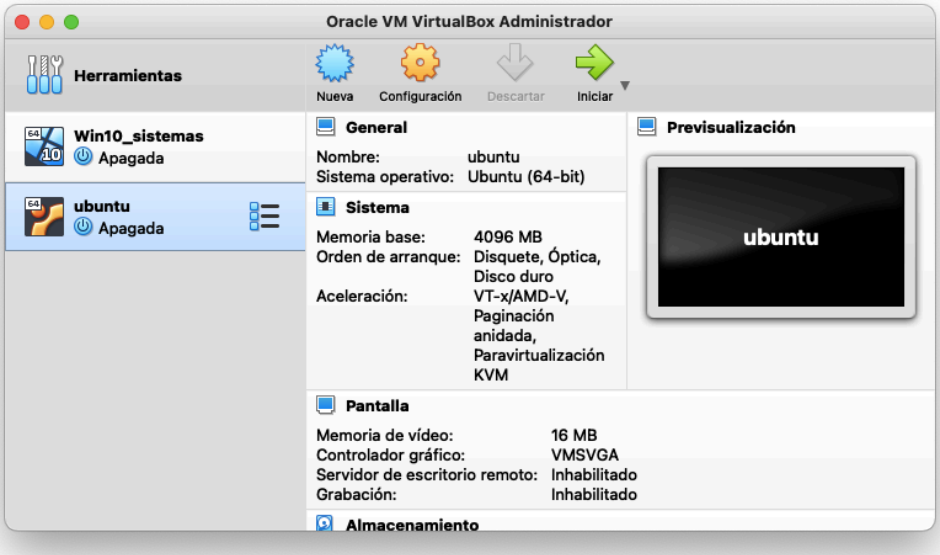
Actividad 2: Configuración de servidor

PUNTO 1: VERIFICACIÓN DE INSTALACIONES

Lo primero de todo que vamos a hacer es crear una maquina virtual dentro de nuestro ordenador donde instalaremos “Ubuntu” para poder realizar toda la configuración desde ahí.



Vamos a instalar los paquetes que se mencionan en el primer punto de la actividad. Tenemos que instalar los siguientes programas: Java, Apache, Tomcat, OpenSSH y MariaDB.

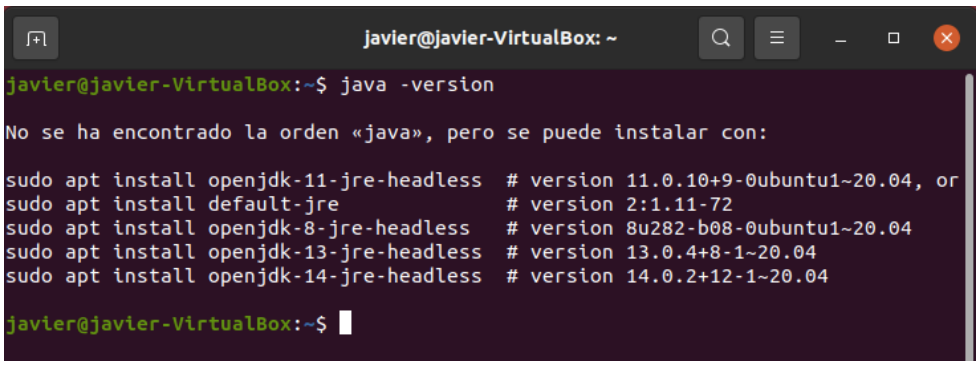
Para ello ejecutamos nuestra máquina virtual con Ubuntu. En mi caso no tenia Ubuntu descargado, así que me descargue el fichero .iso desde la página web y lo introduje dentro de la máquina virtual.

PUNTO 1: INSTALAR JAVA

Lo primero de todo es ver si Java ya está instalado en el dispositivo. En mi caso ya estoy seguro que no lo está, puesto que acabo de instalar Ubuntu desde cero. De todas formas buscamos la versión actual de Java en el ordenador, con el siguiente comando:

java -version

La terminal nos muestra el siguiente mensaje, diciendo que efectivamente NO está instalado y nos da opciones para proceder a la instalación:

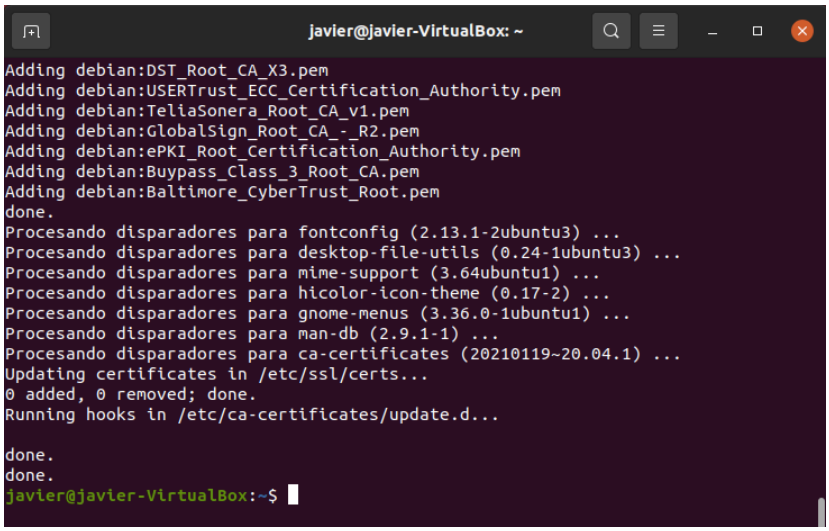


Esa segunda linea que se nos muestra ya nos está indicando el comando que podemos emplear para instalar Java:

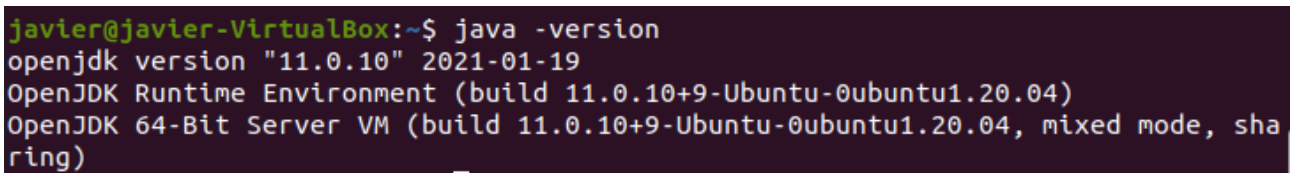
sudo *apt install default-jre*

Le añadimos la coletilla de “sudo” al principio del comando para que no se nos queje y realice esta acción con permisos de super usuario. Nos pedirá la contraseña de super usuario (la del usuario del ordenador) y una vez que lo hagamos, empezará a instalar Java (en Linux ademas se mostrará una barra de progreso).

Una vez finalizado, obtendremos el siguiente mensaje de “done”:



Con ello habremos terminado la instalación de Java. Podemos volver a escribir el primer comando que mencionábamos y esta vez nos dirá la versión instalada:

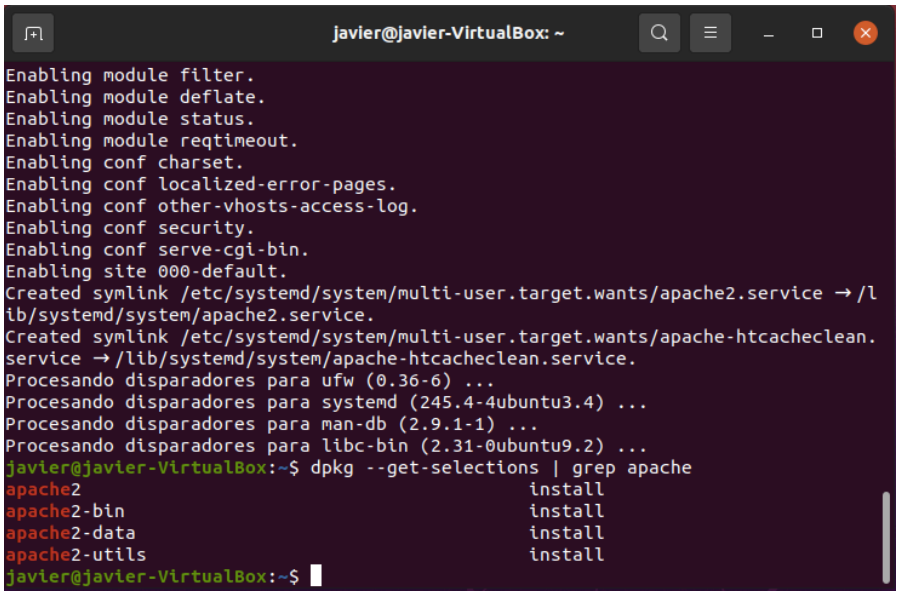


PUNTO 2: INSTALAR Y CONFIGURAR APACHE

Ahora vamos a proceder a instalar Apache. Para ello sencillamente tenemos que introducir el comando que procede a realizar la instalación de apache:

sudo *apt install apache2*

Le añadimos nuevamente “sudo” para no tener problemas de permisos. Una vez terminado el proceso, podemos comprobar con el siguiente comando que efectivamente ya tenemos los archivos de apache instalados en el ordenador:



Antes de la instalación, este mismo comando no nos devolvía nada, de forma que ahora ya sabemos que está efectivamente instalado. Vamos ahora a proceder a configurar Apache para que salga por el puerto adecuado y asegurarnos de que esté abierto el firewall y sea accesible desde el exterior.

Como vimos en clase, configuramos Apache para que utilice el puerto 80. En mi caso ya viene en ese puerto por defecto:

```
1 # If you just change the port or add more ports here, you will likely also
2 # have to change the VirtualHost statement in
3 # /etc/apache2/sites-enabled/000-default.conf
4
5 Listen 80
6
7 <IfModule ssl_module>
8     Listen 443
9 </IfModule>
10
11 <IfModule mod_gnutls.c>
12     Listen 443
13 </IfModule>
14
15 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Ahora tenemos que revisar si este puerto, el 80, está abierto en el Firewall y es accesible desde el exterior. Para ello seguiremos los siguientes pasos:

1. Actualizamos el repositorio de paquetes de Ubuntu, para asegurarnos poder instalar UFW (Uncomplicated FireWall, el sistema de Firewall por defecto en Ubuntu), la herramienta mediante la cual vamos a poder hacer esta configuración. Para ello escribimos **sudo apt update**.
2. Cuando nos confirme que los paquetes están correctamente actualizados, procedemos a instalar UFW mediante el comando **sudo apt install ufw -y**. Como vemos en la imagen, en este caso ya lo teníamos instalado (debería venir siempre por defecto instalado en Ubuntu).

```
javier@javier-VirtualBox:~$ sudo apt install ufw -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ufw ya está en su versión más reciente (0.36-6).
fijado ufw como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
javier@javier-VirtualBox:~$
```

3. Ahora lo activamos. Lo primero miramos si UFW está corriendo o si está parado. En nuestro caso vemos que efectivamente está corriendo y activo. Lo hacemos con el siguiente comando:

```
javier@javier-VirtualBox:~$ systemctl status ufw
● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: e>
     Active: active (exited) since Sat 2021-02-13 13:31:15 CET; 3h 9min ago
       Docs: man:ufw(8)
   Main PID: 234 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 4654)
     Memory: 0B
     CGroup: /system.slice/ufw.service
```

4. Con el servicio corriendo, lo activamos:

```
javier@javier-VirtualBox:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
```

5. Con esto ya tenemos el firewall habilitado. Ahora tenemos que decir qué puertos queremos que habilite o deje pasar desde el exterior, que serán aquellos que vayan a usar nuestros servidores Apache, Tomcat y MariaDB. Gracias a UFW esto se puede hacer de una forma muy sencilla. Lo primero listamos los perfiles que tenemos instalados, mediante un sencillo comando:

```
javier@javier-VirtualBox:~$ sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

6. Vemos como tenemos Apache instalado, al que previamente ya lo hemos configurado para que use el puerto 80. Por lo tanto, ahora solamente tenemos que decirle a UFW que permita Apache, de forma que añade las reglas necesarias para realizar dicha acción:

```
javier@javier-VirtualBox:~$ sudo ufw allow apache
Regla añadida
Regla añadida (v6)
javier@javier-VirtualBox:~$
```

7. Podemos ahora revisar qué esta permitiendo el firewall. Vemos como Apache está permitido. Si nos queremos cerciorar aún mas del puerto del que estamos hablando, podríamos hacer el comando **sudo ufw app info Apache.** Vemos como efectivamente, está actuando sobre el puerto 80 en el que está configurado Apache:

```
javier@javier-VirtualBox:~$ sudo ufw status
Estado: activo

Hasta                      Acción      Desde
-----                      ------      -----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
```

```
javier@javier-VirtualBox:~$ sudo ufw app info Apache
Perfil: Apache
Título: Web Server
Descripción: Apache v2 is the next generation of the omnipresent Apache web
server.

Puerto:
  80/tcp
```

PUNTO 3: INSTALAR Y CONFIGURAR TOMCAT

Ahora procedemos a instalar Tomcat. Lo primero sería actualizar el gestor de paquetes, pero ya lo hemos hecho para Apache, así que podemos proceder directamente a la instalación de Tomcat. Para ello usamos el comando:

sudo apt install tomcat9

```
Creating config file /etc/default/tomcat9 with new version
Created symlink /etc/systemd/system/multi-user.target.wants/tomcat9.service → /l
ib/systemd/system/tomcat9.service.
Procesando disparadores para rsyslog (8.2001.0-1ubuntu1.1) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9.2) ...
javier@javier-VirtualBox:~$
```

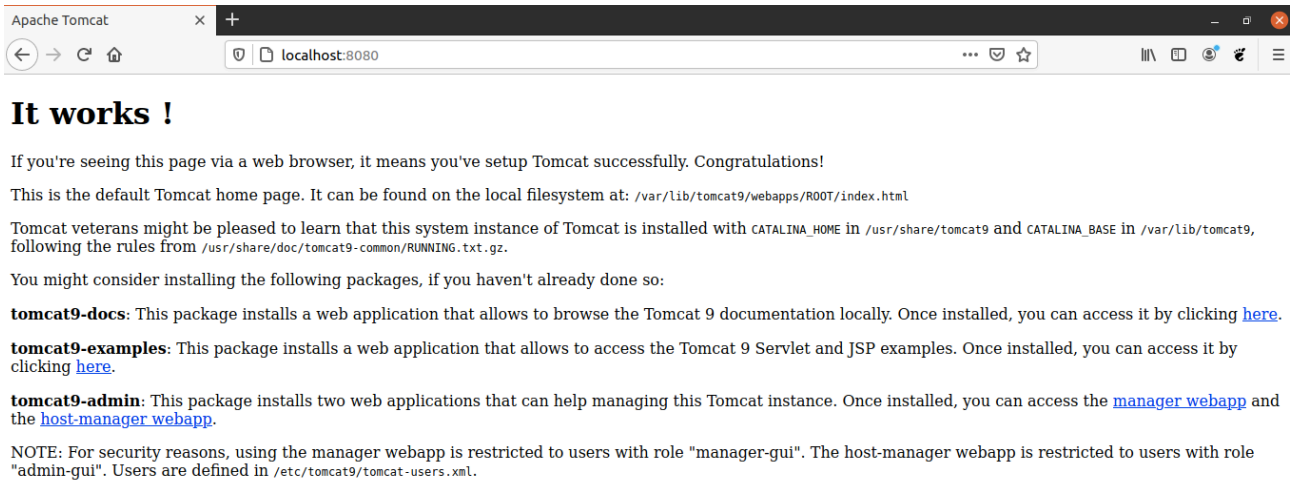
Con tomcat instalado, podemos mediante un comando ver el puerto que está utilizando. Por defecto, Tomcat utiliza el puerto 8080. Como podemos ver en la imagen, el puerto 80 ya sabemos que es Apache y, efectivamente, tenemos activo el puerto 8080:

```
javier@javier-VirtualBox:~$ ss -ltn
State    Recv-Q   Send-Q     Local Address:Port     Peer Address:Port  Process
LISTEN   0        4096       127.0.0.53%lo:53           0.0.0.0:*
LISTEN   0        5              127.0.0.1:631          0.0.0.0:*
LISTEN   0        100                  *:8080               *:*
LISTEN   0        511                  *:80                 *:*
LISTEN   0        5                [::1]:631              [::]:*
```

Añadimos ahora las reglas al UFW para decirle que permita los accesos a ese puerto desde cualquier lugar. Lo hacemos igual que con Apache:

```
javier@javier-VirtualBox:~$ sudo ufw allow 8080
Regla actualizada
Regla actualizada (v6)
```

Con esto realizado, podemos hacer una prueba para ver si efectivamente el servidor Tomcat está funcionando y no es bloqueado por el cortafuegos. Para ello abrimos un navegador y escribimos 127.0.0.1:8080 (también podemos escribir localhost en vez de la dirección IP, es lo mismo). Vemos como efectivamente nos muestra la página de éxito de Tomcat:



Vemos que nos habla de 3 paquetes que sería recomendable instalar. Así que mediante comandos podemos proceder a la instalación de los mismos.

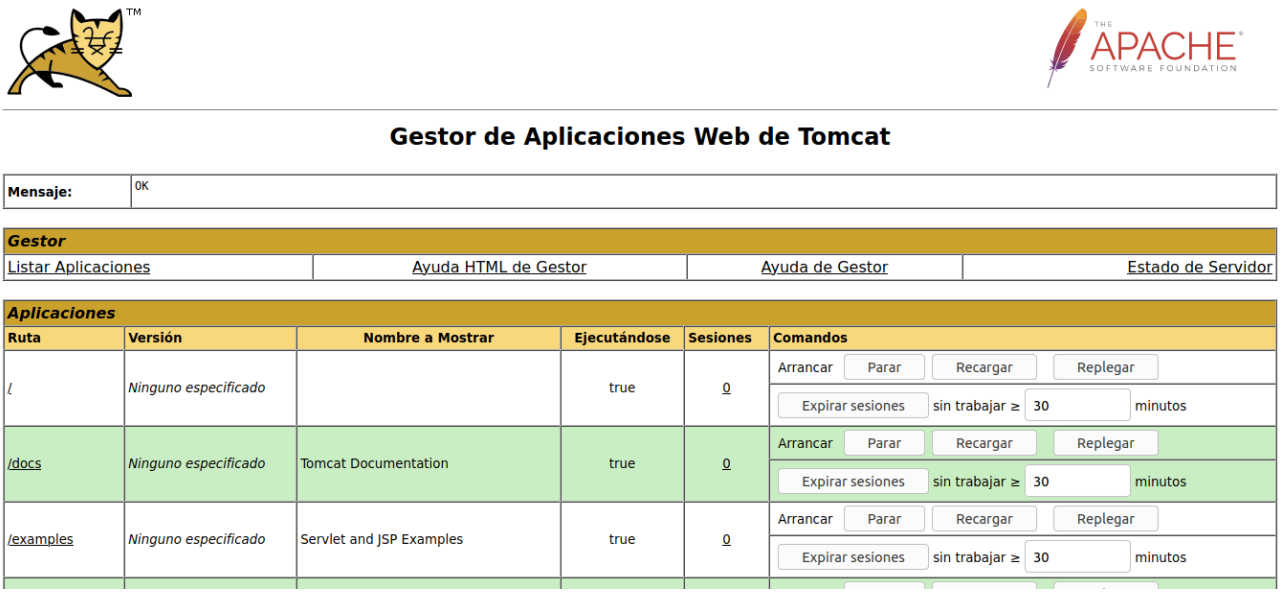
Por último, tenemos que configurar dos usuarios con los permisos de manager-gui y admin-gui. Si intentamos entrar en el fichero tomcat-users.xml para crear esos usuarios, nos dice que no tenemos permisos para ello. Desde la terminal tenemos que modificar dichos permisos para poder hacerlo.

```
javier@javier-VirtualBox:~$ sudo chmod 777 /etc/tomcat9/tomcat-users.xml
```

Ahora ya podremos abrir ese fichero para modificar los usuarios y permisos:

```
<role rolename="admin-gui"/>
<role rolename="manager-gui"/>
<user username="javiadmin" password="javi1" roles="admin-gui"/>
<user username="javimanager" password="javi2" roles="manager-gui"/>
```

Ahora ya podríamos acceder al gestor del servidor Tomcat:



Gestor de Aplicaciones Web de Tomcat

Mensaje:	OK

Gestor			
Listar Aplicaciones	Ayuda HTML de Gestor	Ayuda de Gestor	Estado de Servidor

Aplicaciones					
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

PUNTO 4: INSTALAR OPENSSH

Una vez más, gracias al gestor de paquetes que viene por defecto en Ubuntu, podemos instalar SSH en el sistema mediante un comando:

sudo apt install openssh-client

Vemos como en mi caso, openssh ya estaba preinstalado en mi ordenador:

```
javier@javier-VirtualBox:~$ sudo apt install openssh-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssh-client ya está en su versión más reciente (1:8.2p1-4ubuntu0.1).
fijado openssh-client como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
javier@javier-VirtualBox:~$
```

PUNTO 5: INSTALAR Y CONFIGURAR MARIADB

Por último, vamos a instalar y configurar MariaDB, nuestro sistema gestor de bases relacionales que usaremos con MySQL. Podemos, para asegurarnos, hacer un update primero del repositorio apt de Ubuntu.

Una vez hecho, usamos el siguiente comando para instalar MariaDB:

sudo apt install mariadb-server

Comprobamos ahora el estado del servidor MariaDB:

```
javier@javier-VirtualBox:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.3.25 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese>
     Active: active (running) since Sat 2021-02-13 17:56:37 CET; 3min 11s ago
       Docs: man:mysqld(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 10422 (mysqld)
     Status: "Taking your SQL requests now..."
      Tasks: 31 (limit: 4654)
     Memory: 65.8M
     CGroup: /system.slice/mariadb.service
             └─10422 /usr/sbin/mysqld
```

Vemos que ya nace activo. Instalamos ahora el cliente de mariadb:

```
javier@javier-VirtualBox:~$ sudo apt install mariadb-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
```

Creamos una contraseña, ya que no nos lo ha pedido durante la instalación:

```
javier@javier-VirtualBox:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
```

Y vamos siguiendo los pasos que nos va diciendo. Con el proceso terminado, podemos ejecutar el siguiente comando y, escribiendo nuestra contraseña, vemos que efectivamente mariadb está instalado:

```
javier@javier-VirtualBox:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.3.25-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

Ahora vamos a permitir las conexiones desde fuera. Para ello abrimos el fichero llamado 50-server.cnf que se encuentra en la carpeta mariadb dentro de mysql. Vemos como hay una linea dentro de ese fichero que pone bind-address = 127.0.0.1, es decir, que

actualmente solo se accede desde localhost (desde el propio equipo). Tenemos que cambiarlo por 0.0.0.0 y guardamos el fichero.

Por último, permitimos al firewall que se pueda acceder al puerto 3306 que es el que usa mariadb:

```
javier@javier-VirtualBox:~$ sudo ufw allow 3306/tcp
Regla añadida
Regla añadida (v6)
javier@javier-VirtualBox:~$
```

Y listo, ya tendríamos todos los apartados configurados.