****

**Actividad 3. Tarea de equipo . Configuración de aplicación web**

**PILAR BERMEJO 2 DAW**

[**https://github.com/Pilupbp/despliegueAppWeb**](https://github.com/Pilupbp/despliegueAppWeb)



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. DNS. Se deberá poder acceder a la aplicación y al servidor FTP desde un servidor de DNS externo. Nota: La URL de cada servidor será distinta y no se permitirá que tenga ningún puerto asociado. Se deberá permitir el acceso vía HTTP y HTTPS. Nota: será necesario instalar un certificado digital.
2. Host. Se deberá poder acceder a la aplicación web Java y ésta deberá ser totalmente funcional. Nota: El servidor de aplicaciones Tomcat deberá estar correctamente configurado. Así mismo, se deberá poder acceder a los ficheros HTML y las imágenes en el servidor web.
3. FTP. Se deberá poder acceder al servicio de FTP descargar ficheros con un acceso anónimo y un acceso seguro. Se deberá poder acceder al servicio de FTP para subir ficheros, pero no como usuario anónimo. Nota: Es suficiente con tener 3 perfiles de usuario (anónimo, registrado y administrador).

Se pide que cada integrante genere un manual de instalación de su parte con una explicación precisa de cómo se debe configurar para obtener el resultado pedido en el Requerimiento 1.

**Consideraciones**

Para toda la actividad se valorará el orden y la claridad de la documentación, así como la facilidad de uso.

Para la entrega, se subirá un fichero comprimido con los manuales de instalación a la plataforma.

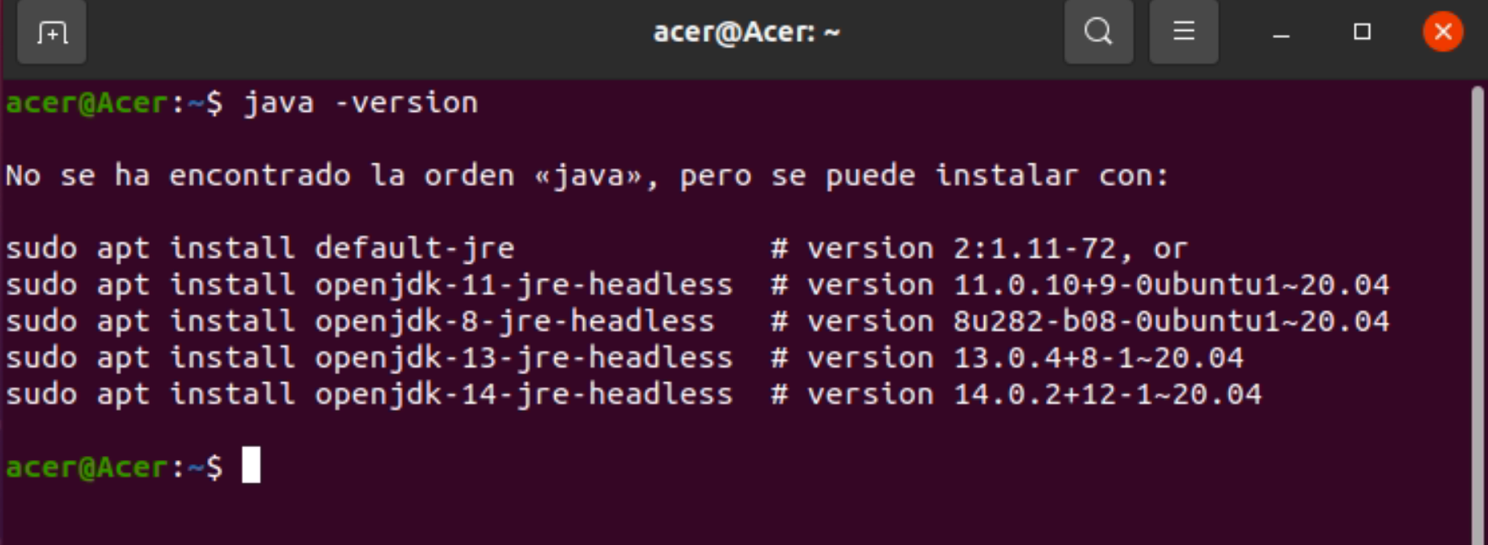
Para la entrega, es necesario la creación de un pequeño documento formal sobre la actividad (portada, explicación, etc.), indicando los componentes del equipo, las decisiones tomadas y la labor de cada integrante del equipo. También se valorarán la explicación de los problemas encontrados y su solución.

Sin asignar a grupo de trabajo.

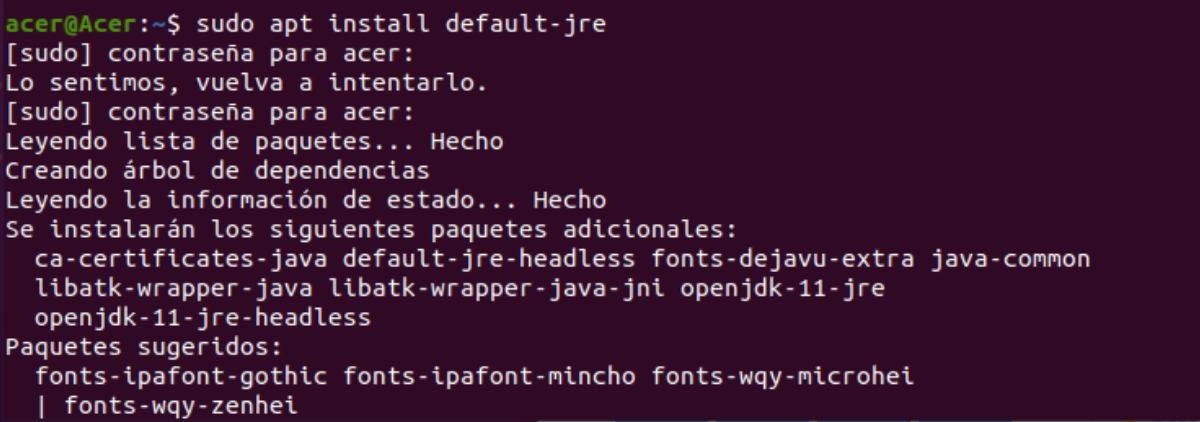
**PILAR BERMEJO – PARTE I**

Accedemos al equipo que hemos preparado para poder realizar una conexión remota cada parte del equipo y configurarlo en una única máquina.

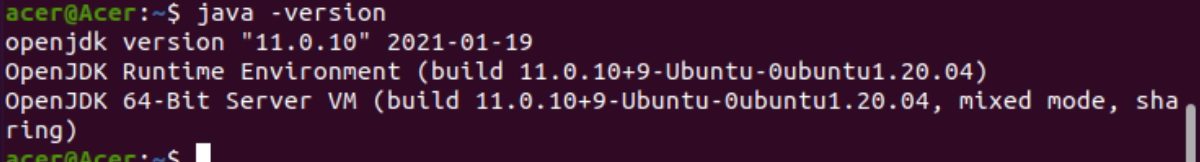
Lo primero que comprobamos para poder instalar las aplicaciones que nos hacen falta es la instalación de JAVA, comprobamos que no está.



Lo instalamos ***$ sudo apt install default-jre***

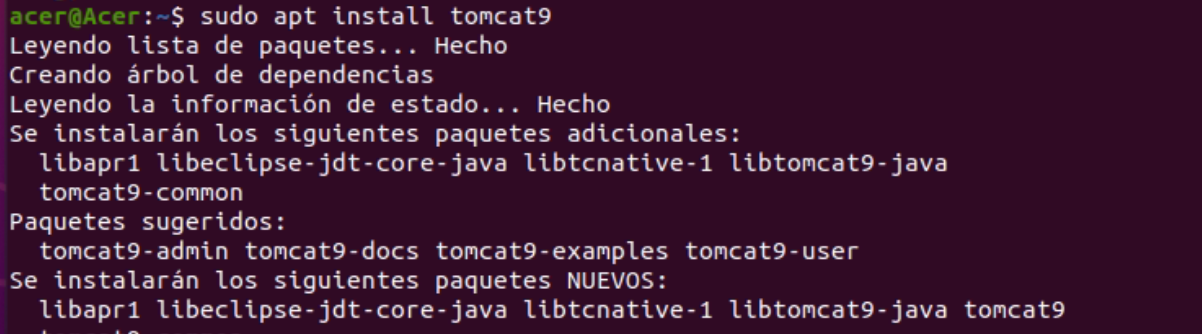


Comprobamos la versión: ***$ java -version***



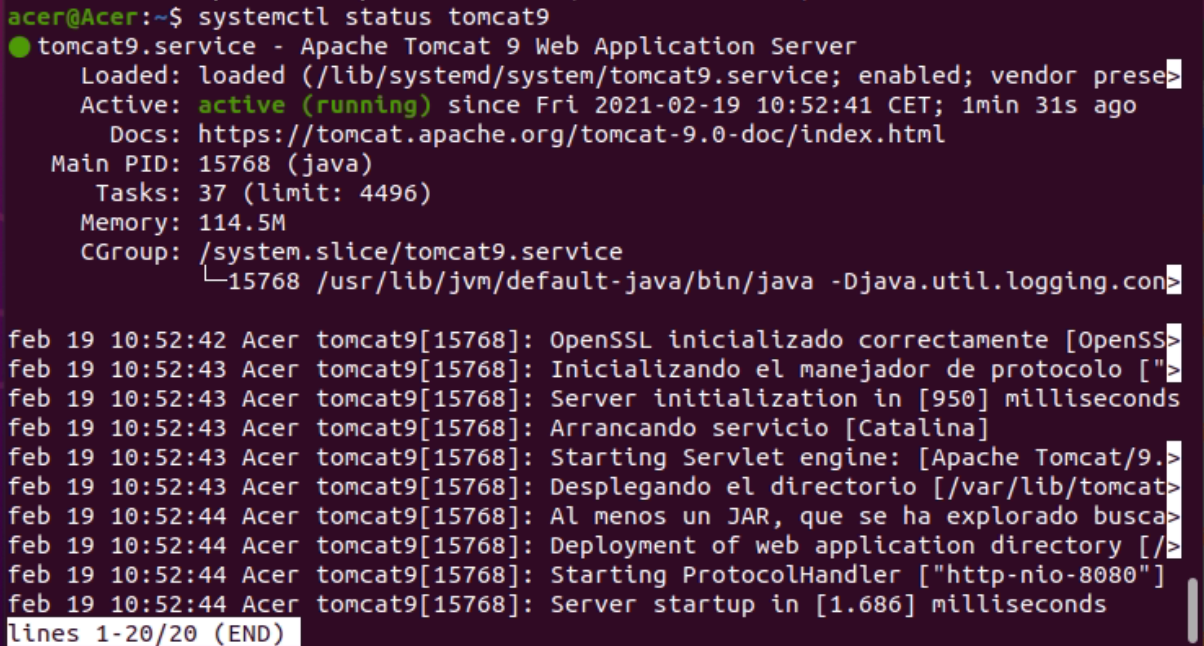
El siguiente paso es instalar APACHE TOMCAT , es un servidor web contenedor de Servlets que utilizamos para presentar aplicaciones Java.

Lo instalamos con el comando: ***$ sudo apt install tomcat9***



Una vez instalado vamos a comprobar el estado:

***$ systemctl status tomcat9***

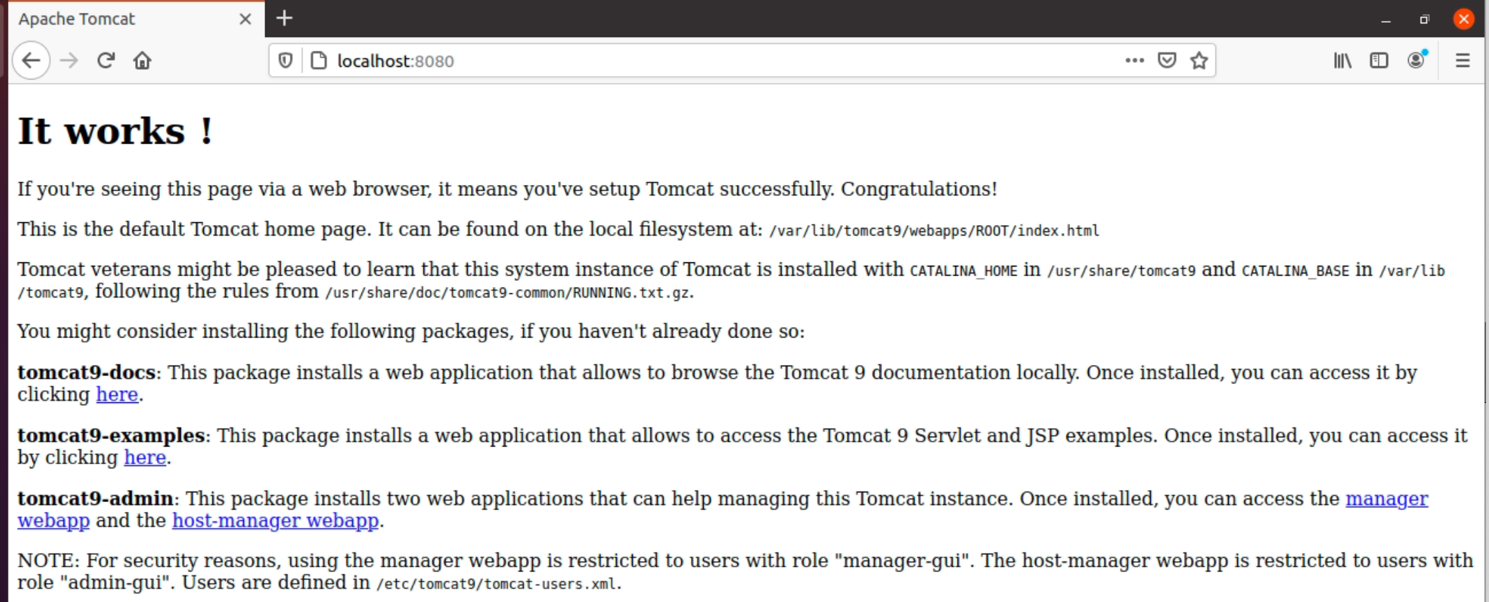


Vamos a acceder al Firewall y a habilitar el puerto por el que escucha Tomcat que es el 8080.

***$ sudo ufw allow 8080/tcp***



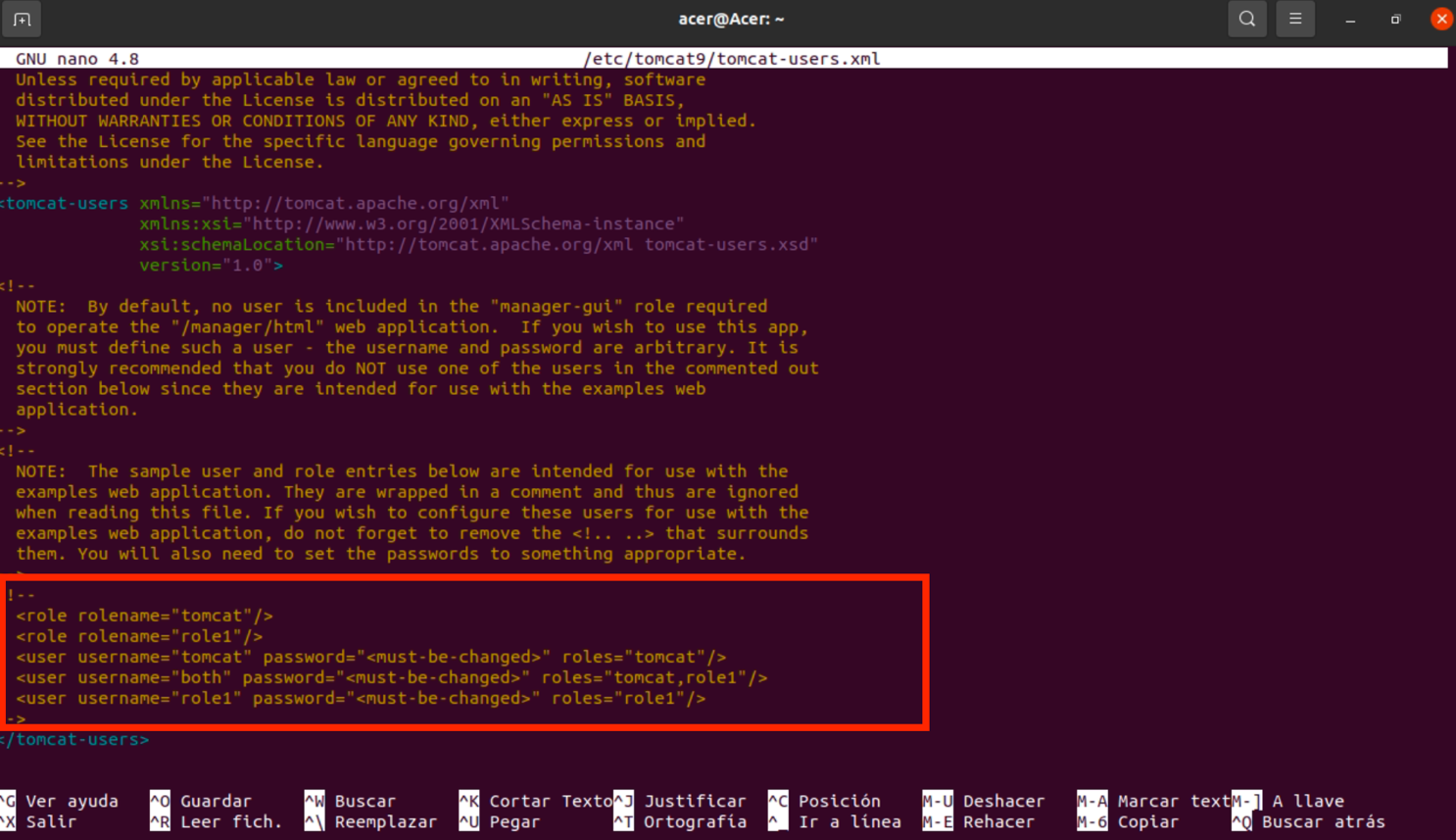
A continuación entramos en la url localhost.:8080 para desplegar el tomcat.

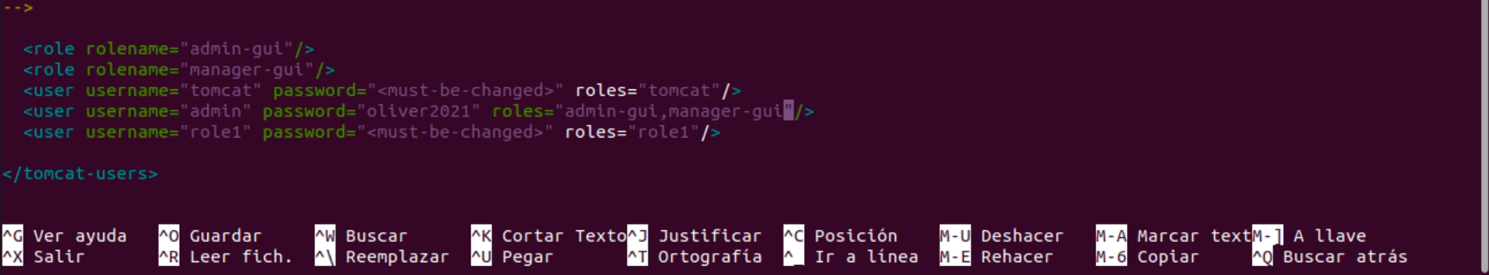


Algunas aplicaciones de Tomcat 9, como las aplicaciones administrativas, requieren el acceso autenticado de usuarios con cierto nivel de privilegios o roles. Por ejemplo, el **Gestor de Aplicaciones Web requiere usuarios con rol manager-gui,** mientras que el **Gestor de Máquina Virtual requiere el rol admin-gui**. Podemos crear los usuarios que consideremos con contraseña y con uno o ambos roles, en este caso será un solo usuario con ambos roles, para lo que editaremos el archivo **tomcat-users.xml**: Y lo editamos.

***$ sudo nano /etc/tomcat9/tomcat-users.xml***

Nos lo encontramos comentado, por lo que tenemos que descomentarlo y cambiar los roles.



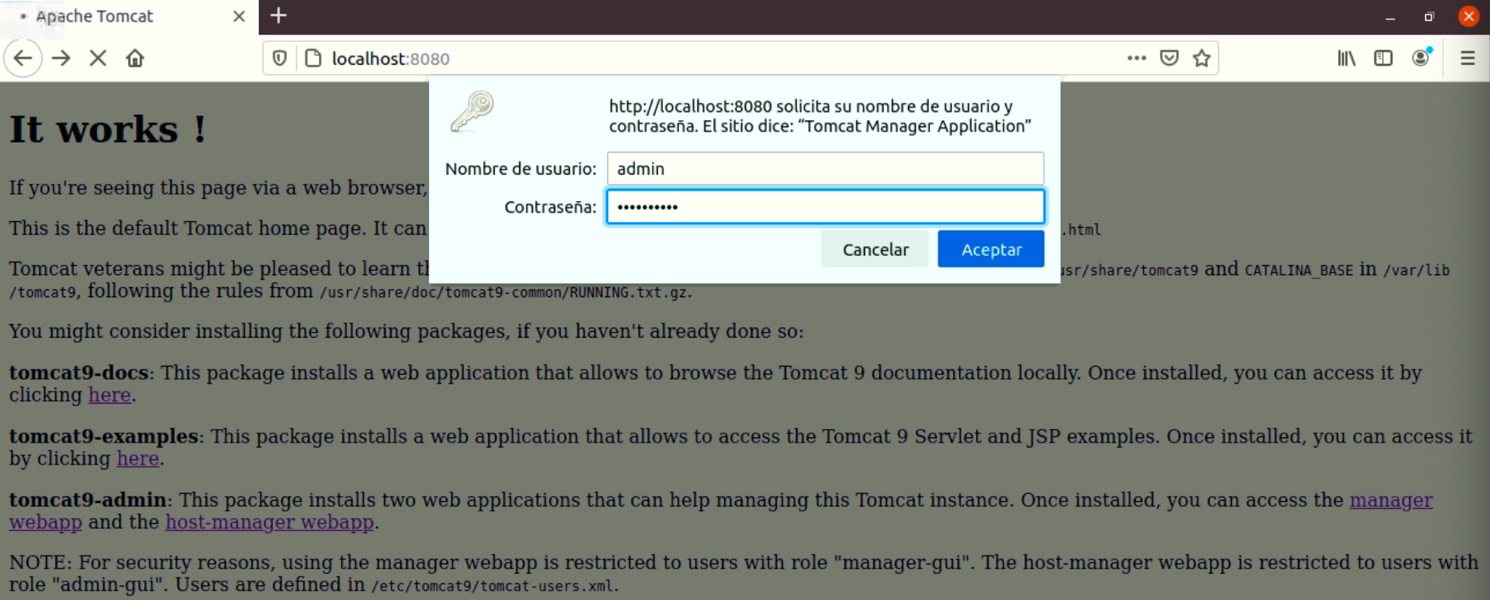


Una vez modificado, lo guardamos. Y paramos y restauramos el servicio para que aplique bien los cambios.

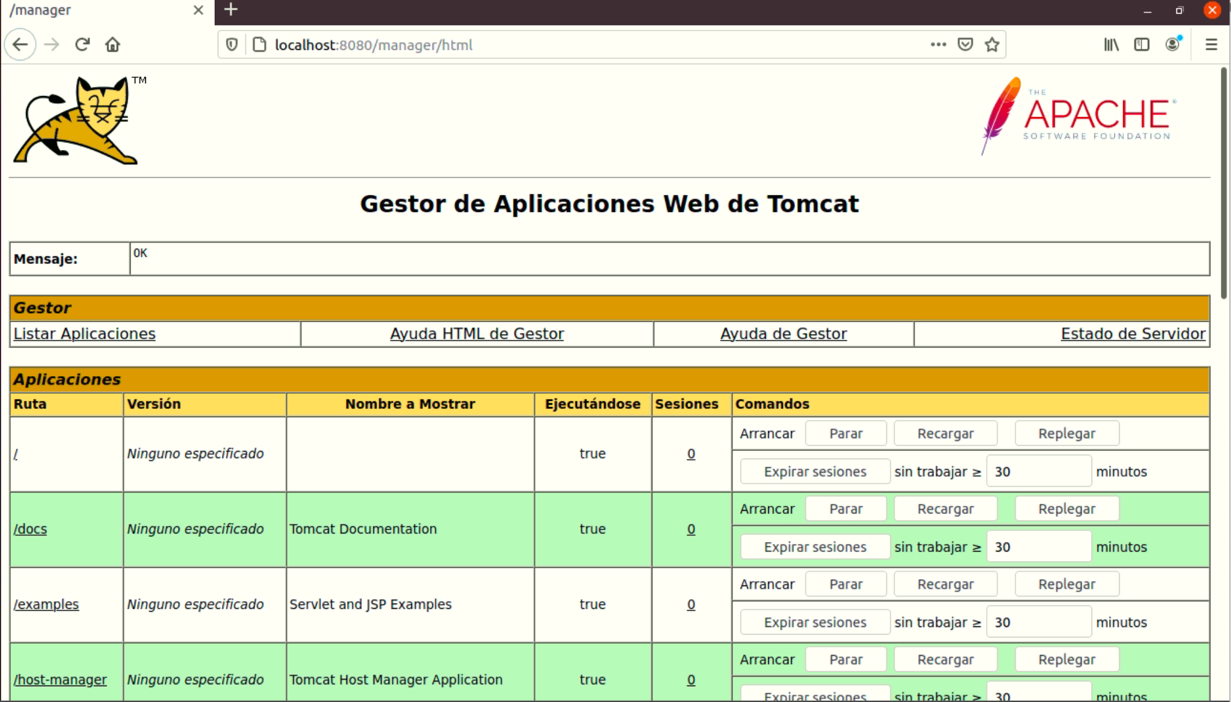
***$ service tomcat9 stop***

***$ service tomcat9 start***

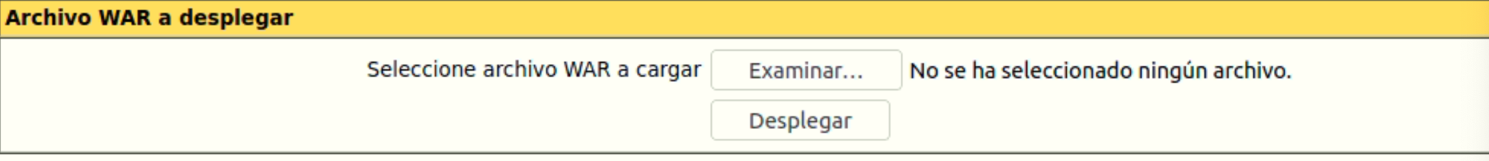
Accedemos a localhost:8080 con el usuario y contraseña que hemos configurado:

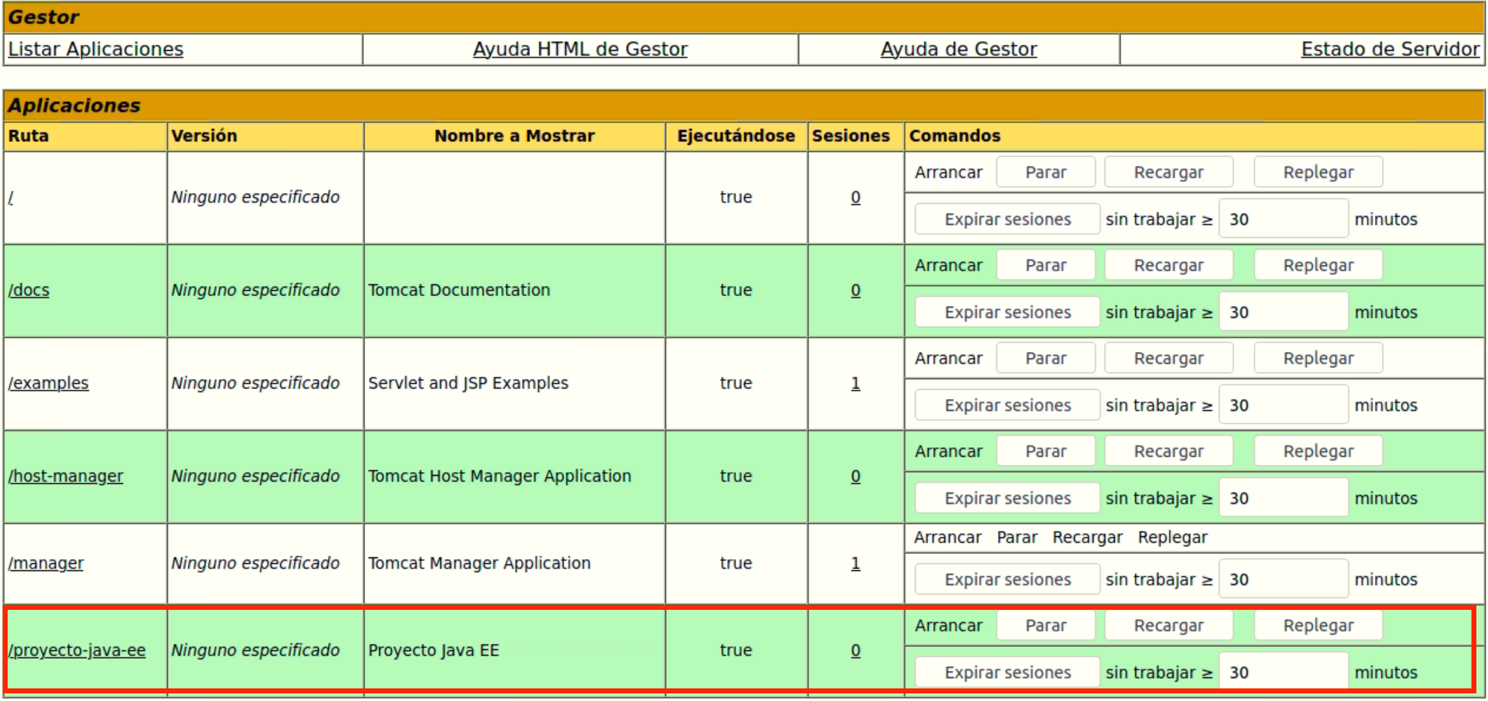


Entramos al gestor de aplicaciones de TOMCAT.



Cargamos el archivo .war que nos hemos descargado, necesario para desplegar la actividad.





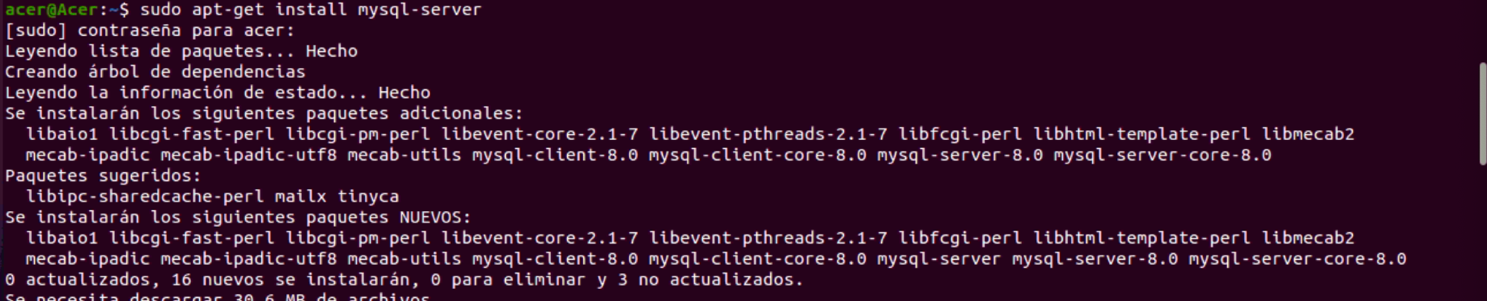
Y una vez que la hemos cargado, la ejecutamos:



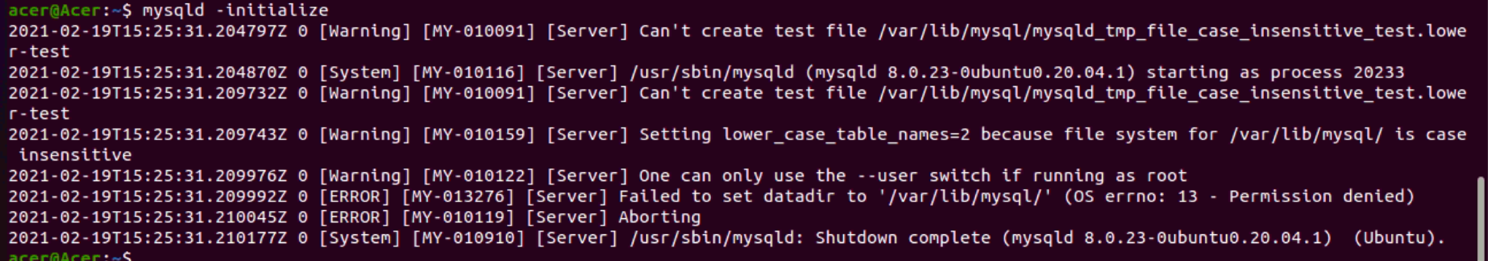
A continuación paramos el servicio para que siga mi compañera **LIDIA MARTINEZ** para la instalación de la BBDD con la creación de la BBDD y el recurso.

**LIDIA MARTINEZ – PARTE II**

***$ sudo apt-get install mysql-server***



Inicializamos el motor de base de datos con ***$ mysqld -initialize***



Comprobamos que el servicio se está ejecutando.

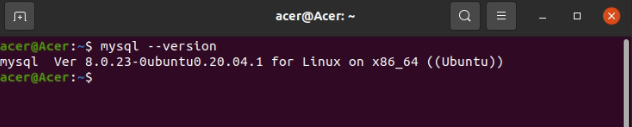
***$ systemctl status mysql.service***

******

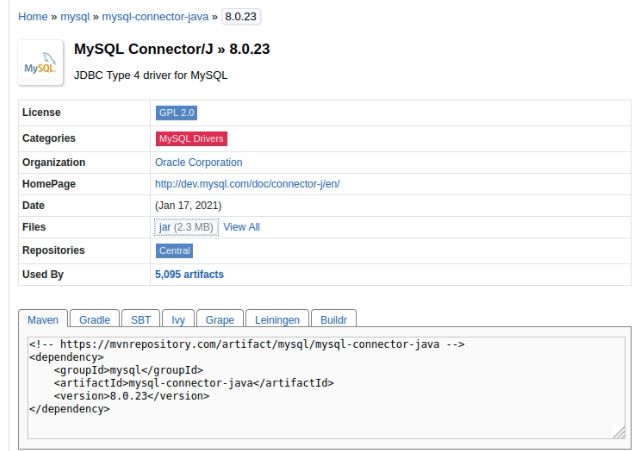
***$ sudo systemctl stop mysql****.*

***$ sudo systemctl start mysql****.*

A continuación, vamos a descargar las librerías necesarias para nuestra conexión de la BBDD. En este caso, vamos a comprobar la version de MySQL para ver cual debemos descargarnos.



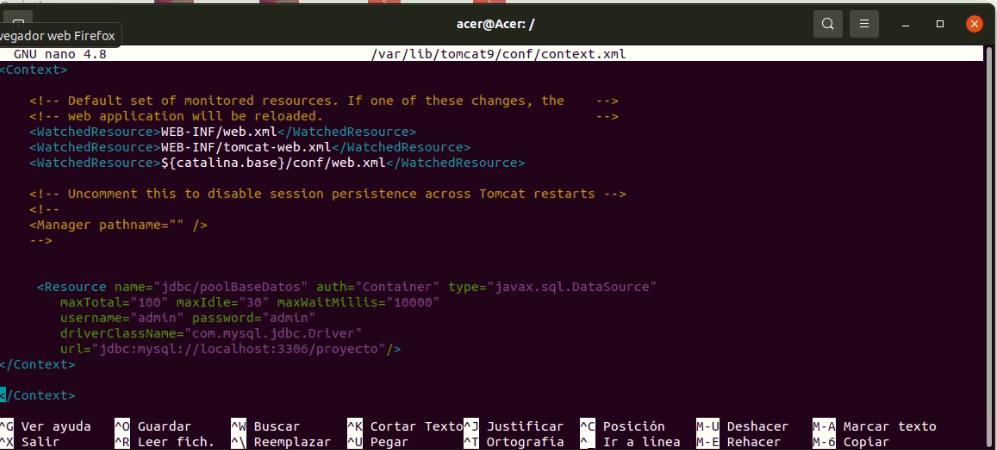
Vemos que nuestra version es la 8.0.23 por lo que vamos a buscar dicho archivo JAR para la conexión y añadirlo en la carpeta lib del tomcat.



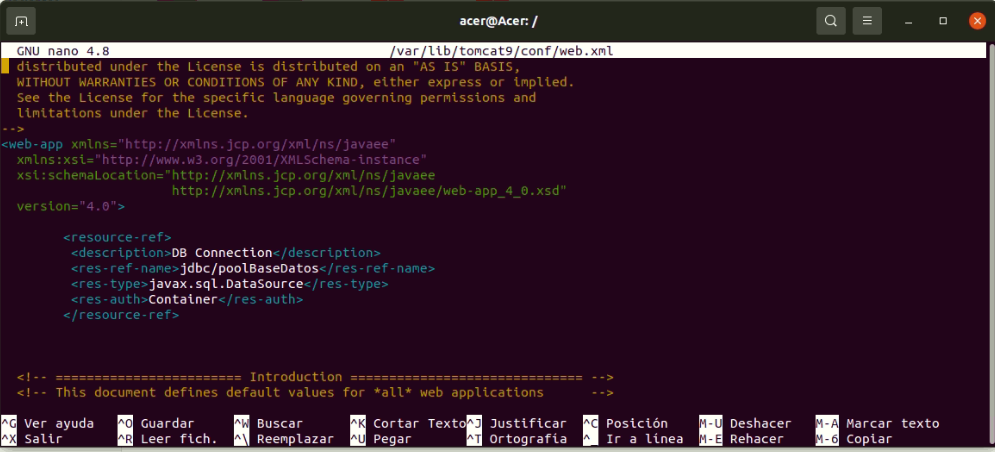


El siguiente paso, es editar el archivo context.xml del Tomcat (/conf/context.xml) y añadir el datasource para después poder recuperarlo desde java con JNDI.

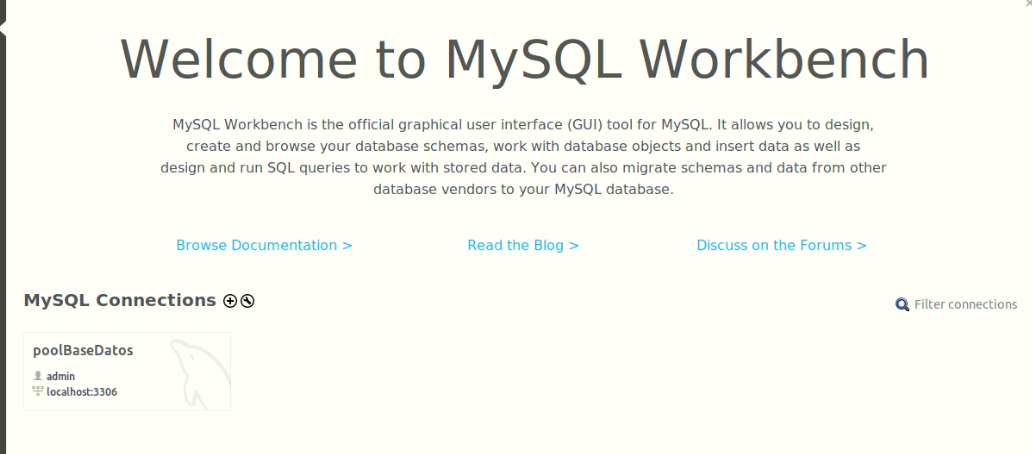
Añadimos lo siguiente:

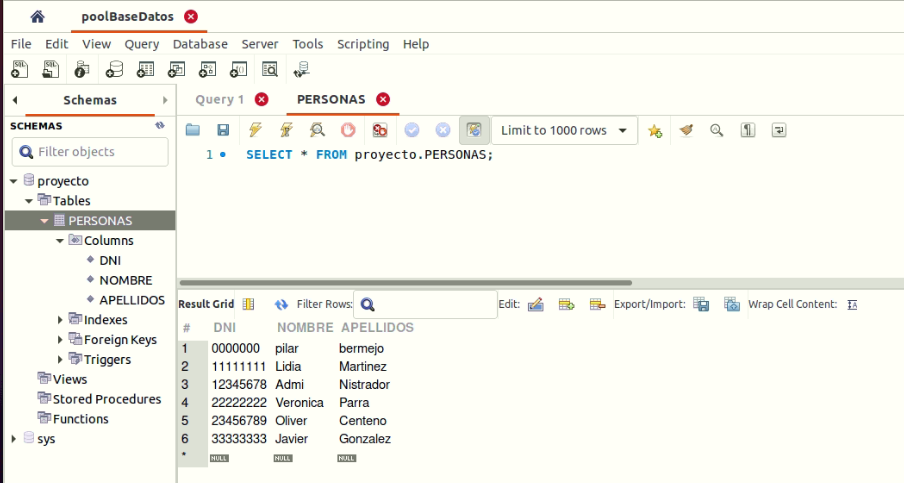


Por otra parte en el archivo web.xml:



A continuación, vamos a crear nuestra BBDD proyecto en MySQL gracias al WorkBench.



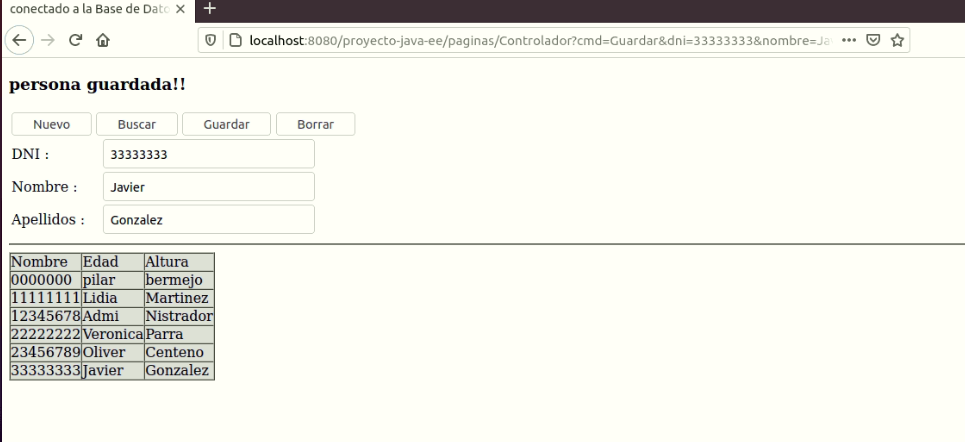


Aquí, podemos ver nuestra base de datos creada.

Una vez configurado todo lo anterior, solo nos quedaría ir al tomcat para ver nuestra base de datos.



Una vez clicamos vemos como nos aparecen nuestras tablas y la base de datos está totalmente operativa.



**VERÓNICA PARRA – PARTE III**

Instalamos el servidor Apache en nuestro equipo con el comando

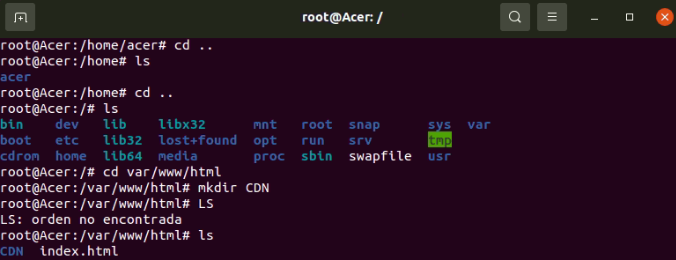
***$* sudo apt install apache2**

A continuación, entramos como super usuario, ya que tenemos que modificar archivos de configuración y necesitamos tener todos los privilegios.



En el terminal, vamos al directorio **/var/www/html**. Dentro de la carpeta html, con el comando mkdir, creamos un directorio al que llamaremos **CDN.**

Listamos el contenido para comprobar que se ha creado correctamente:



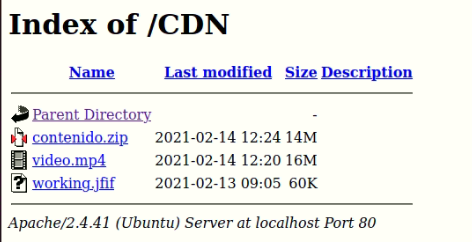
Ahora, vamos directorio en el que se encuentran los tres recursos a desplegar. En mi caso, los recursos los guardé en una carpeta a la que llamé “apache”, dentro de la carpeta del proyecto, para tenerlos más unificados.



Movemos los archivos al directorio que acabamos de crear. Utilizaremos el comando **mv** seguido del archivo que queremos mover y la ruta de destino:

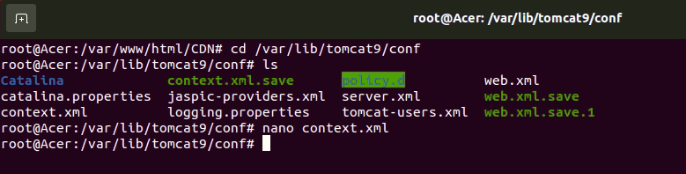


Si entramos en nuestro localhost e indicamos la carpeta, vemos que los archivos aparecen listados:

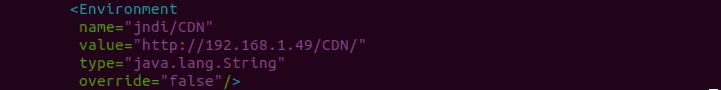


Ahora vamos a crear la variable de entorno para enlazar la aplicación desplegada en Tomcat con los recursos que tenemos en el directorio CDN.

Seguimos la ruta **/var/lib/tomcat/conf** y editamos el archivo **context.xml.**

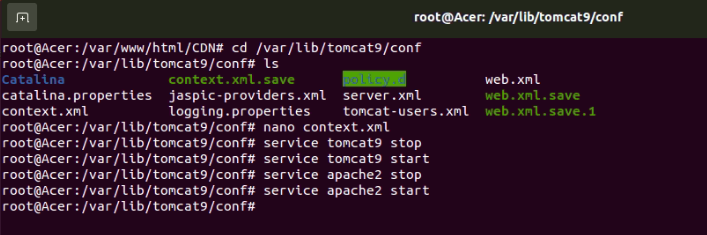


Entramos en el editor y añadimos las líneas:



Le indicamos el nombre del recurso que debe buscar y la dirección ip en la que encontrará los recursos.

Guardamos los cambios y paramos los servidores y volvemos a arrancarlos para que la modificación que hemos realizado en el xml se ejecute.



Abrimos el navegador y accedemos a la aplicación. Pulsamos el botón “CDN” y veremos todos los archivos desplegados en la página.

