**su**

**ÍNDICE**

[BLOQUE 1: Preparando el entorno 1](#_Toc65182954)

[1. General 1](#_Toc65182955)

[2. Base de Datos 3](#_Toc65182956)

[1. Conector 3](#_Toc65182957)

[2. Creación Usuario y Base de Datos en MariaDB 5](#_Toc65182958)

[3. Conexión en el context.xml 7](#_Toc65182959)

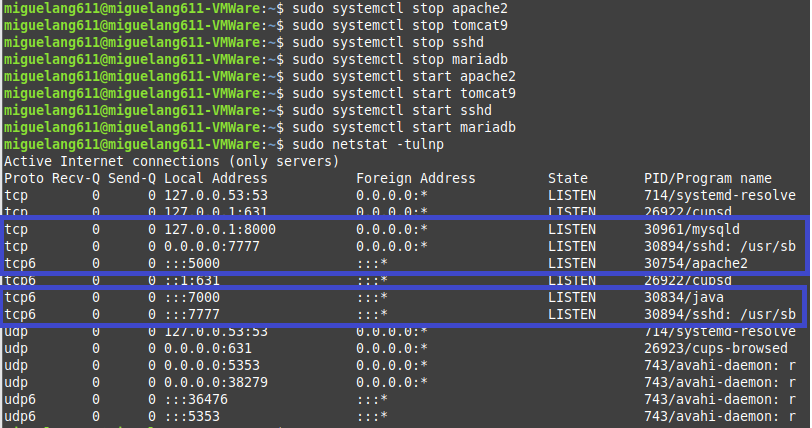
[4. Comprobación funcionamiento 8](#_Toc65182960)

# BLOQUE 1: Preparando el entorno

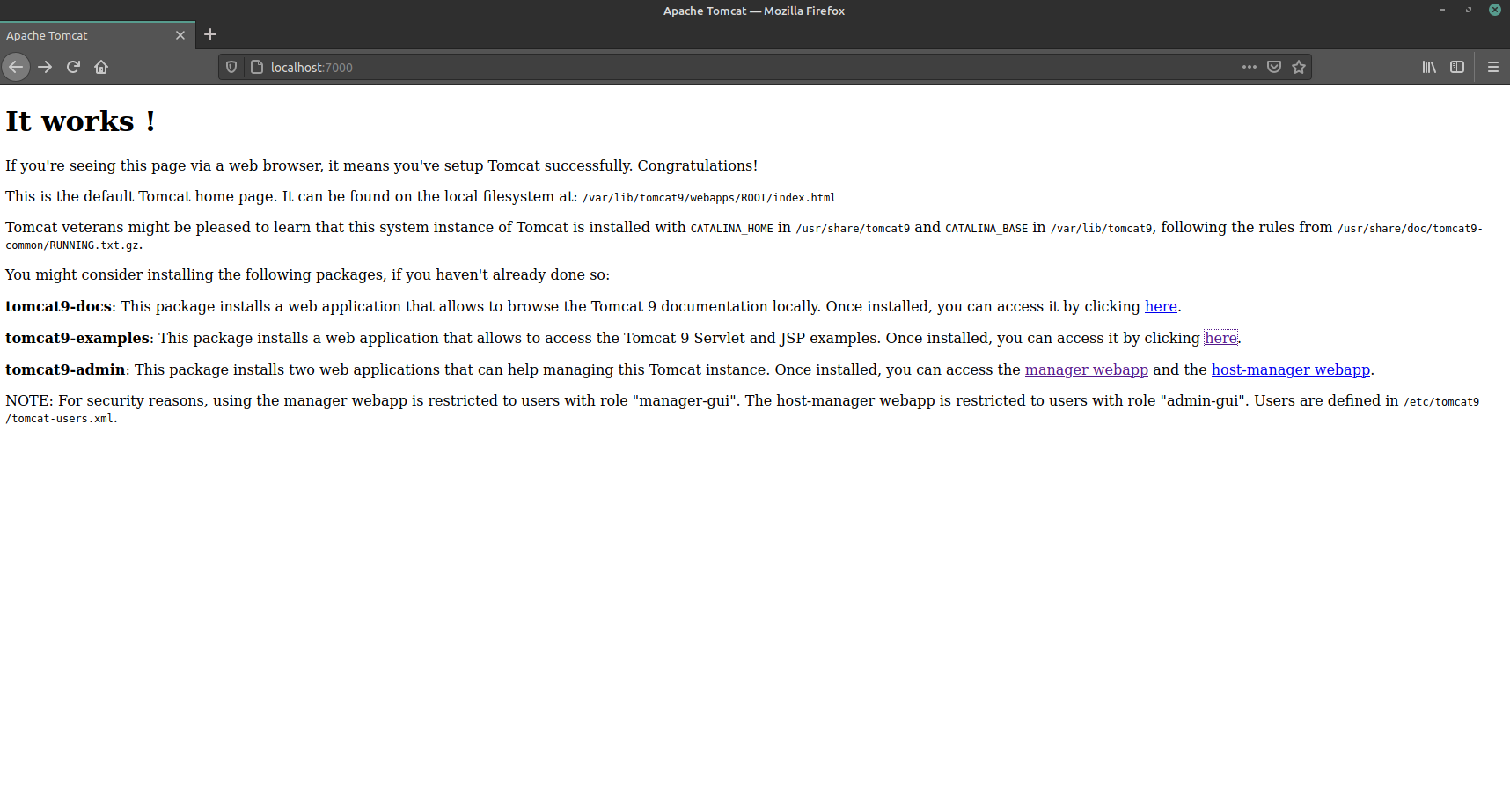
## 1. General

Partiremos de la tarea 2 de Miguel Ángel, lo que implica que tendremos Apache (apache2) corriendo por el puerto 5000, Tomcat (tomcat9) por el puerto 7000, OpenSSH (sshd) por el puerto 7777 y MariaDB (mysqld) por el puerto 8000:

Arrancamos los respectivos servicios, y comprobamos que efectivamente están funcionando por esos puertos:

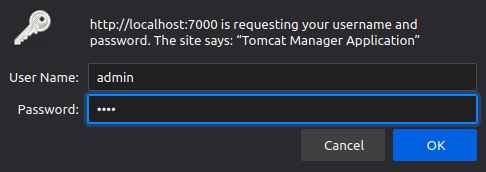


Para comenzar, tenemos que desplegar el archivo .war proporcionado. Para ello nos dirigimos a <http://localhost:7000> , es decir, a la web de Tomcat:

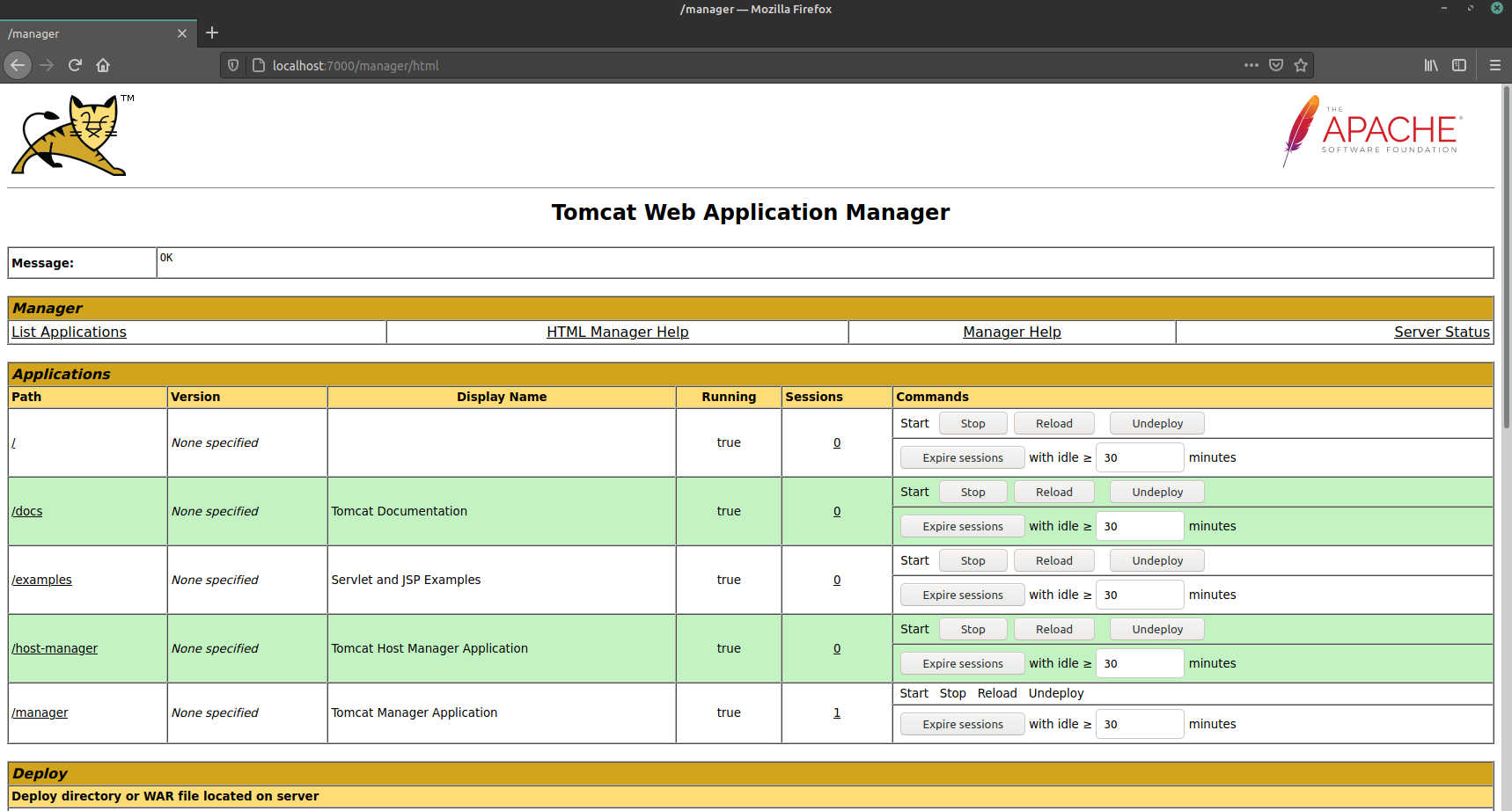


Accedemos al menú de administración web: <http://localhost:7000/manager/html>

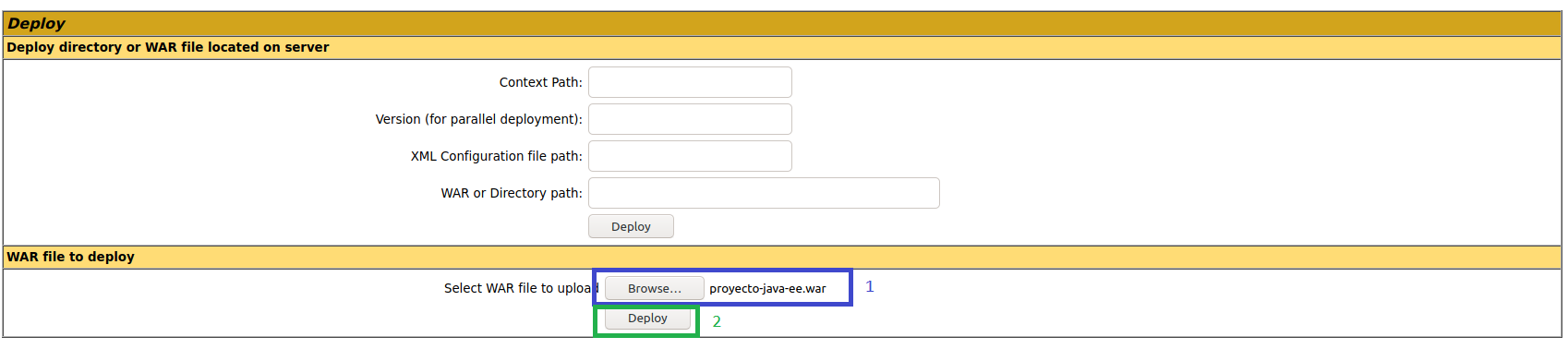
Nos identificamos con las credenciales correspondientes: admin/test:



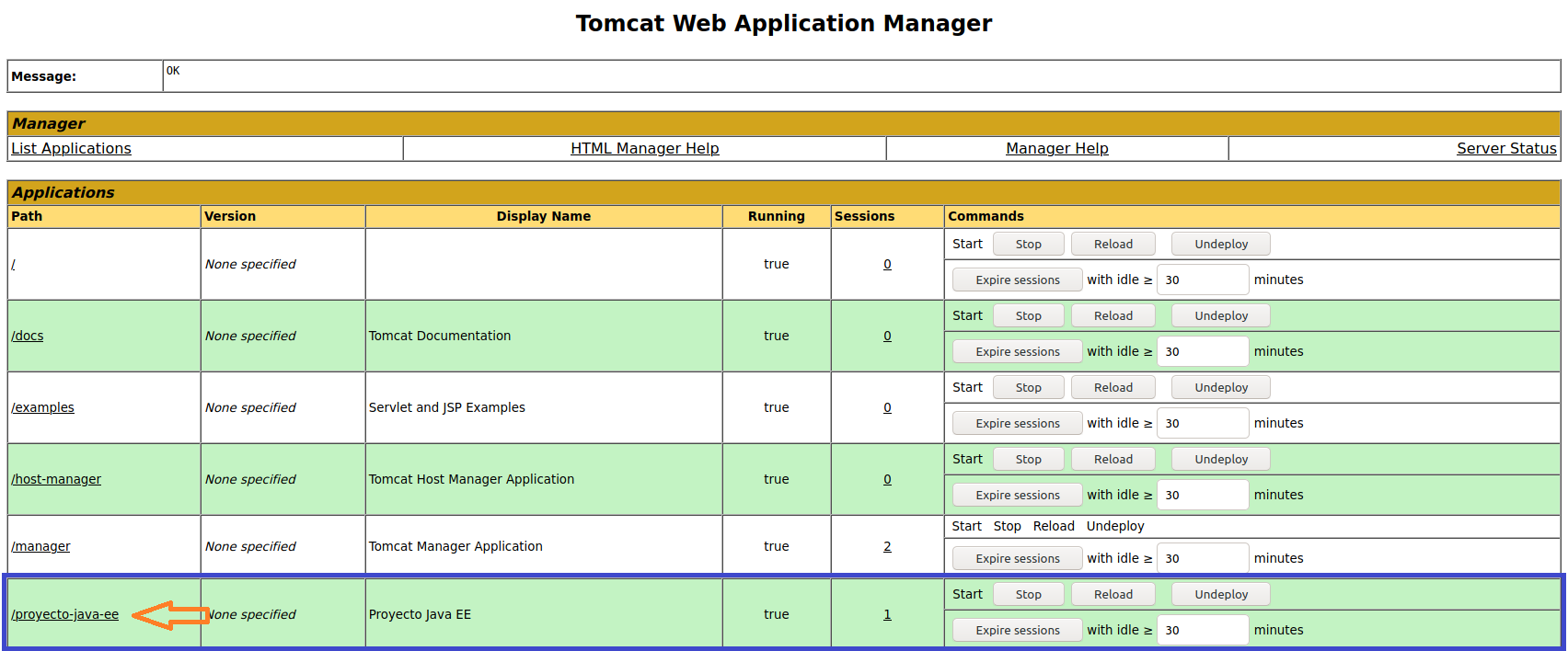
Se nos da la bienvenida al gestor de aplicaciones de Tomcat:



Nos dirigimos al menú de deploy, y dentro de la sección WAR, escogemos el fichero del proyecto, y hacemos click en deploy.



Hecho esto, nos dirigimos de nuevo a la interfaz del manager, observando que el proyecto ha sido cargado. Haremos click sobre /proyecto-java-ee , en la columna path, para dirigirnos a la página principal del proyecto:

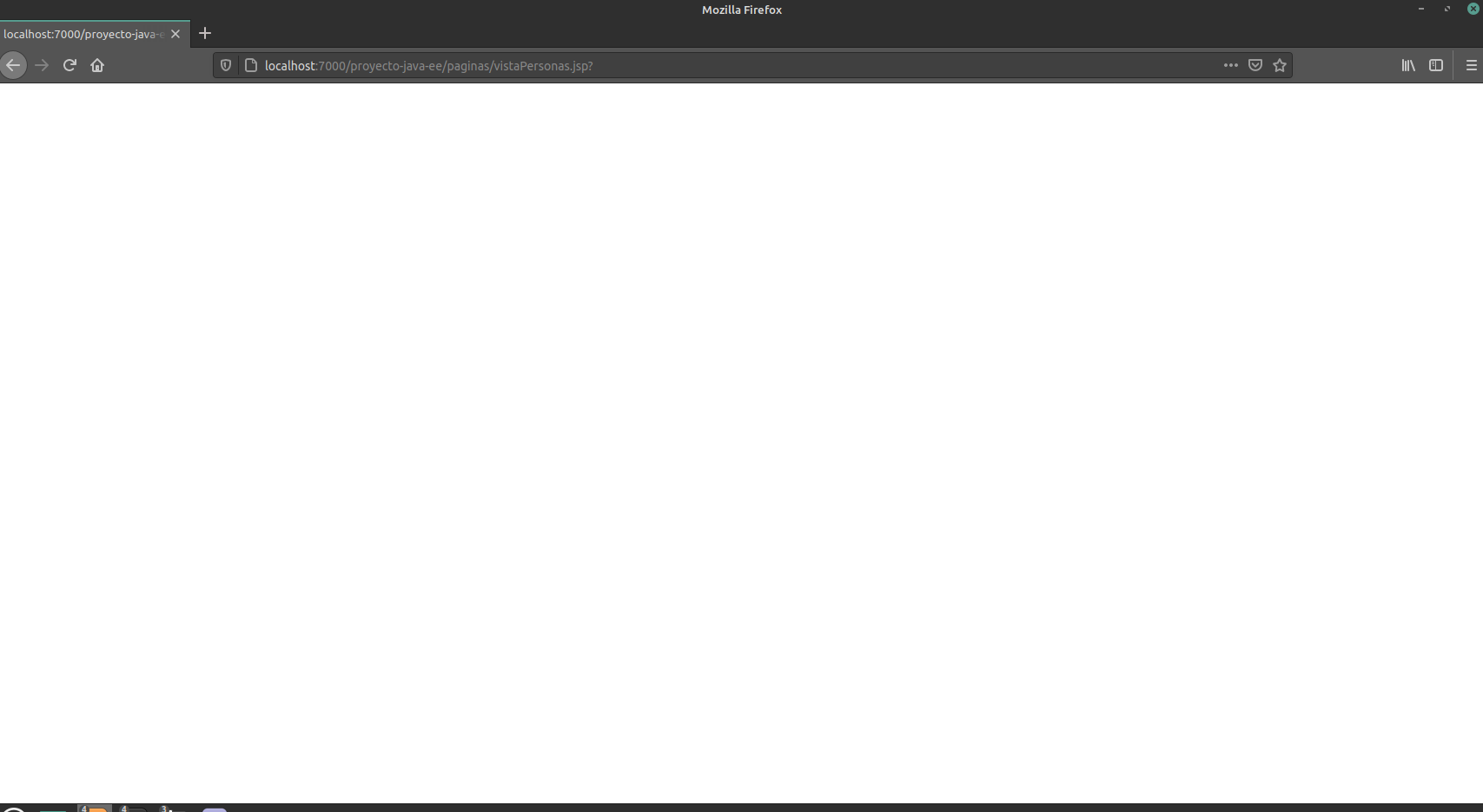


Se nos da la bienvenida:



## 2. Base de Datos

En este momento, vamos a comenzar con la Base de Datos. Si hacemos click en probar, como cabría esperar, no hace nada, ya que no hemos configurado el recurso y conecto JDBC.



Vamos a tratar de poner remedio a esto…

### 1. Conector

Para ello, vamos a descargar inicialmente el conector JDBC de MariaDB:



Escogeremos la versión estable 2.7.2, en su versión .jar

La guardaremos provisionalmente en el Escritorio.

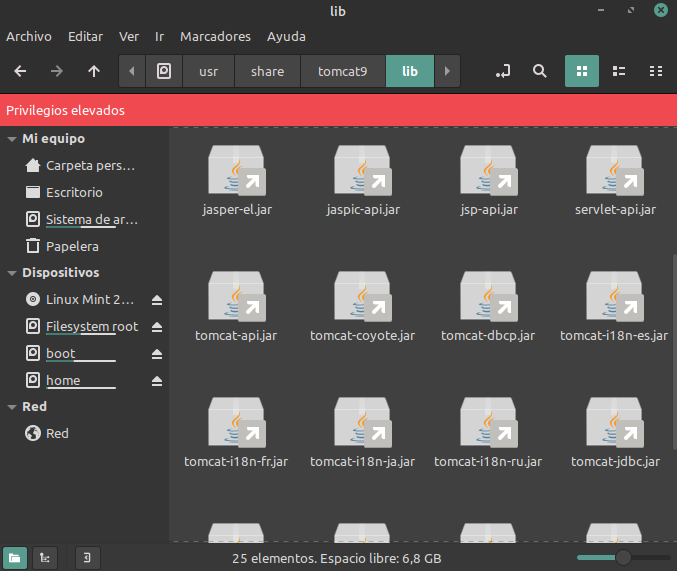
A continuación, vamos a colocar dicha librería en el directorio correspondiente de Tomcat. Puesto que la idea es mantener la carpeta del proyecto intacta, vamos a colocar la librería en el lugar “principal” de las librerías de Tomcat:

/usr/share/tomcat9/lib

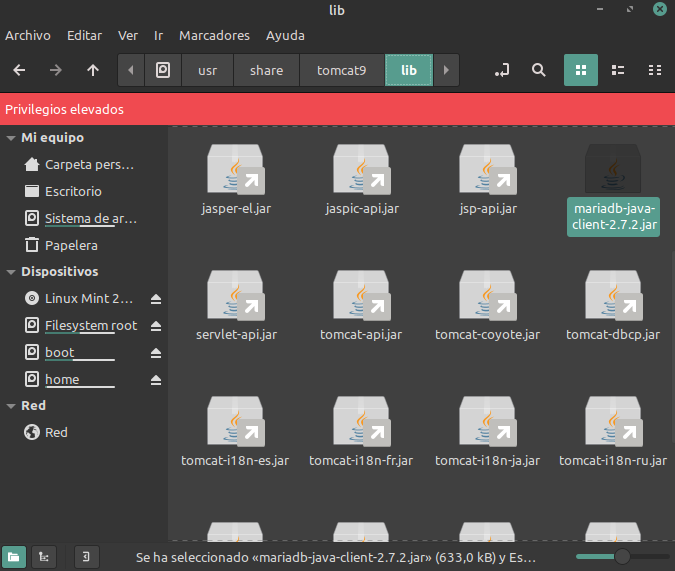
Para ello, ejecutamos desde la Terminal el comando

sudo nemo /usr/share/tomcat9/lib

Lo cual nos dirigirá al explorador de archivos y al directorio correspondiente, con permisos de superusuario:



Copiamos desde el escritorio el conector:



### 2. Creación Usuario y Base de Datos en MariaDB

Hecho esto, procederemos a crear el usuario, contraseña, base de datos y tabla desde MariaDB:

Crear BBDD 🡪 CREATE DATABASE MiBaseDatos;

Crear usuario y darle una contraseña 🡪 CREATE USER 'actividad3'@'localhost' IDENTIFIED BY 'admin';

Otorgarle privilegios 🡪 grant all privileges on MiBaseDatos.\* to 'actividad3'@'localhost';

Limpiar caché 🡪 flush privileges;

Seleccionar BBDD 🡪 use MiBaseDatos

Crear tabla 🡪

CREATE TABLE PERSONAS (

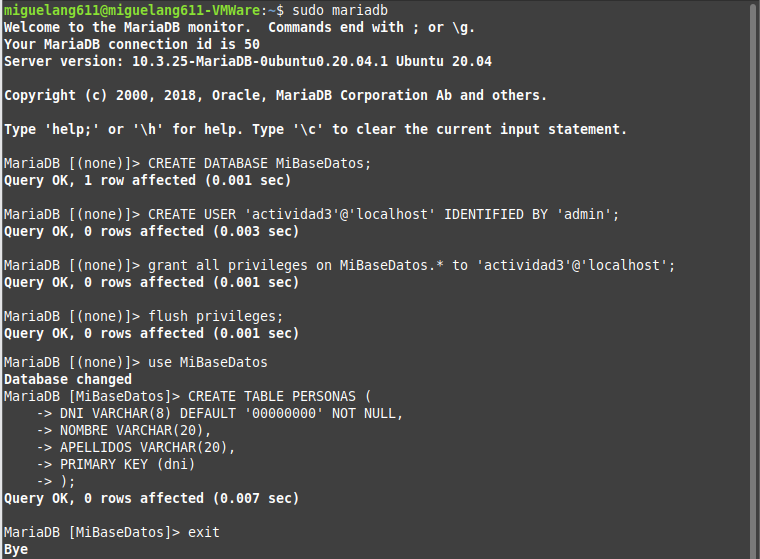
DNI VARCHAR(8) DEFAULT '00000000' NOT NULL,

NOMBRE VARCHAR(20),

APELLIDOS VARCHAR(20),

PRIMARY KEY (dni)

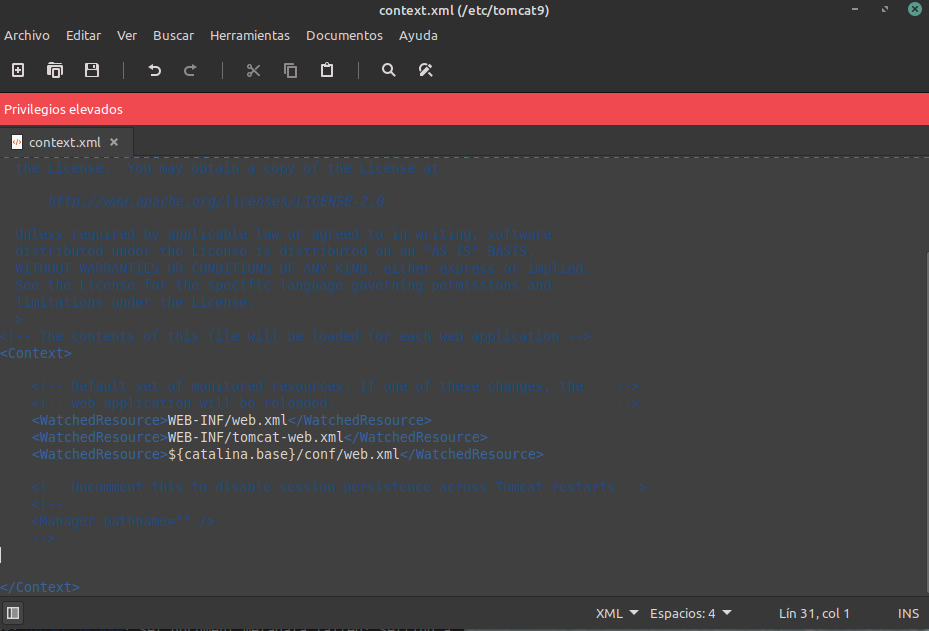
);



### 3. Conexión en el context.xml

Ahora nos toca incluir el recurso correspondiente a la conexión a la base de datos en el context.xml de Tomcat, que se encuentra localizado en /etc/tomcat9/context.xml :

sudo xed /etc/tomcat9/context.xml



Deberemos añadir las siguientes líneas:

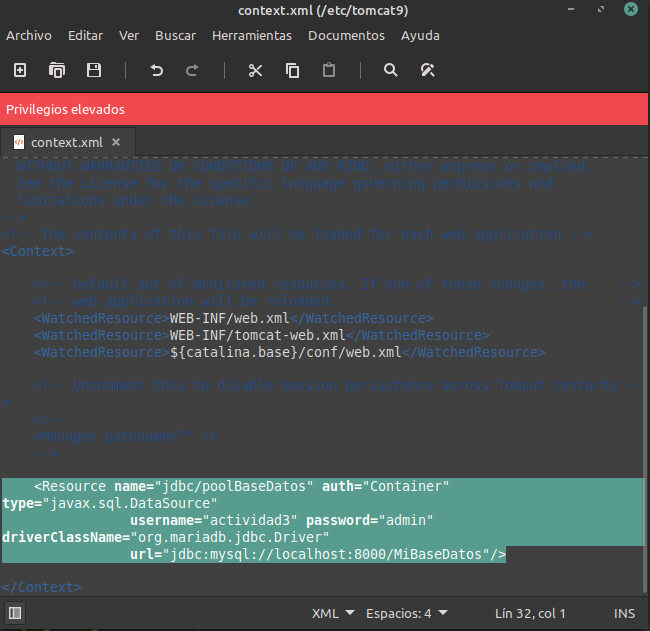
<Resource name="jdbc/poolBaseDatos" auth="Container" type="javax.sql.DataSource"

username="actividad3" password="admin" driverClassName="org.mariadb.jdbc.Driver"

url="jdbc:mysql://localhost:8000/MiBaseDatos"/>

Indicamos pues el nombre de usuario y contraseña que creamos antes, indicamos el nombre de la clase del driver: org.mariadb.jdbc.Driver ; y en la url indicaremos jdbc:mysql//localhost:8000/MiBaseDatos en referencia a que está en local, en el puerto 8000, y que el nombre de la misma es MiBaseDatos.

La tabla se leerá desde la aplicación Java, y esa es la que se denomina Personas.



### 4. Comprobación funcionamiento

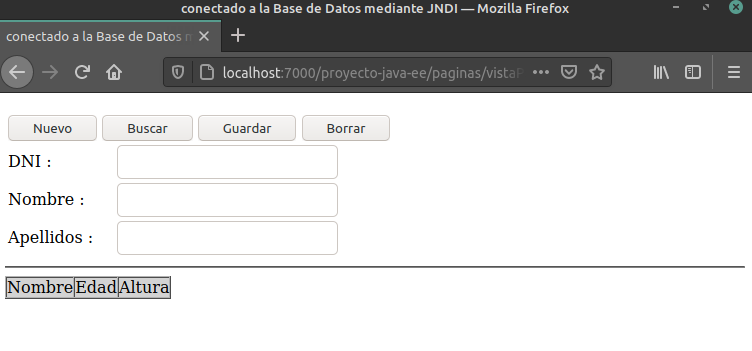
Procedemos a parar e iniciar el servicio de MariaDB y de Tomcat.

Nos dirigimos a la página principal del proyecto:



1. Prueba de BBDD

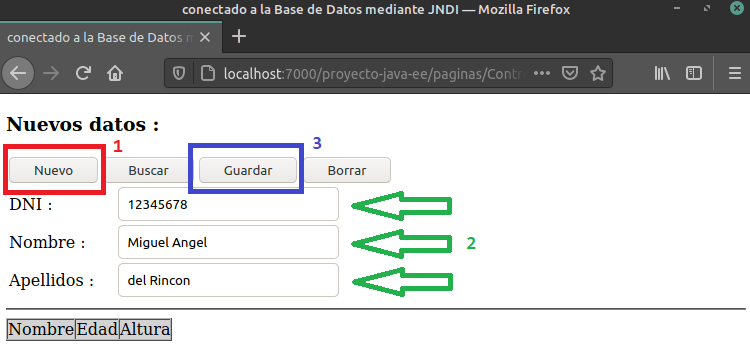
Hacemos click en probar BD:



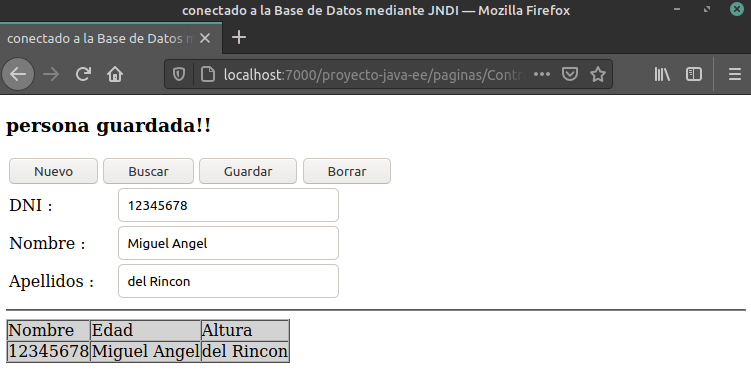
Comprobamos que la conexión es exitosa.

2. Inserción y guardado

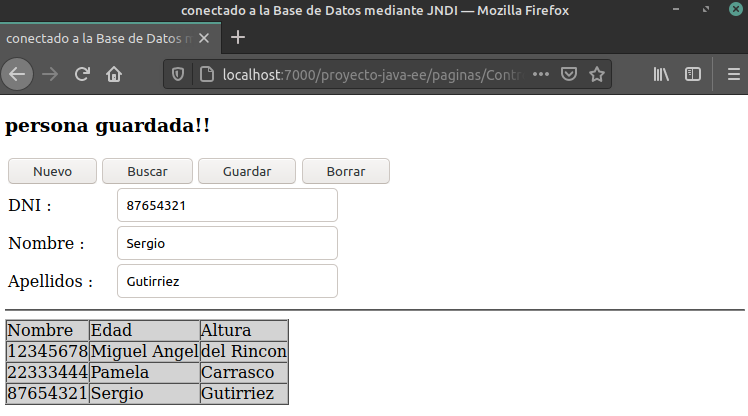
Vamos a probar a insertar datos, haciendo click en Nuevo, introduciendo los datos, y después haciendo click en guardar:



El resultado es el siguiente:

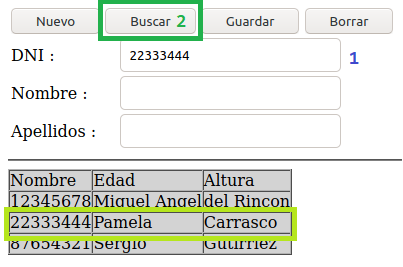


Procedemos análogamente con el nombre de todos los integrantes, siendo este el resultado:

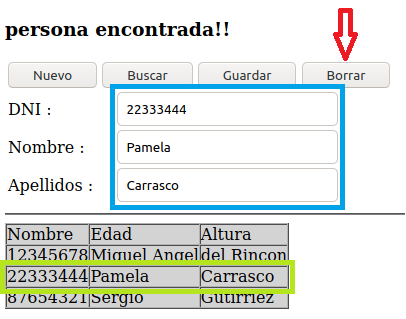


3. Búsqueda

Ahora, podemos introducir un DNI (clave primaria), y hacemos click en buscar:

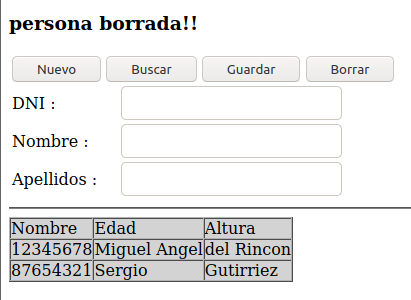


Nos devuelve un aviso de persona encontrada, y la ficha de la persona:



4. Borrado

Si hacemos click sobre borrar, se nos eliminará este registro de la BBDD, de lo que se nos avisa, y además se refresca la lista de datos, desapareciendo Pamela:

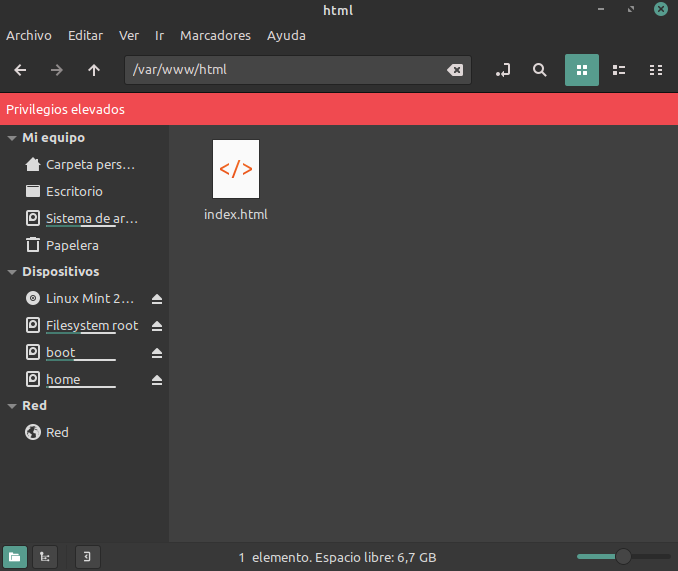


Damos así pues por finalizadas las pruebas de la Base de Datos, siendo satisfactorio su funcionamiento.

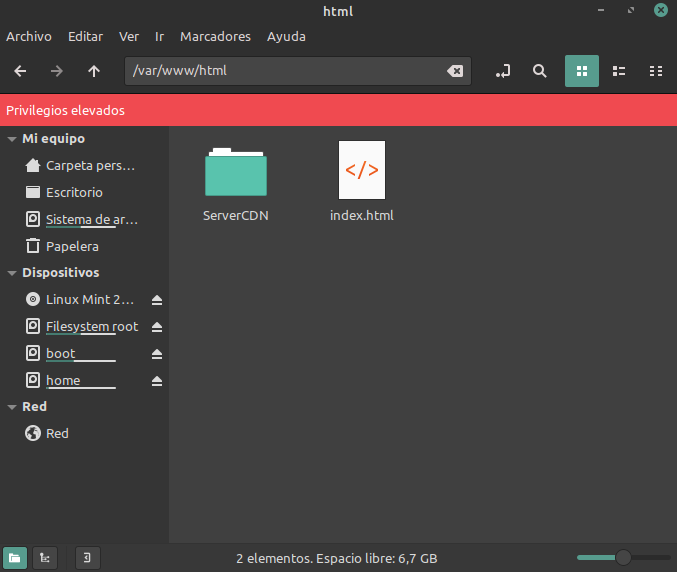
## 3. CDN

Para hacer funcionar el servicio CDN lo que tenemos que hacer es crear una carpeta dentro del servidor Apache donde localizaremos los archivos:

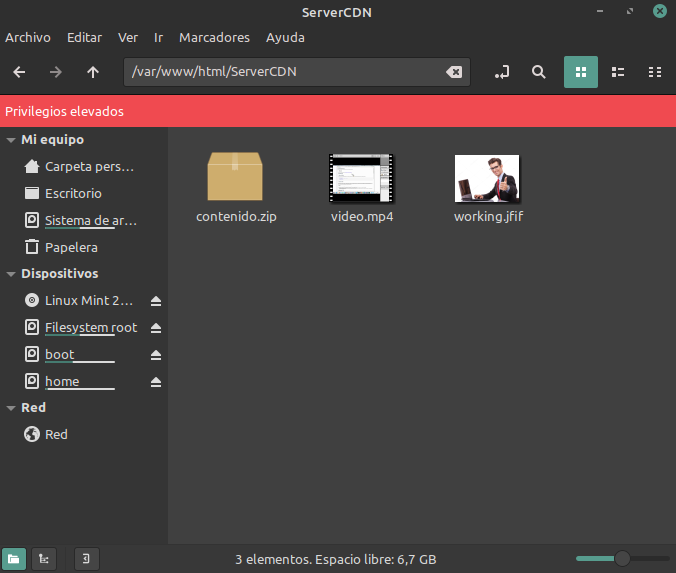
sudo nemo /var/www/html



Crearemos una carpeta llamada ServerCDN, donde alojaremos los 3 archivos, de modo que el directorio /var/www/html quedará así:



Y dentro de dicho directorio tendremos los 3 ficheros:

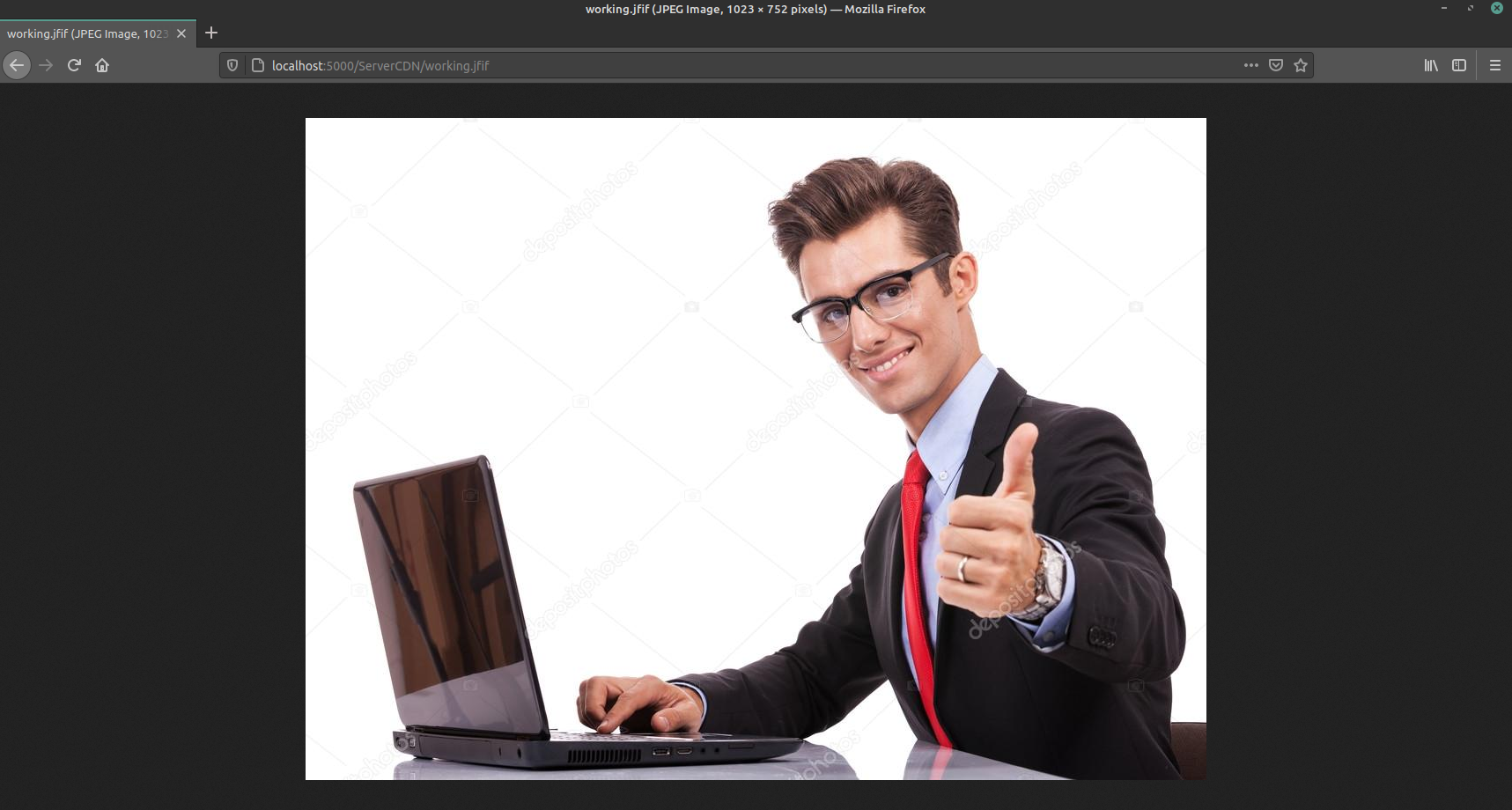


Reiniciaremos el servicio de Apache:

sudo systemctl stop apache2

sudo systemctl start apache2

Ahora, probaremos que podemos acceder desde el navegador web a estos ficheros por medio del servidor Apache, por ejemplo: <http://localhost:5000/ServerCDN/working.jfif>



Una vez comprobado esto, procederemos a realizar los cambios necesarios en Tomcat para que el programa pueda acceder a los ficheros.

Deberemos crear una variable de entorno, que localizaremos dentro del context.xml, para ello ejecutaremos el comando desde la consola para abrir con permisos elevados el editor de texto:

sudo xed /etc/tomcat9/context.xml

Deberemos indicar el nombre de acuerdo a lo requerido por el programa, indicaremos que es una variable de entorno de tipo String; en el value colocaremos la dirección del servidor Apache con el puerto y con la carpeta dentro del Apache en la que está:

<Environment name="jndi/CDN"

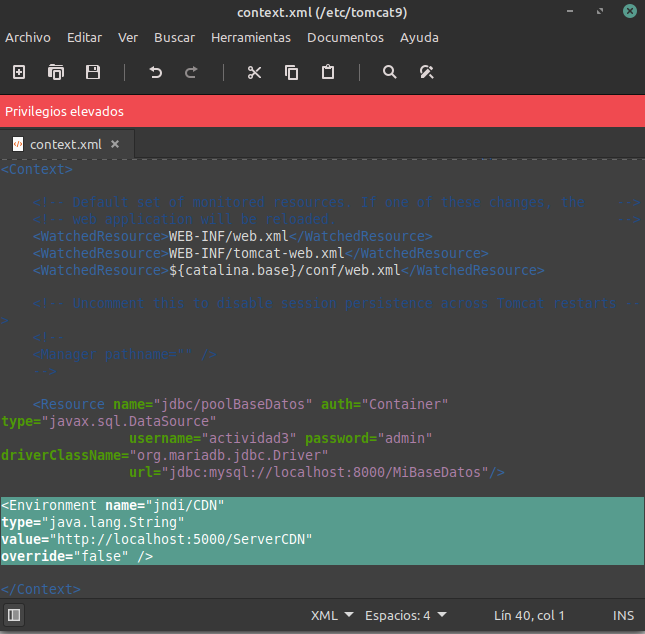
type="java.lang.String"

override="false"

value="http://localhost:5000/ServerCDN"

/>

De esta manera, nos quedaría así el context.xml:



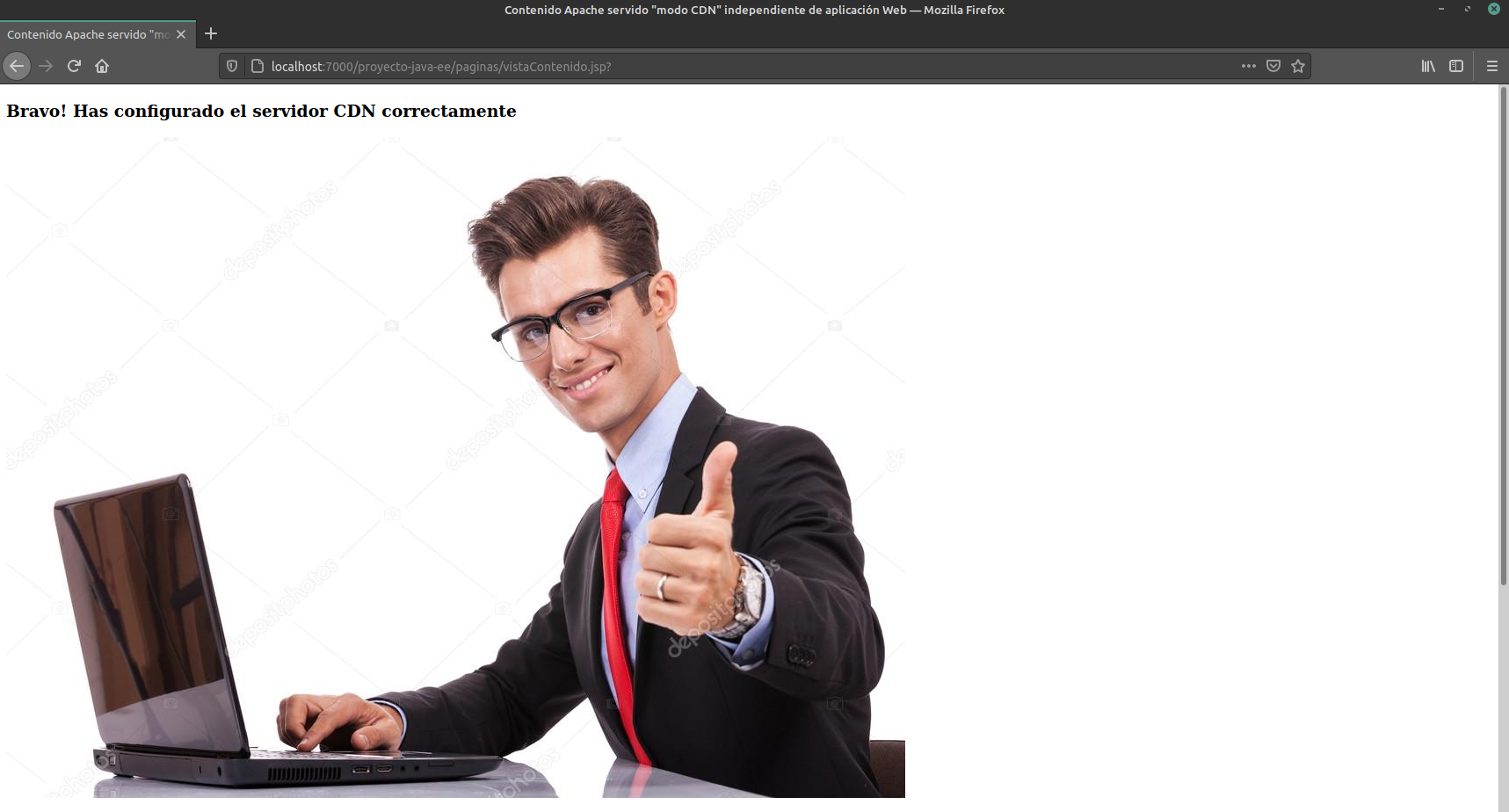
Reiniciaremos el servicio de Tomcat con:

sudo systemctl stop tomcat9

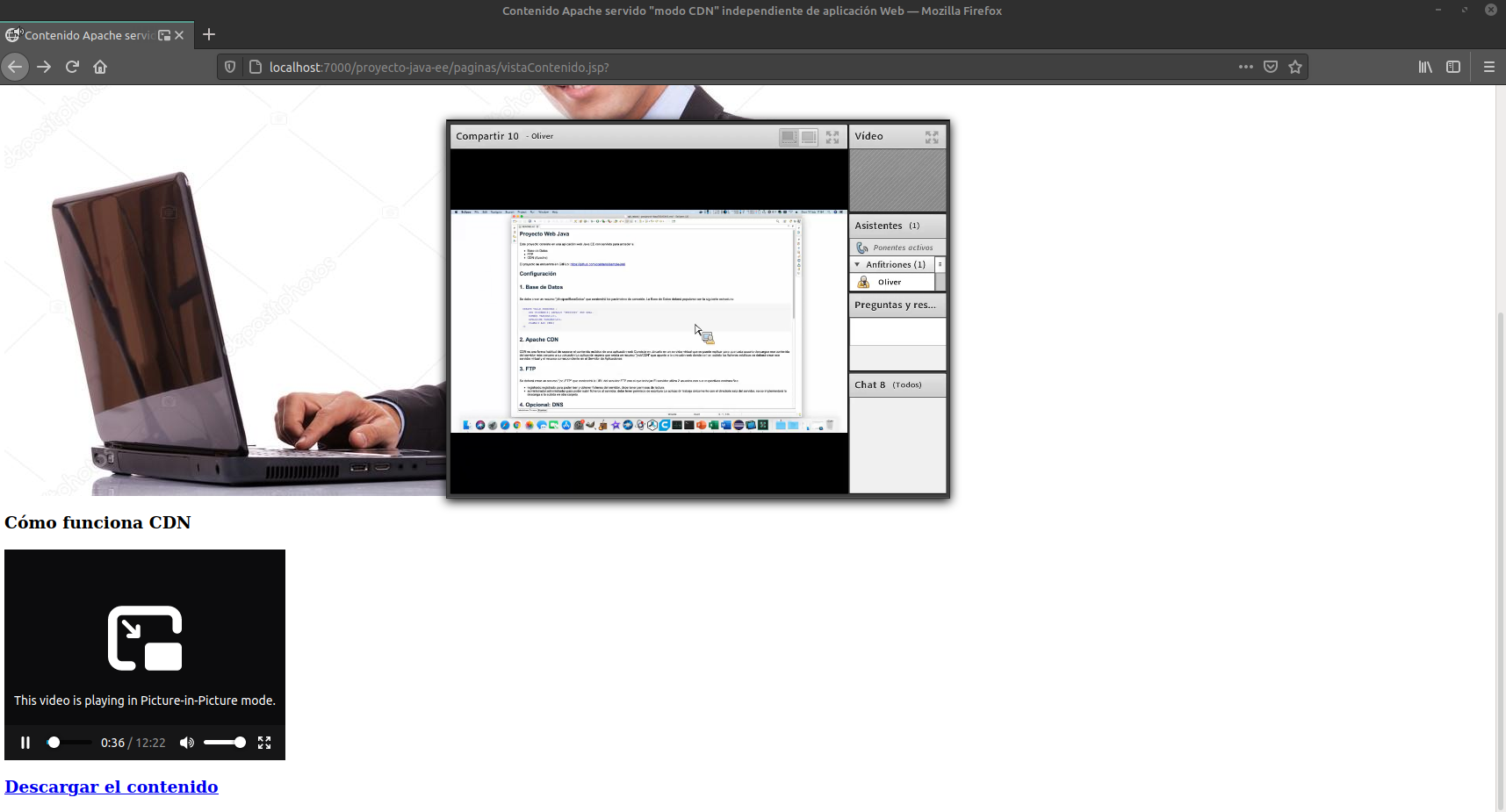
sudo systemctl start tomcat9

Comprobamos que el CDN ha quedado funcionando de manera correcta:

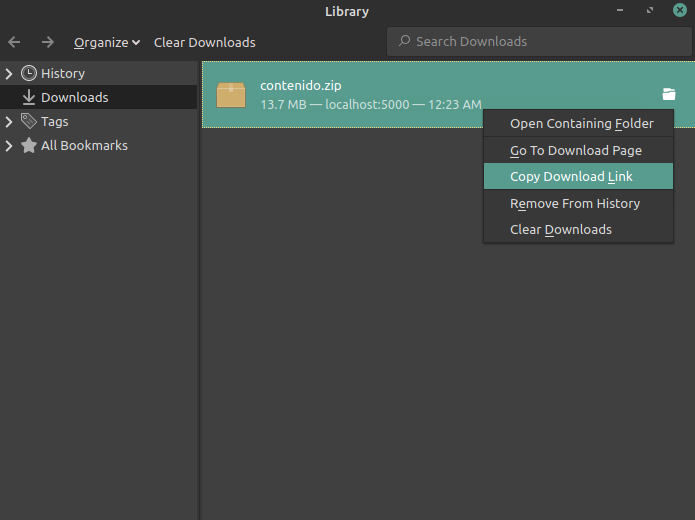
Vemos la imagen:



Vemos el vídeo:



Si hacemos click en descargar, vemos que funciona correctamente, y además, observamos que la dirección de enlace es la del CDN Apache:



<http://localhost:5000/ServerCDN/contenido.zip>

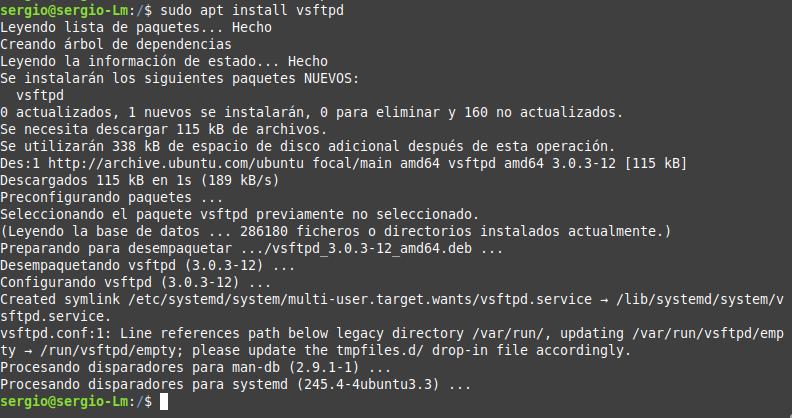
Dentro del zip descargado encontramos lo esperado: el PDF de CDN, el vídeo y la imagen:



## 4. FTP

Instalamos **FTP Server** con el comando :

sudo apt install vsftpd

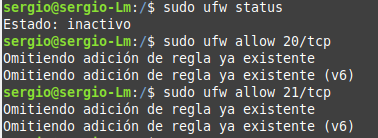


Creamos una copia de seguridad antes de continuar ya que vamos a hacer cambios en el archivo vsftpd.conf

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.confinicial



Ahora pasamos a habilitar y configurar el firewall:

 sudo ufw status

sudo ufw allow 20/tcp

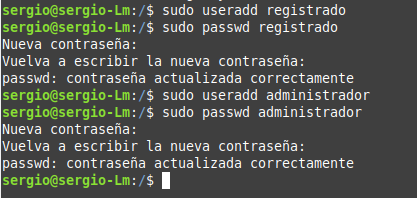
sudo ufw allow 21/tcp

20 y 21 puertos de ftp.

Después vamos a crear los usuarios registrado y administrador y vamos a darles una password.

Primero creamos un directorio donde vamos a guardar una lista con los usuarios:

sudo mkdir /etc/vsftpd/user\_list

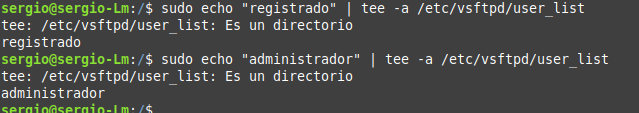


sudo nano /etc/vsftpd/user\_list/user.list

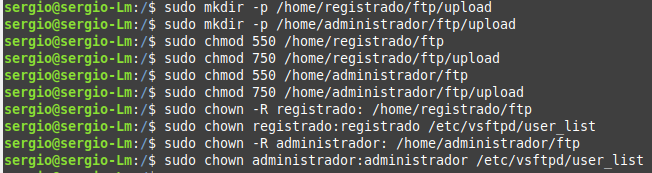
Añadimos restricciones a cada usuario:

sudo echo "registrado" | tee –a /etc/vsftpd/user\_list

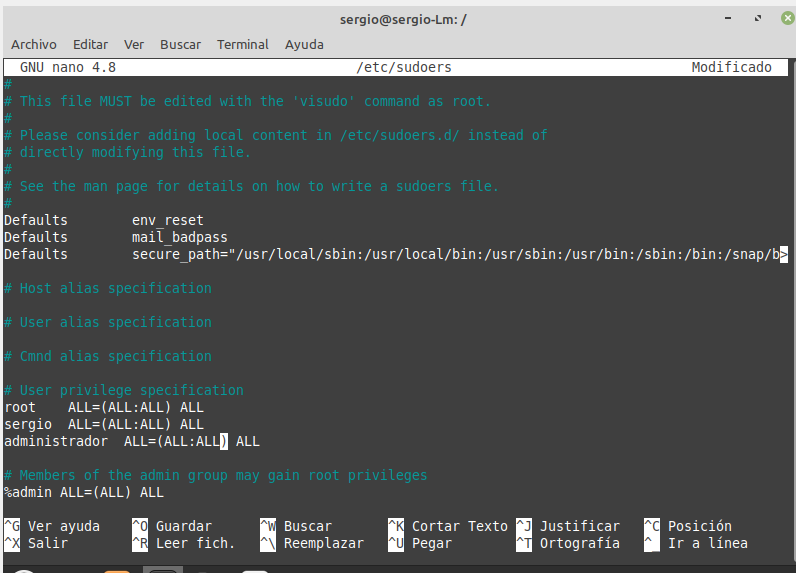
sudo echo "administrador" | tee –a /etc/vsftpd/user\_list



Creamos las carpetas para los usuarios y le asginamos los permisos correspondientes: (leer y escribir)



A traves del commando *sudo nano /etc/sudoers* agregamos los permisos:



Vamos a crear una carpeta en /home que se llama ftp:

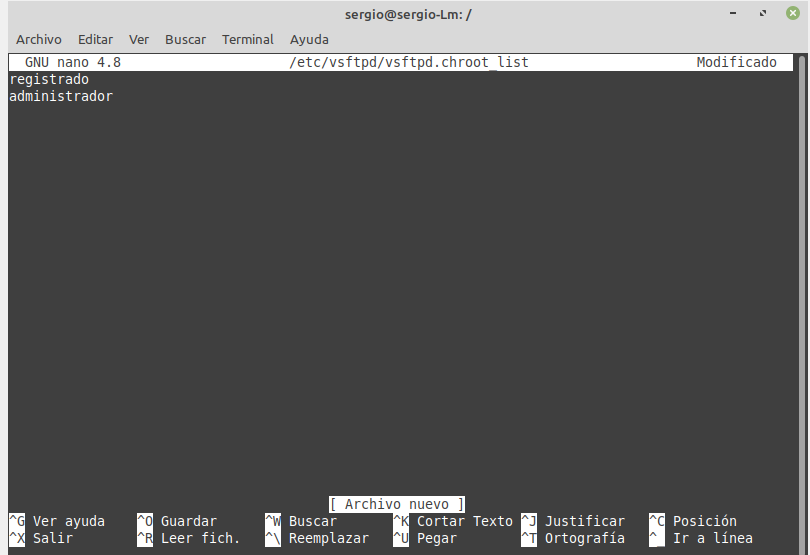
sudo mkdir /home/ftp

Y le agregamos los permisos de lectura y escritura al usuario y al grupo de propietario con el 6, y el 4 al resto, que es de solo lectura.



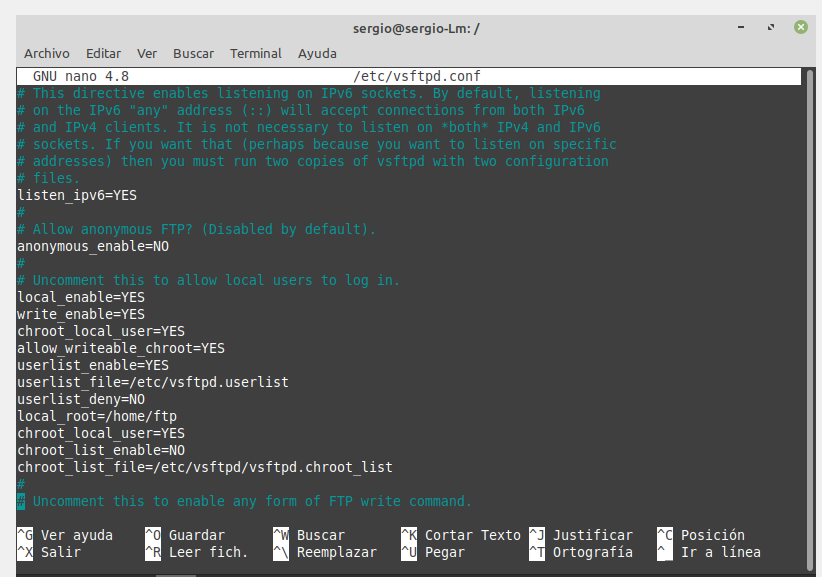
Creamos un archive vsftpd.chroot\_list y añadimos los dos usuarios en el mismo:

sudo nano /etc/vsftpd/vsftpd.chroot\_list



Ahora vamos a configurar los permisos de los usuarios en el archivo vsftpd.conf:

sudo nano /etc/vsftpd.conf

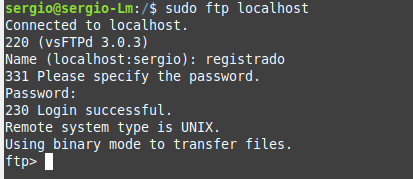
Dentro de este archivo vamos a añadir y a modificar los siguientes parametros:

Despues de todos los cambios realizados vamos a iniciar el servidor vsftpd con :

sudo systemctl start vsftpd

sudo systemctl enable vsftpd

Finalmente utilizamos el commando **sudo ftp localhost**, en el cual nos pedira usuario y password, e iniciaremos conexión.



Como podemos observer lista los archivos y directorios que hay en la carpeta y se logea corractamente, pero por algún motivo que no hemos conseguido resolver de los permisos, no es capaza de subir los archivos.

