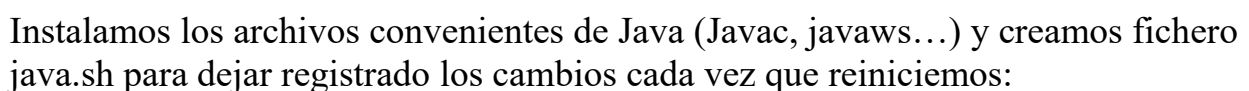


1. Instalación y configuración

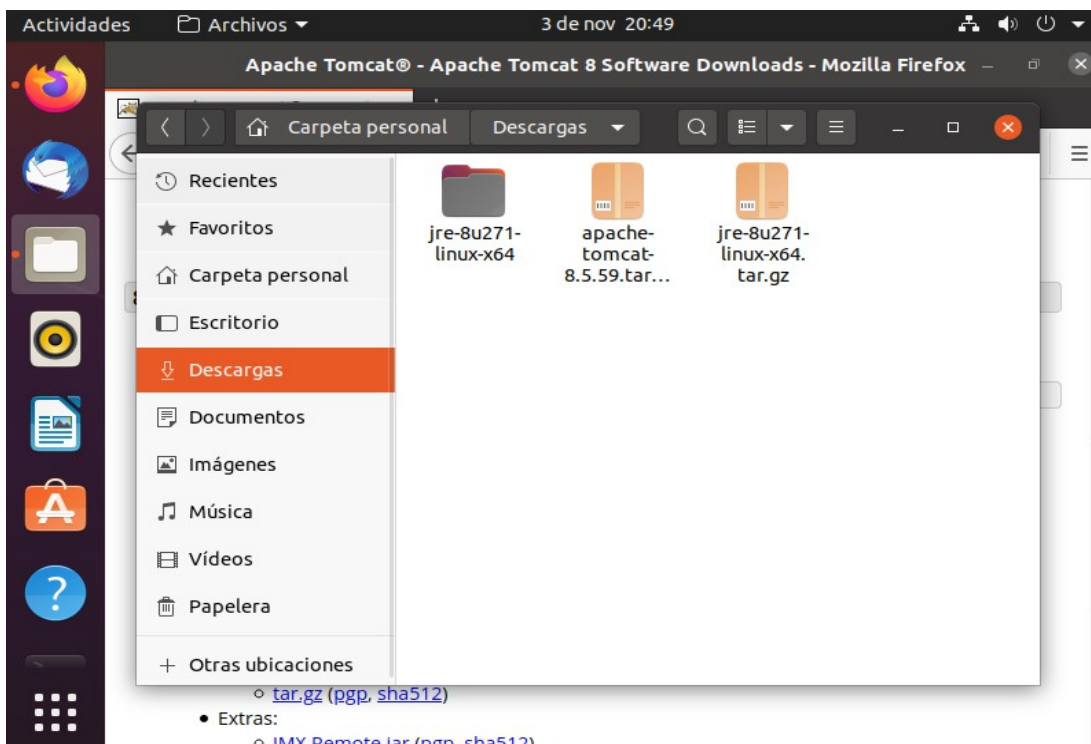
Descargamos y extraemos el JDK 8:



```
root@usuario-B250M-DS3H: ~  
usuario@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271$ cat /etc/profile.d/java.sh  
cat: /etc/profile.d/java.sh: No such file or directory  
usuario@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271$ cat /etc/profile.d/  
cat: /etc/profile.d: Is a directory  
usuario@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271$ cat /etc/profile.d/java.sh  
cat: /etc/profile.d/java.sh: No such file or directory  
usuario@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271$ cat /etc/profile.d/  
cat: /etc/profile.d: Is a directory  
usuario@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271$ sudo su  
root@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271# chmod +x /etc/profile.d/java.sh  
root@usuario-B250M-DS3H: /usr/lib/jvm/jdk1.8.0/jdk-8u271-linux-x64/jdk1.8.0_271# cd  
root@usuario-B250M-DS3H: ~# java -version  
java version "1.8.0_271"  
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)  
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)  
root@usuario-B250M-DS3H: ~# cd  
Command 'cd' not found, did you mean:
```

● Instalación de Apache Tomcat 8

Nos descargamos el archivo “.tar.gz” para así una vez lo tenemos, lo descomprimos:

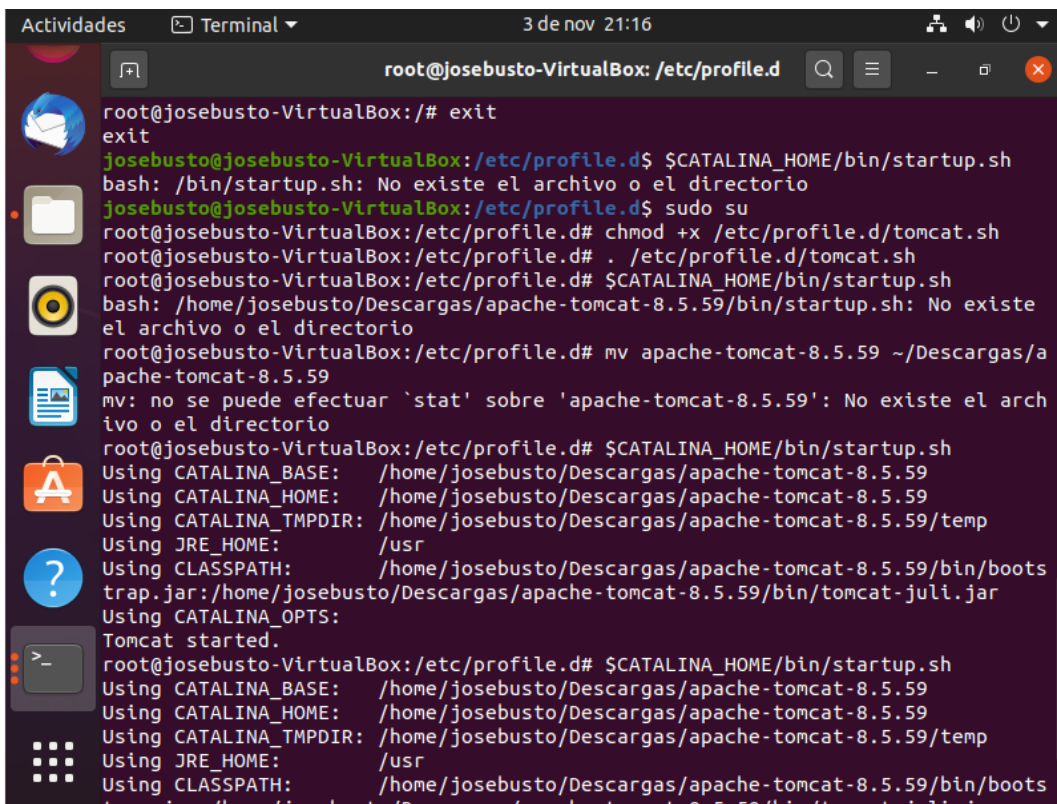


Los siguientes comandos tendremos que tenerlos en cuenta para dar permisos en Ubuntu:

chown -hR usuario:usuario/home/usuario/Downloads/apache-tomcat-8.5.59

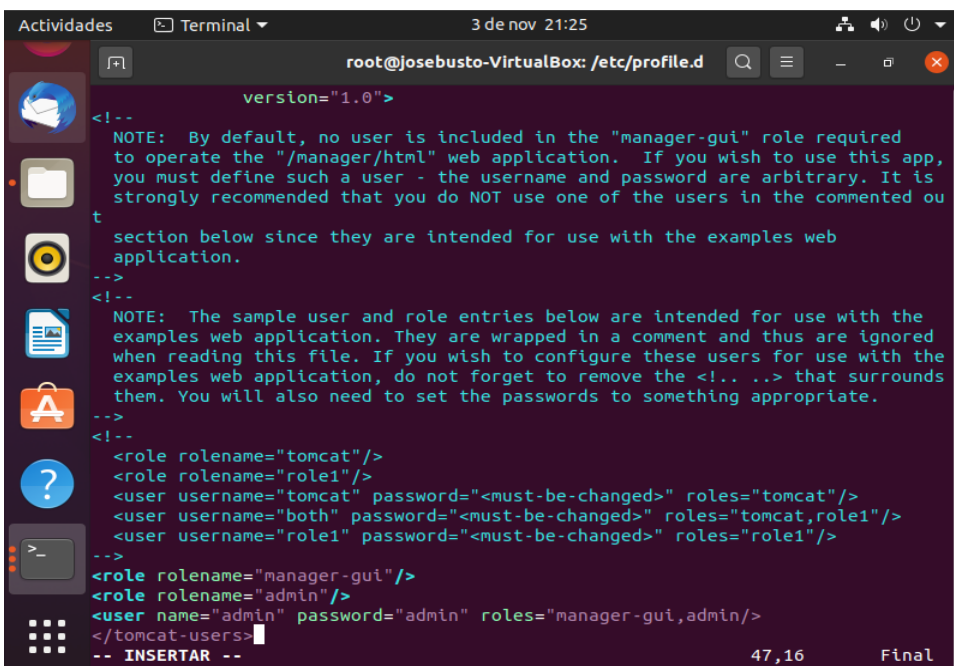
chmod +x /etc/profile.d/tomcat.sh

Ahora una vez hemos colocado Tomcat como servicio y en la ruta requerida lo arrancamos para ver que funciona:



```
root@josebusto-VirtualBox: /etc/profile.d
root@josebusto-VirtualBox: /# exit
exit
josebusto@josebusto-VirtualBox: /etc/profile.d$ $CATALINA_HOME/bin/startup.sh
bash: /bin/startup.sh: No existe el archivo o el directorio
josebusto@josebusto-VirtualBox: /etc/profile.d$ sudo su
root@josebusto-VirtualBox: /etc/profile.d# chmod +x /etc/profile.d/tomcat.sh
root@josebusto-VirtualBox: /etc/profile.d# . /etc/profile.d/tomcat.sh
root@josebusto-VirtualBox: /etc/profile.d# $CATALINA_HOME/bin/startup.sh
bash: /home/josebusto/Descargas/apache-tomcat-8.5.59/bin/startup.sh: No existe
el archivo o el directorio
root@josebusto-VirtualBox: /etc/profile.d# mv apache-tomcat-8.5.59 ~/Descargas/a
pache-tomcat-8.5.59
mv: no se puede efectuar `stat' sobre `apache-tomcat-8.5.59': No existe el arch
ivo o el directorio
root@josebusto-VirtualBox: /etc/profile.d# $CATALINA_HOME/bin/startup.sh
Using CATALINA_BASE:   /home/josebusto/Descargas/apache-tomcat-8.5.59
Using CATALINA_HOME:   /home/josebusto/Descargas/apache-tomcat-8.5.59
Using CATALINA_TMPDIR: /home/josebusto/Descargas/apache-tomcat-8.5.59/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/josebusto/Descargas/apache-tomcat-8.5.59/bin/boots
trap.jar:/home/josebusto/Descargas/apache-tomcat-8.5.59/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@josebusto-VirtualBox: /etc/profile.d# $CATALINA_HOME/bin/startup.sh
Using CATALINA_BASE:   /home/josebusto/Descargas/apache-tomcat-8.5.59
Using CATALINA_HOME:   /home/josebusto/Descargas/apache-tomcat-8.5.59
Using CATALINA_TMPDIR: /home/josebusto/Descargas/apache-tomcat-8.5.59/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/josebusto/Descargas/apache-tomcat-8.5.59/bin/boots
trap.jar:/home/josebusto/Descargas/apache-tomcat-8.5.59/bin/tomcat-juli.jar
Using CATALINA_OPTS:
```

Establecemos los privilegios en el archivo correspondiente dandole un usuario y contraseña:



```
version="1.0">
<!--
NOTE: By default, no user is included in the "manager-gui" role required
to operate the "/manager/html" web application.  If you wish to use this app,
you must define such a user - the username and password are arbitrary.  It is
strongly recommended that you do NOT use one of the users in the commented ou
t
section below since they are intended for use with the examples web
application.
-->
<!--
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="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"/>
<role rolename="admin"/>
<user name="admin" password="admin" roles="manager-gui,admin"/>
</tomcat-users>
-- INSERTAR --
47,16 Final
```

Una vez se establecen los datos podemos entrar en el localhost de Tomcat:

Actividades | Navegador web Firefox | 3 de nov 21:32

/manager - Mozilla Firefox

Apache Tomcat® - Apache | Desarrollo Web con Java | /manager

localhost:8080/manager/html

Gestor de Aplicaciones Web de Tomcat

Mensaje: OK

Gestor

[Listar Aplicaciones](#) | [Ayuda HTML de Gestor](#) | [Ayuda de Gestor](#) | [Estado de Servidor](#)

Aplicaciones

Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
					Arrancar Parar

Vemos como podemos desplegar sin problemas el archivo que acabamos de añadir:

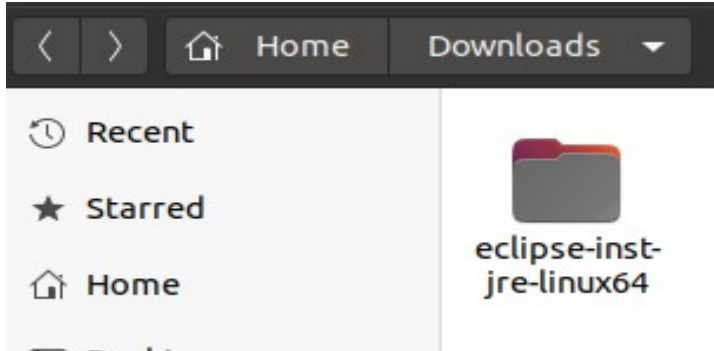
Desarrollo Web con Java | SimpleServlet

localhost:8080/SimpleServlet/simple-servlet

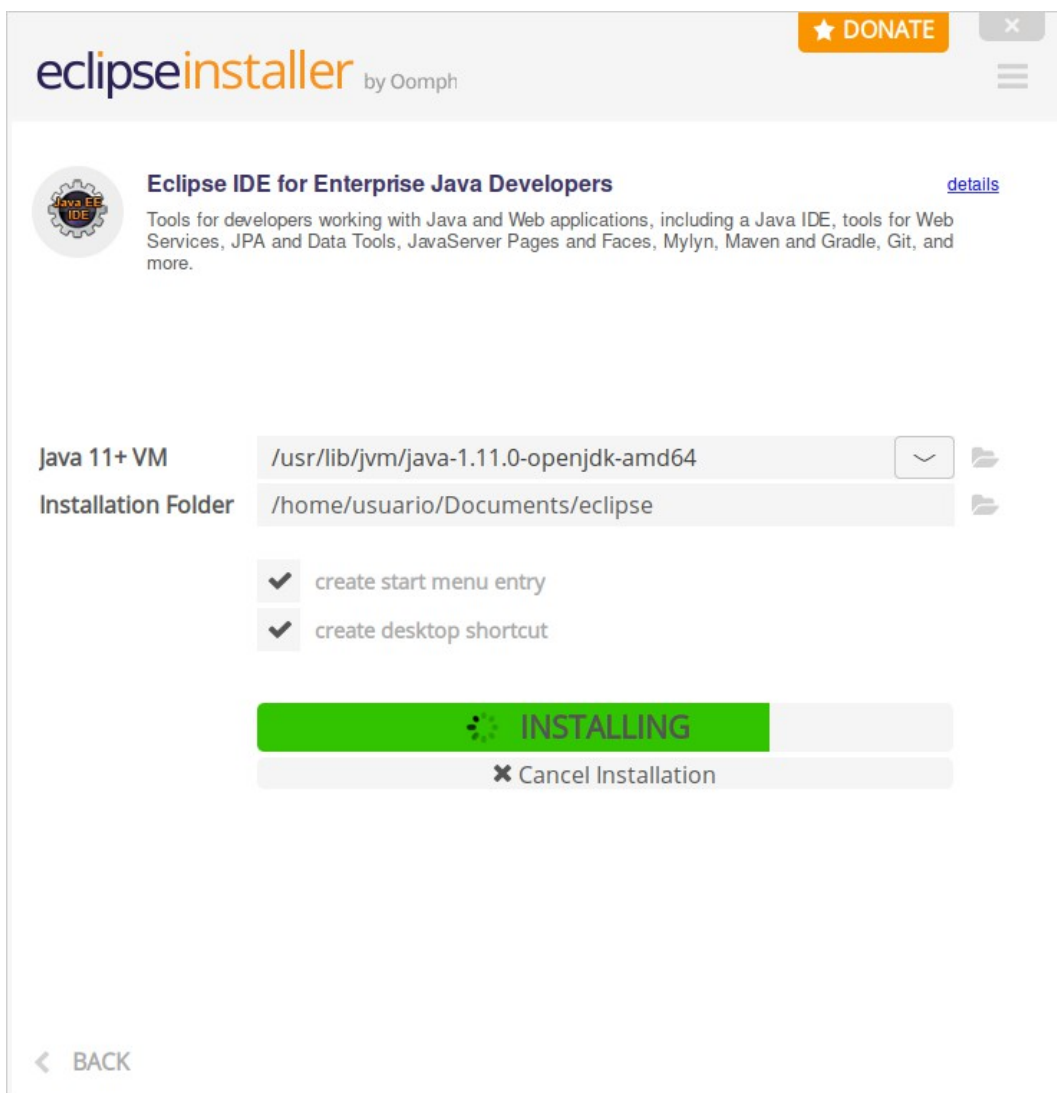
Hello, World

● Instalación de Eclipse IDE

Una vez hemos instalado el JVM y Tomcat, ahora descargamos el IDE Eclipse:

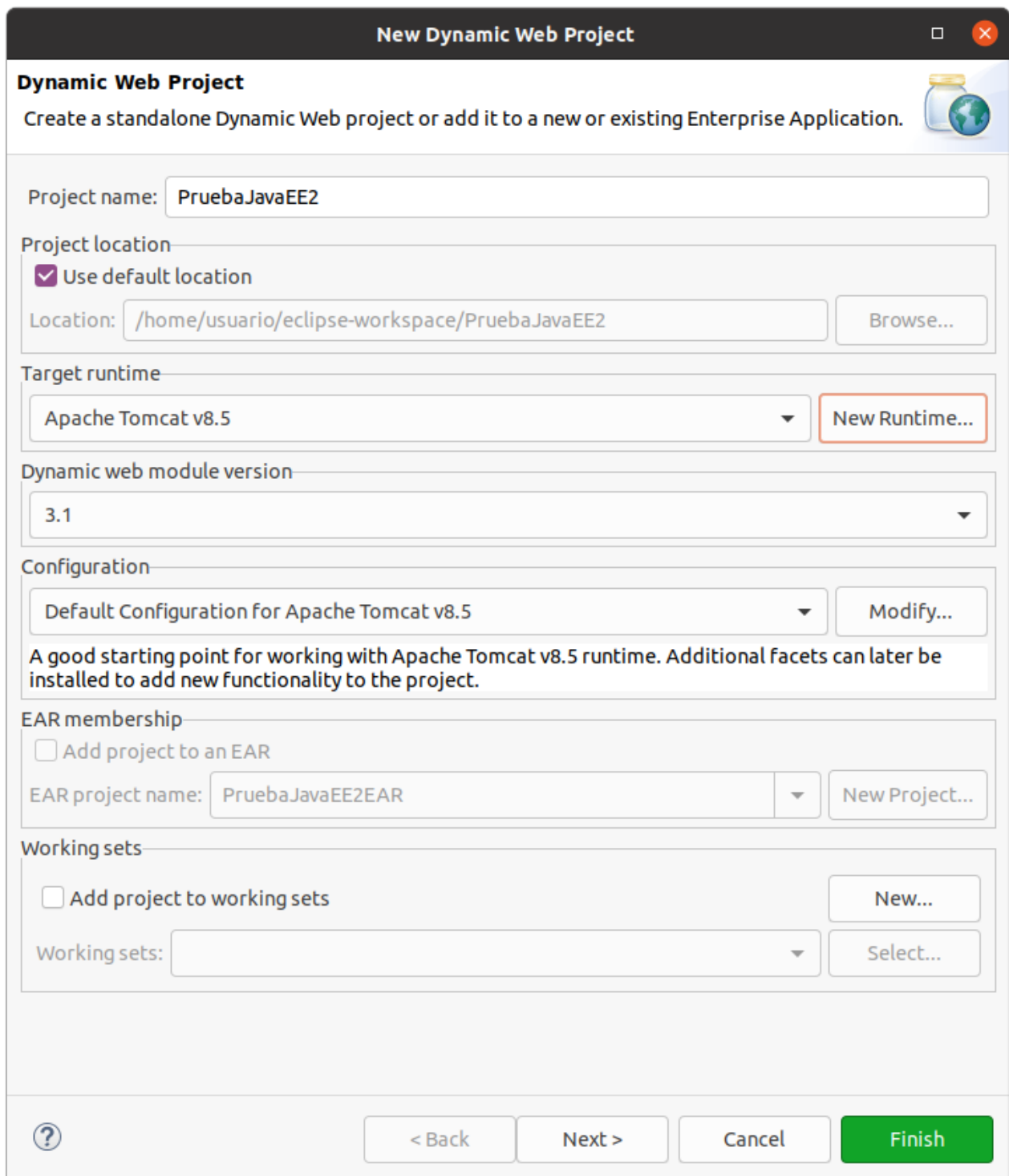


Y lo descargamos e instalamos con su launcher gráfico:



Una vez ha terminado la instalación de Eclipse, procedemos a crear un nuevo proyecto, lo haremos de la siguiente forma: File → New → Dynamic Web Project.

El siguiente paso será dejar integrado e indicado como se llamará nuestro Proyecto e integrar nuestro servidor Tomcat:



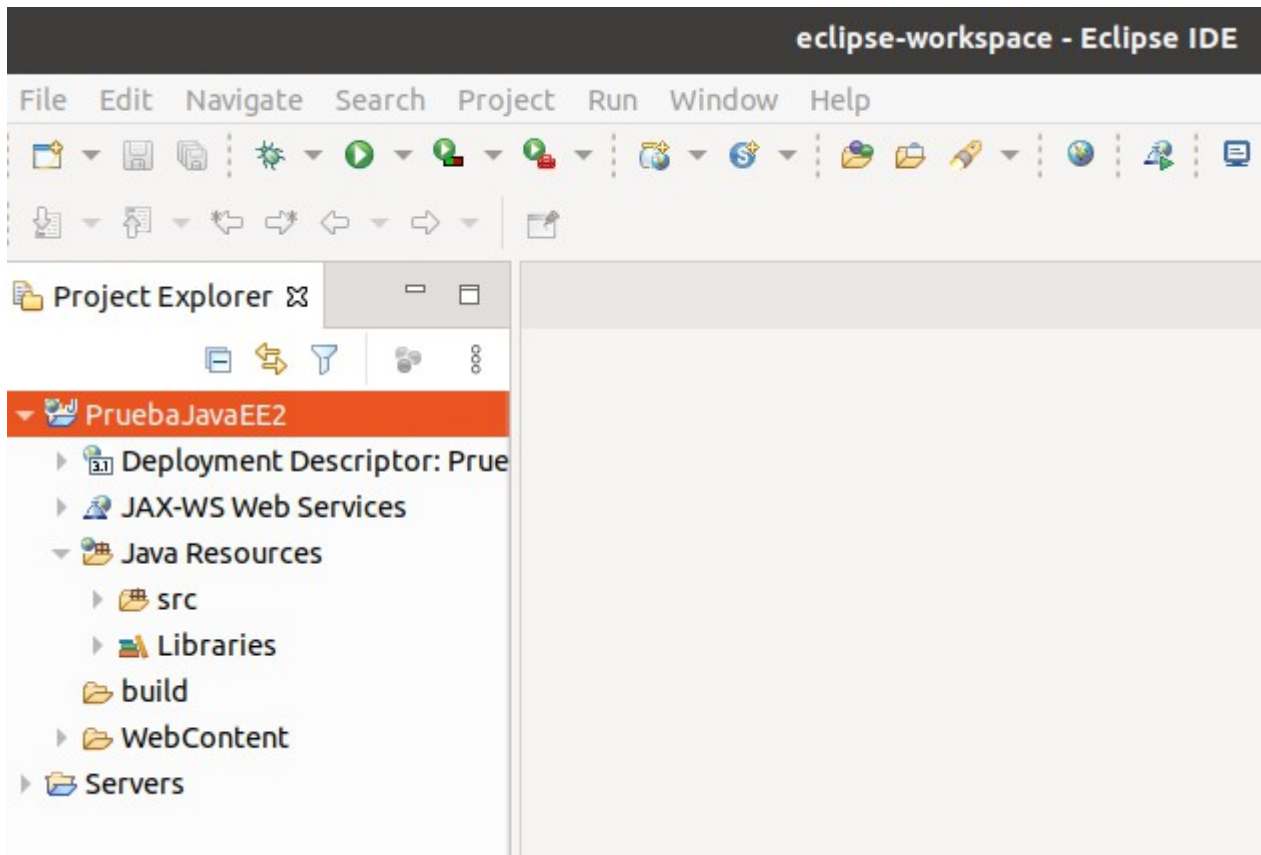
The screenshot shows the 'New Dynamic Web Project' dialog box in Eclipse. The title bar reads 'New Dynamic Web Project'. The main heading is 'Dynamic Web Project' with a subtitle 'Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.' and a small icon of a jar and globe.

The dialog is divided into several sections:

- Project name:** A text field containing 'PruebaJavaEE2'.
- Project location:** A section with a checked checkbox 'Use default location' and a text field 'Location: /home/usuario/eclipse-workspace/PruebaJavaEE2' with a 'Browse...' button.
- Target runtime:** A dropdown menu showing 'Apache Tomcat v8.5' and a 'New Runtime...' button.
- Dynamic web module version:** A dropdown menu showing '3.1'.
- Configuration:** A dropdown menu showing 'Default Configuration for Apache Tomcat v8.5' and a 'Modify...' button. Below this is a text box: 'A good starting point for working with Apache Tomcat v8.5 runtime. Additional facets can later be installed to add new functionality to the project.'
- EAR membership:** A section with an unchecked checkbox 'Add project to an EAR', an 'EAR project name:' text field containing 'PruebaJavaEE2EAR', and a 'New Project...' button.
- Working sets:** A section with an unchecked checkbox 'Add project to working sets', a 'New...' button, a 'Working sets:' dropdown menu, and a 'Select...' button.

At the bottom, there is a help icon (question mark), and four buttons: '< Back', 'Next >', 'Cancel', and a green 'Finish' button.

Aquí podemos ver que nuestro proyecto se ha creado correctamente:



● Instalación de MySQL server y MySQL workbench

Ahora pasamos a descargar MySQL server y MySQL workbench.

El server podemos descargarlo por la terminal de la siguiente forma:

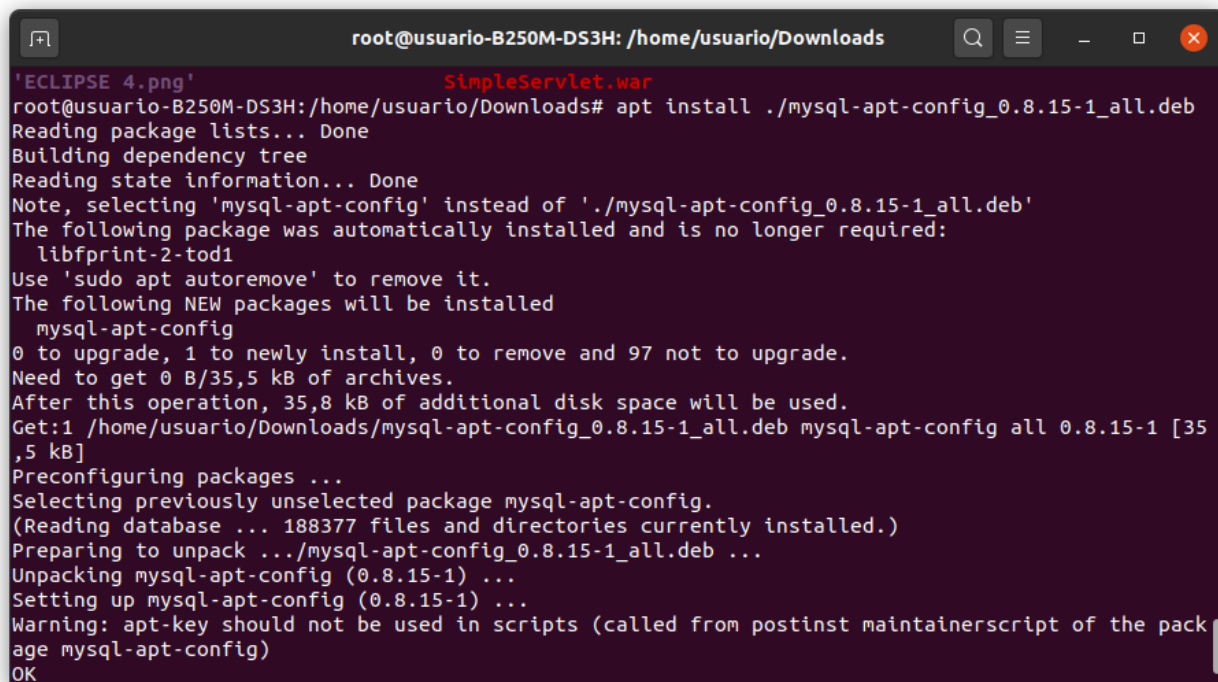
```
root@usuario-B250M-DS3H: /
E: Unable to locate package mysql-workbench
root@usuario-B250M-DS3H: /# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7 libevent-pthreads-2.1-7
  libfcgi-perl libhtml-template-perl libmecab2 mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinycd
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7 libevent-pthreads-2.1-7
  libfcgi-perl libhtml-template-perl libmecab2 mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 to upgrade, 16 to newly install, 0 to remove and 97 not to upgrade.
Need to get 30,4 MB of archives.
After this operation, 248 MB of additional disk space will be used.
Do you want to continue? [S/n] s
Get:1 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client-core-8.0 amd64
8.0.22-0ubuntu0.20.04.2 [4.202 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client-8.0 amd64 8.0.
```

Una vez hemos esperado a que termine de instalarse procedemos con el workbench. Este se debe descargar por Internet en las últimas versiones de Ubuntu.

(Usar este link para descargarlo:

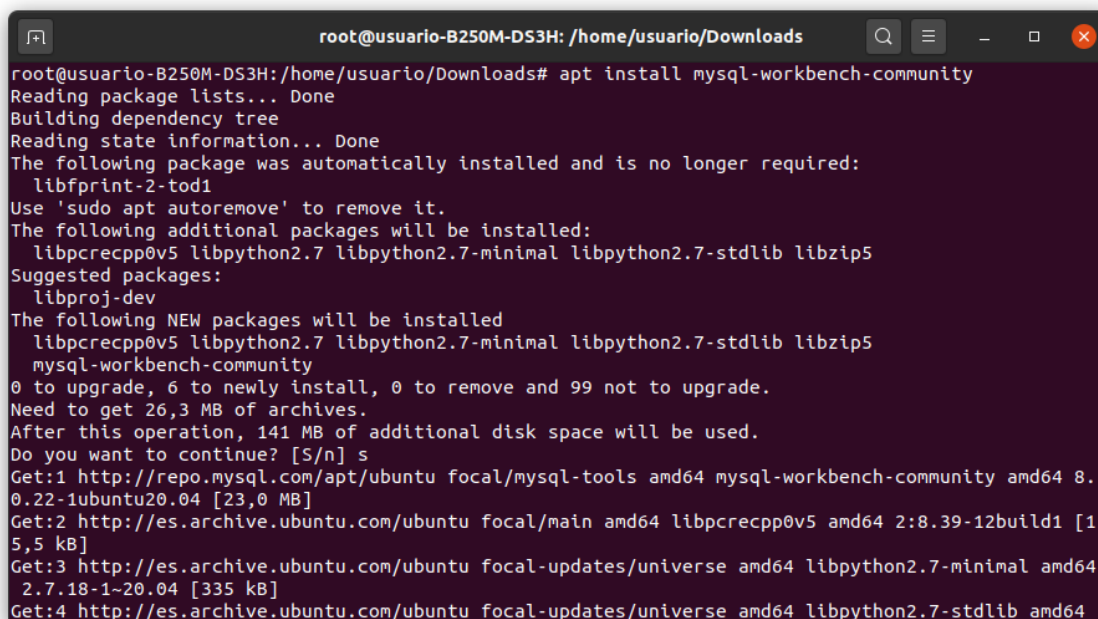
<https://dev.mysql.com/downloads/repo/apt/>.)

Después de descargarnos el instalador de la página anterior, realizamos por terminal lo siguiente:

A terminal window titled 'root@usuario-B250M-DS3H: /home/usuario/Downloads' showing the command 'apt install ./mysql-apt-config_0.8.15-1_all.deb'. The output shows the package being installed, with a note about libfprint-2-tod1 being removed. The terminal text is as follows:

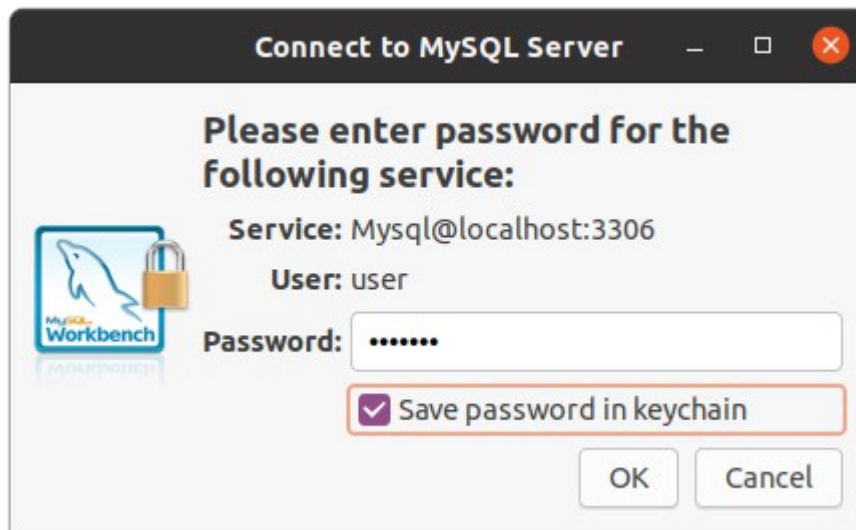
```
root@usuario-B250M-DS3H: /home/usuario/Downloads# apt install ./mysql-apt-config_0.8.15-1_all.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'mysql-apt-config' instead of './mysql-apt-config_0.8.15-1_all.deb'
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  mysql-apt-config
0 to upgrade, 1 to newly install, 0 to remove and 97 not to upgrade.
Need to get 0 B/35,5 kB of archives.
After this operation, 35,8 kB of additional disk space will be used.
Get:1 /home/usuario/Downloads/mysql-apt-config_0.8.15-1_all.deb mysql-apt-config all 0.8.15-1 [35,5 kB]
Preconfiguring packages ...
Selecting previously unselected package mysql-apt-config.
(Reading database ... 188377 files and directories currently installed.)
Preparing to unpack .../mysql-apt-config_0.8.15-1_all.deb ...
Unpacking mysql-apt-config (0.8.15-1) ...
Setting up mysql-apt-config (0.8.15-1) ...
Warning: apt-key should not be used in scripts (called from postinst maintainerscript of the package mysql-apt-config)
OK
```

De esa forma nos lo instalamos en nuestro equipo.

A terminal window titled 'root@usuario-B250M-DS3H: /home/usuario/Downloads' showing the command 'apt install mysql-workbench-community'. The output shows the package being installed, with a note about libfprint-2-tod1 being removed. The terminal text is as follows:

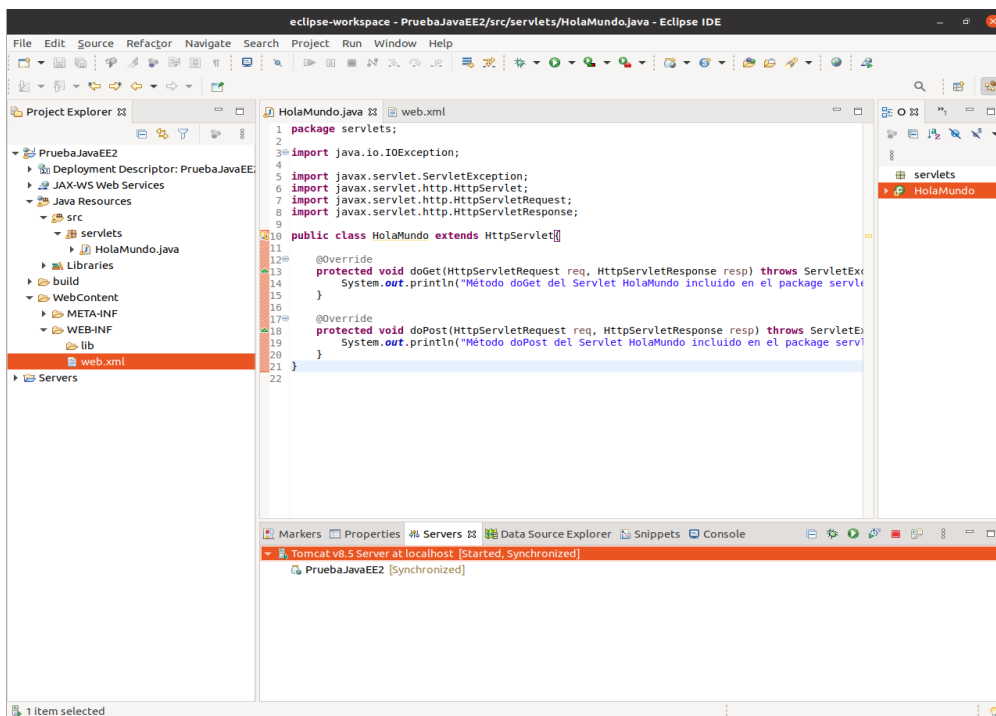
```
root@usuario-B250M-DS3H: /home/usuario/Downloads# apt install mysql-workbench-community
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libpcrecpp0v5 libpython2.7 libpython2.7-minimal libpython2.7-stdlib libzip5
Suggested packages:
  libproj-dev
The following NEW packages will be installed:
  libpcrecpp0v5 libpython2.7 libpython2.7-minimal libpython2.7-stdlib libzip5
  mysql-workbench-community
0 to upgrade, 6 to newly install, 0 to remove and 99 not to upgrade.
Need to get 26,3 MB of archives.
After this operation, 141 MB of additional disk space will be used.
Do you want to continue? [S/n] s
Get:1 http://repo.mysql.com/apt/ubuntu focal/mysql-tools amd64 mysql-workbench-community amd64 8.0.22-1ubuntu20.04 [23,0 MB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/main amd64 libpcrecpp0v5 amd64 2:8.39-12build1 [15,5 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libpython2.7-minimal amd64 2.7.18-1~20.04 [335 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libpython2.7-stdlib amd64
```


Arrancamos MySQL workbench y clickamos sobre nuestro localhost creado por defecto, poniendo así nuestra contraseña:



Conceptos esenciales de Java Servlets

Ahora crearemos el primer servlet a través del método normal de usar el New – Class:



Ahora lo creamos a través de New – Servlet:

Create Servlet
Specify class file destination.

Project: PruebaJavaEE2

Source folder: /PruebaJavaEE2/src Browse...

Java package: servlets Browse...

Class name:

Superclass: javax.servlet.http.HttpServlet Browse...

☐ Use an existing Servlet class or JSP

Class name: Browse...

? < Back Next > Cancel Finish

Esta sería la clase hecha con la opción Servlet:

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left shows the project structure: PruebaJavaEE2 > src > servlets > HolaMundoIDE.java. The main editor displays the code for HolaMundoIDE.java, which is a Servlet implementation class. The code includes package declarations, imports, and annotations for web services. The bottom of the IDE shows the Servers tab with Tomcat v8.5 Server at localhost [Started, Restart] and the PruebaJavaEE2 [Synchronized] project.

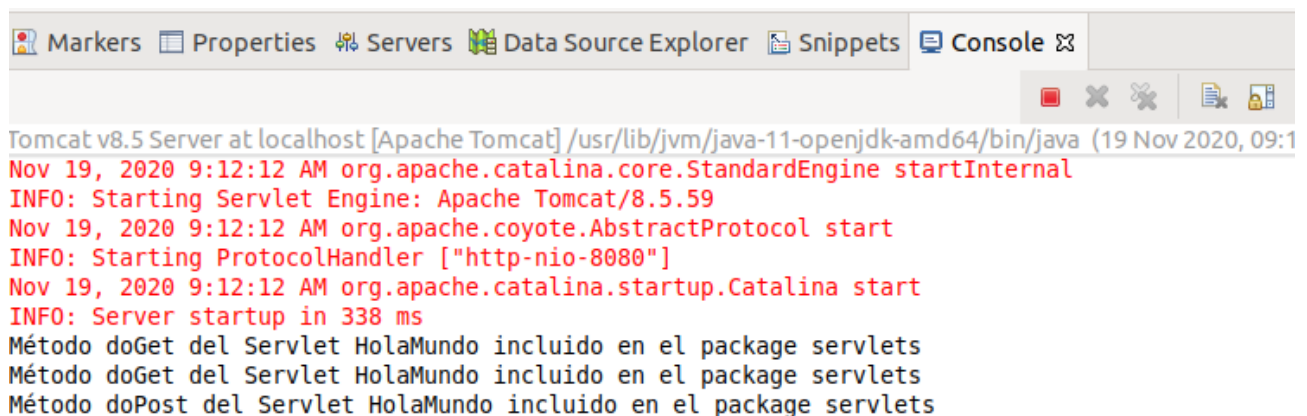
```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class ServletPrueba
7  */
8 @WebServlet("/servletprueba")
9 public class HolaMundoIDE extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14      */
15     public HolaMundoIDE() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22      */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         response.getWriter().append("Served at: ").append(request.getContextPath());
26     }
27
28     /**
29      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
30      */
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32         // TODO Auto-generated method stub
33         doGet(request, response);
34     }
35 }
```

- **Ejecutar un servlet desde una URL**

Ahora vamos a generar una conexión con nuestro servidor y gracias a un HTML y un enlace generaremos nuestro servlet para conectarnos a él:

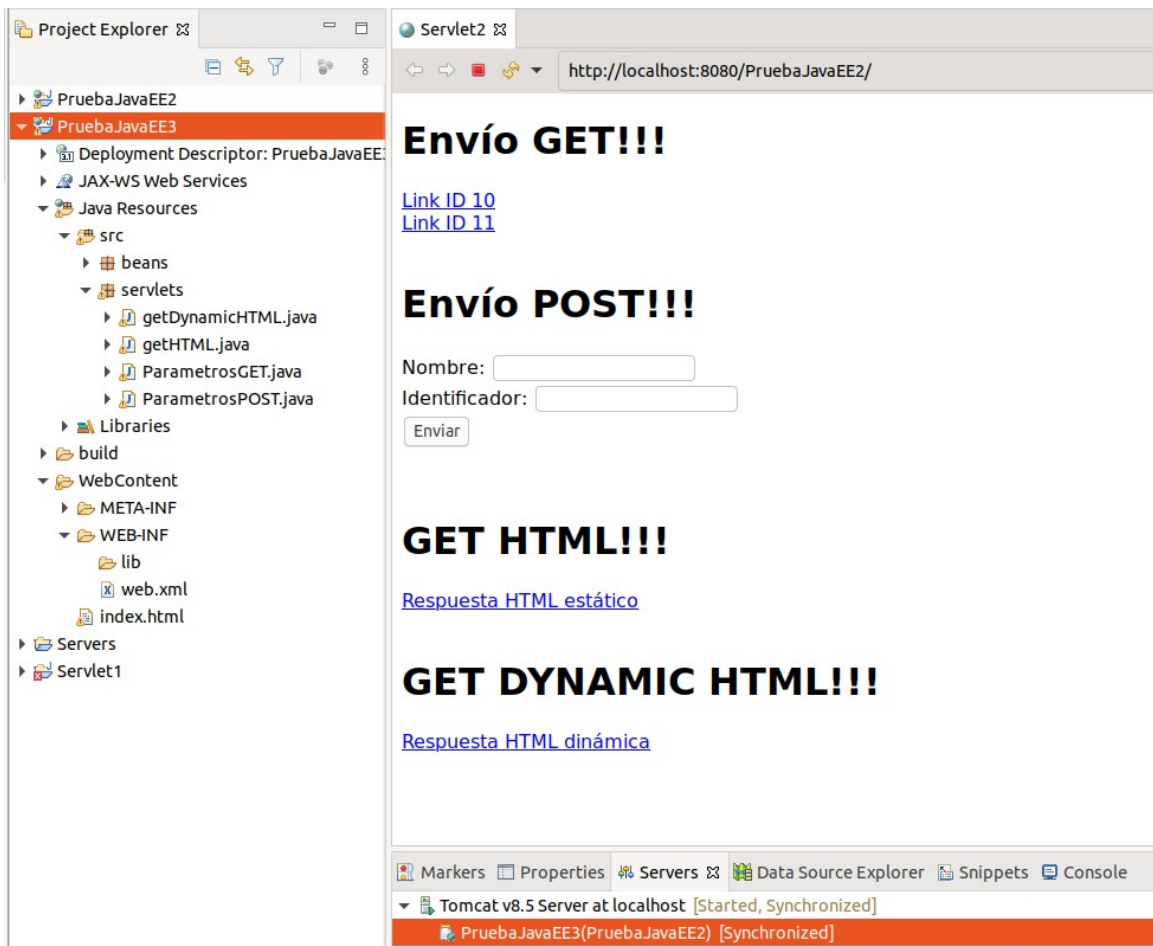


En la consola de Eclipse, Tomcat nos va dejando los diferentes errores, registros y validaciones que se hacen cuando interactuamos con el servlet:

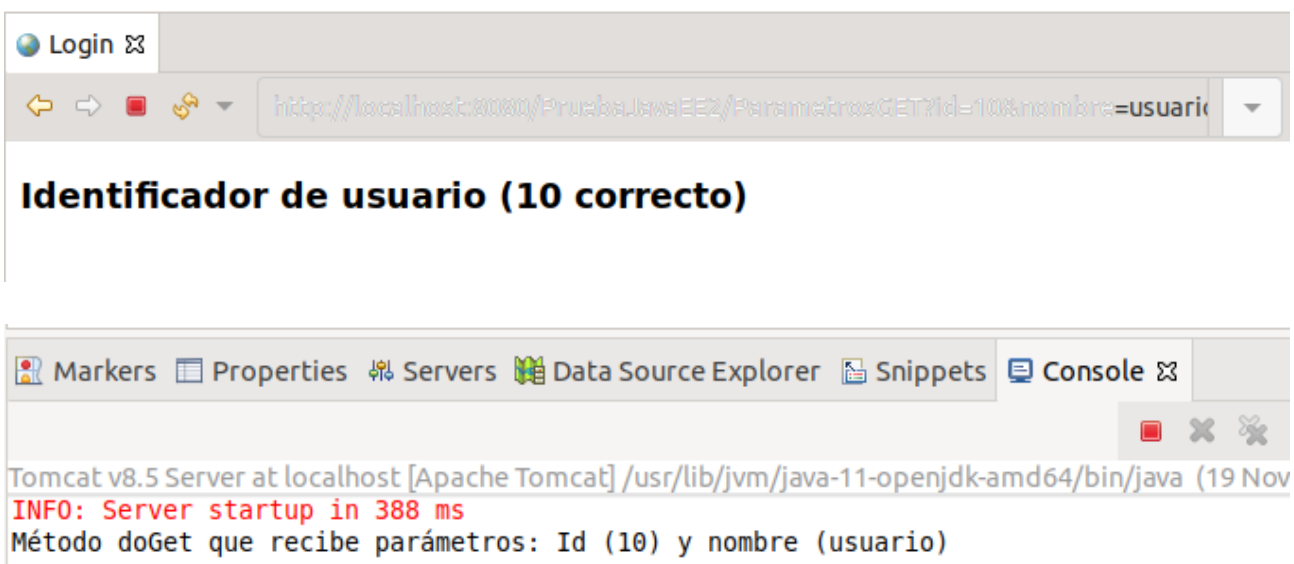


- **Enviar y recibir parámetros en un Java Servlet**

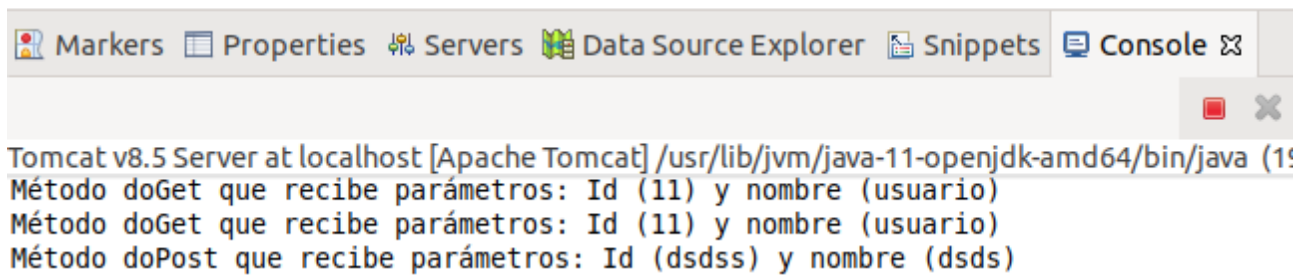
En la siguiente imagen se muestra la ejecución de un proyecto Servlet, el cual tiene sus clases Java y sus funcionalidades HTML y Web. Este ejemplo muestra como se pasan parámetros y como se reciben:



En la siguiente imagen vemos en su url como se ha pasado mediante el link dos parámetros que son: id y nombre.



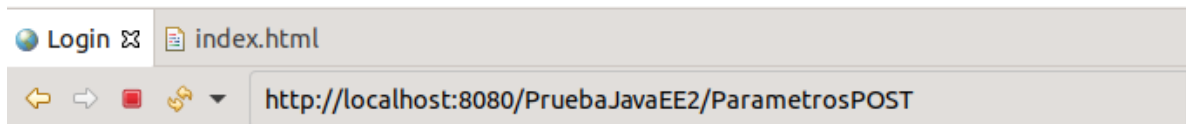
Ahora podemos ver la diferencia registrada de pasarle los parámetros por el método doGet a pasárselo por doPost():



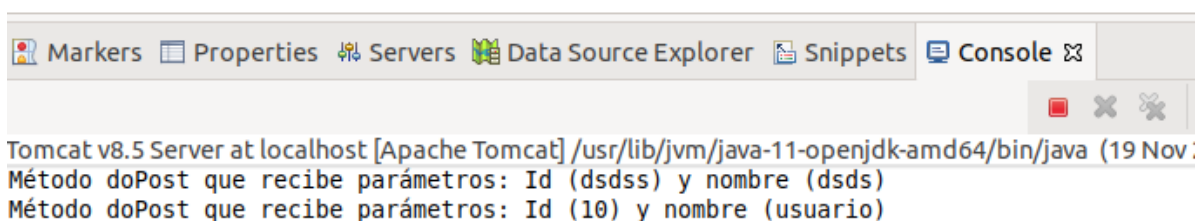
Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v8.5 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (19
Método doGet que recibe parámetros: Id (11) y nombre (usuario)
Método doGet que recibe parámetros: Id (11) y nombre (usuario)
Método doPost que recibe parámetros: Id (dsdss) y nombre (dsds)

En cambio si ponemos los datos correctos que nos pide el formulario de id y nombre nos enviará una nueva información la consola y el HTML:



Identificador de usuario (10 correcto)

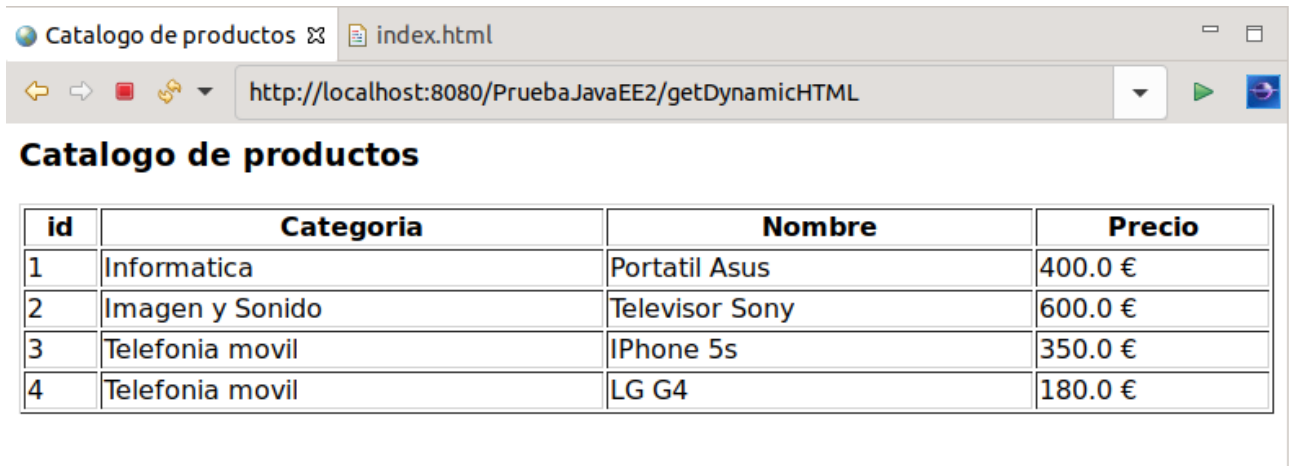


Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v8.5 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (19 Nov:
Método doPost que recibe parámetros: Id (dsdss) y nombre (dsds)
Método doPost que recibe parámetros: Id (10) y nombre (usuario)

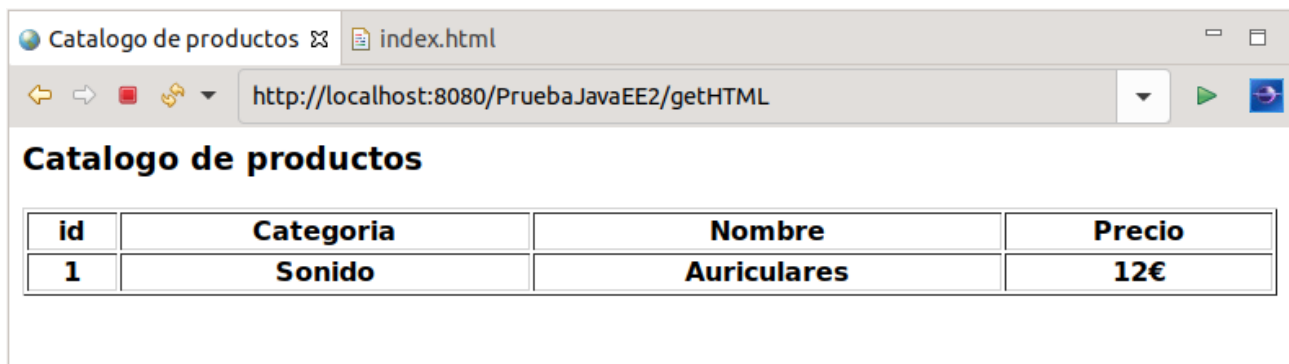
- **Generar código HTML dinámico en un Servlet**

Si queremos usar esta forma diferente de HTML se puede realizar de forma que en nuestra clase Java cargaremos nuestro contenido HTML dinámico para mostrarlo de esta forma:



The screenshot shows a web browser window titled 'Catalogo de productos' with the file 'index.html' open. The address bar shows the URL 'http://localhost:8080/PruebaJavaEE2/getDynamicHTML'. The page content is titled 'Catalogo de productos' and displays a table with 4 rows of product data.

id	Categoria	Nombre	Precio
1	Informatica	Portatil Asus	400.0 €
2	Imagen y Sonido	Televisor Sony	600.0 €
3	Telefonia movil	IPhone 5s	350.0 €
4	Telefonia movil	LG G4	180.0 €



The screenshot shows a web browser window titled 'Catalogo de productos' with the file 'index.html' open. The address bar shows the URL 'http://localhost:8080/PruebaJavaEE2/getHTML'. The page content is titled 'Catalogo de productos' and displays a table with 1 row of product data.

id	Categoria	Nombre	Precio
1	Sonido	Auriculares	12€

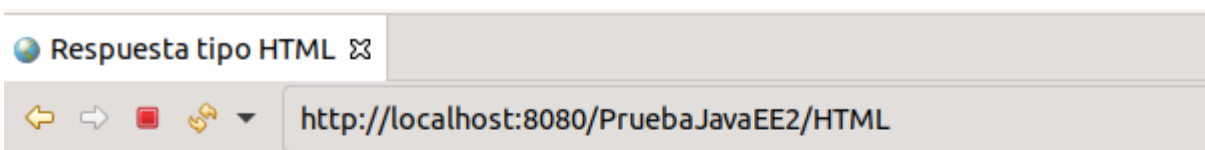
La primera imagen usa HTML Dinámico y en la segunda se usa el estático.

- **Encabezado de respuesta (Content-Type)**

En el siguiente ejemplo vemos las diferentes respuestas para los tipos de encabezados con nuestros servlets:



Por ejemplo algunos tipos son:



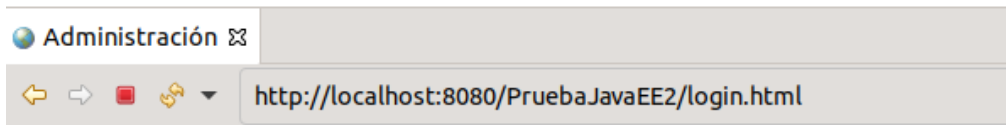
Hemos recibido tu informacion.
Gracias por tus comentarios.



- **Redireccionar respuesta a otra URL**

En este apartado se ve como el método setRedirect nos servirá para a través de un html Login, poner nuestros datos y si es incorrecto se dirige a la clase de ERROR y si es correcto se redirige al HTML correcto con los enlaces.

En las siguientes imagenes se muestra este proceso:

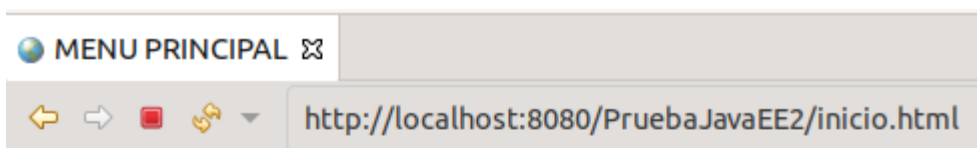
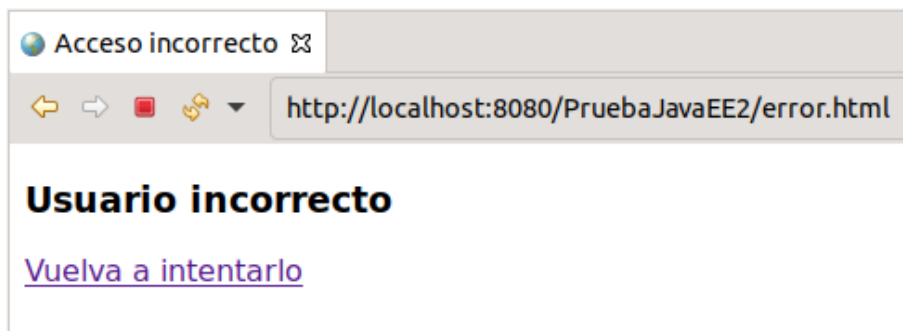


Sistema de Administracion



Usuario:

Password:



Bienvenido al menu principal

[Menu 1](#)

[Menu 2](#)

[Menu 3](#)

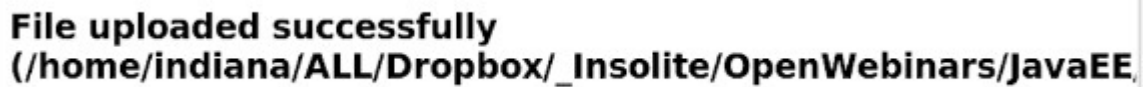
[Menu 4](#)

[Salir](#)

- **Subir archivos al servidor utilizando un Servlet**

En este caso se verá dos casos en los que la única diferencia será la longitud de URL a la hora de subir un archivo externo por nuestra página HTML. En el primer caso será muy larga y en el segundo será diferente ya que se hace desde un entorno de preproducción.

En las siguientes imágenes se muestran ejemplos (imágenes no tomadas por mi ya que a mi me da error):



File uploaded successfully
(/home/indiana/ALL/Dropbox/_Insolite/OpenWebinars/JavaEE

File uploaded successfully (/home/indiana/tomcat/webapps/Servlet4//uploads)!