

Actividad 03 Grupal

Configuración de una Aplicación Web

Daniel Díaz-Delgado Menéndez
Víctor López Gómez
Francisco José de Vicente Escolante



ÍNDICE

Requerimiento 1:

- 1.- BBDD → Fco. José de Vicente Escolante Págs. 01 - 14
- 2.- Apache CDN → Daniel Díaz-Delgado Menéndez Págs. 14 -
- 3.- FTP → Víctor Gómez López Págs.

Repositorio GitHub <https://github.com/grupoed1daw/ActividadGrupoDespliegues2>

Actividad 3. Tarea de equipo (3 personas). Configuración de aplicación web

Tarea en equipo



Requerimiento 1

Se pide desplegar y configurar una aplicación web compuesta de diversos módulos. El fichero desplegable se publicará en la plataforma el 1 de Febrero de 2021.

Cada integrante del equipo tendrá la labor de configurar uno de los elementos integrantes del despliegue: DNS, Host, FTP; pero cada uno trabajará con su propia máquina Linux, por lo que deberán interactuar y compartir sus configuraciones.

1.- BASE DE DATOS

El proyecto se encuentra en GitHub: <https://github.com/ocenteno/sample-jndi>

Configuración

1. Base de Datos

Se debe crear un recurso "jdbc/poolBaseDatos" que contendrá los parámetros de conexión. La Base de Datos deberá popularse con la siguiente estructura:

```
CREATE TABLE PERSONAS (  
    DNI VARCHAR(8) DEFAULT '00000000' NOT NULL,  
    NOMBRE VARCHAR(20),  
    APELLIDOS VARCHAR(20),  
    PRIMARY KEY (dni)  
);
```

IMPORTANTE. Es vuestra responsabilidad crear el usuario de conexión a la BD y *añadirlo* a la configuración del recurso jdbc/poolBaseDatos.

Configuración

Primero: Antes de nada, tenemos que tener instalado Apache Tomcat. Para ello, abrimos nuestra terminal de Ubuntu. Tenemos que comprobar si tenemos instalado java, ya que sin ello no podremos acceder al servidor. Podemos comprobarlo con el comando "java -version"

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
fran@fran-VirtualBox:~$ java -version  
No se ha encontrado la orden «java», pero se puede instalar con:  
sudo apt install default-jre  
sudo apt install openjdk-11-jre-headless  
sudo apt install openjdk-8-jre-headless  
fran@fran-VirtualBox:~$
```

Como vemos, no tenemos instalado java. Para instalarlo, utilizamos el comando “`sudo apt install default-jre`” como nos muestra Ubuntu en las sugerencias.

```
Archivo Editar Ver Buscar Terminal Ayuda
Seleccionando el paquete openjdk-11-jre:amd64 previamente no seleccionado.
Preparando para desempaquetar .../5-openjdk-11-jre_11.0.10+9-0ubuntu1~18.04_amd64.deb ...
Desempaquetando openjdk-11-jre:amd64 (11.0.10+9-0ubuntu1~18.04) ...
Seleccionando el paquete default-jre previamente no seleccionado.
Preparando para desempaquetar .../6-default-jre_2%3a1.11-68ubuntu1~18.04.1_amd64.deb ...
Desempaquetando default-jre (2:1.11-68ubuntu1~18.04.1) ...
Seleccionando el paquete fonts-dejavu-extra previamente no seleccionado.
Preparando para desempaquetar .../7-fonts-dejavu-extra_2.37-1_all.deb ...
Desempaquetando fonts-dejavu-extra (2.37-1) ...
Seleccionando el paquete libatk-wrapper-java previamente no seleccionado.
Preparando para desempaquetar .../8-libatk-wrapper-java_0.33.3-20ubuntu0.1_all.deb ...
Desempaquetando libatk-wrapper-java (0.33.3-20ubuntu0.1) ...
Seleccionando el paquete libatk-wrapper-java-jni:amd64 previamente no seleccionado.
Preparando para desempaquetar .../9-libatk-wrapper-java-jni_0.33.3-20ubuntu0.1_amd64.deb ...
Desempaquetando libatk-wrapper-java-jni:amd64 (0.33.3-20ubuntu0.1) ...
Configurando java-common (0.68ubuntu1~18.04.1) ...
Configurando libgif7:amd64 (5.1.4-2ubuntu0.1) ...
Configurando fonts-dejavu-extra (2.37-1) ...
Configurando libatk-wrapper-java (0.33.3-20ubuntu0.1) ...
Configurando libatk-wrapper-java-jni:amd64 (0.33.3-20ubuntu0.1) ...
Configurando openjdk-11-jre-headless:amd64 (11.0.10+9-0ubuntu1~18.04) ...
Progreso: [ 80%] [#####.....]
```

Una vez lo hayamos instalado, podemos volver a ejecutar el primer comando “`java -version`” para comprobar que se ha instalado correctamente y además, ver qué versión de java tenemos.

```
fran@fran-VirtualBox:~$ java -version
openjdk version "11.0.10" 2021-01-19
OpenJDK Runtime Environment (build 11.0.10+9-Ubuntu-0ubuntu1.18.04)
OpenJDK 64-Bit Server VM (build 11.0.10+9-Ubuntu-0ubuntu1.18.04, mixed mode, sharing)
```

Segundo: Ahora que ya tenemos java instalado, vamos a proceder a instalar nuestro servidor Tomcat. Para ello, ejecutamos el siguiente comando “`sudo apt install tomcat9`”

```
fran@fran-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
fran@fran-VirtualBox:~$ sudo apt install tomcat9
[sudo] contraseña para fran:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9-common
Paquetes sugeridos:
  tomcat9-admin tomcat9-docs tomcat9-examples tomcat9-user
Se instalarán los siguientes paquetes NUEVOS:
  libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9
  tomcat9-common
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 370 no actualizados.
Se necesita descargar 12,1 MB de archivos.
Se utilizarán 14,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libeclipse-jdt-core-java all 3.15.0+eclipse4.9-1~18.04 [6.169 kB]
15% [1 libeclipse-jdt-core-java 2.201 kB/6.169 kB 36%]
```

Una vez hayamos instalado tomcat, nos dirigimos a nuestro navegador web y en la dirección de búsqueda escribimos "localhost:8080 " si todo ha ido bien, nos aparecerá una página indicándonos que la instalación ha ido correctamente pero además, nos indica que deberemos también descargar la documentación correspondiente a tomcat (admin, examples y docs) para poder configurarlo correctamente.

Para poder descargar la documentación, tenemos que ejecutar el comando

- `sudo apt install tomcat9-admin`
- `sudo apt install tomcat9-examples`
- `sudo apt install tomcat9-docs`

Una vez hayamos descargado la documentación, comprobamos el estado del tomcat para saber que está ejecutándose y lo hacemos mediante el comando "systemctl status tomcat9"y "sudo ufw allow 8080/tcp" (para que escuche por el puerto 80)

```
fran@fran-VirtualBox:~$ systemctl status tomcat9
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset:
   Active: active (running) since Tue 2021-02-23 13:16:04 CET; 4min 12s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Main PID: 7651 (java)
    Tasks: 34 (limit: 3556)
   CGroup: /system.slice/tomcat9.service
           └─7651 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.

feb 23 13:17:27 fran-VirtualBox tomcat9[7651]: Al menos un JAR, que se ha explor
feb 23 13:17:27 fran-VirtualBox tomcat9[7651]: Deployment of deployment descript
feb 23 13:18:17 fran-VirtualBox tomcat9[7651]: Desplieque del descriptor de conf
feb 23 13:18:17 fran-VirtualBox tomcat9[7651]: The path attribute with value [/e
feb 23 13:18:19 fran-VirtualBox tomcat9[7651]: Al menos un JAR, que se ha explor
feb 23 13:18:19 fran-VirtualBox tomcat9[7651]: Deployment of deployment descript
feb 23 13:18:50 fran-VirtualBox tomcat9[7651]: Desplieque del descriptor de conf
feb 23 13:18:50 fran-VirtualBox tomcat9[7651]: The path attribute with value [/d
feb 23 13:18:51 fran-VirtualBox tomcat9[7651]: Al menos un JAR, que se ha explor
feb 23 13:18:51 fran-VirtualBox tomcat9[7651]: Deployment of deployment descript
lines 1-19/19 (END)
```

Tercero: Ahora que ya tenemos tomcat instalado, hemos descargado su documentación y hemos comprobado que se está ejecutando, vamos a acceder a dichos documentos para modificarlos y poder acceder al servidor desde nuestro localhost.

El primer documento al que vamos a acceder, es el documento de los usuarios. Ejecutamos el comando "sudo nano /etc/tomcat9/tomcat-users.xml". Nos aparecerá el siguiente documento:

```
fran@fran-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/tomcat9/tomcat-users.xml

distributed under the license is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the license for the specific language governing permissions and
limitations under the license.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  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 o$
  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.
  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 o$
  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"/>
  -->
</tomcat-users>
```

Nos fijamos que en el documento aparece comentado los usuarios de tomcat. Esta parte del documento habrá que modificarla.

Lo que hacemos es descomentar los usuarios que vienen por defecto y en ellos, modificamos los roles y los usuarios según nuestra necesidad.

Para nosotros, nos interesa que haya 2 roles por ahora. El primer rol, será el de "manager-gui" y el

segundo rol, el de "admin-gui" y como usuarios, vamos a dejar uno que será "admin" con password "admin" y role "manager-gui,admin-gui". Insistiendo en que este usuario será el primero e iremos modificando más adelante si es preciso.

```
<role rolename="manager-gui">
<role rolename="admin-gui">
<user username="admin" password="admin" roles="manager-gui,admin-gui"/>
```

Ahora que ya tenemos nuestros usuarios modificados, vamos a parar y arrancar de nuevo el servidor tomcat. Para ello, ejecutamos los siguientes comandos:

- `service tomcat9 stop` (nos pedirá la contraseña de nuestra máquina)
- `service tomcat9 start` (nos volverá a solicitar la contraseña)

Una vez reiniciado el servidor, vamos a acceder a los siguientes documentos. Estos documentos serán por un lado, el **manager** y por otro, el **host-manager** con los siguientes comandos:

- `sudo nano /etc/share/tomcat9-admin/manager/META-INF/context.xml`
- `sudo nano /etc/share/tomcat9-admin/host-manager/META-INF/context.xml`

Ambos documentos deberán aparecer así

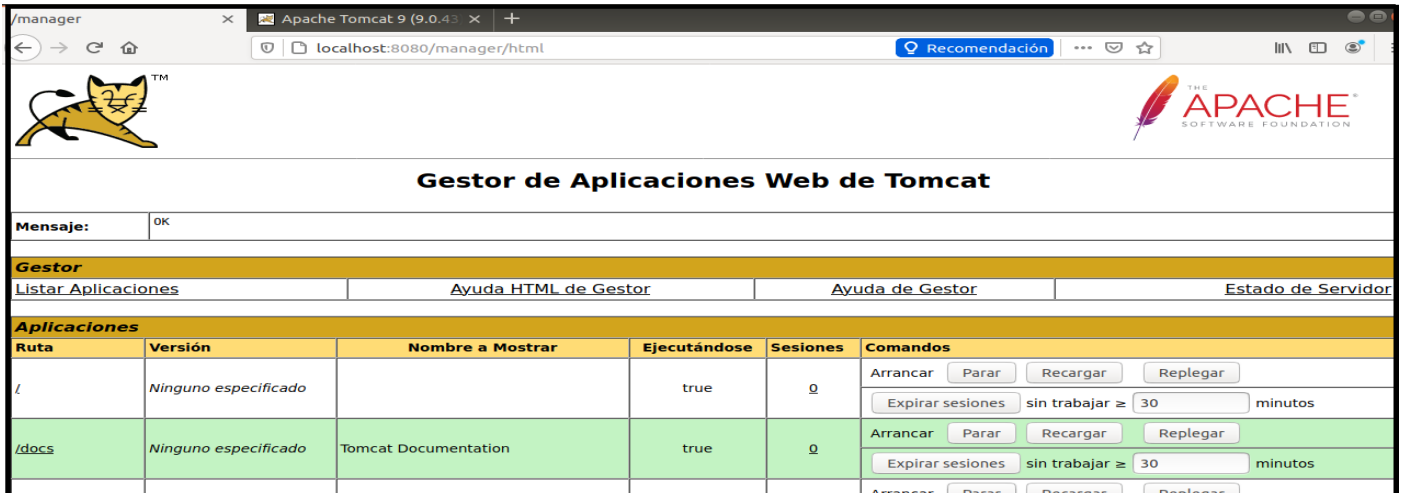
```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /opt/tomcat/webapps/manager/META-INF/context.xml

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the license for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcess$
    sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Intege$
</Context>
```

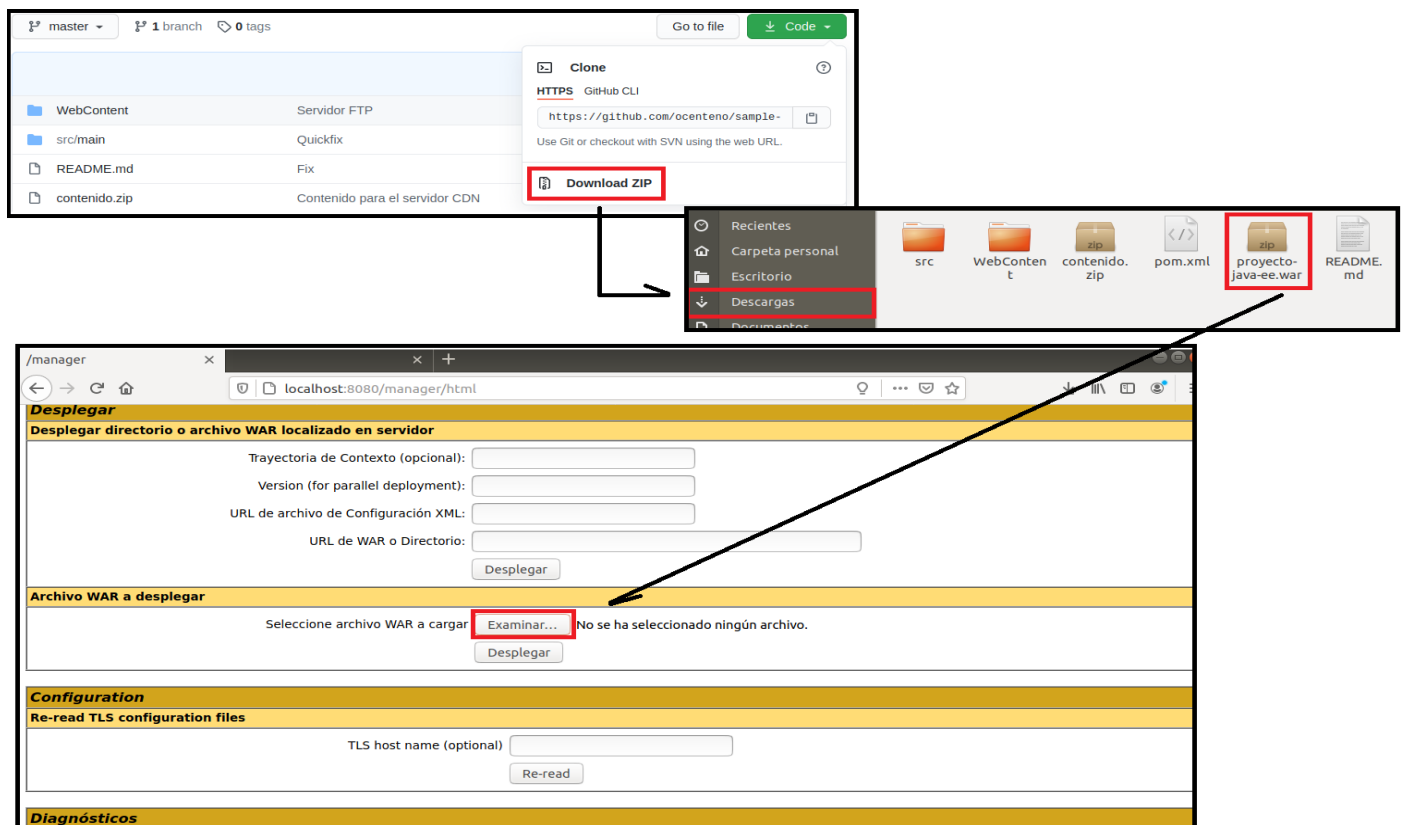

Cuando hayamos comprobado que ambos documentos aparecen iguales, salimos del “context.xml” y reiniciamos el tomcat con el comando:

- sudo systemctl restart tomcat9

Y una vez hayamos reiniciado el servidor, nos dirigimos nuevamente a nuestro navegador y escribimos en la dirección de búsqueda “localhost:8080/manager/html” y nos deberá aparecer la página principal del manager de tomcat.



Cuarto: Ya tenemos todo a punto para manejar tomcat. Ahora, lo que vamos a hacer es desplegar el archivo war que facilita el profesor en su proyecto git. Lo descargamos en nuestra máquina virtual y en el apartado de tomcat donde dice "desplegar archivo war" lo subimos.



Una vez hemos cargado el archivo, en el apartado "Aplicaciones" de nuestro tomcat, aparecerá. Pulsamos en el y nos lleva a la página de bienvenida. Si intentamos acceder a la Base de Datos, nos aparecerá en blanco porque aún no la hemos cargado.

The screenshot shows the Tomcat Manager interface. On the left, a table lists applications. The 'projecto-java-ee' application is highlighted with a red box. An arrow points from this box to a browser window on the right. The browser window shows the 'Bienvenido a la aplicación Web Java de ejemplo para JNDI, Apache y SQL' page. Another arrow points from the 'projecto-java-ee' application to a text box that says 'No nos sale nada, pues aún no hemos cargado BBDD'.

Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
/projecto-java-ee	Ninguno especificado	Proyecto Java EE	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos

Bienvenido a la aplicación Web Java de ejemplo para JNDI, Apache y SQL

Probar BD
Probar CDN
Probar FTP

No nos sale nada, pues aún no hemos cargado BBDD

Quinto: Llegados a éste punto, el siguiente paso será instalar nuestra base de datos. Para ello, abrimos nuestra terminal de Ubuntu. Antes de nada, **se recomienda** actualizar nuestra terminal por si hubiese algún fichero o algo sin actualizar, pero esto simplemente como consejo. Si se quiere actualizar nuestra terminal, se usará el comando "sudo apt-get update" y si aparecen ficheros listos para actualizar, ejecutaremos el comando sudo apt-get upgrade". En nuestro caso, decidimos que sí lo hacemos y la actualizamos.

Dicho esto, ahora sí, vamos a instalar nuestra BBDD. El equipo hemos deliberado qué opción puede ser mejor para trabajar. Hemos decidido, escoger la BBDD de MySQL ya que nos permite trabajar desde nuestra terminal, directamente por comandos y ahorrarnos el descargar mediante la interfaz de usuario otras, como puedan ser WorkBrench, Oracle, etc.

Para la instalación de MySQL, ejecutamos el siguiente comando:

- sudo apt install mysql-server

```
fran@fran-VirtualBox: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
fran@fran-VirtualBox:~$ sudo apt install mysql-server
```


Una vez hemos instalado MySQL, ahora vamos a dar una contraseña. Ejecutamos el comando:

- `sudo mysql_secure_installation`

Con este último comando, podemos acceder con nuestra contraseña del S.O. o cambiarla por otra que consideres oportuna.

El comando anterior nos irá preguntando modalidades como ¿desea utilizar su contraseña habitual? ¿quiere eliminar el usuario anónimo? etc.

```
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
fran@fran-VirtualBox:~$
```

Una vez hemos configurado las opciones que se muestran en la imagen anterior, ahora iniciamos MySQL con el comando

- `sudo mysql -u root -p`

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
fran@fran-VirtualBox:~$ sudo mysql -u root -p
[sudo] contraseña para fran:
[sudo] contraseña para fran:
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.33-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Cuando estamos dentro de MySQL, lo siguiente que vamos a hacer es crear la tabla con los datos que se adjuntan en el enunciado.

Para crear la tabla usamos el comando

- CREATE DATABASE nombreBaseDatos;
- show databases; (Para comprobar que la hemos creado correctamente)

```
fran@fran-VirtualBox:~$ sudo mysql -u root -p
[sudo] contraseña para fran:
Lo sentimos, vuelva a intentarlo.
[sudo] contraseña para fran:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.33-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE PERSONAS;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| PERSONAS |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.07 sec)

mysql>
```

Cuando ya hemos creado la BBDD, el siguiente paso es crear la tabla. Vamos a seguir el patrón del enunciado y creamos la tabla personas con los datos del DNI como primaryKey, el nombre y apellidos. Se ejecuta **todo el comando en la misma línea** el patrón que tenemos que seguir es primero "use table" y ejecutar el comando **CREATE TABLE** nombreTabla (**campo** tipoDato(caracteres) **primaryKey**) y todos los datos que queramos añadir separados entre comas

```
mysql> CREATE TABLE PERSONAS (DNI VARCHAR(8) DEFAULT '00000000' NOT NULL PRIMARY KEY, NOMBRE VARCHAR(20), APELLIDOS VARCHAR(20));
Query OK, 0 rows affected (0.83 sec)

mysql> show tables;
+-----+
| Tables_in_PERSONAS |
+-----+
| PERSONAS |
+-----+
1 row in set (0.00 sec)

mysql>
```

```
fran@fran-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
mysql> use proyecto;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_proyecto |
+-----+
| PERSONAS            |
+-----+
1 row in set (0.00 sec)

mysql> describe PERSONAS;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| DNI         | varchar(8)    | NO   | PRI | 00000000 |       |
| NOMBRE      | varchar(20)   | YES  |     | NULL    |       |
| APELLIDOS   | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Sexto: Cuando ya hemos creado las tablas, lo siguiente que tenemos que hacer es acceder al **context** para cambiar el recurso que se indica en el enunciado. Para ello, usamos el comando:

- `sudo nano /var/lib/tomcat9/conf/context.xml`

Una vez accedemos al archivo del context, vamos a añadir el recurso que se especifica. Una vez hemos accedido, procedemos a modificar el contenido como se indica en los apuntes

```
GNU nano 2.9.3 /var/lib/tomcat9/conf/context.xml
<!-- The contents of this file will be loaded for each web application -->
<Context>

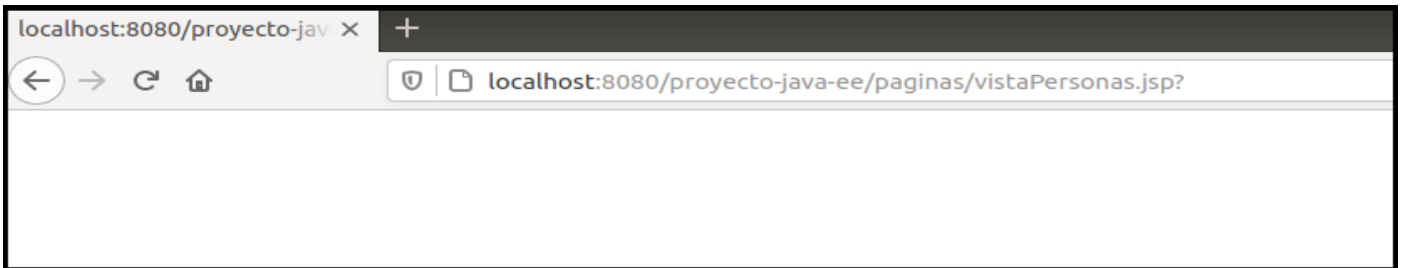
  <Resource name="jdbc/PoolBaseDatos" auth="Container" type="java.sql.DataSource"
    maxTotal="100" maxIdle="30" maxWaitMillis="10000"
    username="root" password="actividad2021"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/PERSONAS"/>

  <!-- Default set of monitored resources. If one of these changes, the
  <!-- web application will be reloaded.

  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
  <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>-->

  <!-- Uncomment this to disable session persistence across Tomcat restarts -->
  <!--
  <Manager pathname="" />
  -->
</Context>
```

Guardamos el archivo pulsando **^x** y volvemos a restablecer el tomcat con el comando "systemctl tomcat9 restart". Una vez lo hayamos reseteado, volvemos a acceder a la BBDD



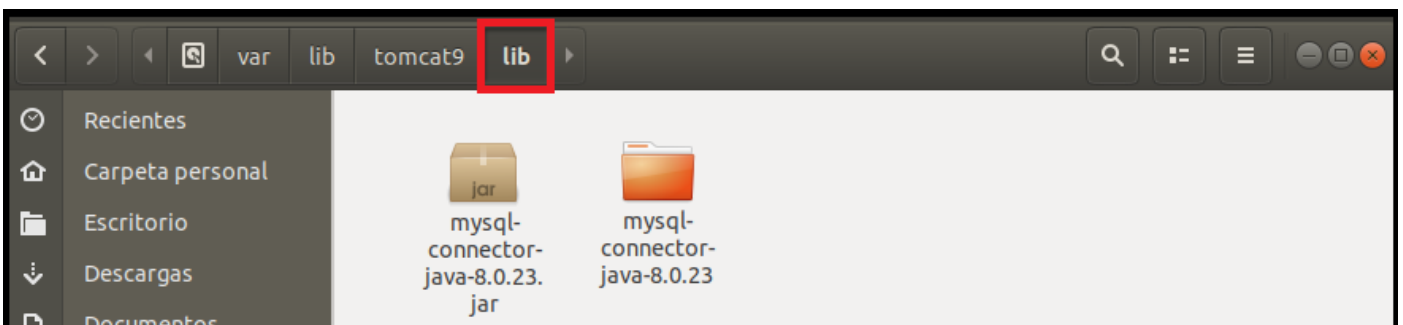
Séptimo: Nos encontramos con el problema de que no podemos acceder. Para solucionar este inconveniente, tenemos que descargar el conector de mysql. Para ello, nos dirigimos a: <https://mvnrepository.com/artifact/mysql/mysql-connector-java> y descargamos el que sea acorde a nuestro mysql.

Conector MySQL / J
Controlador JDBC Tipo 4 para MySQL

Licencia: GPL 2.0
Categorías: Controladores MySQL
Etiquetas: mysql base de datos conector conductor
Usado por: 5.106 artefactos

Versión	Repositorio	Usos	Fecha
8.0.23	Central	40	Enero de 2021
8.0.22	Central	159	Oct, 2020

Una vez lo hayamos descargado, tenemos que llevar el archivo a la librería del tomcat. Cuando lo hayamos pegado en dicha carpeta, procedemos a extraerlo en la misma.



Nota: Si vemos que no conecta la BBDD, tenemos que volver a resetear el servidor y comprobar que tenemos los puertos abiertos en el firewall

```
fran@fran-VirtualBox: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
fran@fran-VirtualBox:~$ sudo ufw enable  
[sudo] contraseña para fran:  
El cortafuegos está activo y habilitado en el arranque del sistema  
fran@fran-VirtualBox:~$ sudo ufw status  
Estado: activo  
  
Hasta          Acción      Desde  
-----  
Apache         ALLOW      Anywhere  
8080/tcp       ALLOW      Anywhere  
3306/tcp       ALLOW      Anywhere  
22/tcp         ALLOW      Anywhere  
443/tcp        ALLOW      Anywhere  
Apache (v6)    ALLOW      Anywhere (v6)  
8080/tcp (v6)  ALLOW      Anywhere (v6)  
3306/tcp (v6)  ALLOW      Anywhere (v6)  
22/tcp (v6)    ALLOW      Anywhere (v6)  
443/tcp (v6)   ALLOW      Anywhere (v6)
```

Octavo: Volvemos a acceder a la BBDD y vemos como ya sí que se visualiza correctamente

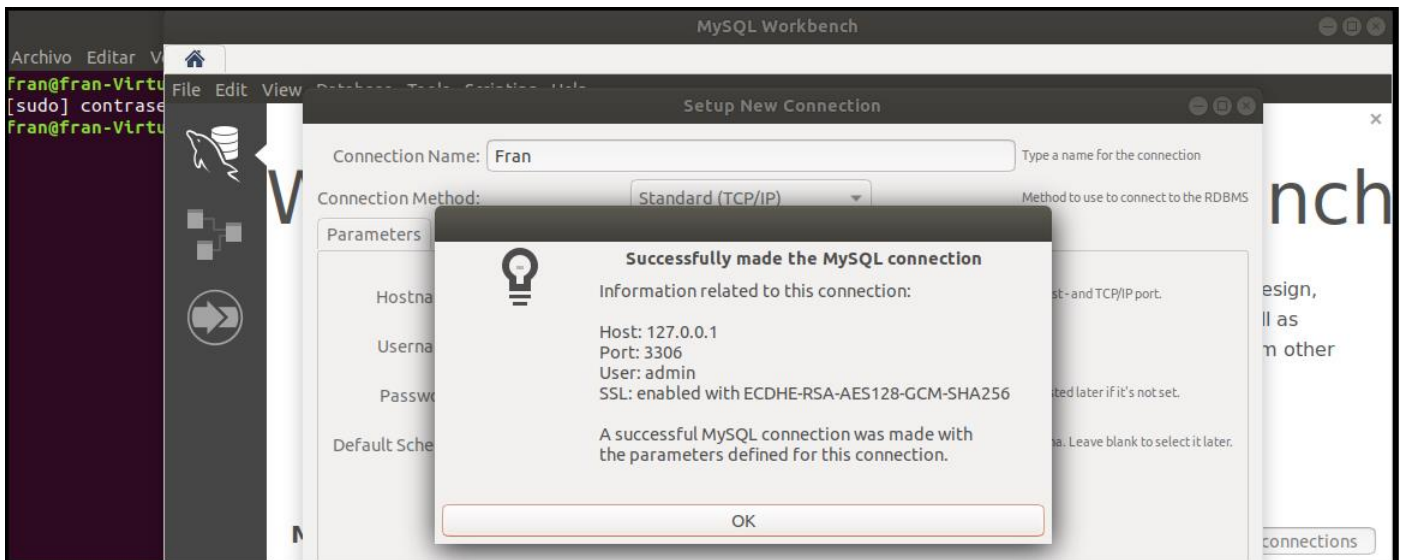


Por último, podemos insertar datos bien utilizando el entorno gráfico que nos aparece en el localhost o bien por comandos. El uso del comando se procesa de la siguiente estructura:

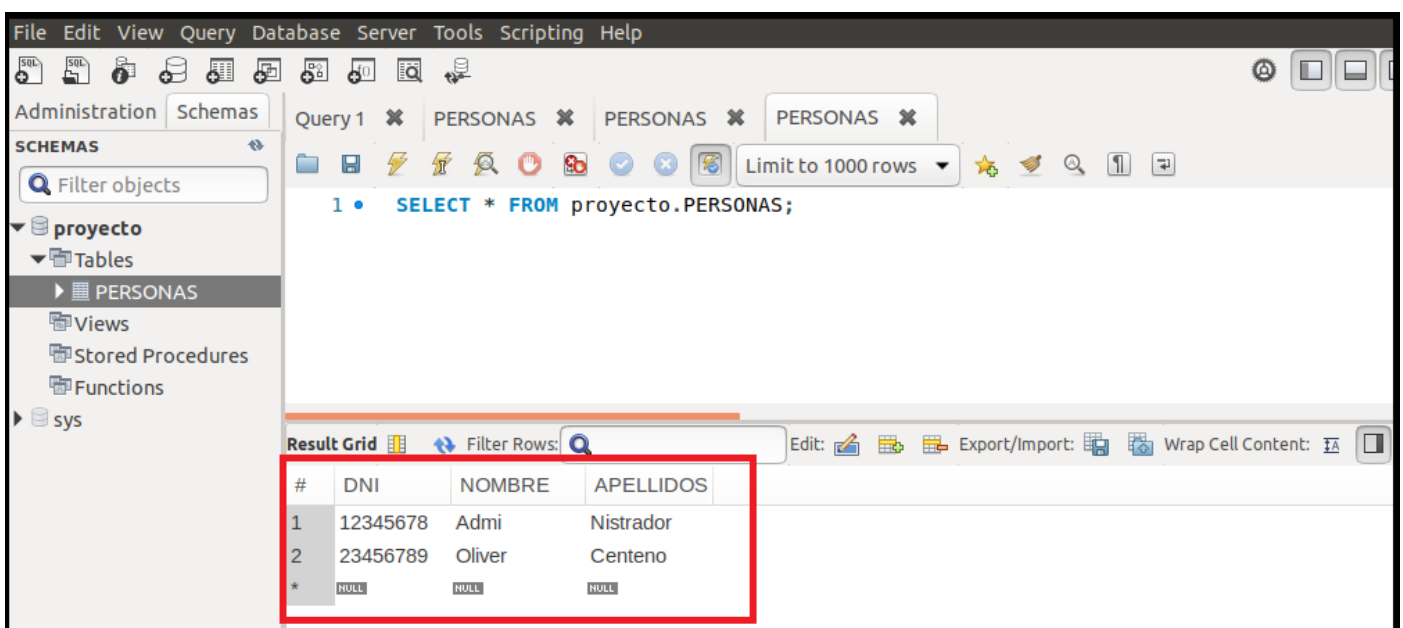
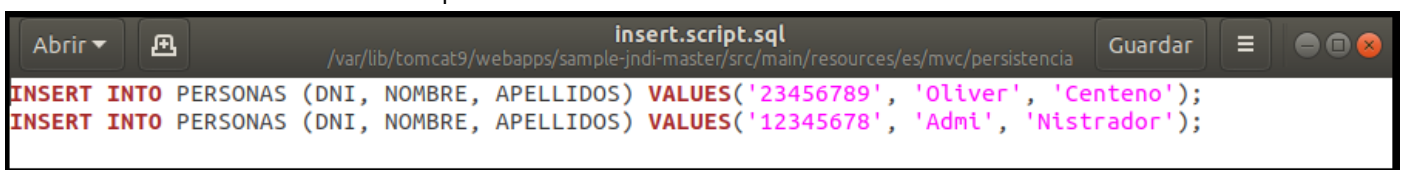
Insert into Table (atributo) values (valor);

```
mysql> insert into PERSONAS (NOMBRE) values ('Francisco');  
Query OK, 1 row affected (0.19 sec)
```

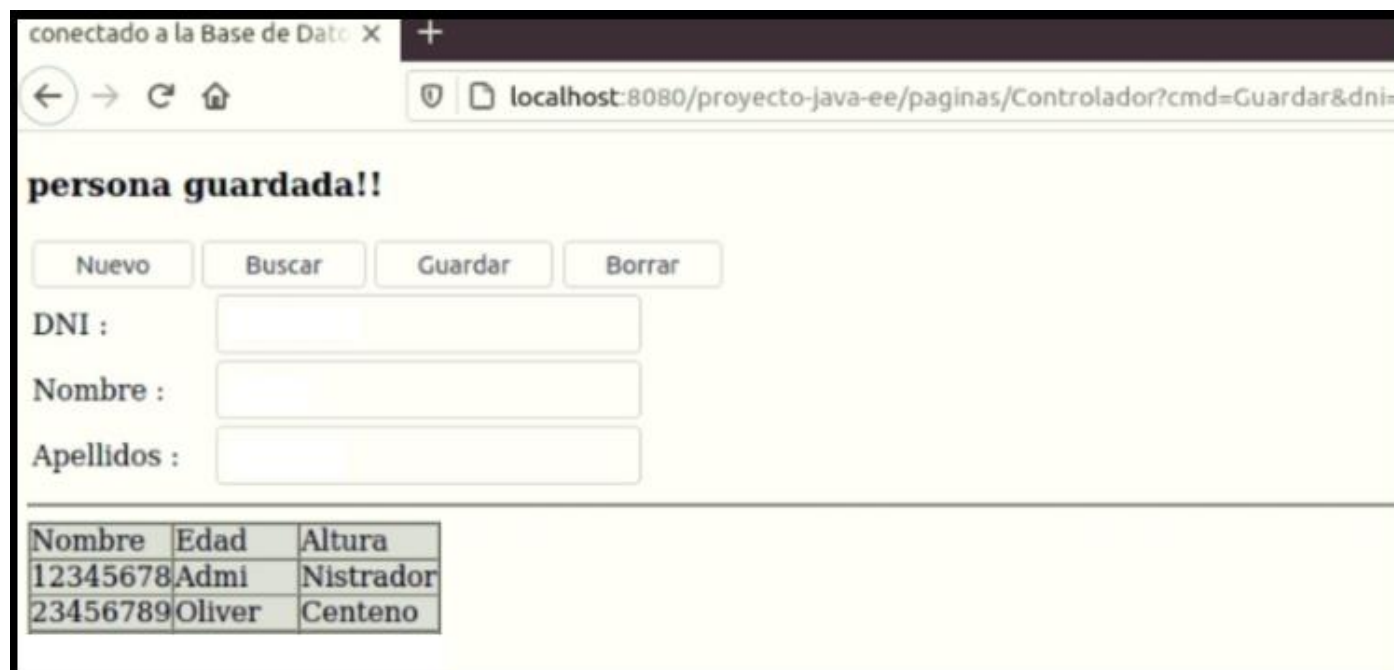
Nota: Podemos hacerlo mediante entorno gráfico también a través de workbench (o cualquier otra base de datos) primero tenemos que asegurarnos que establece la conexión perfectamente usando nuestro usuario y contraseña:



Noveno: Insertamos los campos como se indica en el enunciado



Finalmente, volvemos a cargar la página de la BBDD y comprobamos que se han insertado correctamente los campos:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/proyecto-java-ee/paginas/Controlador?cmd=Guardar&dni=...'. The page content includes a success message 'persona guardada!!' and four buttons: 'Nuevo', 'Buscar', 'Guardar', and 'Borrar'. Below the buttons are three input fields labeled 'DNI:', 'Nombre:', and 'Apellidos:'. At the bottom, there is a table with three columns: 'Nombre', 'Edad', and 'Altura'.

Nombre	Edad	Altura
12345678Admi		Nistrador
23456789Oliver		Centeno

2.- Apache CDN

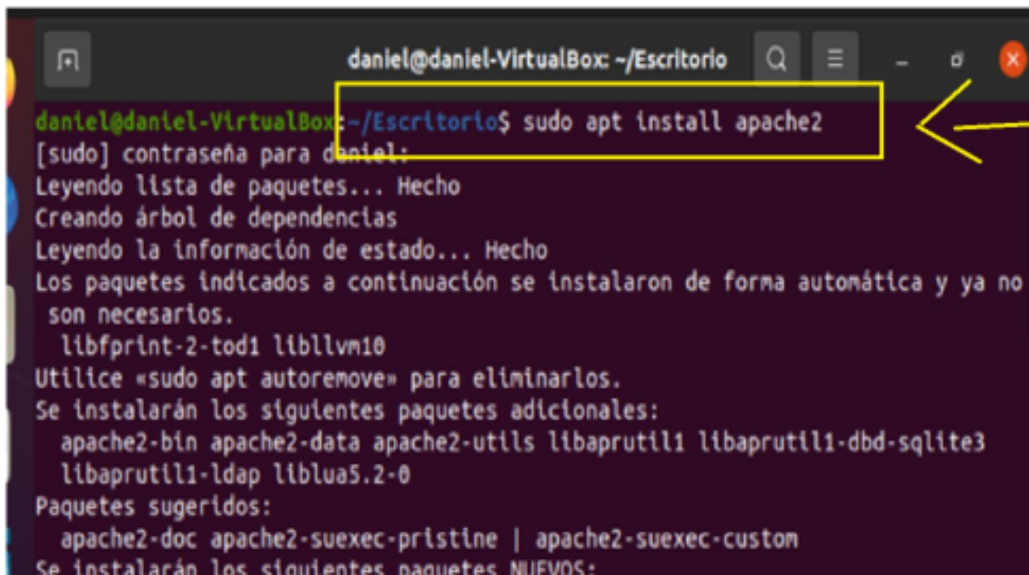
CDN es una forma habitual de separar el contenido estático de una aplicación web. Consiste en ubicarlo en un servidor virtual que se puede replicar para que cada usuario descargue ese contenido del servidor más cercano a su ubicación. La aplicación espera que exista un recurso "jndi/CDN" que apunte a la dirección web donde se han subido los ficheros estáticos; se deberá crear ese servidor virtual y el recurso correspondiente en el Servidor de Aplicaciones.

2. Contenido

Este fichero "contenido.zip" contiene

- Una imagen que deberá desplegarse en el servidor Apache
- Un video que deberá desplegarse en el servidor Apache
- El propio fichero contenido.zip deberá desplegarse en el servidor Apache

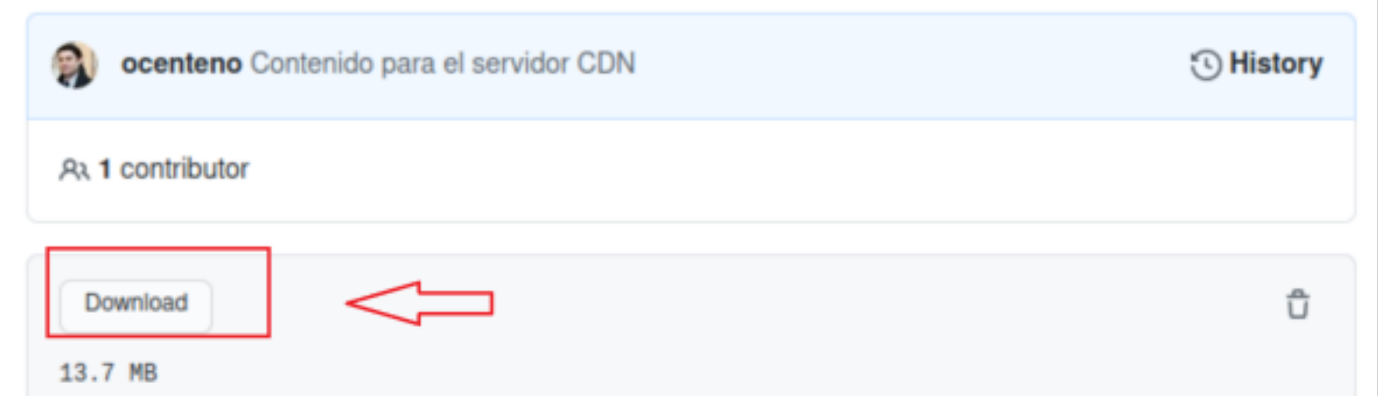
Lo primero que tenemos que hacer es descargarnos apache :



```
daniel@daniel-VirtualBox: ~/Escritorio
daniel@daniel-VirtualBox:~/Escritorio$ sudo apt install apache2
[sudo] contraseña para daniel:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libfprint-2-tod1 libllvm10
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
```

Para ellos hemos utilizado el comando “sudo apt install apache 2”.
Luego nos metemos en el github de la actividad para descargarnos tanto el war. Como el zip con el contenido CDN.

[sample-jndi](#) / [contenido.zip](#)



ocenteno Contenido para el servidor CDN History

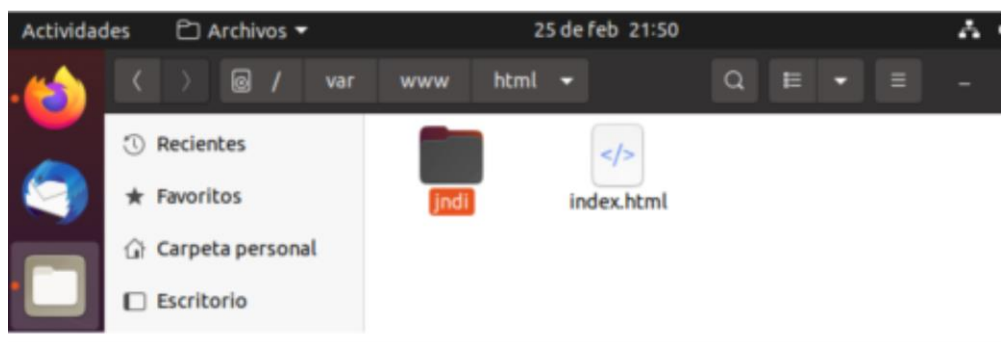
1 contributor

Download 13.7 MB

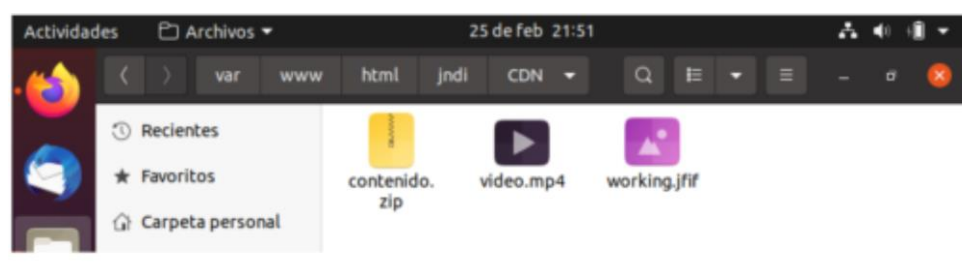
Para conectar el tomcat con el apache iremos a la carpeta “var/lib/tomcat9” y nos metemos en el context.xml , añadiendo al mismo el Environment con un name, value y type cambiados

```
9
10 http://www.apache.org/licenses/LICENSE-2.0
11
12 Unless required by applicable law or agreed to in writing, software
13 distributed under the License is distributed on an "AS IS" BASIS,
14 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 See the License for the specific language governing permissions and
16 limitations under the License.
17 -->
18 <!-- The contents of this file will be loaded for each web application -->
19 <Context>
20
21 <!-- Default set of monitored resources. If one of these changes,
22 the -->
23 <!-- web application will be
24 reloaded. -->
25 <WatchedResource>WEB-INF/web.xml</WatchedResource>
26 <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
27 <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
28 <!-- Uncomment this to disable session persistence across Tomcat
29 restarts -->
30 <!--
31 <Manager pathname="" />
32 -->
33 <Environment name="jndi/CDN" value="http://localhost:80/jndi/CDN"
34 type="java.lang.String" override="false"/>
35 </Context>
```

Para alojar el zip descargados nos vamos a la carpeta "var/www/html"



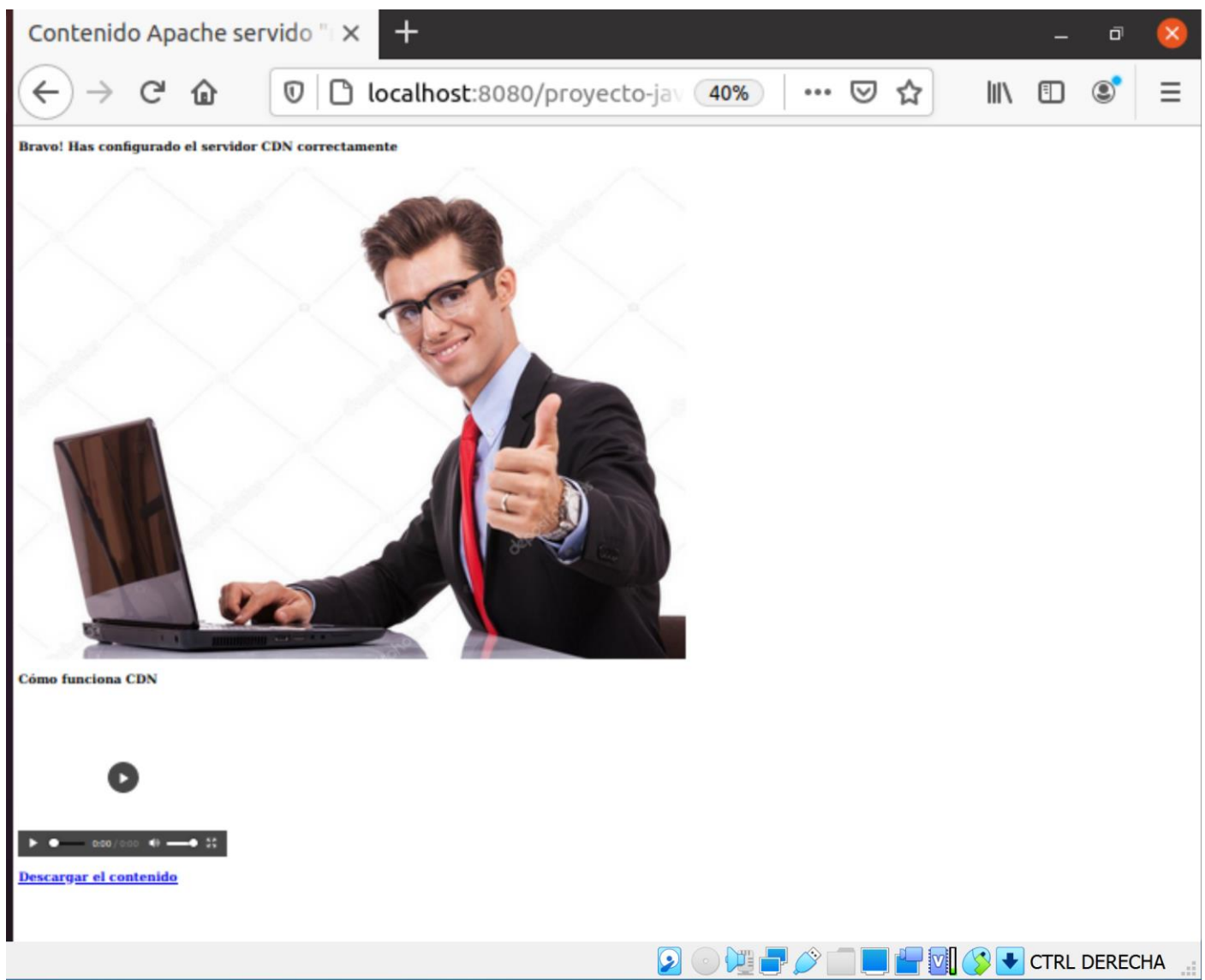
Y pegamos el contenido:



Despues de hacer estos pasos probamos que todo este funcionando,



Y vemos el resultado:



3.- FTP

3. FTP

Se deberá crear un recurso "jndi/FTP" que contendrá la URL del servidor FTP con el que trabajar. El servidor utiliza 2 usuarios con sus respectivas contraseñas:

- registrado: registrado para poder leer y obtener ficheros del servidor, debe tener permisos de lectura
- administrador: administrador para poder subir ficheros al servidor, debe tener permisos de escritura. La aplicación trabaja únicamente con el directorio raíz del servidor, no se implementará la descarga o la subida en otra carpeta

Para la configuración del servidor FTP vamos a utilizar vsftpd. En primer lugar, lo instalamos mediante "apt install":

```
victor@ubuntu:~$ sudo apt install vsftpd
[sudo] contraseña para victor:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  vsftpd
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 115 kB de archivos.
Se utilizarán 338 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 vsftpd amd64 3.0.3-12
[115 kB]
Descargados 115 kB en 0s (269 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete vsftpd previamente no seleccionado.
(Leyendo la base de datos ... 197841 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../vsftpd_3.0.3-12_amd64.deb ...
Desempaquetando vsftpd (3.0.3-12) ...
Configurando vsftpd (3.0.3-12) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /li
b/systemd/system/vsftpd.service.
vsftpd.conf:1: Line references path below legacy directory /var/run/, updating /
var/run/vsftpd/empty → /run/vsftpd/empty; please update the tmpfiles.d/ drop-in
```

Una vez instalado, aseguramos que tenemos abierto el puerto 21, por el que se ejecutará nuestro servidor:

```
victor@ubuntu:/var/www/html$ sudo ufw status
Estado: activo

Hasta      Acción      Desde
-----
8080        ALLOW        Anywhere
8080/tcp    ALLOW        Anywhere
3306/tcp    ALLOW        Anywhere
443/tcp     ALLOW        Anywhere
Apache      ALLOW        Anywhere
22/tcp      ALLOW        Anywhere
21/tcp      ALLOW        Anywhere
20/tcp      ALLOW        Anywhere
8080 (v6)   ALLOW        Anywhere (v6)
8080/tcp (v6) ALLOW        Anywhere (v6)
3306/tcp (v6) ALLOW        Anywhere (v6)
443/tcp (v6) ALLOW        Anywhere (v6)
Apache (v6) ALLOW        Anywhere (v6)
22/tcp (v6) ALLOW        Anywhere (v6)
21/tcp (v6) ALLOW        Anywhere (v6)
20/tcp (v6) ALLOW        Anywhere (v6)
```

configuramos el servidor FTP para que cumpla con lo solicitado en la práctica. Accedemos al fichero vsftpd.conf mediante el comando nano:

```
GNU nano 4.8          vsftpd.conf
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
```

Por un lado, “descomentaremos” las opciones vistas en clase para que los usuarios solo puedan compartir vía FTP su directorio /home/:

```
# chroot)
chroot_local_user=YES
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list
#
```

Cabe destacar que para permitir que el usuario “administrador” pueda tener permiso de escritura, se debió de incorporar la instrucción “allow_writeable_chroot=YES”, en caso contrario el servidor no dejaba logarse con dicho usuario:

```
chroot_local_user=YES
allow_writeable_chroot=YES
```

Crearemos el usuario “registrado”, que incluiremos en la lista antes mencionada:

```
victor@ubuntu:/var/lib/tomcat9/conf$ sudo adduser registrado
[sudo] contraseña para victor:
Añadiendo el usuario `registrado' ...
Añadiendo el nuevo grupo `registrado' (1001) ...
Añadiendo el nuevo usuario `registrado' (1001) con grupo `registrado' ...
Creando el directorio personal `/home/registrado' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiano la información de usuario para registrado
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []:
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
```


Creamos el fichero:

```
victor@ubuntu:/etc$ sudo touch vsftpd.chroot_list
```

Introducimos al usuario "registrado":

```
GNU nano 4.8 vsftpd.chroot_list
registrado
```

Generamos el usuario "administrador" del mismo modo, así como lo incorporamos a la lista anterior, y añadimos el siguiente comando para que su carpeta home corresponda con la que se compartirá vía FTP:

```
victor@ubuntu:~$ sudo usermod -d /var/www/html/jndi/FTP administrador
```

Posteriormente se crea un grupo "FTP", donde introduciremos ambos usuarios; de este modo el usuario "registrado" tendrá acceso de lectura y ejecución, pero no de escritura. Asimismo, nos aseguramos que la carpeta a compartir tiene asignado el grupo creado:

```
victor@ubuntu:~$ sudo addgroup ftp
victor@ubuntu:~$ sudo usermod administrador -G ftp
victor@ubuntu:~$ sudo usermod registrado -G ftp
victor@ubuntu:~$ sudo chgrp -R ftp /var/www/html/jndi/FTP
```

Generamos la variable en el fichero context.xml. Es interesante observar que, si se introducía el puerto o el usuario, la aplicación no funciona correctamente:

```
<Environment name="jndi/CDN" value="http://localhost:80/jndi/CDN"
type="java.lang.String" override="false" />
<Environment name="jndi/FTP"
value="localhost"
type="java.lang.String" override="false" />
```

Tras esto, el servidor FTP ya habría quedado configurado para su uso con la aplicación facilitada, como último paso, introducimos los recursos requeridos dentro del directorio.

NOTA IMPORTANTE: observando el código del .war adjunto, hemos determinado que este predefinía que el usuario debe es siempre "registrado", motivo por el cual no permite adjuntar archivos desde la aplicación (ya que ese usuario ha sido configurado conforme no dispone de permisos de escritura):

←

→

🏠

localhost:8080/proyecto-java-ee/paginas/Ficheros

Salida FTP

Subiendo fichero a localhost220 (vsFTPD 3.0.3)
230 Login successful.
Subiendo null220 (vsFTPD 3.0.3)
230 Login successful.

Indica el parámetro para el comando que quieres ejecutar :

CD

Get

Selecciona un fichero para subir y el nombre que quieres darle :

Put

Examinar...

 No se ha seleccionado ningún archivo.

Contenido del servidor FTP :

Tipo	Nombre	Tamaño	Fecha de creación
f	contenido.zip	14387322 bytes	Mon Mar 01 20:49:00 CET 2021

Finalmente, hemos logrado que los ficheros de configuración de la propia aplicación web se encuentren disponibles en la carpeta habilitada en el servidor FTP (faltando únicamente este manual que será incorporado a su finalización:

Salida FTP

220 (vsFTPD 3.0.3)
230 Login successful.

Indica el parámetro para el comando que quieres ejecutar :

Selecciona un fichero para subir y el nombre que quieres darle : No se ha seleccionado ningún archivo.

Contenido del servidor FTP :

Tipo	Nombre	Tamaño	Fecha de creación
f	archivos_configuracion.zip	5074 bytes	Tue Mar 02 11:50:00 CET 2021
f	context.xml	1916 bytes	Tue Mar 02 11:51:00 CET 2021
f	tomcat-users.xml	2021 bytes	Tue Mar 02 11:36:00 CET 2021
f	vsftpd.chroot_list	25 bytes	Tue Mar 02 11:40:00 CET 2021
f	vsftpd.conf	5874 bytes	Tue Mar 02 11:39:00 CET 2021

Si solicitamos un fichero, este aparece en el navegador (en este caso, el listado de usuarios que han sido limitados a su carpeta home):

Salida FTP

220 (vsFTPD 3.0.3)
230 Login successful.

Indica el parámetro para el comando que quieres ejecutar :

Selecciona un fichero para subir y el nombre que quieres darle : No se ha seleccionado ningún archivo.

Contenido del servidor FTP :

Tipo	Nombre	Tamaño	Fecha de creación
f	archivos_configuracion.zip	5074 bytes	Tue Mar 02 11:50:00 CET 2021
f	context.xml	1916 bytes	Tue Mar 02 11:51:00 CET 2021
f	tomcat-users.xml	2021 bytes	Tue Mar 02 11:36:00 CET 2021
f	vsftpd.chroot_list	25 bytes	Tue Mar 02 11:40:00 CET 2021
f	vsftpd.conf	5874 bytes	Tue Mar 02 11:39:00 CET 2021

localhost:8080/proyecto-java-ee/paginas/Ficheros?cmd=Get¶metro=vsftpd.chroot_list

registrado
administrador