local facilita las tareas de la realización de las pruebas a la aplicación que se este desarrollando, evitando tener que estar actualizando los cambios de código realizados en algún equipo remoto, que este funcionando como host de la aplicación. Existen diversas opciones de servidores web comerciales y gratuitos, aquí se propone el uso de Apache en sus dos versiones y el servidor web Cherokee, ambos ofrecen diferentes herramientas y se pueden configurar de tal forma que ofrezcan una buena plataforma para poder desarrollar y probar las aplicaciones antes de ser liberadas.

3.1 CHEROKEE

Es un servidor web multiplataforma, de alto rendimiento, rápido flexible y de fácil configuración, ofrece soporte para las tecnologías: FastCGI, SCGI, PHP, CGI, SSI, TLS y conexiones SSL encriptadas, hosts virtuales, autenticación, sobre la codificación en marcha, equilibrio de Cargar, los archivos de log de Apache compatibles, Base de datos Balancer, sin tiempo de inactividad actualizaciones y mejoras, Reverse Proxy HTTP, entre otras. Esta escrito totalmente en lenguaje C. Dentro de sus principales características la más destacada es la interfaz de administración, llamada cheerokee-admin que proporciona una manera sencilla y sin complicaciones de configurar cada característica única del servidor.

Tipo de licencia: OpenSource.

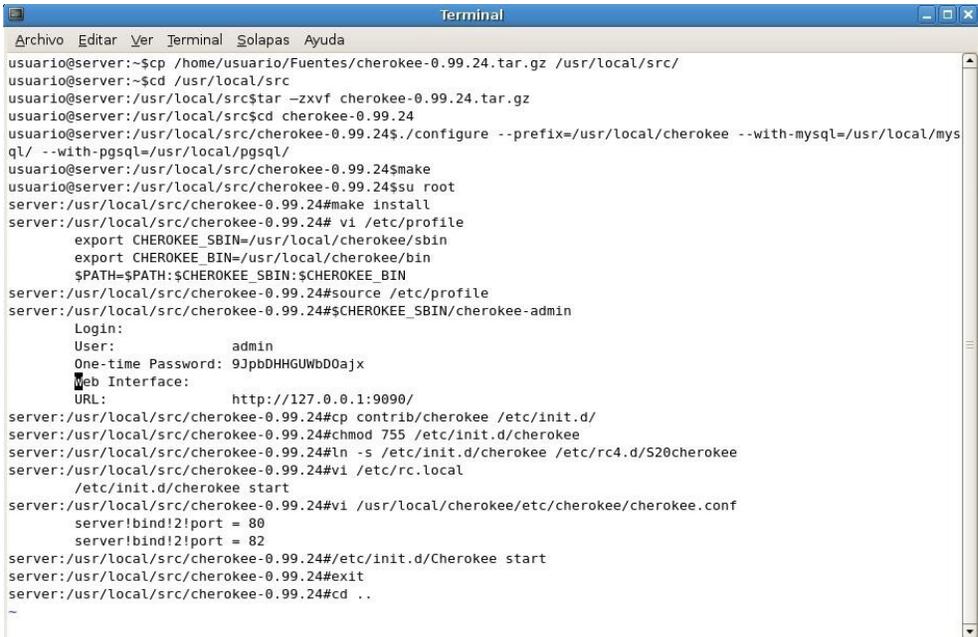
Proceso de instalación

Versión utilizada: *cherokee-0.99.24.tar.gz*

URL de descarga: <http://www.cherokee-project.com/download/0.99/0.99.24/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en*

caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 3.1 se puede apreciar el flujo de comandos de instalación.



```
Terminal
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
usuario@server:~$ cp /home/usuario/Fuentes/ Cherokee-0.99.24.tar.gz /usr/local/src/
usuario@server:~$ cd /usr/local/src
usuario@server:/usr/local/src$ tar -zxvf Cherokee-0.99.24.tar.gz
usuario@server:/usr/local/src$ cd Cherokee-0.99.24
usuario@server:/usr/local/src/Cherokee-0.99.24$ ./configure --prefix=/usr/local/Cherokee --with-mysql=/usr/local/mysql/
--with-pgsql=/usr/local/pgsql/
usuario@server:/usr/local/src/Cherokee-0.99.24$ make
usuario@server:/usr/local/src/Cherokee-0.99.24$ su root
server:/usr/local/src/Cherokee-0.99.24# make install
server:/usr/local/src/Cherokee-0.99.24# vi /etc/profile
export CHEROKEE_SBIN=/usr/local/Cherokee/sbin
export CHEROKEE_BIN=/usr/local/Cherokee/bin
$PATH=$PATH:$CHEROKEE_SBIN:$CHEROKEE_BIN
server:/usr/local/src/Cherokee-0.99.24# source /etc/profile
server:/usr/local/src/Cherokee-0.99.24# $CHEROKEE_SBIN/Cherokee-admin
Login:
User:          admin
One-time Password: 9JpbDHHGUWbD0ajx
Web Interface:
URL:          http://127.0.0.1:9090/
server:/usr/local/src/Cherokee-0.99.24# cp contrib/Cherokee /etc/init.d/
server:/usr/local/src/Cherokee-0.99.24# chmod 755 /etc/init.d/Cherokee
server:/usr/local/src/Cherokee-0.99.24# ln -s /etc/init.d/Cherokee /etc/rc4.d/S20Cherokee
server:/usr/local/src/Cherokee-0.99.24# vi /etc/rc.local
/etc/init.d/Cherokee start
server:/usr/local/src/Cherokee-0.99.24# vi /usr/local/Cherokee/etc/Cherokee/Cherokee.conf
server!bind!2!port = 80
server!bind!2!port = 82
server:/usr/local/src/Cherokee-0.99.24#/etc/init.d/Cherokee start
server:/usr/local/src/Cherokee-0.99.24# exit
server:/usr/local/src/Cherokee-0.99.24# cd ..
~
```

Imagen 3.1. Ejecución del flujo de instalación del Servidor web Cherokee vía compilación de código fuente.

Copiar el archivo fuente del paquete a la ruta `/usr/local/src`.

```
usuario@server:~$ cp /home/usuario/Fuentes/Cherokee-0.99.24.tar.gz /usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$ cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$ tar -zxvf Cherokee-0.99.24.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$ cd Cherokee-0.99.24
```

Ejecutar el comando *configure* indicando la ruta de instalación de *cherokee* y las rutas donde se encuentran ubicadas las instalaciones de *mysql* y *postgres*.

```
usuario@server:/usr/local/src/cherokee-0.99.24$ ./configure --  
prefix=/usr/local/cherokee --with-mysql=/usr/local/mysql/ --with-  
pgsql=/usr/local/pgsql/
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/cherokee-0.99.24$make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/cherokee-0.99.24$ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/cherokee-0.99.24#make install
```

Utilizar el editor de textos de preferencia para agregar al archivo *profile* las variables de ambiente que contienen las rutas de los binarios de *cherokee* y agregarlas al *PATH* del sistema.

```
server:/usr/local/src/cherokee-0.99.24# vi /etc/profile  
export CHEROKEE_SBIN=/usr/local/cherokee/sbin  
export CHEROKEE_BIN=/usr/local/cherokee/bin  
$PATH=$PATH:$CHEROKEE_SBIN:$CHEROKEE_BIN
```

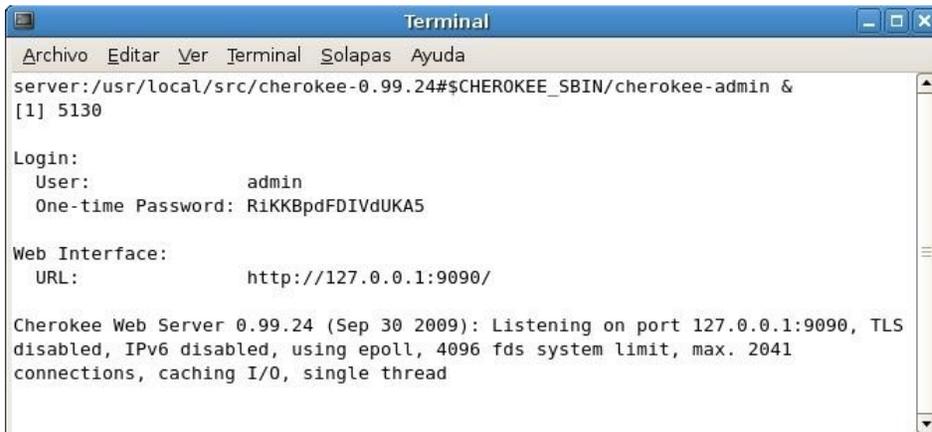
Actualizar las variables de ambiente en la consola de comandos ejecutando el comando *source*.

```
server:/usr/local/src/cherokee-0.99.24# source /etc/profile
```

Ejecutar el comando *cherokee-admin* en segundo plano para obtener los datos de configuración de *cherokee*. En la imagen 3.2 se puede apreciar el resultado de la ejecución del comando.

```
server:/usr/local/src/cherokee-0.99.24#$CHEROKEE_SBIN/cherokee-admin  
&
```

Abrir un navegador web e Ingresar la ruta <http://localhost:9999/> para ingresar a la pagina web de configuración de *cherokee*. En la imagen 3.3 se puede apreciar la pagina web de configuración del servidor *cherokee*.



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
server:/usr/local/src/cherokee-0.99.24#$CHEROKEE_SBIN/cherokee-admin &
[1] 5130

Login:
User:          admin
One-time Password: RiKKBpdFDIVdUKA5

Web Interface:
URL:          http://127.0.0.1:9090/

Cherokee Web Server 0.99.24 (Sep 30 2009): Listening on port 127.0.0.1:9090, TLS
disabled, IPv6 disabled, using epoll, 4096 fds system limit, max. 2041
connections, caching I/O, single thread
```

Imagen 3.2. Activación del portal de configuración del Servidor web cherokee.

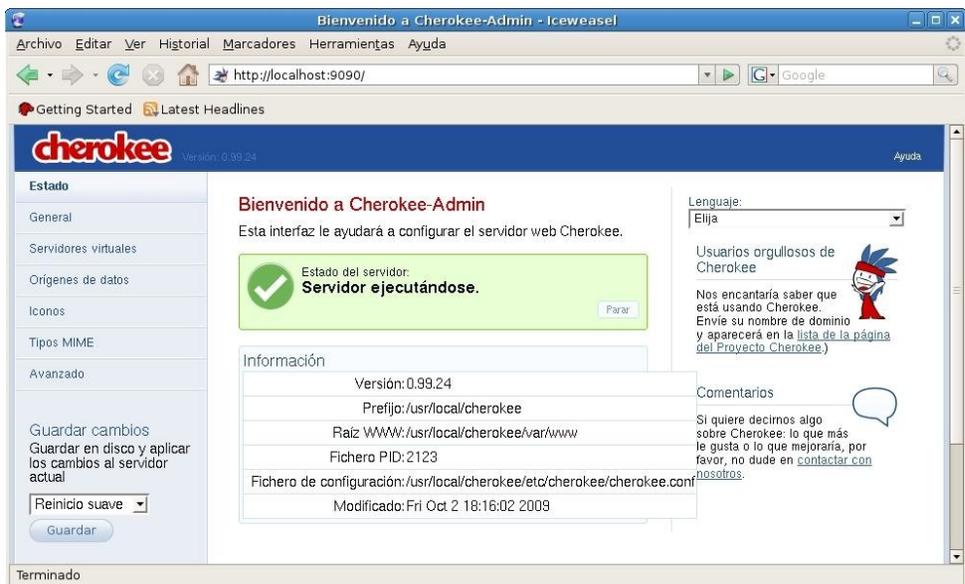


Imagen 3.3 Portal de configuración del Servidor web cherokee.

Copiar el archivo *cherokee* que se encuentra bajo la ruta *contrib/* a la ruta */etc/init.d/* con el mismo nombre.

```
server:/usr/local/src/cherokee-0.99.24# cp contrib/cherokee /etc/init.d/
```

Asignar los permisos 755 al archivo copiado.

```
server:/usr/local/src/cherokee-0.99.24# chmod 755 /etc/init.d/cherokee
```

Crear el enlace simbólico del archivo *cherokee* a la ruta */etc/rc4.d/S20cherokee*.

```
server:/usr/local/src/cherokee-0.99.24# ln -s /etc/init.d/cherokee  
/etc/rc4.d/S20cherokee
```

Utilizar el editor de textos de preferencia para agregar al archivo *rc.local* el comando que iniciara el servicio *cherokee* al cargar el sistema.

```
server:/usr/local/src/cherokee-0.99.24# vi /etc/rc.local  
  
/etc/init.d/cherokee start
```

Utilizar el editor de textos para modificar el archivo *cherokee.conf* y cambiar el puerto donde escuchara el servidor las peticiones.

```
server:/usr/local/src/cherokee-0.99.24#vi  
/usr/local/cherokee/etc/cherokee/cherokee.conf
```

Buscar la línea *server!bind!2!port = 80* y sustituye el valor 80 por el número de puerto que desee y se encuentre libre.

```
server!bind!2!port = 80  
  
server!bind!2!port = 82
```

Ejecutar el comando *cherokee start* para levantar el servidor.

```
server:/usr/local/src/cherokee-0.99.24#/etc/init.d/Cherokee start
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/cherokee-0.99.24#exit
```

Regresar a la carpeta *src*.

```
server:/usr/local/src/cherokee-0.99.24$cd ..
```

Abrir un navegador web e Ingresar la ruta <http://localhost:82> para comprobar que el servidor se haya levantado correctamente. En la imagen 3.4 se puede apreciar el contenido de la pagina web por Default del servidor web *cherokee*.

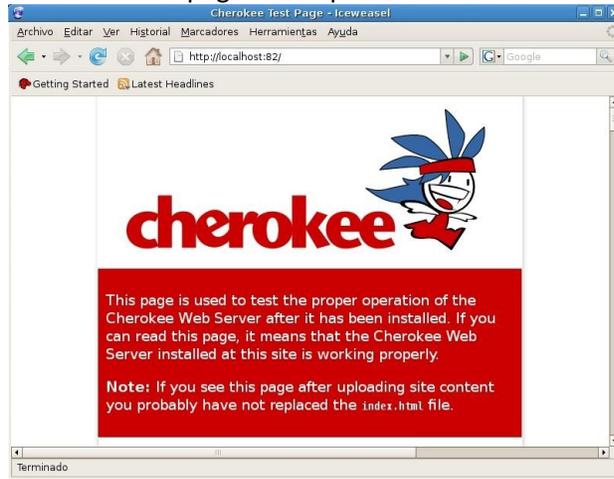


Imagen 3.4 Visualización de la pagina de inicio de prueba del Servidor web cherokee .

3.2 APACHE

Es un servidor web que implementa el protocolo HTTP, es de código abierto y funciona sobre plataformas Unís, Windows y Macintosh entre otras. Actualmente se desarrolla dentro del proyecto http Server de la Apache Software Foundation. Dentro de sus principales características destacan las siguientes: es altamente configurable, soporte a bases de datos de autenticación y negociado de contenido, soporte a virtualización de sitios. Apache esta estructurado en módulos, la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo, los módulos se clasifican en tres: Módulos base, módulos multiproceso y módulos adicionales. Inicialmente el proyecto se llamaba apache y luego a la versión 1.3.42, a partir de la versión 2.0 se llama http Server Project.

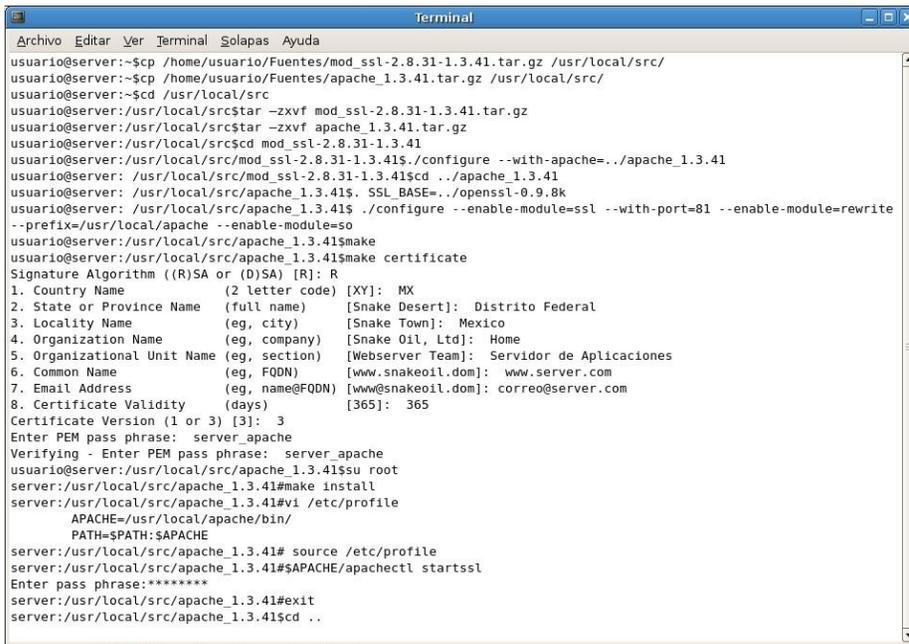
Tipo de licencia: OpenSource.

Servidor APACHE 1.3.x

Versión utilizada: *apache_1.3.41.tar.gz , mod_ssl-2.8.31-1.3.41.tar.gz*

URL de descarga: <http://httpd.apache.org/download.cgi>,
<http://www.modssl.org/source/>

Indicaciones previas: La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 3.5 se puede apreciar el flujo de comandos de la instalación.



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cp /home/usuario/Fuentes/mod_ssl-2.8.31-1.3.41.tar.gz /usr/local/src/
usuario@server:~$cp /home/usuario/Fuentes/apache_1.3.41.tar.gz /usr/local/src/
usuario@server:~$cd /usr/local/src
usuario@server:~/usr/local/src$tar -zxvf mod_ssl-2.8.31-1.3.41.tar.gz
usuario@server:~/usr/local/src$tar -zxvf apache_1.3.41.tar.gz
usuario@server:~/usr/local/src$cd mod_ssl-2.8.31-1.3.41
usuario@server:~/usr/local/src/mod_ssl-2.8.31-1.3.41$./configure --with-apache=../apache_1.3.41
usuario@server:~/usr/local/src/mod_ssl-2.8.31-1.3.41$cd ../apache_1.3.41
usuario@server:~/usr/local/src/apache_1.3.41$ SSL_BASE=../openssl-0.9.8k
usuario@server:~/usr/local/src/apache_1.3.41$ ./configure --enable-module=ssl --with-port=81 --enable-module=rewrite
--prefix=/usr/local/apache --enable-module=so
usuario@server:~/usr/local/src/apache_1.3.41$make
usuario@server:~/usr/local/src/apache_1.3.41$make certificate
Signature Algorithm ((R)SA or (D)SA) [R]: R
1. Country Name (2 letter code) [XY]: MX
2. State or Province Name (full name) [Snake Desert]: Distrito Federal
3. Locality Name (eg, city) [Snake Town]: Mexico
4. Organization Name (eg, company) [Snake Oil, Ltd]: Home
5. Organizational Unit Name (eg, section) [Webserver Team]: Servidor de Aplicaciones
6. Common Name (eg, FQDN) [www.snakeoil.dom]: www.server.com
7. Email Address (eg, name@FQDN) [www@snakeoil.dom]: correo@server.com
8. Certificate Validity (days) [365]: 365
Certificate Version (1 or 3) [3]: 3
Enter PEM pass phrase: server apache
Verifying - Enter PEM pass phrase: server apache
usuario@server:~/usr/local/src/apache_1.3.41$su root
server:~/usr/local/src/apache_1.3.41$make install
server:~/usr/local/src/apache_1.3.41#vi /etc/profile
APACHE=/usr/local/apache/bin/
PATH=$PATH:$APACHE
server:~/usr/local/src/apache_1.3.41# source /etc/profile
server:~/usr/local/src/apache_1.3.41#$APACHE/apachectl startssl
Enter pass phrase:*****
server:~/usr/local/src/apache_1.3.41#exit
server:~/usr/local/src/apache_1.3.41$cd ..
```

Imagen 3.5. Ejecución del flujo de instalación del Servidor web apache vía compilación del código fuente.

Copiar el archivo fuente del paquete `mod_ssl-2.8.31-1.3.41.tar.gz` y `apache_1.3.41.tar.gz` a la ruta `/usr/local/src`.

```
usuario@server:~$cp /home/usuario/Fuentes/mod_ssl-2.8.31-1.3.41.tar.gz /usr/local/src/
```

```
usuario@server:~$cp /home/usuario/Fuentes/apache_1.3.41.tar.gz /usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar los archivos fuente.

```
usuario@server:/usr/local/src$tar -zxvf mod_ssl-2.8.31-1.3.41.tar.gz
usuario@server:/usr/local/src$tar -zxvf apache_1.3.41.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$cd mod_ssl-2.8.31-1.3.41
```

Ejecutar el comando *configure* indicando la ubicación de los archivos fuente del servidor apache.

```
usuario@server:/usr/local/src/mod_ssl-2.8.31-1.3.41$./configure --with-
apache=../apache_1.3.41
```

Ingresar a la ubicación de los archivos fuente de *apache*.

```
usuario@server:/usr/local/src/mod_ssl-2.8.31-1.3.41$cd ../apache_1.3.41
```

Declarar la variable de ambiente *SSL_BASE* y asígnale la ruta de la ubicación de los archivos fuente de *Openssl* instalado previamente.

```
usuario@server:/usr/local/src/apache_1.3.41$. SSL_BASE=../openssl-
0.9.8k
```

Ejecutar el comando *configure* activando el modulo de *ssl*, *rewrite* y *so* para módulos compartidos, indicando el puerto donde el servidor apache atenderá las peticiones y por ultimo indica la ubicación de la instalación de los binarios.

```
usuario@server:/usr/local/src/apache_1.3.41$. ./configure --enable-
module=ssl --with-port=81 --enable-module=rewrite --
prefix=/usr/local/apache --enable-module=so
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/apache_1.3.41$make
```

Ejecutar el comando *make certificate* para generar el certificado *ssl* de apache.

```
usuario@server:/usr/local/src/apache_1.3.41$make certificate
```

Ingresar la letra *R* para indicar que se requiere utilizar el algoritmo de encriptación RSA.

```
Signature Algorithm ((R)SA or (D)SA) [R]: R
```

Ingresa los valores indicados para cada uno de los 8 registros solicitados.

1. *Country Name (2 letter code) [XY]:* MX
2. *State or Province Name (full name) [Snake Desert]:* Distrito Federal
3. *Locality Name (eg, city) [Snake Town]:* México
4. *Organization Name (eg, company) [Snake Oil, Ltd]:* Home
5. *Organizational Unit Name (eg, section) [Webserver Team]:* Servidor de Aplicaciones
6. *Common Name (eg, FQDN) [www.snakeoil.dom]:* www.server.com
7. *Email Address (eg, name@FQDN) [www@snakeoil.dom]:*
correo@server.com
8. *Certificate Validity (days) [365]:* 365

Selecciona la versión del certificado.

Certificate Version (1 or 3) [3]: 3

Ingresa y confirma el password del certificado.

Enter PEM pass phrase: server_apache

Verifying - Enter PEM pass phrase: server_apache

Después de configurar la instalación y generar el certificado, iniciar sesión en la consola de comandos como usuario root.

*usuario@server:/usr/local/src/apache_1.3.41\$*su root

Ejecutar el comando *make install*.

*server:/usr/local/src/apache_1.3.41#*make install

Utilizar el editor de textos de preferencia para agregar al archivo *profile* la variable de ambiente que contiene la ruta de acceso a los binarios de apache.

*server:/usr/local/src/apache_1.3.41#*vi /etc/profile

APACHE=/usr/local/apache/bin/

PATH=\$PATH:\$APACHE

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando `source`.

```
server:/usr/local/src/apache_1.3.41# source /etc/profile
```

Ejecutar el comando `apachectl startssl` para levantar el servidor apache con el protocolo `https`.

```
server:/usr/local/src/apache_1.3.41#$APACHE/apachectl startssl
```

Ingresa el password para `openssl`.

```
Enter pass phrase:*****
```

Cerrar la sesión de `root` al terminar la ejecución del comando anterior.

```
server:/usr/local/src/apache_1.3.41#exit
```

Regresar a la carpeta `src`.

```
server:/usr/local/src/apache_1.3.41$cd ..
```

Abre un navegador web e ingresa la ruta <https://localhost> (imagen 3.6) para comprobar que el servidor se haya levantado correctamente.

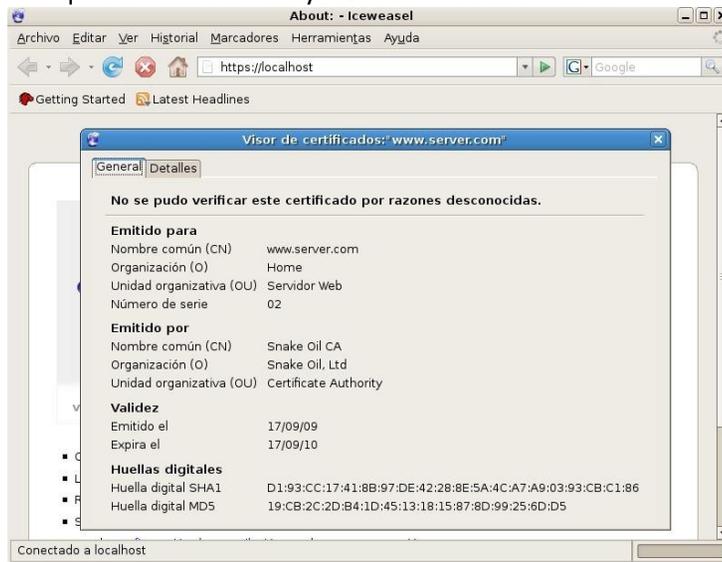


Imagen 3.6. Visor de certificado al levantar el servicio apache con ssl.

Servidor HTTPD 2.2.x

Versión utilizada: `httpd-2.2.11.tar.gz`

URL de descarga: <http://httpd.apache.org/download.cgi>

Indicaciones previas: La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 3.7 se puede apreciar el flujo de los comandos de la instalación.

Copiar el archivo fuente del servidor HTTPD a la ruta `/usr/local/src`.

```
usuario@server:~$cp /home/usuario/Fuentes/httpd-2.2.11.tar.gz  
/usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$ tar -zxvf httpd-2.2.11.tar.gz
```



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cp /home/usuario/Fuentes/httpd-2.2.11.tar.gz /usr/local/src/
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$ tar -zxvf httpd-2.2.11.tar.gz
usuario@server:/usr/local/src$ cd httpd-2.2.11
usuario@server:/usr/local/src/httpd-2.2.11$./configure --with-perl --enable-module=ssl --enable-rewrite=shared
--prefix=/usr/local/apache2 --enable-module=so --enable-module=all
usuario@server:/usr/local/src/httpd-2.2.11$make
usuario@server:/usr/local/src/httpd-2.2.11$su root
server:/usr/local/src/httpd-2.2.11#make install
server:/usr/local/src/httpd-2.2.11#v1 /etc/profile
APACHE2=/usr/local/apache2/bin/
PATH=$PATH:$APACHE2
server:/usr/local/src/httpd-2.2.11#source /etc/profile
server:/usr/local/src/httpd-2.2.11#$APACHE2/apachectl startssl
server:/usr/local/src/httpd-2.2.11#v1 /etc/rc.local
/usr/local/apache2/bin/apachectl start
server:/usr/local/src/httpd-2.2.11#exit
server:/usr/local/src/httpd-2.2.11$cd ..
```

Imagen 3.7. Ejecución del flujo de instalación del Servidor web HTTPD.

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$ cd httpd-2.2.11
```

Ejecutar el comando `configure` indicando la activación de los módulos `perl`, `rewrite` y `ssl`, indicar también la ubicación de la instalación de los archivos binarios.

```
usuario@server:/usr/local/src/httpd-2.2.11$./configure --with-perl --  
enable-module=ssl --enable-rewrite=shared --prefix=/usr/local/apache2 --  
enable-module=so --enable-module=all
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/httpd-2.2.11$make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/httpd-2.2.11$su root
```

Ejecutar el comando *make install* para realizar la instalación.

```
server:/usr/local/src/httpd-2.2.11#make install
```

Utilizar el editor de textos de preferencia para agregar al archivo *profile* la variable de ambiente que contiene la ruta de acceso a los binarios del servidor *httpd*.

```
server:/usr/local/src/httpd-2.2.11#vi /etc/profile  
  
APACHE2=/usr/local/apache2/bin/  
PATH=$PATH:$APACHE2
```

Actualizar las variables de ambiente en la consola de comandos Ejecutando el comando *source*.

```
server:/usr/local/src/httpd-2.2.11#source /etc/profile
```

Ejecutar el comando *apachectl start* para levantar el servidor *httpd*.

```
server:/usr/local/src/httpd-2.2.11#$APACHE2/apachectl startssl
```

Utilizar el editor de textos para agregar al archivo *rc.local* el comando que iniciara el servicio *httpd* al cargar el sistema. En la imagen 3.8 se puede apreciar la página por *Default* del servidor *HTTPD*.

```
server:/usr/local/src/httpd-2.2.11#vi /etc/rc.local  
  
/usr/local/apache2/bin/apachectl start
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/httpd-2.2.11#exit
```

Regresar a la carpeta `src`.

```
usuario@server:/usr/local/src/httpd-2.2.11$cd ..
```

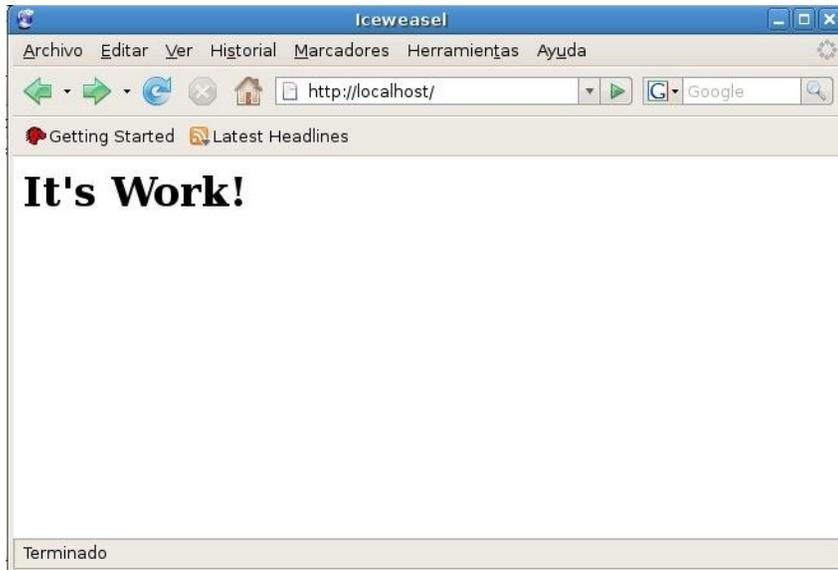


Imagen 3.8. Página de prueba del Servidor HTTPD.

El tener instalado diferentes opciones de servidores web permite probar las aplicaciones que se estén desarrollando bajo diferentes plataformas y así se mejorara las metodologías y técnicas de desarrollo que se estén utilizando. El servidor web cherokee ofrece funciones para realizar la configuración con gran facilidad aún siendo principiantes y también provee la extensión de realizar configuraciones más complejas si así se requiere. Se realizo la instalación de la versión 1.3.x de apache ya que tienen la característica de poder implementar el protocolo HTTPS, utilizando para ello el módulo de MODSSL. Actualmente la gran mayoría de los proveedores de hosting cuentan con esta versión de apache en sus servidores, por lo que se requiere seguir probando esta versión del servidor web con los desarrollos de aplicaciones web. La nueva versión del servidor web apache llamada HTTPD cuenta con grandes mejoras en rendimiento y funcionalidades agregadas como el manejo de rutas amigables y soporte mejorado en el manejo de host virtuales.

Capítulo 4 Lenguajes de programación para aplicaciones Desktop y Web

Con el desarrollo y avance de la web 2.0, la tendencia es migrar al desarrollo de aplicaciones web, no dejando atrás el desarrollo aún necesario de aplicaciones de escritorio, el lenguaje de programación JAVA y RUBY ofrecen la posibilidad de realizar aplicaciones web y de escritorio, PHP por su parte sólo ofrece la opción de desarrollo web y es el lenguaje más popular y más usado actualmente por su facilidad de uso y la posibilidad de desarrollar aplicaciones y sitios web con un buen desempeño. RUBY es un lenguaje relativamente nuevo que esta tomando cada vez más fama entre los desarrolladores que trabajan bajo la filosofía de software libre y JAVA por su parte es usado más en el ámbito empresarial por el soporte que ofrece y la robustez en aplicaciones que necesitan el manejo de grandes cantidades de transacciones.

4.1 JAVA

Lenguaje de programación orientado a objetos, multiplataforma y multiprocesos, gran parte de su sintaxis esta basada en C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel que suelen producir errores como la manipulación directa de punteros o memoria. La plataforma Java consta de las siguientes partes:

- El lenguaje de programación mismo.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

Java tiene diferentes plataformas como son Java SE, Java ME y Java EE. Java EE es un estándar para el desarrollo de aplicaciones empresariales las cuales necesitan ser robustas, escalables y seguras, Java EE es una especificación y no se debe confundir con un producto.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: `java_ee_sdk-5_07-jdk-6u16-linux-ml.bin`

URL de descarga:

<http://java.sun.com/javase/downloads/index.jsp?userOsIndex=1&userOsId=linux&userOsName=Linux>

Indicaciones previas: La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.

Copiar el archivo binario de instalación a la ruta `/usr/local/src`.

```
usuario@server:~$scp /home/usuario/Fuentes/java_ee_sdk-5_07-jdk-6u16-linux-ml.bin /usr/local/src/
```

Ingresa a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Modificar los permisos del binario y asignarle todos los niveles 777.

```
usuario@server:/usr/local/src$chmod 777 java_ee_sdk-5_07-jdk-6u16-linux-ml.bin
```

Iniciar sesión en la consola de comandos como usuario root.

```
usuario@server:/usr/local/src$su root
```

Ejecutar el binario de instalación.

```
server:/usr/local/src#./java_ee_sdk-5_07-jdk-6u16-linux-ml.bin
```

Seleccionar la opción **Next** (imagen 4.1).



Imagen 4.1. Wizard de instalación de JAVA EE.

Aceptar los términos de uso y selecciona la opción **Next** (imagen 4.2).

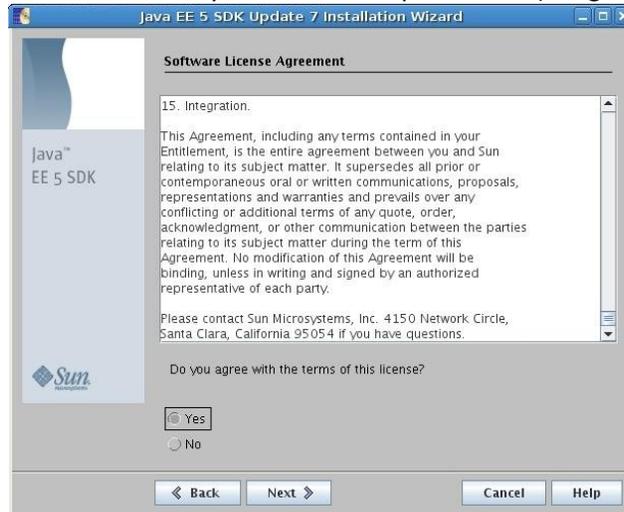


Imagen 4.2. Wizard de instalación de JAVA EE.

Ingresa la ruta de instalación y selecciona la opción **Next** (imagen 4.3).



Imagen 4.3. Wizard de instalación de JAVA EE.

Selecciona la opción *Create Directory* (imagen 4.4).



Imagen 4.4. Wizard de instalación de JAVA EE.

Ingresar los datos del usuario y modifica los puertos si es necesario y selecciona la opción *Next* (imagen 4.5).



Imagen 4.5. Wizard de instalación de JAVA EE.

Desactivar ambas casillas y selecciona la opción *Next* (imagen 4.6).

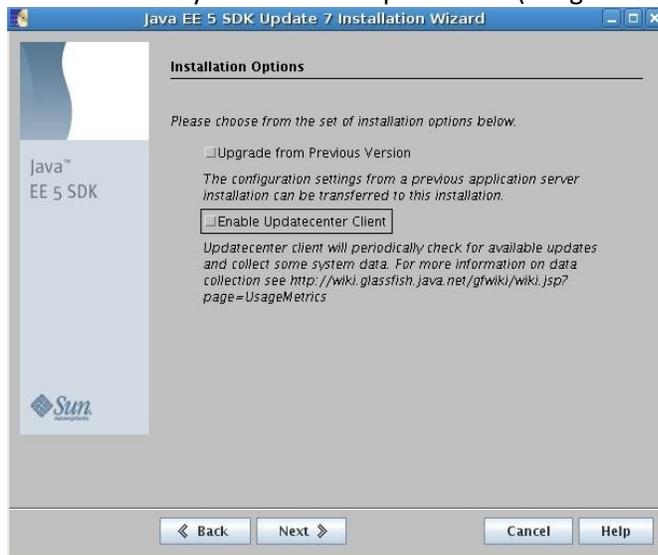


Imagen 4.6. Wizard de instalación de JAVA EE.

Seleccionar la opción *Install Now* (imagen 4.7).

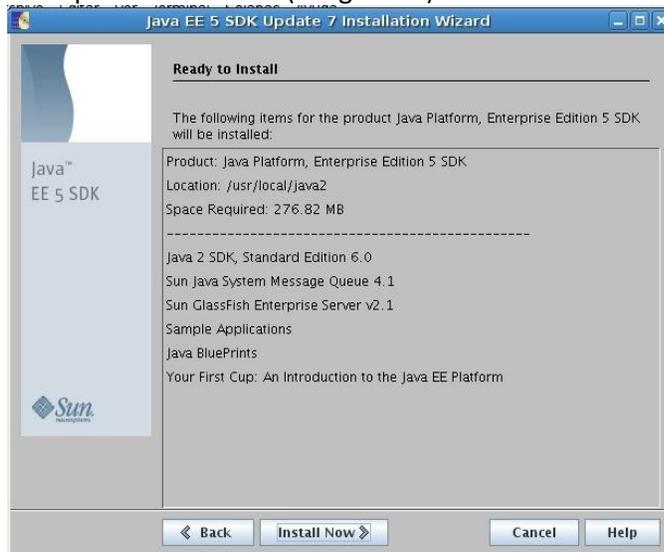


Imagen 4.7. Wizard de instalación de JAVA EE.

Seleccionar la opción *Skip Registration* y después la opción *Next* (imagen 4.8).



Imagen 4.8. Wizard de instalación de JAVA EE.

Seleccionar la opción *Finish* (imagen 4.9).



Imagen 4.9. Wizard de instalación de JAVA EE.

Utilizar el editor de textos de preferencia para agregar al archivo *profile* las variables de ambiente que contienen las rutas de acceso a los binarios de *java*.

```
server:/usr/local/src#vi /etc/profile  
  
export JAVA_HOME=/usr/local/java/jdk  
  
export JAVA_BIN=/usr/local/java/jdk/bin  
  
export PATH=$PATH:$JAVA_HOME:$JAVA_BIN
```

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando *source*.

```
server:/usr/local/src#source /etc/profile
```

Eliminar todos los enlaces simbólicos del comando *java*.

```
server:/usr/local/src#update-alternatives --remove-all java
```

Verificar que no existan los enlaces simbólicos del comando *java*.

```
server:/usr/local/src#update-alternatives --list java
```

Instalar el enlace simbólico de la ubicación de los binarios de *java* instalados previamente.

```
server:/usr/local#update-alternatives --install "/usr/bin/java" "java"  
"/usr/local/java/jdk/bin/java" 1
```

Verificar que exista el enlace simbólico al comando *java*.

```
server:/usr/local#update-alternatives --display java
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local#exit
```

4.2 TOMCAT

Su principal función es ser contenedor de servlets y JSPs. Tomcat no es un servidor de aplicaciones, tiene incluido el compilador Jasper, el cual convierte los JSPs en servlets. Es muy común encontrar trabajando el contenedor de servlets en conjunto con el servidor web Apache. Tomcat puede funcionar como servidor web por si mismo, pero se recomienda utilizarlo bajo este esquema sólo en ambientes de desarrollo. También es llamado Jakarta Tomcat o Apache Tomcat.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *apache-tomcat-6.0.20.tar.gz*

URL de descarga: <http://tomcat.apache.org/download-60.cgi>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 4.10 se puede apreciar el flujo de los comandos de instalación.*



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$scp /home/usuario/Fuentes/apache-tomcat-6.0.20.tar.gz /usr/local/src/
usuario@server:~$cd /usr/local/src
usuario@server: /usr/local/src$tar -zxvf apache-tomcat-6.0.20.tar.gz
usuario@server:/usr/local/src$mv apache-tomcat-6.0.20 ../tomcat
usuario@server:/usr/local/src$cd ../tomcat
usuario@server:/usr/local/tomcat$su root
server:/usr/local/tomcat#apt-get install automake
server:/usr/local/tomcat#vi /etc/profile
    export CATALINA_HOME=/usr/local/tomcat
    export CATALINA_BASE=/usr/local/tomcat
    export CATALINA_TMPDIR=/usr/local/tomcat/temp
PATH=$PATH:$CATALINA_HOME:$CATALINA_BASE
server:/usr/local/tomcat# source /etc/profile
server:/usr/local/tomcat#bin/startup.sh
server:/usr/local/tomcat#vi /etc/rc.local
    /usr/local/tomcat/bin/startup.sh
```

Imagen 4.10. Ejecución del flujo de instalación de TOMCAT.

Copiar el archivo fuente a la ruta */usr/local/src*.

```
usuario@server:~$scp /home/usuario/Fuentes/apache-tomcat-6.0.20.tar.gz /usr/local/src/
```

Ingresa a la ruta */usr/local/src*.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server: /usr/local/src$tar -zxvf apache-tomcat-6.0.20.tar.gz
```

Copiar la carpeta generada a la carpeta `/usr/local` renombrándola como `tomcat`.

```
usuario@server:/usr/local/src$mv apache-tomcat-6.0.20 ../tomcat
```

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$cd ../tomcat
```

Iniciar sesión en la consola de comandos como usuario `root`.

```
usuario@server:/usr/local/tomcat$su root
```

Instalar el comando `automake` por medio de `apt-get`.

```
server:/usr/local/tomcat#apt-get install automake
```

Utilizar el editor de textos de preferencia para agregar al archivo `profile` las variables de ambiente que contienen las rutas de acceso a los binarios de `tomcat`.

```
server:/usr/local/tomcat#vi /etc/profile
```

```
export CATALINA_HOME=/usr/local/tomcat
```

```
export CATALINA_BASE=/usr/local/tomcat
```

```
export CATALINA_TMPDIR=/usr/local/tomcat/temp
```

```
PATH=$PATH:$CATALINA_HOME:$CATALINA_BASE
```

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando `source`.

```
server:/usr/local/tomcat# source /etc/profile
```

Iniciar el servicio de `tomcat`.

```
server:/usr/local/tomcat#bin/startup.sh
```

Utilizar el editor de textos para agregar al archivo `rc.local` el comando que iniciara el servicio de `tomcat` al iniciar el sistema. En la imagen 4.11 se puede apreciar la pagina de prueba del servidor `Tomcat`.

```
server:/usr/local/tomcat#vi /etc/rc.local
```

```
/usr/local/tomcat/bin/startup.sh
```

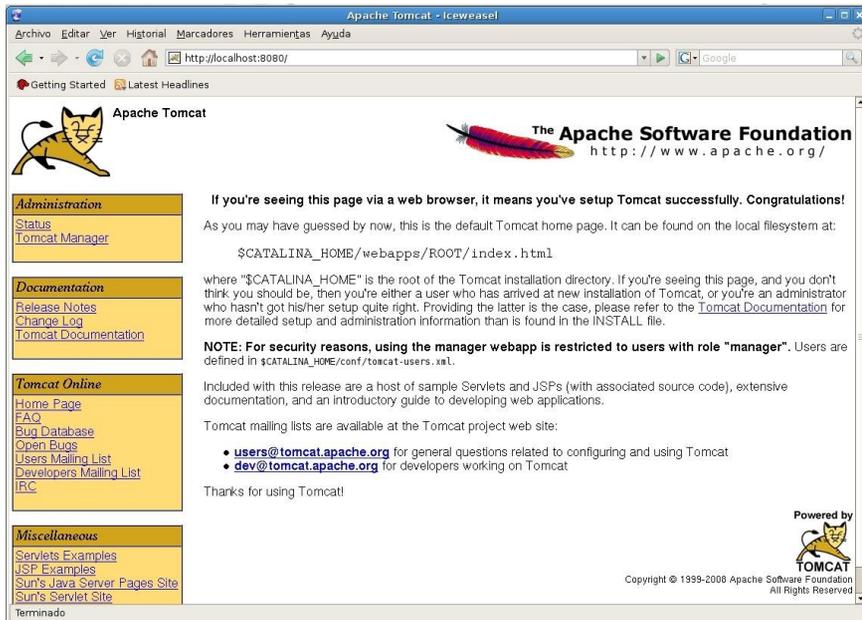


Imagen 4.11. Navegador web mostrando el servicio de Tomcat levantado en el puerto 8080.

Tomcat con Apache

Versión utilizada: *tomcat-connectors-1.2.28-src.tar.gz*

URL de descarga: <http://tomcat.apache.org/download-connectors.cgi?Preferred=http%3A%2F%2Fapache.mirrors.hoobly.com>

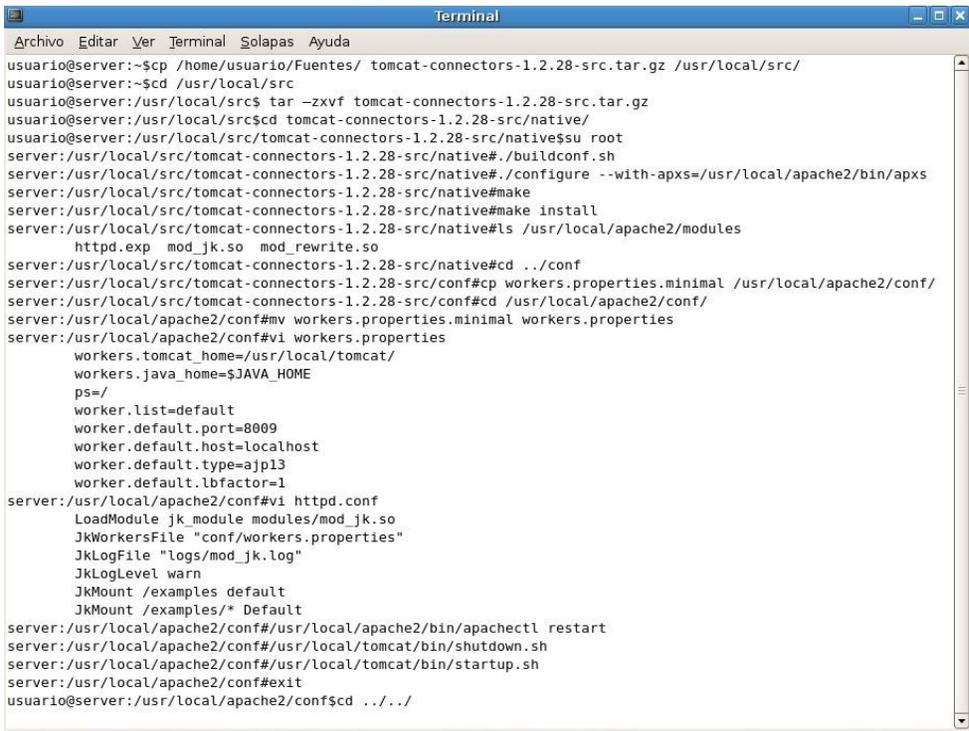
En la imagen 4.14 se puede apreciar el flujo de ejecución de los comandos de instalación.

Copiar el archivo fuente a la ruta */usr/local/src*.

```
usuario@server:~$cp /home/usuario/Fuentes/tomcat-connectors-1.2.28-src.tar.gz /usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$cd /usr/local/src
```



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$ scp /home/usuario/Fuentes/ tomcat-connectors-1.2.28-src.tar.gz /usr/local/src/
usuario@server:~$ cd /usr/local/src
usuario@server:~$ tar -zxvf tomcat-connectors-1.2.28-src.tar.gz
usuario@server:~$ cd tomcat-connectors-1.2.28-src/native/
usuario@server:~$ su root
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# ./buildconf.sh
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# ./configure --with-apxs=/usr/local/apache2/bin/apxs
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# make
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# make install
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# ls /usr/local/apache2/modules
httpd.exp  mod_jk.so  mod_rewrite.so
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# cd ../conf
server:/usr/local/src/tomcat-connectors-1.2.28-src/conf# cp workers.properties.minimal /usr/local/apache2/conf/
server:/usr/local/src/tomcat-connectors-1.2.28-src/conf# cd /usr/local/apache2/conf/
server:/usr/local/apache2/conf# mv workers.properties.minimal workers.properties
server:/usr/local/apache2/conf# vi workers.properties
workers.tomcat_home=/usr/local/tomcat/
workers.java_home=$JAVA_HOME
ps=/
worker.list=default
worker.default.port=8009
worker.default.host=localhost
worker.default.type=ajp13
worker.default.lbfactor=1
server:/usr/local/apache2/conf# vi httpd.conf
LoadModule jk_module modules/mod_jk.so
JkWorkersFile "conf/workers.properties"
JkLogFile "logs/mod_jk.log"
JkLogLevel warn
JkMount /examples default
JkMount /examples/* Default
server:/usr/local/apache2/conf# /usr/local/apache2/bin/apachectl restart
server:/usr/local/apache2/conf# /usr/local/tomcat/bin/shutdown.sh
server:/usr/local/apache2/conf# /usr/local/tomcat/bin/startup.sh
server:/usr/local/apache2/conf# exit
usuario@server:~$ cd ../../
```

Imagen 4.12. Ejecución del flujo de instalación del conector Tomcat.

Ingresar a la carpeta *native* que se encuentra dentro de la carpeta fuente.

```
usuario@server:~$ cd tomcat-connectors-1.2.28-src/native/
```

Iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:~$ su root
root
```

Ejecutar el script *buildconf.sh*

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# ./buildconf.sh
```

Ejecutar el comando *configure* indicando la ubicación del comando *apxs* del servidor *httpd*.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/native# ./configure --with-apxs=/usr/local/apache2/bin/apxs
```

Ejecutar el comando *make*.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/native#make
```

Ejecutar el comando *make install* para realizar la instalación.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/native#make install
```

Verificar que se genere el archivo *mod_jk.so* en la carpeta de módulos de apache que es el módulo de conexión entre *tomcat* y *apache*.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/native#ls  
/usr/local/apache2/modules  
httpd.exp mod_jk.so mod_rewrite.so
```

Ingresar a la carpeta *conf* que se encuentra un nivel arriba.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/native#cd ../conf
```

Copiar el archivo *workers.properties.minimal* a la carpeta de configuración del servidor *httpd*.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/conf#cp  
workers.properties.minimal /usr/local/apache2/conf/
```

Ingresar a la carpeta de configuración del servidor *httpd*.

```
server:/usr/local/src/tomcat-connectors-1.2.28-src/conf#cd  
/usr/local/apache2/conf/
```

Renombrar el archivo *workers.properties.minimal* y asignarle el nombre de *workers.properties*.

```
server:/usr/local/apache2/conf#mv workers.properties.minimal  
workers.properties
```

Utilizar el editor de textos para modificar el archivo *workers.properties*.

```
server:/usr/local/apache2/conf#vi workers.properties
```

Buscar la sección *Defining a worker named ajp13w and of type ajp13* y agregar los siguientes comandos.

```
workers.tomcat_home=/usr/local/tomcat/  
workers.java_home=$JAVA_HOME
```

```
ps=/  
worker.list=default  
worker.default.port=8009  
worker.default.host=localhost  
worker.default.type=ajp13  
worker.default.lbfactor=1
```

Utilizar el editor de textos para agregar al archivo *httpd.conf* el módulo de tomcat.

```
server:/usr/local/apache2/conf#vi httpd.conf
```

Buscar la sección *LoadModules* y agregar la línea *LoadModule jk_module modules/mod_jk.so*.

```
LoadModule jk_module modules/mod_jk.so
```

Al final del archivo agregar las siguientes líneas.

```
JkWorkersFile "conf/workers.properties"  
JkLogFile "logs/mod_jk.log"  
JkLogLevel warn  
JkMount /examples default  
JkMount /examples/* Default
```

Reiniciar el servidor *apache* y *tomcat*. En la imagen 4.13 se puede apreciar la ejecución de un archivo *JSP* bajo el puerto 80 que es utilizado por el servidor *HTTPD*.

```
server:/usr/local/apache2/conf#/usr/local/apache2/bin/apachectl restart  
server:/usr/local/apache2/conf#/usr/local/tomcat/bin/shutdown.sh  
server:/usr/local/apache2/conf#/usr/local/tomcat/bin/startup.sh
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/apache2/conf#exit
```

Regresar a la carpeta *local*.

```
usuario@server:/usr/local/apache2/conf$cd ../../
```



Imagen 4.13. Ejecución de servlet desde el puerto 80 a través de HTTPD.

4.3 PHP

Es un lenguaje de programación interpretado de alto nivel que es incrustado en código HTML del lado del servidor. El gran parecido que existe con los lenguajes de programación estructurada como C o Perl, permite que los desarrolladores reduzcan la curva de aprendizaje de este lenguaje y realicen más rápidamente páginas web dinámicas. Sus principales características son: multiplataforma, conexión con la mayoría de los gestores de bases de datos, permite aplicar técnicas de programación orientada a objetos, biblioteca de funciones nativa amplia y bien documentada, no requiere definición de tipos de variables, tiene manejo de excepciones y fácil implementación del modelo de programación web MVC. Con PHP se puede hacer cualquier cosa que se pueda realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *php-5.3.0.tar.gz*

URL de descarga: <http://mx.php.net/get/php-5.3.0.tar.gz/from/a/mirror>

Indicaciones previas: La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root.

Copiar el archivo fuente a la ruta `/usr/local/src`.

```
usuario@server:~$scp /home/usuario/Fuentes/php-5.3.0.tar.gz  
/usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$tar -zxvf php-5.3.0.tar.gz
```

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$cd php-5.3.0
```

PHP en APACHE

En la imagen 4.14 se puede apreciar el flujo de comandos de instalación de PHP como módulo de APACHE.



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$tar -zxvf php-5.3.0.tar.gz
usuario@server:/usr/local/src$cd php-5.3.0
usuario@server:/usr/local/src/php-5.3.0$./configure --prefix=/usr/local/php_apache --with-apxs=/usr/local/apache
/bin/apxs --with-config-file-path=/usr/local/php_apache --enable-sockets --with-mysql=/usr/local/mysql --with-pg
sql=/usr/local/pgsql --enable-calendar --with-zlib-dir=/usr/local --with-gd=/usr/local/gd --with-png-dir=/usr/in
clude/libpng12 --with-freetype-dir=/usr --with-jpeg-dir=/usr
usuario@server:/usr/local/src/php-5.3.0$make
usuario@server:/usr/local/src/php-5.3.0$make test
usuario@server:/usr/local/src/php-5.3.0$su root
server:/usr/local/src/php-5.3.0#make install
server:/usr/local/src/php-5.3.0#cp php.ini-dist /usr/local/php_apache/lib/php.ini.
server:/usr/local/src/php-5.3.0#vi /usr/local/apache/conf/httpd.conf
LoadModule php5_module          libexec/libphp5.so
AddType application/x-httpd-php .php .html .htm .html
AddType application/x-httpd-php-source .phps
AddType image/x-icon .ico
UserDir public_html htdocs
server:/usr/local/src/php-5.3.0#./usr/local/apache/bin/apachectl restart
server:/usr/local/src/php-5.3.0#vi /usr/local/apache/htdocs/index.php
<? phpinfo(); ?>
server:/usr/local/src/php-5.3.0#exit
```

Imagen 4.14. Ejecución del flujo de instalación de PHP como modulo del Servidor web Apache.

Ejecutar el comando *configure* indicando la ubicación de instalación de los archivos binarios, el comando *apxs* de *apache*, la ruta de instalación de *mysql* y *postgres*.

```
usuario@server:/usr/local/src/php-5.3.0$./configure --  
prefix=/usr/local/php_apache --with-apxs=/usr/local/apache/bin/apxs --  
with-config-file-path=/usr/local/php_apache --enable-sockets --with-  
mysql=/usr/local/mysql --with-pgsql=/usr/local/pgsql --enable-calendar --  
with-zlib-dir=/usr/local --with-gd=/usr/local/gd --with-png-  
dir=/usr/include/libpng12 --with-freetype-dir=/usr --with-jpeg-dir=/usr
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/php-5.3.0$make
```

Ejecutar el comando *make test* para comprobar la configuración.

```
usuario@server:/usr/local/src/php-5.3.0$make test
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/php-5.3.0$su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/php-5.3.0#make install
```

Copiar el archivo *php.ini-dist* a la ruta */usr/local/php_apache/lib/php.ini*.

```
server:/usr/local/src/php-5.3.0#cp php.ini-dist  
/usr/local/php_apache/lib/php.ini.
```

Utilizar el editor de textos de preferencia para modificar el archivo *httpd.conf* .

```
server:/usr/local/src/php-5.3.0#vi /usr/local/apache/conf/httpd.conf
```

Buscar la sección de *LoadModule* y Agregar la siguiente línea.

```
LoadModule php5_module libexec/libphp5.so
```

Buscar la sección *AddTypeAddType* y Agregar las siguientes líneas.

```
AddType application/x-httpd-php .php .phtml .htm .html  
AddType application/x-httpd-php-source .phps
```

```
AddType image/x-icon .ico
```

```
UserDir public_html htdocs
```

Reiniciar el servidor *apache* para que los cambios en el archivo tengan efecto.

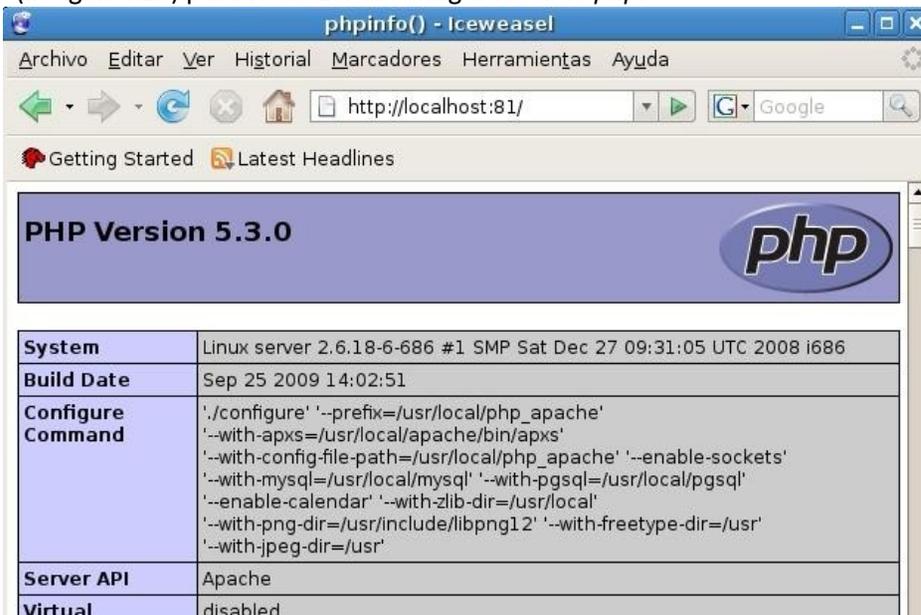
```
server:/usr/local/src/php-5.3.0#usr/local/apache/bin/apachectl restart
```

Utilizar el editor de textos para crear un archivo con extensión php y así verificar que la instalación sea correcta.

```
server:/usr/local/src/php-5.3.0#vi /usr/local/apache/htdocs/index.php
```

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

Ingresar a un navegador web e introduce la dirección <http://localhost:81> (imagen 4.15) para visualizar la configuración de *php*.



System	Linux server 2.6.18-6-686 #1 SMP Sat Dec 27 09:31:05 UTC 2008 i686
Build Date	Sep 25 2009 14:02:51
Configure Command	'./configure' '--prefix=/usr/local/php_apache' '--with-apxs=/usr/local/apache/bin/apxs' '--with-config-file-path=/usr/local/php_apache' '--enable-sockets' '--with-mysql=/usr/local/mysql' '--with-pgsql=/usr/local/pgsql' '--enable-calendar' '--with-zlib-dir=/usr/local' '--with-png-dir=/usr/include/libpng12' '--with-freetype-dir=/usr' '--with-jpeg-dir=/usr'
Server API	Apache
Virtual	disabled

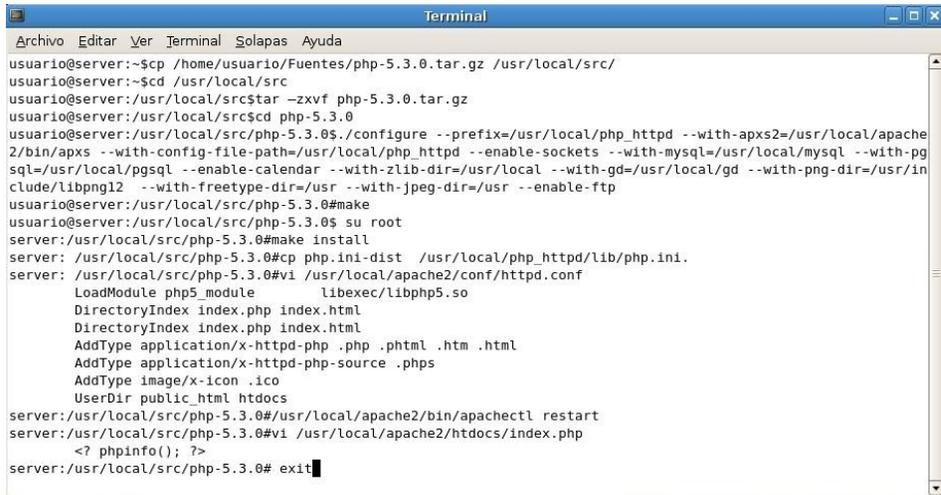
Imagen 4.15. Visualización de página de PHP sobre el Sobre el servidor web Apache.

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/php-5.3.0#exit
```

PHP con HTTPD

En la imagen 4.16 se puede apreciar el flujo de los comandos de instalación de PHP como módulo del servidor web HTTPD.



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$ cp /home/usuario/Fuentes/php-5.3.0.tar.gz /usr/local/src/
usuario@server:~$ cd /usr/local/src
usuario@server:~/usr/local/src$ tar -zxvf php-5.3.0.tar.gz
usuario@server:~/usr/local/src/php-5.3.0$
usuario@server:~/usr/local/src/php-5.3.0$ ./configure --prefix=/usr/local/php_httpd --with-apxs2=/usr/local/apache2/bin/apxs --with-config-file-path=/usr/local/php_httpd --enable-sockets --with-mysql=/usr/local/mysql --with-pgsql=/usr/local/pgsql --enable-calendar --with-zlib-dir=/usr/local --with-gd=/usr/local/gd --with-png-dir=/usr/include/libpng12 --with-freetype-dir=/usr --enable-ftp
usuario@server:~/usr/local/src/php-5.3.0$ make
usuario@server:~/usr/local/src/php-5.3.0$ su root
server:~/usr/local/src/php-5.3.0$ make install
server:~/usr/local/src/php-5.3.0$ cp php.ini-dist /usr/local/php_httpd/lib/php.ini
server:~/usr/local/src/php-5.3.0$ vi /usr/local/apache2/conf/httpd.conf
LoadModule php5_module          libexec/libphp5.so
DirectoryIndex index.php index.html
DirectoryIndex index.php index.html
AddType application/x-httpd-php .php .phtml .htm .html
AddType application/x-httpd-php-source .phps
AddType image/x-icon .ico
UserDir public_html htdocs
server:~/usr/local/src/php-5.3.0$ /usr/local/apache2/bin/apachectl restart
server:~/usr/local/src/php-5.3.0$ vi /usr/local/apache2/htdocs/index.php
<? phpinfo(); ?>
server:~/usr/local/src/php-5.3.0$ exit
```

Imagen 4.16. Ejecución del flujo de instalación de PHP sobre el Servidor web HTTPD.

Ejecutar el comando *configure* indicando la ubicación de instalación de los archivos binarios, el comando *apxs2* del servidor *httpd*, la ruta de instalación de *mysql* y *postgres*.

```
usuario@server:~/usr/local/src/php-5.3.0$ ./configure --
prefix=/usr/local/php_httpd --with-apxs2=/usr/local/apache2/bin/apxs --
with-config-file-path=/usr/local/php_httpd --enable-sockets --with-
mysql=/usr/local/mysql --with-pgsql=/usr/local/pgsql --enable-calendar --
with-zlib-dir=/usr/local --with-gd=/usr/local/gd --with-png-
dir=/usr/include/libpng12 --with-freetype-dir=/usr --with-jpeg-dir=/usr --
enable-ftp
```

Ejecutar el comando *make*.

```
usuario@server:~/usr/local/src/php-5.3.0$ make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:~/usr/local/src/php-5.3.0$ su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/php-5.3.0#make install
```

Copiar el archivo *php.ini-dist* a la ruta */usr/local/php_httpd/lib/php.ini*.

```
server:/usr/local/src/php-5.3.0#cp php.ini-dist  
/usr/local/php_httpd/lib/php.ini.
```

Utilizar el editor de textos de preferencia para modificar el archivo *httpd.conf*.

```
server:/usr/local/src/php-5.3.0#vi /usr/local/apache2/conf/httpd.conf
```

Buscar la sección de *LoadModule* y agregar la siguiente línea.

```
LoadModule php5_module libexec/libphp5.so
```

Buscar la sección *AddTypeAddType* y agregar las siguientes líneas.

```
DirectoryIndex index.php index.html
```

```
DirectoryIndex index.php index.html
```

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

```
AddType application/x-httpd-php-source .phps
```

```
AddType image/x-icon .ico
```

```
UserDir public_html htdocs
```

Reiniciar el servidor *apache* para que los cambios en el archivo sufran efecto.

```
server:/usr/local/src/php-5.3.0#/usr/local/apache2/bin/apachectl restart
```

Utilizar el editor de textos para crear un archivo con extensión *php* y así verificar que la instalación sea correcta.

```
server:/usr/local/src/php-5.3.0#vi /usr/local/apache2/htdocs/index.php  
  
<? phpinfo(); ?>
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server:/usr/local/src/php-5.3.0# exit
```

Ingresar a un navegador web e introduce la dirección <http://localhost/> (imagen 4.17) para visualizar la configuración de *php*.

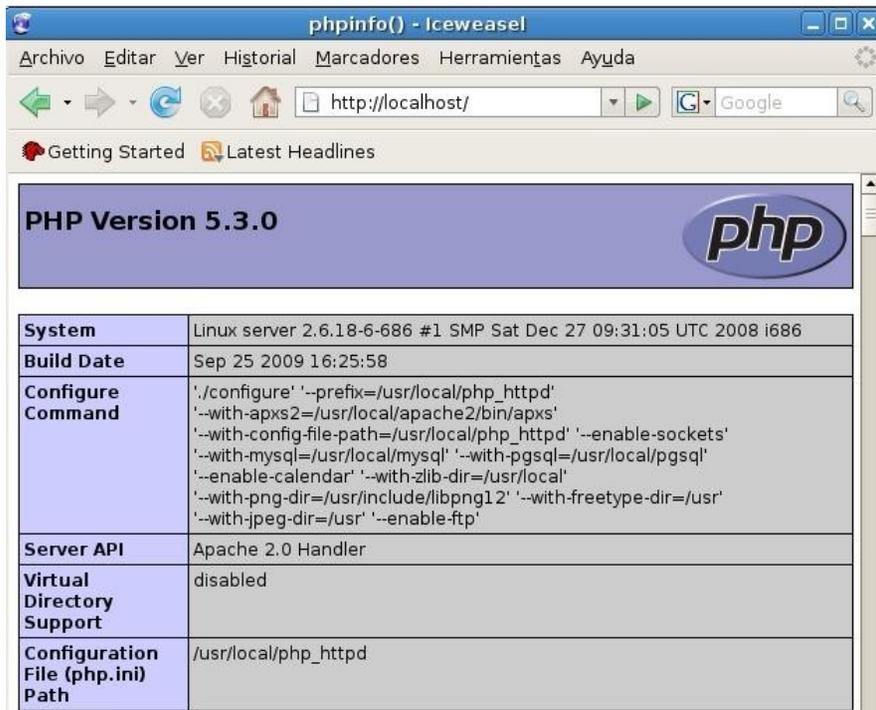


Imagen 4.17. Visualización de pagina de PHP sobre el Sobre el servidor web HTTPD.

PHP como CGI para CHEROKEE

Ejecutar el comando *configure* indicando la ruta de instalación y activando la bandera de redireccionamiento de CGI, .

```
usuario@server:/usr/local/src/php-5.3.0$./configure --  
prefix=/usr/local/php_cgi --enable-force-cgi-redirect --with-config-file-  
path=/usr/local/php_cgi --enable-sockets --with-mysql=/usr/local/mysql --  
with-pgsql=/usr/local/pgsql --enable-calendar --with-zlib-dir=/usr/local --  
with-png-dir=/usr/include/libpng12 --with-freetype-dir=/usr --with-jpeg-  
dir=/usr --enable-ftp
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/php-5.3.0$make
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/php-5.3.0$su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/php-5.3.0#make install
```

Instalar el enlace simbólico de la ubicación de los binarios de *php-cgi*.

```
server:/usr/local/src/php-5.3.0#update-alternatives --install  
"/usr/bin/php-cgi" "php-cgi" "/usr/local/php_cgi/bin/php-cgi" 1
```

Arrancar el servicio de *cherokee-admin* en segundo plano (imagen 4.18), esto iniciará la interfaz administrativa de *cherokee*, que es accesible mediante un navegador ingresando a la ruta <http://localhost:9090>.

```
server:/usr/local/src/php-5.3.0#cherokee-admin -b &
```



Imagen 4.18. Usuario y password para el administrador gráfico de cherokee.

Ingresar el usuario y password que se muestra en la consola al levantar el administrador de cherokee.

```
Usuario: admin          password: XLm17pmW461T0hbb
```

Seleccionar la opción de *Servidores virtuales* (imagen 4.19).



Imagen 4.19. Menú principal del administrador cherokee.

Seleccionar el servidor virtual dando click sobre la liga *default* (imagen 4.20).

Servidores virtuales

 'Servidor virtual' es un mecanismo de abstracción que permite definir un número de parámetros y reglas personalizado que se aplicará a uno o más dominios.

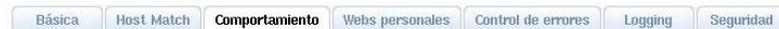
Nombre	Raíz	Dominios	Logging	Stats
default	/usr/local/cherokee/var/www	1	✓	✓

[Asistentes](#) [Clonar servidor virtual](#) [Añadir nuevo servidor virtual](#)

Imagen 4.20. Servidor virtual activado en el servidor cherokee.

Seleccionar la pestaña *comportamiento* (imagen 4.21).

Servidor Virtual: default



 Las reglas se evalúan de **arriba a abajo**. Arrastre y suéltelas para reordenarlas.

Objetivo	Tipo	Gestor	Raíz	Auth	Enc	Cad	Final	
(Extensions: php,html) Y (Bind: 82)	Compleja	FastCGI	✗	✗	✓	✗	✓	🔒
/about	Directory	Info. del servidor	✗	✗	✓	✗	✓	🔒
/cgi-bin	Directory	CGI	✓	✗	✗	✗	✓	🔒
/cherokee_themes	Directory	Contenido estático	✓	✗	✗	✗	✓	🔒
/icons	Directory	Contenido estático	✓	✗	✗	✗	✓	🔒
Defecto	Default	Listar y Enviar	✗	✗	✗	✗		

[Asistentes](#) [Añadir nueva regla](#)

Imagen 4.21. Opciones presentadas al seleccionar la pestaña de comportamiento.

Seleccionar la opción *Asistentes* (imagen 4.22).



Imagen 4.22. Opciones visualizadas al seleccionar la opción asistente

Seleccionar la categoría lenguajes y seleccionar la opción *Run Wizard* que se muestra en la Imagen 4.23.



Imagen 4.23. Wizard de los lenguajes configurados.

Verificar que se agregue la nueva regla con nombre *php* y seleccionar la casilla de la opción *final* (imagen 4.24).

Objetivo	Tipo	Gestor	Raíz	Auth	Enc	Cad	Final	
php	Extensions	FastCGI	✗	✗	✓	✗	<input checked="" type="checkbox"/>	🔒
(Extensions: php,html) Y (Bind: 82)	Compleja	FastCGI	✗	✗	✓	✗	<input checked="" type="checkbox"/>	🔒
/about	Directory	Info. del servidor	✗	✗	✓	✗	<input checked="" type="checkbox"/>	🔒
/cgi-bin	Directory	CGI	✓	✗	✗	✗	<input checked="" type="checkbox"/>	🔒
/cherokee_themes	Directory	Contenido estático	✓	✗	✗	✗	<input checked="" type="checkbox"/>	🔒
/icons	Directory	Contenido estático	✓	✗	✗	✗	<input checked="" type="checkbox"/>	🔒
Defecto	Default	Listar y Enviar	✗	✗	✗	✗		

Imagen 4.24. Al inicio de la lista se muestra la nueva regla para php.

Seleccionar la opción *Guardar* (imagen 4.25).

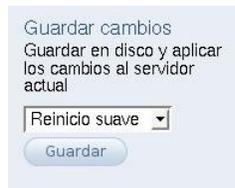


Imagen 4.25. Se deben guardar los cambios al realizar cualquier modificación a la configuración.

Seleccionar la regla que se creó dando click sobre el link [php](#) (imagen 4.26).
default - Extensiones: php



Imagen 4.26. Opciones mostradas al seleccionar la regla de php creada.

Dentro de la pestaña *Regla* seleccionar el botón de la opción Y (imagen 4.27).

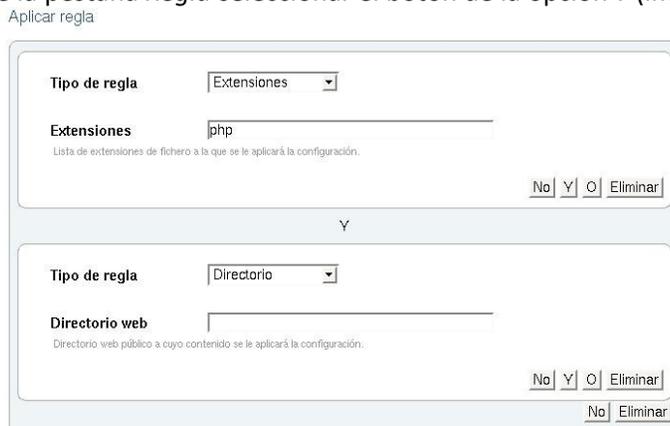
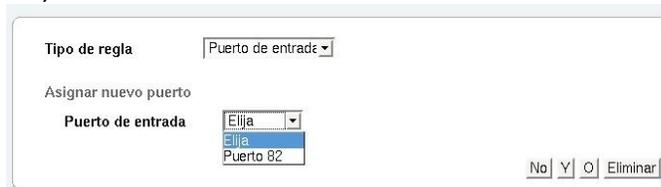


Imagen 4.29. Al seleccionar la opción "Y" se despliegan opciones para agregar criterios a la regla.

Seleccionar la lista de *Tipo de regla* y selecciona la opción *Puerto de entrada* (imagen 4.28).



Tipo de regla: Puerto de entrada

Asignar nuevo puerto

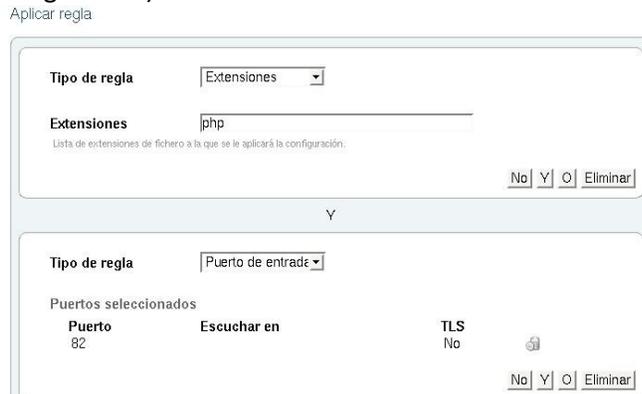
Puerto de entrada: Eljja, Eljja, Puerto 82

No Y O Eliminar

Imagen 4.28. Al seleccionar la opción *Puerto de entrada* se muestra la lista de puertos.

Seleccionar el puerto donde se atenderán las peticiones de *PHP* en el servidor Cherokee(imagen 4.29).

Aplicar regla



Tipo de regla: Extensiones

Extensiones: php

Lista de extensiones de fichero a la que se le aplicará la configuración.

No Y O Eliminar

Y

Tipo de regla: Puerto de entrada

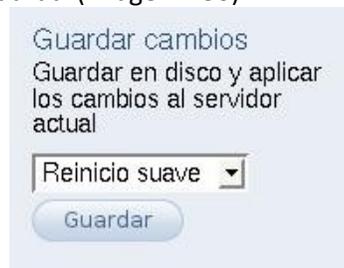
Puertos seleccionados

Puerto	Escuchar en	TLS
82		No

No Y O Eliminar

Imagen 4.29. Configuración de la regla para atención de peticiones de *PHP* en el puerto **82**.

Selecciona la opción *Guardar*(imagen 4.30).



Guardar cambios

Guardar en disco y aplicar los cambios al servidor actual

Reinicio suave

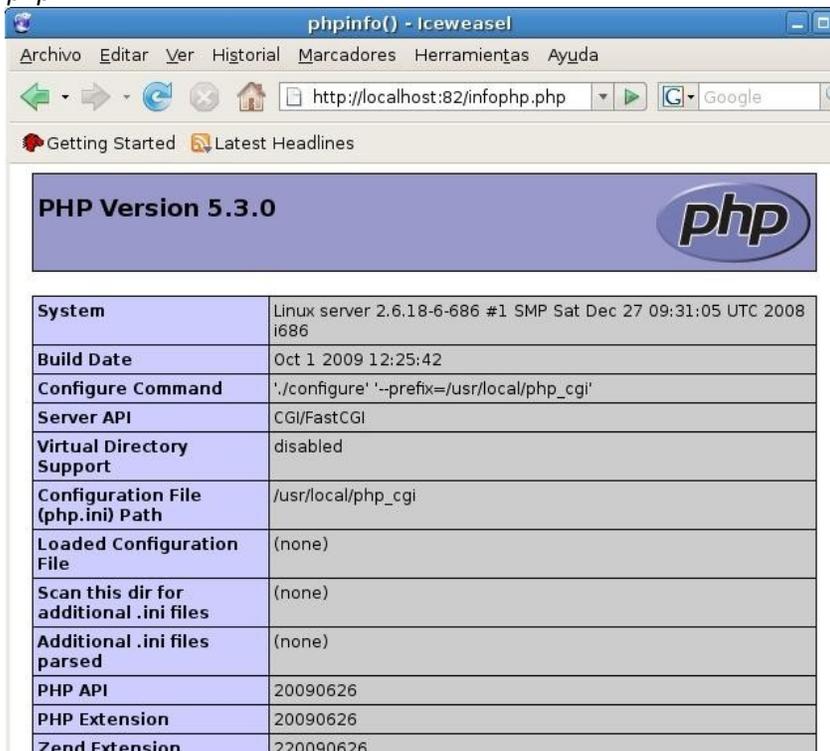
Guardar

Imagen 4.30. Se deben guardar los cambios al realizar cualquier modificación a la configuración.

Utilizar el editor de textos para crear un archivo con extensión *php* y así verificar que la instalación sea correcta.

```
server:/usr/local/src/php-5.3.0#vi  
/usr/local/cherokee/var/www/infophp.php  
  
<? phpinfo(); ?>
```

Ingresar a un navegador web e introducir la dirección <http://localhost:82/infophp.php> (imagen 4.31) para visualizar la configuración de *php*.



PHP Version 5.3.0	
System	Linux server 2.6.18-6-686 #1 SMP Sat Dec 27 09:31:05 UTC 2008 i686
Build Date	Oct 1 2009 12:25:42
Configure Command	'./configure' '--prefix=/usr/local/php_cgi'
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/php_cgi
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626

Imagen 4.31. Visualización de la pagina de información de *php* en el servidor web Cherokee.

Cerrar la sesión de *root* al terminar la ejecución de la configuración.

```
server: /usr/local/src/php-5.3.0#exit
```

Regresar a la carpeta *local*.

```
usuario@server:/usr/local/src/php-5.3.0$cd ../../
```

4.4 RUBY

Es un lenguaje de programación interpretado del lado del servidor, reflexivo y orientado a objetos. Su sintaxis esta basada en Phyton y Perl, con características de programación orientada a objetos de Smalltalk. Maneja todos los tipos de datos como un objeto, incluidas las clases, toda función es un método. Cuenta con soporte para la meta-programación, que son programas que manipulan a otros programas, también cuenta con soporte para hilos de ejecución gestionados por el interprete de Ruby. Ha sido descrito como lenguaje de programación de multiparadigma ya que permite utilizar diferentes paradigmas de programación lo que le permite al desarrollador tomar la mejor herramienta para cada trabajo, admitiendo que ningún paradigma resuelve todos los problemas.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *ruby-1.9.1-p0.tar.gz*

URL de descarga: <http://www.ruby-lang.org/es/downloads/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 4.32 se puede apreciar el flujo de los comandos de instalación.*

Copiar el archivo fuente a la ruta */usr/local/src*.

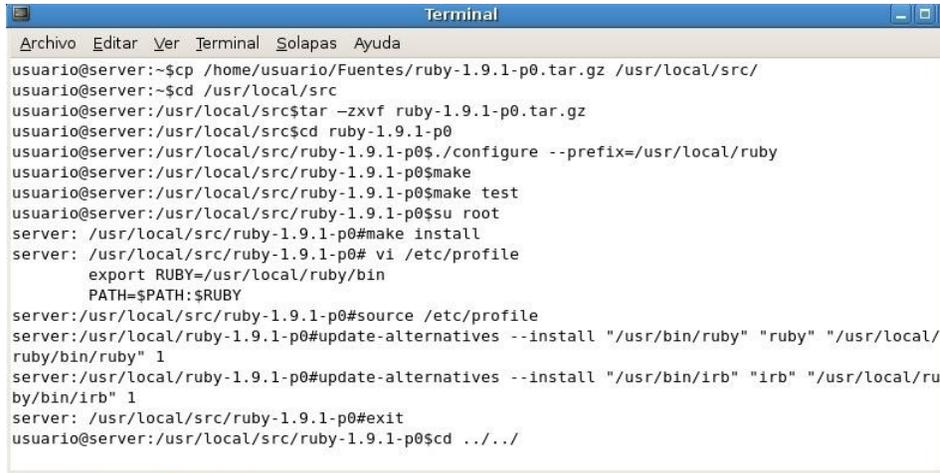
```
usuario@server:~$scp /home/usuario/Fuentes/ruby-1.9.1-p0.tar.gz  
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$tar -zxvf ruby-1.9.1-p0.tar.gz
```



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cp /home/usuario/Fuentes/ruby-1.9.1-p0.tar.gz /usr/local/src/
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$tar -zxvf ruby-1.9.1-p0.tar.gz
usuario@server:/usr/local/src$cd ruby-1.9.1-p0
usuario@server:/usr/local/src/ruby-1.9.1-p0$./configure --prefix=/usr/local/ruby
usuario@server:/usr/local/src/ruby-1.9.1-p0$make
usuario@server:/usr/local/src/ruby-1.9.1-p0$make test
usuario@server:/usr/local/src/ruby-1.9.1-p0$su root
server:/usr/local/src/ruby-1.9.1-p0#make install
server:/usr/local/src/ruby-1.9.1-p0# vi /etc/profile
export RUBY=/usr/local/ruby/bin
PATH=$PATH:$RUBY
server:/usr/local/src/ruby-1.9.1-p0#source /etc/profile
server:/usr/local/ruby-1.9.1-p0#update-alternatives --install "/usr/bin/ruby" "ruby" "/usr/local/
ruby/bin/ruby" 1
server:/usr/local/ruby-1.9.1-p0#update-alternatives --install "/usr/bin/irb" "irb" "/usr/local/ru
by/bin/irb" 1
server:/usr/local/src/ruby-1.9.1-p0#exit
usuario@server:/usr/local/src/ruby-1.9.1-p0$cd ../../
```

Imagen 4.32. Ejecución del flujo de instalación de RUBY.

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$cd ruby-1.9.1-p0
```

Ejecutar el comando *configure* indicando la ubicación donde se instalaran los binarios de *ruby*.

```
usuario@server:/usr/local/src/ruby-1.9.1-p0$./configure --
prefix=/usr/local/ruby
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/ruby-1.9.1-p0$make
```

Ejecutar el comando *make test* para comprobar la configuración.

```
usuario@server:/usr/local/src/ruby-1.9.1-p0$make test
```

Después de configurar la instalación, iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/ruby-1.9.1-p0$su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/ruby-1.9.1-p0#make install
```

Utilizar el editor de textos de preferencia para agregar al archivo *profile* la variable de ambiente que contiene la ruta de los binarios de ruby.

```
server: /usr/local/src/ruby-1.9.1-p0# vi /etc/profile
export RUBY=/usr/local/ruby/bin
PATH=$PATH:$RUBY
```

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando *source*.

```
server:/usr/local/src/ruby-1.9.1-p0#source /etc/profile
```

Instalar el enlace simbólico de la ubicación de los binarios del comando *ruby*.

```
server:/usr/local/ruby-1.9.1-p0#update-alternatives --install
"/usr/bin/ruby" "ruby" "/usr/local/ruby/bin/ruby" 1
```

Instalar el enlace simbólico de la ubicación de los binarios del comando *irb*.

```
server:/usr/local/ruby-1.9.1-p0#update-alternatives --install "/usr/bin/irb"
"irb" "/usr/local/ruby/bin/irb" 1
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server: /usr/local/src/ruby-1.9.1-p0#exit
```

Regresar a la carpeta *local*.

```
usuario@server:/usr/local/src/ruby-1.9.1-p0$cd ../../
```

Ruby GEM

Versión utilizada: *rubygems-1.3.5.tgz*

URL de descarga: http://rubyforge.org/frs/?group_id=126

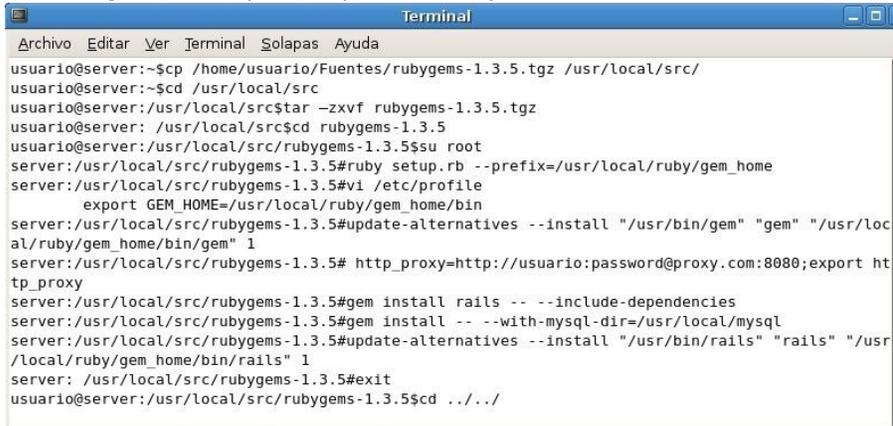
Copiar el archivo fuente a la ruta */usr/local/src*.

```
usuario@server:~$scp /home/usuario/Fuentes/rubygems-1.3.5.tgz
/usr/local/src/
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$cd /usr/local/src
```

En la imagen 4.33 se puede apreciar el flujo de los comandos de instalación.



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$scp /home/usuario/Fuentes/rubygems-1.3.5.tgz /usr/local/src/
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$tar -zxvf rubygems-1.3.5.tgz
usuario@server: /usr/local/src$cd rubygems-1.3.5
usuario@server:/usr/local/src/rubygems-1.3.5$su root
server:/usr/local/src/rubygems-1.3.5#ruby setup.rb --prefix=/usr/local/ruby/gem_home
server:/usr/local/src/rubygems-1.3.5#vi /etc/profile
export GEM_HOME=/usr/local/ruby/gem_home/bin
server:/usr/local/src/rubygems-1.3.5#update-alternatives --install "/usr/bin/gem" "gem" "/usr/local/ruby/gem_home/bin/gem" 1
server:/usr/local/src/rubygems-1.3.5# http_proxy=http://usuario:password@proxy.com:8080;export http_proxy
server:/usr/local/src/rubygems-1.3.5#gem install rails -- --include-dependencies
server:/usr/local/src/rubygems-1.3.5#gem install -- --with-mysql-dir=/usr/local/mysql
server:/usr/local/src/rubygems-1.3.5#update-alternatives --install "/usr/bin/rails" "rails" "/usr/local/ruby/gem_home/bin/rails" 1
server: /usr/local/src/rubygems-1.3.5#exit
usuario@server:/usr/local/src/rubygems-1.3.5$cd ../../
```

Imagen 4.33. Ejecución del flujo de instalación de RUBY Gem para la instalación de Rails.

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$tar -zxvf rubygems-1.3.5.tgz
```

Ingresar a la carpeta generada.

```
usuario@server:/usr/local/src$cd rubygems-1.3.5
```

Iniciar sesión en la consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src/rubygems-1.3.5$su root
```

Ejecutar el script *setup.rb* utilizando el comando *ruby* e indicando la ruta de instalación.

```
server:/usr/local/src/rubygems-1.3.5#ruby setup.rb --
prefix=/usr/local/ruby/gem_home
```

Utilizar el editor de textos para agregar al archivo *profile* la variable de ambiente que contiene la ruta de los binarios de la gema.

```
server:/usr/local/src/rubygems-1.3.5#vi /etc/profile
export GEM_HOME=/usr/local/ruby/gem_home/bin
```

Instalar el enlace simbólico de la ubicación de los binarios del comando *gem*.

```
server:/usr/local/src/rubygems-1.3.5#update-alternatives --install
"/usr/bin/gem" "gem" "/usr/local/ruby/gem_home/bin/gem" 1
```

En caso de realizar la conexión a Internet a través de un *proxy*, declara la variable de ambiente *http_proxy* asignándole los parámetros de conexión al *proxy*.

```
server:/usr/local/src/rubygems-1.3.5#  
http_proxy=http://usuario:password@proxy.com:8080;export http_proxy
```

Ejecutar el comando *gem install* indicándole que se requiere instalar la gema de *rails* incluyendo dependencias.

```
server:/usr/local/src/rubygems-1.3.5#gem install rails -- --include-  
dependencies
```

Ejecutar el comando *gem install* indicándole que se requiere instalar la gema de *mysql* e indicando la ruta donde se encuentran los binarios de *mysql* en nuestro equipo.

```
server:/usr/local/src/rubygems-1.3.5#gem install -- --with-mysql-  
dir=/usr/local/mysql
```

Instalar el enlace simbólico de la ubicación de los binarios del comando *rails*.

```
server:/usr/local/src/rubygems-1.3.5#update-alternatives --install  
"/usr/bin/rails" "rails" "/usr/local/ruby/gem_home/bin/rails" 1
```

Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server: /usr/local/src/rubygems-1.3.5#exit
```

Regresar a la carpeta *local*.

```
usuario@server:/usr/local/src/rubygems-1.3.5$cd ../../
```

En la imagen 4.34 se puede apreciar la página de prueba de *Ruby on Rails*, ejecutándose sobre su propio servidor web.

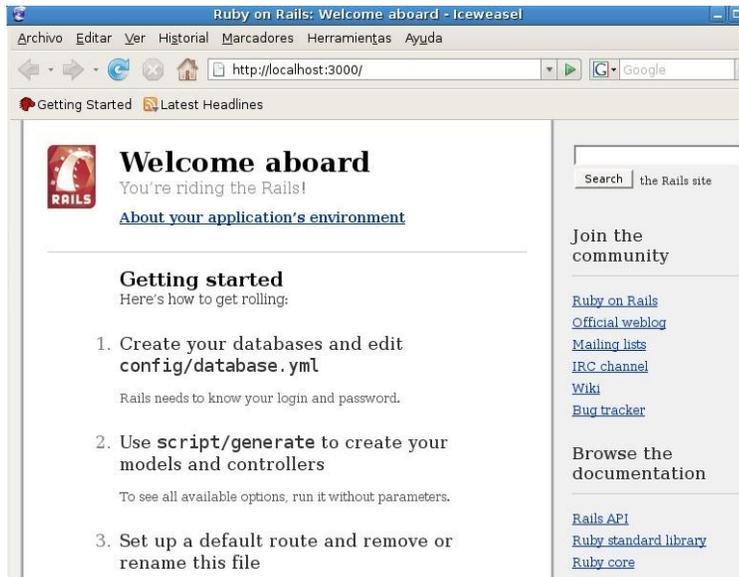


Imagen 4.34. Visualización de pagina de Ruby On Rails en el puerto 3000.

Con la instalación de estos lenguajes de programación se cuenta con la base necesaria para comenzar con la tarea de desarrollo de aplicaciones, se eligieron estos lenguajes por su gran demanda en el desarrollo de sistemas tanto para aplicaciones web como para sistemas de escritorio. El lenguaje de *PHP* se instaló para los distintos servidores que se tienen instalados, así como también se instaló el framework de Tomcat para el manejo de servlets y applets con conexión para atender peticiones a través del servidor web HTTPD. Por la parte de *Ruby* a su vez podría configurarse para atender las peticiones a través de los servidores web instalados, pero se recomienda usarlo mejor bajo su propio servidor que viene integrado.

Capítulo 5 Aplicaciones utilitarias para desarrollo

Para este momento ya se tienen todos los requerimientos básicos para comenzar con el desarrollo de aplicaciones tanto de tipo de escritorio como aplicaciones web, se puede programar en cualquier lenguaje de programación sólo utilizando un editor de textos de modo gráfico o vía consola, como por ejemplo “vi”, además, la configuración del equipo se puede realizar igualmente sólo utilizando la consola de comandos, o para facilitar esas tareas existen diferentes aplicaciones que realizan las funciones a través de una interfaz gráfica, la cual facilita las acciones requeridas para un usuario que tenga experiencia media-básica en el uso de Debian. En este Capítulo se proponen algunas aplicaciones primordiales que serán de gran ayuda para los desarrolladores que se encuentran iniciando en el mundo del desarrollo de software.

5.1 Eclipse como IDE de desarrollo

Es un entorno integrado de desarrollo creado inicialmente por la compañía IBM, su principal característica es que esta pensado y construido a base de módulos o *plug-in's*, lo cual permite agregar funcionalidades y herramientas de desarrollo o eliminar de las vistas de desarrollo las funciones que no sean requeridas. El desarrollo y soporte de la aplicación es realizado por la comunidad de Eclipse que es una organización sin fines de lucro que fomenta el desarrollo de software libre para ser integrado en Eclipse. La facilidad de creación y libre uso de módulos para Eclipse ha permitido que se desarrollen extensiones para el manejo de otros lenguajes de programación tal como C++, Phython, Ruby, Perl e incluso la interacción con manejadores de bases de datos.

Tipo de licencia: OpenSource.

Proceso de instalación

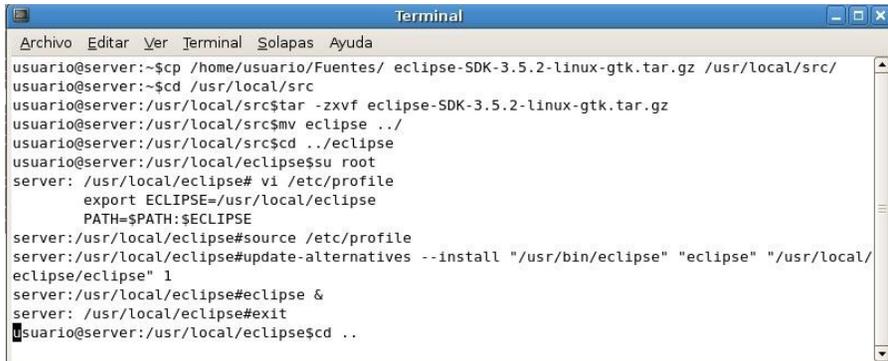
Versión utilizada: *eclipse-SDK-3.5.2-linux-gtk.tar.gz*

URL de descarga: <http://www.eclipse.org/downloads/>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 5.1 se puede apreciar el flujo de los comandos de instalación.*

Copiar el archivo fuente a la ruta `/usr/local/src`.

```
usuario@server:~$scp /home/usuario/Fuentes/ eclipse-SDK-3.5.2-linux-gtk.tar.gz /usr/local/src/
```



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$scp /home/usuario/Fuentes/ eclipse-SDK-3.5.2-linux-gtk.tar.gz /usr/local/src/
usuario@server:~$cd /usr/local/src/
usuario@server:/usr/local/src$tar -zxvf eclipse-SDK-3.5.2-linux-gtk.tar.gz
usuario@server:/usr/local/src$mv eclipse ../
usuario@server:/usr/local/src$cd ../eclipse
usuario@server:/usr/local/eclipse$su root
server: /usr/local/eclipse# vi /etc/profile
      export ECLIPSE=/usr/local/eclipse
      PATH=$PATH:$ECLIPSE
server:/usr/local/eclipse#source /etc/profile
server:/usr/local/eclipse#update-alternatives --install "/usr/bin/eclipse" "eclipse" "/usr/local/eclipse/eclipse" 1
server:/usr/local/eclipse#eclipse &
server: /usr/local/eclipse#exit
usuario@server:/usr/local/eclipse$cd ..
```

Imagen 5.1. Ejecución del flujo de instalación del IDE de desarrollo Eclipse.

Ingresa a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo fuente.

```
usuario@server:/usr/local/src$tar -zxvf eclipse-SDK-3.5.2-linux-gtk.tar.gz
```

Mover el directorio generado a la ruta `/usr/local/`.

```
usuario@server:/usr/local/src$mv eclipse ../
```

Ingresa al directorio de `eclipse`.

```
usuario@server:/usr/local/src$cd ../eclipse
```

Iniciar sesión en la consola de comandos como usuario `root`.

```
usuario@server:/usr/local/eclipse$su root
```

Utilizar el editor de textos de preferencia para agregar al archivo `profile` la variable de ambiente que contiene la ruta de los binarios de `Eclipse`.

```
server: /usr/local/eclipse# vi /etc/profile
```

```
export ECLIPSE=/usr/local/eclipse
```

```
PATH=$PATH:$ECLIPSE
```


Cerrar la sesión de *root* al terminar la ejecución del comando anterior.

```
server: /usr/local/eclipse#exit
```

Regresar a la carpeta *local*.

```
usuario@server:/usr/local/eclipse$cd ..
```

5.2 SUBVERSION

Es una aplicación cuyo funcionamiento se centra en el manejo de versiones de los archivos utilizados dentro de un grupo de trabajo. Surgió para reemplazar a la aplicación CVS el cual tiene bastantes deficiencias. Esta aplicación puede ser utilizada desde modo línea de comando e incluso existen extensiones para su implementación con IDE's de desarrollo como Eclipse, Jdeveloper, NetBeans o Visual Estudio. También cuenta con la facilidad de configuración de la administración del acceso de los usuarios mediante simples instrucciones o reglas de acceso en su archivo de configuración principal. Esta aplicación es muy utilizada en la comunidad de software libre y se utiliza en muchos proyectos como en Apache, KDE, GNOME, FreeBSD por mencionar algunas.

Tipo de licencia: OpenSource.

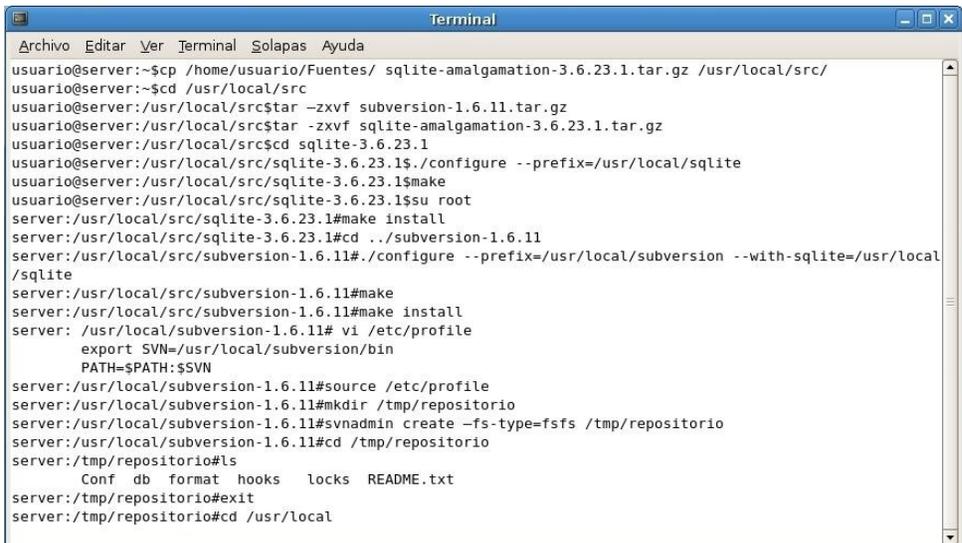
Proceso de instalación

Versiones utilizadas: *subversion-1.6.11.tar.gz*, *sqlite-amalgamation-3.6.23.1.tar.gz*

URL de descarga:

<http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=260&expandFolder=74> , <http://www.sqlite.org/download.html>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 5.4 se puede apreciar el flujo de los comandos de instalación.*



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cp /home/usuario/Fuentes/ sqlite-amalgamation-3.6.23.1.tar.gz /usr/local/src/
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$tar -zxvf subversion-1.6.11.tar.gz
usuario@server:/usr/local/src$tar -zxvf sqlite-amalgamation-3.6.23.1.tar.gz
usuario@server:/usr/local/src$cd sqlite-3.6.23.1
usuario@server:/usr/local/src/sqlite-3.6.23.1$./configure --prefix=/usr/local/sqlite
usuario@server:/usr/local/src/sqlite-3.6.23.1$make
usuario@server:/usr/local/src/sqlite-3.6.23.1$su root
server:/usr/local/src/sqlite-3.6.23.1#make install
server:/usr/local/src/sqlite-3.6.23.1#cd ../subversion-1.6.11
server:/usr/local/src/subversion-1.6.11#./configure --prefix=/usr/local/subversion --with-sqlite=/usr/local/
sqlite
server:/usr/local/src/subversion-1.6.11#make
server:/usr/local/src/subversion-1.6.11#make install
server:/usr/local/subversion-1.6.11# vi /etc/profile
export SVN=/usr/local/subversion/bin
PATH=$PATH:$SVN
server:/usr/local/subversion-1.6.11#source /etc/profile
server:/usr/local/subversion-1.6.11#mkdir /tmp/repositorio
server:/usr/local/subversion-1.6.11#svnadmin create -fs-type=fsfs /tmp/repositorio
server:/usr/local/subversion-1.6.11#cd /tmp/repositorio
server:/tmp/repositorio#ls
Conf db format hooks locks README.txt
server:/tmp/repositorio#exit
server:/tmp/repositorio#cd /usr/local
```

Imagen 5.4. Ejecución del flujo de instalación de Subversion.

Copiar los archivos fuente a la ruta `/usr/local/src`.

```
usuario@server:~$cp /home/usuario/Fuentes/subversion-1.6.11.tar.gz /usr/local/src/
```

```
usuario@server:~$cp /home/usuario/Fuentes/sqlite-amalgamation-3.6.23.1.tar.gz /usr/local/src/
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo fuente de `subversion`.

```
usuario@server:/usr/local/src$tar --zxvf subversion-1.6.11.tar.gz
```

Desempaquetar el archivo fuente de `sqlite`.

```
usuario@server:/usr/local/src$tar --zxvf sqlite-amalgamation-3.6.23.1.tar.gz
```

Ingresar a la carpeta generada de `sqlite`.

```
usuario@server:/usr/local/src$cd sqlite-3.6.23.1
```

Ejecutar el comando *configure* indicando la ruta de instalación de *sqlite*.

```
usuario@server:/usr/local/src/sqlite-3.6.23.1$./configure --  
prefix=/usr/local/sqlite
```

Ejecutar el comando *make*.

```
usuario@server:/usr/local/src/sqlite-3.6.23.1$make
```

Iniciar sesión como usuario *root* en al consola de comandos.

```
usuario@server:/usr/local/src/sqlite-3.6.23.1$su root
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/sqlite-3.6.23.1#make install
```

Ingresar a la carpeta de *subversion* generada al desempaquetar el código fuente.

```
server:/usr/local/src/sqlite-3.6.23.1#cd ../subversion-1.6.11
```

Ejecutar el comando *configure* indicando la ubicación de la ruta de instalación de *subversio* y la ubicación de la instalación de *sqlite*.

```
server:/usr/local/src/subversion-1.6.11#./configure --  
prefix=/usr/local/subversion --with-sqlite=/usr/local/sqlite
```

Ejecutar el comando *make*.

```
server:/usr/local/src/subversion-1.6.11#make
```

Ejecutar el comando *make install* para aplicar la instalación.

```
server:/usr/local/src/subversion-1.6.11#make install
```

Utilizar el editor de textos de preferencia para agregar al archivo *profile* la variable de ambiente que contiene la ruta de los binarios de *Subversion*.

```
server: /usr/local/subversion-1.6.11# vi /etc/profile
```

```
export SVN=/usr/local/subversion/bin
```

```
PATH=$PATH:$SVN
```

Actualizar las variables de ambiente en la consola de comandos ejecutando el comando *source*.

```
server:/usr/local/subversion-1.6.11#source /etc/profile
```

Crear un directorio bajo la ruta `/tmp` con nombre de *repositorio*.

```
server:/usr/local/subversion-1.6.11#mkdir /tmp/repositorio
```

Crear un repositorio de prueba en el directorio previamente creado.

```
server:/usr/local/subversion-1.6.11#svnadmin create -fs-type=fsfs  
/tmp/repositorio
```

Ingresa al directorio `/tmp/repositorio`.

```
server:/usr/local/subversion-1.6.11#cd /tmp/repositorio
```

Enlistar los archivos del directorio.

```
server:/tmp/repositorio#ls
```

Verificar que se muestren los siguientes archivos y directorios creados.

```
Conf db format hooks locks README.txt
```

Cerrar la sesión de *root*.

```
server:/tmp/repositorio#exit
```

Regresar a la carpeta `/usr/local`.

```
server:/tmp/repositorio#cd /usr/local
```

5.3 DBDesigner 4

Es una aplicación de diseño de bases de datos y sus funcionalidades son: diseño de bases de datos, modelado, creación y mantenimiento de scripts SQL bajo un entorno gráfico que ofrece una interfaz muy sencilla y potente a la vez, lo que facilita el manejo de la aplicación por parte del usuario. Su optimización y desarrollo está orientado para trabajar con el manejador de bases de datos MySQL, aunque también es capaz de generar código de transacciones SQL para Postgres, Oracle entre otros. Su funcionalidad se puede comparar con la de productos comerciales como Oracle Designer o IBM's Rational Rose pero DBDesigner se encuentra bajo licencia GPL.

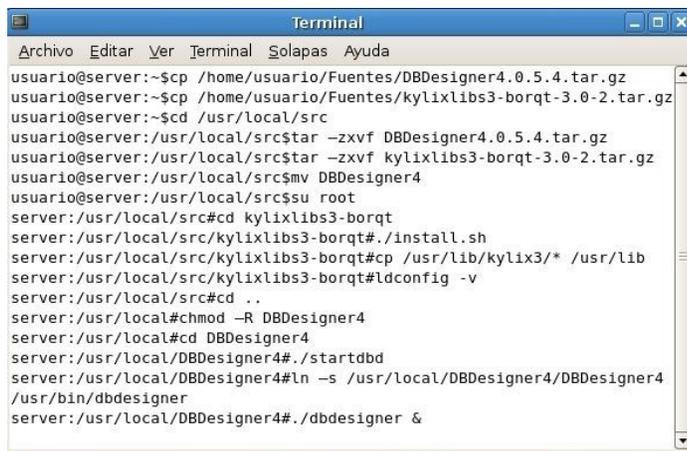
Tipo de licencia: OpenSource.

Proceso de instalación

Versiones utilizadas: DBDesigner4.0.5.4.tar.gz, kyllixlibs3-borqt-3.0-2.tar.gz

URL de descarga: <http://www.fabforce.net/downloads.php>,
<http://kylixlibs.sourceforge.net/down.html>

Indicaciones previas: La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 5.5 se puede apreciar el flujo de los comandos de instalación.



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cp /home/usuario/Fuentes/DBDesigner4.0.5.4.tar.gz
usuario@server:~$cp /home/usuario/Fuentes/kylixlibs3-borqt-3.0-2.tar.gz
usuario@server:~$cd /usr/local/src
usuario@server:~/local/src$tar -zxvf DBDesigner4.0.5.4.tar.gz
usuario@server:~/local/src$tar -zxvf kyllixlibs3-borqt-3.0-2.tar.gz
usuario@server:~/local/src$mv DBDesigner4
usuario@server:~/local/src$su root
server:~/local/src#cd kyllixlibs3-borqt
server:~/local/src/kylixlibs3-borqt#./install.sh
server:~/local/src/kylixlibs3-borqt#cp /usr/lib/kylix3/* /usr/lib
server:~/local/src/kylixlibs3-borqt#ldconfig -v
server:~/local/src#cd ..
server:~/local#chmod -R DBDesigner4
server:~/local#cd DBDesigner4
server:~/local/DBDesigner4#./startdbd
server:~/local/DBDesigner4#ln -s /usr/local/DBDesigner4/DBDesigner4
/usr/bin/dbdesigner
server:~/local/DBDesigner4#./dbdesigner &
```

Imagen 5.5 Ejecución del flujo de instalación de la aplicación DBDesigner4.

Copiar los archivos de la aplicación y la librería a la ruta `/usr/local/src`.

```
usuario@server:~$cp /home/usuario/Fuentes/DBDesigner4.0.5.4.tar.gz
```

```
usuario@server:~$cp /home/usuario/Fuentes/kylixlibs3-borqt-3.0-2.tar.gz
```

Ingresar a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo de los binarios de DBDesigner.

```
usuario@server:~/local/src$tar -zxvf DBDesigner4.0.5.4.tar.gz
```

Desempaquetar el archivo de la librería.

```
usuario@server:~/local/src$tar -zxvf kyllixlibs3-borqt-3.0-2.tar.gz
```

Mover el directorio generado de *DBDesigner* a la ruta */usr/local*.

```
usuario@server:/usr/local/src$mv DBDesigner4
```

Iniciar session en consola de comandos como usuario *root*.

```
usuario@server:/usr/local/src$su root
```

Ingresar a la carpeta *kylixlibs3-borqt*.

```
server:/usr/local/src#cd kylixlibs3-borqt
```

Ejecutar el script de instalación *install.sh*

```
server:/usr/local/src/kylixlibs3-borqt#./install.sh
```

Copiar todo el contenido del directorio */usr/lib/kylix3* al directorio */usr/lib*.

```
server:/usr/local/src/kylixlibs3-borqt#cp /usr/lib/kylix3/* /usr/lib
```

Ejecutar el comando *ldconfig -v* para crear los links y el cache de la nueva librería instalada.

```
server:/usr/local/src/kylixlibs3-borqt#ldconfig -v
```

Descender al directorio */usr/local*.

```
server:/usr/local/src#cd ..
```

Asignar los permisos 755 al directorio *DBDesigner4* y a todos los archivos contenidos en ella.

```
server:/usr/local#chmod -R DBDesigner4
```

Ingresar al directorio *DBDesigner4*.

```
server:/usr/local#cd DBDesigner4
```

Ejecutar el script de instalación *startdbd*

```
server:/usr/local/DBDesigner4#./startdbd
```

Crear un link del script *DBDesigner4* en la ruta de los binarios de usuario.

```
server:/usr/local/DBDesigner4#ln -s /usr/local/DBDesigner4/DBDesigner4  
/usr/bin/dbdesigner
```

Ejecutar el comando *DBDesigner* en segundo plano para lanzar la aplicación. En la imagen 5.6 se puede apreciar la pantalla de inicio de la aplicación.

```
server:/usr/local/DBDesigner4#./dbdesigner &
```

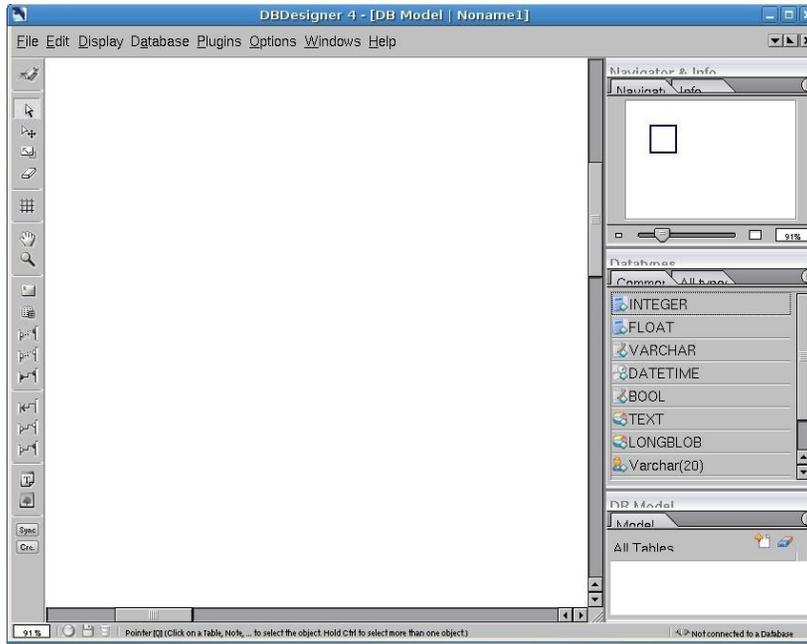


Imagen 5.6 Pantalla principal de la aplicación *DBDesigner*.

Cerrar la sesión de *root*.

```
server:/usr/local/DBDesigner4#exit
```

Regresar a la carpeta */usr/local*.

```
usuario@server:/usr/local/DBDesigner4$cd ..
```

5.4 Webmin

Es una aplicación a la cual se accede a través de una interfaz web que sirve para la administración de sistemas de tipo UNIX, en este caso GNU/Linux en su distribución Debia. Para acceder a esta aplicación se puede utilizar cualquier navegador para configurar las cuentas de usuario, Apache, DNS, apagado del equipo, compartir archivos, etc. Esto facilita la tarea de administrar el equipo local

o remotamente. Esta escrito en perl y Ejecutar su propio servidor web pero eso no significa que no se pueda realizar la conexión y acceder a la aplicación a través de Apache o Cherokee. Por defecto se comunica a través del puerto TCP 10.000, y puede ser configurado para usar SSL.

Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: *webmin-1.510.tar.gz*

URL de descarga: <http://www.webmin.com/download.html>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 5.7 se puede apreciar el flujo de los comandos de instalación.*

Copiar los archivos de la aplicación a la ruta */usr/local/src*.

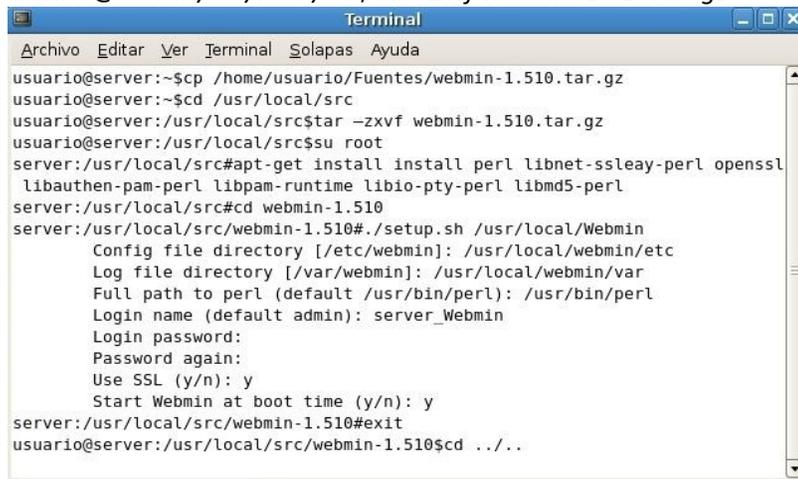
```
usuario@server:~$cp /home/usuario/Fuentes/webmin-1.510.tar.gz
```

Ingresar a la ruta */usr/local/src*.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo de los binarios de DBDesigner.

```
usuario@server:/usr/local/src$tar -zxvf webmin-1.510.tar.gz
```



```
Terminal
Archivo  Editar  Ver     Terminal  Solapas  Ayuda
usuario@server:~$cp /home/usuario/Fuentes/webmin-1.510.tar.gz
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$tar -zxvf webmin-1.510.tar.gz
usuario@server:/usr/local/src$su root
server:/usr/local/src#apt-get install install perl libnet-ssleay-perl openssl
libauthen-pam-perl libpam-runtime libio-pty-perl libmd5-perl
server:/usr/local/src#cd webmin-1.510
server:/usr/local/src/webmin-1.510#./setup.sh /usr/local/Webmin
Config file directory [/etc/webmin]: /usr/local/webmin/etc
Log file directory [/var/webmin]: /usr/local/webmin/var
Full path to perl (default /usr/bin/perl): /usr/bin/perl
Login name (default admin): server_Webmin
Login password:
Password again:
Use SSL (y/n): y
Start Webmin at boot time (y/n): y
server:/usr/local/src/webmin-1.510#exit
usuario@server:/usr/local/src/webmin-1.510$cd ../../
```

Imagen 5.7 Ejecución del flujo de instalación de la aplicación Webmin.

Iniciar sesión como *root* en la consola de comandos.

usuario@server:/usr/local/src\$su root

Instalar las dependencias de perl para poder ejecutar la aplicación.

*server:/usr/local/src#apt-get install install perl libnet-ssleay-perl openssl
libauthen-pam-perl libpam-runtime libio-pty-perl libmd5-perl*

Ingresar al directorio *webmin-1.510*.

server:/usr/local/src#cd webmin-1.510

Ejecutar el script *setup.sh* indicando la ruta de instalación de la aplicación.

server:/usr/local/src/webmin-1.510#./setup.sh /usr/local/Webmin

Ingresar la ruta de instalación de los archivos de configuración.

Config file directory [/etc/webmin]: /usr/local/webmin/etc

Ingresar la ruta donde se ubicaran los archivos Log.

Log file directory [/var/webmin]: /usr/local/webmin/var

Ingresar la ruta del binario del comando *perl*.

Full path to perl (default /usr/bin/perl): /usr/bin/perl

Ingresar el *Login* del usuario.

Login name (default admin): server_Webmin

Ingresar y confirma el *password*.

Login password:

Password again:

Seleccionar la opción “y” para usar el protocolo *https*

Use SSL (y/n): y

Seleccionar la opción “y” para que se inicie la aplicación al encender el servidor.

Start Webmin at boot time (y/n): y

Utilizar un navegador web para ingresar a la ruta <https://localhost:10000> (imagen 5.8), iniciar sesión como administrador.



Imagen 5.8 Pantalla de inicio de sesión de Webmin.

Verificar que se muestre la información del equipo (imagen 5.9).

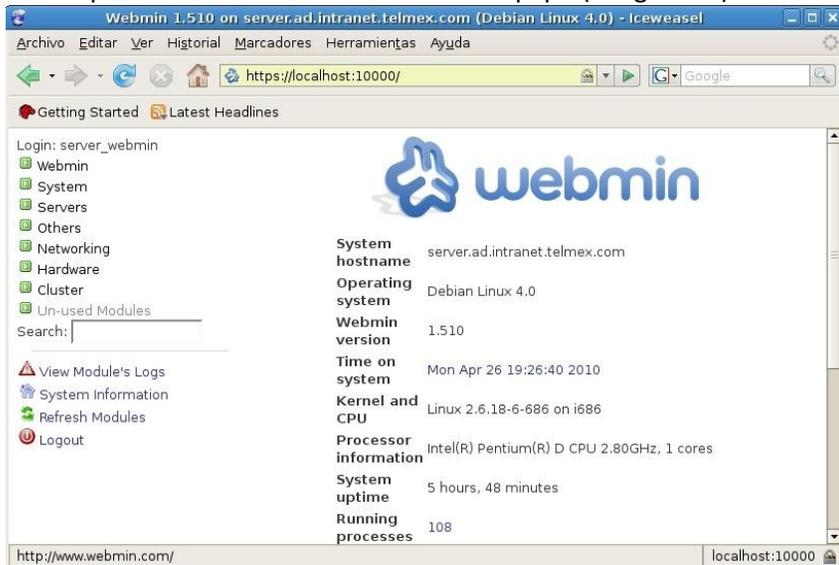


Imagen 5.9 Pantalla principal de la aplicación Webmin.

Cerrar la sesión de root.

```
server:/usr/local/src/webmin-1.510#exit
```

Regresar a la carpeta `/usr/local`.

```
usuario@server:/usr/local/src/webmin-1.510$cd ../..
```

5.5 Terminator

Es una aplicación muy útil para los desarrolladores que gustan por el uso de la terminal de comandos para realizar sus tareas diarias, ya que permite abrir múltiples consolas de comandos en una sola ventana, dividiéndola horizontal o verticalmente. Su principal objetivo es ser una herramienta útil para la organización de terminales. Su desarrollo se baso en aplicaciones como gnome-multi-term, quadkonsole entre otras.

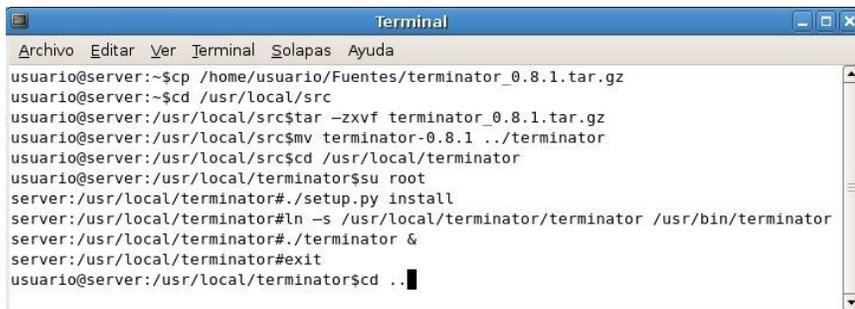
Tipo de licencia: OpenSource.

Proceso de instalación

Versión utilizada: `terminator_0.8.1.tar.gz`

URL de descarga: <https://launchpad.net/terminator/trunk/0.8.1>

Indicaciones previas: *La ejecución de los comandos se realizará dentro de una consola de comandos firmada con un usuario sin privilegios de root, en caso de indicarse lo contrario se ejecutará como usuario root. En la imagen 5.10 se puede apreciar el flujo de los comandos de instalación.*



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
usuario@server:~$cp /home/usuario/Fuentes/terminator_0.8.1.tar.gz
usuario@server:~$cd /usr/local/src
usuario@server:/usr/local/src$tar -zxvf terminator_0.8.1.tar.gz
usuario@server:/usr/local/src$mv terminator-0.8.1 ../terminator
usuario@server:/usr/local/src$cd /usr/local/terminator
usuario@server:/usr/local/terminator$su root
server:/usr/local/terminator#. /setup.py install
server:/usr/local/terminator#ln -s /usr/local/terminator/terminator /usr/bin/terminator
server:/usr/local/terminator#. /terminator &
server:/usr/local/terminator#exit
usuario@server:/usr/local/terminator$cd ..
```

Imagen 5.10 Ejecución del flujo de instalación del emulador de terminales.

Copiar los archivos de la aplicación a la ruta `/usr/local/src`.

```
usuario@server:~$cp /home/usuario/Fuentes/terminator_0.8.1.tar.gz
```

Ingresa a la ruta `/usr/local/src`.

```
usuario@server:~$cd /usr/local/src
```

Desempaquetar el archivo de los binarios de DBDesigner.

```
usuario@server:/usr/local/src$tar -zxvf terminator_0.8.1.tar.gz
```

Mover el directorio generado a la ruta `/usr/local/` y renómbralo como `terminator`.

```
usuario@server:/usr/local/src$mv terminator-0.8.1 ../terminator
```

Ingresa al directorio generado.

```
usuario@server:/usr/local/src$cd /usr/local/terminator
```

Iniciar sesión como usuario `root` en la consola de comandos.

```
usuario@server:/usr/local/terminator$su root
```

Ejecutar el script `setup.py install` para instalar la aplicación.

```
server:/usr/local/terminator#./setup.py install
```

Crear un link del script `terminator` en la ruta de los binarios de usuario.

```
server:/usr/local/terminator#ln -s /usr/local/terminator/terminator  
/usr/bin/terminator
```

Ejecutar el comando `terminator` en segundo plano para lanzar la aplicación. En la imagen 5.11 se puede apreciar la ejecución de la aplicación `Terminator`.

```
server:/usr/local/terminator#./terminator &
```

Cerrar la sesión de `root`.

```
server:/usr/local/terminator#exit
```

Regresar a la carpeta `/usr/local`.

```
usuario@server:/usr/local/terminator$cd ..
```

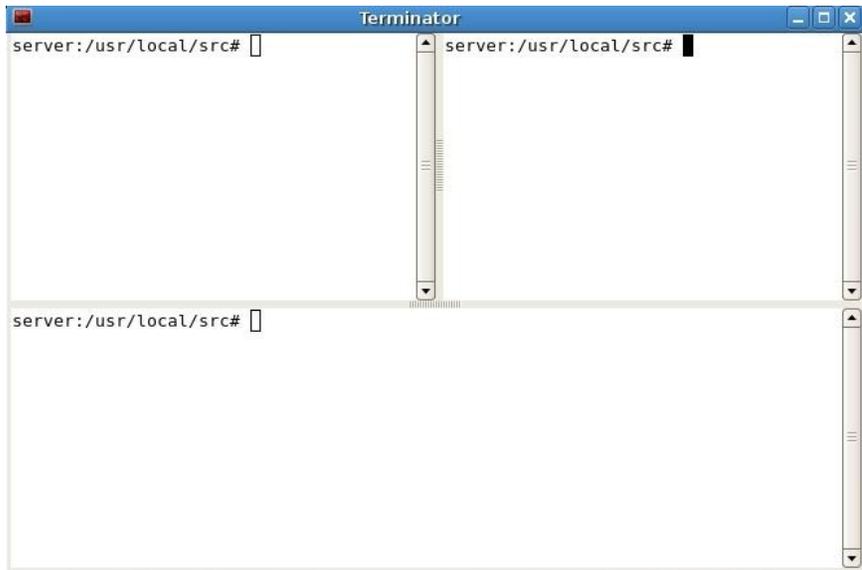


Imagen 5.11 Pantalla de la aplicación Terminator dividida en tres secciones.

Existe una gran cantidad de aplicaciones bajo licencia de software libre orientadas a la administración de equipos en modo gráfico y para el desarrollo de programas, las que se propusieron como instalación son algunas de las más populares y a su vez de gran utilidad para comenzar a trabajar. Eclipse tiene la principal característica de agregar diferentes extensiones para ampliar su funcionalidad y proporcionar el entorno de desarrollo para diferentes lenguajes de programación, podría convertirse en la única aplicación que se necesitara utilizar durante la tarea de editar código e incluso como interfaz gráfica para realizar tareas de administración de los manejadores de bases de datos. Subversion por su parte facilita la tarea de llevar un mejor control de las versiones de los archivos que conforman algún proyecto de desarrollo y más cuando en el proyecto intervienen más de 2 desarrolladores simultáneamente. DBDesigner4 ayuda primordialmente al momento de encontrarse en la etapa de diseño de la base de datos por el hecho de mostrar gráficamente como se interrelacionan las tablas. Webmin es útil para la administración del equipo ya sea vía local o remotamente, esto para los usuarios que no son tan expertos y opten por métodos de administración gráficos. El emulador de terminales Terminator será muy utilizado y requerido por los usuarios más avanzados y que opten por realizar todas las tareas de configuración y administración a través de la consola de comandos.

Conclusiones

La forma en como se realizó la instalación de cada una de las aplicaciones descritas en este manual se efectuó conforme a la experiencia personal y de compañeros de trabajo. Todas las aplicaciones cuentan con su manual de instalación en sus páginas oficiales, pero esas instrucciones en la mayoría de los casos sólo hacen mención a las dependencias de programas y bibliotecas que se requieren previamente instalar, pero no indican la forma de llevarlo a cabo e incluso al momento de estar realizando la compilación e instalación de los archivos fuente o la instalación directa de los binarios, surgen problemas que no están documentados en los manuales ni documentos de las aplicaciones en sus sitios oficiales.

Los foros que existen en los sitios oficiales de algunas aplicaciones también son de gran utilidad para resolver dudas, aunque existe el problema de que en la gran mayoría de las veces las soluciones descritas no sirven en el flujo de instalación de los programas en los equipos, lo que lleva a probar todas las soluciones encontradas hasta encontrar la combinación que lleve a un flujo de instalación efectivo para nuestro equipo. El tiempo de instalación de una nueva aplicación ira disminuyendo con base a la experiencia ganada, porque con el pasar del tiempo y la información adquirida, ayuda a realizar deducciones e incluso probar nuevos flujos de instalación de una aplicación. También se puede encontrar en la web bastantes blog's de personas que realizan la instalación y configuración de las aplicaciones aquí descritas, pero de igual forma no siempre funciona como ellos lo describen u olvidan configuraciones extras que ellos realizan previamente en su equipo. Además la forma en que se realiza la instalación muchas veces lo hacen con comandos que instalan con parámetros por omisión las aplicaciones lo que puede llegar a complicar la tarea de configurar nuevamente o desinstalar una aplicación.

Tratando el tema sobre utilizar un equipo con sistema operativo OpenSource como lo es GNU/Linux en su distribución Debian, por experiencia propia puedo decir que es bastante recomendable, ya que permite realizar las configuraciones del servidor de desarrollo de manera muy personalizada, lo cual facilitara la tarea del mantenimiento del equipo, un mejor control sobre el sistema de archivos y posteriores instalaciones de aplicaciones que necesiten la conexión o convivencia con las que ya se tienen en el equipo, porque como se puede observa los flujos de

instalación siguen la siguiente estructura: todos los archivos de instalación, en su estado de archivos comprimidos se encuentran en la ruta `/usr/local/src` esto con el fin de que si se requiere instalar nuevamente cualquier aplicación, se puedan localizar sus archivos fuente ágilmente. Las aplicaciones se instalan en la ruta `/usr/local/`, se elige este directorio por ser el indicado para contener las aplicaciones del usuario. Las librerías y bibliotecas de las cuales dependen las aplicaciones, también se encuentran instaladas bajo la ruta `/usr`.

Después de analizar los flujos de instalación realizados se puede llegar a una conclusión sobre como se debe organizar el sistema de archivos en el equipo, las recomendaciones que proporciono son: crear particiones independientes en el disco duro para los directorio `/home`, `/usr`, `/etc` y el de raíz(`/`), además crear una Copiar de respaldo de los archivos principales de configuración de las aplicaciones instaladas antes de cada configuración y si esta fue exitosa después de hacerla. Aunque simples las recomendaciones, serán de gran ayuda por si se requiere volver a instalar el sistema operativo.

Por último se enfatiza que los flujos de instalación que se ejecutaron y los cuales se describen en este manual fueron reunidos a través de la experiencia que surgió por la necesidad de trabajar con un equipo que se encontrara configurado para poder realizar tareas de desarrollo de software. Aunque existe más de una forma de instalar y configurar las aplicaciones aquí propuestas, las descritas en este manual son las que mejor funcionaron y más por hecho de que siguen una secuencia para evitar los problemas de dependencias. Sin embargo sugiero que realicen la instalación de estas mismas aplicaciones pero siguiendo las instrucciones de instalación que pueden localizar en los sitios oficiales, ya que conforme vayan avanzando surgirán una infinidad de errores de instalación, los cuales incitaran a iniciar la búsqueda de soluciones en la web y eso conlleva a leer y aprender más sobre GNU/Linux lo cual enriquece mucho y proporciona los conocimientos necesarios para realizar deducciones asertivas sobre problemas que surjan al realizar tareas de configuración y mantenimiento en momentos posteriores.

Glosario de comandos

Apt-get: Es utilizado para realizar la administración de los paquetes .deb que se encuentran en los repositorios de paquetes de Debian en la web. Con el se puede instalar, buscar o borrar paquetes.

Sintaxis: apt-get [opciones]

Cd: Es utilizado para desplazarse entre los directorio del sistema de archivos.

Sintaxis: cd [/url destino]

Chgrp: Es utilizado para asignar o cambiar el grupo de trabajo al cual pertenece un archivo o un directorio.

Sintaxis: chgrp [opciones] [archivos o directorios]

Chmod: Es utilizado para cambiar los permisos de acceso a los archivos y directorios del sistema.

Sintaxis: chmod [opciones] [archivos o directorios]

Chown: Es utilizado para asignar o cambiar de propietario los archivos y directorios del sistema.

Sintaxis: chown [opciones] [archivos o directorios]

Configure: Es un script que se encuentra en el paquete de archivos fuente, el cual contiene las instrucciones necesarias para configurar el ambiente antes de compilar una aplicación.

Sintaxis: ./configure [parámetros de configuración]

Cp: Es utilizado para realizar el copiado de archivos o directorios completos.

Sintaxis: cp [opciones] archivo-origen directorio-destino

Exit: Es utilizado para terminar una sesión en la consola de comandos.

Sintaxis: exit

Groupadd: Es utilizado para agregar un grupo de trabajo en el sistema operativo.

Sintaxis: groupadd [opciones] [nombre de grupo]

Ldconfig: Es utilizado para cargar en el cache las ligas de las bibliotecas compartidas.

Sintaxis: ldconfig [-v]

Ln: Es utilizado para la creación de ligas simbólicas que funcionan como accesos directos a archivos y directorio.

Sintaxis: ln [opciones] [Ruta de archivo origen] [Ruta de archivo destino]

Make: Es utilizado para realizar la recopilación de las piezas del programa si así se requiere.

Sintaxis: make [install]

Pkg-config: Es utilizado para configurar la bibliotecas de los programas instalados.

Sintaxis: pkg-config [opciones]

Source: Es utilizado para evaluar el contenido de un script e interpretarlo en la consola de comandos.

Sintaxis: source [ruta de script]

Su: Iniciar sesión con la configuración de usuario que se le pase como parámetro.

Sintaxis: su [usuario]

Tar : Es utilizado para empaquetar o desempaquetar archivos, empaquetar significa guardar en un sólo archivo una lista de varios archivos, o el contenido de todo un directorio.

Sintaxis: tar [opciones] [archivo1][archivo2]...[archivoN]

Unzip: Es utilizado para desempaquetar un archivo comprimido en formato zip.

Sintaxis: unzip [opciones] [archivo1][archivo2]...[archivoN]

Update-alternatives: Es utilizado para darle mantenimiento a los enlaces simbólicos de las alternativas de ejecución de comandos en la consola de línea de comandos.

Useradd: Es utilizado para agregar usuarios al sistema operativo y asignarle un grupo de trabajo.

Sintaxis: `useradd [opciones] [nombre de usuario]`

Vi: Editor de textos en línea de comandos que se encuentra instalado por default en la mayoría de las distribuciones GNU/Linux, puede ser muy difícil de manejar para usuarios principiantes en el manejo de línea de comandos.

Referencias

Apuntes de los cursos de capacitación del plan de becas de alto rendimiento de la Subdirección de Sistemas de la UNAM.

Diccionario de términos de informática online:
<http://lawebdelprogramador.com/diccionario/>

Sitios oficiales

MySQL: <http://www.mysql.com/>
Postgres: <http://www.postgresql.org/>
Cherokee: <http://www.cherokee-project.com/>
Apache: <http://www.apache.org/>
Tomcat: <http://tomcat.apache.org/>
Ruby: <http://www.ruby-lang.org/es/>
Java: <http://java.sun.com/javaee/>
PHP: <http://php.net/>
Eclipse: <http://www.eclipse.org/>
DBDesigner4: <http://www.fabforce.net/dbdesigner4/>
Subversion: <http://subversion.tigris.org/>
Terminator: <http://www.tenshu.net/terminator/>