**RETO: GESTIÓN DE RUTAS EN ACTIVIDADES AL AIRE LIBRE**



**Dam1 – Equipo2**

**Integrantes:**

* Rubén Cobo González
* Roberto Sainz Arjona
* Guillermo Muñoz Pando
* Pablo Palencia Castillo
* Abraham Bekele Sánchez

Contenido

[SERVIDOR DE BASE DE DATOS 2](#_Toc197609878)

[SERVIDOR WEB 2](#_Toc197609879)

[SERVIDOR FTP 3](#_Toc197609880)

[SERVIDOR SSH 3](#_Toc197609881)

[CONEXIÓN DESDE CLIENTES 4](#_Toc197609882)

[BIBLIOGRAFIA Y FUENTES 4](#_Toc197609883)

**Diagrama de Red de la Arquitectura de la Aplicación**

Diagrama

El contenido generado por IA puede ser incorrecto.

# SERVIDOR DE BASE DE DATOS

## 1. Descripción y Justificación de la Elección

* **Tecnología elegida:** Se ha utilizado **VirtualBox** para montar una máquina virtual Ubuntu Server con MySQL y AWS con Ubuntu server.
* **Motivo de elección:** Se comparó VirtualBox con AWS y se eligió VirtualBox porque:
  + Es una solución gratuita y de código abierto.
  + Permite trabajar de manera local sin necesidad de conexión permanente a Internet.
  + Es ideal para entornos de laboratorio y pruebas en equipos con recursos limitados.
  + Facilita la exportación de máquinas en formato .ova, cumpliendo los requisitos del reto.
    - Aspectos negativos de AWS:
      * AWS no es 100% gratuito, mientras VirtualBox si lo es.
      * Es más complejo que VirtualBox.
      * AWS depende completamente de tu conexión a internet, mientras con VirtualBox puedes trabajar 100% offline sin depender de la red.

## 2. Requisitos Técnicos

* **Requisitos de Hardware de la Máquina Virtual**
  + **CPU**: 2 núcleos (mínimo)
  + **Memoria RAM**: 2 GB (mínimo), recomendado 4 GB o más para mejor rendimiento
  + **Almacenamiento**: 20 GB (mínimo), recomendado 50 GB si se espera crecimiento en la base de datos
  + **Adaptador de red**: Configurado en modo *Adaptador puente* o *NAT* con puertos redirigidos
  + **Acceso remoto**: Puertos abiertos para MySQL (3306), SSH (22)
* **Requisitos de software - Sistema Operativo**:
  + Ubuntu Server 22.04 LTS (recomendado)
  + imagen ISO descargada desde: <https://ubuntu.com/download/server>
* **Base de Datos**:
  + MySQL Server última versión estable
  + Cliente MySQL para administración y conexión remota
* **Herramientas adicionales (ambas se han utilizado en la configuración del servidor)**:
  + ufw (Firewall simple de Ubuntu para proteger puertos)
  + nano (Editor de texto para configuración)
* **Requisitos de Red**
  + **IP fija** en la red local
* **Puertos**:
  + 3306/TCP → MySQL
  + 22/TCP → SSH para administración remota
* **Requisitos de Seguridad**
* Configuración de usuarios y contraseñas seguras en MySQL
* Uso de autenticación *caching\_sha2\_password* (Requiere mayúscula, minúscula, carácter numérico y carácter especial)
* Configuración de Firewall (ufw) para permitir solo IPs específicas

## 3. Proceso de Instalación, Configuración y Despliegue

* **Paso 1: Descarga de la Imagen Base de la Máquina Virtual**
* Descargar la imagen ISO de Ubuntu Server 22.04 LTS desde la página oficial:  
  <https://ubuntu.com/download/server>
* Crear una nueva máquina virtual en VirtualBox con los requisitos técnicos especificados.
* **Paso 2: Instalación del Sistema Operativo (Ubuntu Server)**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

* + Montar la ISO en la VM y arrancar la instalación.
  + Seleccionar idioma, teclado y configuración regional.
  + Asignar un nombre al servidor
  + Configurar la red: se recomienda asignar una IP fija para facilitar las conexiones.
  + Crear un usuario administrador y establecer una contraseña segura.
  + Habilitar la opción de instalación de **OpenSSH Server** para administración remota.
  + Finalizar la instalación y reiniciar la máquina virtual.
* **Paso 3: Instalación de MySQL**
* **Paso 4: Configuración de Usuarios, Contraseñas y Puertos**
* Asegurar la instalación con este comando:

sudo mysql\_secure\_installation

* Establecer contraseña para el usuario root.
* Eliminar usuarios anónimos.
* Deshabilitar acceso remoto para root (recomendado).
* Eliminar base de datos de prueba.
* Recargar las tablas de privilegios.
* Crear un usuario para la aplicación con permisos específicos:

sudo mysql -u root -p

* Dentro del prompt de MySQL:

CREATE USER 'equipo2'@'%' IDENTIFIED BY 'Equipo2@';

GRANT ALL PRIVILEGES ON \*.\* TO 'equipo2'@'%' WITH GRANT OPTION;

FLUSH PRIVILEGES;

* Editar el archivo de configuración para permitir conexiones remotas:

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

* Dentro de este menú hay que cambiar:

bind-address = 0.0.0.0

* Reiniciar el servicio de MySQL:

sudo systemctl restart mysql

* Abrir el puerto 3306 en el firewall:

sudo ufw allow 3306/tcp

sudo ufw enable

sudo ufw status

* **Paso 5: Proceso de Carga de Datos**

Transferir el archivo .sql de la base de datos al servidor (por FTP, SCP o USB).

* Importar la base de datos:

mysql -u appuser -p < /ruta/a/rutas\_airelibre.sql

* Verificar que las tablas y registros están cargados:

SHOW DATABASES;

USE rutas\_airelibre;

SHOW TABLES;

## 5. Acceso desde la Aplicación

* **IP del servidor**: 10.0.2.15
* **Puerto**: 3306
* **Usuario**: equipo2
* **Contraseña**: Equipo2@
* **Pruebas de conexión realizadas**:
  + Desde clientes remotos mediante **MySQL Workbench** o mediante la propia aplicación de escritorio.

# SERVIDOR WEB

**1. Descripción y Justificación de la Elección**

* **Tecnologías probadas:** Se ha utilizado VirtualBox para montar una máquina virtual Ubuntu Server con Apache y con Nginx.
* **Elección final y motivo:** Al final se ha elegido Apache porque:
  + Al ser uno de los servidores más antiguos y usados, hay muchísima documentación, foros, y guías para casi cualquier cosa.
  + Muchos scripts están diseñados pensando en Apache y sus módulos.
  + Apache permite proteger directorios con autenticación de forma sencilla.
  + En entornos donde varios usuarios tienen sitios web en el mismo servidor, Apache permite configurar reglas por usuario gracias a su sistema de permisos.
    - Aspectos negativos de Nginx:
      * Nginx para configuraciones complejas sigue teniendo menos recursos disponibles online.
      * Los módulos en Nginx son menos flexibles.
      * Nginx requiere más configuración manual para PHP y reescrituras.

**2. Proceso de Instalación, Configuración y Despliegue**

* Paso 1: Instalación de Apache
* **Actualiza los paquetes:**

sudo apt update && sudo apt upgrade -y

* **Instala Apache:**

sudo apt install apache2 -y

* **Habilita y arranca Apache:**

sudo systemctl enable apache2

sudo systemctl start apache2

* **Prueba que funciona (en el navegador):**
* Escribe en tu navegador: http://10.0.16.31
* Deberías ver la página de bienvenida de Apache.
* Paso 2: Configuración del Virtual Host
  + - Crea un directorio para tu web:

sudo mkdir -p /var/www/miweb

* + - Pon los permisos correctos

sudo chown -R $USER:$USER /var/www/miweb

* + - Crea un archivo de configuración para tu sitio:

sudo nano /etc/apache2/sites-available/miweb.conf

* + - Agrega este Código para alojar la página HTML:

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

ServerName IP\_o\_Dominio

DocumentRoot /var/www/miweb

<Directory /var/www/miweb>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

* + - Activa el sitio y desactiva el otro sitio por defecto:

sudo a2dissite 000-default.conf

sudo a2ensite miweb.conf

* + - Por ultimo reinicia Apache :

sudo systemctl reload apache2

* Paso 3: Despliegue de la web generada en el módulo de Lenguajes de Marcas
  + - Copia los archivos de tu proyecto web a la carpeta:

sudo cp -r /ruta/del/proyecto/\* /var/www/miweb/

* + - Dale los permisos correctos:

sudo chown -R equipo2:equipo2 /var/www/miweb

sudo chmod -R 755 /var/www/miweb

* + - Reinicia Apache de nuevo:

Sudo systemctl restart apache2

**3. Acceso desde Clientes**

* URL de prueba: https://10.0.16.31
* Resultado esperado: carga correcta del sitio web

## SERVIDOR FTP

**1. Descripción y Justificación**

* **Servidor FTP elegido:** vsftpd
* **Motivo de elección:**
  + **Seguridad:** vsftpd es conocido por ser uno de los servidores FTP más seguros disponibles.
  + **Rendimiento:** Ligero y eficiente, ideal para entornos productivos y servidores con pocos recursos.
  + **Compatibilidad:** Soporta FTPS (FTP sobre TLS), lo que permite conexiones seguras.
  + **Estabilidad:** Utilizado por grandes distribuciones Linux como Ubuntu, Debian y CentOS.
  + **Facilidad de integración:** Compatible con herramientas gráficas como WinSCP, FileZilla, etc.

**2. Proceso de Instalación, Configuración y Despliegue**

* Paso 1: Instalación del servidor FTP

sudo apt update

sudo apt install vsftpd

* Paso 2: Configuración con autenticación y modo seguro (FTPS)
  + - Haz una copia de seguridad del archivo de configuración original por si acaso:

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.backup

* + - Edita el archivo de configuración:

sudo nano /etc/vsftpd.conf

* + - Modifica o asegura las siguientes líneas:

# Habilitar escritura

write\_enable=YES

# Restringir a su carpeta home

chroot\_local\_user=YES

# Habilitar SSL/TLS

ssl\_enable=YES

rsa\_cert\_file=/etc/ssl/certs/ssl-cert-snakeoil.pem

rsa\_private\_key\_file=/etc/ssl/private/ssl-cert-snakeoil.key

* + - Reinicia el servidor FTP para aplicar los cambios:

sudo systemctl restart vsftpd

* Paso 3: Creación de usuarios FTP
  + - Crea un usuario para usar el FTP:

sudo adduser usuarioftp

* + - Establece permisos para su carpeta para poder subir los archivos que quieras:

sudo mkdir -p /home/usuarioftp/ftp

sudo chown nobody:nogroup /home/usuarioftp/ftp

sudo chmod a-w /home/usuarioftp/ftp

# Carpeta donde puede subir archivos

sudo mkdir /home/usuarioftp/ftp/archivos

sudo chown usuarioftp:usuarioftp /home/usuarioftp/ftp/archivos

**3. Acceso desde Clientes**

* Cliente FTP usado: Winscp
* Protocolo: SFTP
* Servidor: 10.0.16.31
* Usuario: usuario ftp o equipo2
* Contraseña: equipo2
* Puerto: 22

## SERVIDOR SSH

**1. Descripción y Justificación**

* **SSH estándar de OpenSSH**
* **Motivo:**

**2. Proceso de Instalación, Configuración y Despliegue**

* Paso 1:
* Paso 2: **3. Acceso desde Clientes**
* Cliente usado:
* Comando usado:
* Pruebas realizadas y capturas adjuntas

# CONEXIÓN DESDE CLIENTES

**Pruebas Realizadas**

| **Servicio** | **Herramienta Cliente** | **Resultado** |
| --- | --- | --- |
| Navegador (HTTPS) | Chrome / Firefox | OK |
| SSH |
| FTP seguro | WinScp | OK |

*(Falta Incluir capturas de pantalla de cada conexión exitosa)*

# BIBLIOGRAFIA Y FUENTES

* Documentación oficial de MySQL: <https://dev.mysql.com/doc/>
* Guía oficial de Nginx: https://nginx.org/en/docs/
* Foro StackOverflow
* Tutoriales DigitalOcean

**Anexo**

* **Enlace a la máquina virtual exportada (.ova):** []
* **Repositorio GitHub con documentación y enlaces:** []

[**--- VOLVER AL INICIO ---**](#INICIO)