



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.

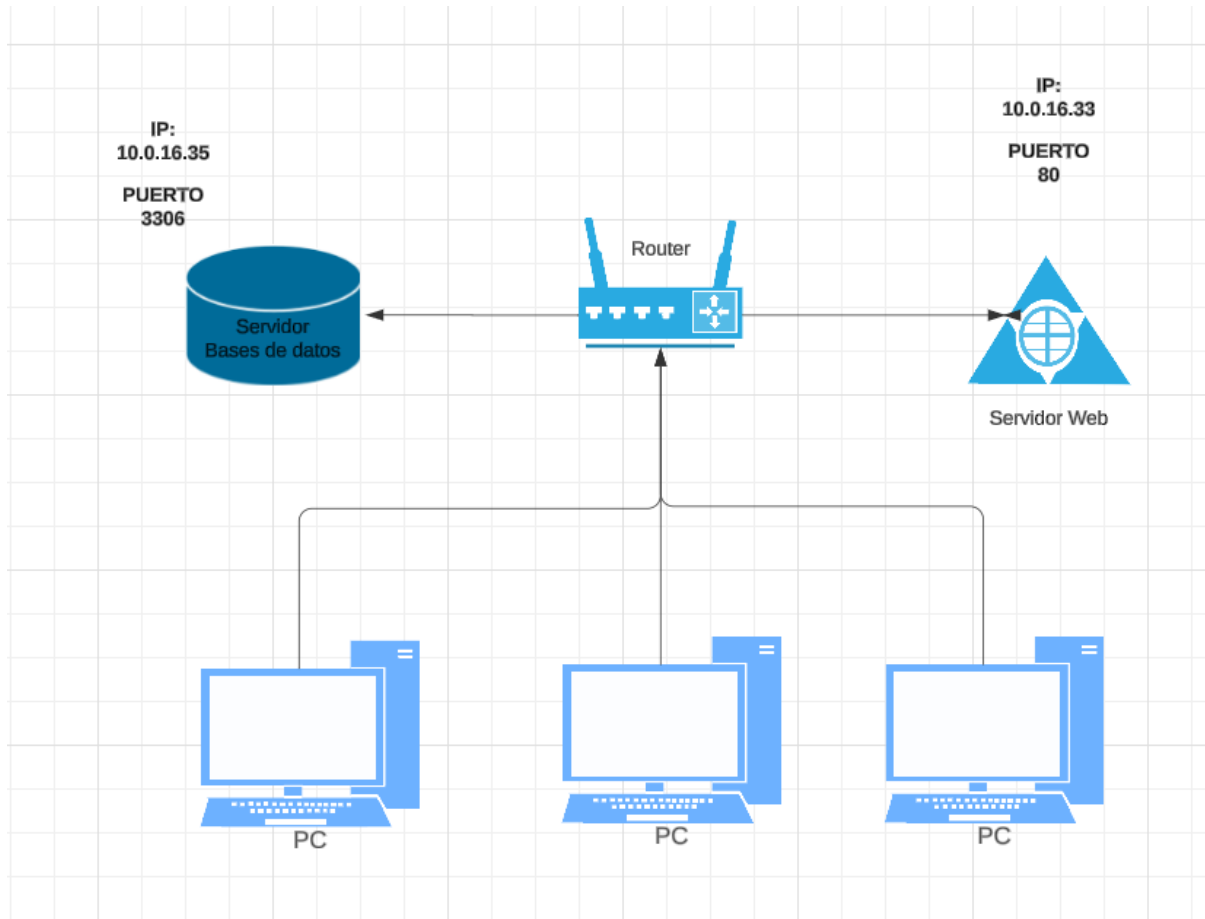
Alumno 5: Francisco Sitjar de Cos-Estrada.

Alumno 6: Diego Bolloque Somoano.

INDICE

1.Diagrama de red de la arquitectura de la aplicación.	3
2.Servidor de base de datos.	4
2.1.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:	4
2.2.2. Proceso de instalación, configuración, despliegue del servidor, proceso de carga de datos y acceso desde la aplicación:	4
3.Servidor web/FTP/SSH.	6
3.1.WEB:	6
3.1.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:	6
3.1.2. Proceso de instalación, configuración, despliegue del servidor y el acceso desde clientes:	6
3.2.SSH:	7
3.2.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:	7
3.2.2. Proceso de instalación, configuración, despliegue del servidor y el acceso desde clientes:	7
3.3.FTP:	8
3.2.1. Descripción y justificación de la elección y de la infraestructura sobre la que se ha montado:	8
3.2.2. Proceso de instalación, configuración, despliegue del servidor y el acceso desde clientes:	8
4.Bibliografía /Fuentes.....	10

1. Diagrama de red de la arquitectura de la aplicación.



En este diagrama de red de la aplicación 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.

Las máquinas clientes representan entidades de software que realizan peticiones de servicios a los proveedores de este, una máquina cliente lanzará una petición en forma de mensaje a un servidor de red que las procesará y después de este proceso, el servidor transmitirá la respuesta al cliente.

2.Servidor de base de datos.

DESCARGAR MAQUINA VIRTUAL

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

Descripción:

-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.

Justificación de la elección y de la infraestructura:

-Hemos elegido la máquina virtual de Alpine porque es perfecta para trabajar con nuestra base de datos debido a que cumple con los requisitos pedidos, también porque pesa muy poco lo cual es una gran ventaja y hemos usado docker porque es más fácil encontrar máquinas bien configuradas.

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

Para ello, hemos añadido docker y hemos hecho que se inicie automáticamente al encender la maquina Alpine

Comandos usados para crear e iniciar el contendor usado:

```
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
```

Usuario de Alpine:Root

Contraseña de Alpine:Root

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

GUÍA DE DESPLIEGUE

donde quitamos los privilegios y creamos el usuario gestor con permisos de administrador para nosotros.

Acceso al servidor:

A la base accedemos desde Workbench introduciendo la dirección ip 10.0.16.35, el usuario gestor, la contraseña root y el puerto 3306 correspondientes.

Acceso desde la aplicación:

```
private Connection conn = null;
// constantes a definir
private static final String BD = "Base_Datos_Reto";
private static final String USUARIO = "gestor";
private static final String CLAVE = "root";
private static final String URL = "jdbc:mysql://10.0.16.33:3306/" + BD;

/**
 * Constructor privado que crea una instancia para la conexión con la base de datos.
 * Utiliza el patrón Singleton para instanciar la clase una unica vez.
 */
private AccesoBaseDatos() {

    try {
        Properties properties = new Properties();
        properties.setProperty(key:"user", value: USUARIO);
        properties.setProperty(key:"password", value: CLAVE);
        properties.setProperty(key:"useSSL", value: "false");
        properties.setProperty(key:"autoReconnect", value: "true");
        conn = (Connection) DriverManager.getConnection(url:URL, info:properties);
        if (conn == null) {
            System.out.println("Error en conexion");
        } else {
            System.out.println("Conexion correcta a: " + URL);
        }
    } catch (SQLException ex) {
        // handle any errors
        System.out.println("SQLException: " + ex.getMessage());
        System.out.println("SQLState: " + ex.getSQLState());
        System.out.println("VendorError: " + ex.getErrorCode());
    }
}

/**
 * Este método devuelve la instancia para la conexión.
 * @return La instancia AccesoBaseDatos.
 */
public static AccesoBaseDatos getInstance() {
    return AccesoBaseDatosHolder.INSTANCE;
}
```

3.Servidor web/FTP/SSH.

3.1.WEB:

[DESCARGAR MAQUINA VIRTUAL](#)

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

Descripción:

-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.

Justificación de la elección y de la infraestructura:

Hemos elegido la máquina virtual de Alpine porque es perfecta para trabajar con nuestra base de datos debido a que cumple con los requisitos pedidos y también porque pesa muy poco. Se ha elegido Apache2 porque es un software estable, fiable, de código abierto y gratuito.

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 comentario de la línea 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:

Descripción:

-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.

Justificación de la elección y de la infraestructura:

-Para el servidor SSH usamos una máquina virtual de Alpine, por el poco espacio que ocupa y porque para su instalación es muy sencillo en Alpine.

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 máquina 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:

Descripción:

-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.

Justificación de la elección y de la infraestructura:

-Para el servidor FTP usamos “vsftpd” porque es el predeterminado de la máquina Alpine.

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 también almacenaremos la página web.
- -s /sbin/nologin establecemos que el usuario no pueda iniciar sesión directamente en línea de comandos y que solo pueda acceder a unos recursos específicos.

A continuación, 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

GUÍA DE DESPLIEGUE

Luego accedemos al archivo `sshd_config` por medio de `nano /etc/ssh/sshd_config` para configurar el servicio `ssh` donde habilitaremos la autenticación por contraseña cambiando la línea `PasswordAuthentication` no por `PubkeyAuthentication` yes.

Finalmente reiniciamos los servicios `ssh` y `vsftpd` 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

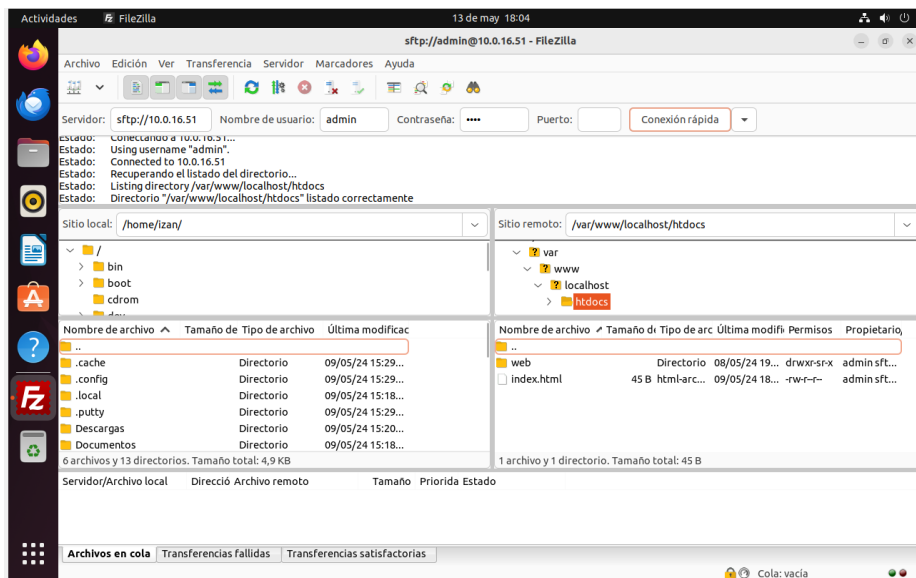
Usuario: admin

Contraseña: root

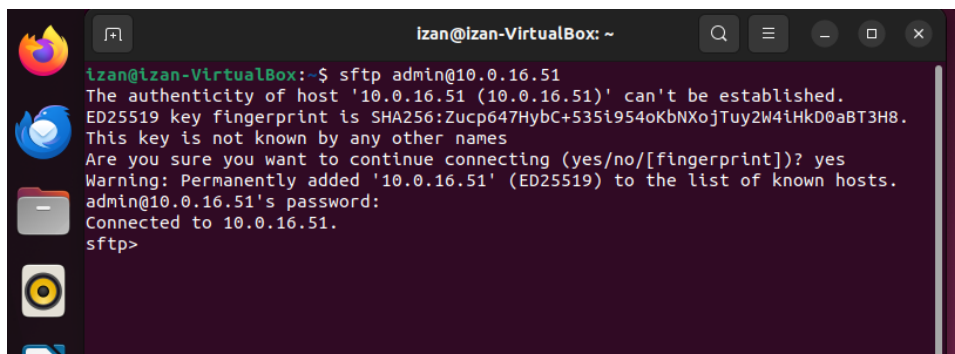
Ip servidor: 10.0.16.51 (ip dinámica)

Puerto: 22

Conexión desde cliente Ubuntu:

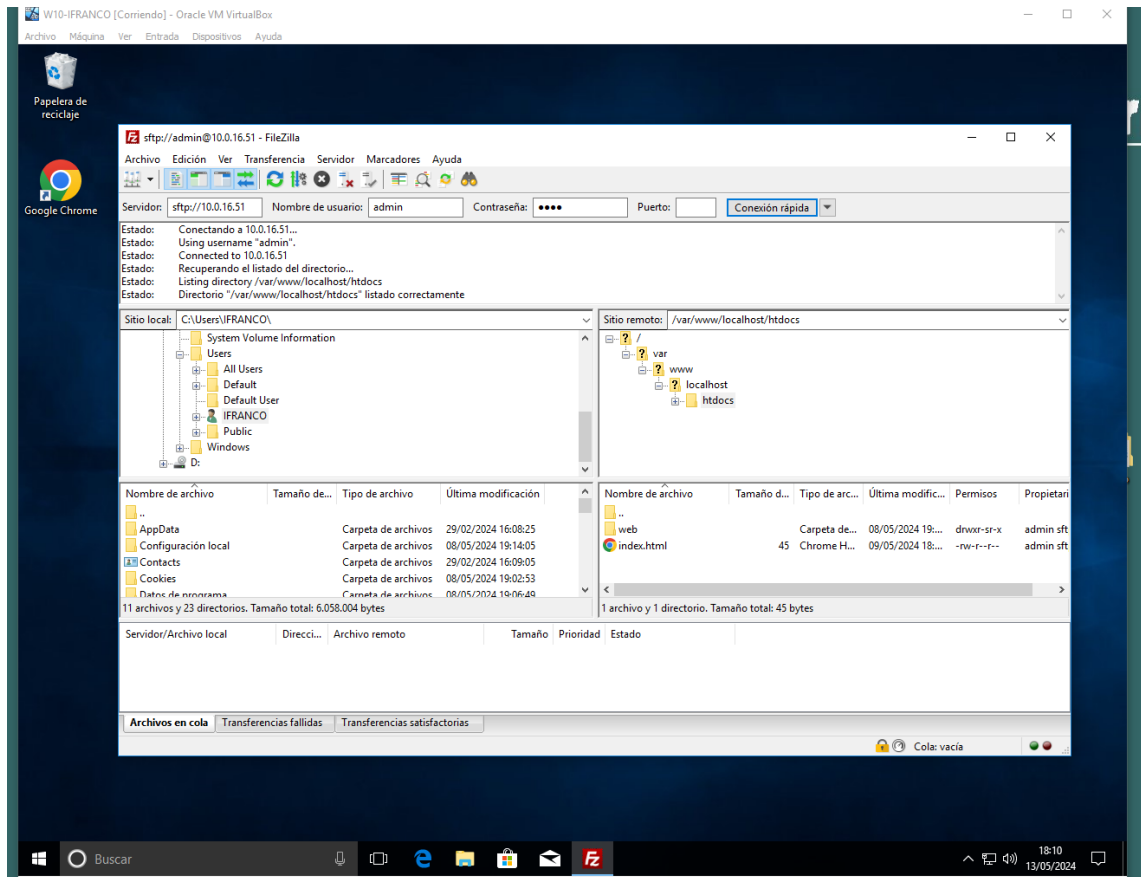


Conexión al servidor por línea de comandos Ubuntu:



Conexión desde cliente Windows:

GUÍA DE DESPLIEGUE



Conexión al servidor por la línea de comandos Windows:

```
Símbolo del sistema - sftp admin@10.0.16.51

Microsoft Windows [Versión 10.0.19045.4291]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\Usuarios\dam122>sftp admin@10.0.16.51
The authenticity of host '10.0.16.51 (10.0.16.51)' can't be established.
ECDSA key fingerprint is SHA256:Uh8kSwQM8gpIpPxMdGkQ6d1C5VwVUIGyXVITsP2wOEY.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '10.0.16.51' (ECDSA) to the list of known hosts.
admin@10.0.16.51's password:
Connected to 10.0.16.51.
sftp>
```

4.Bibliografía /Fuentes.

<https://www.youtube.com/watch?v=dEStH-1trdc> ==> Apache

https://wiki.alpinelinux.org/wiki/Setting_up_a_SSH_server ==> SSH

GUÍA DE DESPLIEGUE

<https://wiki.alpinelinux.org/wiki/FTP> ==> FTP

<https://wiki.alpinelinux.org/wiki/MariaDB> ==> MariaDB

