22-2-2021

[Título del documento]

ACTIVIDAD 3

David Losa

instituto tecnologico telefonica

Contenido

[BBDD 2](#_Toc65402691)

[Instalar MySQL 2](#_Toc65402692)

[Configurar MySQL 2](#_Toc65402693)

[Crear usuario 3](#_Toc65402694)

[Crear BBDD 3](#_Toc65402695)

[Descargar el conector 4](#_Toc65402696)

[Modificar “context.xml” 5](#_Toc65402697)

[Revisión del proyecto 6](#_Toc65402698)

# GIT

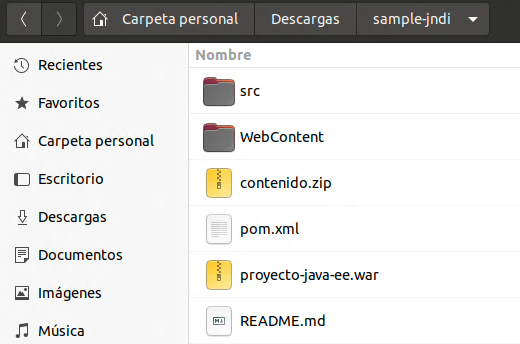
## Descargar proyecto

Abrimos un terminal y nos dirigimos al directorio de “Descargas”

Usamos el siguiente comando

* git clone <https://github.com/ocenteno/sample-jndi.git>

Veremos como se ha descargado el repositorio



## Desplegar proyecto

Ahora en la terminal nos dirigimos al directorio de “webapps” de tomcat y escribimos el siguiente comando

* cp /home/David/Descargas/sample-jndi/proyecto-java-ee.war .

Veremos como el .war se lleva a ese directorio y si dejamos pasar un momento, veremos como se despliega automáticamente (ya que tomcat, en ese directorio si ve algún .war, lo desplegará automáticamente)

Una vez descargado, renombramos el directorio, en mi caso se ha renombrado a “proyecto”

**Reiniciamos Tomcat**

# BBDD

## Instalar MySQL

En Ubuntu 20.04, podemos instalar MySQL usando el repositorio de paquetes APT.

Para ello seguiremos los siguientes pasos:

* Abrimos la terminal de Ubuntu y escribimos

**sudo apt update**

* A continuación, instalamos el paquete mysql-server

**sudo apt install mysql-server**

## Configurar MySQL

Para configurar MySQL comenzaremos por el siguiente comando

**sudo mysql\_secure\_installation**

* Se mostrarán los siguientes mensajes e indicaremos la respuesta abajo descrita en color rojo

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords and improve security. It checks the strength of password and allows the users to set only those passwords which are secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: **Y**

There are three levels of password validation policy:

LOW Length >= 8  
MEDIUM Length >= 8, numeric, mixed case, and special characters  
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: **2**

* Acto seguido crearemos una password

**Please set the password for root here.**

**New password:**

**Re-enter new password:**

* Una vez introducida, podemos pulsar Y y luego ENTER para aceptar los valores predeterminados para todas las preguntas siguientes.

## Crear usuario

* Accedemos a “mysql” con el siguiente comando

**sudo mysql**

* Creamos un usuario con el siguiente comando

**CREATE USER 'test'@'localhost' IDENTIFIED BY 'wmHTy53-cie';**

* Ahora concedemos los privilegios necesarios al usuario

**GRANT ALL PRIVILEGES ON \*.\* TO 'test'@'localhost' WITH GRANT OPTION;**

* Si realizamos la siguiente select, deberemos ver a nuestro user creado

**SELECT user,authentication\_string,plugin,host FROM mysql.user;**

* Salimos de mysql con el comando

**Exit**

## Crear BBDD

* Accedemos a “mysql” con el siguiente comando

**sudo mysql**

* Creamos una BBDD conel siguiente nombre

**CREATE DATABASE PROYECTO;**

* Ahora debemos usarla con el siguiente comando

**USE PROYECTO;**

* Ahora debemos crear la tabla pertinente para la actividad

**CREATE TABLE PERSONAS (**

**DNI VARCHAR(8) DEFAULT '00000000' NOT NULL,**

**NOMBRE VARCHAR(20),**

**APELLIDOS VARCHAR(20),**

**PRIMARY KEY (dni)**

**);**

## Descargar el conector

Accedemos a la siguiente URL para descargar e conector. En este caso descargamos el 8.0.23 ya que coincide con la versión de MySQL

[Maven Repository: mysql » mysql-connector-java » 8.0.23 (mvnrepository.com)](https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.23)

Una vez descargado debemos llevarlo al directorio de nuestro proyecto

* **WEB-INF\lib\** mysql-connector-java-8.0.23.jar

## Modificar “context.xml”

Nos dirigimos al directorio donde tengamos instalado tomcat

* **/conf/context**

Abrimos el fichero **context.xml** y escribimos lo siguiente:

<Resource name="jdbc/poolBaseDatos"

auth="Container"

type="javax.sql.DataSource"

factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"

testWhileIdle="true"

testOnBorrow="true"

testOnReturn="false"

validationQuery="SELECT 1"

validationInterval="30000"

timeBetweenEvictionRunsMillis="30000"

maxActive="20"

minIdle="2"

maxWait="10000"

initialSize="10"

removeAbandonedTimeout="60"

removeAbandoned="true"

logAbandoned="true"

minEvictableIdleTimeMillis="30000"

jmxEnabled="true"

username="test"

password="wmHTy53-cie"

driverClassName="com.mysql.cj.jdbc.Driver"

url="jdbc:mysql://localhost:3306/PROYECTO"

/>

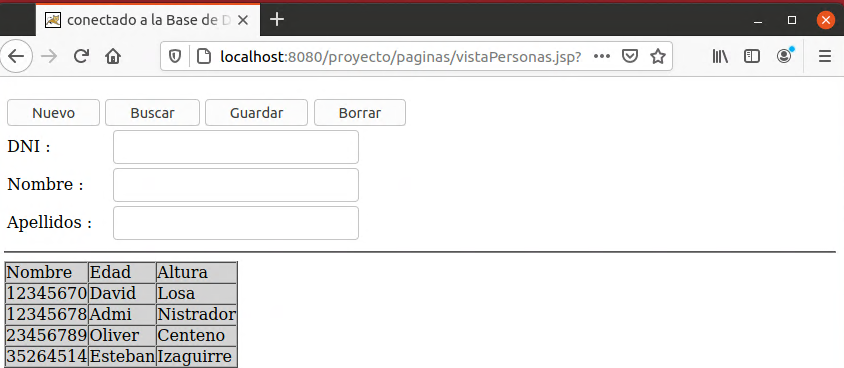
**En este punto Guardamos el fichero y reiniciamos Tomcat**

## Revisión del proyecto

Accedemos a la URL del proyecto

* <http://localhost:8080/proyecto>





# CDN

## Instalación

Abrimos un terminal y escribimos el siguiente comando

* **sudo apt install apache2**

En la siguiente ruta del sistema creamos un directorio desde la consola, llamado por ejemplo, CDN

* **/var/www/html** - CDN

Reiniciamos con el comando

* **sudo /etc/init.d/apache2 restart**

## Modificar “context.xml”

Nos dirigimos al directorio donde tengamos instalado tomcat

* **/conf/context**

Abrimos el fichero **context.xml** y escribimos lo siguiente:

**<Environment name="jndi/CDN"**

**value="http://localhost:80/CDN"**

**type="java.lang.String"**

**override="false"/>**

En el directorio CDN que hemos creado, debemos descargar todo el contenido que nos ha proporcionado el profesor para la actividad

Reiniciamos con el comando

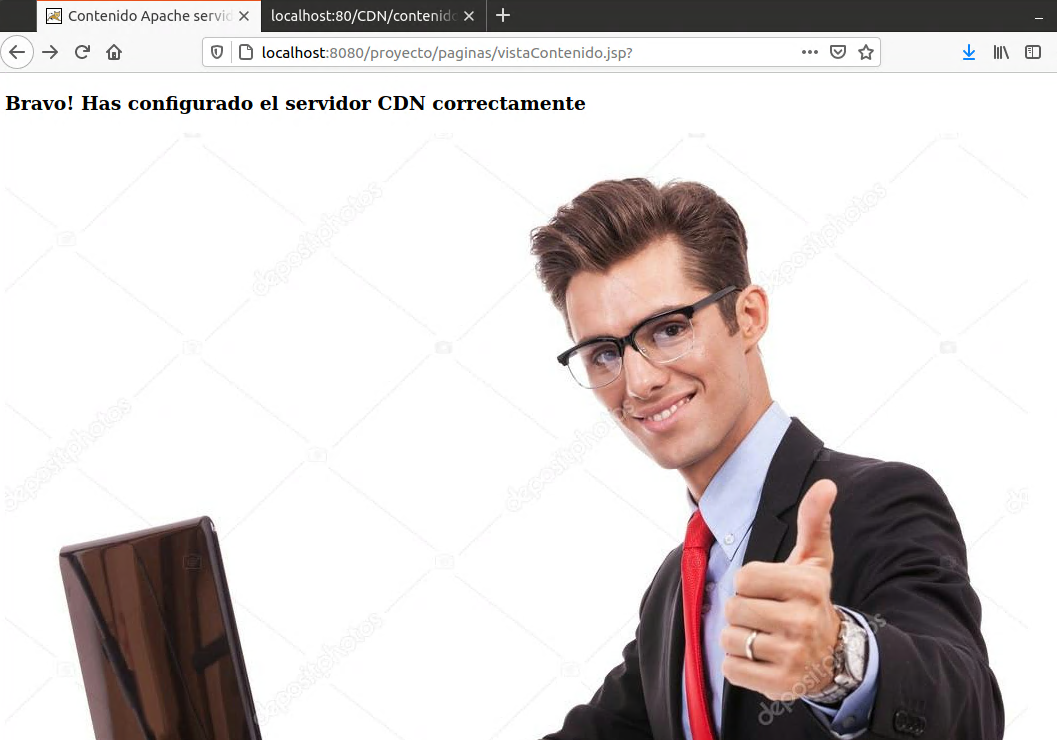
* **sudo /etc/init.d/apache2 restart**

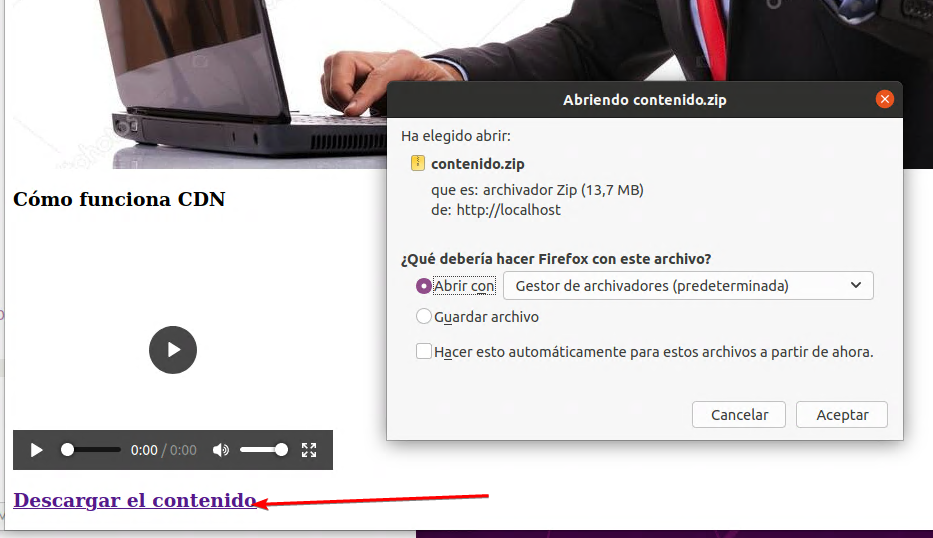
## Revisión del proyecto

Accedemos a la URL del proyecto

* <http://localhost:8080/proyecto>







# FTP

Para crear el servidor ftp usamos el comando

**sudo apt-get install vsftpd**

Despues creamos las cuentas del administrador y del usuario registrado con

**sudo adduser administrador**

y

**sudo adduser registrado**

Tras ejecutar el comando adduser se nos pedira una contraseña para el usuario, en este caso las contraeñas serán las mismas que los nombres de usuario, administrador para administrador y registrado para registrado.

Creamos un directorio donde almcenar los ficheros de nuestro servidor FTP:

**sudo mkdir var/www/html/FTP**

Y convertimos los directorios “home” de los usuario del Ftp en esta carpeta:

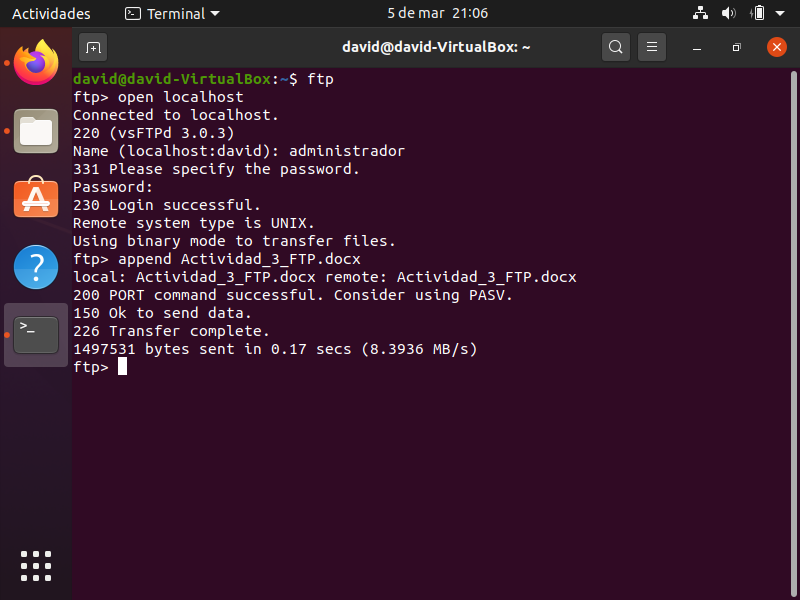
**sudo usermod -d /var/www/html/FTP administrador**

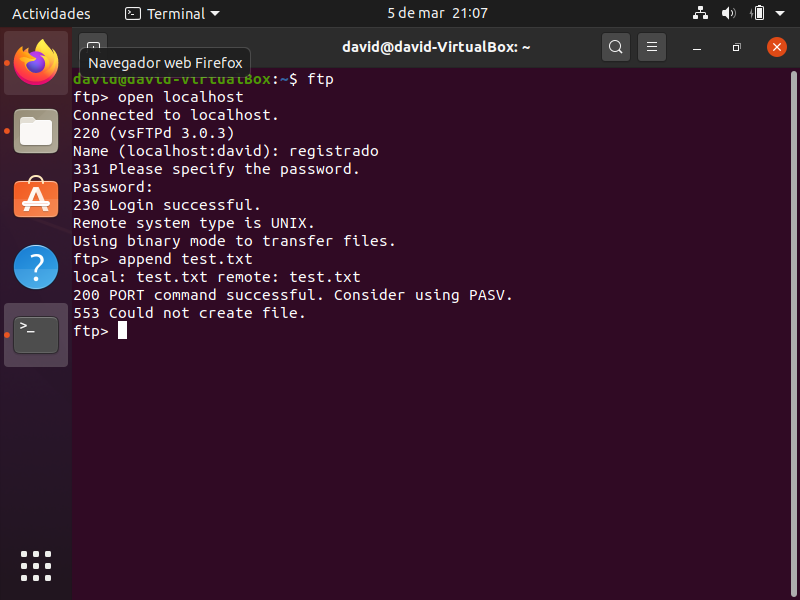
**sudo usermod -d /var/www/html/FTP registrado**

Y para que el administrador tenga permisos de escritura en el directorio:

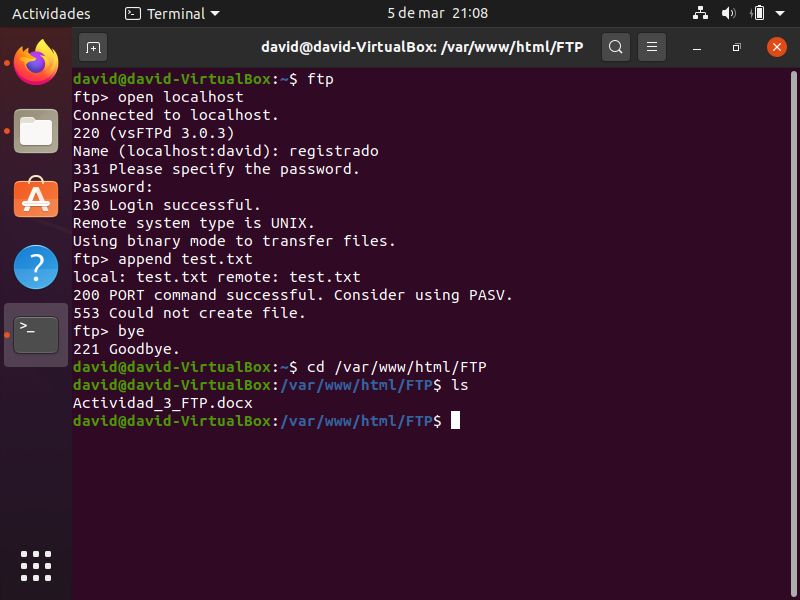
**sudo chown -R administrador /var/www/html/FTP/**

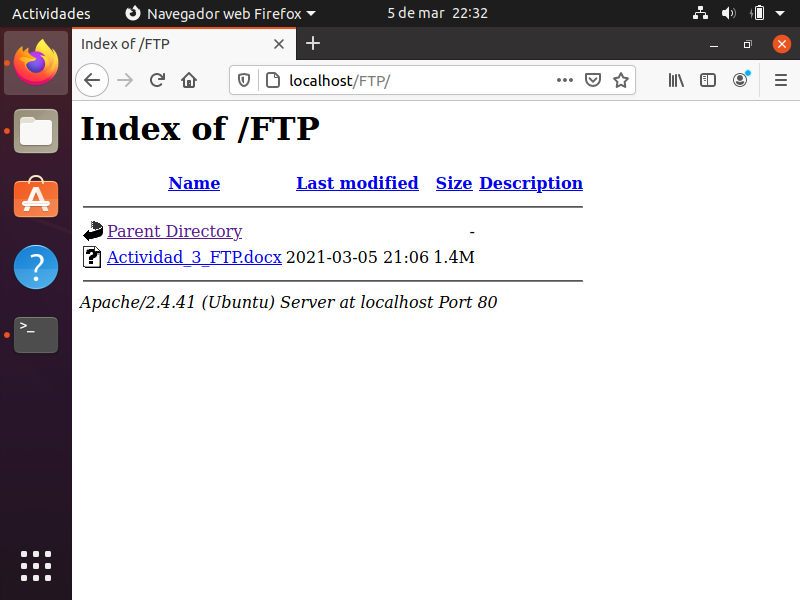
Esto nos permitira acceder a el servidor FTP usando estos usuarios mediante un cliente FTP:

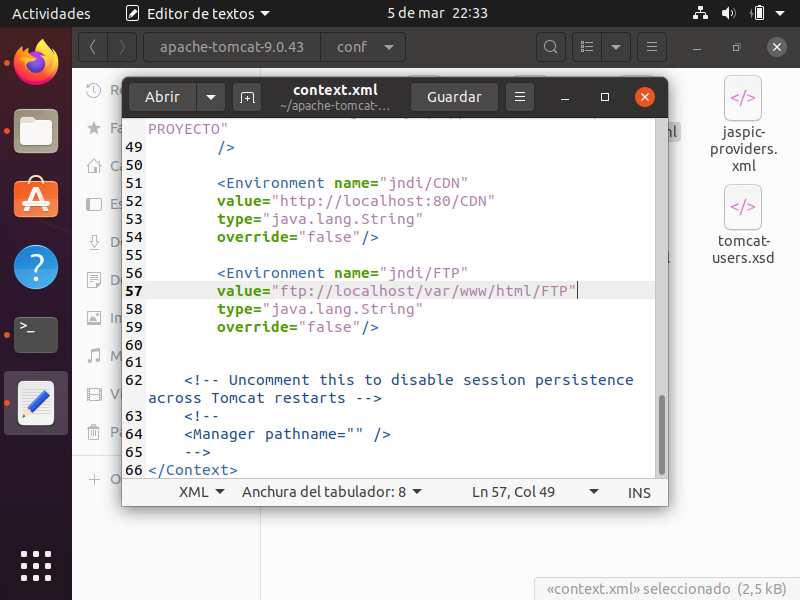
Vemos que con la cuenta administrador nos deja subir el archivo

Pero con la cuenta “registrado” no ya que no tiene permisos.

Podemos comprobar que el archivo efectivamente esta subido al directorio FTP en el que habiamos configurado los “home” de los usuarios:

Y que se puede acceder al directorio de archivos FTP mediante el servidor web:

He intentado configurar el servidor para que la apicación en tomcat pudiese acceder al el:

Pero pese a probar distintas urls en el campo value, no he dado con el formato necesario para que la aplicación pueda trabajar con el servidor ftp, se accede al menú, pero el ver y subir los archivos no funciona:

