

**RETO: GESTIÓN DE ACTIVIDADES EXTRAESCOLARES**

**DAM1\_Equipo1 :**

Alumno 1: Fabian Saiz Landeras.

Alumno 2: Izan Franco Anduaga.

Alumno 3: Ciro Galán Vertí.

Alumno 4: Raúl Buenaga García.

Alumno 5: Francisco Sitjar de Cos-Estrada.

Alumno 6: Diego Bolloque Somoano.

**INDICE**

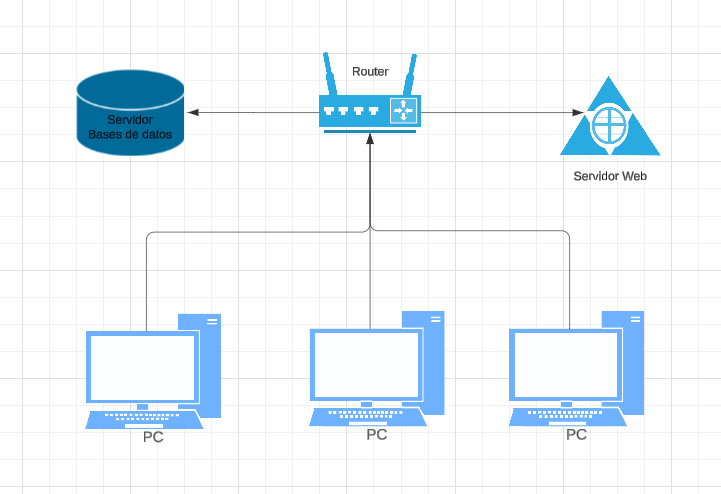
[1.Diagrama de red de la arquitectura de la aplicación. 2](#_Toc532061909)

[2.Servidor de base de datos. 3](#_Toc198491790)

[3.Servidor web/FTP/SSH. 5](#_Toc245894532)

[4.Bibliografía /Fuentes. 6](#_Toc1197748952)

# **Diagrama de red de la arquitectura de la infraestructura.**



En este diagrama de red de la infraestructura se puede ver como 3 ordenadores o PC’s están conectados a un router en conjunto que a su vez este está conectado al servidor de base de datos de MariaDB y también está conectado el router al servidor Web.

# **2. Servidor de base de datos.**

**2.1.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:**

El servidor donde se alojan los datos está en una base de datos MariaDB, en una máquina virtual de Linux Alpine la cual es compatible con la versión de MySQL 8.3.0 y para hacer la base de datos hemos creado un cliente MySQL.

Para ello instalamos el docker en el Alpine, se ha elegido Alpine porque ocupa poco espacio y es sencillo de usar.

**2.2.2. Proceso de instalación, configuración, despliegue del servidor, proceso de carga de datos y acceso desde la aplicación.**

Una vez dentro de Alpine, hacemos un apk update.

Y actualizados los paquetes usamos apk add --update docker openrc.

Usamos rc-update add docker boot para que se inicie al encender la maquina Alpine.

Comandos usados:

**docker run --name mariadb -p 3306:3306**

**-v $HOME/docker/mariadb/config:/var/lib/mysql**

**-v $HOME/docker/mariadb/conf.d:/etc/mysql/conf.d**

**-e MYSQL\_ROOT\_PASSWORD=123**

**-d mariadb**

**rc-update add docker boot**

**docker update mariadb –restart unless-stopped**

Con estos dos últimos comandos aseguramos que al iniciar la maquina donde se almacena el servidor la base de datos se inicialice automáticamente.

La base de datos va desde el puerto 3306 del host al puerto 3306 del contenedor mariadb.

Tenemos una copia de seguridad y un archivo de configuración del servidor donde quitamos privilegios.

Creamos el usuario gestor con permisos de administrador para nosotros.

A la base accedemos desde workbench donde metimos los datos de los cursos, grupos, departamentos y profesores.

La aplicación devuelve una sentencia sql a la base de datos cada vez que recibe una modificación

# **3. Servidor web/FTP/SSH.**

**3.1. WEB:**

**3.1.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:**

Un servidor web almacena los archivos que forman parte de un sitio electrónico o de una base de datos. El servidor funge como transmisor de esta información de acuerdo con las demandas de sus usuarios. De este modo, clientes y consumidores pueden acceder a la información solicitada vía remota.

Para el servidor web usamos una máquina virtual de Alpine, por el poco espacio que ocupa y por su facilidad al usarla.

**3.1.2. Proceso de instalación, configuración, despliegue del servidor y el acceso desde clientes:**

Instalaremos apache con el comando apk add nano apache2 apache2-ssl apache2-http2 –U.

Hacemos una copia de seguridad del archivo ssl.conf como ssl.conf.bak que es la extensión de la copia de seguridad con el comando:

-cp /etc/apache2/conf.d/ssl.conf /etc/apache2/conf.d/ssl:conf.bak

Después con el nano hemos abierto el archivo httpd.conf :

-nano /etc/apache2/httpd.conf.

Y buscamos la parte del Servername con el nombre del dominio y el puerto.

Lo cambiamos a www.retodam1.com:80 y hacemos un rc-update add apache2.

Iniciamos el servidor con rc-service apache2 start.

Volvemos a entrar al httpd.conf y en <IfModule dir\_module>

Se encuentra el archivo que abre por defecto que en este caso es index.html.

Vamos a activar el UserDir que permite a los usuarios tener su propia página web alojada en una carpeta (dentro de su carpeta personal) llamada public\_html.

Para ello volvemos a entrar en el httpd.conf y quitamos la almohadilla de comentado de la linea que pone:

- #LocalModule userdir\_module modules/mod\_userdir.so.

**3.2. SSH:**

**3.2.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:**

El SSH (Secure Shell) es un protocolo de red que permite el acceso remoto a través de una conexión cifrada. Proporciona una autenticación robusta y es compatible con el inicio de sesión remoto seguro, la ejecución de comandos, la transferencia de archivos, el control de acceso, el reenvío de TCP/IP, etc.

Para el servidor SSH usamos una máquina virtual de Alpine, por el poco espacio que ocupa y por su sencillez a la hora de usarla.

**3.2.2. Proceso de instalación, configuración, despliegue del servidor y el acceso desde clientes:**

Para el SSH instalamos el openssh con:

- Apk add openssh.

Actualizamos los paquetes del openssh con:

- rc-update add sshd.

Hacemos que se habilite de manera automática al abrir la maquina con:

- rc-service sshd start.

Quitamos comentarios en:

- UseDNS no # By setting this to no, connection speed can increase.

- Port 443 # Use whichever port number fits your needs

**3.3. FTP:**

**3.2.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:**

El FTP es un protocolo que se utiliza para transferir todo tipo de archivos entre equipos conectados a una red, por ejemplo Internet. Las cuentas de FTP son el medio estándar para gestionar el contenido alojado en un servidor web: enviar archivos, descargarlos, crear directorios, borrar ficheros, etc.

Para el servidor FTP usamos una máquina virtual de Alpine, por el poco espacio que ocupa y por su sencillez a la hora de usarla.

**3.2.2. Proceso de instalación, configuración, despliegue del servidor y el acceso desde clientes:**

En nuestro caso el Servidor FTP que usaremos será vsftpd, para instalarlo usamos:

- Apk add vsftpd.

Para configurar el servicio sftp desde nuestro usuario root creamos un grupo con addgroup sftpgroup y configuramos un usuario

-adduser –G sftpgroup –D –h /var/www/localhost/htdocs -s /sbin/nologin admin

* -G sftpgroup incluimos el usuario ‘admin’ en el grupo creado anteriormente
* –D hacemos que el usuario no tenga contraseña ya que la configuraremos después
* –h /var/www/localhost/htdocs especificamos que la carpeta raíz de nuestro usuario sea /var/www/localhost/htdocs donde tambien almacenaremos la pagina web
* –s /sbin/nologin establecemos que el usuario no pueda iniciar sesion directamente en linea de comandos y que solo pueda acceder a unos recursos específicos

A continuacion asignamos una contraseña al usuario con passwd admin

Configuramos los permisos de lectura y ejecución para el grupo en el directorio donde alojaremos nuestras páginas web con chmod g+rx /var/www/localhost/htdocs

Luego accedemos al archivo sshd\_config por medio de nano /etc/ssh/sshd\_config para coinfigurar el servicio ssh donde habilitaremos la autenticacion por contraseña cambiando la linea PasswordAuthentication no por PasswordAuthentication yes

Finalmente reiniciamos los servicios ssh y vsftp con rc-service sshd restart y rc-service vsftpd restart y comprobamos la funcionalidad del servidor accediendo con un cliente sftp en nuestro caso utilizando Filezilla desde un S.O. Windows y un S.O. Ubuntu

**4.Bibliografía /Fuentes.**

<https://www.youtube.com/watch?v=dEStH-1trdc> ==> Apache  
<https://wiki.alpinelinux.org/wiki/Setting_up_a_SSH_server> ==> SSH  
<https://wiki.alpinelinux.org/wiki/FTP> ==> FTP  
<https://wiki.alpinelinux.org/wiki/MariaDB> ==> MariaDB