

INTRODUCCIÓN

Este tema de monografía fue escogido por la evidente necesidad de documentación clara sobre el proceso de Instalación y variantes dentro de la configuración del Servidor de Aplicaciones Tomcat, el mismo que es relativamente nuevo y de gran utilidad para el alojamiento de aplicaciones web.

Una de las ventajas de utilizar Tomcat como servidor de aplicaciones y el sistema Operativo Linux, es que son programas de libre distribución, lo que facilita la obtención de los mismos y evidenciará un notable ahorro económico a quién lo utilice.

El sistema Operativo Linux es considerado un sistema operativo robusto y estable, es multiusuario y multitarea, es decir pueden existir varios usuarios utilizando un computador al mismo tiempo, y varios procesos ejecutándose a la vez.

Nuestra investigación lo que pretende es elaborar el manual de instalación de Tomcat, para que sirva de referencia para futuras instalaciones, la metodología utilizada para el desarrollo de la monografía se basa en un sustento teórico, debidamente probado con la ejecución de una aplicación.

CAPÍTULO 1

SERVIDOR DE APLICACIONES TOMCAT

1.1 Introducción a Tomcat

Tomcat, también llamado “Jakarta Tomcat” o “Apache Tomcat”, es un servidor de aplicaciones, fue desarrollado bajo el proyecto Jakarta, en Apache Software Foundation, creado James Duncan Davidson, un empleado de Sun a mediados del año 1998, quién utilizó el nombre de Tomcat, traducido al lenguaje español como “gato”, pretendiendo representar las capacidades del programa, que se identifican con las del animal de ser independiente y de cuidarse por si mismo.

Al ser el Servidor de Aplicaciones Tomcat mantenido y desarrollado por Apache Software Foundation, los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence, pudiendo de esta forma manipular directamente el código, de acuerdo a los intereses y necesidades de cada usuario.

En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era solo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día Tomcat es usado como servidor web autónomo, en entornos con alto nivel de tráfico y alta disponibilidad. Tomcat funciona como un contenedor de servlets. Un Servlet es un programa que se ejecuta en un servidor, y un contenedor de servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario, puede funcionar como servidor web por sí mismo.

Las primeras versiones de Tomcat fueron las 3.0x, en el año 2001 se distribuye la versión 4.0x y se utiliza el contenedor de Servlets Catalina, Catalina fue un rediseño completo de la arquitectura de Tomcat, hoy en día se distribuyen las versiones 6.x.

1.2 Características de Tomcat

A continuación se mencionan algunas de las características principales de cada una de las versiones conocidas de Tomcat.

Tomcat 3.x

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de servlets

Tomcat 4.x

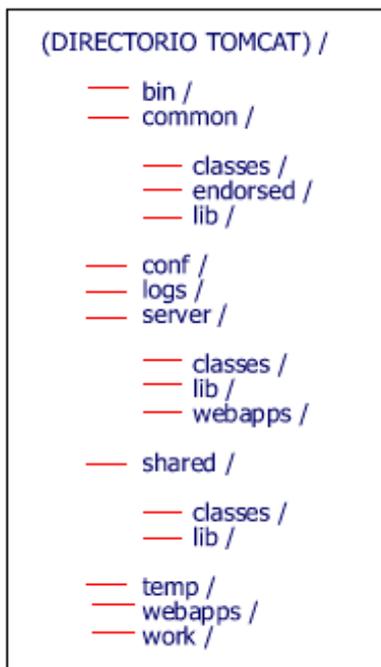
- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de servlets rediseñado como Catalina
- Motor JSP rediseñado con Jasper
- Conector Coyote
- Java Management Extensions (JMX), JSP y administración basada en Struts.

Tomcat 5.x

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas.

1.3 Estructura de directorios de Tomcat

Una vez que se ha descomprimido la distribución binaria de Tomcat, deberíamos tener la siguiente estructura de directorios:



Directorio	Descripción
/bin	Contiene los Scripts de arrancar- parar Tomcat
/common	Clases comunes que pueden utilizar Catalina y las aplicaciones web
/conf	Contiene varios ficheros de configuración incluyendo server.xml (Fichero principal de Tomcat) y web.xml que configura los valores por defecto para las distintas aplicaciones en Tomcat
/logs	Aquí es donde Tomcat guarda los archivos de registros.
/server	Contiene clases para uso interno y exclusivo de Catalina
/shared	Contiene clases compartidas por todas las aplicaciones web
/webapps	Directorio base para las aplicaciones Web.

Adicionalmente se creará los siguientes directorios.

Directorio	Descripción
/work	Generando automáticamente por Tomcat, aquí se sitúan los ficheros intermedios. Si se elimina este directorio mientras se esta ejecutando Tomcat no se podrá ejecutar paginas JSP.
/classes	Podemos crear este directorio para añadir clases adicionales al classpath.

1.4 Scripts de Tomcat

Tomcat es un programa Java, por lo tanto es posible ejecutarlo desde la línea de comandos, después de configurar algunas variables de entorno. El equipo de desarrollo de Tomcat, proporciona algunos scripts para arrancar y parar Tomcat fácilmente, a continuación se detalla algunos de los scripts más importantes:

Script	Descripción
tomcat	Es el Script principal, configura el entorno apropiado, incluyendo CLASSPATH, TOMCAT HOME y JAVA HOME, también arranca Tomcat con los parámetros de la línea de comando apropiados.
startup	Script para arrancar Tomcat en segundo plano, acceso directo para tomcat start.
shutdown	Script para parar o apagar Tomcat, acceso directo para Tomcat stop.

El script más importante para los usuarios es tomcat, (tomcat.sh/tomcat.bat). Los otros scripts relacionados con tomcat realizan tareas específicas de forma simplificada desde la línea de comandos. A continuación se detallan las acciones que realiza el script tomcat.

- Averigua donde está TOMCAT_HOME si no se especifica
- Averigua donde está JAVA_HOME si no se especifica
- Configura una CLASSPATH que contiene:
 - El directorio \${TOMCAT_HOME}/classes(si existe).
 - Todo el contenido de \${TOMCAT_HOME}/lib.
 - \${JAVA_HOME}/lib/tools.jar
- Ejecuta java con los parámetros de la línea de comandos que han configurado un entorno de sistema Java, llamado tomcat.home, con org.apache.tomcat.startup.Tomcat como la clase de arranque. También procesa los parámetros de la línea de comandos para org.apache.tomcat.startup.Tomcat, como:
 - La operación a ejecutar start/stop/run.etc.
 - Un path al fichero server.xml usado por este proceso Tomcat.

1.5 Ficheros de configuración de Tomcat

La configuración de Tomcat se basa en dos ficheros:

1. server.xml
2. web.xml

1.5.1 Fichero server.xml

Es el fichero de configuración principal de Tomcat. Sirve para dos objetivos.

1. Proporcionar la configuración inicial para los componentes de Tomcat.
2. Especifica la estructura de Tomcat, lo que significa permitir que Tomcat arranque y se construya a sí mismo ejemplarizando los componentes especificados en server.xml.

Es importante conocer la arquitectura de Tomcat, ya que cada uno de sus componentes está cercanamente asociado con las funciones del servidor. La arquitectura de Tomcat está estructurada en diferentes niveles de jerarquía, algunos de estos componentes son llamados “top level” o componentes de alto nivel, porque se encuentran en el nivel mas alto de la jerarquía. También existen los llamados “Containers” o Contenedores, quienes pueden contener una colección de otros componentes. Los llamados “nested components” o nidos de componentes pueden residir en contenedores pero no pueden ellos mismos contener otros componentes.

El archivo server.xml se encuentra ubicado dentro del directorio /usr/share/tomcat/conf donde /usr/share/Tomcat es el directorio definido en CATALINA_HOME.

Como cualquier otro documento en XML todo contenido entre <!--> se lo considera como un comentario, y por lo tanto cualquier parámetro que se encuentre dentro de estos caracteres no es utilizado por “Tomcat”, los parámetros que no sean definidos dentro del archivo server.xml son asignados un valor “Default” por Tomcat.

El siguiente listado contiene los parámetros principales del archivo de configuración server.xml e indica su utilización:

Server: <Server> es el elemento principal del archivo server.xml y todas las demás secciones deben encontrarse entre estos nodos; el atributo port indica el puerto TCP donde se espera la señal de cierre (shutdown) de Tomcat.

Listener: Con este elemento se configuran las extensiones JMX (“Java Management Extensions”) que serán utilizados por Tomcat, estos elementos toman dos atributos: className que indica la Clase diseñada para escuchar sobre eventos JMX y debug para especificar el nivel de “debug” generado al tiempo de ejecución.

Service: Este parámetro permite configurar Tomcat para diferentes modalidades de ejecución, en el archivo server.xml. La definición asignada a través del atributo name= “Catalina” es empleada para ejecutar Tomcat individualmente.

Connector: Este elemento representa las conexiones (Puertos TCP) que serán abiertas cuando Tomcat arranque, dentro de cada elemento Connector se definen diversos atributos los cuales dan mas detalles acerca de la conexión. El elemento connector más importante es aquel que define la clase HttpConnector.



La declaración anterior indica que Tomcat está dispuesto a recibir requerimientos que arriben por el puerto 8080 del “Host” donde el Tomcat está instalado. El puerto 80 es ampliamente conocido como puerto TCP 80 con el que intenta comunicarse cualquier navegador en Internet.

Engine: Los elementos <Engine> deben encontrarse dentro de <Service>, estos representan el mecanismo que atenderá todas las solicitudes requeridas, todas las definiciones Connector serán procesadas por <Engine>.

Los atributos de este elemento son los siguientes:



En donde “defaultHost” representa el nombre del servidor Tomcat mientras “debug” indica el nivel de debug generado al tiempo de ejecución.

Logger: Los elementos Logger indican a Tomcat hacia donde deben ser enviados los registros “Logs”.



El gráfico anterior indica que los registros de Tomcat deben ser enviados al archivo localhost_log.txt.

Host: Estos elementos permiten definir varios Hosts Virtuales para Tomcat, esto es, a través del elemento <Engine> se define un sitio (localhost) para atender solicitudes, a través de Host es posible definir sitios “Virtuales”.

Context: Es un elemento utilizado para indicar la ubicación de las aplicaciones ejecutadas en Tomcat, en su configuración “Default”, estas aplicaciones se encuentran dentro del directorio webapps bajo el directorio raíz de Tomcat (/usr/share/tomcat).

Una aplicación en Tomcat o cualquier Máquina Servlet es un conjunto de (Java Server Pages) “JSP’s” y/o Servlets agrupados con ciertos parámetros de arranque y seguridad, este conjunto de archivos es conocido como un WAR (Web-Archive).

5.1.2 Fichero Web.xml

Este fichero es utilizado en dos lugares en Tomcat: En el directorio `$CATALINA_BASE/conf` y en cada aplicación web. Hay una pequeña característica de Tomcat que esta relacionada con `web.xml`. Tomcat permite al usuario definir los valores por defecto de `web.xml` para todos los contextos poniendo un fichero `web.xml` por defecto en el directorio `conf`. Cuando se construye un nuevo contexto, Tomcat usa el fichero `web.xml` por defecto como la configuración base y el fichero `web.xml` específico de la aplicación.

`Web.xml` contiene varios elementos que se encuentran definidos dentro del elemento `web-app`, que es el elemento `root`, a continuación figura un listado, con la descripción de cada uno de estos:

Display-name: Aquí se configura el nombre de la aplicación web que se mostrará en los listados de Tomcat.

```
<display-name>Aplicación web ejemplo </display-name>
```

Description: Configura la descripción de la aplicación Web a visualizar.

```
<description> Descripción de la aplicación </description>
```

Listener: Configura la aplicación web para que se ejecute un oyente de contexto Servlet cuando arranque o pare la aplicación Web. En el ejemplo, se selecciona que el oyente de contexto Servlet de la aplicación web ejemplo sea `webejemplo.ContextListener`

```
<listener>  
<listener-class>webejemplo.ContextListener</listener-class>
```

Welcome-file-list: Configura el punto de entrada de la aplicación web. Las preferencias por defecto para este parámetro se seleccionan en el fichero `conf/web.xml` de Tomcat. Por defecto el archivo `web.xml` se visualiza de la siguiente manera:

```
<welcome-file-list>  
<welcome-file>index.html</welcome-file>  
<welcome-file>index.htm</welcome-file>  
<welcome-file>Home.jsp</welcome-file>  
</welcome-file-list>
```

CAPÍTULO 2

DESCRIPCIÓN DE MYSQL

2.1 Introducción a MySQL

El lenguaje de Lectura Estructurado (SQL) fue comercializado en 1981 por primera vez por la empresa IBM, desde entonces SQL ha sido considerado un estándar para las bases de datos estructuradas. En 1990 se intento utilizar SQL, sin embargo al darse cuenta que no satisfacía necesidades de flexibilidad y rapidez, sus creadores se vieron obligados a crear una API SQL denominada MySQL.

MySQL es un sistema de administración de base de datos, multiusuario y multihilo, es la base de datos de código fuente abierto mas utilizada en el mundo, tiene un esquema de licenciamiento dual, es decir que adicional a la GNU (GPL General Public License), que es de uso libre ofrece productos privativos que pueden ser adquiridos mediante una versión comercial licenciada.

Existen algunas APIs que permiten, a aplicaciones realizadas en diferentes lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, Java, entre otros, cada uno de estos utiliza una API especifica. MySQL es muy utilizado en aplicaciones web, bajo plataformas Linux.

2.1 Características de MySQL.

- Licenciamiento GNU (GPL General Public License).
- Arquitectura flexible
- OLTP y Transacciones
- ANSI SQL Estándar
- Replicación Cluster y alta Disponibilidad
- Respaldos
- Manejo del Almacenamiento
- Características optimizadas para aplicaciones Web

- Características optimizadas para aplicaciones Data Warehouse.
- Seguridad
- Buen desempeño, fácil de usar.
- Herramientas graficas y multiplataforma.

CAPÍTULO 3

DESCRIPCIÓN DEL LENGUAJE DE PROGRAMACIÓN JAVA

3.1 Introducción a Java

Java es el lenguaje de programación desarrollado por Sun Microsystems, creado para resolver simultáneamente todos los problemas que encuentran los desarrolladores de software, debido a la incompatibilidad encontrada en el mercado causado por la gran cantidad de sistemas operativos y maquinas existentes. Algunos autores hablan sobre el origen de Java, algunos de estos coinciden en que inicialmente no se le dio la atención requerida, hasta que finalmente encontró su nicho de mercado de Internet.

3.2 Características del lenguaje de Programación Java

Las Características principales que nos ofrece el lenguaje de Programación Java respecto a cualquier lenguaje de programación son:

Simple: Java se diseñó para ser parecido a C++, excepto por la no-incorporación de características confusas y no utilizadas en el lenguaje C++, facilitando de esta manera su utilización y el aprendizaje.

Orientado a objetos: Java trabaja con sus datos como objetos y con interfaces a esos objetos, Java soporta las características de encapsulamiento, herencia y polimorfismo, características de la orientación a objetos. Las plantillas de objetos son llamadas clases y sus copias instancias, las instancias, necesitan ser construidas y destruidas en espacios de memoria. Java incorpora una importante funcionalidad que es la resolución dinámica de métodos.

Distribuido: Java se ha construido con importantes capacidades de interconexión TCP/IP. Con la existencia de librerías de rutinas para interactuar con protocolos como http y ftp, se puede acceder a la información a través de la red con mucha facilidad.

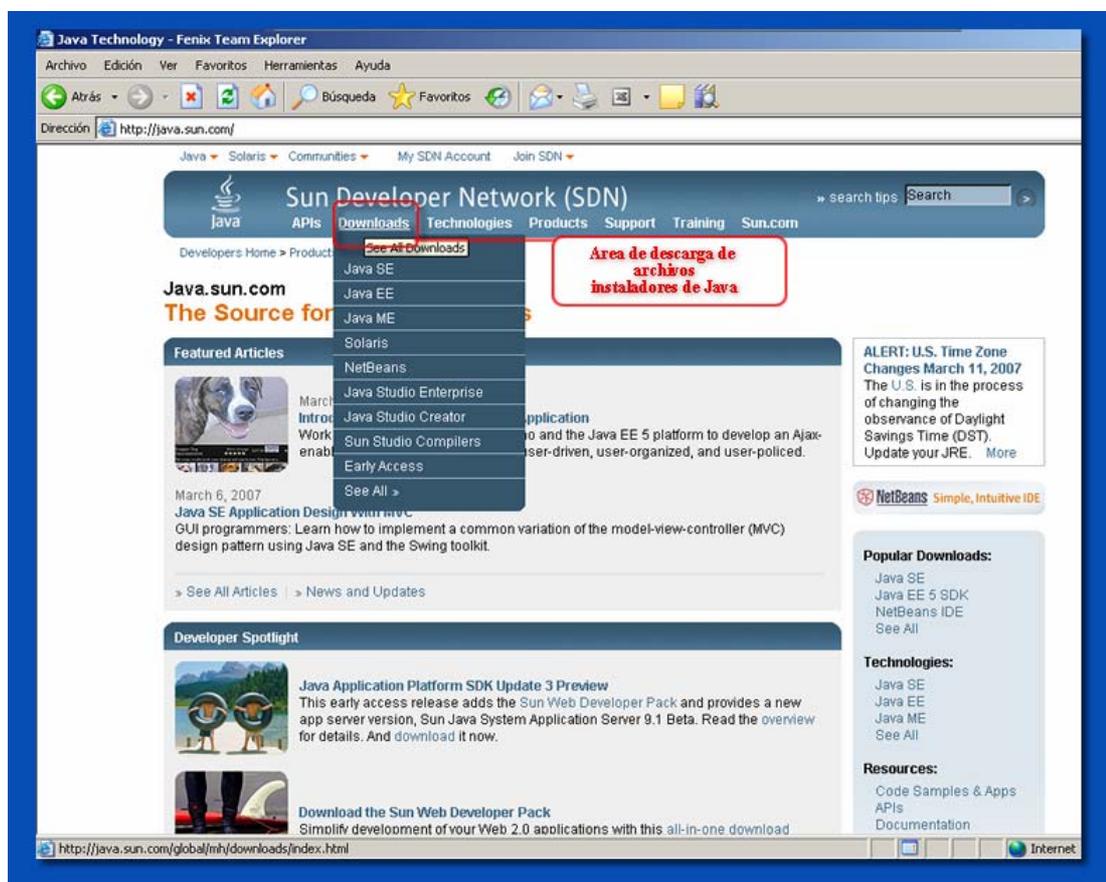
Seguro: Java a sido diseñado poniendo una especial atención en el tema de la seguridad, ya que dada la naturaleza distribuida de este lenguaje de programación, donde los applets se bajan desde cualquier punto de la red, se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

Robusto: Java fue desarrollado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores(aritmética de punteros), ya que se han prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

CAPÍTULO 4

REQUISITOS PREVIOS A LA INSTALACIÓN DE TOMCAT

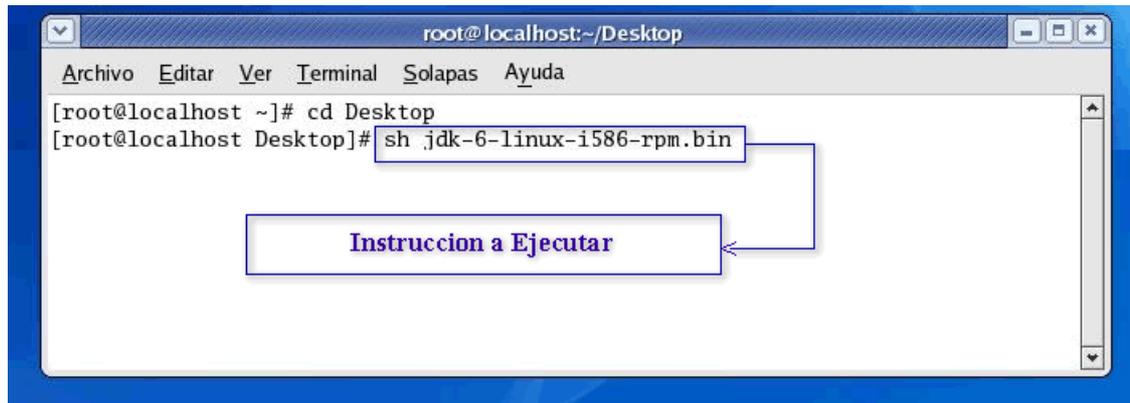
Utilizando el sistema Operativo Linux, antes de empezar con la instalación de Tomcat, se debe instalar Java. Para la instalación de Java primeramente se necesita acceder a la dirección web www.java.sun.com y descargar el archivo `jdk-6-linux-i586-rpm.bin`.



Una vez que se ha descargado el archivo y su contenido ha sido extraído como un archivo de extensión `bin`, se ejecuta la siguiente instrucción en la terminal de comandos:

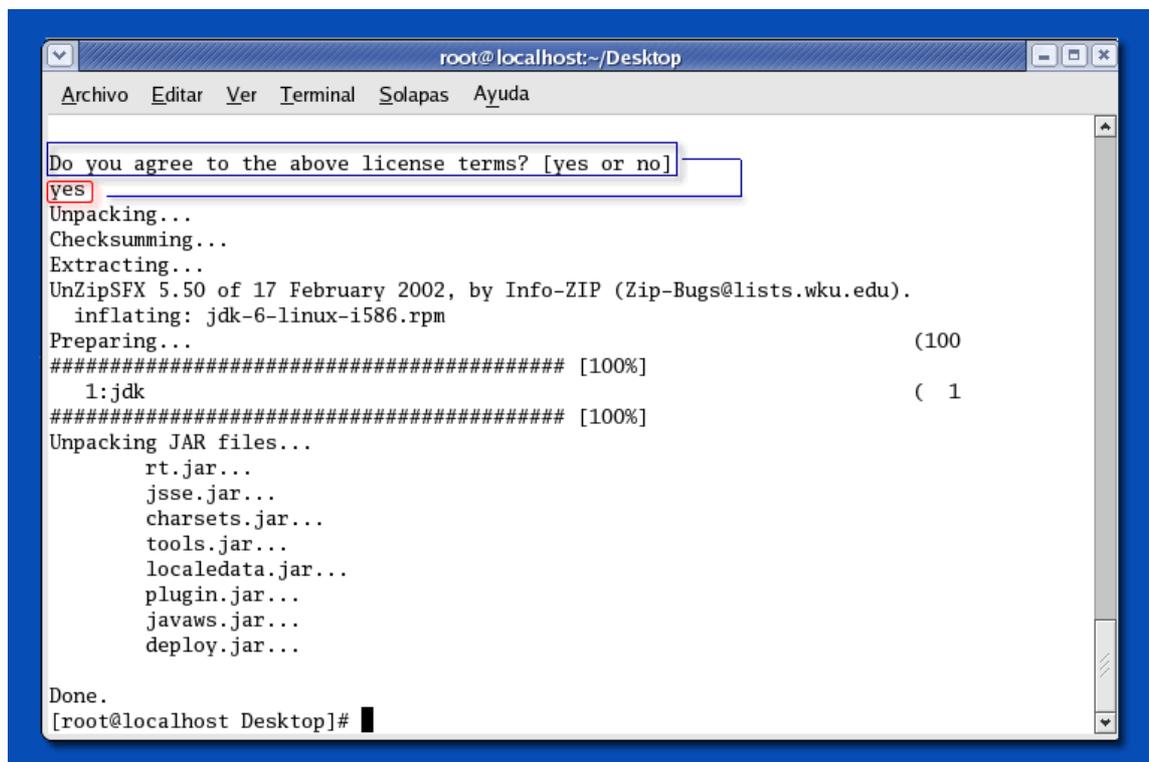
```
sh archivo_de_instalación_java.bin
```

Se puede observar en el siguiente grafico



En este caso el archivo de instalación se denomina jdk-6-linux-i586-rpm.bin.

A continuación se podrá observar una pantalla como la que se muestra en el siguiente gráfico, donde se deberá responder “yes” si está de acuerdo con los términos de la licencia.



Al finalizar la instalación de Java, en la terminal de comandos, aparecerá el mensaje “DONE”..

CAPÍTULO 5

INSTALACIÓN Y CONFIGURACIÓN DE TOMCAT

5.1 Primeros pasos

Como requisito previo a la Instalación de Tomcat es necesario que se haya instalado Java. Una vez que la instalación de Java ha finalizado, se deberá modificar el archivo “profile” que se encuentra ubicado en /etc/, aquí se deberán agregar la variable de entorno \$JAVA_HOME como se indica a continuación, ya que JAVA_HOME es la variable que utiliza Tomcat para poder arrancar.

```
JAVA_HOME=/ruta_a_la_carpeta_principal_de_Java
export JAVA_HOME
```

Adicionalmente a esto se puede crear la variable JAVA_OPTS que es la variable de entorno, que sirve para dar otras opciones de ejecución a Java cuando arranca Tomcat, opciones relacionadas con la memoria y también para generar pdf sin que el servidor grafico de Linux este levantado.

El servidor de aplicaciones Apache, trabaja con 64 MB de memoria, si bien es cierto el rendimiento es bueno pero en muchos de los casos esto podría resultar insuficiente, por esto la memoria utilizable por Tomcat es un parámetro configurable, a través de la siguiente opción, que se hará en el archivo utilizado anteriormente, y se deberá incluir las siguientes líneas:

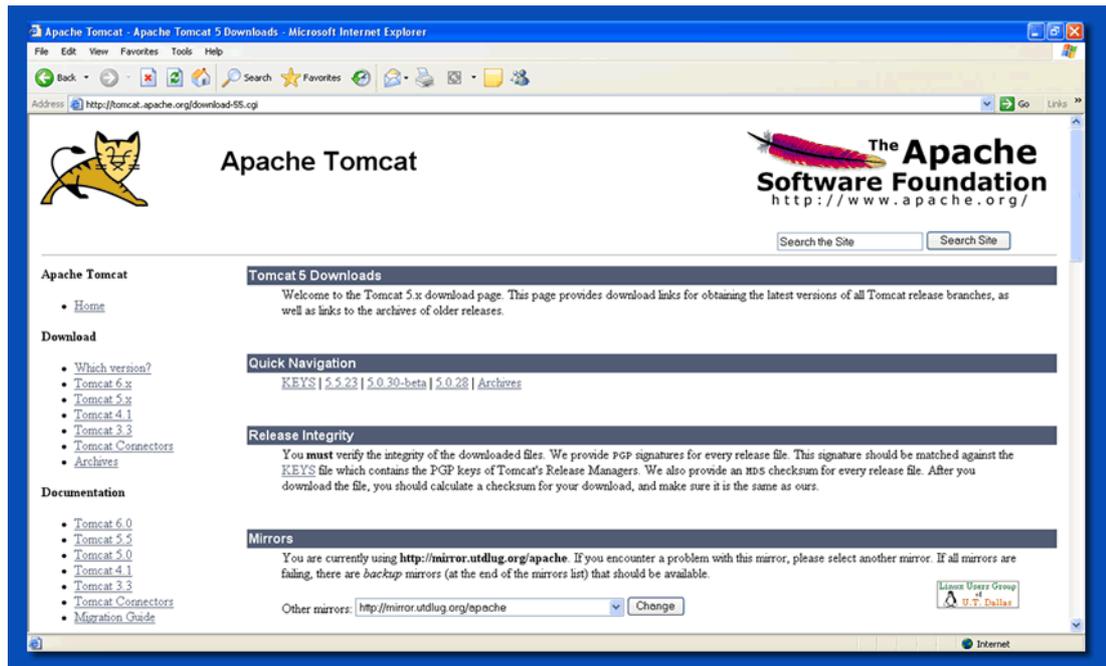
```
JAVA_OPTS="-Xms256m -Xmx256m -Djava.awt.headless=true"
export JAVA_OPTS
```

Las opciones (-Xms y -Xmx) indican que la memoria máxima y memoria mínima a utilizar es de 256, al no declarar estos valores se asume 64MB por defecto, y la tercera opción es la requerida para generar pdf sin interfaz gráfica.

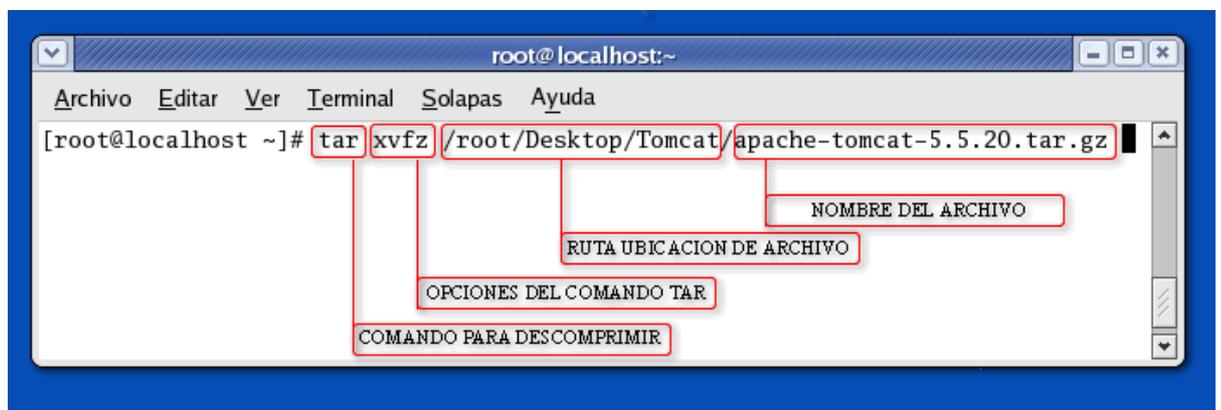
5.2 Instalación de Tomcat

La instalación de Tomcat sobre Linux es sencilla, primeramente se deberá descargar el archivo `apache-tomcat-5.5.20.tar` desde la siguiente dirección:

<http://jakarta.apache.org/tomcat>



Una vez descargado el archivo, se deberá descomprimirlo, esta instrucción se la realiza con el comando "tar", tal como se muestra a continuación:



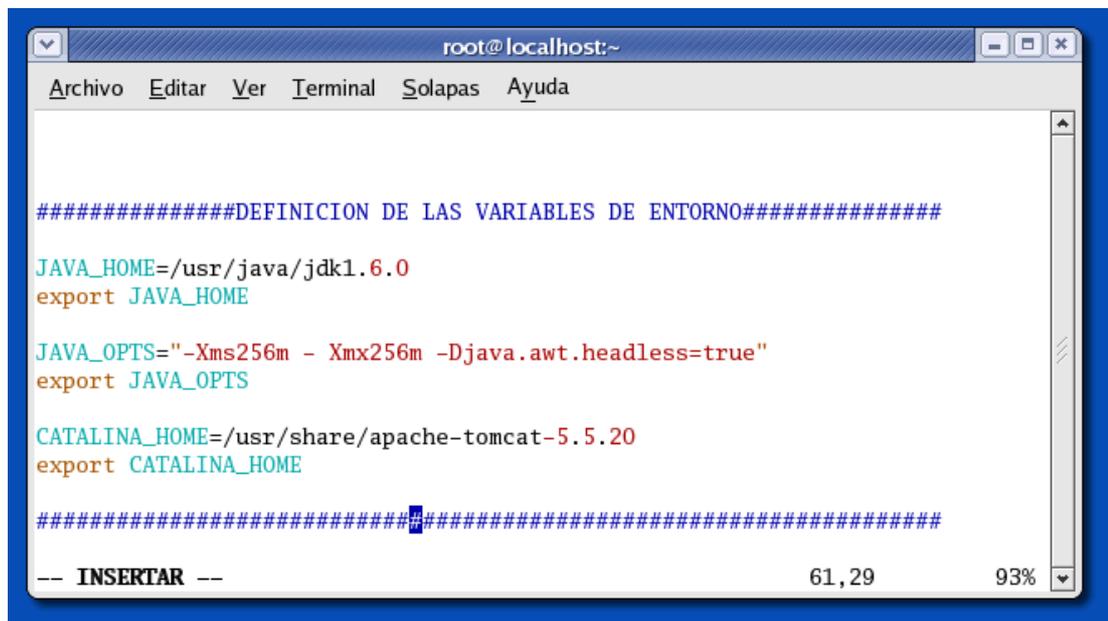
Al descomprimir el archivo se generará una nueva carpeta en el mismo directorio donde se encuentra el archivo comprimido de Tomcat. Se puede mover el directorio a otra carpeta como la siguiente:

```
/usr/share/apache-tomcat-5.5.20
```

Seguido de esto se debería exportar la variable \$CATALINA_HOME, esto se lo realiza desde la terminal de comandos, se puede también agregar dicha instrucción al archivo profile utilizado en el capítulo anterior. La instrucción para exportar la variable \$CATALINA_HOME es la siguiente.

```
CATALINA_HOME=/usr/share/apache-tomcat-5.5.20
export CATALINA_HOME.
```

Al final el archivo "profile" se verá como se indica a continuación.



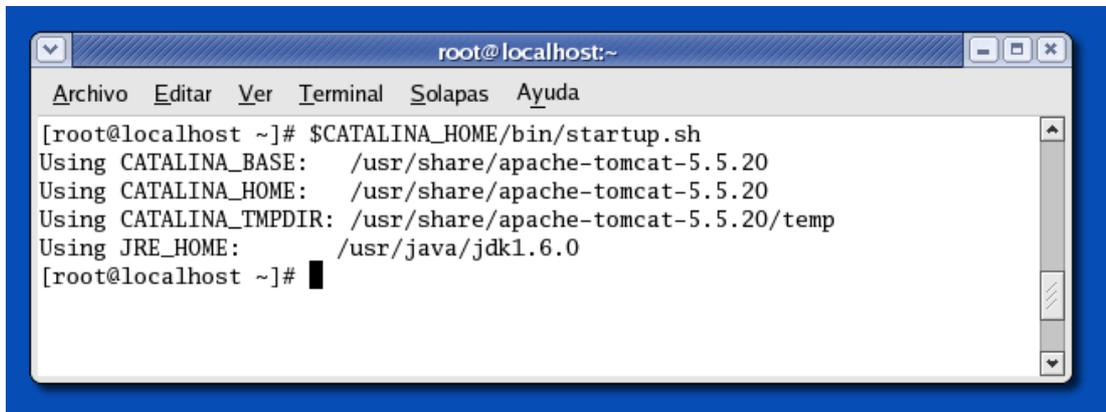
```
root@localhost:~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

#####DEFINICION DE LAS VARIABLES DE ENTORNO#####
JAVA_HOME=/usr/java/jdk1.6.0
export JAVA_HOME
JAVA_OPTS="-Xms256m -Xmx256m -Djava.awt.headless=true"
export JAVA_OPTS
CATALINA_HOME=/usr/share/apache-tomcat-5.5.20
export CATALINA_HOME
#####
-- INSERTAR --                               61,29          93%
```

Se puede arrancar Tomcat, con la siguiente instrucción:

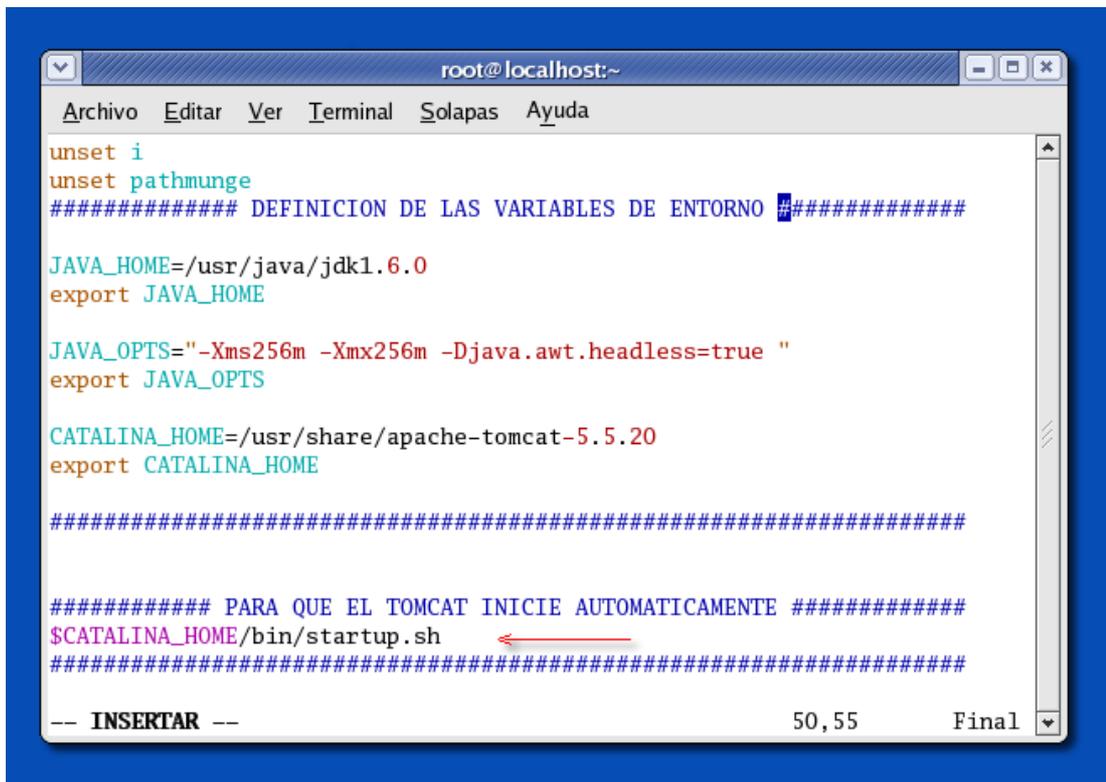
```
$CATALINA_HOME/bin/startup.sh
```

Si todo lo indicado anteriormente se realizó correctamente, y el servidor arrancó sin ningún problema, en la terminal se podrá visualizar una pantalla como la que se muestra a continuación:



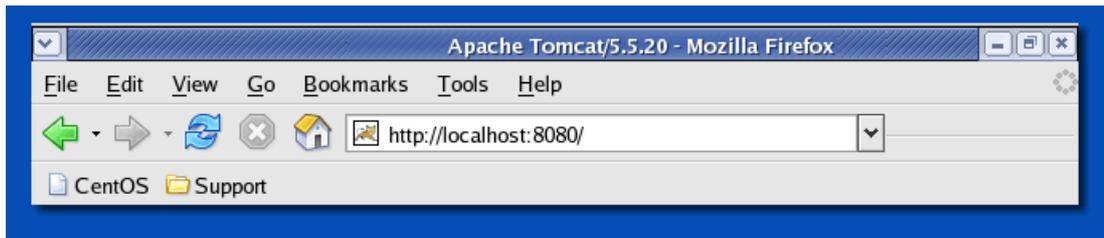
```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@localhost ~]# $CATALINA_HOME/bin/startup.sh  
Using CATALINA_BASE:   /usr/share/apache-tomcat-5.5.20  
Using CATALINA_HOME:   /usr/share/apache-tomcat-5.5.20  
Using CATALINA_TMPDIR: /usr/share/apache-tomcat-5.5.20/temp  
Using JRE_HOME:        /usr/java/jdk1.6.0  
[root@localhost ~]#
```

Si se desea que el Servidor de Aplicaciones Tomcat arranque automáticamente al iniciar el sistema, se debería incluir la línea mencionada anteriormente al archivo "profile", de esta manera:

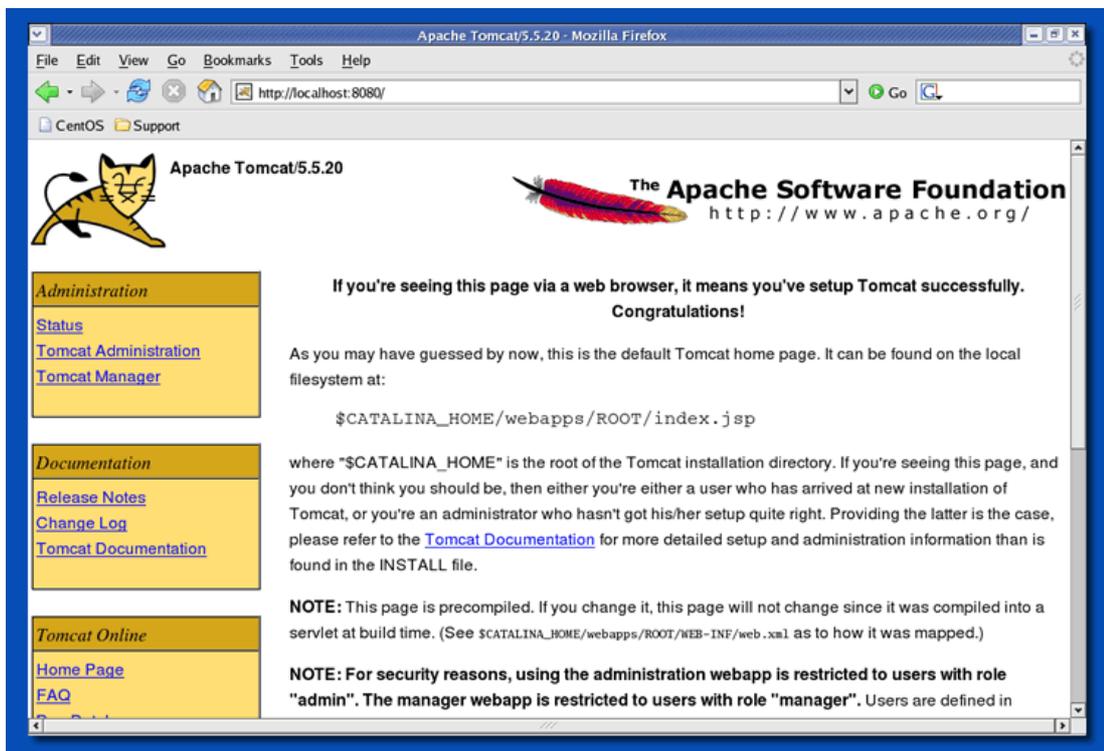


```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
unset i  
unset pathmunge  
##### DEFINICION DE LAS VARIABLES DE ENTORNO #####  
  
JAVA_HOME=/usr/java/jdk1.6.0  
export JAVA_HOME  
  
JAVA_OPTS="-Xms256m -Xmx256m -Djava.awt.headless=true "  
export JAVA_OPTS  
  
CATALINA_HOME=/usr/share/apache-tomcat-5.5.20  
export CATALINA_HOME  
  
#####  
  
##### PARA QUE EL TOMCAT INICIE AUTOMATICAMENTE #####  
$CATALINA_HOME/bin/startup.sh  
#####  
  
-- INSERTAR -- 50,55 Final
```

Para confirmar que Tomcat esta ejecutándose correctamente, colocar lo siguiente en la barra de direcciones del explorador.



Luego de escribir la dirección mencionada anteriormente en la barra de direcciones, se podrá visualizar la siguiente pantalla de Tomcat, si todo lo realizado anteriormente esta correcto.



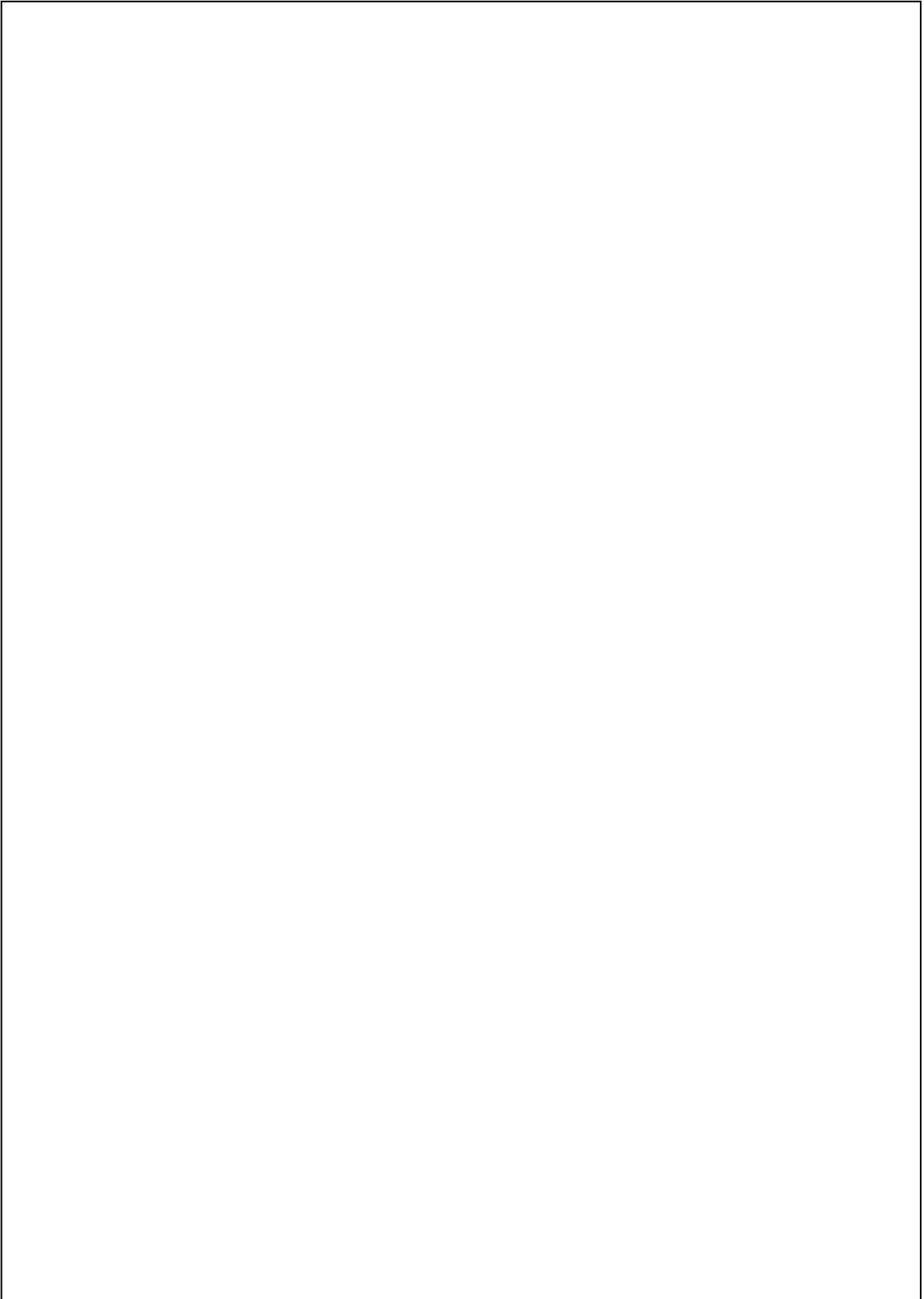
La Instrucción para parar el servidor Tomcat es la siguiente:

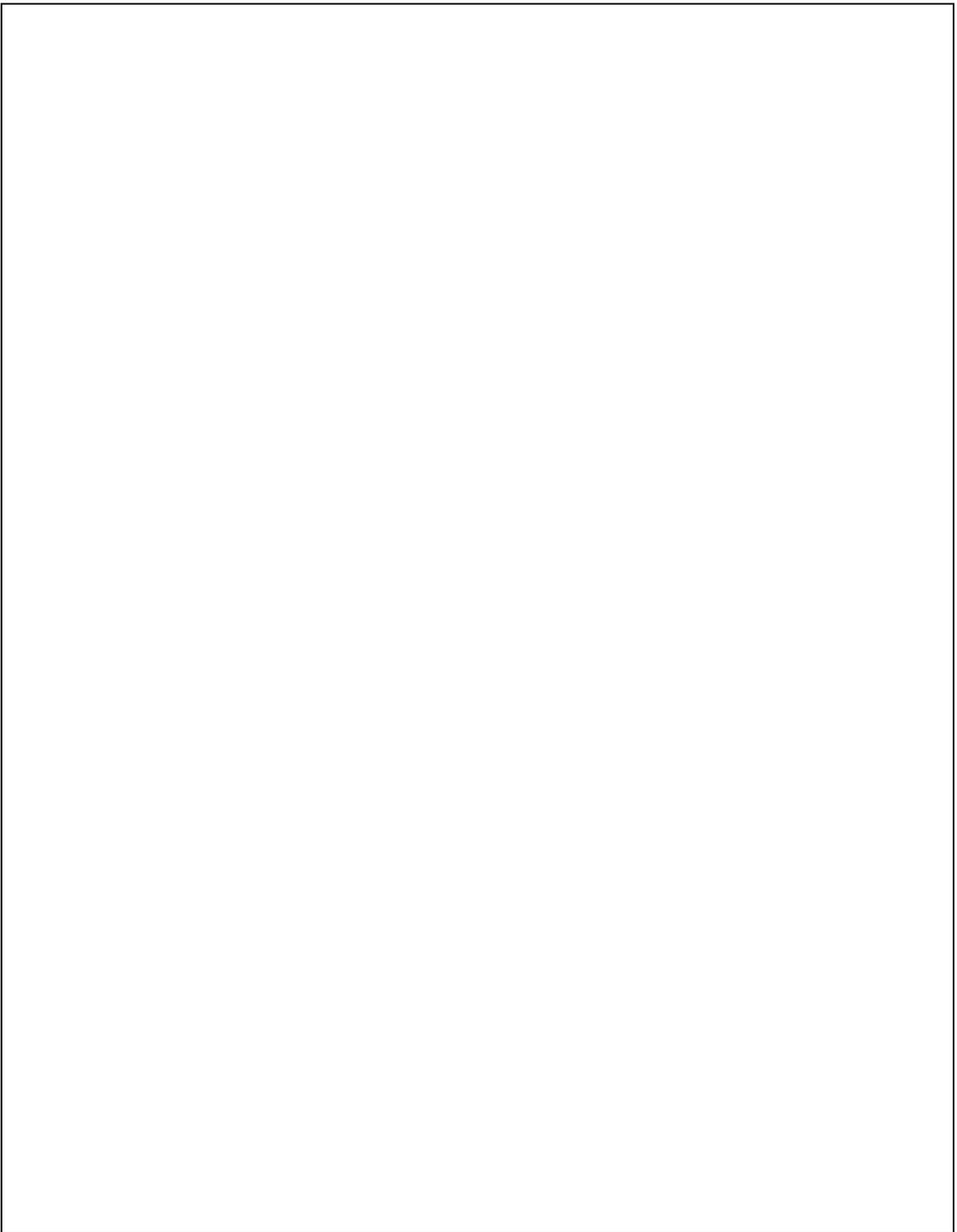
```
$CATALINA_HOME=/bin/shutdown.sh
```

Otra opción para arrancar automáticamente el servidor Tomcat, es realizar un script, de esta manera, crearemos un servicio llamado Tomcat, el cual se va a cargar al iniciar la máquina y además se podrá ejecutar con los parámetros “start”, “stop”, “restart” y “status”, al igual que cualquier otro servicio del sistema.

Los pasos para realizar el script son los siguientes:

Creamos un archivo llamado “tomcat” en cualquier editor de textos, ubicándolo en /etc/init.d/, dicho archivo contendrá las líneas detalladas a continuación:





Luego de que el archivo se ha creado, desde la terminal de comandos ejecutar las siguientes líneas:

Creación del servicio:

```
# chmod 755 /etc/init.d/tomcat  
# chkconfig --add tomcat
```

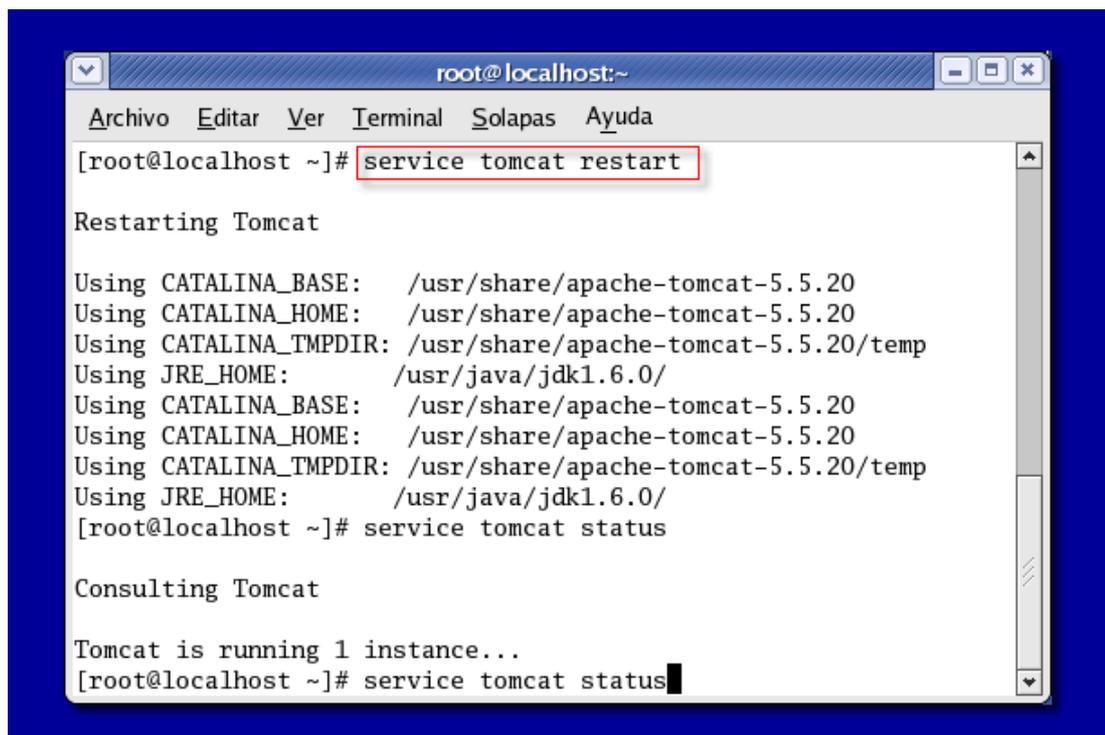
Para hacer que el servicio inicie al arrancar el sistema:

```
# chkconfig --level 35 tomcat on
```

Luego de esto se podrá invocar el servicio tomcat con el comando service, de la siguiente manera:

```
service tomcat start → Inicia el servicio  
service tomcat stop → Para el servicio  
service tomcat restart → Reinicia el servicio  
service tomcat status → Indica el estado del servicio
```

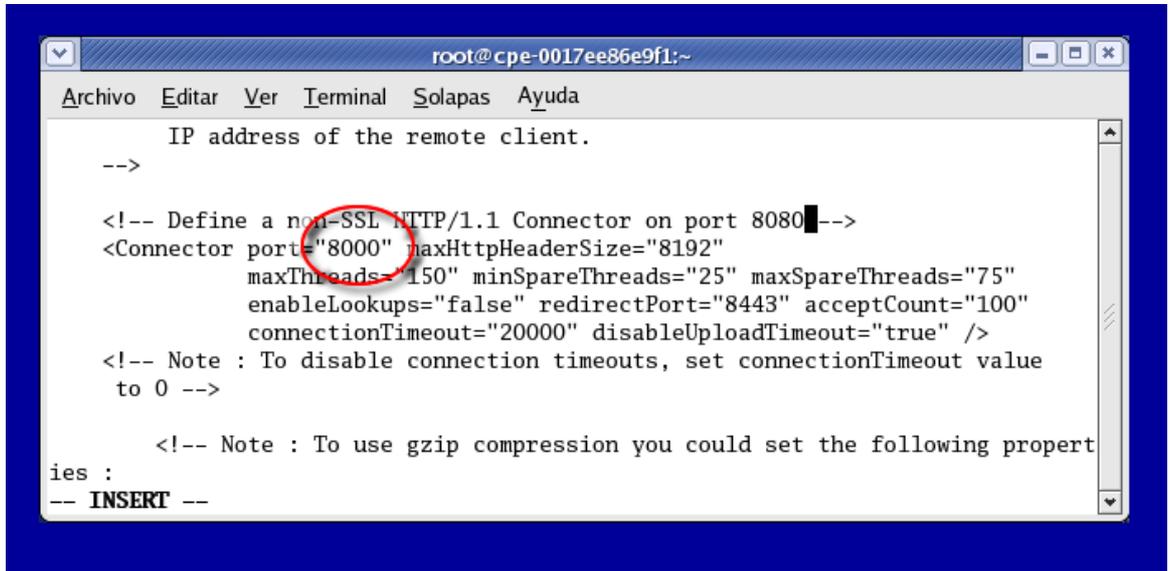
El siguiente gráfico ilustra la manera en la que se ejecuta el servicio tomcat.



```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@localhost ~]# service tomcat restart  
  
Restarting Tomcat  
  
Using CATALINA_BASE:   /usr/share/apache-tomcat-5.5.20  
Using CATALINA_HOME:   /usr/share/apache-tomcat-5.5.20  
Using CATALINA_TMPDIR: /usr/share/apache-tomcat-5.5.20/temp  
Using JRE_HOME:        /usr/java/jdk1.6.0/  
Using CATALINA_BASE:   /usr/share/apache-tomcat-5.5.20  
Using CATALINA_HOME:   /usr/share/apache-tomcat-5.5.20  
Using CATALINA_TMPDIR: /usr/share/apache-tomcat-5.5.20/temp  
Using JRE_HOME:        /usr/java/jdk1.6.0/  
[root@localhost ~]# service tomcat status  
  
Consulting Tomcat  
  
Tomcat is running 1 instance...  
[root@localhost ~]# service tomcat status
```

El puerto de conexión que utiliza Tomcat por defecto es el puerto 8080, pero este puede estar bloqueado en algunos servidores para impedir el uso del Messenger y otros programas que utilizan el mismo, para dar solución a este problema se

recomienda cambiarlo. En este caso, se escogió el 8000. Esta acción se la realiza en el archivo /conf/server.xml de la siguiente manera.



```
root@cpe-0017ee86e9f1:~
Archivo Editar Ver Terminal Solapas Ayuda
IP address of the remote client.
-->
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
<Connector port="8000" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true" />
<!-- Note : To disable connection timeouts, set connectionTimeout value
to 0 -->

<!-- Note : To use gzip compression you could set the following properties :
-- INSERT --
```

5.3 Manager de Tomcat

El manager es una aplicación para la administración del servidor Tomcat, que usa una interfaz web.

Se puede acceder al Manager de Tomcat luego de crear un usuario, con los accesos necesarios. Para crear un usuario se deberá modificar el archivo de configuración de usuarios de Tomcat “tomcat-users.xml”, este archivo se encuentra bajo la ruta:

`$CATALINA_HOME/conf/tomcat-users.xml`

La variable `$CATALINA_HOME` representa la ruta de ubicación del Apache Tomcat `/usr/share/apache-tomcat-5.5.20`

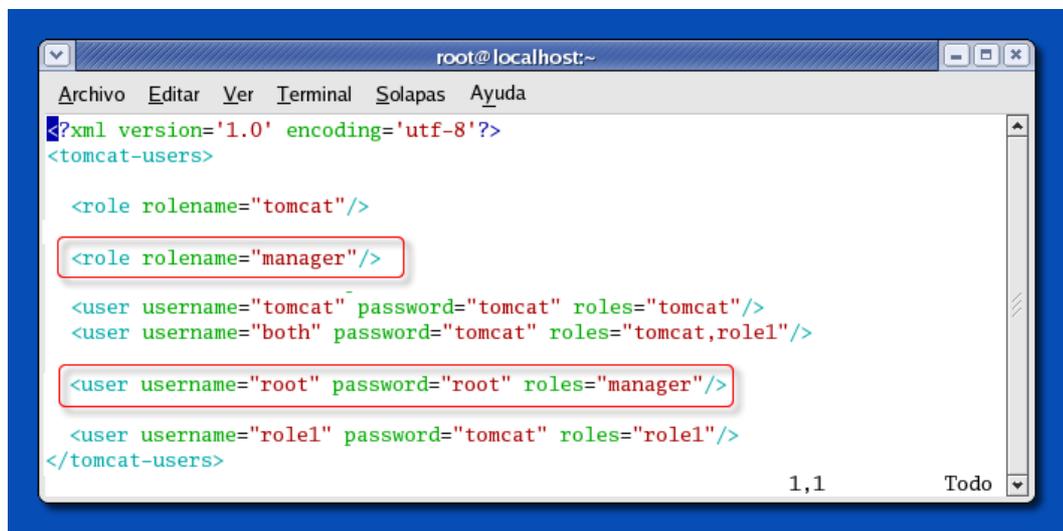
Al archivo `tomcat-users.xml` se le debe agregar las siguientes líneas:

```
<role rolename="manager"/>
```

```
<user username="root" password="password" roles="manager"/>
```

El documento que se ha modificado es un documento con extensión xml, al ser un documento de este tipo se debe respetar el orden de las etiquetas, es decir cuando se agreguen las líneas de <role>, se lo debe realizar debajo de las ya existentes, así mismo se insertarán las líneas de <user>. Con respecto al username se puede colocar el nombre que se desee, no es necesario que el nombre sea “root”, la condición de administrador es definida en el atributo ”role”.

En el siguiente gráfico se puede visualizar como quedará el archivo tomcat-users.xml luego de realizar los cambios:



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>

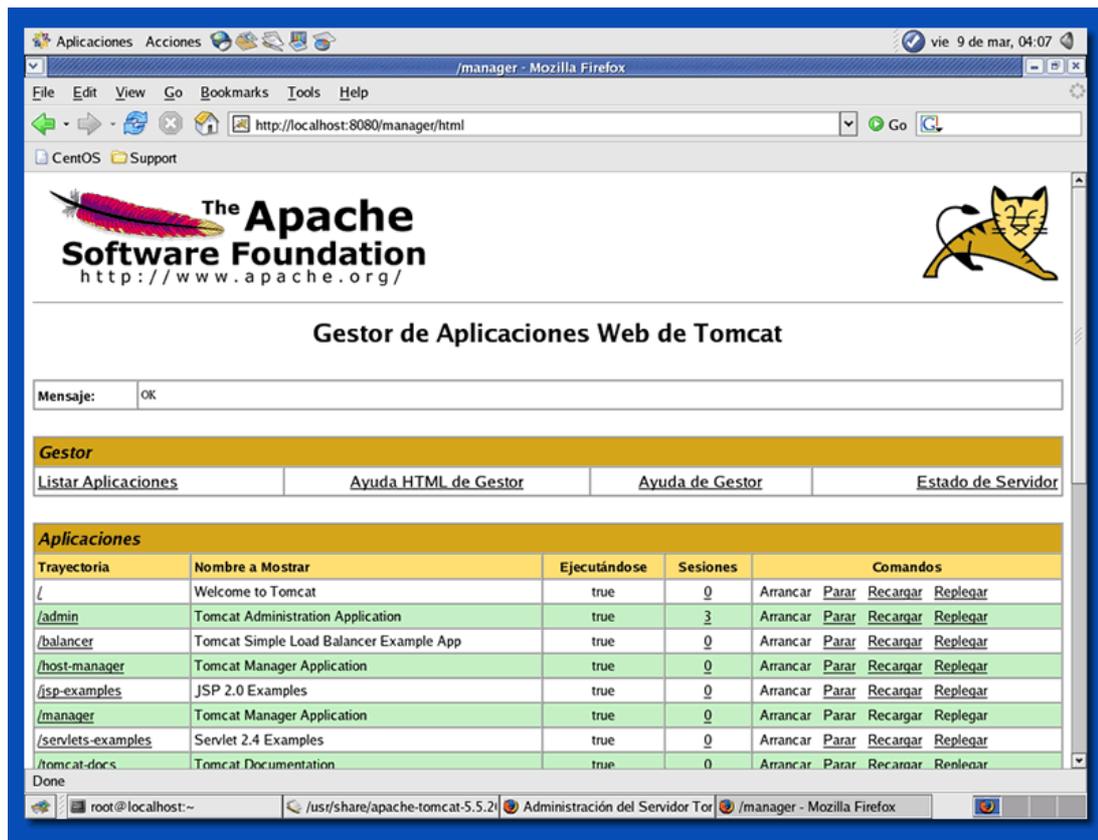
  <role rolename="tomcat"/>
  <role rolename="manager"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="root" password="root" roles="manager"/>
  <user username="role1" password="tomcat" roles="role1"/>
</tomcat-users>
1,1 Todo
```

Una vez añadido el usuario, se deberá reiniciar el servidor Tomcat, esto se lo realiza con las instrucciones de shutdown.sh y startup.sh.

Una vez que el servidor haya sido reiniciado, desde el navegador se deberá acceder a la siguiente dirección

<http://localhost:8080/manager/html>

Se pide ingresar los datos de username y password, del usuario recién creado, y aparecerá en el navegador la interfaz del manager que es la siguiente:



La interfaz del Manager o Gestor de Aplicaciones Web de Tomcat consta de 5 partes:

Mensaje: En esta parte se mostrará el resultado de las órdenes que se le den al manager de Tomcat, estos resultados pueden ser OK o FAILED.

Gestor: Dentro de esta parte, se tiene 4 opciones. La primera opción, recarga la lista de aplicaciones instaladas actualmente, las dos opciones siguientes, permiten el acceso a la documentación del manager y la última opción permite verificar el estado del motor de Tomcat.

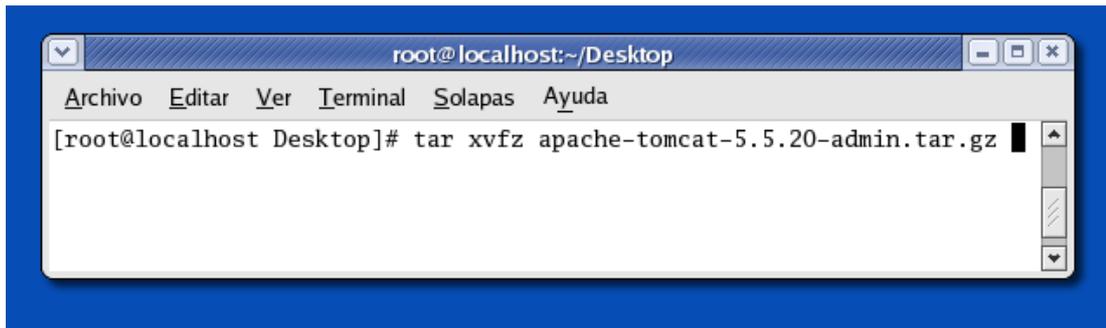
Aplicaciones: Dentro de esta opción se listan las aplicaciones web que está ejecutando el servidor Tomcat. Se encuentran los comandos `parar` `arrancar`, `parar`, `recargar` y `borrar`.

Desplegar: Desde esta opción se podrá subir aplicaciones directamente al servidor.

Información de Servidor: Contiene información sobre el servidor Tomcat.

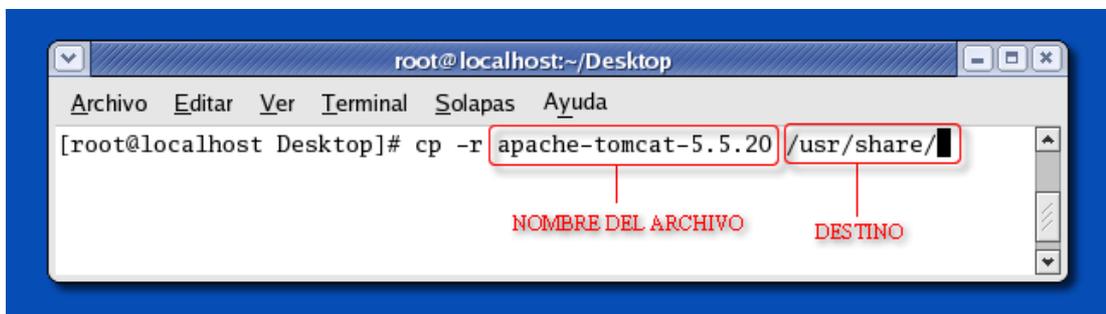
5.4 Administración del Servidor Tomcat

Otra herramienta principal es el Administrador de Tomcat, para lo cual se deberá descargar el archivo “apache-tomcat-5.5.20-admin.tar” desde la dirección <http://jakarta.apache.org/tomcat>, luego de que el archivo ha sido descargado, se deberá descomprimirlo, utilizando el comando “tar”, como se indica a continuación:



```
root@localhost:~/Desktop
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost Desktop]# tar xvfz apache-tomcat-5.5.20-admin.tar.gz
```

El siguiente paso a realizar es copiar la carpeta descomprimida en la ubicación /usr/share/. Esto se lo realiza de la siguiente manera:



```
root@localhost:~/Desktop
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost Desktop]# cp -r apache-tomcat-5.5.20 /usr/share/
```

NOMBRE DEL ARCHIVO DESTINO

Una vez que se hayan realizado todos los pasos indicados anteriormente, se podrá acceder al Administrador del Servidor Tomcat, para esto también es necesario, modificar el archivo tomcat-users.xml, agregando las siguientes líneas:

```
root@localhost:~  
Archivo  Editar  Ver  Terminal  Solapas  Ayuda  
<?xml version='1.0' encoding='utf-8'?>  
<tomcat-users>  
  <role rolename="manager" />  
  <role rolename="tomcat" />  
  <role rolename="admin" />  
  <role rolename="role1" />  
  <user username="admin" password="admin" roles="admin" />  
  <user username="tomcat" password="tomcat" roles="tomcat" />  
  <user username="both" password="tomcat" roles="tomcat,role1" />  
  <user username="root" password="root" roles="manager" />  
  <user username="role1" password="tomcat" roles="role1" />  
</tomcat-users>
```

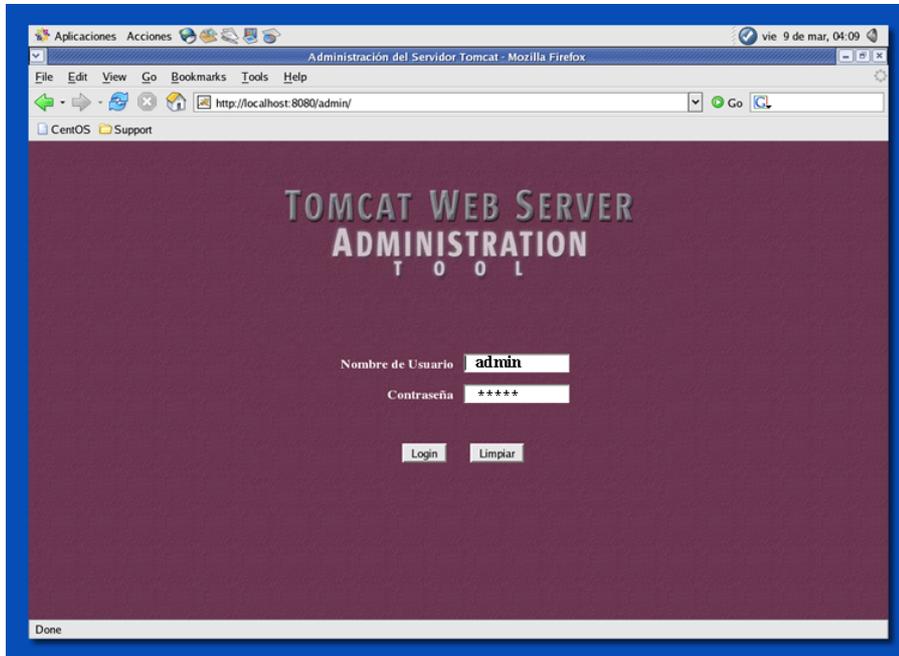
Luego de la modificación del archivo tomcat-users.xml, reiniciar el servidor Tomcat, utilizando las instrucciones de parada “shutdown.sh” y la instrucción de arrancar el servicio “startup.sh”.

```
root@localhost:~  
Archivo  Editar  Ver  Terminal  Solapas  Ayuda  
[root@localhost ~]# $CATALINA_HOME/bin/shutdown.sh → INSTRUCCION  
Using CATALINA_BASE:  /usr/share/apache-tomcat-5.5.20  PARA PARAR O  
Using CATALINA_HOME:  /usr/share/apache-tomcat-5.5.20  APAGAR  
Using CATALINA_TMPDIR: /usr/share/apache-tomcat-5.5.20/temp TOMCAT  
Using JRE_HOME:       /usr/java/jdk1.6.0  
  
[root@localhost ~]# $CATALINA_HOME/bin/startup.sh → INSTRUCCION  
Using CATALINA_BASE:  /usr/share/apache-tomcat-5.5.20  PARA REINICIAR  
Using CATALINA_HOME:  /usr/share/apache-tomcat-5.5.20  TOMCAT  
Using CATALINA_TMPDIR: /usr/share/apache-tomcat-5.5.20/temp  
Using JRE_HOME:       /usr/java/jdk1.6.0  
[root@localhost ~]#
```

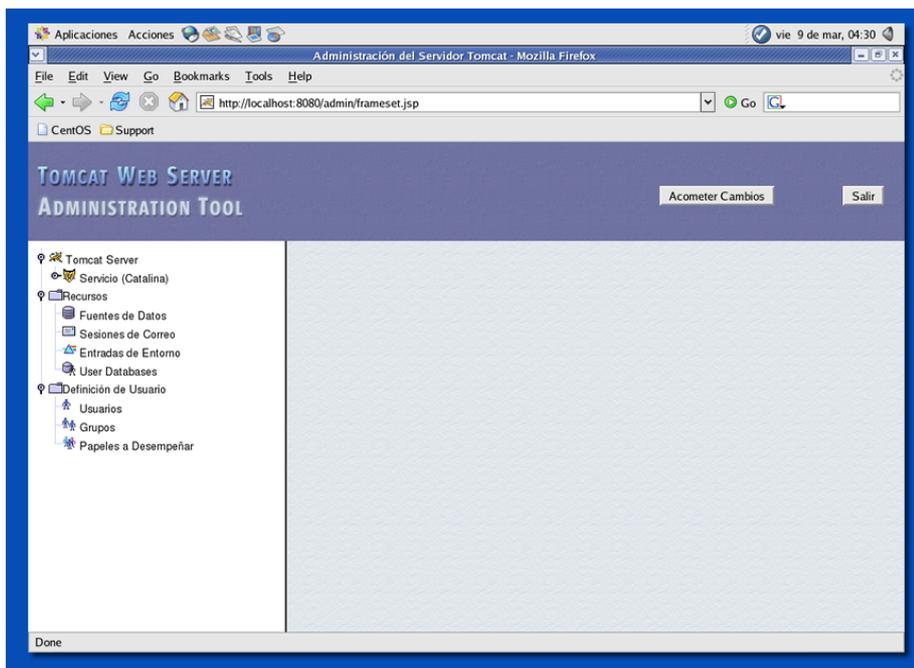
Una vez que se han realizado las instrucciones y el servidor Tomcat haya reiniciado, se podrá verificar que todo lo realizado anteriormente este correcto, esto se lo puede hacer accediendo al navegador web y colocando la siguiente dirección:

<http://localhost:8080/admin/>

La siguiente pantalla se visualizará en el navegador, allí serán ingresados los datos de usuario y password del administrador, dichos datos se ingresarán en una pantalla similar a la que se muestra en el siguiente gráfico.



Si el nombre de usuario y la contraseña ingresadas son correctos, se visualizará sin problemas la pantalla del Administrador del Servidor Tomcat.



5.5 Creación de un Servlet

Como se mencionó en el Capítulo II, los Servlets son programas escritos en lenguaje Java, que se encuentran residentes en la memoria de la PC, que se ejecutan dentro de un contenedor de Servlets, también corren o se ejecutan dentro de un servidor de aplicaciones.

Dicho de otro modo, un Servlet es un programa que se ejecuta en un servidor y al estar residente en la memoria, pueden rápidamente responder a una petición.

A continuación se procederá a crear un Servlet de ejemplo, como primer paso, se creará un archivo PruebaServlet.java el cual contendrá el siguiente código.

```
import java.io.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class PruebaServlet extends HttpServlet
{
public void doGet (HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException
    {
        PrintWriter out = res.getWriter();
        out.println ("Prueba de creacion de un Servlet");
        out.close ();
    }
}
```

Es recomendable para todas las aplicaciones, que se desarrollen en un directorio ajeno al de Tomcat. También es importante montar la estructura de directorios propia de un Servlet. Dicha estructura es la siguiente:

```
> Nombre de la aplicación
> META-INF (opcional)
> WEB-INF (Se incluye el archivo web.xml)
> classes (Se guardan los archivos compilados)
> src (opcional)
> lib (Opcional)
```

Una vez que el código este grabado, se deberá compilarlo, para esto bajo el directorio /WEB-INF/classes se ejecutan los siguientes comandos:

```
export PATH=/usr/java/jdk1.6.0/bin:$PATH
export CLASSPATH=$CATALINA_HOME/common/lib/servlet.jar:.
```

Luego de haber ejecutado los comandos anteriores se debe compilar el programa, con la instrucción:

```
Javac PruebaServlet.java
```

Se puede verificar si el programa compiló con éxito, listando el directorio donde se encuentra el archivo con extensión java ya que allí deberá aparecer el mismo archivo con extensión class.

El archivo PruebaServlet.class se deberá copiar al directorio classes. Se puede copiar el archivo PruebaServlet.java al directorio src, esto es opcional. Adicional a esto, se creará el archivo web.xml el cuál contendrá lo siguiente:

```
<web-app>
  <servlet>
    <servlet-name>prueba</servlet-name>
    <servlet-class>PruebaServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>invoker</servlet-name>
    <url-pattern>/servlet/*</servlet-class>
  </servlet-mapping>
```

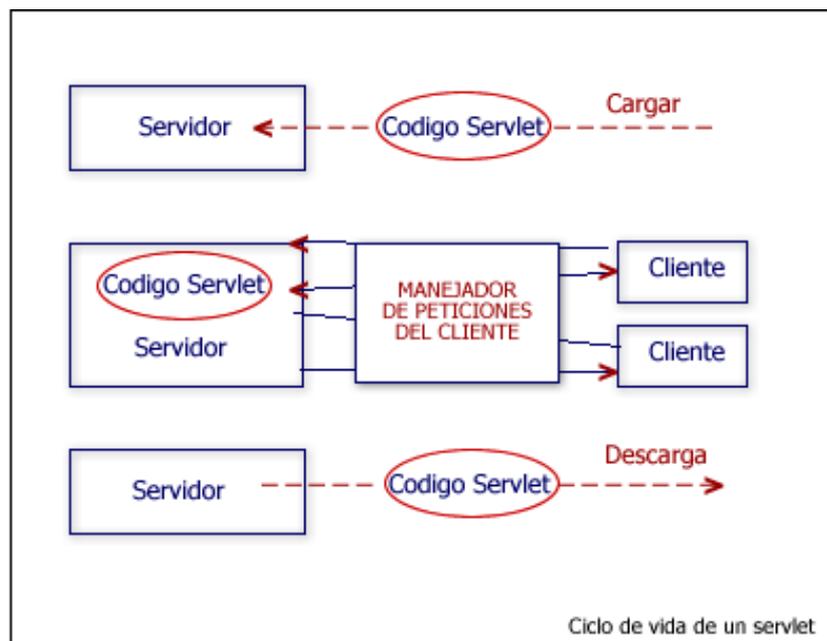
</web-app>

En servlet-name se indicara el nombre de la aplicación, esta deberá coincidir con el directorio donde se encuentra guardada. En Servlet-class se indica la clase principal de la aplicación y en url-pattern la dirección desde la cual se accederá al Servlet.

Para comprobar que todo esta correcto se ingresa en el explorador la dirección <http://localhost/prueba/PruebaServlet>, si todo esta correcto se deberá observar en el explorador el mensaje de Prueba de creación de un Servlet.

5.5.1 Ciclo de vida de un Servlet

- Cada Servlet tiene el mismo ciclo de vida.
- Un servidor carga e inicializa el Servlet.
- El servidor elimina el Servlet.



Cuando un servidor carga un Servlet, ejecuta el método “init” del Servlet. Esta inicialización se completa antes de manejar las peticiones del cliente y antes de que el Servlet sea destruido. El servidor llama una vez al método “init” y no lo llamará de nuevo a menos que vuelva a recargar el Servlet. El servidor no puede recargar un Servlet sin primero haber destruido el mismo, llamando al método “destroy”.

CAPITULO 6

CONFIGURACIÓN DE MYSQL

6.1 Pasos para la Instalación de MYSQL

Ya que el presente proyecto, está realizado en la versión CentOS 4.1, la instalación de un paquete adicional de MySQL no fue necesaria, ya que dicho Sistema Operativo, incorpora la versión MySQL 4.1.10a.

En el caso de requerir alguna versión de MySQL diferente a la que incorpore el sistema Operativo, podrá ingresar a la dirección www.mysql.com y descargar la versión necesaria.

Una vez realizado el enlace simbólico, se deberá cambiar de directorio y ejecutar el script que establece los servicios por defecto para los usuarios en el servidor de MySQL.

```
/usr/bin/mysql_install_db
```

Cuando se han realizado los pasos anteriores, se procederá a iniciar el servicio de MySQL

```
# service mysqld start
```

Si se desea que el servidor de MySQL corra en la máquina como demonio, es decir que cada vez que la máquina inicie, el servidor esté corriendo, se deberá ingresar en la terminal de comandos la siguiente sentencia.

```
# chkconfig --level 345 mysqld on
```

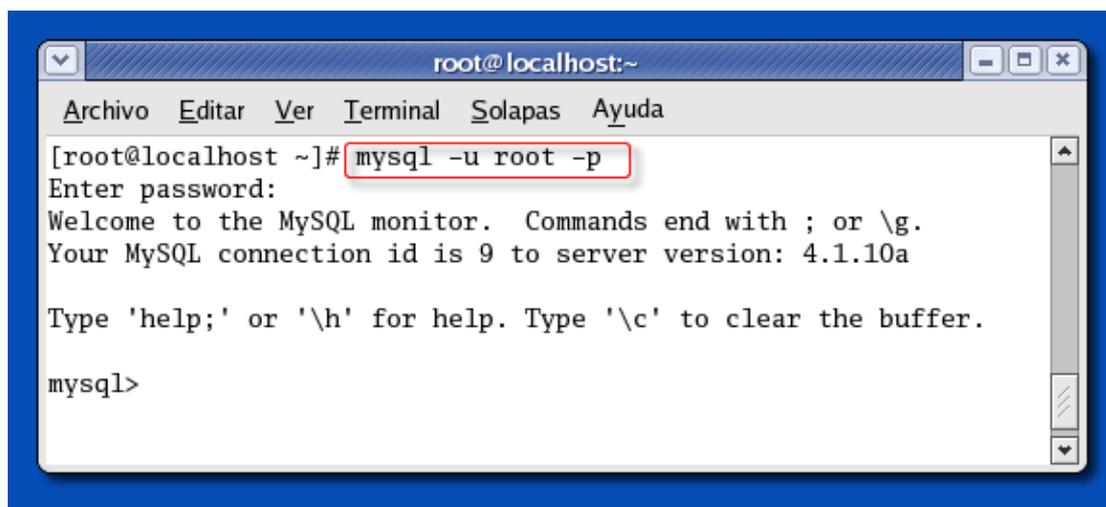
Después de iniciado el servicio, se debe establecer la contraseña para el usuario root utilizando el comando

```
#mysqladmin -u root password 'nueva contraseña'
```

Se deberá reemplazar ' nueva contraseña ' por la deseada, en este momento se tiene ya creado una cuenta del usuario root y su password.

El siguiente paso será ingresar a MySQL, para esto en la terminal de comandos ingresar la siguiente sentencia:

```
mysql -u root -p
```



Luego se deberá ingresar el password definido anteriormente, y se ingresará al shell de MySQL.

Después se realiza una petición para conocer las tablas ya creadas mediante la instalación del servidor.

```
mysql> use mysql
```

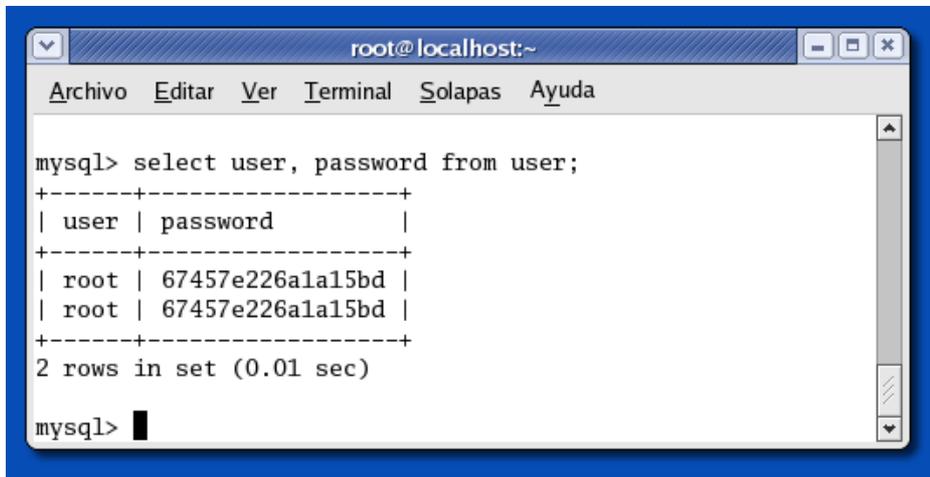
Y se puede hacer una petición para que se muestre las tablas ya creadas en MySQL

```
mysql> show tables;
```

Una vez que se ha ingresado el comando, se mostrarán los nombres de las tablas ya existentes, luego se podrá acceder a la tabla llamada “user”, que es la tabla que posee los usuarios y es donde está el root al que se desea acceder. Una vez hecho esto se realiza una selección de los campos user y password de la tabla user.

```
mysql> select user from user;
```

Mostrando como resultado lo siguiente:



The screenshot shows a terminal window titled "root@localhost:~" with a menu bar containing "Archivo", "Editar", "Ver", "Terminal", "Solapas", and "Ayuda". The terminal displays the following text:

```
mysql> select user, password from user;
+-----+-----+
| user | password          |
+-----+-----+
| root | 67457e226a1a15bd |
| root | 67457e226a1a15bd |
+-----+-----+
2 rows in set (0.01 sec)

mysql> █
```

Una vez realizadas todas las operaciones necesarias dentro de MySQL, se podrá abandonar MySQL con la instrucción “quit”.

La instrucción para reiniciar el servidor es la siguiente:

```
# service mysqld restart
```

Para crear una nueva base de datos se lo puede realizar utilizando el comando:

```
# mysqladmin -u root -p create "nueva_base_de_datos"
```

Luego se ingresará el password y estará creada la base de datos con el nombre asignado en “nueva_base_de_datos”.

Si por el contrario se desea eliminar una base de datos, el comando a utilizar seria el siguiente:

```
#mysqladmin -u root -p drop "base_de_datos_a_eliminar"
```

Es recomendable que el usuario root solo se encargue de las tareas administrativas y creación de nuevas bases de datos, para esto es conveniente delegar a los usuarios ordinarios el manejo de sus propias bases de datos.

Una vez que se ha generado una base de datos, se deberá también determinar con que usuario y desde que equipo de la red local se podrá tener acceso, así como los privilegios para modificarla.

Lo mas recomendable es que las asignaciones de acceso se realicen desde el mismo servidor (localhost), a menos que el desarrollo Web o aplicación se localice en otro de los equipos. Un ejemplo de cómo brindar estos permisos a un usuario se muestra a continuación:

```
mysql> GRANT select, insert, update, create, alter, delete, drop ON base_de_datos.*  
TO usuario@localhost. IDENTIFIED BY 'password_usuario'.
```

CAPÍTULO 7

INSTALACIÓN DE LA APLICACIÓN EJEMPLO

Para ejecutar la aplicación, se necesitará crear una base de datos, en este caso el nombre que se escogió es monografía. Para realizar esta acción desde la terminal de comandos poner la siguiente sentencia:

```
# mysqladmin -u root -p create monografía
```

Una vez creada la base de datos, el proyecto utilizará dos tablas: PRODUCTOS y USUARIOS que contendrán diferentes campos, para la creación de la base de datos y de los campos que la componen se utilizo la siguiente sentencias SQL.

```
DROP TABLE IF EXISTS PRODUCTOS;
DROP TABLE IF EXISTS USUARIOS;
CREATE TABLE PRODUCTOS
(
  PRO_CODIGO          INT          NOT NULL AUTO_INCREMENT,
  PRO_NOMBRE          VARCHAR(128) NOT NULL,
  PRO_CODIGO_BARRAS  VARCHAR(64)  NOT NULL,
  PRO_PRECIO          DECIMAL(13,4) NOT NULL,
  PRIMARY KEY (PRO_CODIGO)
)
COMMENT = "PRODUCTOS"
TYPE = INNODB;
```

```
ALTER TABLE PRODUCTOS
  MODIFY COLUMN PRO_CODIGO INT
  AUTO_INCREMENT COMMENT 'Codigo';
```

```
ALTER TABLE PRODUCTOS
  MODIFY COLUMN PRO_NOMBRE VARCHAR(128)
  COMMENT 'Nombre';
```

```
ALTER TABLE PRODUCTOS
  MODIFY COLUMN PRO_CODIGO_BARRAS VARCHAR(64)
  COMMENT 'Codigo barras';
```

```
ALTER TABLE PRODUCTOS
  MODIFY COLUMN PRO_PRECIO DECIMAL(13,4)
```

```

COMMENT 'Precio';

CREATE TABLE USUARIOS
(
  USU_CODIGO          INT          NOT NULL
  AUTO_INCREMENT,
  USU_NOMBRE          VARCHAR(128) NOT NULL,
  USU_PASSWORD        VARCHAR(128) NOT NULL,
  PRIMARY KEY (USU_CODIGO)
)
COMMENT = "USUARIOS"
TYPE = INNODB;

ALTER TABLE USUARIOS
  MODIFY COLUMN USU_CODIGO INT
  AUTO_INCREMENT COMMENT 'Codigo';

ALTER TABLE USUARIOS
  MODIFY COLUMN USU_NOMBRE VARCHAR(128)
  COMMENT 'Nombre';

```

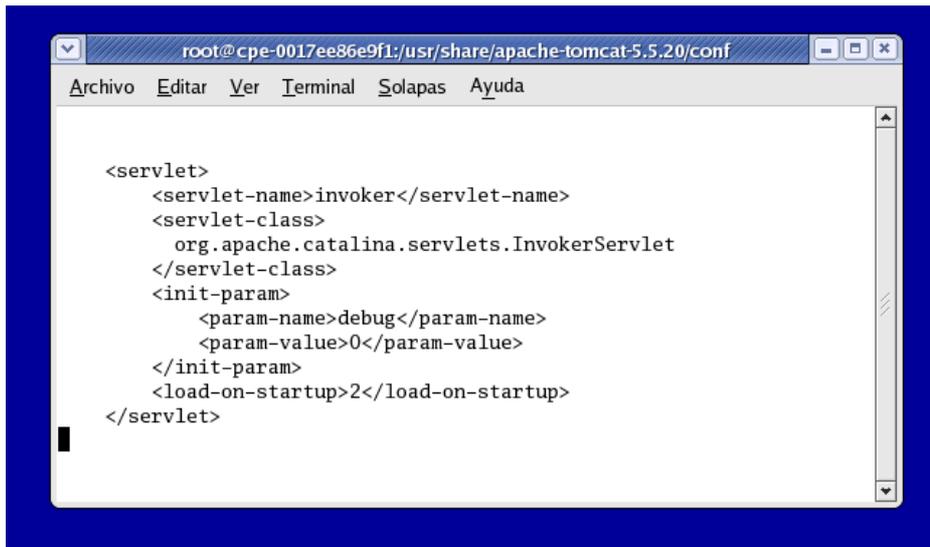
Para poder acceder a la base de datos, se deberá crear un usuario administrador, el usuario se crea con la siguiente sentencia SQL.

```

insert into USUARIOS (USU_NOMBRE, USU_PASSWORD) values ('usu',
'usu');

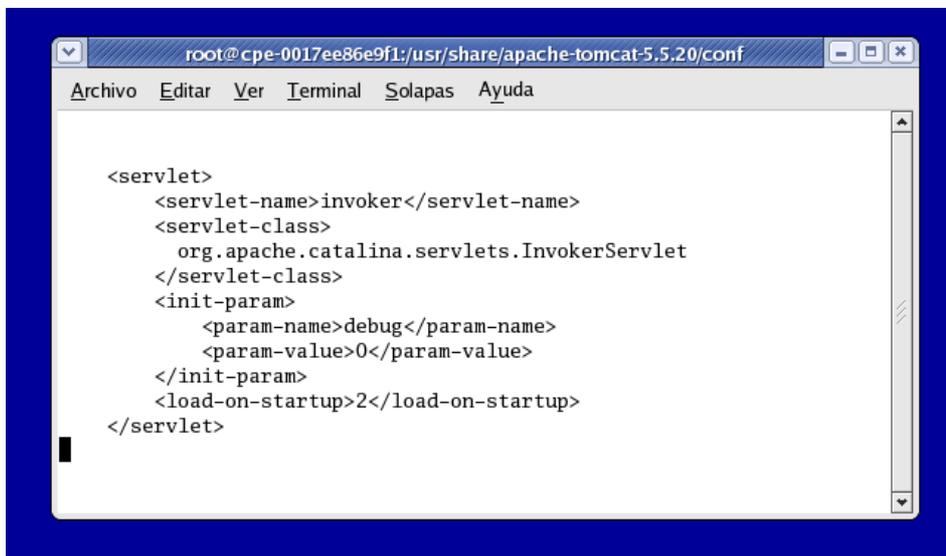
```

Antes de empezar con la ejecución de la aplicación, se sugiere modificar el archivo web.xml ubicado en /conf/ del que se habló en capítulos anteriores, descomentando las líneas que se indican en los dos cuatros a continuación. Esto se realiza para no detallar las opciones de los archivos web.xml de cada una de las aplicaciones creadas, ya que éstos tomarán las configuraciones generales del archivo web.xml global (/conf/web.xml).



```
root@cpe-0017ee86e9f1:/usr/share/apache-tomcat-5.5.20/conf
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

<servlet>
  <servlet-name>invoker</servlet-name>
  <servlet-class>
    org.apache.catalina.servlets.InvokerServlet
  </servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>2</load-on-startup>
</servlet>
```

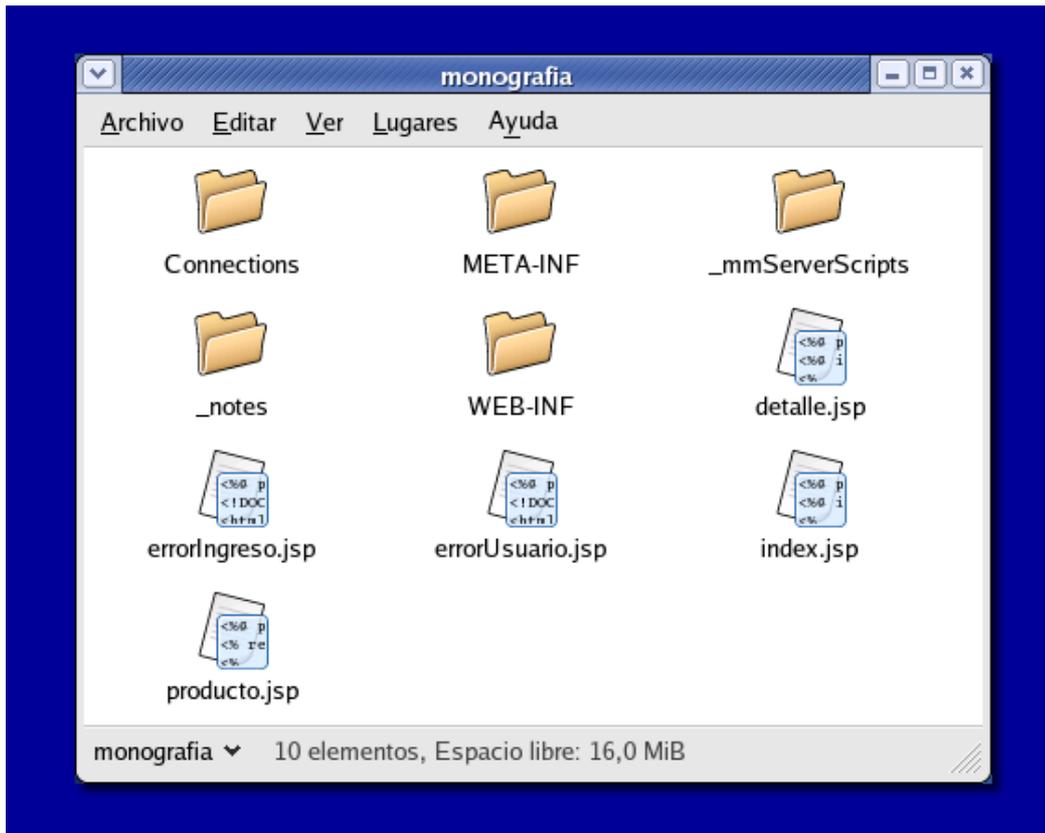


```
root@cpe-0017ee86e9f1:/usr/share/apache-tomcat-5.5.20/conf
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

<servlet>
  <servlet-name>invoker</servlet-name>
  <servlet-class>
    org.apache.catalina.servlets.InvokerServlet
  </servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>2</load-on-startup>
</servlet>
```

El siguiente paso es copiar la aplicación dentro de la estructura de CATALINA_HOME/webapps/ siguiendo la estructura de directorios de Tomcat.

A continuación se mostrará el gráfico con la estructura de directorios Tomcat.



Realizado lo anterior, los archivos web.xml correspondientes a las aplicaciones no necesitarán configuraciones específicas, únicamente se recomienda utilizar el elemento `<display-name>` para poder identificar a cada aplicación:

Un ejemplo del archivo web.xml correspondiente a una aplicación ejemplo, se muestra a continuación:

```
<display-name>Aplicacion de Ejemplo</display-name>

<session-config>
  <session-timeout>
    30|
  </session-timeout>
</session-config>
<welcome-file-list>
```

Una vez realizado esto, si todas las configuraciones están correctas se visualizará la aplicación consta dentro del Servidor.



The screenshot displays the Tomcat Manager web interface. At the top, there is a navigation bar with links for 'Gestor', 'Listar Aplicaciones', 'Ayuda HTML de Gestor', 'Ayuda de Gestor', and 'Edor'. Below this is a section titled 'Aplicaciones' containing a table with the following columns: 'Trayectoria', 'Nombre a Mostrar', 'Ejecutándose', 'Sesiones', and 'Comandos'. The table lists several applications, including 'monografía', which is circled in red. Below the table is a 'Desplegar' section with the text 'Desplegar directorio o archivo WAR localizado en servidor'.

Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar
/admin	Tomcat Administration Application	true	0	Arrancar Parar Recargar Replegar
/balancer	Tomcat Simple Load Balancer Example App	true	0	Arrancar Parar Recargar Replegar
/host-manager	Tomcat Manager Application	true	0	Arrancar Parar Recargar Replegar
/jsp-examples	JSP 2.0 Examples	true	0	Arrancar Parar Recargar Replegar
/manager	Tomcat Manager Application	true	0	Arrancar Parar Recargar Replegar
/monografía	Aplicacion de Ejemplo	true	1	Arrancar Parar Recargar Replegar
/servlet-examples	Servlet 2.4 Examples	true	0	Arrancar Parar Recargar Replegar
/tomcat-docs	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar
/webdav	Webdav Content Management	true	0	Arrancar Parar Recargar Replegar

CAPÍTULO 8

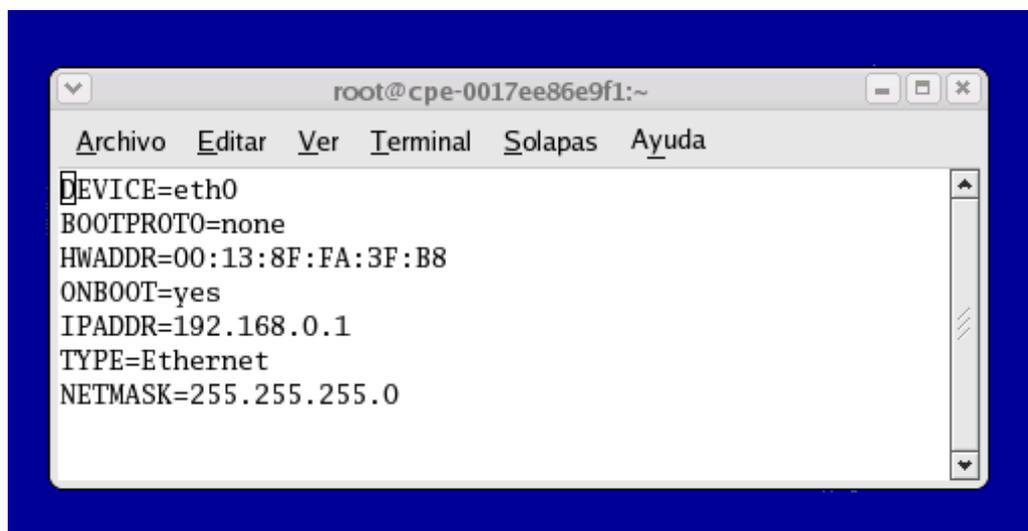
PRUEBAS DE COMUNICACIÓN CLIENTE-SERVIDOR

8.1 Configuración del Servidor

Para que un computador tenga conectividad Ip, y el cliente pueda acceder al servidor, se deberá realizar la configuración necesaria dentro de los mismos. En el servidor se configuraran los parámetros de red necesarios, para esto en la terminal de comandos, se deberá colocar lo siguiente:

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

Se edita el archivo, el cual deberá contener las líneas indicadas a continuación:

A screenshot of a terminal window with a blue background. The window title is "root@cpe-0017ee86e9f1:~". The menu bar includes "Archivo", "Editar", "Ver", "Terminal", "Solapas", and "Ayuda". The terminal content shows the following configuration lines:

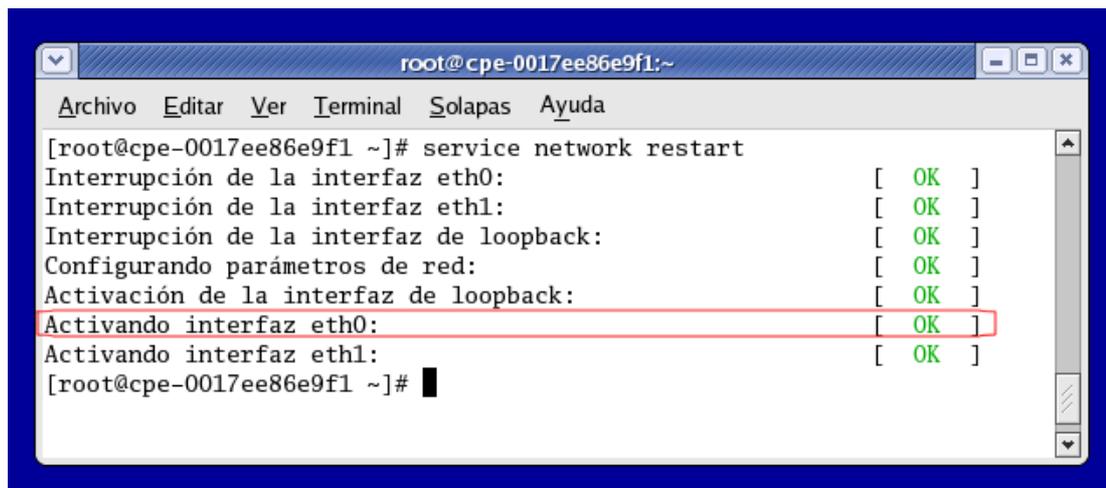
```
DEVICE=eth0
BOOTPROTO=none
HWADDR=00:13:8F:FA:3F:B8
ONBOOT=yes
IPADDR=192.168.0.1
TYPE=Ethernet
NETMASK=255.255.255.0
```

Existen varias opciones dentro el archivo ifcfg-eth0, estas opciones serán modificadas de acuerdo a las necesidades. En la casilla IPADDR se deberá indicar la dirección Ip del servidor, para 192.168.0.1 y se definirá una mascara de subred o NETMASK=255.255.255.0

Para que los cambios tengan efecto se deberá reiniciar el servicio de la configuración de red, utilizando el comando:

```
service network restart
```

Si todos los cambios se han realizado correctamente luego de reiniciar el servicio se mostrará una pantalla como la siguiente:



```
root@cpe-0017ee86e9f1:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@cpe-0017ee86e9f1 ~]# service network restart
Interrupción de la interfaz eth0: [ OK ]
Interrupción de la interfaz eth1: [ OK ]
Interrupción de la interfaz de loopback: [ OK ]
Configurando parámetros de red: [ OK ]
Activación de la interfaz de loopback: [ OK ]
Activando interfaz eth0: [ OK ]
Activando interfaz eth1: [ OK ]
[root@cpe-0017ee86e9f1 ~]#
```

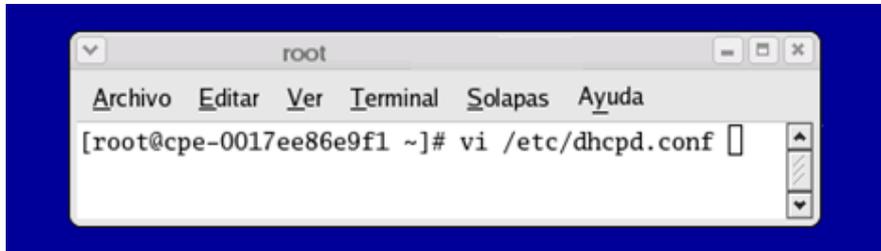
Para que la dirección Ip sea asignada de manera automática al cliente, es necesario configurar el servicio DHCP en el servidor, dicha configuración se la realiza modificando el archivo dhcpd.conf existente en el sistema, para esto se copiará el archivo dhcop.conf al directorio /etc/dhcdp.conf

A continuación se indica como copiar el archivo existente en otra ubicación:

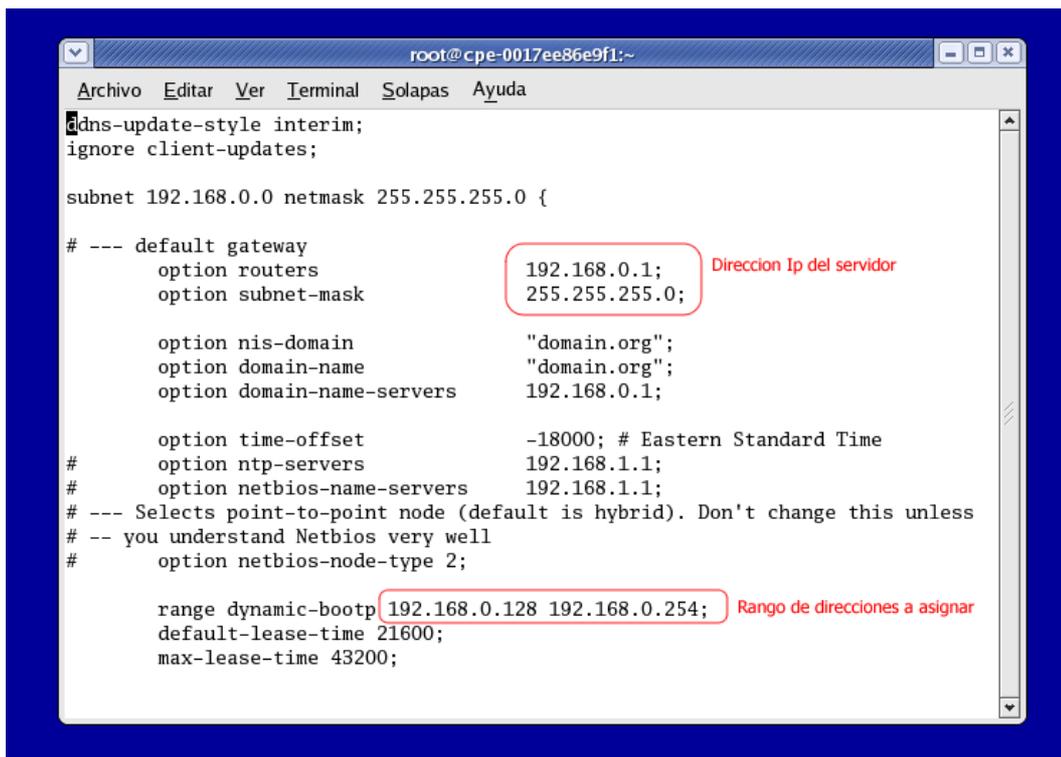


```
root@
Archivo Editar Ver Terminal Solapas Ayuda
[root]# cp /usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample /etc/dhcp.conf
```

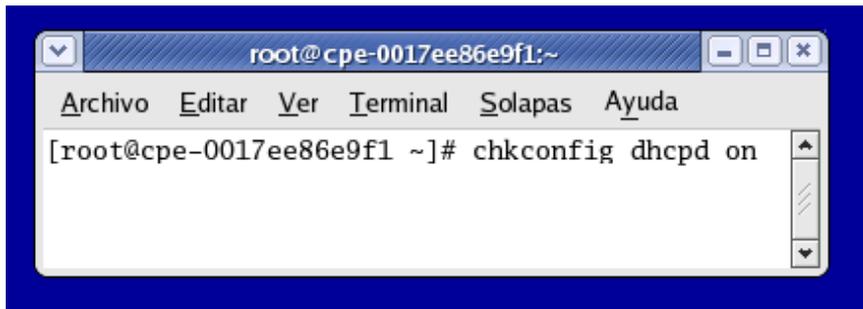
En este archivo se indica el rango de direcciones IP que el servidor asignará a las Pc's de la misma red, esto lo hace de manera automática. Para editar el archivo, se utiliza el comando vi.



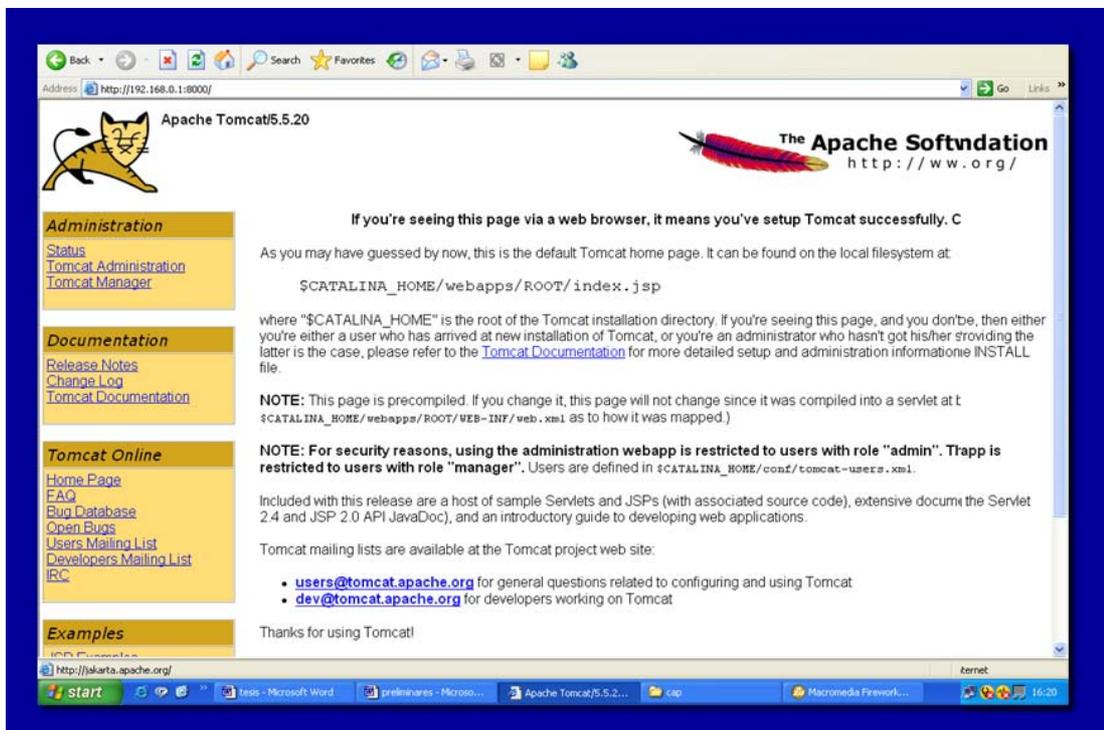
Para la asignación de las direcciones Ip se debe modificar las casillas señaladas en el siguiente gráfico, tomando en cuenta la dirección Ip del servidor, y el rango de direcciones IP a asignar.



Para que las configuraciones anteriormente mencionadas tengan efecto, se deberá reiniciar el servicio dhcp, esto se lo realiza con el comando. Esta instrucción es importante, hay que tener en cuenta que cada vez que el servidor inicie, se deberá arrancar el servicio dhcp.



Realizadas las configuraciones anteriormente detalladas, desde cualquier máquina de la red local, se podrá visualizar la página del servidor de aplicaciones Tomcat, accediendo a la siguiente dirección.

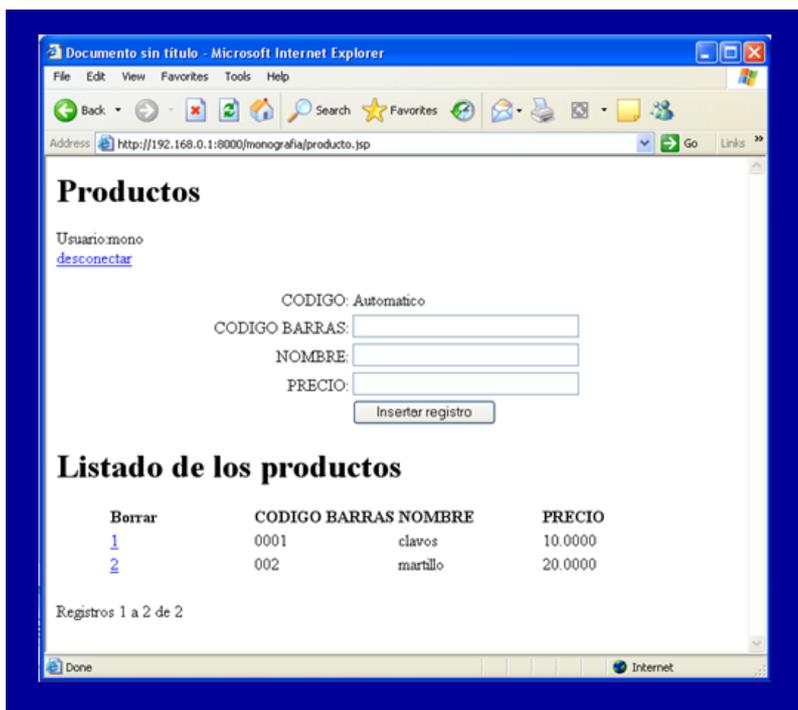


Esto demuestra que la configuración ha sido realizada correctamente, si se desea acceder a cualquier aplicación existente en el sistema, se deberá ingresar la dirección de la misma, la cual constara de la dirección Ip del servidor, el puerto al que se desea conectar y el nombre de la misma.

<http://192.168.0.1:8000/monografia>



En la pagina inicial de la aplicación, se deberá colocar el nombre de usuario y contraseña creados en la tabla USUARIO, si los datos son correctos se visualizara la pantalla de productos, en donde el usuario podra insertar nuevos registros o asi mismo listarlos.



Para esto la aplicación se conecta con la base de datos creada anteriormente, y se visualizara la información que contenga la misma. Se podr'an hacer modificaciones dentro de la base de datos, eliminando registros que esta contenga.



CONCLUSIONES

Luego de haber concluido con la realización de esta monografía, nos pudimos dar cuenta que en el mercado existen herramientas importantes, como Tomcat, que deberían ser mas utilizadas ya que son de enorme importancia en el momento de la administración de aplicaciones.

Cabe recalcar, que los objetivos trazados al inicio de este trabajo fueron alcanzados satisfactoriamente. La instalación y configuración del servidor Tomcat, se lo realizó sin ningún inconveniente, siguiendo todos los pasos descritos en esta monografía.

Así mismo se alcanzaron otros objetivos importantes como los de elaborar el manual de instalación de aplicaciones, y la instalación de la aplicación de prueba.

RECOMENDACIONES

Luego de realizar el presente proyecto, se recomienda no utilizar las versiones beta de ninguna de las herramientas que hemos usado ya que pueden presentar ciertos errores e incompatibilidades.

Se debe tomar en cuenta que las indicaciones que se han realizado en este manual sobre los procesos de configuración e instalación de las herramientas utilizadas solamente se aplican a las versiones aquí nombradas, cualquier otra versión de software puede presentar variantes en la instalación y configuración.

Se deberá tener especial cuidado al momento de escoger el puerto en el que queremos que Tomcat trabaje, ya que existen algunos puertos que son utilizados por otras aplicaciones, y otros que pueden estar bloqueados por un firewall tanto en el cliente como en el servidor.

Se recomienda utilizar la versión de MySQL que es proporcionada en la misma distribución del sistema operativo, ya que utilizar una versión diferente de la base de datos puede ocasionar algunas complicaciones.

BIBLIOGRAFÍA

- Tomcat: The Definitive Guide, Guía del Servidor de aplicaciones Tomcat, Jason Brittain, Ian F. Darwin.
- Professional Apache Tomcat 5, Libro electrónico Apache Tomcat, Vivek Chopra, Amit Bakore, Jon Eaves, Ben Galbraith, Sing Li, Chanoch Wiggers
- Professional Apache Tomcat 5, Libro electrónico Apache Tomcat, Vivek Chopra, Amit Bakore, Jon Eaves, Ben Galbraith, Sing Li, Chanoch Wiggers
- <http://www.programacion.net/java/tutorial/tomcatintro/1/>, Tutorial de Instalación y configuración de Tomcat, Programación en Castellano, S. L, Ciudad Real España.
- <http://www.mysql-hispano.org/page.php?id=25&pag=1>, Aplicaciones web con Tomcat y MySQL en Linux, MySQL Hispano.

ANEXOS

UNIVERSIDAD DEL AZUAY

**FACULTAD DE CIENCIAS DE LA
ADMINISTRACION**

ESCUELA DE INGENIERIA DE SISTEMAS

PROYECTO DE MONOGRAFIA

TEMA:

**“Instalación y Utilización del Servidor de Aplicaciones Tomcat de Apache sobre
Linux”**

AUTORES:

Durán Medina Maria Alexandra

Torres López Edén José

CUENCA-ECUADOR

2007

1. Título del Proyecto

“Instalación y Utilización del Servidor de Aplicaciones Tomcat de Apache sobre Linux”

2. Selección y Delimitación del Tema

Contenido:

La presente monografía detalla la instalación del Servidor de Aplicaciones Tomcat, sobre la plataforma LINUX, así como la utilización práctica de la misma, para lo cuál se crea una aplicación realizada en Lenguaje de Programación Java, utilizando un servidor de base de datos “MySQL”.

Clasificación:

Se escogió trabajar sobre LINUX, Tomcat de Apache y MySQL, por tratarse de Software de uso libre, facilitando así la obtención de los mismos y además son considerados como Software robusto y estable.

Espacio:

Para la investigación y desarrollo se utilizaran los laboratorios de la Universidad del Azuay y equipos particulares.

Tiempo:

El tiempo de duración del proyecto es de dos meses.

3. Resumen del Proyecto.

En la presente monografía se realizará la instalación del Servidor de Aplicaciones Tomcat, para lo cuál es necesario la previa instalación del sistema Operativo LINUX, realizando las configuraciones necesarias tanto del sistema Operativo como del Servidor de Aplicaciones para su correcto funcionamiento, luego de lo cuál se realizarán las debidas pruebas de funcionamiento con la ayuda de una aplicación desarrollada en Java que utiliza MySQL.

El proyecto comprenderá:

- Instalación y configuración del Sistema Operativo LINUX.
- Instalación y configuración del Servidor de Aplicaciones Tomcat.
- Instalación y configuración de MySQL
- Desarrollo de la aplicación en Java
- Pruebas de funcionamiento

Para el desarrollo de los puntos antes mencionados, se deberá realizar una investigación de los fundamentos teóricos y prácticos de los mismos que nos ayudarán con el buen funcionamiento del presente tema.

Además contaremos con dos PC, la una cumplirá la función de Servidor con el sistema Operativo Linux, y la otra PC, servirá de cliente para las pruebas de la aplicación desarrollada.

4. Contexto sobre la Tesis.

En la actualidad se ha incrementado de manera interesante la utilización del servidor de Aplicaciones “Tomcat” y del Sistema Operativo “LINUX” ya que estos son de uso libre, además Tomcat es compatible con las API mas recientes de Java y no es muy exigente en cuanto a recursos de hardware.

5. Impacto tecnológico.

La reducción de los costos en cuanto a compra de licencias y de equipos para el funcionamiento de servidores de pequeña y gran magnitud hace que se de preferencia a la utilización del software escogido en esta monografía.

6. Impacto Social.

Los beneficios económicos obtenidos por la preferencia del uso del software libre versus el software pagado, lo que constituye un aspecto de gran importancia

dentro de la economía de las empresas, y de la sociedad en general relacionada con esta área.

7. Objetivos

7.1 Objetivo General:

El Objetivo general es la instalación, configuración y utilización del servidor de aplicaciones Tomcat en un ambiente LINUX.

7.2 Objetivos Específicos:

- Elaborar un manual de Instalación de Tomcat
- Elaborar un manual de Instalación de Aplicaciones
- Instalar una aplicación de prueba en Java para Tomcat.

8. Teoría Referencial.

Para el adecuado desarrollo del proyecto, se utilizarán los conocimientos adquiridos en el Curso de Graduación, así como manuales e información adquirida en Internet.

- LINUX

Podríamos definir Linux como un sistema operativo multiusuario y multitarea basado en la filosofía de Unix. Es decir, puede haber varios usuarios utilizando un computador al mismo tiempo, y varios procesos ejecutándose a la vez. Actualmente es el sistema operativo que más plataformas soporta.

- TOMCAT

Tomcat es un contenedor de Servlets con un entorno JSP. Un contenedor de Servlets es un shell de ejecución que maneja e invoca Servlets por cuenta del usuario.

- MYSQL

Es un sistema de gestión de base de datos, multiusuario y multihilo, con más de seis millones de instalaciones. MySQL es desarrollado como software libre y

además puede ser adquirido por empresas que quieran incorporarlo en productos privados.

- JAVA

Java es un lenguaje de programación con el se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems y esta enfocado a cubrir las necesidades tecnológicas más importantes.

9. Plan Operativo

Introducción

Capítulo 1: Servidor de Aplicaciones Tomcat

- 1.1 Introducción a Tomcat
- 1.2 Características de Tomcat
- 1.3 Estructura de Directorios de Tomcat
- 1.4 Scripts de Tomcat
- 1.5 Ficheros de Configuración de Tomcat
 - 1.5.1 Fichero server.xml
 - 1.5.2 Fichero web.xml

Capítulo 2: Descripción de MySQL

- 2.1 Introducción a MySQL
- 2.2 Características de MySQL

Capítulo 3: Descripción del Lenguaje de Programación Java

- 3.1 Introducción a Java
- 3.2 Características del Lenguaje de Programación Java

Capítulo 4: Requisitos Previos a la Instalación de Tomcat

Capítulo 5: Instalación y Configuración de Tomcat

- 5.1 Primeros Pasos
- 5.2 Instalación de Tomcat
- 5.3 Manager de Tomcat
- 5.4 Administración del Servidor Tomcat
- 5.5 Creación de un Servlet

5.5.1 El ciclo de vida de un Servlet

Capítulo 6: Instalación de MySQL

6.1 Pasos para la Instalación de MySQL

Capítulo 7: Instalación de la aplicación ejemplo

Capítulo 8: Pruebas de Comunicación Cliente-Servidor

8.1 Configuración del Servidor

10. Recursos Humanos

1 Director del Proyecto

2 Investigadores / Estudiantes Alexandra Durán Medina, Edén Torres López.

11. Recursos Materiales

Hardware:

2 PC`s 386 o mayor

Tarjeta de red para la comunicación de al menos 2PC`s

Cable de red

Software:

Sistema Operativo LINUX basados en el Kernel de Red Hat 2.3.0.7 en adelante.

Sistema Operativo Windows

Tomcat

Java

MYSQL

12. Bibliografía.

Para la realización de este proyecto es primordial la investigación en Internet, por ser una fuente actualizada y de uso común en la actualidad, algunas de las páginas consultadas son:

<http://tomcat.apache.org/> - Sitio Oficial de Apache-Tomcat- 2007-01-04

<http://sunsite.dcaa.unam.mx> – Sun Microsystems – 2007-01-05

<http://www.linux-es.org> - El Rincón de Linux – 2007-01-05

<http://www.gra2.com/article.php/20050721165958576> - Blog sobre tecnología y Opinión de Barcelona-España -2007-01-05

13. Cronograma de Actividades

	SEM 1	SEM 2	SEM 3	SEM 4	SEM 5	SEM 6
Servidor de Aplicaciones Tomcat	■					
Descripcion de MySQL	■					
Descripcion de Lenguaje Java		■				
Instalacion y Configuracion de Tomcat		■	■			
Instalacion de MySQL			■			
Instalacion de Java			■			
Desarrollo de Aplicacion Ejemplo				■	■	
Pruebas Cliente Servidor						■
Conclusiones						■
Recomendaciones						■