**GUIA DE DESPLIEGUE DE LA APLICACIÓN:**



INDICE DE CONTENIDOS:

1. INTRODUCCIÓN.
2. DIAGRAMA DE RED DE LA APLICACIÓN.
3. REQUERIMIENTOS DEL SISTEMA.
4. ELECCIÓN DE INFRAESTRUCTURA.
5. SERVIDOR DE BASE DE DATOS Y FTP.
6. BIBLIOGRAFÍA Y ENLACES DE EQUIPO.
7. CONTACTO Y SOPORTE

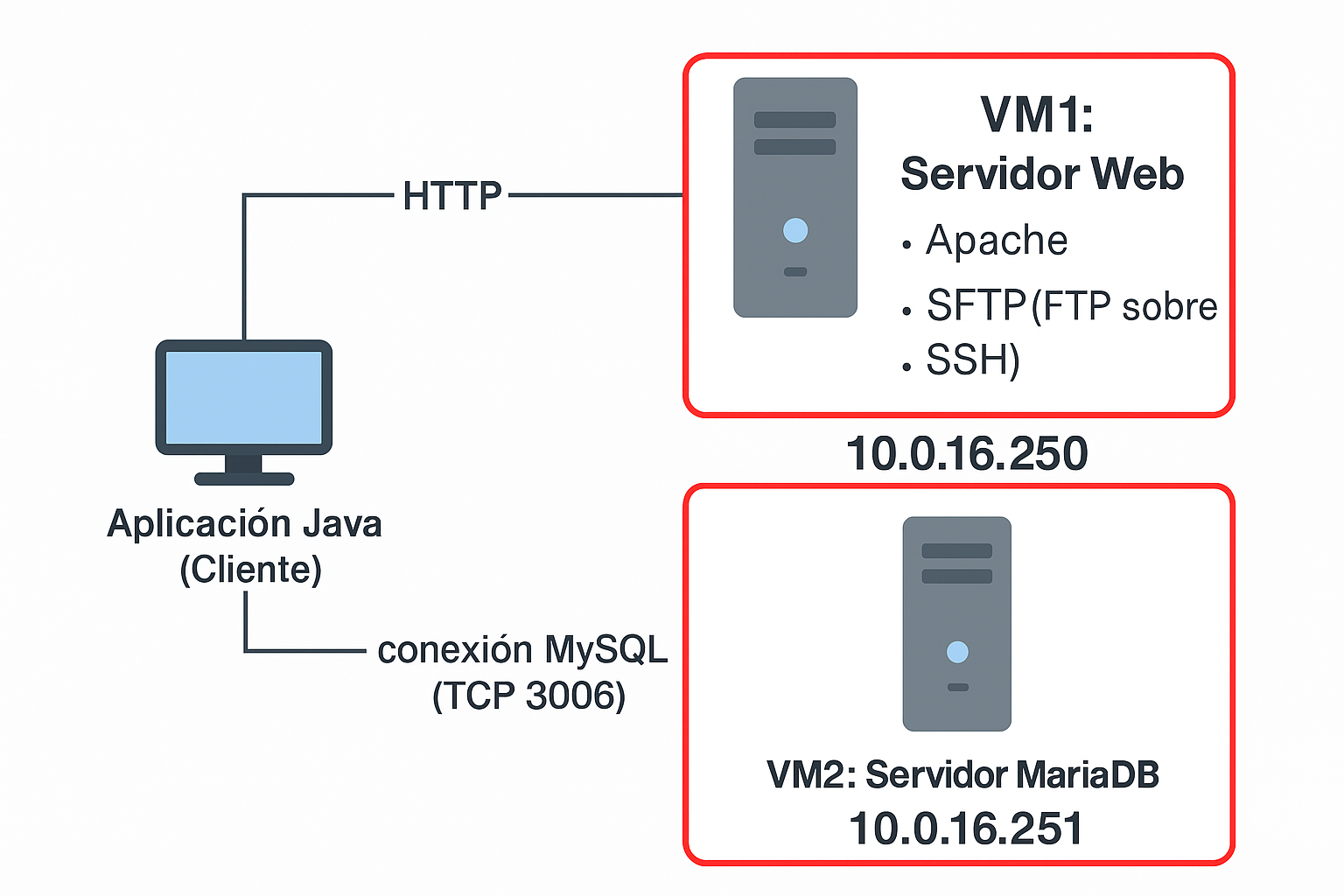
**1. INTRODUCCIÓN:**

En este documento se explicará de manera detallada sobre cómo se ha montado y alojado el servidor web de nuestra aplicación para su correcto funcionamiento. Esto incluye la instalación de un servidor de base de datos, un servidor web, y la configuración de servicios para la transferencia segura de archivos.

El sistema está pensado para una aplicación que utiliza MySQL como base de datos y un servidor Apache para alojar el sitio web. También incluye mecanismos de seguridad para la transferencia de archivos, usando FTP y SSH.

**2. DIAGRAMA DE RED DE LA APLICACIÓN:**

Hemos desarrollado el siguiente diagrama de red para tener claro su uso y poder también visualizar su infraestructura de la aplicación:



En el vemos como contaremos con 2 equipos de servidor diferentes, ambas montadas en máquinas virtuales.

El primero utilizará la IP 10.0.16.250, y contará con el servidor web APACHE que se conecta por HTTP por el puerto TCP 80 y un servidor FTP sobre SSH para transferencia segura de archivos por el puerto TCP 22.

La segunda maquina con IP 10.0.16.251 incorporará un servidor de bases de datos MARIADB que se conecta por el puerto TCP 3306 e igualmente utilizará FTP sobre SSH para transferencia segura del script de la base de datos en caso de necesitar actualizarle igualmente mediante el puerto TCP 22.

**3. SERVIDOR SQL/SSH:**

-Elección de infraestructura:

Para el servidor de base de datos, al moverle a Alpine vimos que no cuenta con un servidor MySQL puro, por lo que hemos instalado MariaDB que trabaja con MySQL de nativa sin problemas de compatibilidad, ya que nuestra base de datos estaba creada originalmente en MySQL.

-Requerimientos del sistema:

**Hardware**:

* CPU: 2 núcleos
* RAM: 4 GB
* Almacenamiento: 8 GB
* Red: La conexión de red debe ser estable.

**Software**:

* Sistema Operativo: Alpine Linux
* Versiones necesarias de servicios:

-MariaDB: última versión estable.

-OpenSSH: última versión estable.

-Instalación y configuración del servidor:

-En primer lugar, instalamos tanto el servidor de la base de datos como el servidor ssh con la instrucción:

$apk add mariadb mariadb-cliente $apk add sshd

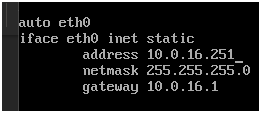
-La primera vez que montamos un servidor mariadb debemos crear su sistema de directorios mediante la instrucción (ejecutar solo 1 vez):

$mysql\_install\_db --user=mysql --basedir=/usr --datedir=/var/lib/mysql

-De igual forma que hicimos antes, en esta máquina virtual estableceremos una IP estática y un sistema de seguridad mediante FTP sobre SSH con ChrootDirectory.

-Primero cambiamos la IP estática de igual forma que antes modificando el archivo en la ruta:

$nano /etc/network/interfaces



-Después haremos las mismas operaciones que llevamos antes a cabo, pero esta vez el directorio de trabajo será directamente el /home del usuario.

-Primero hacemos que la carpeta padre y superiores sean propiedad del root:

$chown root:root /home

y damos los permisos requeridos a la ruta

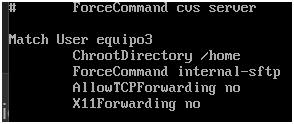
$chmod –R 755 /home

-Después haremos que el usuario (en este caso equipo3) sea propietario del directorio home, pero en este caso al ser su directorio /home por defecto este paso no es necesario, y tampoco será necesario mover su directorio /home de lugar.

-Tras esto, modificaremos igual que antes la configuración del SSH para que utilice la función de ChrootDirectory y enjaular al usuario, modificando el archivo en:

$nano /etc/ssh/sshd\_config

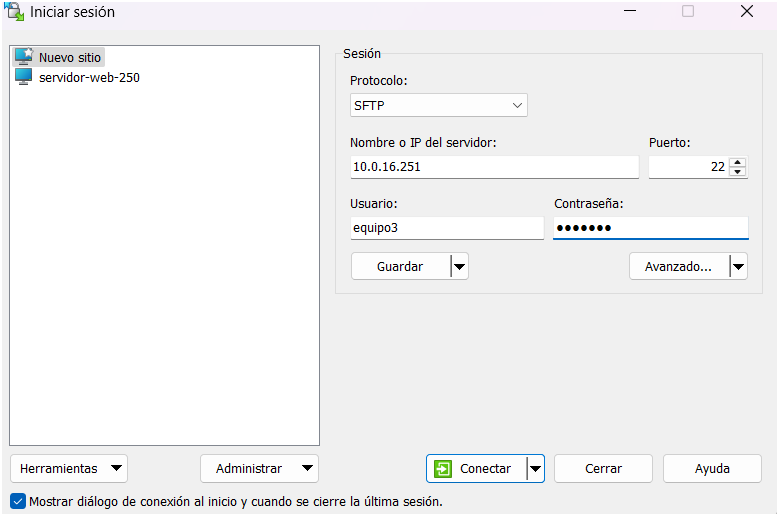
y realizamos la misma configuración anterior



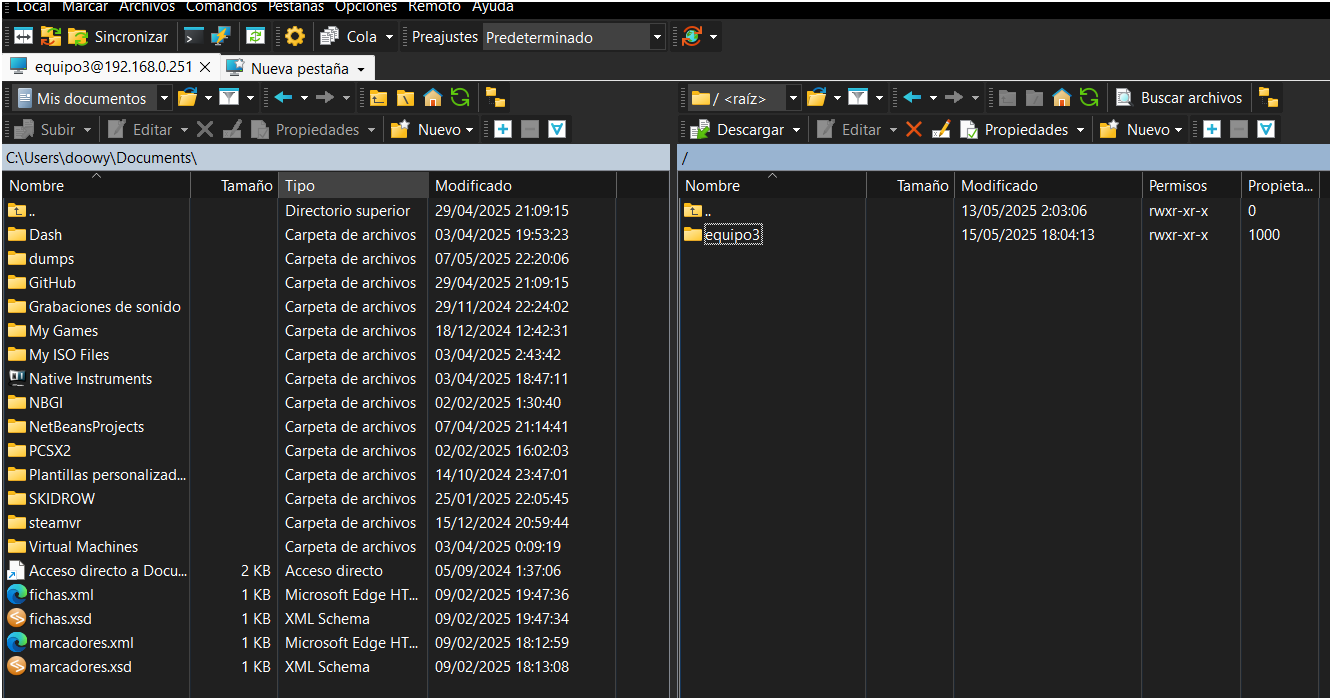
-De esta forma nos queda el servidor de base de datos preparado con un sistema de conexión a él por SFTP (FTP sobre SSH).

-Lo único que faltaría sería utilizar este servicio SFTP para pasar a la máquina virtual nuestro script SQL e iniciarle. Para ello nos conectaremos con WinSCP.

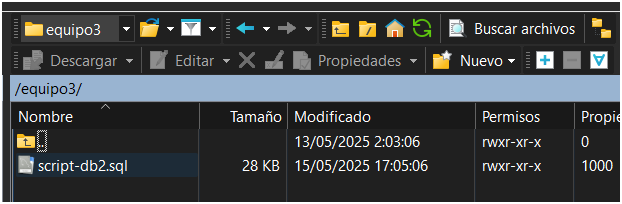
-Abrimos el programa e introducimos los datos del servidor:



-Y ahora veremos la pantalla principal del programa, donde aparece a la izquierda los archivos de nuestro equipo y a la derecha los archivos presentes en la máquina virtual.



-Ahora solo arrastramos el script a la parte de la derecha:



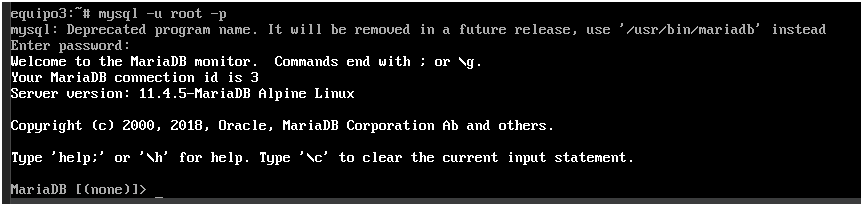
-Y por último inicializamos la base de datos con ese script con la instrucción:

$mysql –u root –p < /home/equipo3/script-db.sql

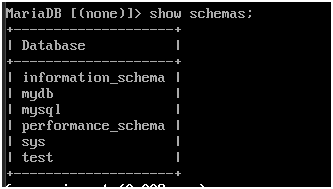
-Si queremos comprobar que todo ha funcionado bien, podemos acceder a la base de datos mediante un cliente SQL con la instrucción:

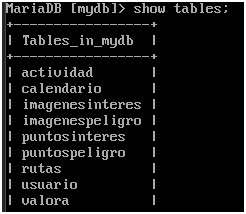
$mysql –u root –p

lo cual nos abrirá un terminal SQL



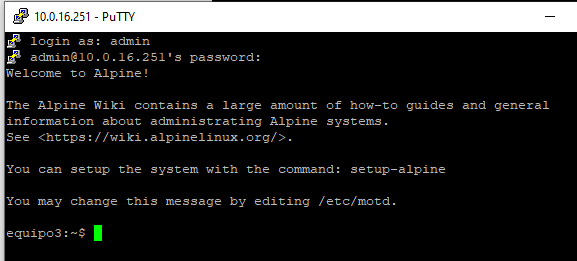
donde podremos buscar nuestra base de datos:





Acceso de clientes por SSH

Para el acceso por SSH a través de puerto 22 a la máquina virtual, hemos creado un usuario llamado admin con la misma contraseña que no tiene restricciones en cuanto al Chrootdirectory ni restricciones adicionales de SSH. Nos conectaremos a la máquina mediante el software Putty con la IP 10.0.16.251



**4. SERVIDOR WEB/SSH:**

-Requerimientos del sistema:

**Hardware**:

* CPU: 2 núcleos
* RAM: 1 GB
* Almacenamiento: 8 GB
* Red: La conexión de red debe ser estable.

**Software**:

* Sistema Operativo: Alpine Linux
* Versiones necesarias de servicios:

-Apache: última versión estable.

-OpenSSH: última versión estable.

-Elección de infraestructura:

Para el servidor web hemos elegido Apache debido a su sencillez de uso y mantenimiento. También ha sido la elección debido a que nos resulta más familiar su uso por haber trabajado con el previamente.

-Instalación y configuración del servidor:

-Respecto al servidor web, en primer lugar, hemos instalado el servidor Apache en nuestra máquina virtual Alpine mediante el uso del comando

$apk add apache2

-Sabemos por la documentación que la ruta donde leerá Apache los archivos de la web es /var/www/localhost/htdocs en Alpine.

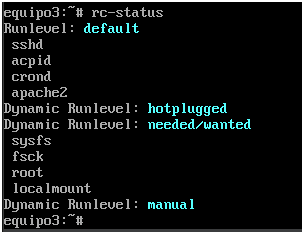
-De igual forma hemos instalado el servidor SSH con la instrucción

$apk add sshd

-Ambos los hemos configurado para que se inicien al arrancar el sistema, mediante el uso de la instrucción

$rc-update add sshd $r-update add apache2

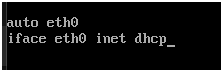
-Como podemos ver en la imagen, ambos servicios se encuentran en el nivel default, es decir, el arranque principal del sistema.



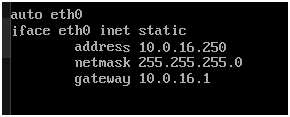
-Tras eso, modificaremos la configuración de red de la máquina virtual a nivel interno para asigne a la máquina una IP estática en lugar de asignarla por dhcp. Para ello modificaremos el archivo en la ruta:

$nano /etc/network/interfaces

y modificaremos la parte de dhcp



por lo siguiente



donde asignaremos en address la IP deseada, en netmask la máscara de red y en gateway la puerta de enlace predeterminada.

-Para aumentar la seguridad, haremos que el usuario que se conecta al servidor por FTPS (en este caso equipo3) este enjaulado en su directorio home sin poder salir, para ello utilizaremos la funcionalidad de ChrootDirectory.

-Para utilizar ChrootDirectory se requiere que la carpeta padre y superiores pertenezcan a root y la carpeta que se va a utilizar pertenezca al usuario, por lo que cambiaremos el propietario de las carpetas superiores con la instrucción:

$chown –R root:root /var (-R indica aplicar de forma recursiva)

y le daremos permisos con la instrucción

$chmod –R 755 /var (-R indica que se aplique de forma recursiva)

-El siguiente paso será crear la carpeta dentro de ese directorio ../htdocs que va a contener los archivos de la web con:

$ mkdir /var/www/localhost/htdocs/web

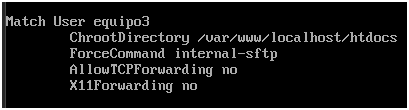
Ahora nos falta hacer que el usuario sea propietario de ese directorio con:

$chown equipo3:equipo3 /var/www/localhost/htdocs/web

-Por último, modificaremos la configuración de SSH para que utilice la función del ChrootDirectory, para lo cual modificaremos el siguiente archivo:

$nano /etc/ssh/sshd\_config

y añadiremos al final del documento lo siguiente



De este modo, al conectarse por SFTP, el usuario únicamente podrá moverse por el directorio destinado a los archivos de la página web.

-Por último, modificaremos la configuración de Apache para que lea los archivos de la web desde este nuevo directorio creado en ../htdocs/web. Para ello entraremos a la configuración de Apache con la instrucción:

$nano /etc/apache2/httpd.conf

y modificaremos la siguiente parte para incluir la nueva carpeta



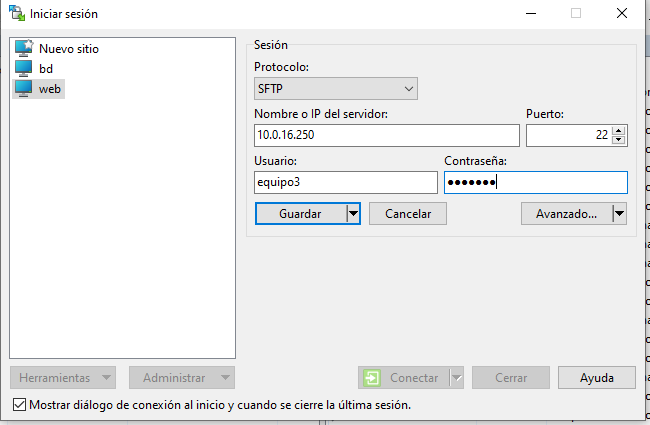
-De esta forma hemos creado un enjaulamiento para mejorar aún más la seguridad de nuestro servidor SFTP.

-Para conectarnos utilizaremos la herramienta WINScp, con las credenciales:

-IP: 10.0.16.250

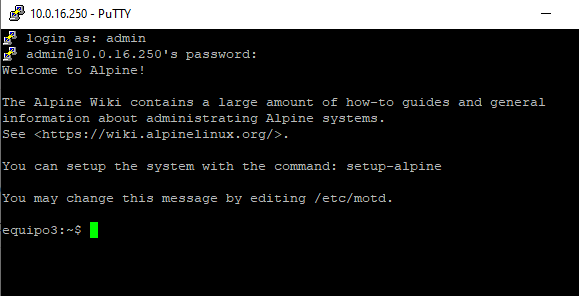
-Usuario: equipo3

-Contraseña: equipo3



Acceso de clientes por SSH

Para el acceso por SSH a través de puerto 22 a la máquina virtual, hemos creado un usuario llamado admin con la misma contraseña que no tiene restricciones en cuanto al Chrootdirectory ni restricciones adicionales de SSH. Nos conectaremos a la máquina mediante el software Putty.



**6. BIBLIOGRAFÍA Y ENLACES DE EQUIPO:**

**Fuentes técnicas y documentación oficial**

* MySQL Documentation. <https://dev.mysql.com/doc/>
* Apache HTTP Server Project. <https://httpd.apache.org/docs/>
* Docker Documentation. https://docs.docker.com/
* VirtualBox User. <https://www.virtualbox.org/manual/UserManual.html>

**Enlaces de Equipo**:

<https://github.com/MiguelIGP23/DAM1_EQUIPO3_2425>

**7. Licencia, Términos y Condiciones de Uso:**

La licencia recomendada sería la MIT. Esto significa que puedes usarla, modificarla y distribuirla bajo los términos establecidos en la licencia.

Términos y Condiciones de Uso

1. Uso Permitido:
   * El usuario puede utilizar la aplicación solo para fines personales.
   * El usuario no podrá modificar ni redistribuir la aplicación sin la debida autorización.
2. Responsabilidad:
   * El equipo de desarrollo no se hace responsable de los daños causados por el uso inapropiado de la aplicación.
3. Actualizaciones:
   * El equipo proporcionara diferentes actualizaciones para mejorar la funcionalidad de la aplicación para tener un mejor uso.

**8. Contacto y Soporte**

Si necesitas ayuda o tienes alguna consulta, contáctanos a través de los siguientes medios:

Correo Electrónico: **equipo3DAM1@gmail.com**

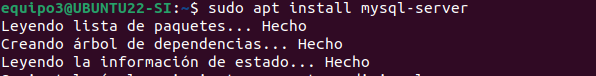
Teléfono de Soporte: 608664371

**9. Anexo**

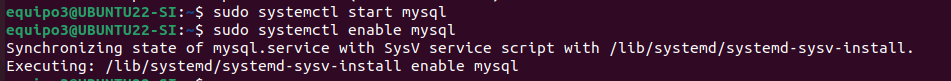
Anexo 1- Servidor de base de datos en Ubuntu Desktop

Hemos optado por el servicio de MYSQL ya que se tiene más familiarizado y por lo tanto a la hora de alojar la base de datos nos permite tener una mayor comodidad y conocimiento, además de que es compatible con una mayor cantidad de opciones.

Para la preparación del servidor de base de datos hemos tenido que instalar el paquete de mysql desde la terminal de la máquina virtual de Ubuntu:



Luego se inició y se comprobó el servicio de sql a través de los siguientes comandos para verificar la instalación.



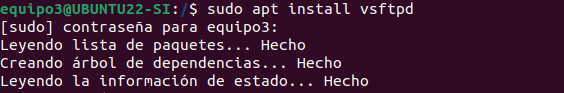
Una vez realizado esto, decidimos enviar el script de nuestra base de datos a través del servidor ftp que necesitaremos posteriormente a la hora de montar el servidor web.

**-SERVIDOR FTP:**

El servidor FTP que hemos elegido nos ha parecido el más manejable ya que nos ha permitido transferir los archivos de nuestra base de datos de manera eficaz y rápida además del servidor. Para garantizar una transferencia segura de archivos, hemos optado por un servidor FTP que soporta **FTPS** (FTP sobre SSL/TLS). Ya que cifra la conexión entre el cliente y el servidor, protegiendo los datos durante la transmisión, lo que hace una conexión optima entre ambas partes.

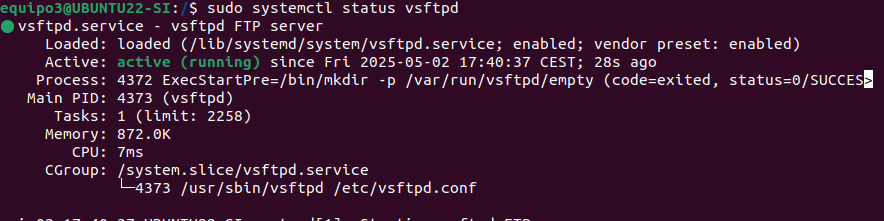
El protocolo FTP es compatible con la gran mayoría de sistemas operativos y clientes, nuestra empresa opto por WINSCP por su gran manejo y facilidad en el uso, lo que asegura que los usuarios puedan acceder al servidor desde diferentes plataformas sin problemas.

Para ello instalamos vsftpd que es el paquete que contiene el servidor ftp



Luego comprobaremos el estado del ftp para ver si está instalado y funcionando

En este caso, vemos que esta activa.

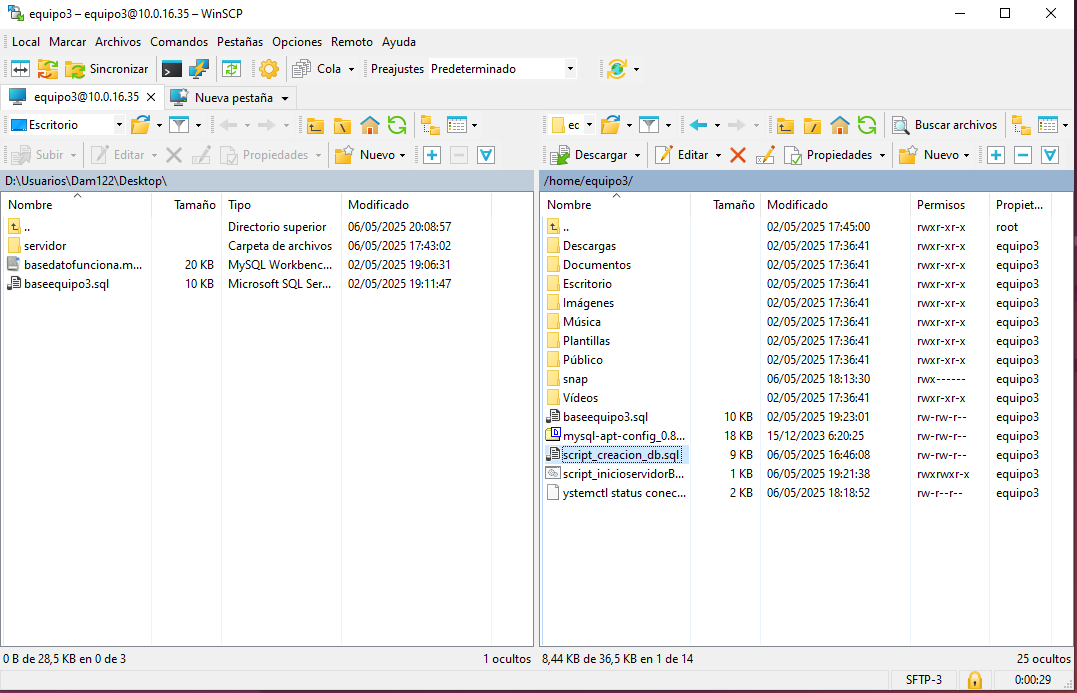


Para poder configurar nuestra sesión en WINSCP que será el cliente que utilizaremos, deberemos poner la IP de nuestro servidor como el usuario y contraseña del usuario creado

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Ahora nos mostrara la siguiente pantalla y en la dirección /home/equipo3 de nuestro servidor enviaremos mediante ftp el script .sql que hemos creado y trabajado con el proyecto.



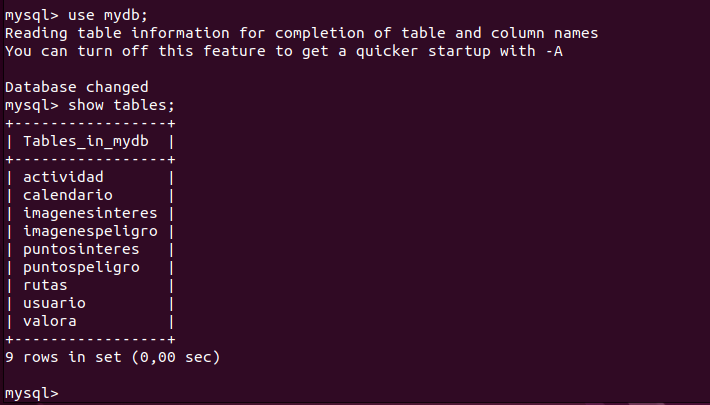
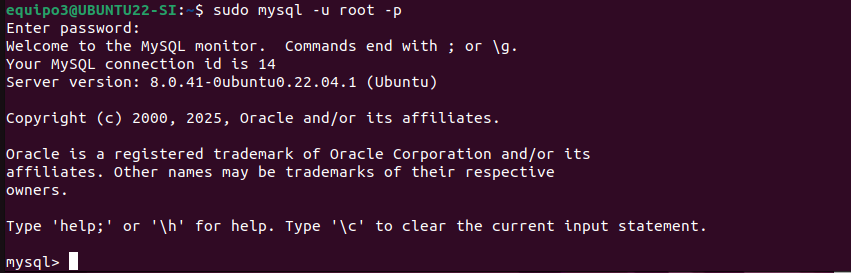
Ahora nos dirigimos a la terminal de ubuntu en la que utilizaremos lo siguiente para poder redirigir el script al sql.



Con esto hemos usado redirecciones para transferir a la base de datos scriptdb el contenido sql de nuestra base de datos.

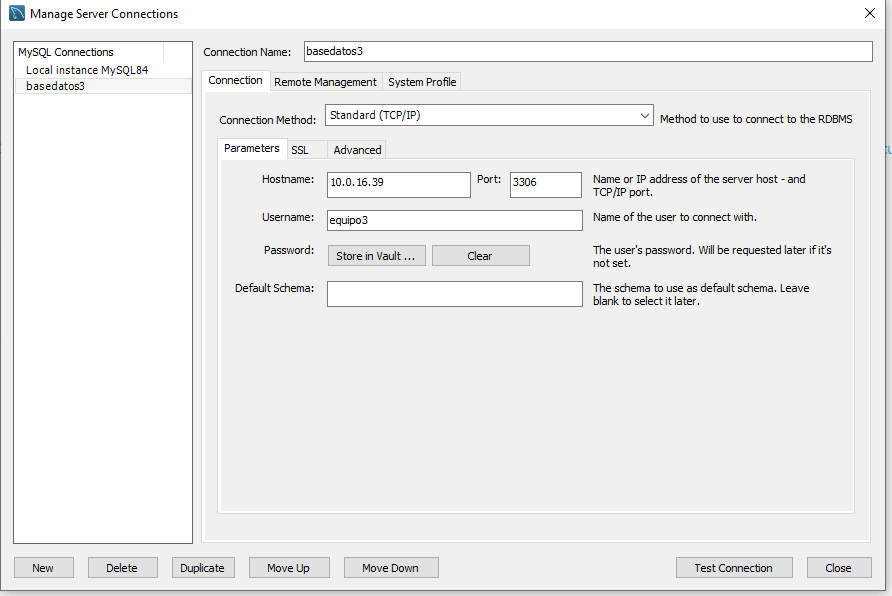
De esta forma tenemos el script metido en nuestro servidor de la base de datos.

Se puede comprobar entrando en el sql

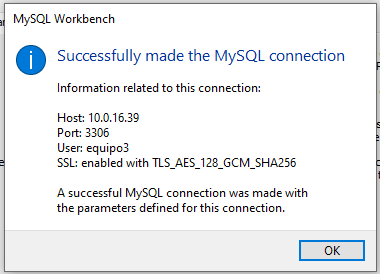
Vemos que se han subido de forma correcta la base de datos con todo su contenido.

Por otra parte, hemos creado una conexión local desde mysql worbench en nuestro ordenador anfitrión de la siguiente manera:

En el que el username y contraseña es el mismo que en el equipo servidor



Hacemos un Test Connection y veremos que funciona de manera correcta.



Ahora hemos querido automatizar con un script estas funciones:

-Iniciar automaticamente el usuario (equipo3).

-Iniciar de manera automatica la conexión de sql mediante el shell.

Para este caso instalamos el paquete de mysqlsh que nos permite tener la terminal de sql.

- A raíz de ello, podremos tener la conexión activa de manera automatizada para un funcionamiento estable.

El script que hemos utilizado es el siguiente:

**#!/bin/bash**

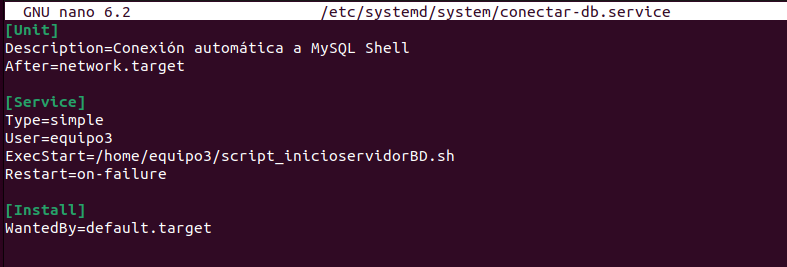
**mysqlsh --sql --uri equipo3@10.0.16.39:3306<<EOF**

**select now();**

**EOF**

**echo "Conexión establecida”**

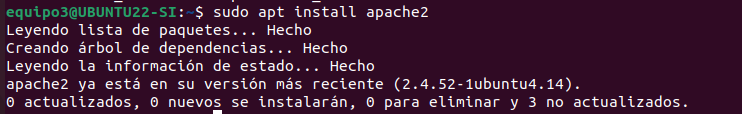
Para que se inicie de forma automática se deberá modificar el siguiente archivo:

Para que se inicie el script en el arranque de la máquina virtual

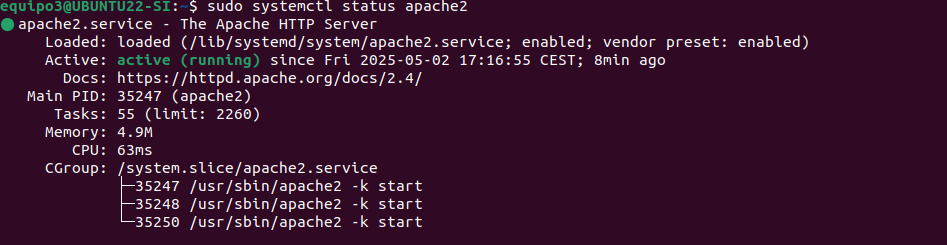
Anexo 2- Servidor web en Ubuntu Desktop

Para el caso del servidor web, previamente esta explicado por qué se ha elegido el servicio de apache y sus ventajas respecto a la competencia.

Lo primero fue instalar el paquete de apache2 en el servidor.



Luego se comprueba el estado del apache y su instalación



Ahora hemos decidido que se inicie automáticamente al arrancar el servidor

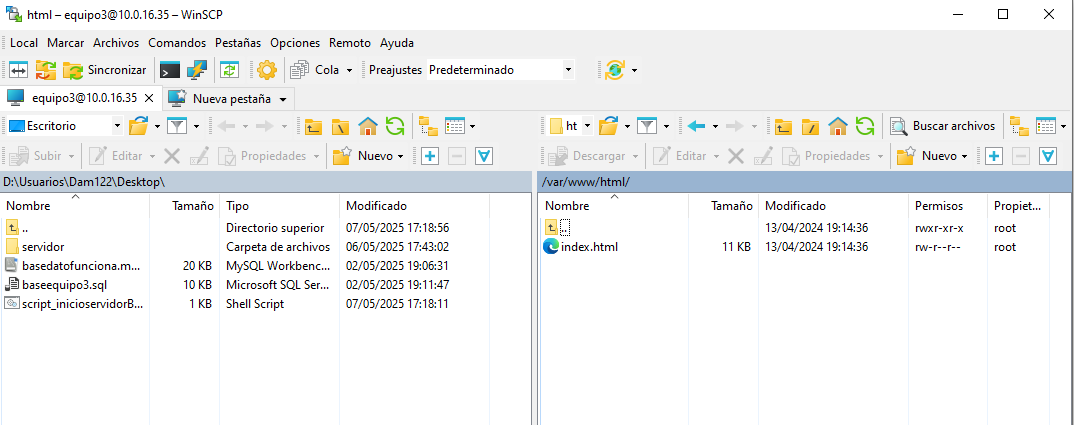


Después, para poder comprobar que funciona de manera correcta buscamos en internet http: ip

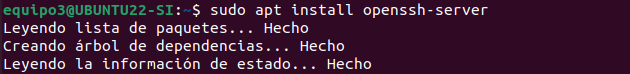
Interfaz de usuario gráfica

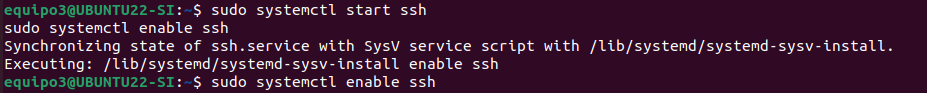
El contenido generado por IA puede ser incorrecto.

Una vez hemos comprobado eso, nos decidimos por mandar a través del ftp el index.html de nuestra página web del proyecto (más adelante) de manera sencilla, con la ruta /var/www/html que es donde está el archivo

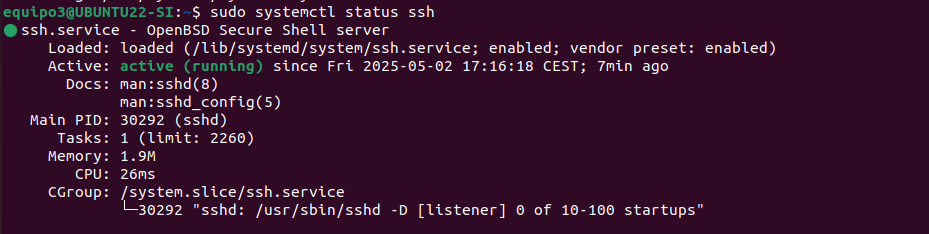


Para el caso del SSH simplemente hemos instalado el paquete y comprobado su funcionamiento:





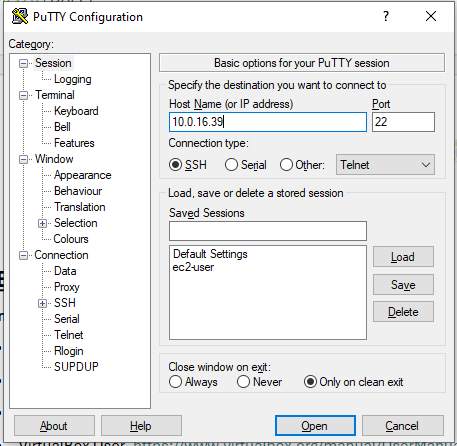
Se activara el ssh y su estado:

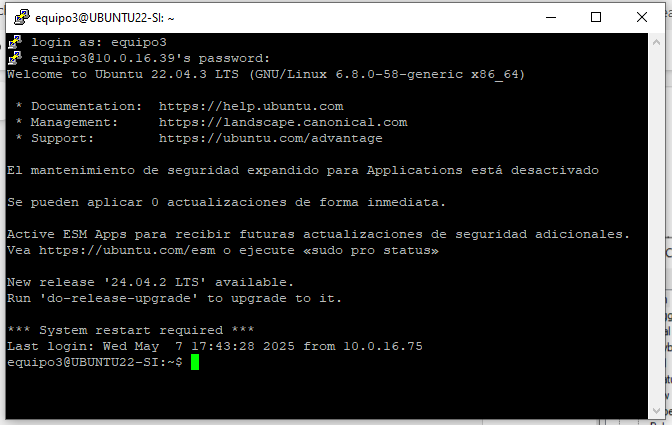


La guía para conectarte desde acceso remoto es la siguiente:

-Conéctate a un cliente e inicialó (putty,aws..)

-Introduce en la IP la correspondiente en este caso 10.0.16.39



Introduce las siguientes credenciales:

Username: equipo3

Password: equipo3