qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| Despliegue Proyecto Web Java  Actividad 3. Grupal.  27/02/2021  Raúl Montero Peregrina Javier Pérez Vivar Patricia Villanueva Polo Jaime Juárez Fernández |

ÍNDICE

[I. Enunciado 4](#_Toc65340028)

[II. Introducción 6](#_Toc65340029)

[III. Patricia. Instalación de Apache, Tomcat y BBDD 6](#_Toc65340030)

[Instalar y configurar el servidor Apache 6](#_Toc65340031)

[Instalación y configuración de la BBDD 8](#_Toc65340032)

[3. Instalación y configuración de TomCat 10](#_Toc65340033)

[La BBDD 17](#_Toc65340034)

[IV. Raúl. Replica anterior. Incluye CDN apache 20](#_Toc65340035)

[Configuraciones y BBDD 20](#_Toc65340036)

[APACHE CDN 24](#_Toc65340037)

[V. PARTE JAVIER. REPLICA ANTERIOR Y AÑADE CERTIFICADO DIGITAL EN APACHE 26](#_Toc65340038)

[Crear el certificado SSL 26](#_Toc65340039)

[Configurar Apache para usar SSL 28](#_Toc65340040)

[Ajustar el firewall 32](#_Toc65340041)

[Habilitar los cambios en Apache 33](#_Toc65340042)

[Probar el cifrado 34](#_Toc65340043)

[VI. Jaime. Replica anterior. Instalación de FTP y gestión de usuarios. 36](#_Toc65340044)

[Habilitar los cambios en Apache 36](#_Toc65340045)

[Modificar archivo de configuración 37](#_Toc65340046)

[Crear nuevos usuarios. 39](#_Toc65340048)

[VII. Apéndices/Bibliografía 41](#_Toc65340049)

[VIII. Observaciones 41](#_Toc65340050)

# Enunciado

**Proyecto Web Java**

Este proyecto consiste en una aplicación web Java EE con servlets para acceder a

* Base de Datos
* FTP
* CDN (Apache)

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)**

**);**

**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

**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

**4. Opcional: DNS**

Opcionalmente se podrá registrar nuestro servidor en un DNS (local o público) para poder hacer uso de URLs más representativas para cada una de las piezas del sistema: Apache, FTP y Tomcat

**5. Librerías y ficheros**

1. Base de Datos

Para la conexión a Base de Datos se requieren las librerías Java correspondientes a la Base de Datos en cuestión.

Para MySQL y MariaDB se deberá utilizar: https://mvnrepository.com/artifact/mysql/mysql-connector-java

Para Oracle se deberá utilizar: https://mvnrepository.com/artifact/oracle/oracle-jdbc

Para otras Bases de Datos se deberá utilizar el JDBC correspondiente

2. Contenido

Éste 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

3. Aplicación

Éste fichero "proyecto-java-ee.war" contiene la aplicación Java que deberá desplegarse en el servidor de aplicaciones

4. FTP

Deberá hacerse accesible en el servidor FTP el PDF que explica la resolución de la práctica y un fichero ZIP que contenga los ficheros de configuración que se han creado. Además, se deberán añadir estos ficheros de configuración al repositorio GitHub que se ha utilizado para las actividades anteriores

5. Despliegue

Se deberán publicar las librerías necesarias en la carpeta /lib del Servidor de Aplicaciones

Se deberá configurar el recurso "jdbc/poolBaseDatos" en el Contexto del Servidor de Aplicaciones

Se deberán configurar las variables de entorno "jndi/FTP" y "jndi/CDN" en el Contexto del Servidor de Aplicaciones

Se deberá desplegar el WAR del proyecto en el Servidor de Aplicaciones

Se deberán crear los usuarios "registrado" y "administrador" en el Servidor FTP

Se deberá configurar el Servidor Apache para poder acceder al directorio CDN y desplegar ahí los ficheros de prueba.

# Introducción

La idea es que los 4 componentes del equipo trabajen juntos desplegando una App Web dada sobre una misma configuración en una máquina virtual de Ubuntu. Para ello cada uno de los 4 componentes.

Para ello debemos instalar lo siguiente:

1. **Servidor Apache**: Este servidor nos permitirá mostrar el contenido web de la aplicación java.
2. **Tomcat**: Es un contenedor de servlets que se puede usar para compilar y ejecutar aplicaciones web realizadas en Java. Implementa y da soporte tanto a servlets como a páginas JSP
3. **FTP**: Para poder utilizar una de las funcionalidades de la aplicación, que se conecta a un servidor ftp para subir y descargar archivos
4. **BBDD**: La necesitamos para poder usar una de las funcionalidades de la aplicación, en la cual podemos dar de alta y eliminar registros de una tabla de una BBDD.

# Patricia. Instalación de Apache, Tomcat y BBDD

## Instalar y configurar el servidor Apache

Creo un usuario administrador para mi servidor apache:

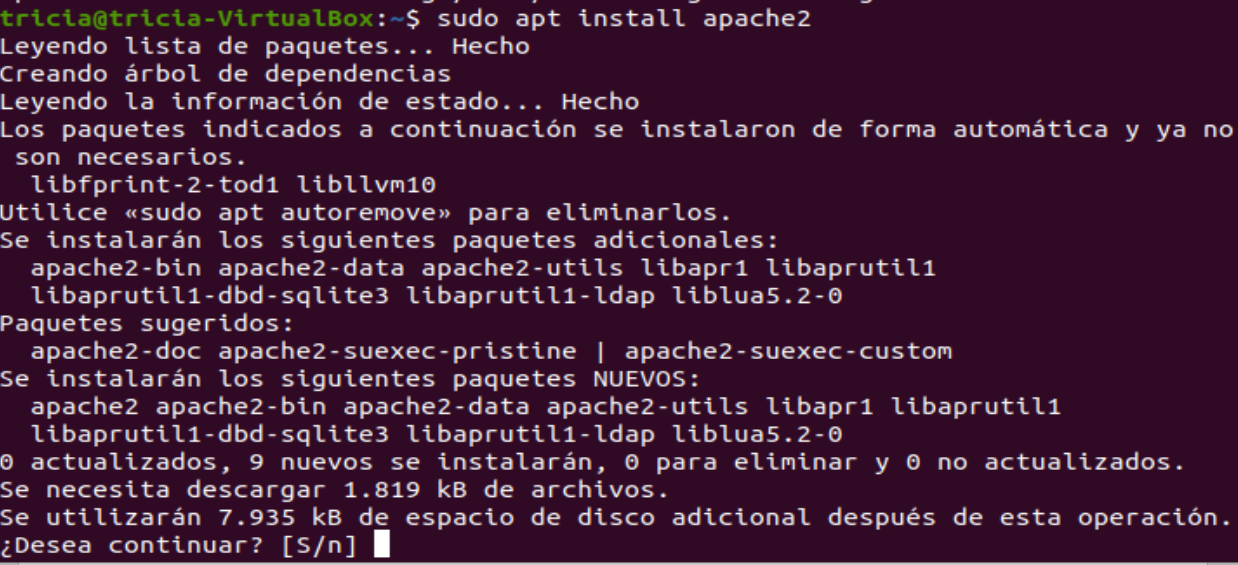
**Sudo adduser tricia**

Y lo meto en el grupo “sudo” para que tenga permisos de administrador:

**usermod -aG sudo user\_apch**

A continuación, instalamos Apache:

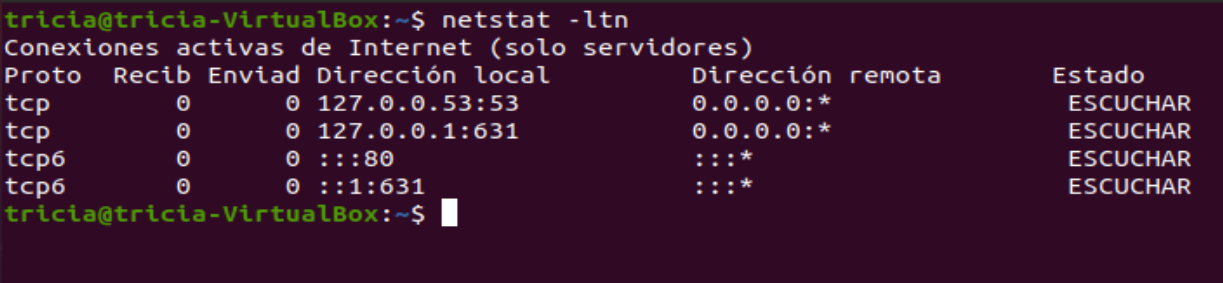
**Sudo aptinstall apache2**



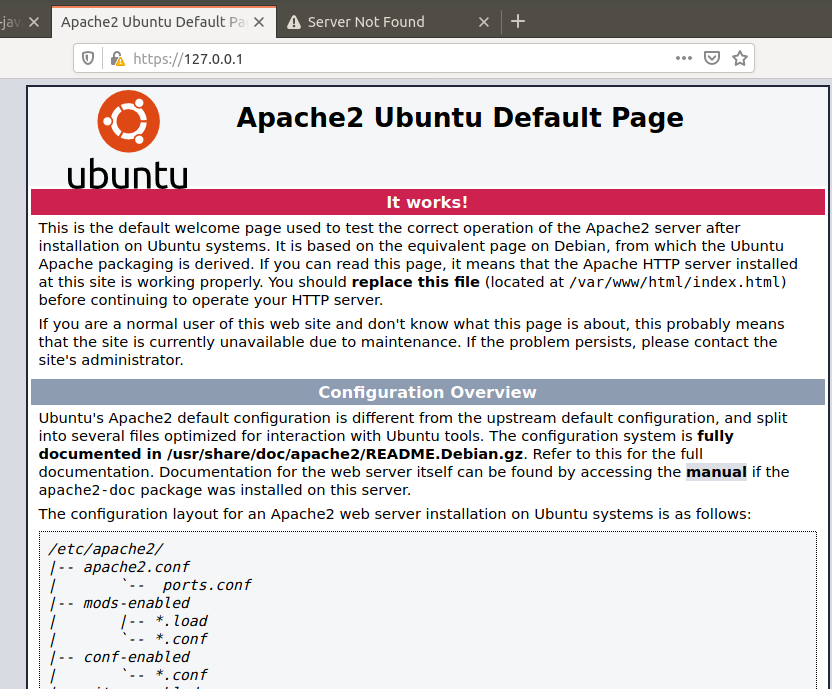
Comprobamos que está corriendo:



Y en que IP lo está haciendo, en nuestro caso en la 127.0.0.1

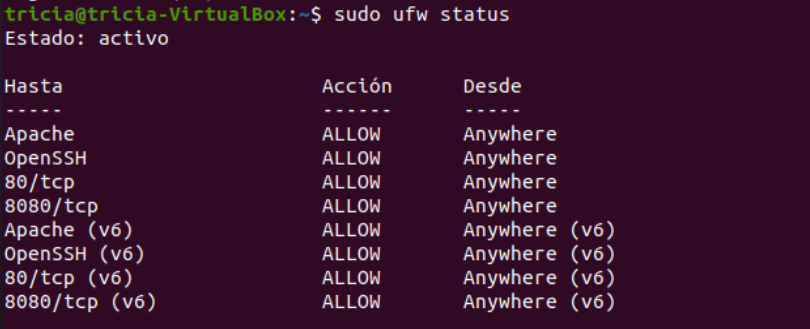


Si vamos a un navegador y tecleamos “localhost” o “127.0.0.1” podremos ver la instalación de Apache:



A continuación, vamos a abrir los puertos de Apache y ApacheFull en nuestro cortafuegos. Serían los puertos 80 y 443. Para ello tecleamos:

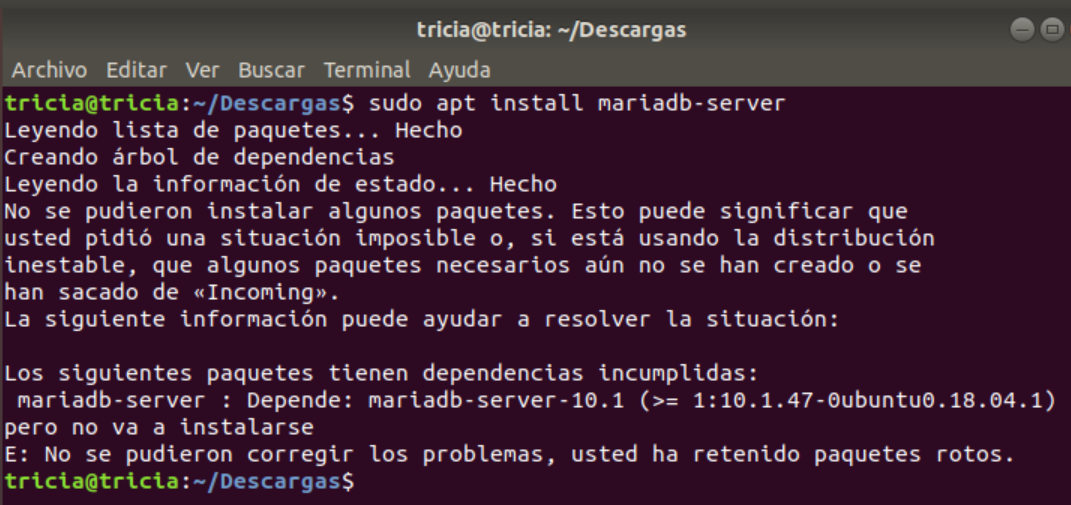
**Sudo ufw allow 80/tcp** y **sudo ufw allow 443/tcp**

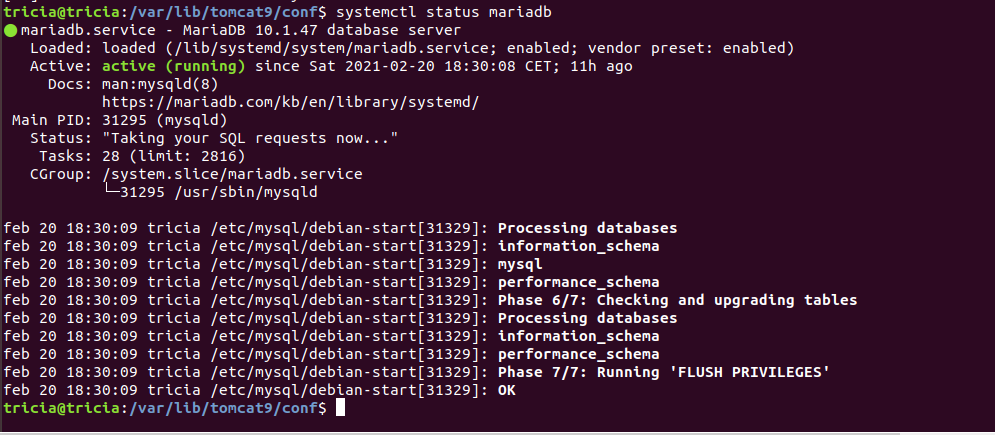


El 80 nos dejará acceder al servidor por http y el 443 por https, es decir por una conexión segura.

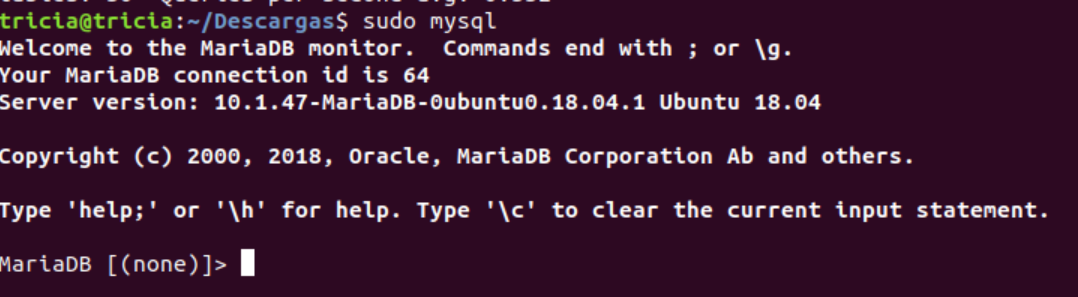
## Instalación y configuración de la BBDD

Para esta práctica vamos a utilizar mariadb:





Accedemos a mariadb:



Creamos un usuario con permisos administrador para la BBDD:

GRANT ALL ON \*.\* TO 'tricia'@'localhost' IDENTIFIED BY 'cursodaw' WITH GRANT OPTION;

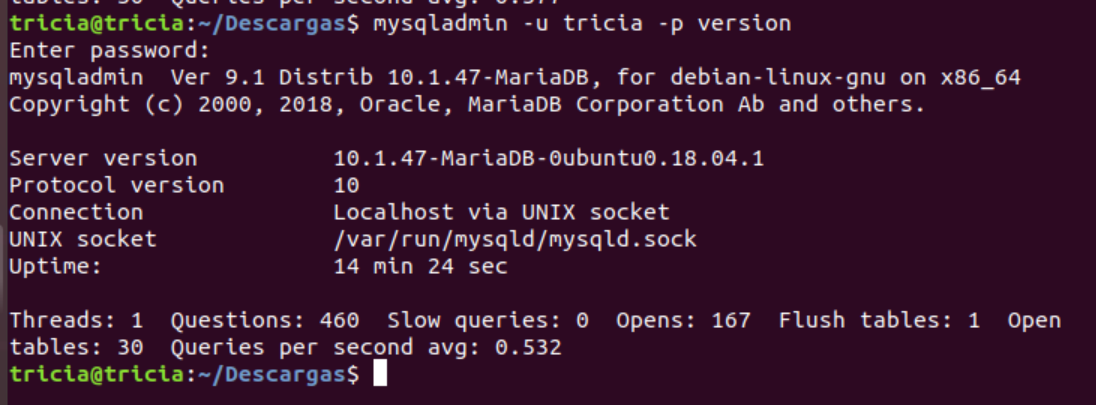
FLUSH PRIVILEGES

Exit

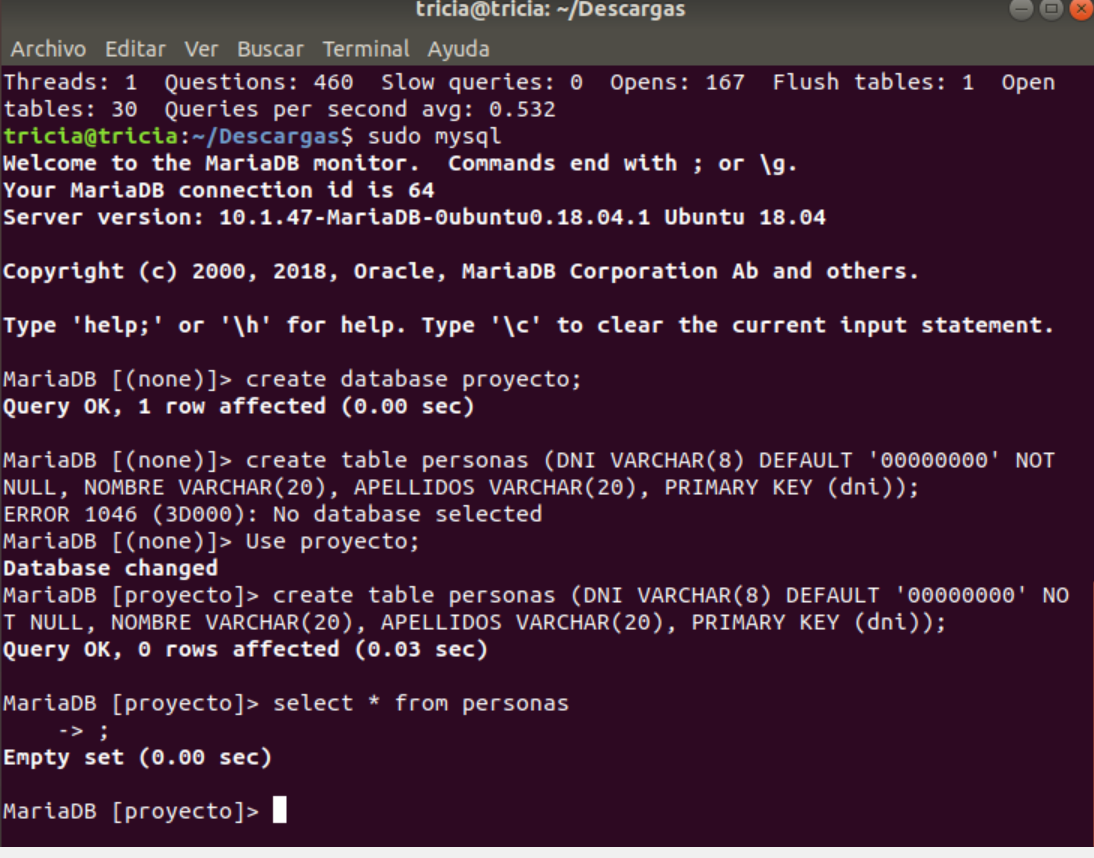
El siguiente comando me indica que puedo establecer conexión con la BBDD como root:



Y este que puedo hacerlo con mi usuario administrador “tricia”



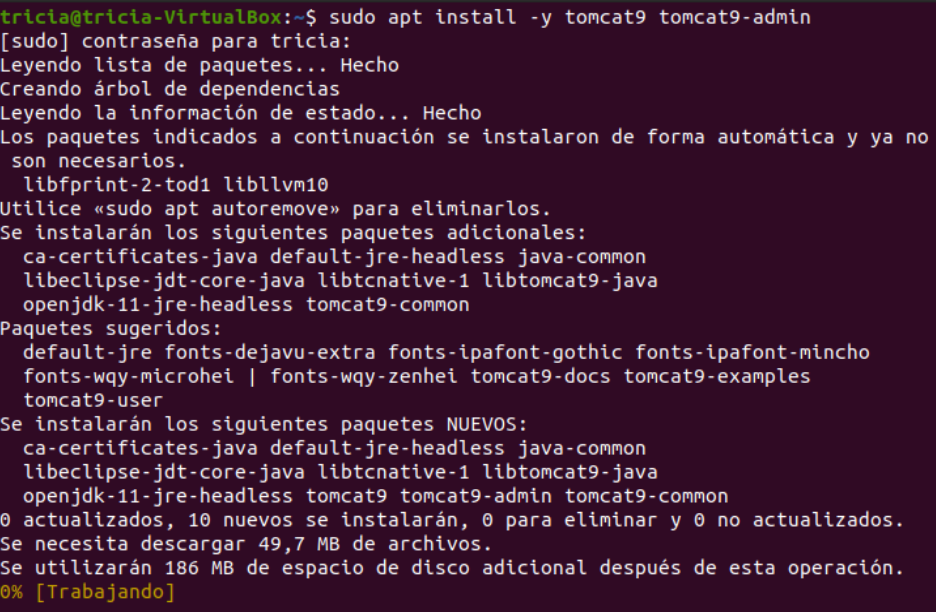
A continuación, accedo a la base de datos **sudo mysql** y creo la base de datos y la tabla para el ejercicio, tal y como se muestra en la captura.



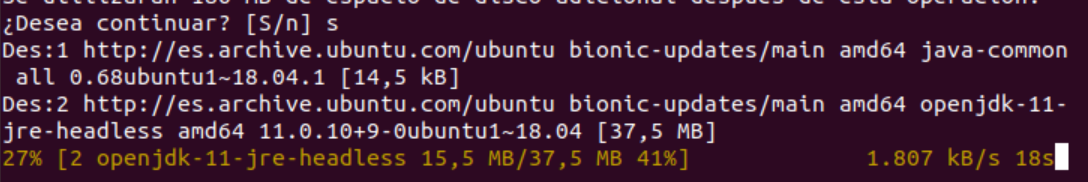
## Instalación y configuración de TomCat

Lo siguiente será instalar el servidor tomcat, para ello tecleamos el siguiente comando:

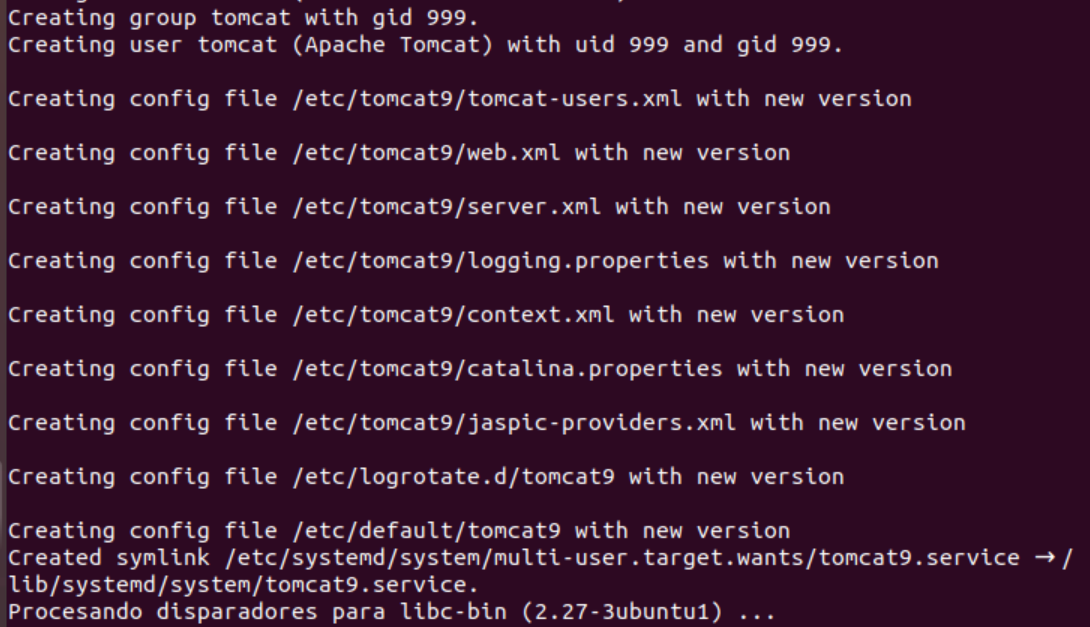
**Sudo apt install -y tomcat9 tomcat9-admin**



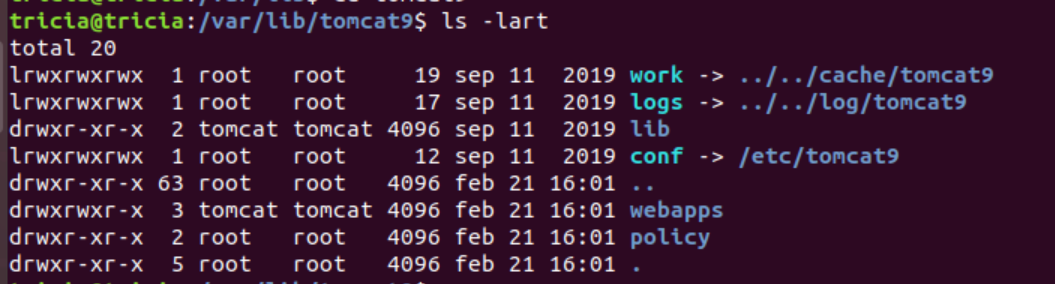
Le das a “s” y comienza la instalación



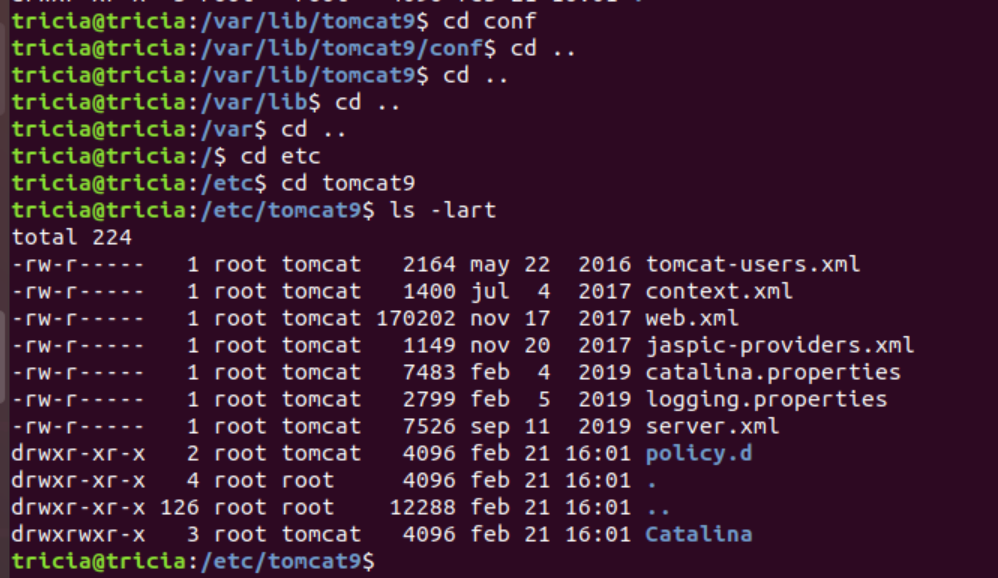
Las carpetas y recursos que está instalando:



En la siguiente captura podemos ver que tomcat se está instalado en var/lib/tomcat9 y haciendo un ls -l podemos ver que los ficheros de configuración, están en /etc/tomcat9.

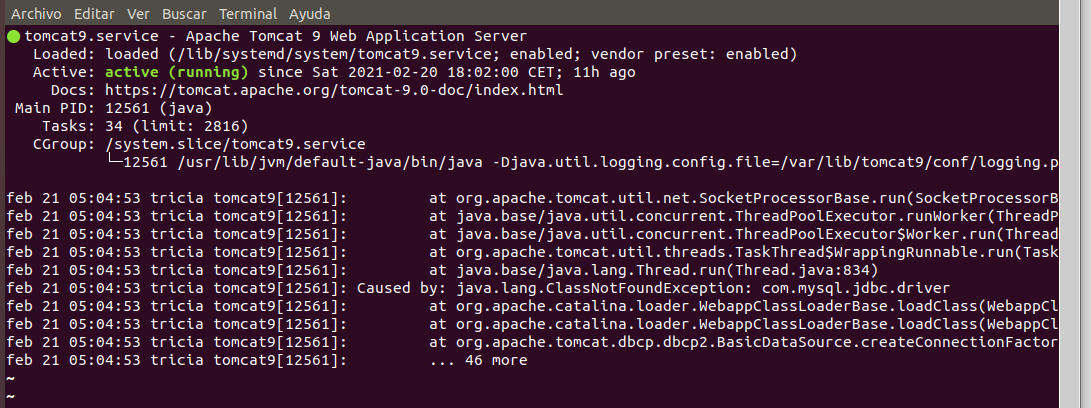


Podemos acceder desde var/lib/tomcat9/conf o desde etc/tomcat9



Cuando finaliza comprobamos el estado:

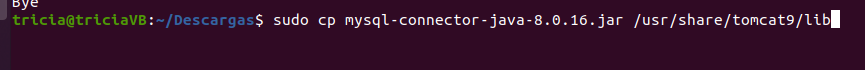
**Sudo systemctl status tomcat9**



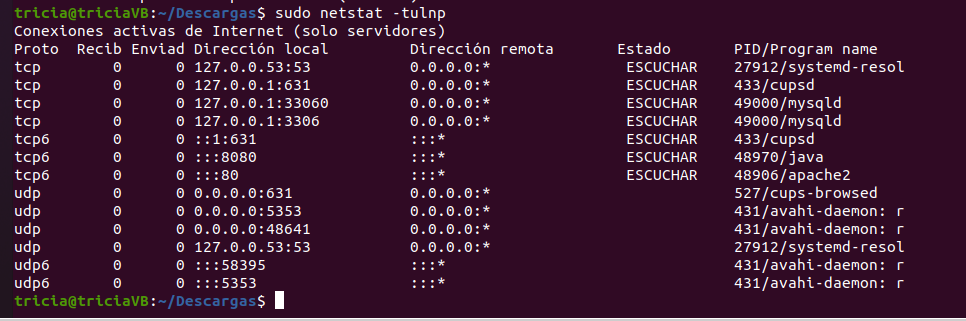
Podemos ver que tomcat no tiene acceso a la bbdd todavía.

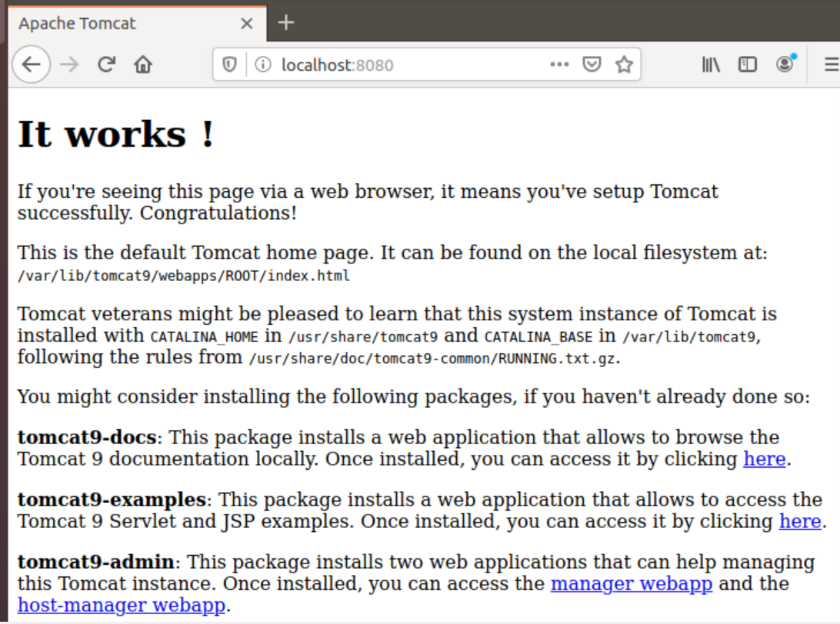
Una vez instalado Tomcat debemos darle acceso a la BBDD a través del conector mysql que hemos descargado.

Dicho conector debe estar en esta ruta:

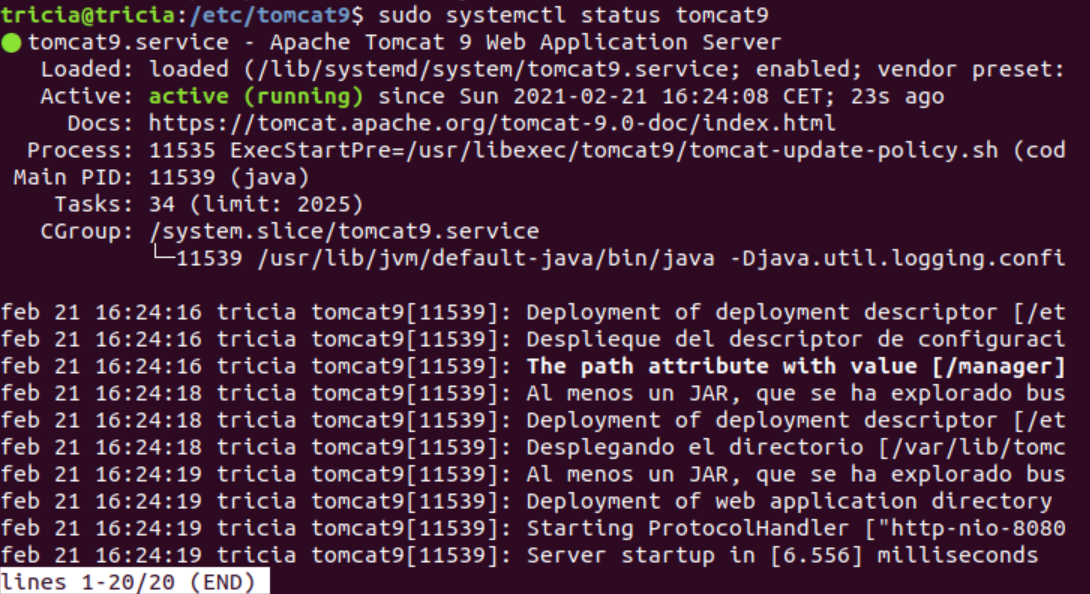


Si vamos a un navegador web y tecleamos **localhost:8080** podremos comprobar que la instalación esté bien hecha, esto puede variar en función del puerto que se haya habilitado para el acceso a tomcat. Para estar seguros podemos ejecutar **sudo netstat -tulpn,** tenemos el de mysql (mysqld), tomcat (java) y apache2.





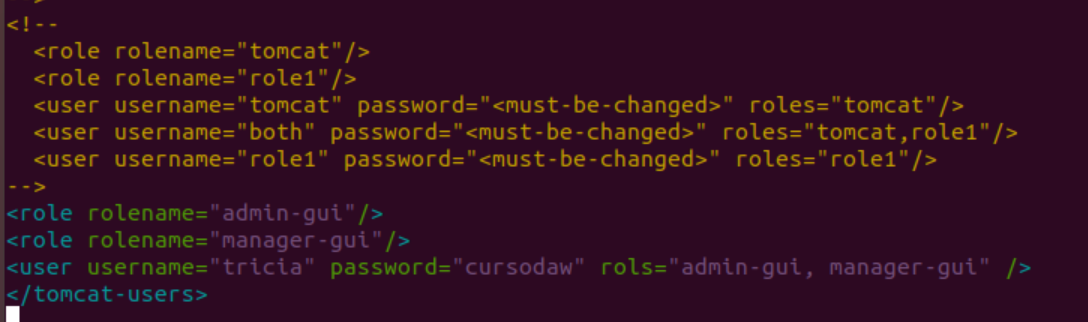
Lo siguiente es arrancar el servicio de tomcat:

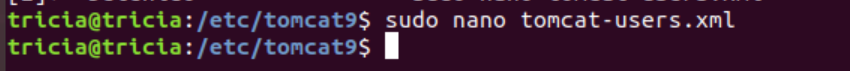


Ahora vamos a configurar tomcat para poder utilizar el webmanager:

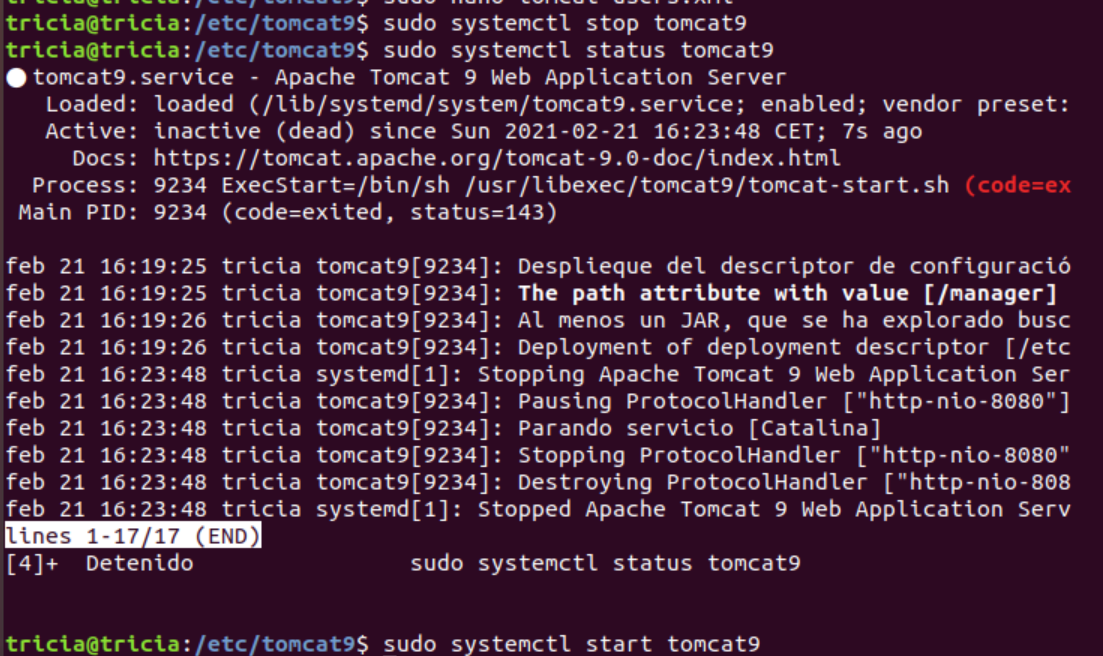
Lo primero es abrir el fichero de tomcat-users.xml y vamos a añadir nuestro usuario administrador con los roles admin-gui y manager-gui para poder acceder a webmanager:

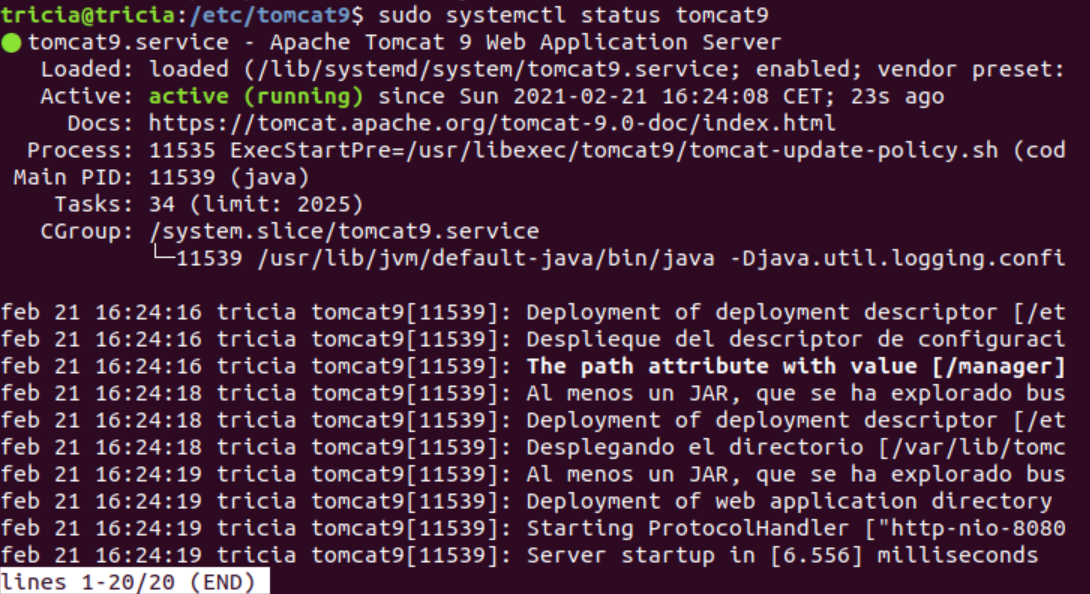




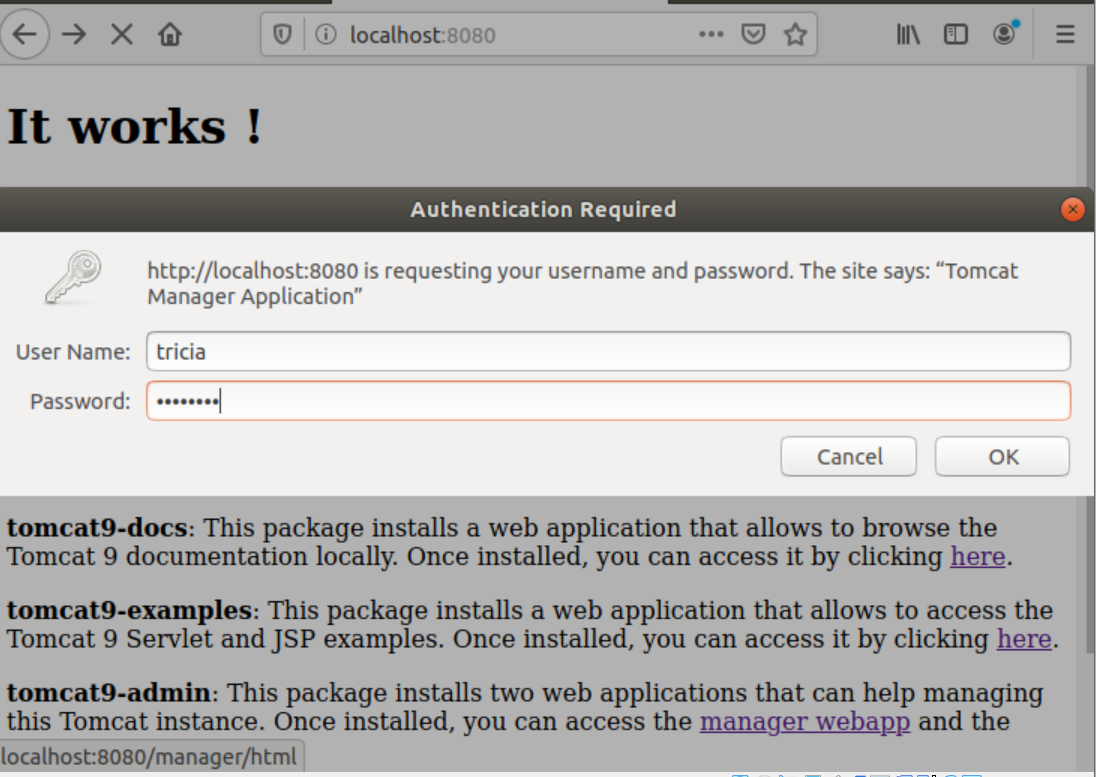


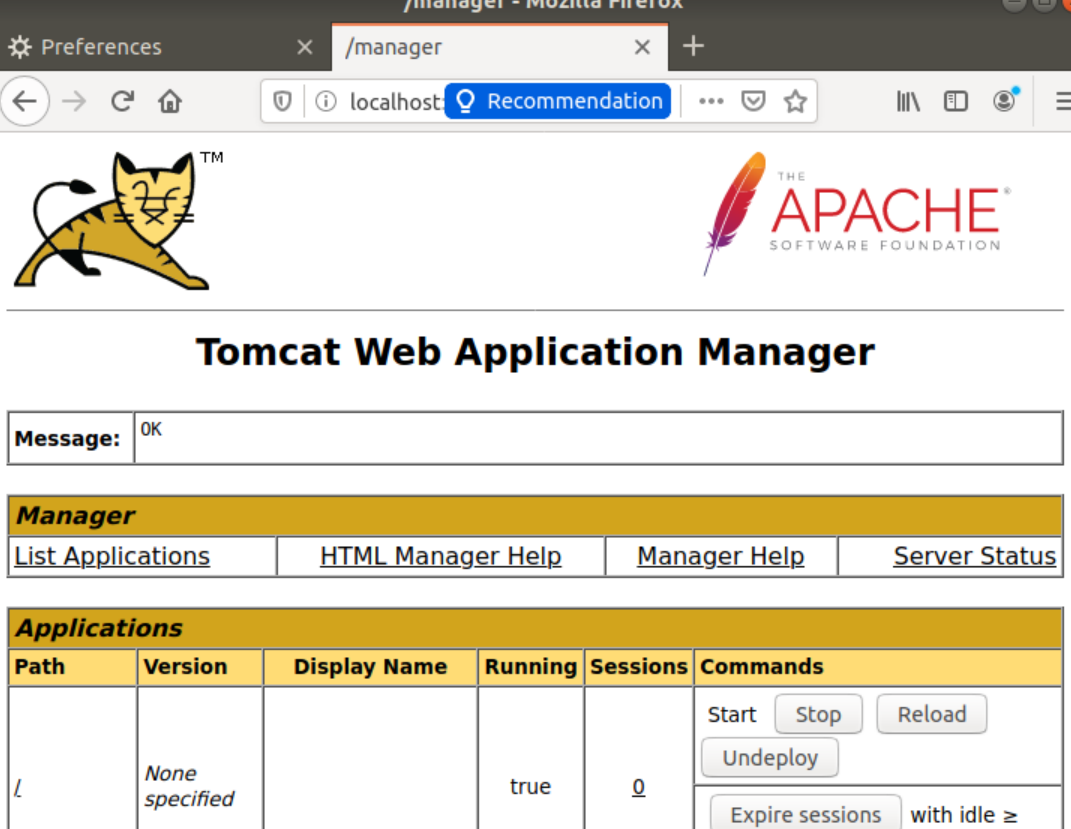
Reiniciamos el servicio:



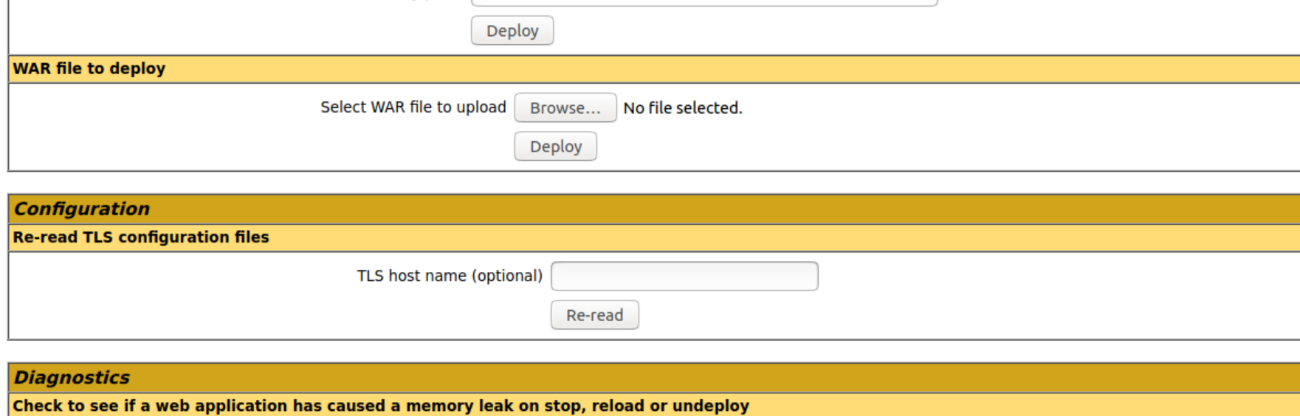


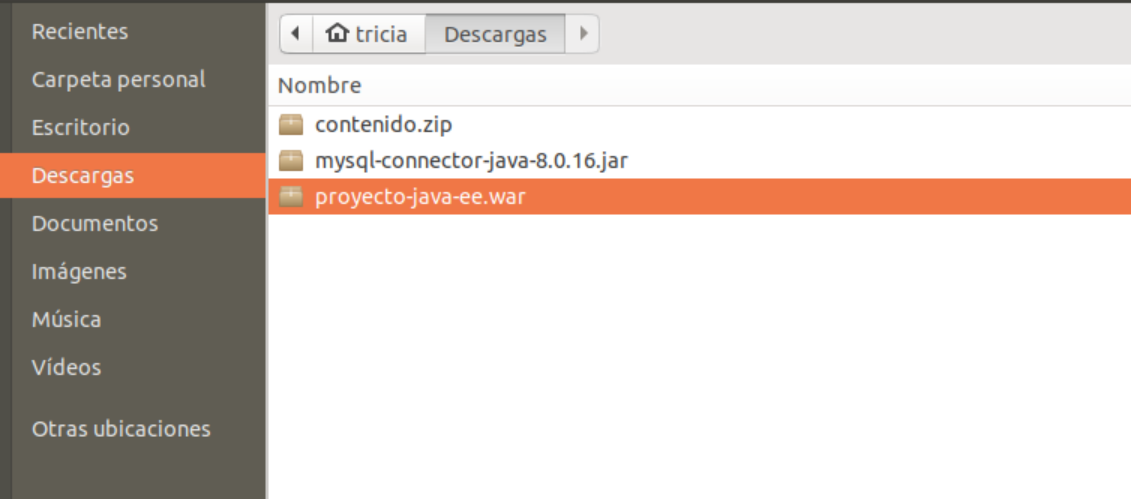
Ahora vamos a probar si ha tenido efecto, volemos a localhost:8080 y pulsamos sobre manager webapp:

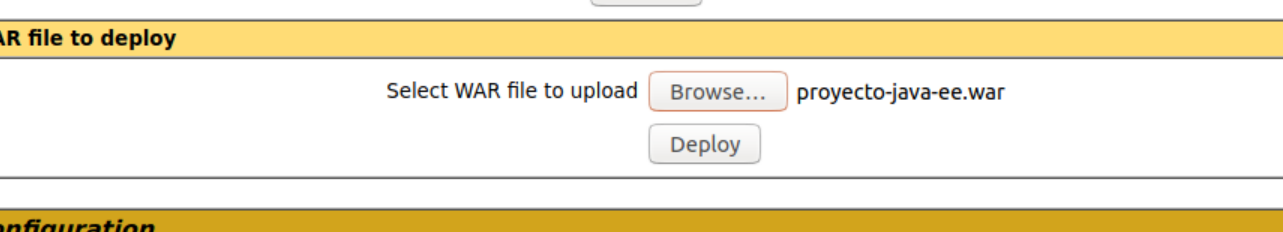




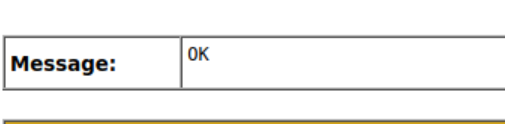
Ahora vamos a desplegar la aplicación .war desde tomcat para ver si lo hemos hecho bien:



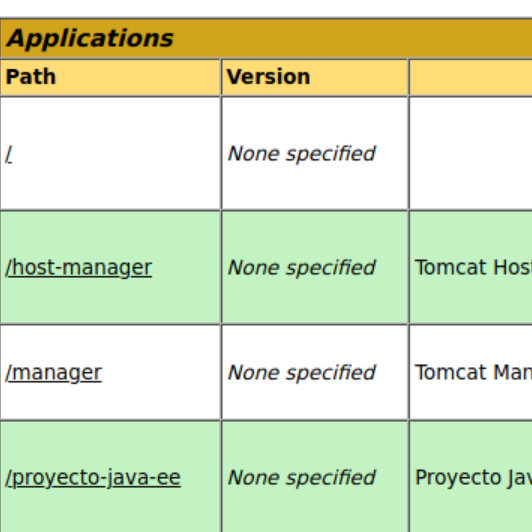




Si lo hemos hecho bien, veremos este mensaje:

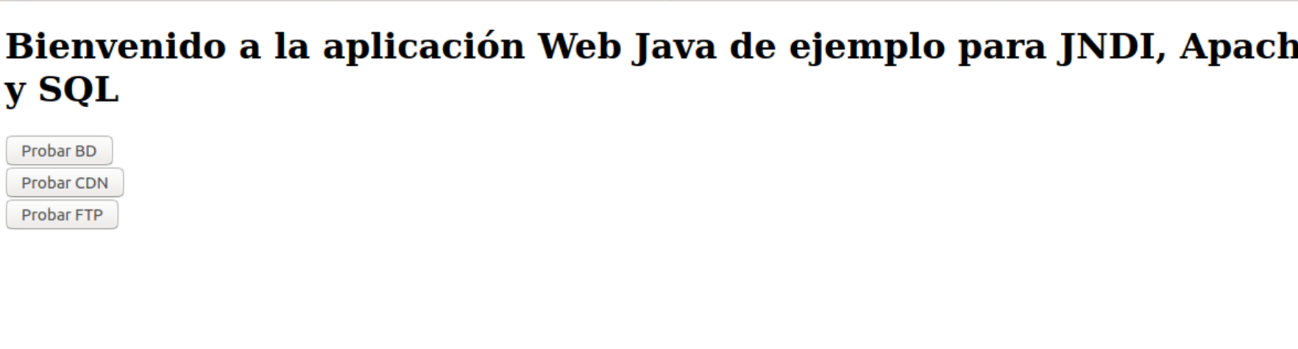


Y que se ha subido la aplicación a tomcat:



Para probarla, pinchamos sobre /proyecto-javaa-ee

Y si vemos esta pantalla es que lo hemos hecho bien:

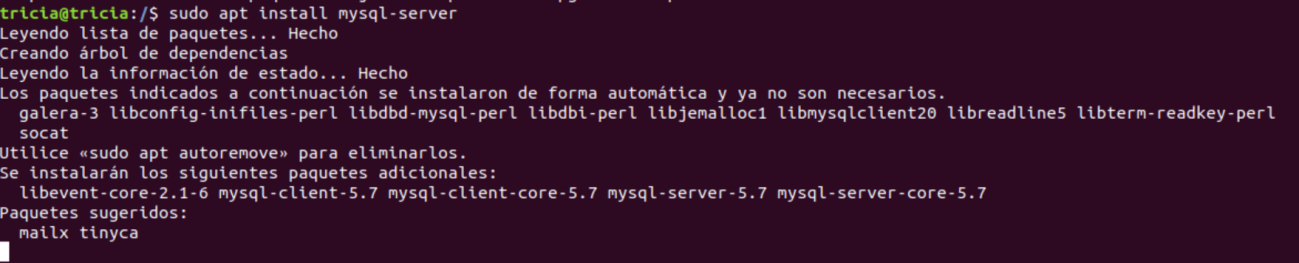


Hasta aquí ya tendríamos instalado tomcat y desplegada la aplicación. Ahora para poder utilizar todas sus funcionalidades necesitamos terminar de instalar y configurar la BBDD, el servidor Apache para el CDN y el FTP.

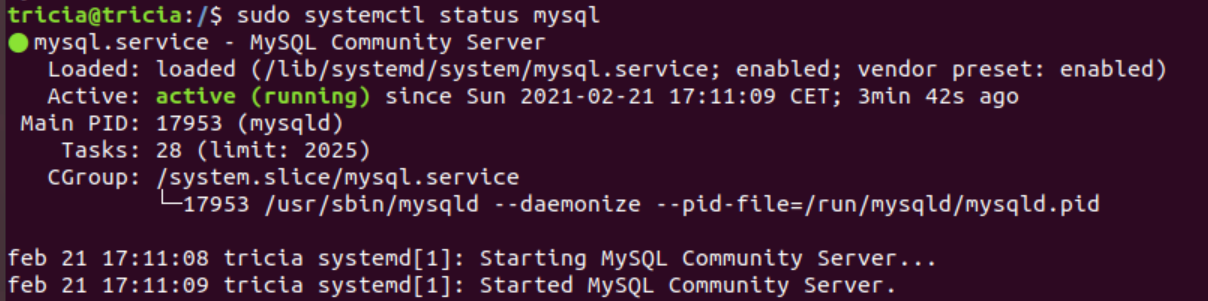
## La BBDD

Vamos a instalar el gestor de base de datos mysql:

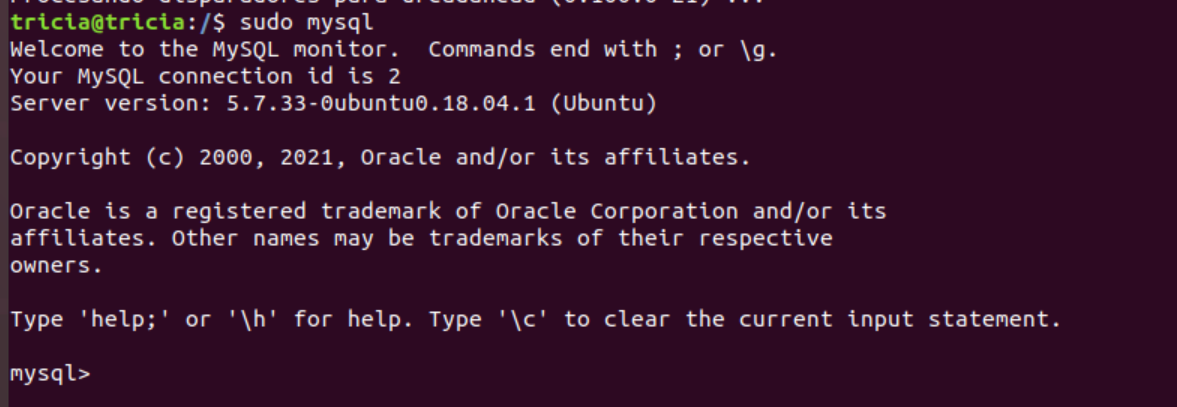
**sudo apt install mysql-server**



Y cuando acaba la instalación comprobamos el estado:

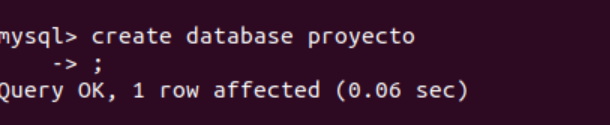


Accedo al gestor de bbdd con sudo mysql:



Ahora puedo crear mi base de datos y la tabla para la práctica con los siguientes comandos:

**CREATE DATABASE proyecto**



**CREATE TABLE PERSONAS (**

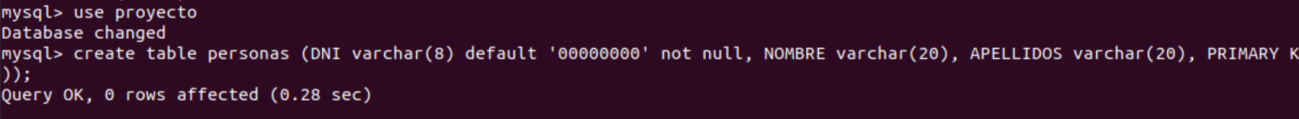
**DNI VARCHAR(8) DEFAULT '00000000' NOT NULL,**

**NOMBRE VARCHAR(20),**

**APELLIDOS VARCHAR(20),**

**PRIMARY KEY (dni)**

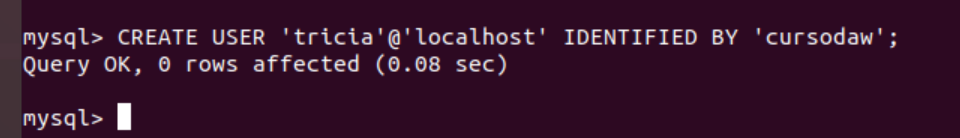
**);**

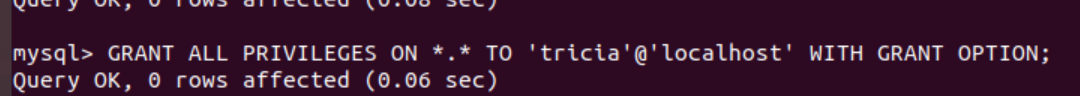


Lo siguiente es crear un usuario que pueda acceder a la base de datos y que podamos utilizar en nuestra posterior configuración de tomcat:

CREATE USER ‘nom\_usuario’@'localhost' IDENTIFIED BY ‘la\_password’;

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





Seguidamente vamos a copiar el driver a las librerías de tomcat:



Ahora que ya tenemos la base de datos creada, es hora de conectarla con tomcat. Para ello deberemos modificar el fichero **context.xml** que está en conf, para añadir un recurso que le indique a tomcat dónde está y cómo acceder a la base de datos:

**Sudo nano context.xml**

Le damos un nombre al recurso, esta vez viene dado por el ejercicio: jdbc/poolBaseDatos. Le indicamos el tipo de clase de este recurso, que es javax.sql.DataSource, algunos parámetros para configurar el máximo de conexiones, cuánto tiempo permanece logado hasta que se corte …etc. Lo siguiente es establecer el usuario y password con el que se accede a la base de datos, que será el usuario que hemos creado al dar de alta la bbdd y, por último, el nombre del Driver y la url de la base de datos que hemos creado.

