

ACTIVIDAD 2. TAREA INDIVIDUAL. CONFIGURACIÓN DEL SERVIDOR

REQUERIMIENTO 1

Queremos preparar nuestro servidor Linux para poder desplegar una aplicación web. Para ello tenemos que verificar que están instalados:

Java

Apache

Tomcat

openSSH

MariaDB

Valoración: 6 puntos sobre 10

REQUERIMIENTO 2

Así mismo, queremos asegurarnos de que los servidores están bien configurados y son accesibles antes de desplegar la aplicación. Por ello debemos configurar y comprobar que los puertos asociados a Apache, Tomcat y MariaDB están abiertos en el Firewall y son accesibles desde el exterior.

Valoración: 4 puntos sobre 10

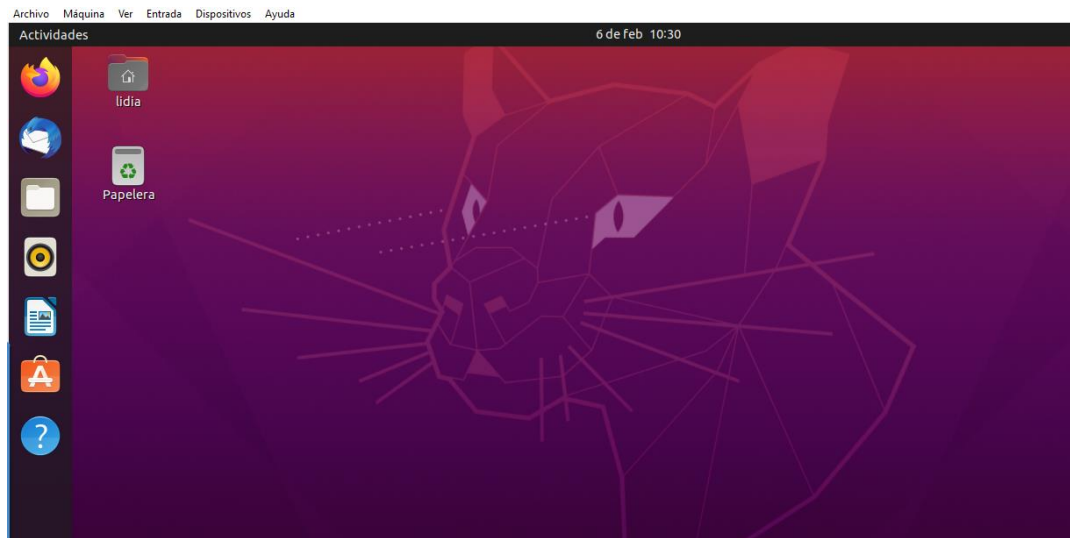
Consideraciones

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

Para la entrega, es necesaria la creación y subida a la plataforma de un pequeño documento formal sobre la actividad (portada, explicación, etc.) y un Manual de instalación que describa y permita realizar todos los pasos para la instalación y configuración de las aplicaciones.

Nótese que más adelante se pedirá que se realicen tareas con un repositorio GIT que contenga la documentación de esta actividad.

En el primer requerimiento nos solicitan la instalación de una serie de aplicaciones y servidores, por lo que vamos a proceder a su instalación. Para ello seguiremos utilizando la máquina virtual de Ubuntu que nos instalamos en actividades anteriores.



La primera aplicación que vamos a instalar es Java, por lo que abrimos el terminal. En primer lugar, nos logamos como root para poder realizar la instalación correctamente.

Seguidamente comenzaremos actualizando las listas de paquetes disponibles:

```
root@lidia-VirtualBox: ~  
lidia@lidia-VirtualBox:~$ sudo su -  
[sudo] contraseña para lidia:  
root@lidia-VirtualBox:~# sudo apt update  
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease  
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Des:3 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]  
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]  
Des:5 http://es.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [415  
kB]  
Des:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [482  
kB]  
Des:7 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [797  
kB]  
Des:8 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [182 k  
B]  
Des:9 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [106  
kB]  
Des:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metada  
ta [24,3 kB]  
Des:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadat  
a [6.164 B]  
Des:12 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Package  
s [121 kB]  
Des:13 http://security.ubuntu.com/ubuntu focal-security/restricted i386 Packages
```

En Ubuntu 20.04 disponemos de las versiones 8, 11, 13 y 14 de Java OpenJDK, correspondiendo la versión por defecto (los paquetes default-jdk y default-jre) a Java OpenJDK 11. En el caso que nos ocupa hemos elegido el JRE.

Por lo que instalamos el paquete con apt.

LIDIA MARTÍNEZ CAPITA

```
root@lidia-VirtualBox:~# sudo apt install -y default-jre
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ca-certificates-java default-jre-headless
  fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni openjdk-11-jre
  openjdk-11-jre-headless
Paquetes sugeridos:
  fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei
Se instalarán los siguientes paquetes NUEVOS:
  ca-certificates-java default-jre default-jre-headless
  fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni openjdk-11-jre
  openjdk-11-jre-headless
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 250 no actualizados.
Se necesita descargar 39,9 MB de archivos.
Se utilizarán 180 MB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 java-common all 0.72
```

Para comprobar la versión instalada basta con utilizar el comando `java -version`.

```
root@lidia-VirtualBox:~# java -version
openjdk version "11.0.9.1" 2020-11-04
OpenJDK Runtime Environment (build 11.0.9.1+1-Ubuntu-0ubuntu1.20.04)
OpenJDK 64-Bit Server VM (build 11.0.9.1+1-Ubuntu-0ubuntu1.20.04, mixed mode, s
haring)
root@lidia-VirtualBox:~#
```

Por otra parte, debemos saber también que en ciertos entornos se necesita la presencia del JRE de Java 8, por lo que será necesario instalar Java OpenJDK 8 en Ubuntu 20.04.

En el caso del entorno de ejecución o JRE de OpenJDK 8 para Ubuntu 20.04 instalamos el paquete `openjdk-8-jre`:

```
root@lidia-VirtualBox:~# sudo apt install -y openjdk-8-jre
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  openjdk-8-jre-headless
Paquetes sugeridos:
  icedtea-8-plugin fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei fonts-wqy-zenhei
Se instalarán los siguientes paquetes NUEVOS:
  openjdk-8-jre openjdk-8-jre-headless
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 250 no actualizados.
Se necesita descargar 28,2 MB de archivos.
Se utilizarán 104 MB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 openjdk-
8-jre-headless amd64 8u275-b01-0ubuntu1~20.04 [28,2 MB]
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 openjdk-
8-jre amd64 8u275-b01-0ubuntu1~20.04 [69,8 kB]
Descargados 28,2 MB en 7s (3.904 kB/s)
```

Y para el kit de desarrollo JDK de OpenJDK 8 seleccionaríamos el paquete `openjdk-8-jdk`:

```

root@lidia-VirtualBox:~# sudo apt install -y openjdk-8-jdk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev
  libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk-headless x11proto-core-dev
  x11proto-dev xorg-sgml-doctools xtrans-dev
Paquetes sugeridos:
  libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-8-demo
  openjdk-8-source visualvm
Se instalarán los siguientes paquetes NUEVOS:
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev
  libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk openjdk-8-jdk-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 actualizados, 14 nuevos se instalarán, 0 para eliminar y 250 no actualizados.
Se necesita descargar 13,1 MB de archivos.
Se utilizarán 50,2 MB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 xorg-sgml-doctools al
ll 1:1.11-1 [12,9 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu focal/main amd64 x11proto-dev all 201
9.2-1ubuntu1 [594 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu focal/main amd64 x11proto-core-dev al

```

Para establecer la versión de Java que se usará por defecto en Ubuntu 20.04 utilizamos el comando `update-alternatives`:

```

root@lidia-VirtualBox:~# sudo update-alternatives --config java
Existen 2 opciones para la alternativa java (que provee /usr/bin/java).

   Selección   Ruta
-----
* 0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111      modo
automático
   1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111      modo
manual
   2            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java    1081      modo
manual

Pulse <Intro> para mantener el valor por omisión [*] o pulse un número de selec
ción:

```

La que está marcada con un asterisco es la versión actualmente seleccionada.

Finalmente, como apunte, muchas aplicaciones basadas en Java necesitan conocer la ruta a la instalación de Java en el sistema a través de la variable de entorno `JAVA_HOME`.

La forma más sencilla de que la variable `JAVA_HOME` esté disponible para cualquier usuario del sistema es configurando su valor en `/etc/environment`:

```

lidia@lidia-VirtualBox:~$ export JAVA_HOME=/usr/lib/jvm/default-java/jre
lidia@lidia-VirtualBox:~$ echo $JAVA_HOME
/usr/lib/jvm/default-java/jre
lidia@lidia-VirtualBox:~$

```

Con estos pasos ya tendríamos instalado y configurada la variable de `JAVA_HOME` en nuestro equipo.

A continuación, vamos a proceder a instalar Apache.

LIDIA MARTÍNEZ CAPITA

Como en la actividad anterior ya instalamos el servidor Apache, vamos a comprobar que lo tenemos instalado.

```
lidia@lidia-VirtualBox: ~  
lidia@lidia-VirtualBox:~$ systemctl status apache2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese  
   Active: active (running) since Sat 2021-02-06 11:29:07 CET; 10min ago  
     Docs: https://httpd.apache.org/docs/2.4/  
   Process: 656 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC  
   Main PID: 733 (apache2)  
     Tasks: 55 (limit: 2316)  
    Memory: 7.3M  
    CGroup: /system.slice/apache2.service  
            └─733 /usr/sbin/apache2 -k start  
              └─734 /usr/sbin/apache2 -k start  
                └─735 /usr/sbin/apache2 -k start  
  
feb 06 11:29:01 lidia-VirtualBox systemd[1]: Starting The Apache HTTP Server...  
feb 06 11:29:07 lidia-VirtualBox apachectl[689]: AH00558: apache2: Could not re  
feb 06 11:29:07 lidia-VirtualBox systemd[1]: Started The Apache HTTP Server.
```

Para averiguar la versión de Apache que tenemos instalada en Ubuntu 20.04 podemos usar el comando `apachectl -v`.

```
lidia@lidia-VirtualBox:~$ apachectl -v  
Server version: Apache/2.4.41 (Ubuntu)  
Server built: 2020-08-12T19:46:17  
lidia@lidia-VirtualBox:~$
```

Como ya permitimos el acceso al servidor por SSH debemos habilitar las conexiones con la aplicación Apache. Esta aplicación nos permite el acceso por HTTP por los puertos comunes de HTTP (80) y HTTPS (443).

Esto lo conseguimos con el comando siguiente:

```
lidia@lidia-VirtualBox:~$ sudo ufw allow "Apache"  
Reglas actualizadas  
Reglas actualizadas (v6)  
lidia@lidia-VirtualBox:~$
```

Finalmente, una vez realizada la configuración, podemos activar ya el firewall, con el comando **ufw enable**, también podemos comprobar la configuración de UFW con el comando **ufw status**. Ahora deberíamos ver que el firewall se encuentra activo y además podremos obtener un listado de aplicaciones habilitadas.

```

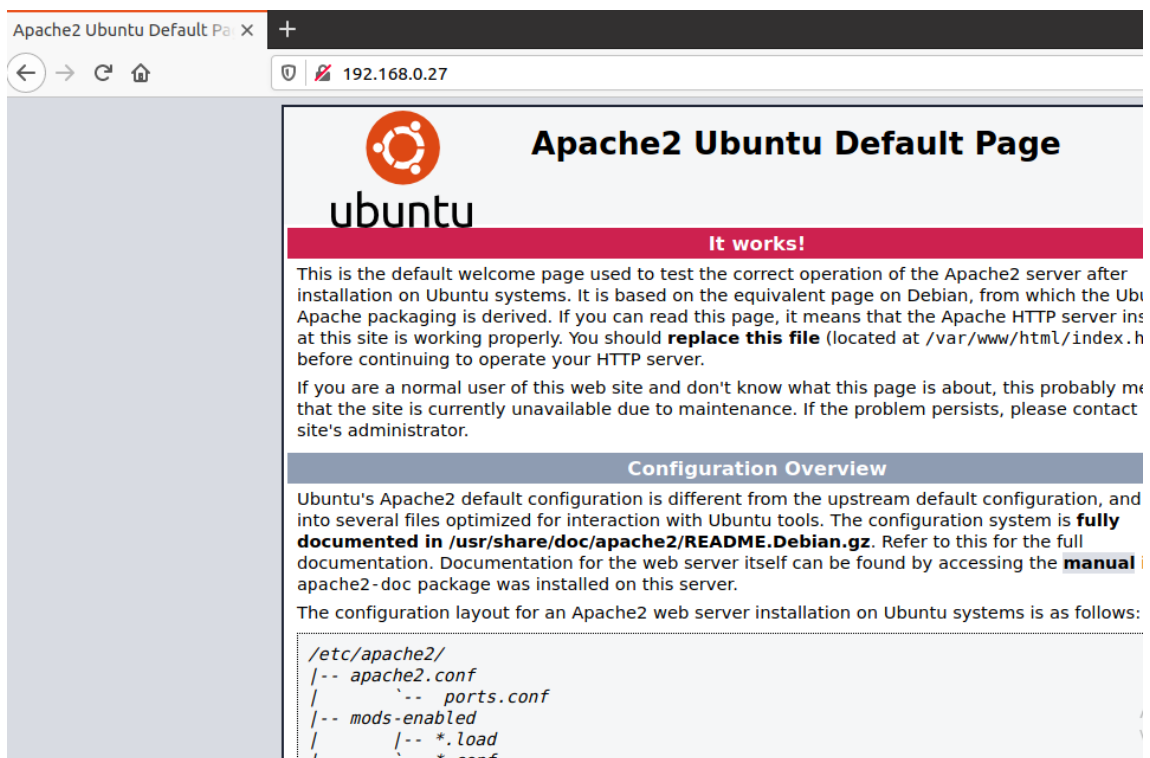
lidia@lidia-VirtualBox:~$ sudo su -
root@lidia-VirtualBox:~# ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
root@lidia-VirtualBox:~# sudo ufw status
Estado: activo

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

root@lidia-VirtualBox:~#

```

Lo último que necesitaríamos hacer sería probar el servidor web, para ello debemos saber nuestra IP. Si todo sale bien, como podemos comprobar en la imagen posterior es que ya tenemos instalado nuestro servidor.



Por otra parte, para saber donde están alojadas las paginas web se usa el comando `ls /var/www`

```

root@lidia-VirtualBox:~# ls /var/www
ls: no se puede acceder a 'var/www': No existe el archivo o el directorio
root@lidia-VirtualBox:~# ls /var/www
html
root@lidia-VirtualBox:~# ls /var/www/html
index.html
root@lidia-VirtualBox:~#

```

Si nos vamos a los archivos de configuración, vemos que Apache está escuchando por el puerto 80.


```
apache2.conf  ×  ports.conf
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
```

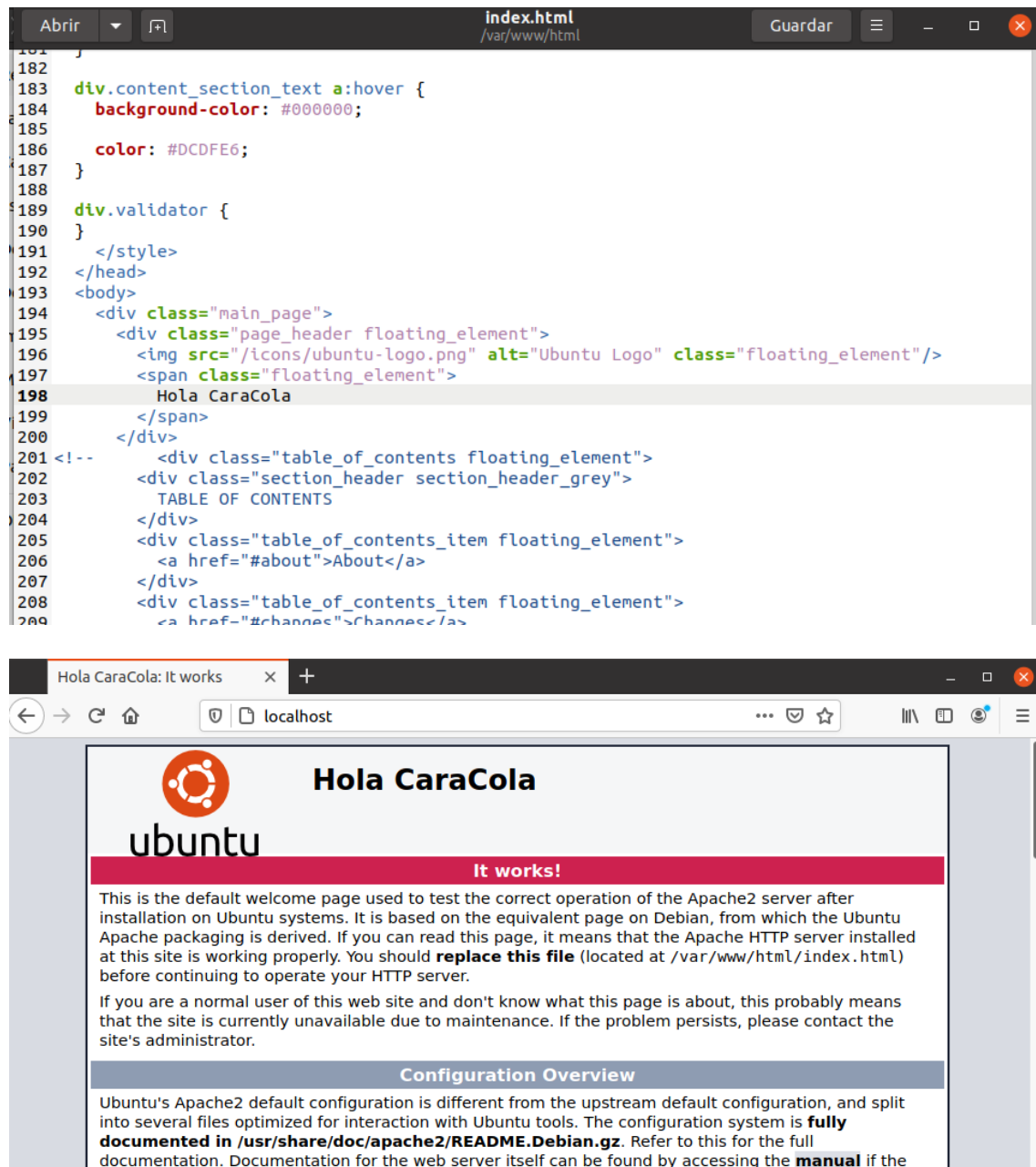
Y el directorio del ServerRoot

```
Abrir  [icon]  apache2.conf [Solo lectura]  Guardar
/et/apache2
61 #
62 # NOTE! If you intend to place this on an NFS (or otherwise network)
63 # mounted filesystem then please read the Mutex documentation (available
64 # at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
65 # you will save yourself a lot of trouble.
66 #
67 # Do NOT add a slash at the end of the directory path.
68 #
69 #ServerRoot "/etc/apache2"
70
71 #
```

También podemos ver que están denegados los permisos para el acceso a mis documentos (mi directorio raíz), pero sí está permitido el acceso para las páginas webs.

```
59 <Directory />
60     Options FollowSymLinks
61     AllowOverride None
62     Require all denied
63 </Directory>
64
65 <Directory /usr/share>
66     AllowOverride None
67     Require all granted
68 </Directory>
69
70 <Directory /var/www/>
71     Options Indexes FollowSymLinks
72     AllowOverride None
73     Require all granted
74 </Directory>
75
```

Como tenemos permisos para modificar el archivo HTML, lo modificamos y vemos como efectivamente la página de inicio cambia.



El siguiente servidor a instalar es Tomcat. Lo primero que podemos hacer es consultar el repositorio para ver qué paquete de Tomcat está disponible para descargar.

```
root@lidia-VirtualBox: ~  
lidia@lidia-VirtualBox:~$ sudo su -  
root@lidia-VirtualBox:~# apt-cache search tomcat  
resource-agents - Cluster Resource Agents  
libapache-mod-jk-doc - Documentación del paquete libapache2-mod-jk  
liblogback-java - biblioteca de acceso flexible para Java  
liblogback-java-doc - biblioteca de acceso flexible para Java - documentación  
solr-jetty - Enterprise search server based on Lucene3 - Jetty integration  
solr-tomcat - Enterprise search server based on Lucene3 - Tomcat integration  
libapache2-mod-jk - Apache 2 connector for the Tomcat Java servlet engine  
libjnlpservlet-java - simple and convenient packaging format for JNLP applicati  
ons  
libnetty-tcnative-java - Tomcat native fork for Netty  
libnetty-tcnative-jni - Tomcat native fork for Netty (JNI library)  
libsolr-java - Enterprise search server based on Lucene - Java libraries  
libspring-instrument-java - modular Java/J2EE application framework - Instrument  
ation  
libtcnative-1 - Tomcat native library using the Apache Portable Runtime  
libtomcat9-embed-java - Apache Tomcat 9 - Servlet and JSP engine -- embed librar  
ies  
libtomcat9-java - Apache Tomcat 9 - Servlet and JSP engine -- core libraries  
libtomcatjss-java - JSSE implementation using JSS for Tomcat  
nagios-plugins-contrib - Plugins for nagios compatible monitoring systems  
python3-ajpy - Python module to craft AJP requests  
solr-common - Enterprise search server based on Lucene3 - common files
```

Podemos ver que se nos va a instalar el Apache Tomcat 9. Por lo tanto, vamos a proceder a su instalación. El comando para ello será:

```
root@lidia-VirtualBox:~# sudo apt install tomcat9 tomcat9-admin  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no  
son necesarios.  
  linux-headers-5.4.0-42 linux-headers-5.4.0-42-generic  
  linux-image-5.4.0-42-generic linux-modules-5.4.0-42-generic  
  linux-modules-extra-5.4.0-42-generic  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:  
  libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9-common  
Paquetes sugeridos:  
  tomcat9-docs tomcat9-examples tomcat9-user  
Se instalarán los siguientes paquetes NUEVOS:  
  libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9 tomcat9-admin  
  tomcat9-common  
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 229 no actualizados.  
Se necesita descargar 12,3 MB de archivos.  
Se utilizarán 14,8 MB de espacio de disco adicional después de esta operación.
```

Una vez que Tomcat haya finalizado la instalación, debería iniciarse automáticamente. Se puede verificar que se ejecute con el sscomando. Debería ver un puerto abierto, número 8080, ya que ese es el puerto predeterminado para Apache Tomcat.

```

root@lidia-VirtualBox: ~
Creating config file /etc/tomcat9/logging.properties with new version
Creating config file /etc/tomcat9/context.xml with new version
Creating config file /etc/tomcat9/catalina.properties with new version
Creating config file /etc/tomcat9/jaspic-providers.xml with new version
Creating config file /etc/logrotate.d/tomcat9 with new version
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) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9) ...
root@lidia-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          [::]:*
root@lidia-VirtualBox:~#

```

Por otra parte, también necesitamos configurar el firewall para permitir el tráfico desde cualquier fuente al puerto 8080.

```

root@lidia-VirtualBox:~# sudo ufw status
Estado: activo

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

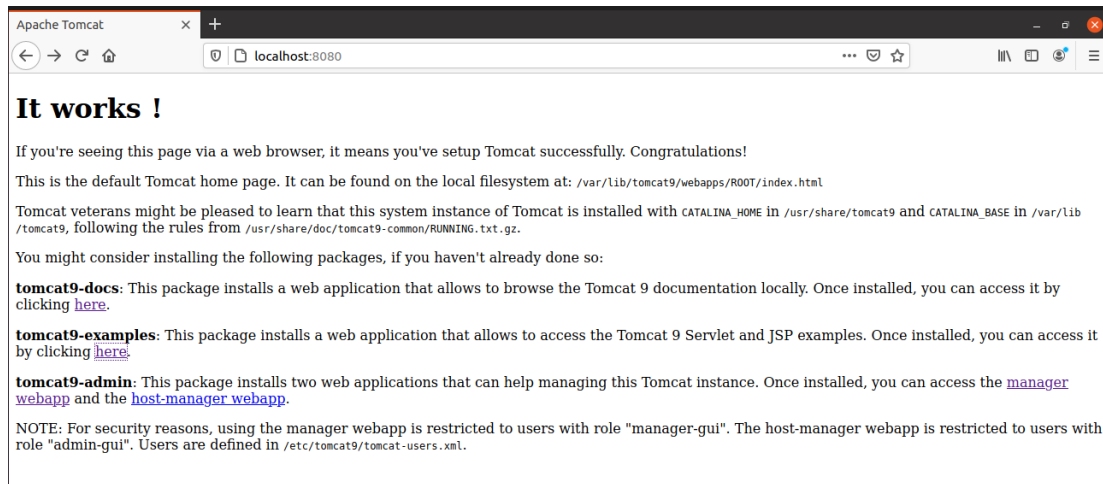
root@lidia-VirtualBox:~# sudo ufw allow 8080/tcp
Regla añadida
Regla añadida (v6)
root@lidia-VirtualBox:~# sudo ufw status
Estado: activo

Hasta      Acción      Desde
-----
Apache     ALLOW       Anywhere
8080/tcp   ALLOW       Anywhere
Apache (v6) ALLOW       Anywhere (v6)
8080/tcp (v6) ALLOW       Anywhere (v6)

root@lidia-VirtualBox:~#

```

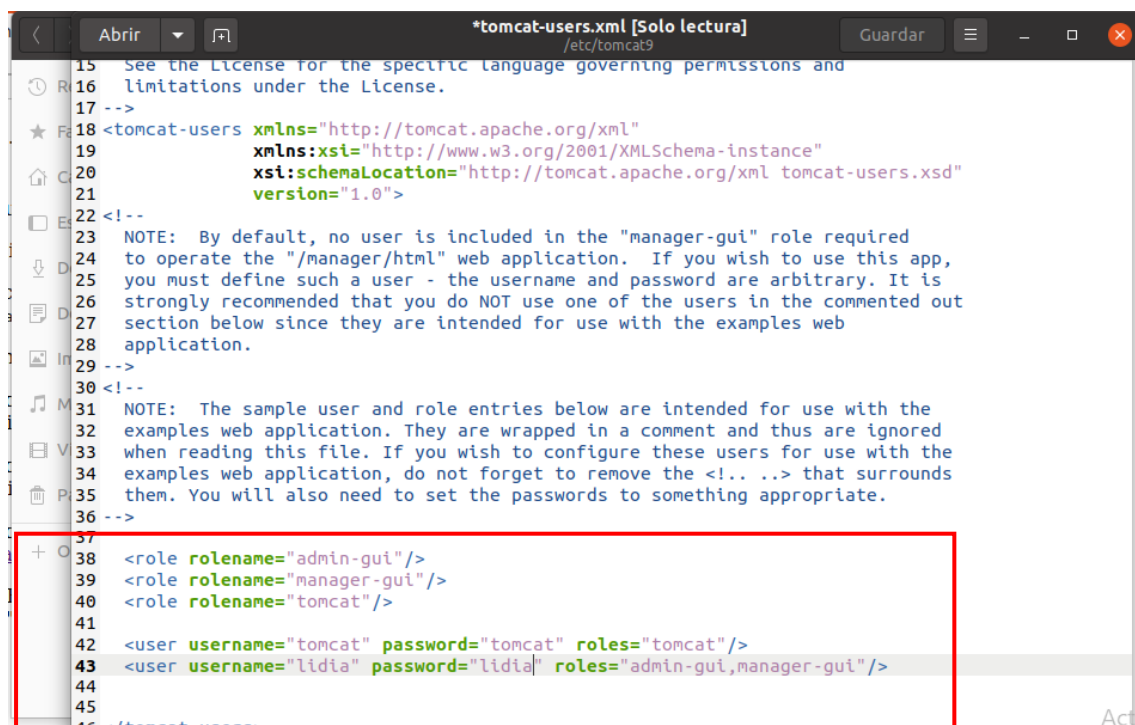
Con Tomcat en funcionamiento, ahora deberíamos poder acceder a él en un navegador web: localhost:8080.



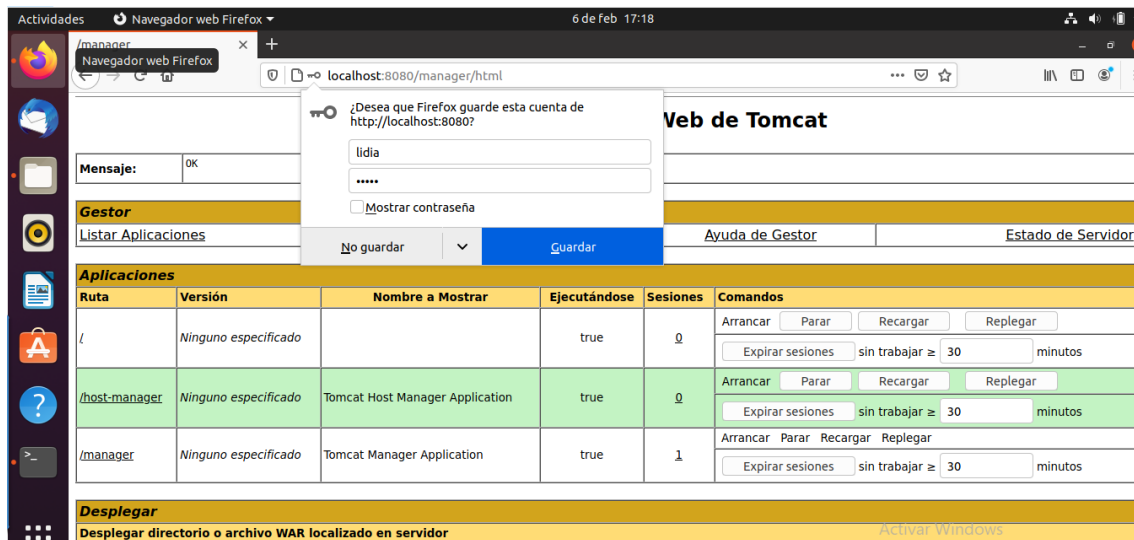
A continuación, vamos a asignar un usuario para el Administrador de aplicaciones web en el servidor Tomcat. Es por ello que debemos crear una nueva cuenta de usuario para utilizar Apache Tomcat Web Application Manager.

Por tanto, los pasos que seguiremos serán los siguientes. Vamos a abrir el archivo tomcat-users.xml en el directorio Tomcat con el editor de texto.

Cambiamos contraseñas y usuarios junto con los roles de admin y manager como se puede observar en la imagen.



Seguidamente y después de reiniciar el servicio, deberemos iniciar sesión en Tomcat Web Application Manager y nos solicitará las credenciales que acabamos de asignar.



Con este paso ya tendríamos instalado correctamente nuestro servidor Tomcat.

La siguiente tarea que se nos solicita es instalar OpenSSH. Instalamos este paquete con apt.

```

lidia@lidia-VirtualBox: ~
lidia@lidia-VirtualBox:~$ sudo apt install -y ssh
[sudo] contraseña para lidia:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  linux-headers-5.4.0-42 linux-headers-5.4.0-42-generic
  linux-image-5.4.0-42-generic linux-modules-5.4.0-42-generic
  linux-modules-extra-5.4.0-42-generic
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
Paquetes sugeridos:
  molly-guard monkeysphere ssh-askpass
Se instalarán los siguientes paquetes NUEVOS:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 229 no actualizados.
Se necesita descargar 693 kB de archivos.
Se utilizarán 6.130 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-sftp-server amd64 1:8.2p1-4ubuntu0.1 [51,5 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-server amd64 1:8.2p1-4ubuntu0.1 [377 kB]

```

Podemos comprobar el estado del servicio SSH.

```

lidia@lidia-VirtualBox:~$ systemctl status ssh
● ssh.service - OpenSSH Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-02-06 19:04:16 CET; 3min 13s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 2164 (sshd)
     Tasks: 1 (limit: 2316)
    Memory: 1.1M
     CGroup: /system.slice/ssh.service
             └─2164 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

```

Además, podemos comprobar el servicio SSH en modo local, usando el comando de consola ssh localhost:

La primera vez que conectamos con un cliente SSH a un servicio se nos pregunta si deseamos guardar la huella digital de la clave pública que usa el servidor, para que en próximas conexiones se compare la huella guardada con la recibida en la nueva conexión, puesto que si son distintas indicaría que los certificados ya no son los mismos, siendo una posible causa la suplantación de identidad del sitio.

```
lidia@lidia-VirtualBox: ~  
lidia@lidia-VirtualBox:~$ ssh localhost  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ECDSA key fingerprint is SHA256:9jbvVAh0q0IdK95k69RHHGSqoJM+epsiQ1ovj0qU8A0.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.  
lidia@localhost's password:  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.8.0-41-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Introducing self-healing high availability clusters in MicroK8s.  
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
  
    https://microk8s.io/high-availability  
  
230 actualizaciones se pueden instalar inmediatamente.  
0 de estas actualizaciones son una actualización de seguridad.  
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable  
  
Your Hardware Enablement Stack (HWE) is supported until April 2025.
```

Finalmente, necesitamos configurar el firewall para realizar conexiones desde red al servicio SSH.

```
lidia@lidia-VirtualBox:~$ sudo ufw allow ssh  
[sudo] contraseña para lidia:  
Regla añadida  
Regla añadida (v6)  
lidia@lidia-VirtualBox:~$ ufw status  
ERROR: Debe ser root (administrador) para ejecutar este guión  
lidia@lidia-VirtualBox:~$ sudo su -  
root@lidia-VirtualBox:~# ufw status  
Estado: activo  
  
Hasta      Acción      Desde  
-----  
Apache     ALLOW       Anywhere  
8080/tcp   ALLOW       Anywhere  
22/tcp     ALLOW       Anywhere  
Apache (v6) ALLOW       Anywhere (v6)  
8080/tcp (v6) ALLOW       Anywhere (v6)  
22/tcp (v6) ALLOW       Anywhere (v6)  
root@lidia-VirtualBox:~#
```

El último paso de esta actividad es instalar MariaDB, un sistema de administración relacional de bases de datos de código abierto.

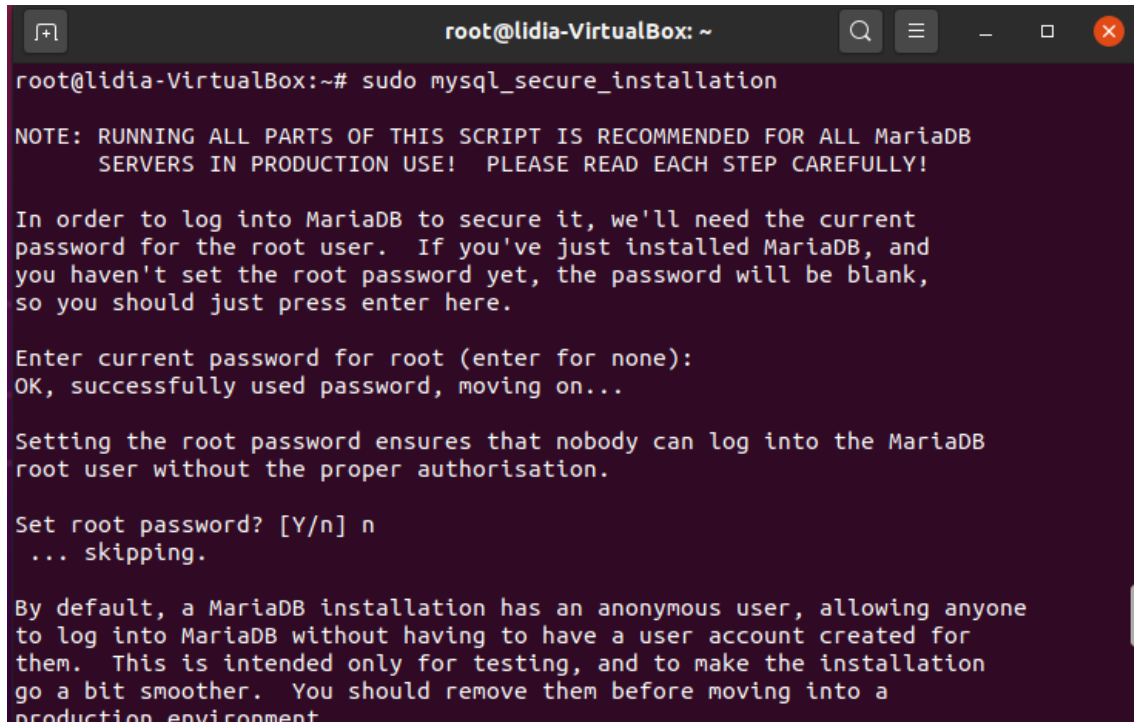
Inicialmente lo que necesitamos hacer es actualizar el índice de paquetes usando apt.

```
root@lidia-VirtualBox: ~  
root@lidia-VirtualBox:~# sudo apt update  
Des:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]  
Obj:2 http://es.archive.ubuntu.com/ubuntu focal InRelease  
Des:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]  
Des:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24,3 kB]  
Des:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [56,6 kB]  
Des:7 http://es.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [415 kB]  
Des:8 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [797 kB]  
Des:9 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [264 kB]  
Des:10 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [737 kB]  
Des:11 http://es.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [547 kB]  
Des:12 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [301 kB]  
Des:13 http://es.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2.468 B]  
Des:14 http://es.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11
```

Seguidamente, instalamos el paquete de mariadb-server usando apt.

```
root@lidia-VirtualBox: ~  
root@lidia-VirtualBox:~# sudo apt install mariadb-server  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
  linux-headers-5.4.0-42 linux-headers-5.4.0-42-generic  
  linux-image-5.4.0-42-generic linux-modules-5.4.0-42-generic  
  linux-modules-extra-5.4.0-42-generic  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:  
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl  
  libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libfcgi-perl  
  libhtml-template-perl libreadline5 libsigsegv2 libsnappy1v5  
  libterm-readkey-perl mariadb-client-10.3 mariadb-client-core-10.3  
  mariadb-common mariadb-server-10.3 mariadb-server-core-10.3 socat  
Paquetes sugeridos:  
  gawk-doc libclone-perl libmldbm-perl libnet-daemon-perl  
  libsql-statement-perl libipc-sharedcache-perl mailx mariadb-test tinycat  
Se instalarán los siguientes paquetes NUEVOS:  
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl  
  libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libfcgi-perl  
  libhtml-template-perl libreadline5 libsigsegv2 libsnappy1v5  
  libterm-readkey-perl mariadb-client-10.3 mariadb-client-core-10.3
```


En las nuevas instalaciones de MariaDB, el siguiente paso es ejecutar la secuencia de comandos de seguridad incluida. Esta secuencia de comandos cambia algunas de las opciones predeterminadas que son menos seguras. La usaremos para bloquear las conexiones de root remotas y eliminar los usuarios de la base de datos no utilizados.



```
root@lidia-VirtualBox: ~  
root@lidia-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):  
OK, successfully used password, moving on...  
  
Setting the root password ensures that nobody can log into the MariaDB  
root user without the proper authorisation.  
  
Set root password? [Y/n] n  
... skipping.  
  
By default, a MariaDB installation has an anonymous user, allowing anyone  
to log into MariaDB without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.
```

Ahora, veremos una serie de solicitudes mediante las cuales podremos realizar cambios en las opciones de seguridad de su instalación de MariaDB. En la primera solicitud se pide que metamos la contraseña root de la base de datos actual. Debido a que no configuramos una aún, pulsamos no y le damos a enter.

Para todas las siguientes cuestiones, también pulsamos enter y de este modo aceptamos los valores predeterminados para todas las preguntas siguientes. Con esto, se eliminarán algunos usuarios anónimos y la base de datos de prueba.

```
root@lidia-VirtualBox: ~  
By default, MariaDB comes with a database named 'test' that anyone can  
access. This is also intended only for testing, and should be removed  
before moving into a production environment.  
  
Remove test database and access to it? [Y/n]  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!  
  
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.  
  
Reload privilege tables now? [Y/n]  
... Success!  
  
Cleaning up...  
  
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.  
  
Thanks for using MariaDB!  
root@lidia-VirtualBox:~#
```

El siguiente paso será crear una cuenta nueva llamada administrador con las mismas capacidades que la cuenta root, ya que normalmente se recomienda crear una cuenta administrativa independiente para el acceso basado en contraseña. Para hacer este paso introducimos en el terminal el siguiente comando:

```
lidia@lidia-VirtualBox: ~  
lidia@lidia-VirtualBox:~$ sudo mysql  
[sudo] contraseña para lidia:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 36  
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.  
MariaDB [(none)]>
```

Ahora, crearemos un nuevo usuario con privilegios root y acceso basado en contraseña y vaciamos los privilegios.

```
MariaDB [(none)]> GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'pas  
sword' WITH GRANT OPTION;  
Query OK, 0 rows affected (0.570 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES  
->
```

Seguidamente cerramos el terminal y pasamos a probar la instalación.

```
lidia@lidia-VirtualBox:~$ sudo systemctl status mariadb
[sudo] contraseña para lidia:
● mariadb.service - MariaDB 10.3.25 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Sun 2021-02-07 13:58:57 CET; 46min left
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 774 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 31 (limit: 2316)
    Memory: 82.4M
    CGroup: /system.slice/mariadb.service
            └─774 /usr/sbin/mysqld

Feb 07 13:58:40 lidia-VirtualBox systemd[1]: Starting MariaDB 10.3.25 database
Feb 07 13:58:49 lidia-VirtualBox mysqld[774]: 2021-02-07 13:58:49 0 [Note] /usr
Feb 07 13:58:49 lidia-VirtualBox mysqld[774]: 2021-02-07 13:58:49 0 [Warning] C
Feb 07 13:58:57 lidia-VirtualBox systemd[1]: Started MariaDB 10.3.25 database s
Feb 07 13:58:59 lidia-VirtualBox /etc/mysql/debian-start[1030]: Looking for 'my
Feb 07 13:58:59 lidia-VirtualBox /etc/mysql/debian-start[1030]: Looking for 'my
Feb 07 13:58:59 lidia-VirtualBox /etc/mysql/debian-start[1030]: This installati
Feb 07 13:58:59 lidia-VirtualBox /etc/mysql/debian-start[1048]: Checking for in
Feb 07 13:58:59 lidia-VirtualBox /etc/mysql/debian-start[1052]: Triggering myis
```

Como comprobación adicional, puede intentar establecer conexión con la base de datos usando la herramienta mysqladmin, que es un cliente que le permite ejecutar comandos administrativos.

Además, como hemos configurado un usuario administrativo independiente con la autenticación de contraseña, podemos realizar la misma operación tal y como se muestra en la imagen posterior.

```
lidia@lidia-VirtualBox: ~
lidia@lidia-VirtualBox:~$ sudo mysqladmin version
[sudo] contraseña para lidia:
mysqladmin Ver 9.1 Distrib 10.3.25-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.3.25-MariaDB-0ubuntu0.20.04.1
Protocol version        10
Connection               Localhost via UNIX socket
UNIX socket              /var/run/mysqld/mysqld.sock
Uptime:

lidia@lidia-VirtualBox:~$ mysqladmin -u admin -p version
Enter password:
mysqladmin Ver 9.1 Distrib 10.3.25-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.3.25-MariaDB-0ubuntu0.20.04.1
Protocol version        10
Connection               Localhost via UNIX socket
UNIX socket              /var/run/mysqld/mysqld.sock
Uptime:

lidia@lidia-VirtualBox:~$
```

Esto significa que MariaDB está activo y que nuestro usuario puede autenticarse correctamente.

```
lidia@lidia-VirtualBox:~$ sudo su -  
[sudo] contraseña para lidia:  
root@lidia-VirtualBox:~# ufw status  
Estado: activo  
  
Hasta          Acción      Desde  
-----  
Apache         ALLOW      Anywhere  
8080/tcp       ALLOW      Anywhere  
22/tcp         ALLOW      Anywhere  
3306           ALLOW      Anywhere  
Apache (v6)    ALLOW      Anywhere (v6)  
8080/tcp (v6)  ALLOW      Anywhere (v6)  
22/tcp (v6)    ALLOW      Anywhere (v6)  
3306 (v6)      ALLOW      Anywhere (v6)  
  
root@lidia-VirtualBox:~#
```

Con esta última instalación damos por terminada la segunda actividad.