

MANUAL DE DESPLIEGUE DAM1 - EQUIPO 1

Gestión de rutas en actividades al aire libre



RDosM2

Maya García Velasco

Miguel Ángel Quian Díaz

Daniel Cabeza Berrazueta

Raúl Buenaga García

ÍNDICE

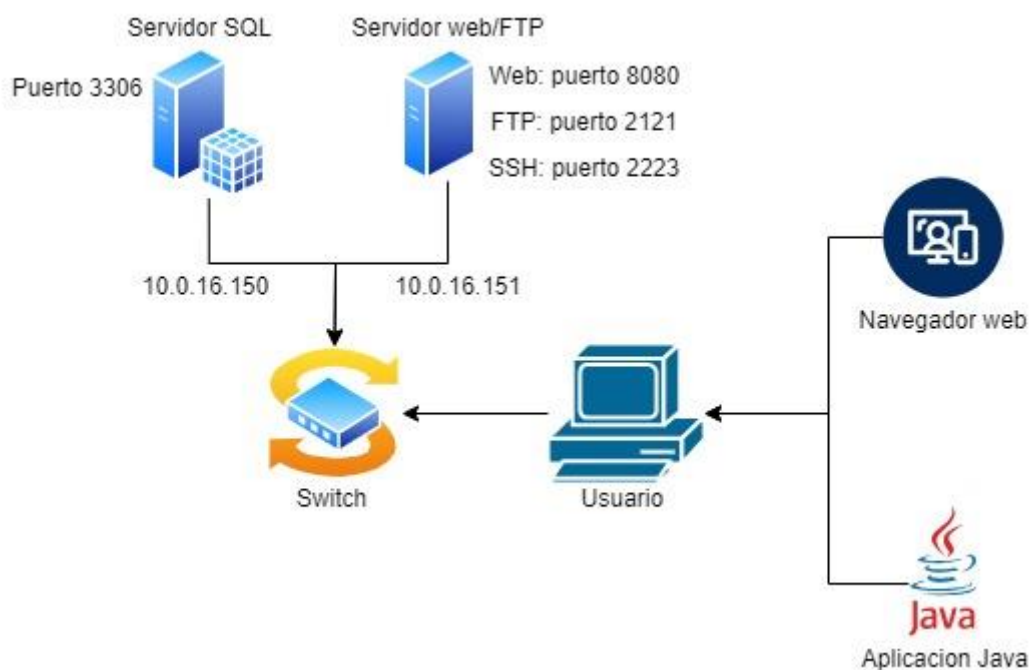
- 1. OBJETIVO*
- 2. DIAGRAMA DE LA ARQUITECTURA DE LA APLICACIÓN*
- 3. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE BASE DE DATOS*
 - 3.1 DESCRIPCIÓN Y JUSTIFICACIÓN*
 - 3.2 REQUISITOS DEL SISTEMA*
 - 3.3 INSTALACIÓN Y CONFIGURACIÓN*
 - 3.4 CARGA DE DATOS*
 - 3.5 ACCESO DESDE LA APLICACIÓN*
- 4. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR WEB*
 - 4.1 DESCRIPCIÓN Y JUSTIFICACIÓN*
 - 4.2 INSTALACIÓN Y CONFIGURACIÓN*
 - 4.3 ACCESO COMO CLIENTE*
 - 4.4 SERVIDOR FTP*
 - 4.4.1 DESCRIPCIÓN Y JUSTIFICACIÓN*
 - 4.4.2 INSTALACIÓN Y CONFIGURACIÓN*
 - 4.4.3 ACCESO COMO CLIENTE*
 - 4.5 SERVIDOR SSH*
 - 4.5.1 DESCRIPCIÓN Y JUSTIFICACIÓN*
 - 4.5.2 INSTALACIÓN Y CONFIGURACIÓN*
 - 4.5.3 ACCESO COMO CLIENTE*
- 5. BIBLIOGRAFÍA*

1. Objetivo

El objetivo de este manual es la documentación sobre los procedimientos usados para la creación y configuración del servidor de base de datos y de la página web, incluyendo porqué hemos escogido unas infraestructuras, cómo las hemos implementado y cómo tienen acceso a ello desde clientes.

2. Diagrama de arquitectura de la aplicación

Esta es la infraestructura que hemos planteado para la aplicación



3. Instalación y configuración del servidor de base de datos

3.1 Descripción y justificación

Elegimos Ubuntu server como sistema operativo y MySQL Server como sistema de gestión de bases de datos por su estabilidad, seguridad y eficiencia. Ubuntu Server es una distribución Linux de código abierto ampliamente utilizada en entornos empresariales y de servidores debido a su bajo consumo de recursos, soporte a largo plazo y una comunidad activa que garantiza actualizaciones constantes y soluciones rápidas a problemas comunes. Por otra parte, MySQL Server es un motor de bases de datos relacional robusto, confiable y altamente compatible con una amplia gama de aplicaciones, que ofrece un excelente

rendimiento incluso bajo cargas de trabajo intensas. Ambos componentes, al ser gratuitos y bien documentados, permiten una implementación rentable y flexible, ideal tanto para pequeños proyectos como este.

3.2 Requisitos del sistema

Tener instalada la última versión de VirtualBox, y MySQL Workbench.

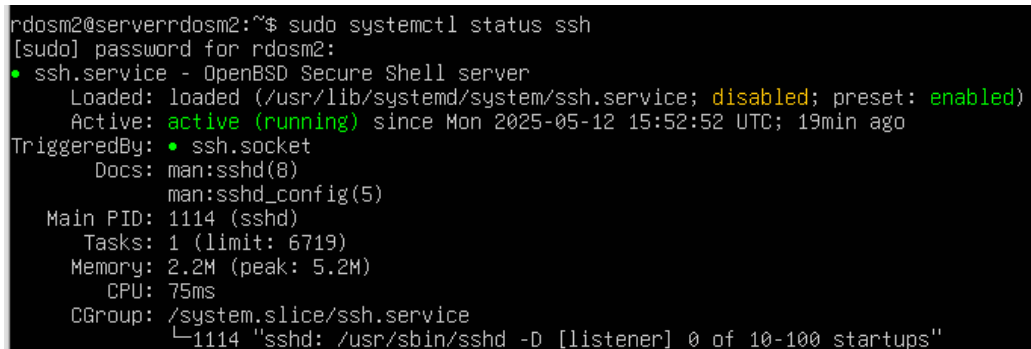
3.3 Instalación y configuración

Se crea una máquina virtual desde VirtualBox para alojar el servidor de base de datos, en este caso el sistema operativo elegido ha sido Ubuntu server.

Para poder hacer cambios de forma más sencilla en la máquina virtual, optamos por instalar un servicio ssh y acceder desde fuera utilizando PuTTY. El proceso es el siguiente:

- `sudo apt update`
- `sudo apt install openssh-server -y`
- `reboot`

Como se puede comprobar en la imagen, con el comando `sudo systemctl status ssh` el servicio ssh está funcional y estará operativo automáticamente cada vez que se encienda la máquina.



```
rdosm2@serverrdosm2:~$ sudo systemctl status ssh
[sudo] password for rdosm2:
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Mon 2025-05-12 15:52:52 UTC; 19min ago
   TriggeredBy: • ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 1114 (sshd)
      Tasks: 1 (limit: 6719)
     Memory: 2.2M (peak: 5.2M)
        CPU: 75ms
     CGroup: /system.slice/ssh.service
            └─1114 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Para facilitar el acceso al servidor hemos también configurado una ip estática. Para lograrlo se debe editar el archivo `.yaml` de la carpeta `netplan` como se muestra en la imagen:



```
GNU nano 7.2
network:
  version: 2
  renderer: networkd
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 10.0.16.150/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
      routes:
        - to: default
          via: 10.0.16.1
```

Los comandos utilizados para ello son los siguientes:

- `sudo nano /etc/netplan/00-installer-config.yaml`
- `sudo netplan try`
- Reboot

Para que el cambio sea efectivo, es necesario hacer un reboot de la máquina virtual.

La base de datos cuenta con un usuario root, y un usuario administrador con todos los permisos para alterar la base de datos. Para configurar la base de datos hemos seguido el proceso de instalación segura que ofrece mysql, usando el comando:

- `sudo mysql_secure_installation.`

Debido a un problema de la instalación segura en versiones recientes de mysql, se debe configurar el usuario root y su contraseña antes de usarlo con estas instrucciones:

- `sudo mysql`
- `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';`

Después de terminar la instalación segura, lo siguiente ha sido crear el usuario administrador, y concederle todos los permisos para poder gestionar la base de datos con los siguientes comandos:

- `CREATE USER 'admin'@localhost IDENTIFIED BY 'password';`
- `GRANT PRIVILEGE on *.* to 'admin'@localhost IDENTIFIED BY 'password';`

3.4 Carga de datos

En esta ocasión hemos decidido cargar la base de datos desde fuera de la máquina virtual con MySQL Workbench. Desde la máquina local, creamos una nueva conexión con las credenciales del usuario administrador de base de datos (log in: admin, contraseña: password), puerto 3306 y la ip 10.0.16.150.

Una vez establecida la conexión, importamos el script de la base de datos para su creación y almacenamiento dentro el servidor.

3.5 Acceso desde la aplicación

La aplicación de escritorio podrá acceder a la base de datos gracias a los drivers que implementan las interfaces JDBC, que permiten la comunicación de la app con la base de datos. Para asegurarnos de que no hay más de una conexión abierta utilizamos una clase que sigue el patrón singleton, utilizando el nombre de la base de datos, el usuario, la contraseña y la url.

4. Instalación y configuración del servidor web

4.1 Descripción y justificación

Toda la configuración del servidor web está hecha en un sistema operativo Ubuntu Server 24.04. Usamos la infraestructura con Apache porque hemos probado con Nginx ya que era óptima para este proyecto, pero hemos tenido problemas configurándolo, así que decidimos trabajarla con Apache porque nos ha funcionado con menos problema y es la que más hemos usado en clase. Además, Apache es una plataforma segura, estable y compatible con diferentes tecnologías web. Esta es la configuración del docker-compose.yml que hemos usado:

```
GNU nano 7.2
version: '3'

services:
  apache:
    build: ./apache
    ports:
      - "8080:80"
    volumes:
      - ./apache:/usr/local/apache2/htdocs/
    restart: always

  ftp:
    image: fauria/vsftpd
    container_name: ftp
    ports:
      - "2121:21"
      - "2020:20"
      - "21100-21110:21100-21110"
    environment:
      - FTP_USER=usuario
      - FTP_PASS=contrasena
      - PASV_ADDRESS=127.0.0.1
      - PASV_MIN_PORT=21100
      - PASV_MAX_PORT=21110
      - VSFTPD_ENABLE_SSL=YES
    volumes:
      - ./apache:/home/vsftpd
    restart: always

  ssh:
    build: ./ssh
    ports:
      - "2223:22"
    volumes:
      - ./apache:/home/sshuser/www
    restart: always
```

4.2 Instalación y configuración

Guardamos los archivos que componen la página web dentro de la carpeta de apache, aquí estará el archivo index.html donde se aloja nuestra página web, a este se accede con el comando nano. Así sería la estructura de los directorios y archivos para la web, se observan los css y otros html que hemos usado para crear la página:

```
equipo1@servidor-web:~/servidor$ cd servidor
equipo1@servidor-web:~/servidor$ ls
apache  docker-compose.yml  ftp  ssh
equipo1@servidor-web:~/servidor$ cd apache
equipo1@servidor-web:~/servidor/apache$ ls
Dockerfile  estilos.css  GestorDeRutas.html  imagenesWeb  index.html  mierlo.html  nosotros.html  ordesa.html  usuario  web
equipo1@servidor-web:~/servidor/apache$
```

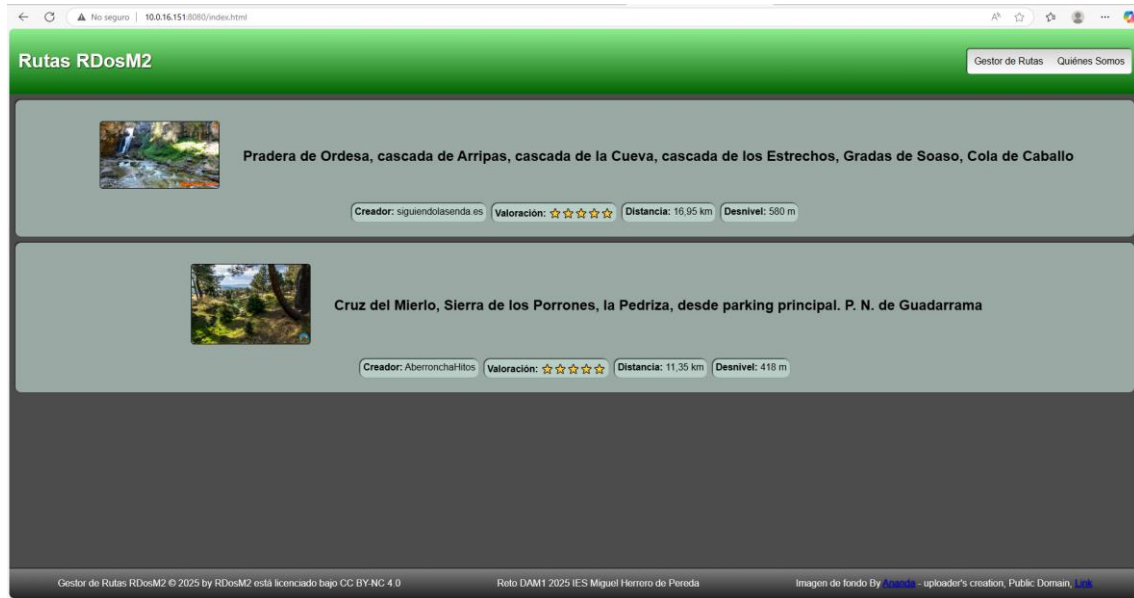
4.3 Acceso como cliente

-Para el cliente desde un sistema Windows:

Para acceder a ello con buscar en el navegador <http://10.0.16.151:8080> y vemos nuestra página web.

-Para el cliente desde un sistema Linux:

Al igual que en Windows, escribes en el navegador la dirección <http://10.0.16.151:8080>



4.4 Servidor FTP

4.4.1 Descripción y justificación

Se ha montado con una infraestructura docker y servidor vsftpd porque es rápido y muy seguro, también es compatible con FTP y SFTP y es estable.

4.4.2 Instalación y configuración

Escribimos el siguiente comando dentro de un contenedor de DockerHub, usamos la imagen fauria/vsftpd. En la configuración que hemos mostrado antes, se puede ver los puertos usados y el usuario y contraseña que hemos asignado (usuario y contraseña).

```
ftp:
  image: fauria/vsftpd
  container_name: ftp
  ports:
    - "2121:21"
    - "2020:20"
    - "21100-21110:21100-21110"
  environment:
    - FTP_USER=usuario
    - FTP_PASS=contrasena
    - PASV_ADDRESS=127.0.0.1
    - PASV_MIN_PORT=21100
    - PASV_MAX_PORT=21110
    - VSFTPD_ENABLE_SSL=YES
  volumes:
    - ./apache:/home/vsftpd
  restart: always
```

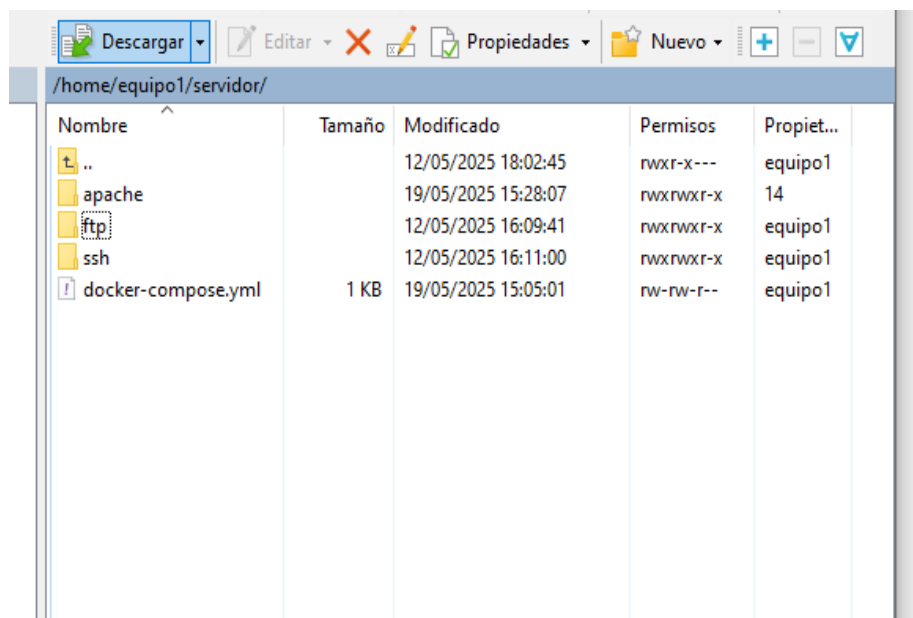

4.4.3 Acceso como cliente

-Para conectarse desde un sistema Windows:

Con una aplicación cliente de FTP como FileZilla o WinSCP por ejemplo, que es la que nosotros hemos usado para comprobar que funcione correctamente. También a través de la consola de comandos Hay que introducir la ip del servidor (10.0.16.151), el puerto 22, el usuario es “equipo1” y la contraseña es “equipo1”.

-Para acceder desde un sistema Linux:

Se puede a través de una aplicación cliente FTP como FileZilla o WinSCP. También desde la consola de comandos como cliente ssh nos podemos conectar, sería con el comando ssh [usuario@10.0.16.151](#).



Nombre	Tamaño	Modificado	Permisos	Propiet...
..		12/05/2025 18:02:45	rw-r-x---	equipo1
apache		19/05/2025 15:28:07	rw-rwxr-x	14
ftp		12/05/2025 16:09:41	rw-rwxr-x	equipo1
ssh		12/05/2025 16:11:00	rw-rwxr-x	equipo1
docker-compose.yml	1 KB	19/05/2025 15:05:01	rw-rw-r--	equipo1

4.5 Servidor SSH

4.5.1 Descripción y justificación

Se ha montado con una infraestructura docker y servidor OpenSSH porque son los métodos sobre los que hemos trabajado. Además, es un servidor seguro, permite copiar archivos con SFTP que hemos usado antes.

4.5.2 Instalación y configuración

Creamos el directorio ssh donde alojaremos este servidor. Dentro de la carpeta creamos un archivo Dockerfile en el que instalamos el servidor OpenSSH y añadimos nuestro usuario y contraseña. La estructura del directorio sería así:

```
equipo1@servidor-web:~/servidor$ ls
apache  docker-compose.yml  ftp  ssh  web
equipo1@servidor-web:~/servidor$ cd ssh
equipo1@servidor-web:~/servidor/ssh$ ls
Dockerfile
equipo1@servidor-web:~/servidor/ssh$
```

Y el contenido del Dockerfile es este:

```
GNU nano 7.2 Dockerfile
FROM ubuntu:latest

RUN apt update && apt install -y openssh-server && \
    useradd -ms /bin/bash sshuser && \
    echo "sshuser:contrasena" | chpasswd && \
    mkdir /var/run/sshd

EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]
```

4.5.3 Acceso como cliente

-Desde un cliente con sistema Windows:

Se accede desde una aplicación cliente ssh como puede ser Putty o a través de la consola de comandos y se introduce el comando: ssh usuario@10.0.16.151.

-Desde un cliente con sistema Linux:

Se puede acceder desde una aplicación cliente ssh como Putty, que es la que más hemos usado nosotros, o usando la consola de comandos, escribiendo el comando: ssh usuario@10.0.16.151.

Al igual que en el servidor de base de datos, en el del servidor web también hemos modificado la configuración para tener de ip estática la ip 10.0.16.151. Para ello editamos con el nano, de la carpeta netplan, el archivo 00-installer-config.yaml.

```
GNU nano 7.2
network:
  version: 2
  renderer: networkd
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 10.0.16.151/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
      routes:
        - to: default
          via: 10.0.16.1
```

5. Bibliografía

- DigitalOcean. (2020, 6 de noviembre). *How To Install MySQL on Ubuntu 20.04*. <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04>
- Educantabria. (s. f.). *SERVIDORES FTP-WEB-LINUX (RA6)* [Bloc de notas de clase]. SharePoint. https://educantabria.sharepoint.com/:o:/r/sites/DAM1-SISTEMASINFORMATICOScopia/_layouts/15/Doc.aspx?sourcedoc=%7BEF0AC31D-6980-49D3-B133-A636AEB68BDC%7D
- OpenAI. (s. f.). *ChatGPT*. OpenAI. <https://openai.com/es-ES/index/chatgpt/>