

Ad-2 Servidores

Aclaraciones

Para la realización de esta práctica se ha utilizado una máquina virtual con sistema Debian 12. En orden de simular un equipo real, se esta usando de Network settings **Bridged Adapter** para que la máquina tenga su propia ip en la red.

- Nombre de la máquina: **debianServer**
- Usuario: **username**
- Contraseña: **password**

Durante la instalación del sistema no se han instalado ni entornos gráficos ni programas adicionales.

Se ha creado un usuario llamado **username**

Sistema

Especificaciones

Sistema Debian 12

- 2Gb RAM
- 2 CPU

Configuración base

1. Se inicia sesión con el usuario **root**
2. Se actualiza el sistema con el comando

```
apt update && apt upgrade -y
```

3. Se instalan los programas base con el comando

- sudo para dar permisos de administrador al usuario **username**
- vim para editar archivos
- net-tools para comprobar la configuración de red
- passwd para cambiar la contraseña de los usuarios en caso de ser necesario

```
apt install -y sudo vim net-tools passwd
```

1. Se dan permisos de sudo al usuario **username** con el comando

```
usermod -aG sudo username
```

5. Iniciar sesión con el usuario `username` y comprobar que tiene permisos de sudo con el comando

```
su - username  
sudo echo "hola"
```

Firewall

1. Instalar el firewall con el comando

```
sudo apt install -y ufw
```

2. Habilitar el firewall con el comando

```
sudo ufw enable
```

Servicio SSH

1. Instalar el servidor SSH con el comando

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

2. Comprobar que el servicio esta activo con el comando

```
sudo systemctl status ssh
```

3. Añadir una regla al firewall para permitir el tráfico SSH con el comando

```
sudo ufw allow OpenSSH  
sudo ufw allow ssh  
sudo ufw reload
```

4. Conectar por SSH desde el equipo local con el comando

```
ssh -p 22 user@ip
```

Utilizar el comando `sudo ufw status` para comprobar que las reglas se han añadido correctamente. Nos debe mostrar algo similar a esto:

```
user@debianServer:~$ sudo ufw status
Status: active
```

| To | Action | From |
|--------------|--------|---------------|
| -- | ----- | ---- |
| OpenSSH | ALLOW | Anywhere |
| 22/tcp | ALLOW | Anywhere |
| OpenSSH (v6) | ALLOW | Anywhere (v6) |
| 22/tcp (v6) | ALLOW | Anywhere (v6) |

Servicio FTP

1. Instalar el servidor FTP con el comando

```
sudo apt install -y vsftpd ftp
```

2. Añadir una regla al firewall para permitir el tráfico FTP con el comando

```
sudo ufw allow ftp
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw reload
```

Estado de la configuración del firewall

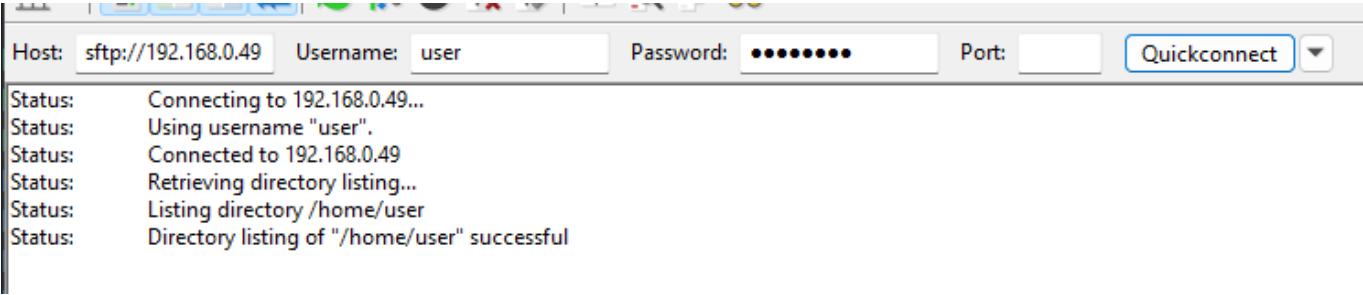
```
Status: active
```

| To | Action | From |
|--------------|--------|---------------|
| -- | ----- | ---- |
| OpenSSH | ALLOW | Anywhere |
| 22/tcp | ALLOW | Anywhere |
| 21/tcp | ALLOW | Anywhere |
| 20/tcp | ALLOW | Anywhere |
| OpenSSH (v6) | ALLOW | Anywhere (v6) |
| 22/tcp (v6) | ALLOW | Anywhere (v6) |
| 21/tcp (v6) | ALLOW | Anywhere (v6) |
| 20/tcp (v6) | ALLOW | Anywhere (v6) |

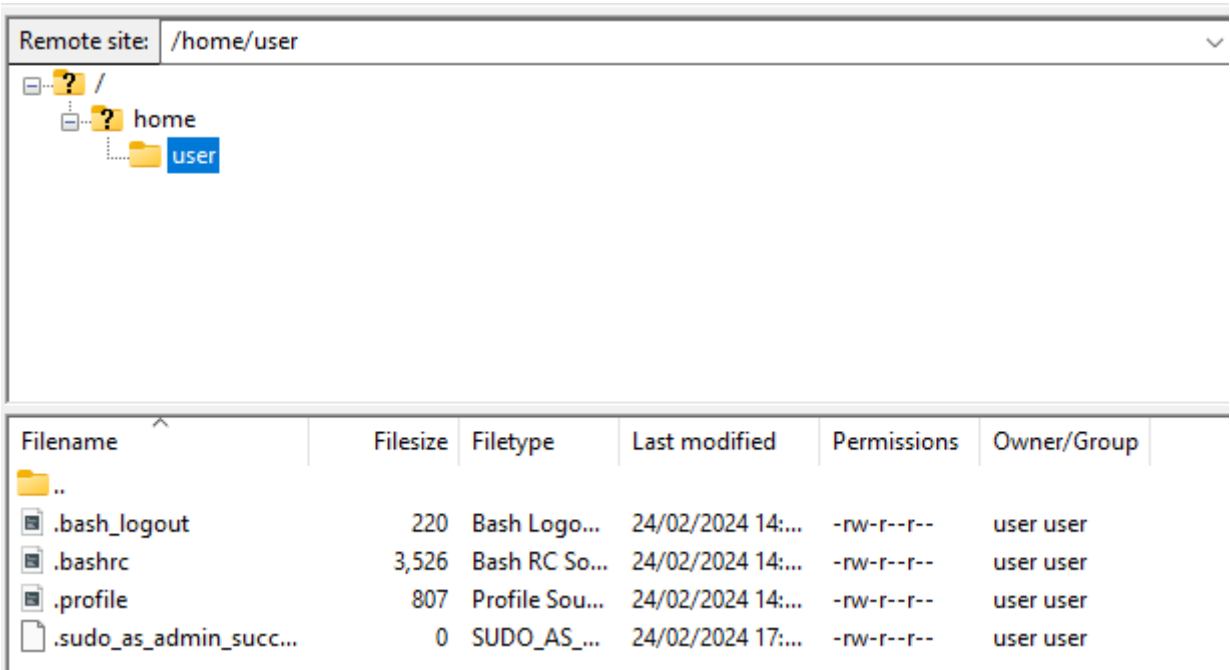
3. Comprobar que el servicio esta activo con el comando

```
sudo systemctl status vsftpd
```

4. Conectar por ftp desde el equipo local con filezilla



Poder ver el contenido de la carpeta `/home/username` desde el equipo local



Conclusiones

Hasta este punto se ha configurado el sistema base con un firewall, un servidor SSH y un servidor FTP. Esto nos permitirá acceder al sistema de forma remota gracias al servicio SSH y subir archivos al sistema gracias al servicio FTP.

Aplicación 1

Para la aplicación 1 se ha decidido montar un servidor Apache encargado de alojar una aplicación web desarrollada con PHP.

Tecnologías

- Lenguaje de front
 - HTML5
 - CSS3
 - Javascript
- Lenguaje de back
 - PHP
- BBDD

- MySQL

Instalación

Servidor Apache

1. Instalar el servidor Apache con el comando

```
sudo apt install -y apache2
```

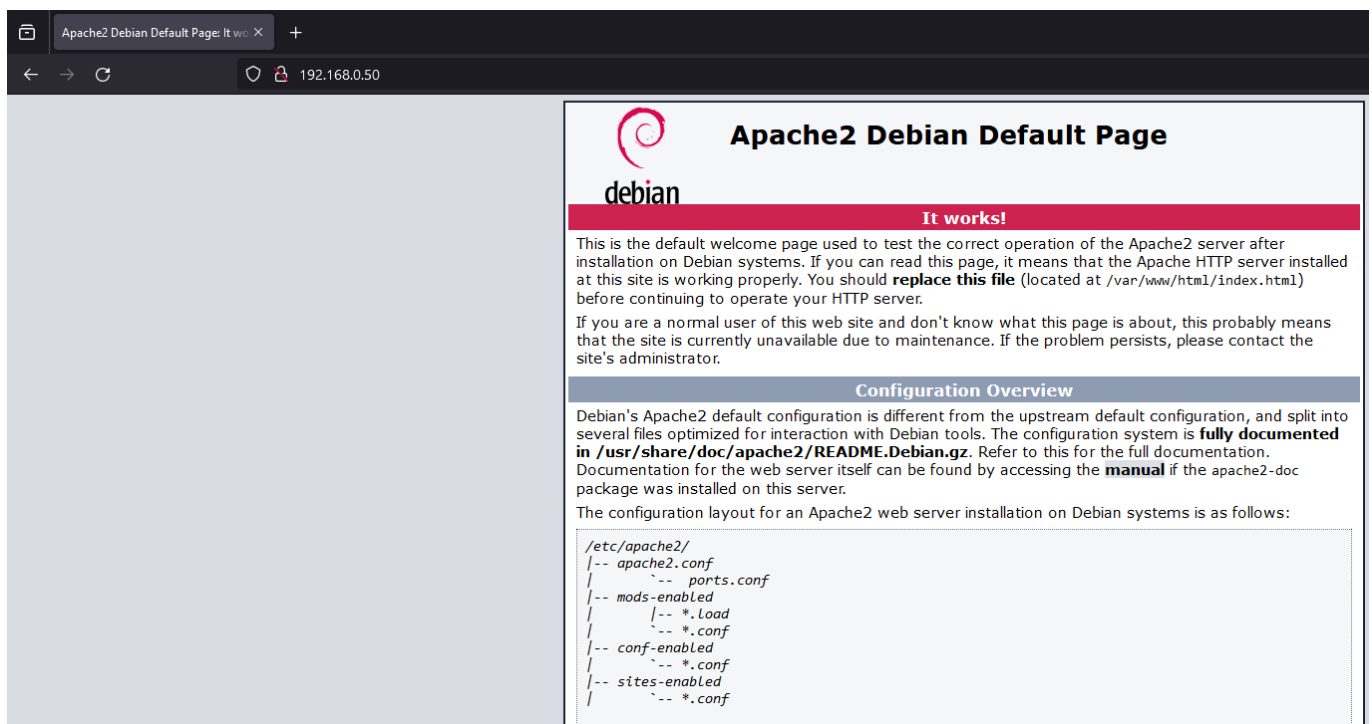
2. Comprobar que el servicio esta activo con el comando

```
sudo systemctl status apache2
```

3. Añadir una regla al firewall para permitir el tráfico HTTP con el comando

```
sudo ufw allow http
sudo ufw allow 80/tcp
sudo ufw reload
```

4. Acceder a la ip de la máquina virtual desde el navegador para comprobar que el servidor esta activo



PHP

1. Instalar el lenguaje de programación PHP con el comando

```
sudo apt install -y php libapache2-mod-php
```

2. Comprobar que php se ha instalado correctamente

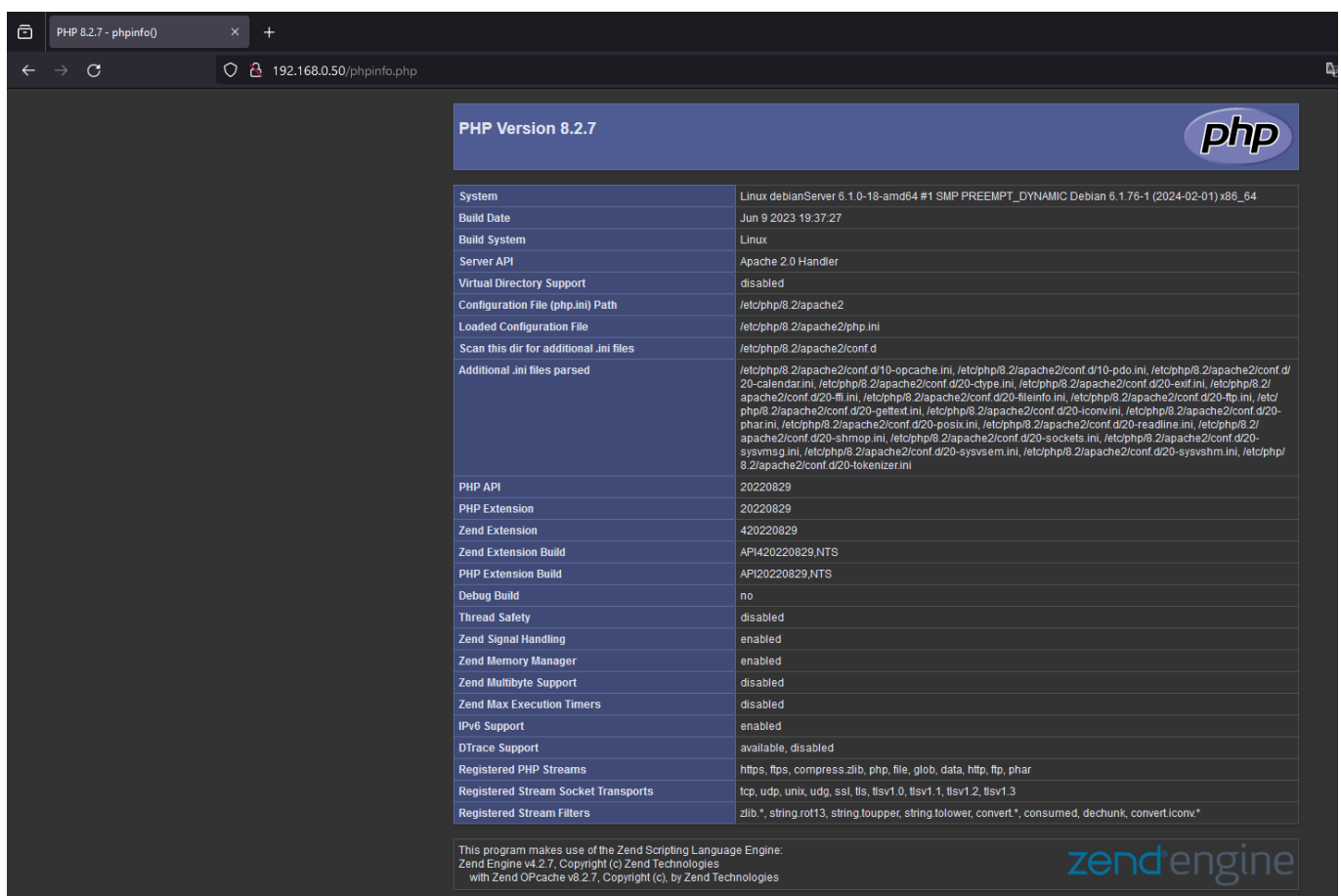
- Crear un archivo `phpinfo.php` en el directorio `/var/www/html` con el comando

```
sudo echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/phpinfo.php
```

3. Habilitar el módulo de Apache con el comando

```
sudo a2enmod php
```

- Acceder a la ip de la máquina virtual desde el navegador y añadir `/phpinfo.php` para comprobar que php esta activo



The screenshot shows a web browser window with the address bar displaying `192.168.0.50/phpinfo.php`. The page title is "PHP 8.2.7 - phpinfo()". The main content area displays the "PHP Version 8.2.7" header and a table of configuration information.

| PHP Version 8.2.7 | |
|---|---|
| System | Linux debianServer 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64 |
| Build Date | Jun 9 2023 19:37:27 |
| Build System | Linux |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.2/apache2 |
| Loaded Configuration File | /etc/php/8.2/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.2/apache2/conf.d |
| Additional .ini files parsed | /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysmsg.ini, /etc/php/8.2/apache2/conf.d/20-syssem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini |
| PHP API | 20220829 |
| PHP Extension | 20220829 |
| Zend Extension | 420220829 |
| Zend Extension Build | API420220829.NTS |
| PHP Extension Build | API20220829.NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | enabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | disabled |
| Zend Max Execution Timers | disabled |
| IPv6 Support | enabled |
| DTrace Support | available, disabled |
| Registered PHP Streams | https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3 |
| Registered Stream Filters | zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.* |

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.2.7, Copyright (c) Zend Technologies
with Zend OPcache v8.2.7, Copyright (c), by Zend Technologies

MySQL

1. Instalar el gestor de base de datos MySQL con el comando

```
sudo apt install -y mariadb-server  
sudo mysql_secure_installation
```

2. Acceder a MySql con el comando

```
sudo mysql
```

3. Configurar el usuario para la base de datos

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost' WITH GRANT OPTION;  
exit
```

4. Acceder a MySql con el usuario creado

```
mysql -u username -p
```

5. Crear una base de datos para la aplicación

```
CREATE DATABASE app1;
```

6. Crear una tabla para la aplicación

```
USE app1;  
CREATE TABLE tareas (id INT AUTO_INCREMENT PRIMARY KEY, tarea VARCHAR(100), fecha  
DATE);  
INSERT INTO tareas (tarea, fecha) VALUES ('tarea1', '2022-12-31');  
INSERT INTO tareas (tarea, fecha) VALUES ('tarea2', '2022-12-31');  
INSERT INTO tareas (tarea, fecha) VALUES ('tarea3', '2022-12-31');
```

7. Reiniciar el servidor Apache con el comando

```
sudo systemctl restart apache2
```

PhpMyAdmin

1. Instalar el gestor de base de datos PhpMyAdmin con el comando

```
sudo apt install -y phpmyadmin
```

- Durante la instalación seleccionar **apache2** como servidor
- Seleccionar **yes** para configurar la base de datos
- Introducir la contraseña para la base de datos

2. Crear un enlace para poder acceder a phpmyadmin desde el servidor web

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

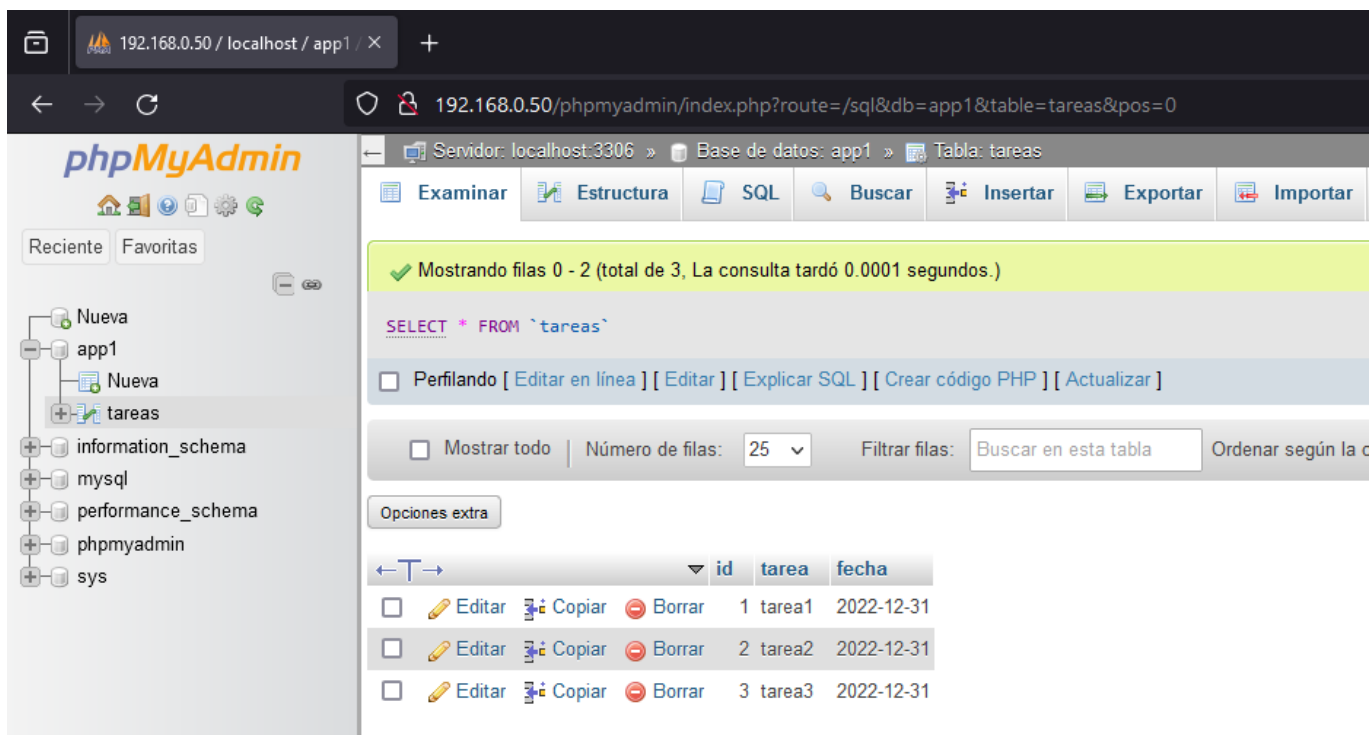
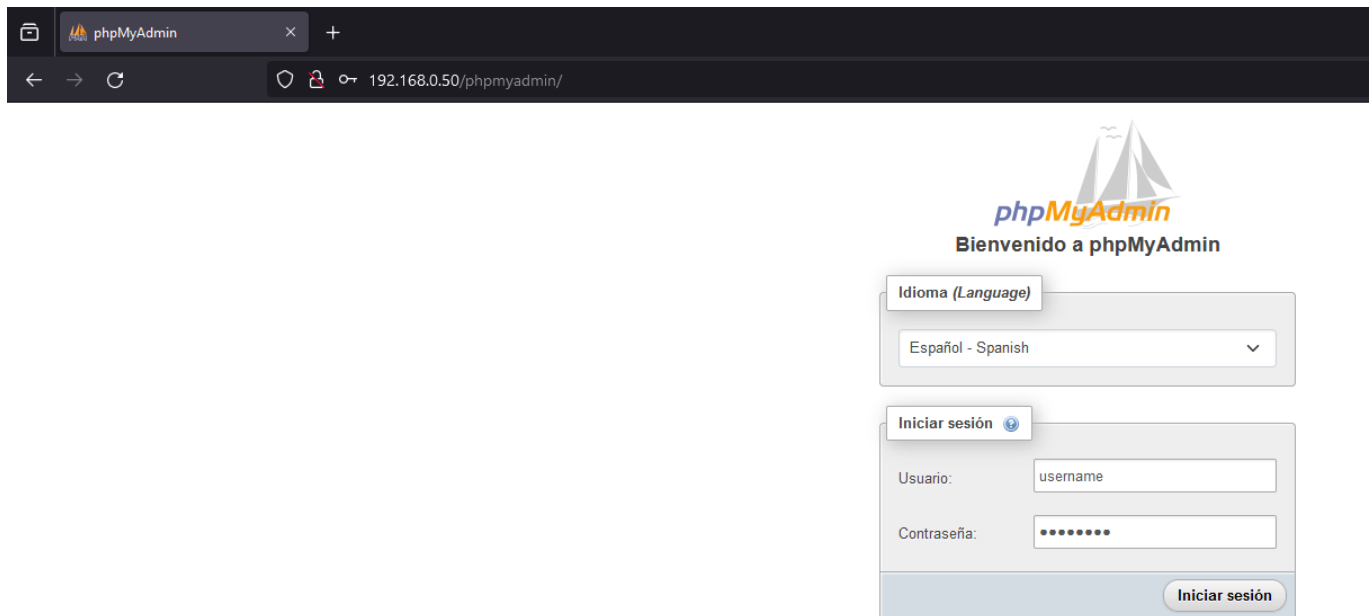
3. Habilitar el módulo de Apache para PHP con el comando

```
sudo phpenmod mbstring  
sudo phpenmod mysqli
```

4. Reiniciar apache para aplicar los cambios

```
sudo service apache2 restart
```

5. Acceder a la ip de la máquina virtual desde el navegador y añadir **/phpmyadmin** para comprobar que phpmyadmin esta activo



Ya tenemos instalado el servidor ftp, ahora vamos a configurar el acceso a la aplicación.

1. Dar permisos de escritura al directorio `/var/www/html` con el comando

```
sudo chmod 775 /var/www/html
sudo chown -R username /var/www/html
```

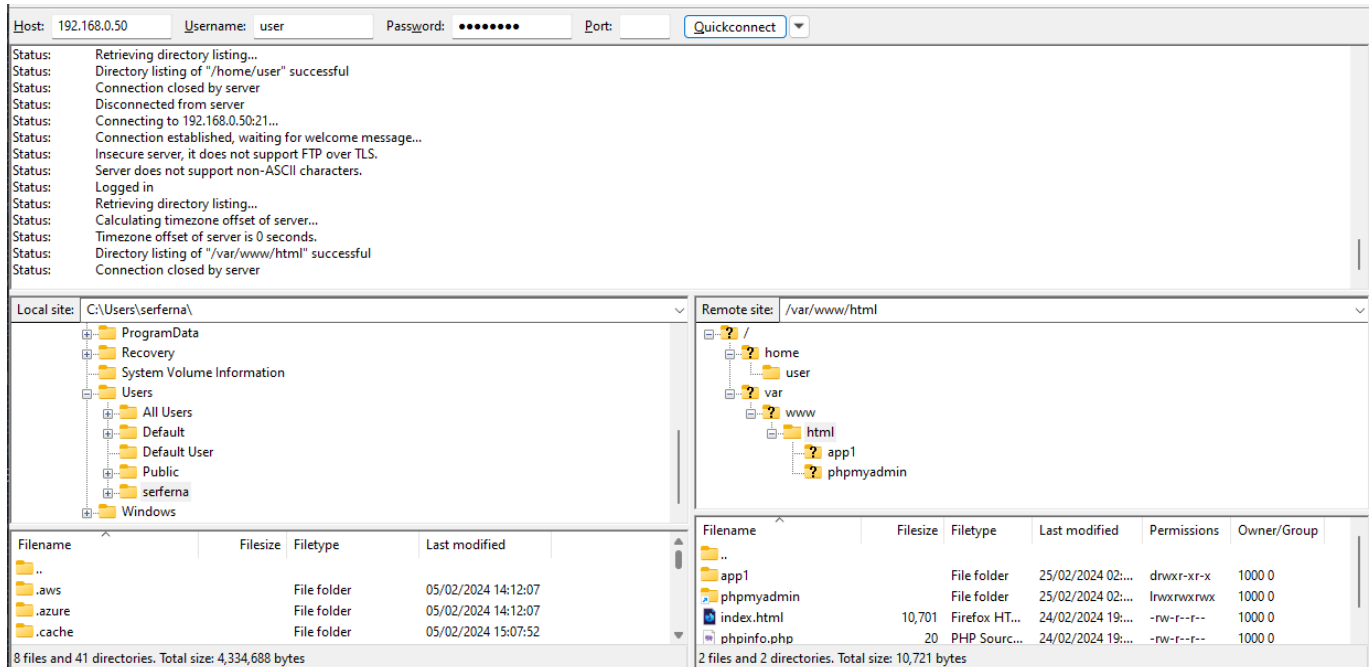
2. Asegurarse que en el archivo `/etc/vsftpd.conf` tiene las siguientes configuraciones

```
local_umask=022
write_enable=YES
```

```
local_root=/var/www/html
chroot_local_user=YES
```

- En filezilla si no se puede establecer la conexión, en ajustes -> ftp -> modo de transferencia -> activo

3. Conectar por ftp desde el equipo local con filezilla



Crear aplicación

1. Crear el directorio de la aplicación en el directorio `/var/www/html` con el comando

```
sudo mkdir /var/www/html/app1
```

Opción 1 - Editar archivos desde el servidor

2. Crear el archivo `index.php` en el directorio `/var/www/html/app1` con el comando

```
sudo echo "<?php echo 'hola'; ?>" | sudo tee /var/www/html/app1/index.php
```

3. Acceder a la ip de la máquina virtual desde el navegador y añadir `/app1` para comprobar que la aplicación esta activa
4. Modificar el archivo `index.php` para que se conecte a la base de datos

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
```

```

$dbname = "app1";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, tarea, fecha FROM tareas";
$result = $conn->query($sql);

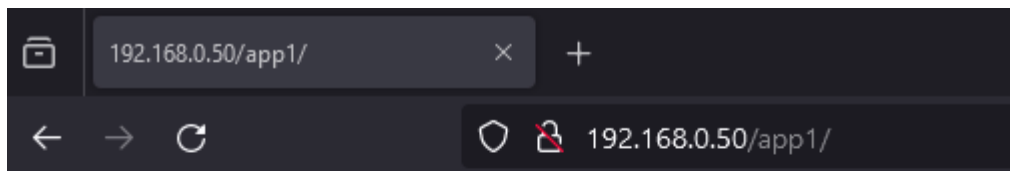
if ($result->num_rows > 0) {
    echo "<table><tr><th>ID</th><th>Tarea</th><th>Fecha</th></tr>";
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["id"]. "</td><td>" . $row["tarea"]. "</td><td>" .
$row["fecha"]. "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}

$conn->close();

?>

```

5. Acceder a la ip de la máquina virtual desde el navegador y añadir `/app1` para comprobar que la aplicación esta activa



| ID | Tarea | Fecha |
|----|--------|------------|
| 1 | tarea1 | 2022-12-31 |
| 2 | tarea2 | 2022-12-31 |
| 3 | tarea3 | 2022-12-31 |

Opción 2 - Subir archivos desde el equipo local usando ftp

1. Conectar por ftp desde el equipo local con filezilla

- Host: ip de la máquina virtual
- Usuario: username
- Contraseña: password

- Puerto: 21
2. Subir el archivo `index.php` al directorio `/var/www/html/app1`
 3. Acceder a la ip de la máquina virtual desde el navegador y añadir `/app1` para comprobar que la aplicación esta activa

Aplicación 2

Tecnologías

- Lenguaje de front
 - HTML5
 - CSS3
 - Javascript
- Lenguaje de back
 - Java
- BBDD
 - MySql
- Servidor de aplicaciones
 - Tomcat

Instalación

Servidor Tomcat

1. Instalar el servidor Tomcat con el comando

- `tomcat10` es la versión de Tomcat que se instala
- `tomcat10-admin` es la interfaz web para administrar Tomcat
- `tomcat10-common` es la documentación y los archivos comunes
- `tomcat10-user` es el usuario que ejecuta Tomcat
- `tomcat10-docs` es la documentación
- `tomcat10-examples` son ejemplos de aplicaciones

```
sudo apt install -y tomcat10 tomcat10-admin tomcat10-common tomcat10-user  
tomcat10-docs tomcat10-examples
```

2. Comprobar que el servicio esta activo con el comando

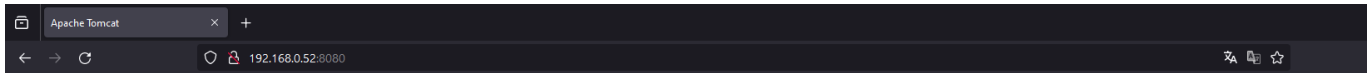
```
sudo systemctl status tomcat10
```

3. Añadir una regla al firewall para permitir el tráfico HTTP con el comando. El puerto por defecto de Tomcat es el 8080

```
sudo ufw allow 8080/tcp

sudo ufw reload
```

4. Acceder a la ip de la máquina virtual desde el navegador y añadir :8080 para comprobar que el servidor esta activo



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat10/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat10` and `CATALINA_BASE` in `/var/lib/tomcat10`, following the rules from `/usr/share/doc/tomcat10-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat10-docs: This package installs a web application that allows to browse the Tomcat 10 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat10-examples: This package installs a web application that allows to access the Tomcat 10 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat10-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat10/tomcat-users.xml`.

MySQL

1. Instalar el gestor de base de datos MySQL con el comando

```
sudo apt install -y mariadb-server
sudo mysql_secure_installation
```

2. Acceder a MySQL con el comando

```
sudo mysql
```

3. Configurar el usuario para la base de datos

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost' WITH GRANT OPTION;
exit
```

4. Acceder a MySQL con el usuario creado

```
mysql -u username -p
```

5. Crear una base de datos para la aplicación

```
CREATE DATABASE app2;
```

6. Crear una tabla para la aplicación

```
USE app2;  
CREATE TABLE tareas (id INT AUTO_INCREMENT PRIMARY KEY, tarea VARCHAR(100), fecha  
DATE);  
INSERT INTO tareas (tarea, fecha) VALUES ('tarea1', '2022-12-31');  
INSERT INTO tareas (tarea, fecha) VALUES ('tarea2', '2022-12-31');  
INSERT INTO tareas (tarea, fecha) VALUES ('tarea3', '2022-12-31');
```

Configuración ftp para subir archivos a la aplicación

Ya tenemos instalado el servidor ftp, ahora vamos a configurar el acceso a la aplicación.

1. Dar permisos de escritura al directorio `/var/lib/tomcat10/webapps` con el comando

```
sudo chmod 777 /var/lib/tomcat10/webapps  
sudo chown -R username /var/lib/tomcat10/webapps
```

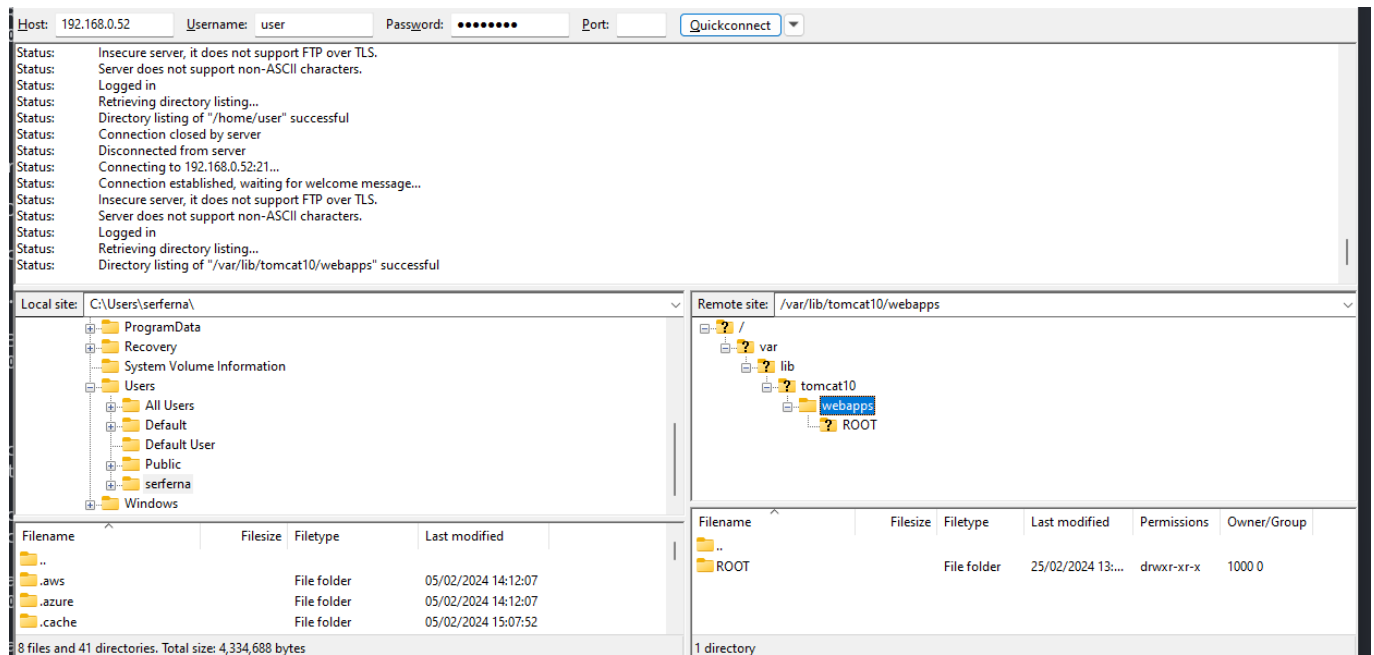
2. Asegurarse que en el archivo `/etc/vsftpd.conf` tiene la siguiente configuración

```
local_umask=022  
write_enable=YES  
local_root=/var/lib/tomcat10/webapps  
chroot_local_user=YES
```

3. Reiniciar el servidor ftp con el comando

```
sudo systemctl restart vsftpd
```

4. Conectar por ftp desde el equipo local con filezilla



Crear un usuario para la interfaz de administración de Tomcat

1. Añadir un usuario al archivo `/etc/tomcat10/tomcat-users.xml` con el comando

```
sudo vim /etc/tomcat10/tomcat-users.xml
```


Descomentar la sección que contiene el usuario y la contraseña


```
<user username="admin" password="<must-be-changed>" roles="manager-gui"/>
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
```

2. Reiniciar el servidor Tomcat con el comando

```
sudo systemctl restart tomcat10
```

3. Acceder a la ip de la máquina virtual desde el navegador y añadir `:8080/manager/html` para comprobar que la interfaz de administración de Tomcat esta activa





Gestor de Aplicaciones Web de Tomcat

Mensaje:

Gestor

Listar Aplicaciones

Ayuda HTML de Gestor

Ayuda de Gestor

Estado de Servi

| Aplicaciones | | | | | |
|---------------|----------------------|---------------------------------|--------------|----------|---|
| Ruta | Versión | Nombre a Mostrar | Ejecutándose | Sesiones | Comandos |
| / | Ninguno especificado | | true | 0 | <div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div> |
| /docs | Ninguno especificado | Tomcat Documentation | true | 0 | <div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div> |
| /examples | Ninguno especificado | Servlet and JSP Examples | true | 0 | <div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div> |
| /host-manager | Ninguno especificado | Tomcat Host Manager Application | true | 0 | <div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div> |
| /manager | Ninguno especificado | Tomcat Manager Application | true | 1 | <div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div> |

Crear aplicación

1. Crear el directorio de la aplicación en el directorio `/var/lib/tomcat10/webapps` con el comando

```
sudo mkdir /var/lib/tomcat10/webapps/app2
```

Opción 1 - Editar archivos desde el servidor

2. Crear el archivo `index.jsp` en el directorio `/var/lib/tomcat10/webapps/app2` y añadir el código

```
<!DOCTYPE html>
<html>
<head>
  <title>Sumar dos Números</title>
</head>
<body>
  <h2>Sumar Números</h2>
  <form action="calculate.jsp" method="post">
    Número 1: <input type="text" name="num1"><br>
    Número 2: <input type="text" name="num2"><br>
    <input type="submit" value="Calcular Suma">
  </form>
</body>
</html>
```

3. Crear el archivo `calculate.jsp` en el directorio `/var/lib/tomcat10/webapps/app2` y añadir el código

```
<%@ page import="java.io.*,java.util.*" %>
<%@ page import="javax.servlet.*,javax.servlet.http.*" %>
```



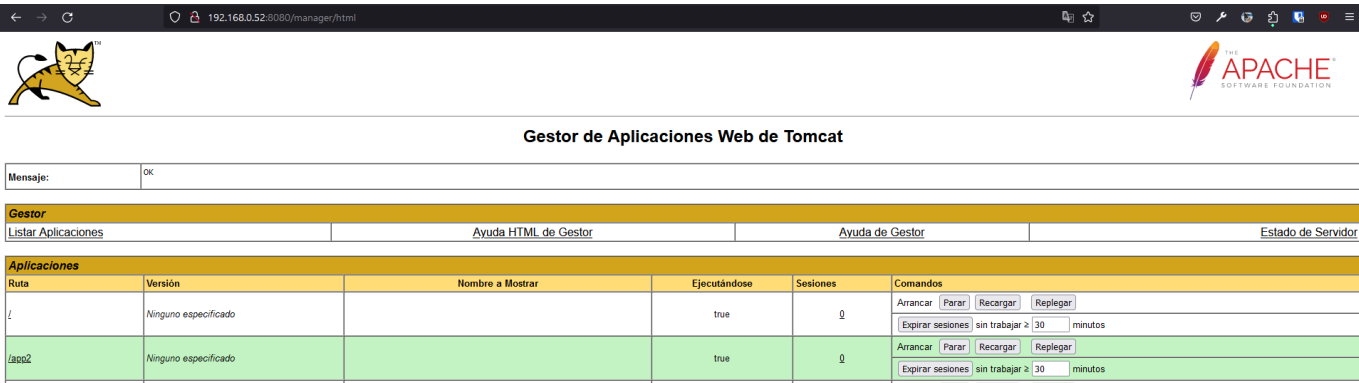
```
<%
    // Se obtienen los parametros del formulario
    String num1Str = request.getParameter("num1");
    String num2Str = request.getParameter("num2");

    // Se convierten los parametros a enteros
    int num1 = Integer.parseInt(num1Str);
    int num2 = Integer.parseInt(num2Str);

    // Se realiza la suma
    int sum = num1 + num2;
%>

<!DOCTYPE html>
<html>
<head>
    <title>Sumar dos Numeros</title>
</head>
<body>
    <h2>Sumar dos Numeros</h2>
    <p>Numero 1: <%= num1 %></p>
    <p>Numero 2: <%= num2 %></p>
    <p>Suma: <%= sum %></p>
</body>
</html>
```

4. Acceder a la ip de la máquina virtual desde el navegador y ir al panel de administración de Tomcat para comprobar que la aplicación esta activa



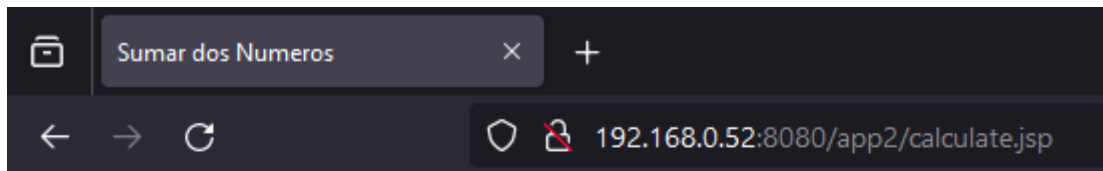
5. Acceder a la ip de la máquina virtual desde el navegador y añadir :8080/app2 para ver la aplicación



Sum Calculator

Enter number 1:

Enter number 2:



Sumar dos Numeros

Numero 1: 3

Numero 2: 2

Suma: 5

Opción 2 - Subir archivos desde el equipo local usando ftp

1. Conectar por ftp desde el equipo local con filezilla
 - Host: ip de la máquina virtual
 - Usuario: username
 - Contraseña: password
 - Puerto: 21
2. Subir el archivo `index.jsp` y `calculate.jsp` al directorio `/var/lib/tomcat10/webapps/app2`
3. Acceder a la ip de la máquina virtual desde el navegador y añadir `/app2` para comprobar que la aplicación esta activa
4. Reiniciar el servidor Tomcat con el comando

```
sudo systemctl restart tomcat10
```

Conclusiones

De forma base se ha configurado un servidor Debian 12 con un firewall, un servidor SSH y un servidor FTP.

Sobre este sistema se han montado dos aplicaciones web.

La primera aplicación se ha montado sobre un servidor Apache con PHP y MySQL.

- Se ha configurado el servidor Apache y se ha comprobado que esta activo
- Se ha configurado PHP y se ha comprobado que esta activo
- Se ha configurado MySQL y se ha comprobado que esta activo
- Se ha instalado PhpMyAdmin y se ha comprobado que esta activo
- Se ha configurado el servidor FTP y se ha configurado para dar acceso al directorio de la aplicación
- Se ha creado una aplicación web y se ha comprobado que esta activa
- Se ha comprobado que se puede subir archivos a la aplicación
- Se ha comprobado que se puede acceder a la base de datos desde la aplicación
- Se ha comprobado que se puede acceder a la base de datos desde PhpMyAdmin
- Se ha comprobado que se puede acceder a la aplicación desde el equipo local

La segunda aplicación se ha montado sobre un servidor Tomcat con MySQL.

- Se ha configurado el servidor Tomcat y se ha comprobado que esta activo
- Se ha configurado MySQL y se ha comprobado que esta activo
- Se ha configurado el servidor FTP y se ha configurado para dar acceso al directorio de la aplicación
- Se ha creado una aplicación web y se ha comprobado que esta activa
- Se ha comprobado que se puede subir archivos a la aplicación
- Se ha comprobado que se puede acceder a la aplicación desde el panel de administración de Tomcat
- Se ha comprobado que se puede acceder a la aplicación desde el equipo local