



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Práctica 0: Configuración del entorno de
desarrollo.

Unidad de aprendizaje: Web Application Development

Grupo: 3CM6

Alumnos(a):

Ramos Diaz Enrique

Profesor(a):

Montes Casiano Hermes Francisco

19 de febrero 2019

Índice

1	Introducción	2
2	Desarrollo	2
2.1	Instalación de Java JDK 8	2
2.2	Instalación de Apache Tomcat 8	3
2.3	Crear un usuario Tomcat	4
2.4	Arranque automático de Tomcat	6
2.5	Configuración de Apache y Apache Tomcat	7
2.5.1	Instalación del servidor Apache	9
2.6	Usuario para Tomcat Manager	9
2.7	Cerrar el puerto 8080	11
2.8	Instalar el IDE Eclipse for Java EE Developers	12
2.9	Instalar Jetty	13
3	Pruebas	15
3.1	Pantalla del navegador al intentar acceder a localhost:8080	15
3.2	Pantalla del navegador al acceder a localhost	15
3.3	Versión de Java instalada	16
3.4	Eclipse y Jetty ejecutándose	16
3.5	Ejemplos de servlets en clase, ejecutándose desde Jetty	16
3.6	Servlet Hola Mundo	17
4	Conclusiones	17
5	Bibliografía	17

1. Introducción

Para el curso de Web Application Development es necesario configurar el entorno de desarrollo en nuestro equipo. Se va a trabajar con el sistema operativo Linux, además se utilizarán los paquetes de desarrollo de Java en su versión 8, así como un servidor web, siendo el elegido Tomcat 8.

Debemos establecer los directorios HOME tanto de Tomcat como de Java, crear usuarios, asignar permisos, conectar los servidores Apache y Tomcat entre sí, preparar el servidor para un balanceador de cargas, cerrar puertos y configurar el arranque automático.

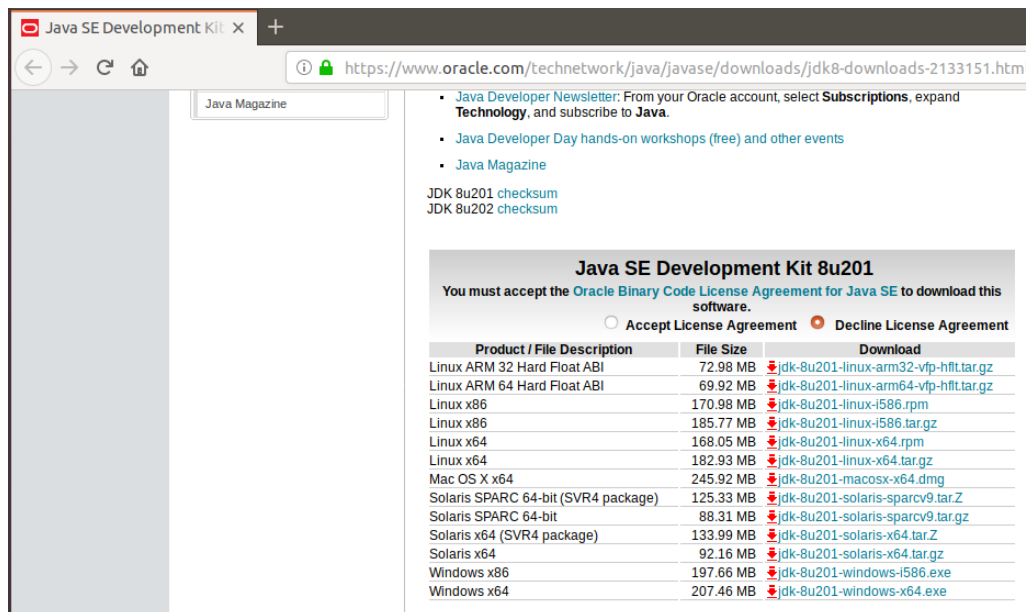
Finalmente, nos auxiliaremos de un IDE para trabajar de una forma más fácil y dinámica, el cual será Eclipse en su paquete para Java EE Web Developers, y un plugin para la ejecución en tiempo real de nuestras aplicaciones llamado Jetty.

2. Desarrollo

Se utilizó la distribución de Ubuntu 16.04 LTS para ésta configuración.

2.1. Instalación de Java JDK 8

1. Debemos ir a la página: <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> y descargar la versión más reciente de JDK 8.



2. Abrimos el directorio en donde se guardó el archivo tar.gz, abrimos una terminal como super usuario con:

```
$ sudo su
```

Y lo descomprimos con el comando:

```
$ tar -xvf jdk-8u201-linux-x64.tar.gz
```

3. Luego creamos un nuevo directorio, que será el HOME de Java en nuestro equipo. Copiamos el contenido extraído en la carpeta a la nueva creada:

```
$ mkdir /usr/local/java
$ mv jdk1.8_202 /usr/local/java
```

4. Ejecutamos los siguientes comandos para indicarle a Ubuntu en donde está localizado Java:

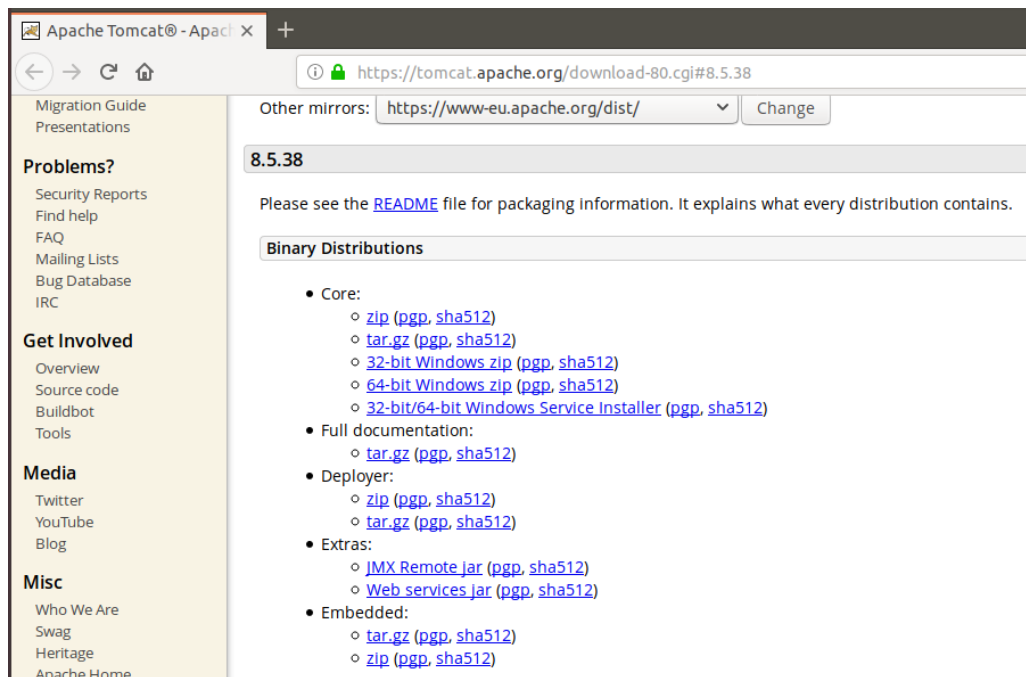
```
$ update-alternatives --install /usr/bin/java java /usr/local/
java/jdk1.8_202/bin/java 1065
$ update-alternatives --install /usr/bin/javac javac /usr/local/
java/jdk1.8_202/bin/javac 1065
$ update-alternatives --install /usr/bin/jar jar /usr/local/
java/jdk1.8_202/bin/jar 1065
$ update-alternatives --install /usr/bin/javaws javaws /usr/local/
java/jdk1.8_202/bin/javaws 1065
```

5. Finalmente, seleccionamos la versión de Java con la que vamos a trabajar:

```
$ update-alternatives --config java
```

2.2. Instalación de Apache Tomcat 8

1. Descargar la versión más reciente de Apache Tomcat 8, según nuestro sistema operativo y arquitectura, desde el siguiente link: <https://tomcat.apache.org/download-80.cgi>



2. Abrimos el directorio en donde se guardó el archivo tar.gz, abrimos una terminal y lo descomprimos:

```
$ tar -xvzf apache-tomcat-8.5.37.tar.gz
```

3. Una vez finalizada la extracción, procedemos a borrar el archivo tar.gz, pues ya no lo utilizaremos:

```
$ rm apache-tomcat-8.5.37.tar.gz
```

4. Accedemos como superusuario a la terminal y copiamos la ruta en donde se extrajeron todos los archivos de configuración y directorios de Tomcat a la siguiente a continuación:

```
# mv apache-tomcat-8.5.37/* /opt/tomcat/apache-tomcat-8.5.37/
```

Y borramos la carpeta extraída original:

```
# rm -rf apache-tomcat-8.5.37
```

5. Por último, es necesario agregar e indicar CATALINA_HOME como variable del sistema. Abrimos el archivo bash.bashrc desde la terminal como superusuario con el comando:

```
# nano /etc/bash.bashrc
```

Y agregamos ésta línea al final del archivo:

```
export CATALINA_HOME=/opt/tomcat/apache-tomcat-8.5.37/
```

```
GNU nano 2.9.3 /etc/bash.bashrc

# check because c-n-f could've been removed in the meantime
if [ -x /usr/lib/command-not-found ]; then
    /usr/lib/command-not-found -- "$1"
    return $?
elif [ -x /usr/share/command-not-found/command-not-found ]; then
    /usr/share/command-not-found/command-not-found -- "$1"
    return $?
else
    printf "%s: command not found\n" "$1" >&2
    return 127
fi

}

fi

export CATALINA_HOME=/opt/tomcat/apache-tomcat-8.5.37
export JAVA_HOME=/usr/lib/jvm/java-8-oracle

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

6. Guardamos los cambios y cerramos el archivo.

2.3. Crear un usuario Tomcat

1. Abrimos una terminal como superusuario y creamos un grupo llamado tomcat:

```
# groupadd tomcat
```

2. Creamos un usuario llamado tomcat, lo agregamos al grupo creado previamente y establecemos el directorio de CATALINA_HOME como home de éste:

```
# useradd -g tomcat -d /opt/tomcat/apache-tomcat-8.5.37/ tomcat
```

3. Agregamos al usuario tomcat al grupo www-data:

```
# usermod -G www-data tomcat
```

4. Establecemos una contraseña para el usuario tomcat:

```
# passwd tomcat
```



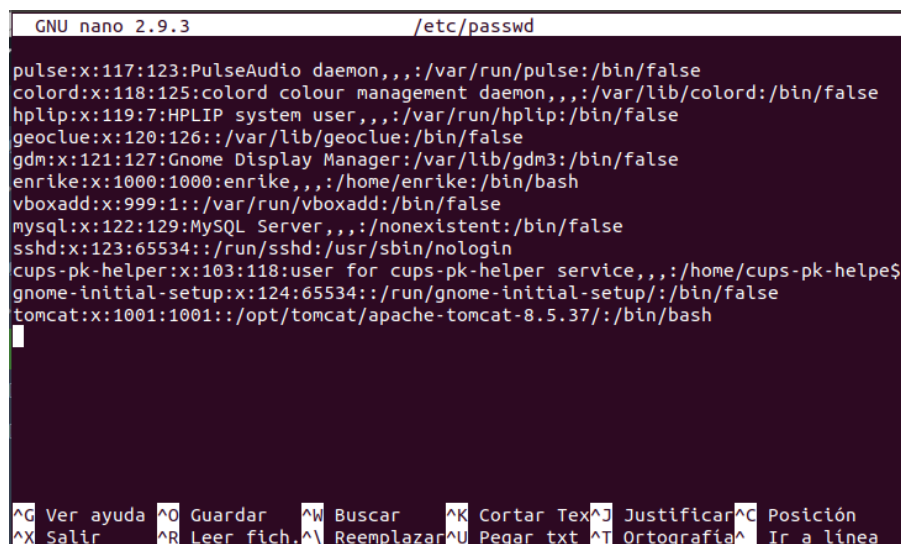
5. Cambiamos el tipo de consola a bash para el usuario tomcat modificando el archivo /etc/passwd con el siguiente comando:

```
# nano /etc/passwd
```

Modificamos la siguiente línea como se muestra a continuación, guardamos los cambios y cerramos el archivo:

Original -> tomcat:x:1001:1001::/usr/**local**/tomcat:/bin/sh

Nueva -> tomcat:x:1001:1001::/opt/tomcat/apache-tomcat-8.5.37/:/bin/bash



6. Por último, cambiamos al propietario del directorio CATALINA_HOME:

```
# cd /opt/
# chown -R tomcat:tomcat tomcat/
# cd /opt/tomcat/
# chown -R tomcat:tomcat apache-tomcat-8.5.37/
```

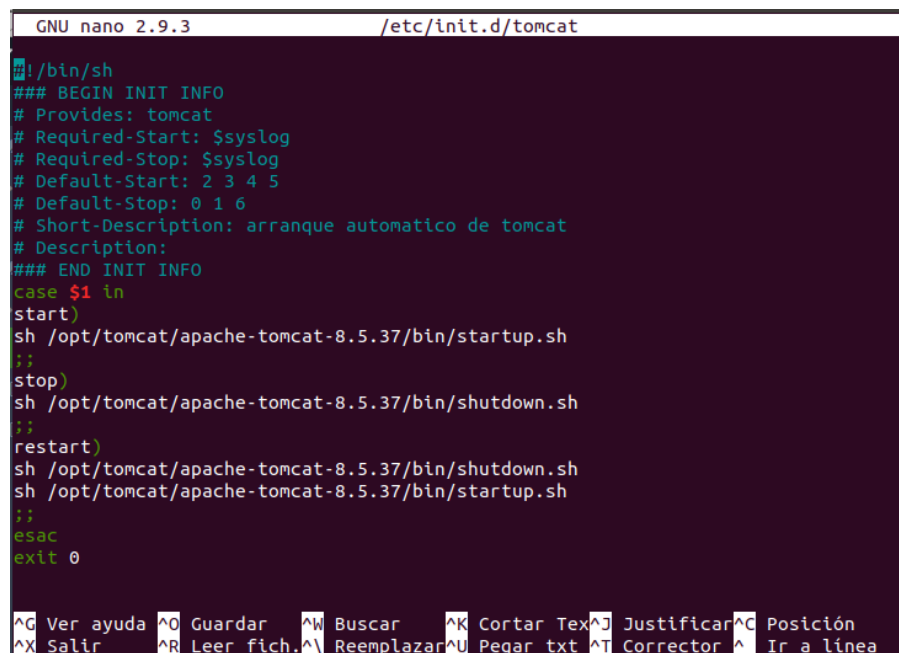
7. A partir de ahora, solo el usuario tomcat puede realizar cambios en éste directorio.

2.4. Arranque automático de Tomcat

1. Abrimos una terminal como superusuario.
2. Creamos un archivo llamado tomcat en /etc/init.d/:

```
# nano /etc/init.d/tomcat
```

3. Escribimos las siguientes líneas en el archivo, que debe quedar como a continuación se muestra:



```
GNU nano 2.9.3 /etc/init.d/tomcat
#!/bin/sh
### BEGIN INIT INFO
# Provides: tomcat
# Required-Start: $syslog
# Required-Stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: arranque automatico de tomcat
# Description:
### END INIT INFO
case $1 in
start)
sh /opt/tomcat/apache-tomcat-8.5.37/bin/startup.sh
;;
stop)
sh /opt/tomcat/apache-tomcat-8.5.37/bin/shutdown.sh
;;
restart)
sh /opt/tomcat/apache-tomcat-8.5.37/bin/shutdown.sh
sh /opt/tomcat/apache-tomcat-8.5.37/bin/startup.sh
;;
esac
exit 0
```

Guardamos los cambios y cerramos el archivo.

4. Modificamos los permisos del archivo creado para poder ser ejecutado:

```
# chmod 755 /etc/init.d/tomcat
```

5. Agregamos un enlace simbólico al script de arranque de Tomcat:

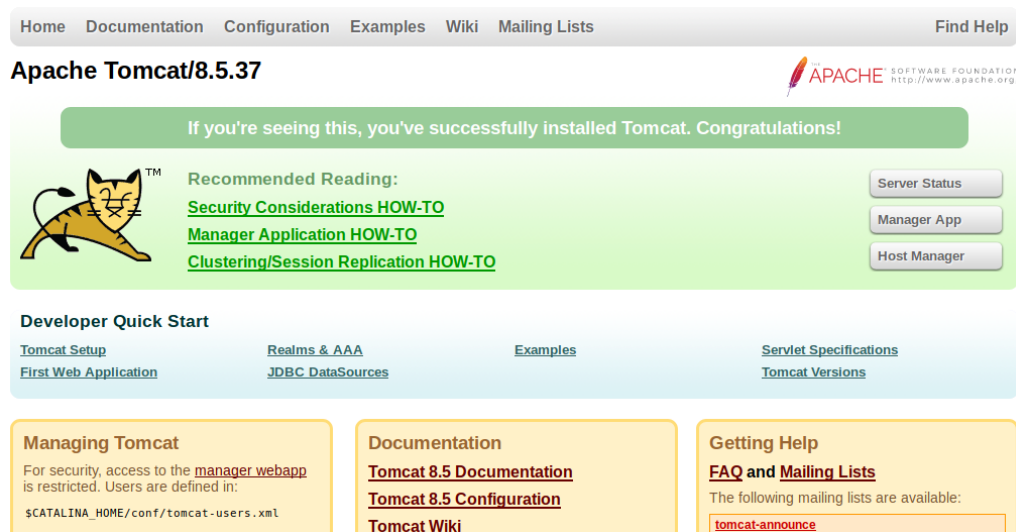
```
# update-rc.d tomcat defaults
```

Ahora cada vez que el sistema operativo inicie, también Tomcat arrancará automáticamente.

6. Por último, verificamos que toda la configuración haya sido correcta ejecutando el siguiente comando:

```
# /etc/init.d/tomcat start
```

7. Abrimos un navegador web y vamos a la dirección: `http://127.0.0.1:8080`. Debe de aparecer una pantalla como la siguiente:



2.5. Configuración de Apache y Apache Tomcat

1. Abrimos una terminal como superusuario. Nos movemos hacia el directorio de CATALINA_HOME, en donde se encuentra apache-tomcat-8.5.37:

```
# cd /opt/tomcat/apache-tomcat-8.5.37/
```

2. Abrimos el archivo `/conf/server.xml` para sólo lectura:

```
# nano /conf/server.xml
```


3. Tenemos que buscar y localizar el siguiente bloque de código.

```
GNU nano 2.9.3 /opt/tomcat/apache-tomcat-8.5.37/conf/server.xml

    </SSLHostConfig>
  </Connector>
  -->

  <!-- Define an AJP 1.3 Connector on port 8009 -->
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

  <!-- An Engine represents the entry point (within Catalina) that processes
  every request. The Engine implementation for Tomcat stand alone
  analyzes the HTTP headers included with the request, and passes them
  on to the appropriate Host (virtual host).
  Documentation at /docs/config/engine.html -->

  <!-- You should set jvmRoute to support load-balancing via AJP ie :
  <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
  -->
  <Engine name="Catalina" defaultHost="localhost">
```

Esto nos va a servir para identificar el puerto que el servidor Apache utiliza para comunicarse con Apache Tomcat, que de forma predeterminada es el 8009. Cerramos el archivo.

4. Abrimos el archivo httpd.conf:

```
# nano /etc/apache2/httpd.conf
```

5. Agregamos las siguientes líneas de código al final de archivo:

```
GNU nano 2.9.3 /etc/apache2/httpd.conf

# Settings so that every request to port 80 are redirected to port 8089 of Tomcat.
MaxClients 250
ProxyPass / ajp://localhost:8009/
ProxyPassReverse / ajp://localhost:8009/
```

- La primera línea especifica el número de clientes que el servidor Apache puede atender concurrentemente.
- La segunda línea redirecciona todo el tráfico al puerto 8009.
- La tercera línea coloca un Reverse Proxy que permite a Apache cambiar los headers de respuesta enviados a Apache Tomcat. Así, cualquier usuario externo verá respuestas provenientes desde Apache y no desde Tomcat.

Guardamos los cambios y cerramos el archivo.

2.5.1. Instalación del servidor Apache

- a) Antes de proseguir y evitar errores, hacemos una desinstalación completa de la versión de Apache que éste instalada en nuestro equipo:

```
# apt-get remove --purge apache2
```

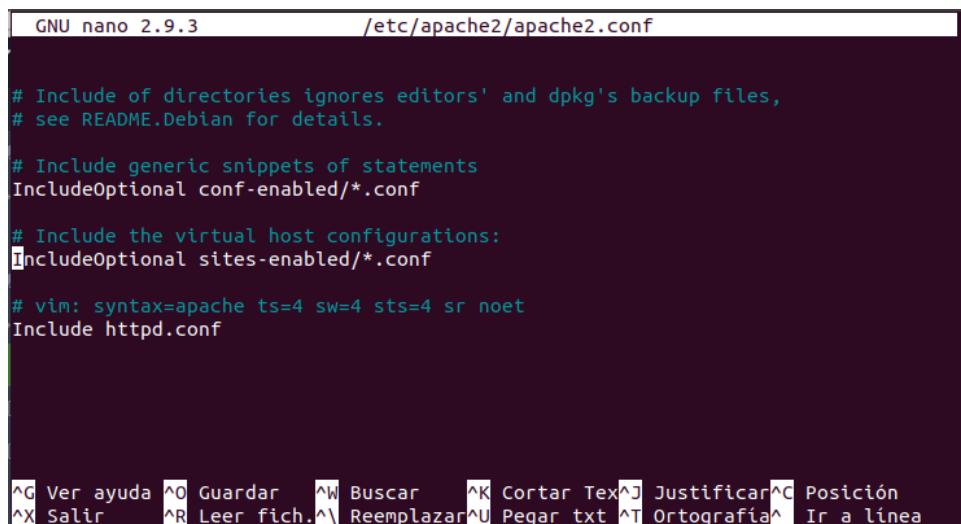
- b) Ahora, instalamos la versión más reciente de Apache:

```
# apt-get install apache2
```

6. Abrimos el archivo `apache2.conf`:

```
# nano /etc/apache2/apache2.conf
```

7. Escribimos al final del archivo la siguiente línea, guardamos los cambios y cerramos el archivo:



```
GNU nano 2.9.3 /etc/apache2/apache2.conf

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
Include httpd.conf

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

8. Instalamos `mod_proxy` con los siguientes comandos:

```
# a2enmod proxy_ajp
# a2enmod proxy
```

9. Reiniciamos el servidor Apache y Apache Tomcat para que se apliquen los cambios:

```
# /etc/init.d/apache2 restart
# /etc/init.d/tomcat restart
```

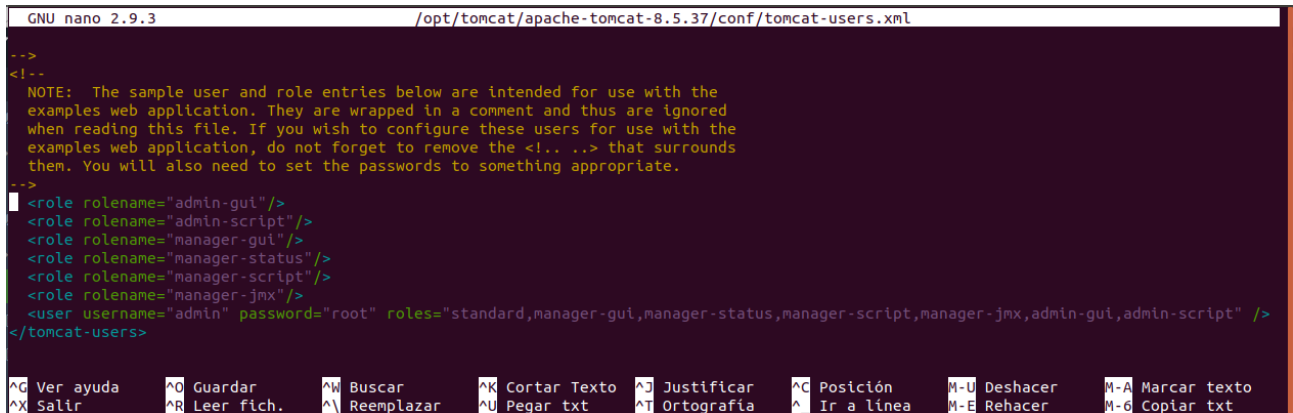
10. Abrimos un navegador web y vamos a la dirección: `http://127.0.0.1:8080`. Debe de aparecer la misma pantalla de Apache Tomcat que en la sección anterior.

2.6. Usuario para Tomcat Manager

1. Vamos a crear al usuario que se encargará de desplegar nuestras aplicaciones en Apache Tomcat.
2. Desde una terminal como superusuario abrimos el archivo `tomcat-users.xml`:

```
# nano /opt/tomcat/apache-tomcat-8.5.37/conf/tomcat-users.xml
```

3. Agregamos las siguientes líneas al final del archivo como se muestra a continuación:



```
GNU nano 2.9.3 /opt/tomcat/apache-tomcat-8.5.37/conf/tomcat-users.xml
-->
<!--
NOTE: The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- ... --> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-status"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<user username="admin" password="root" roles="standard,manager-gui,manager-status,manager-script,manager-jmx,admin-gui,admin-script" />
</tomcat-users>

^G Ver ayuda  ^O Guardar    ^W Buscar     ^K Cortar Texto ^J Justificar  ^C Posición   M-U Deshacer   M-A Marcar texto
^X Salir      ^R Leer fich. ^N Reemplazar ^U Pegar txt    ^T Ortografía ^I Ir a línea  M-E Rehacer   M-6 Copiar txt
```

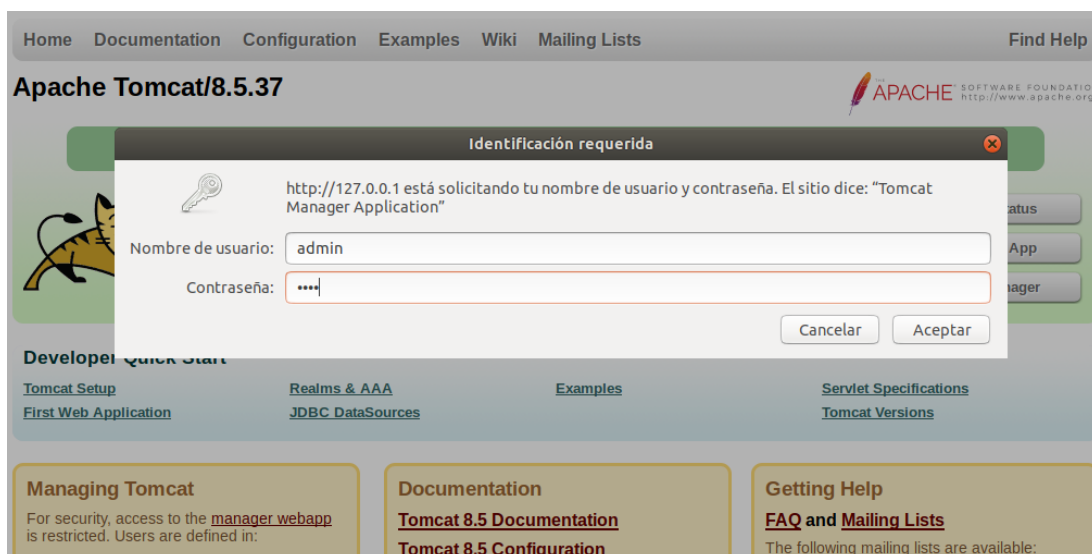
- La primera y ultima linea encapsula al bloque de código para la creación de usuarios.
- La tercera línea crea el rol manager-gui, que es necesario para permitirle a un usuario desplegar aplicaciones en Apache Tomcat.
- La séptima línea crea un usuario especificado con su contraseña especificada.

Guardamos los cambios y cerramos el archivo.

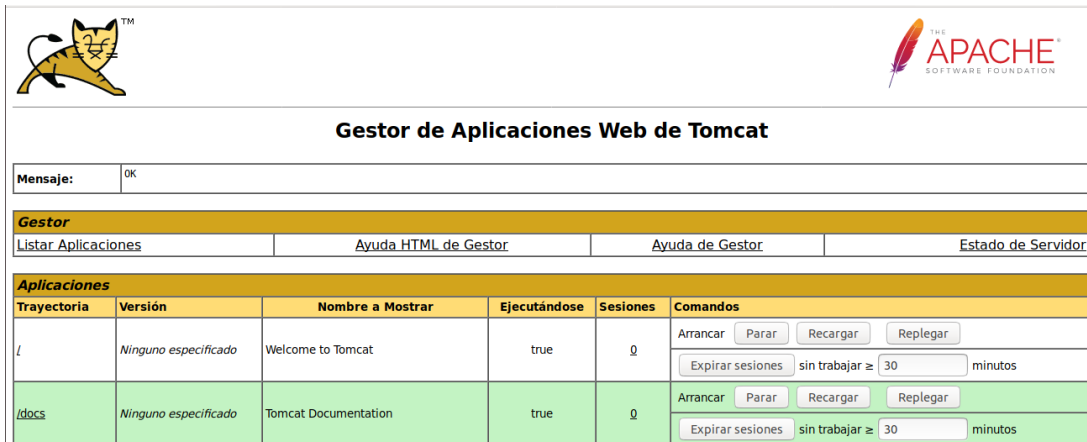
4. Reiniciamos Tomcat:

```
# /etc/init.d/tomcat restart
```

5. Abrimos un navegador web y vamos a la dirección: <http://127.0.0.1:8080>. En la pantalla de inicio de Tomcat, damos clic en el botón de Manager App.
6. Saldrá un cuadro de diálogo solicitándonos un usuario y contraseña, que fue la que escribimos en el archivo tomcat-users.xml. La ingresamos.



7. Si todo salio bien, debe aparecer la siguiente pantalla:



2.7. Cerrar el puerto 8080

1. Abrimos una terminal como superusuario. Nos movemos hacia el directorio de CATALINA_HOME, en donde se encuentra apache-tomcat-8.0.37:

```
# cd /opt/tomcat/apache-tomcat-8.5.37/
```

2. Abrimos el archivo /conf/server.xml para sólo lectura:

```
# nano /conf/server.xml
```

3. Tenemos que buscar y localizar el siguiente bloque de código:

```

/opt/tomcat/apache-tomcat-8.5.37/conf/server.xml

<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html
Java AJP  Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<!-- <Connector port="8080" protocol="HTTP/1.1"
       connectionTimeout="20000"
       redirectPort="8443" />
-->
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
            port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
-->
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation. The default
SSLImplementation will depend on the presence of the APR/native

```

4. Lo comentamos escribiendo:

```
<!-- Bloque... -->
```

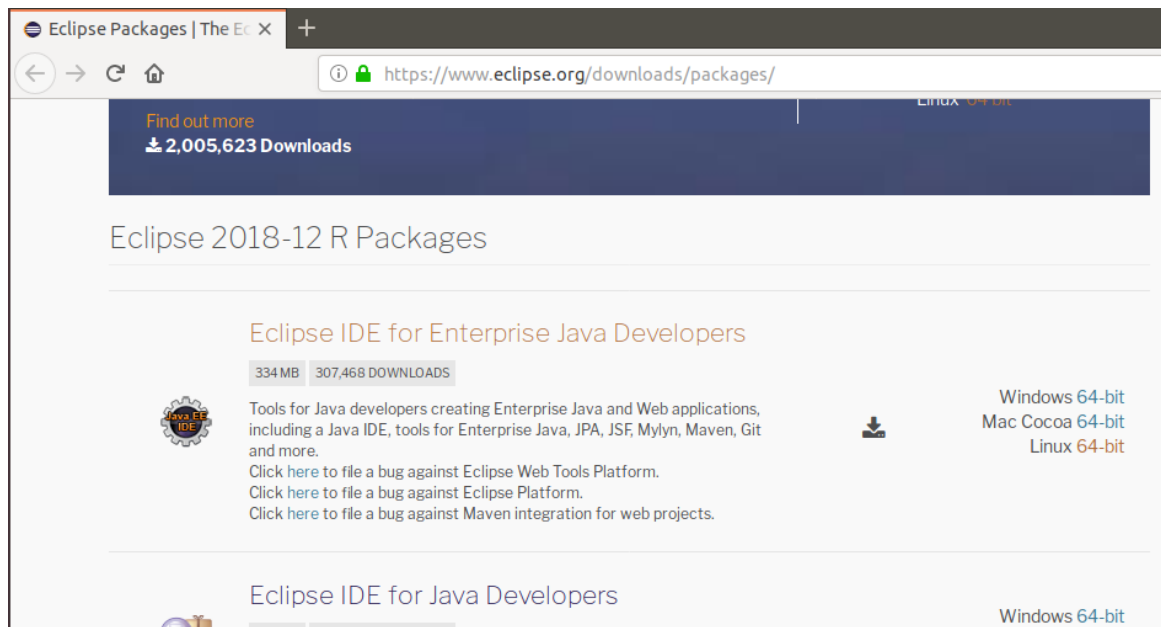
Entre el bloque. Guardamos los cambios y cerramos el archivo.

5. Reiniciamos Tomcat y Apache:

```
# /etc/init.d/apache2 restart
# /etc/init.d/tomcat restart
```

2.8. Instalar el IDE Eclipse for Java EE Developers

1. Descargamos la versión más reciente del IDE Eclipse for Java EE Developers, según nuestro sistema operativo y arquitectura, desde el siguiente link: <https://www.eclipse.org/downloads/packages/>

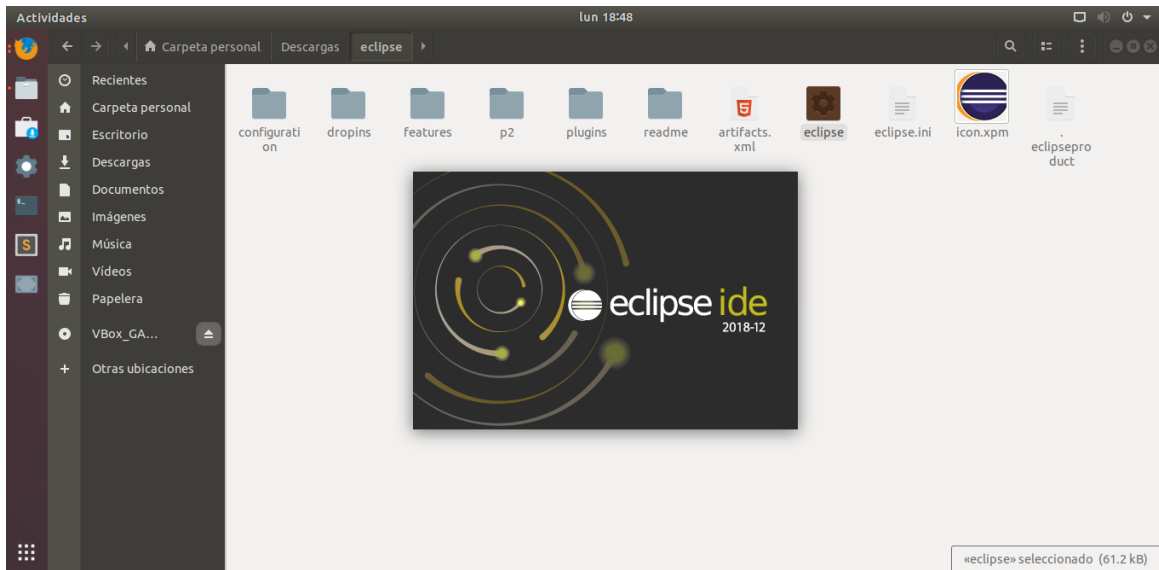


2. Abrimos el directorio en donde se guardó el archivo tar.gz, abrimos una terminal y lo descomprimos:

```
$ tar -xvzf eclipse-jee-2018-12-R-linux-gtk-x86_64.tar.gz
```

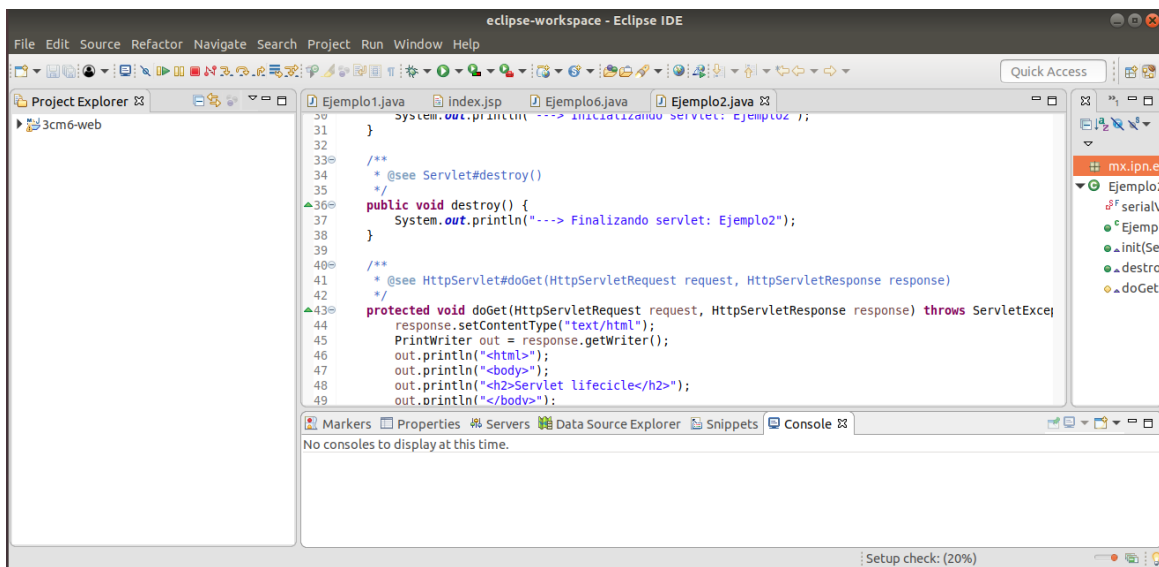
3. Navegamos hasta el directorio en donde se descomprimió el tar.gz llamado eclipse, y abrimos el IDE haciendo doble clic en el archivo *eclipse*, ó abriendo una terminal e ingresando el siguiente comando:

```
$ ./eclipse
```

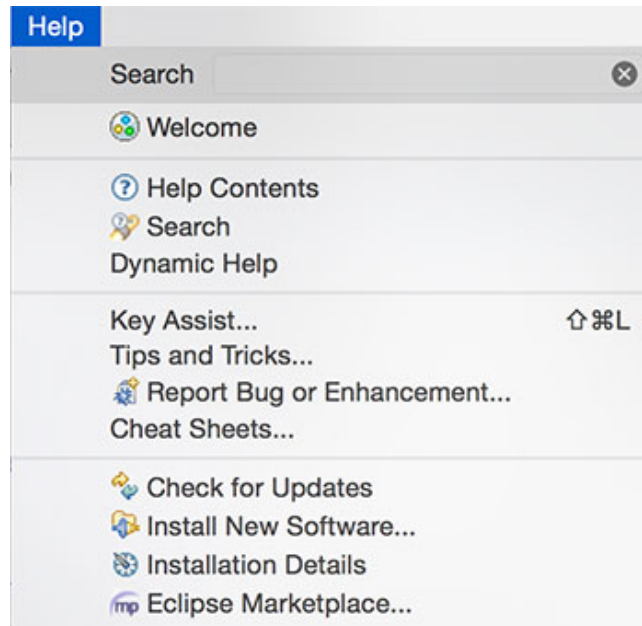


2.9. Instalar Jetty

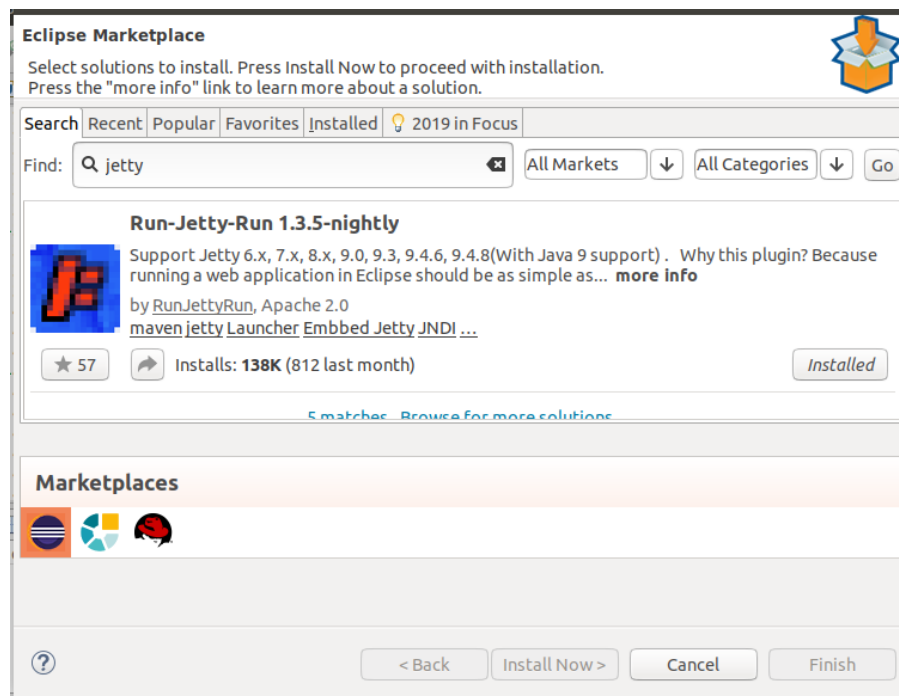
1. Abrimos el IDE Eclipse.



2. Damos clic en la opción Help y luego en Eclipse Marketplace.



3. Se abrirá una ventana. En el campo de Search, escribimos "Jetty" y damos Enter.
4. Seleccionamos la versión de Jetty con el icono azul, y damos clic en el botón instalar.



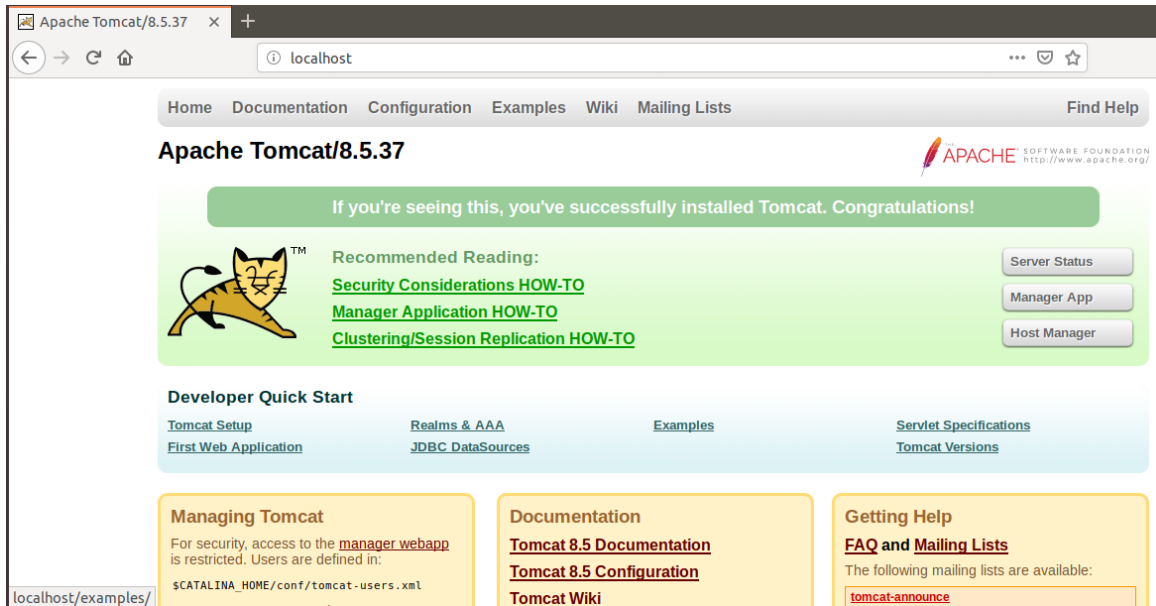
5. Seguimos las instrucciones de la ventana hasta finalizar la instalación.
6. Una vez finalizada, Eclipse aplicará los cambios y se reiniciará.

3. Pruebas

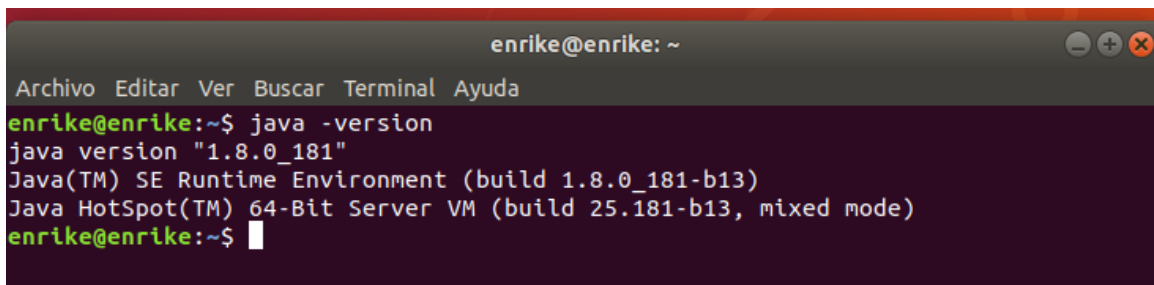
3.1. Pantalla del navegador al intentar acceder a localhost:8080



3.2. Pantalla del navegador al acceder a localhost



3.3. Versión de Java instalada

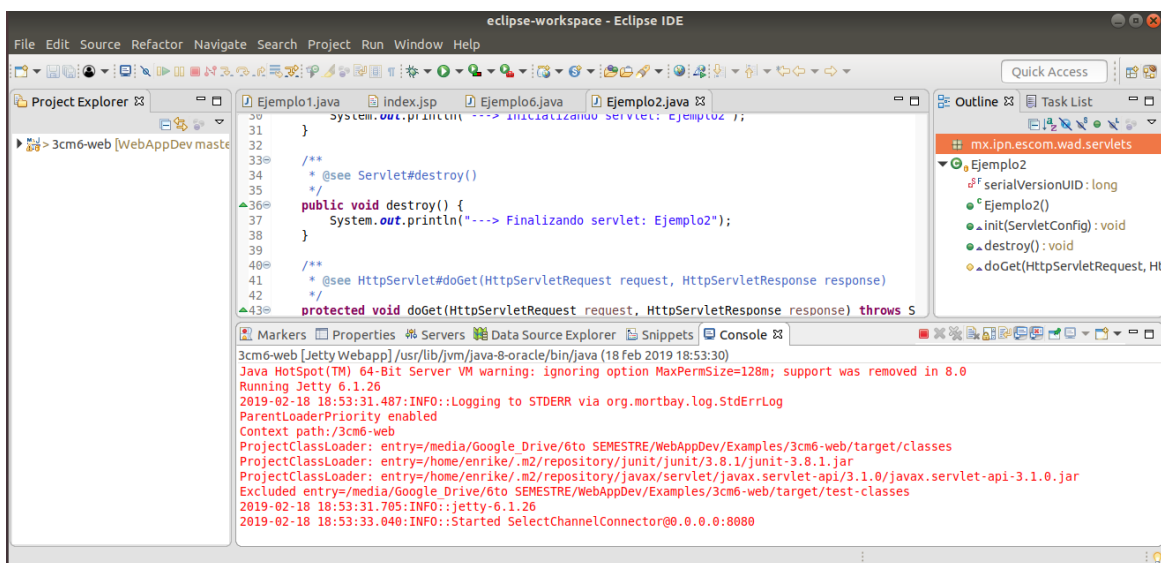


```

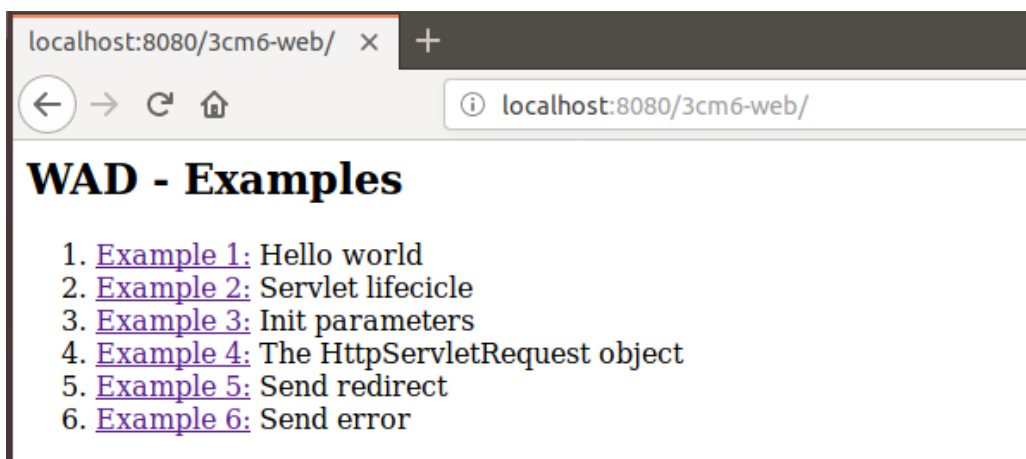
enrike@enrike: ~
Archivo Editar Ver Buscar Terminal Ayuda
enrike@enrike:~$ java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
enrike@enrike:~$

```

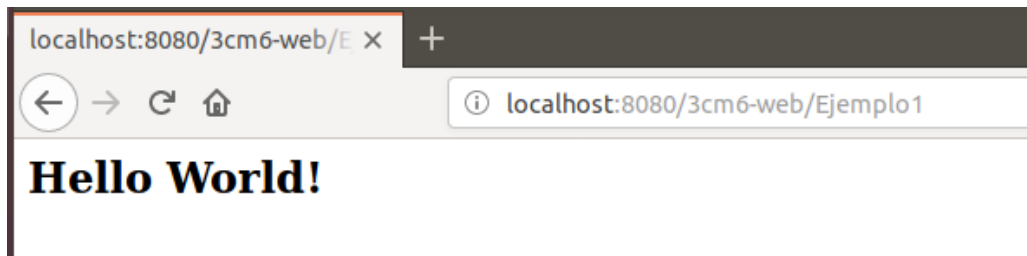
3.4. Eclipse y Jetty ejecutándose



3.5. Ejemplos de servlets en clase, ejecutándose desde Jetty



3.6. Servlet Hola Mundo



4. Conclusiones

Cada configuración del entorno de trabajo en nuestro equipo dependen de las objetivos y necesidades específicas de cada grupo o individuo.

En la primera mitad de ésta práctica configuramos un servidor web, para desplegar y poner en marcha nuestras aplicaciones web, y un lenguaje de programación, en donde se procesará toda la información y se implementarán los métodos de programación. Ésta podría ser la configuración "genérica" para cualquier interesado en empezar a usar servlets, JSP, etc.

Posteriormente, introducimos el servidor Apache a la ecuación, dándole el control sobre el servidor Tomcat: especificando el número de clientes concurrentes, redireccionando el tráfico, y protegiendo la seguridad tanto de Tomcat como de las aplicaciones desarrolladas por el programador.

También cerramos el puerto 8080 para Tomcat, siendo la dirección `localhost` (sin el famoso `:8080`) la única opción para acceder a él. Sin embargo, para ver las aplicaciones web en ejecución por medio de Eclipse y Jetty, sí es necesario agregar el puerto en la url para ver los cambios aplicados en el desarrollo de éstas.

En un futuro, se agregará un balanceador de cargas en el servidor para gestionar y distribuir tráfico y peticiones simultáneas de varios clientes.

Ya por último, instalamos un IDE, que si bien no es del nada obligatorio (se puede desarrollar creando la estructura de directorios especificada), es de mucha ayuda cuando se trabajen en proyectos que requieran el uso de muchas clases, así tenemos un mayor orden y organización al desplegarlas en el servidor web, en tiempo real de compilación y ejecución, con ayuda del plugin Jetty, que nos ahorra todo el trabajo de crear un WAR y desplegarlo manualmente en Tomcat.

5. Bibliografía

[1] (2019) H. F. Montes Casiano. WEB APPLICATION DEVELOPMENT 2018 - 2019 2. Accessed february 2019. [Online]. Available: <http://www.comunidad.escom.ipn.mx/hermes/cursos/WAD20192.xhtml>