

Práctica 2.2 Despliegue de una Aplicación Python sobre la Pila LAMP.

1. Objetivo de la práctica.

El objetivo de esta práctica es que los alumnos aprendan a desplegar una aplicación **Python** básica sobre una pila **LAMP** (Linux, Apache, MariaDB, Python). La aplicación se conectará a una base de datos **MariaDB** para realizar **consultas** sencillas y presentará los resultados a través de una interfaz **web**.

2. Requisitos previos.

- **Apache** instalado y en ejecución en una **máquina virtual** con **Debian 12**.
- **MariaDB** instalado y funcionando.
- Acceso a la **máquina virtual** para realizar configuraciones y editar archivos.

3. Descripción de la práctica.

Se les solicita a los alumnos que realicen las siguientes tareas para desplegar una aplicación **Python** en una pila **LAMP**. La aplicación interactuará con una **base de datos MariaDB** y se servirá a través de **Apache** utilizando el módulo **mod_wsgi**. El objetivo final es que los alumnos puedan acceder a la aplicación desde el **navegador** y ver la lista de alumnos almacenados en la **base de datos**.

4. Tareas a realizar.

Tarea 1: Configuración del entorno.

- **Instalar Python** y las dependencias necesarias:
 1. Actualizar la lista de paquetes e instalar **Python**:

```
sudo apt update
```

```
jferher_usuario@jferher:~$ sudo apt update
Obj:1 http://deb.debian.org/debian bookworm InRelease
Obj:2 http://deb.debian.org/debian bookworm-updates InRelease
Obj:3 http://security.debian.org/debian-security bookworm-security InRelease
Des:4 http://repo.mysql.com/apt/debian bullseye InRelease [22,7 kB]
Err:4 http://repo.mysql.com/apt/debian bullseye InRelease
```

```
sudo apt install python3-pip
```

```
jferher_usuario@jferher:~$ sudo apt install python3-pip
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential
  dpkg-dev fakeroot g++ g++-12 gcc gcc-12 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libbinutils
  libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
  libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
  libgcc-12-dev libgprofng0 libitm1 libjs-jquery libjs-sphinxdoc
  libjs-underscore liblsan0 libnsl-dev libpython3-dev libpython3.11-dev
  libstdc++-12-dev libtirpc-dev libtsan2 libubsan1 linux-libc-dev make
  manpages-dev python3-dev python3-distutils python3-lib2to3
  python3-setuptools python3-wheel python3.11-dev rpcsvc-proto zlib1g-dev
Paquetes sugeridos:
  binutils-doc debian-keyring g++-multilib g++-12-multilib gcc-12-doc
  gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-12-multilib gcc-12-locales glibc-doc bzip2 libstdc++-12-doc make-doc
  python-setuptools-doc
Se instalarán los siguientes paquetes NUEVOS:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential
  dpkg-dev fakeroot g++ g++-12 gcc gcc-12 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libbinutils
  libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
  libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
  libgcc-12-dev libgprofng0 libitm1 libjs-jquery libjs-sphinxdoc
  libjs-underscore liblsan0 libnsl-dev libpython3-dev libpython3.11-dev
  libstdc++-12-dev libtirpc-dev libtsan2 libubsan1 linux-libc-dev make
```

2. Crea un Entorno Virtual para trabajar con Python:

[Entornos Virtuales en Python](#)

Para esta práctica vamos a crear un **entorno virtual** en la ruta **/var/www/html/flaskapp** y como nombre del **entorno virtual** utilizaremos **flaskapp**.

```
jferher_usuario@jferher:/var/www/html/flaskapp$ sudo apt install python3.11-venv
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  python3-pip-whl python3-setuptools-whl
Se instalarán los siguientes paquetes NUEVOS:
  python3-pip-whl python3-setuptools-whl python3.11-venv
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 2.835 kB de archivos.
Se utilizarán 3.164 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://deb.debian.org/debian bookworm/main amd64 python3-pip-whl all 23.0.1+dfsg-1 [1.717 kB]
Des:2 http://security.debian.org/debian-security bookworm-security/main amd64 python3.11-venv amd64 3.11.2-6+deb12u3 [5.896 B]
Des:3 http://deb.debian.org/debian bookworm/main amd64 python3-setuptools-whl all 66.1.1-1 [1.111 kB]
Descargados 2.835 kB en 0s (6.643 kB/s)
Seleccionando el paquete python3-pip-whl previamente no seleccionado.
(Leyendo la base de datos ... 140962 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../python3-pip-whl_23.0.1+dfsg-1_all.deb ...
Desempaquetando python3-pip-whl (23.0.1+dfsg-1) ...
Seleccionando el paquete python3-setuptools-whl previamente no seleccionado.
Preparando para desempaquetar .../python3-setuptools-whl_66.1.1-1_all.deb ..
.
Desempaquetando python3-setuptools-whl (66.1.1-1) ...
Seleccionando el paquete python3.11-venv previamente no seleccionado.
Preparando para desempaquetar .../python3.11-venv_3.11.2-6+deb12u3_amd64.deb
...
Desempaquetando python3.11-venv (3.11.2-6+deb12u3) ...
```

```
jferher_usuario@jferher:/var/www/html/flaskapp$ sudo python3 -m venv --prompt flaskapp .venv
jferher_usuario@jferher:/var/www/html/flaskapp$ sudo source .venv/bin/activate
sudo: source: command not found
jferher_usuario@jferher:/var/www/html/flaskapp$ source .venv/bin/activate
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

3. Instalar el conector PyMySQL de MariaDB para Python:

```
pip3 install pymysql
```

Para evitar **errores** al instalar el paquete en el **entorno virtual** tienes que asegurarte que tu **usuario** tenga los **permisos** necesarios aplicando el siguiente comando:

```
sudo chown -R $USER:$USER /var/www/html/flaskapp/.venv
```

```
fguardia@debian12:/$ sudo mkdir /var/www/html/flaskapp
fguardia@debian12:/$ cd /var/www/html/flaskapp
fguardia@debian12:/var/www/html/flaskapp$ ls -la
total 8
drwxr-xr-x 2 root    root    4096 oct  7 09:58 .
drwxr-xr-x 6 www-data www-data 4096 oct  7 09:58 ..
fguardia@debian12:/var/www/html/flaskapp$ sudo python3 -m venv --prompt flaskapp .venv
fguardia@debian12:/var/www/html/flaskapp$ ls -la
total 12
drwxr-xr-x 3 root    root    4096 oct  7 09:58 .
drwxr-xr-x 6 www-data www-data 4096 oct  7 09:58 ..
drwxr-xr-x 5 root    root    4096 oct  7 09:58 .venv
fguardia@debian12:/var/www/html/flaskapp$ source .venv/bin/activate
(flaskapp) fguardia@debian12:/var/www/html/flaskapp$ deactivate
fguardia@debian12:/var/www/html/flaskapp$ sudo chown -R $USER:$USER /var/www/html/flaskapp/.venv
fguardia@debian12:/var/www/html/flaskapp$ ls -la
total 12
drwxr-xr-x 3 root    root    4096 oct  7 09:58 .
drwxr-xr-x 6 www-data www-data 4096 oct  7 09:58 ..
drwxr-xr-x 5 fguardia fguardia 4096 oct  7 09:58 .venv
fguardia@debian12:/var/www/html/flaskapp$ source .venv/bin/activate
(flaskapp) fguardia@debian12:/var/www/html/flaskapp$ pip3 install mysql-connector-python
Collecting mysql-connector-python
  Using cached mysql_connector_python-9.0.0-cp311-cp311-manylinux_2_17_x86_64.whl (19.3 MB)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.0.0
(flaskapp) fguardia@debian12:/var/www/html/flaskapp$ █
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo chown -R $USER:$USER /var/www/html/flaskapp/.venv
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ pip3 install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

- **Configurar Apache para servir aplicaciones Python:**
 1. Instalar el módulo **mod_wsgi**:

```
sudo apt install libapache2-mod-wsgi-py3
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo apt install libapache2-mod-wsgi-py3
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-wsgi-py3
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 97,3 kB de archivos.
Se utilizarán 296 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 libapache2-mod-wsgi-py3 amd64 4.9.4-1+b2 [97,3 kB]
Descargados 97,3 kB en 0s (383 kB/s)
Seleccionando el paquete libapache2-mod-wsgi-py3 previamente no seleccionado.
(Leyendo la base de datos ... 140980 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ... /libapache2-mod-wsgi-py3_4.9.4-1+b2_amd64.deb ...
Desempaquetando libapache2-mod-wsgi-py3 (4.9.4-1+b2) ...
Configurando libapache2-mod-wsgi-py3 (4.9.4-1+b2) ...
apache2_invoke: Enable module wsgi
```

2. Reiniciar Apache:

```
sudo systemctl restart apache2
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo systemctl restart apache2
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$
```

Entrega:

- Capturas de pantalla mostrando la instalación correcta de **Python**, el **conector MariaDB** y la instalación de **mod_wsgi**.

Tarea 2: Creación de la base de datos.

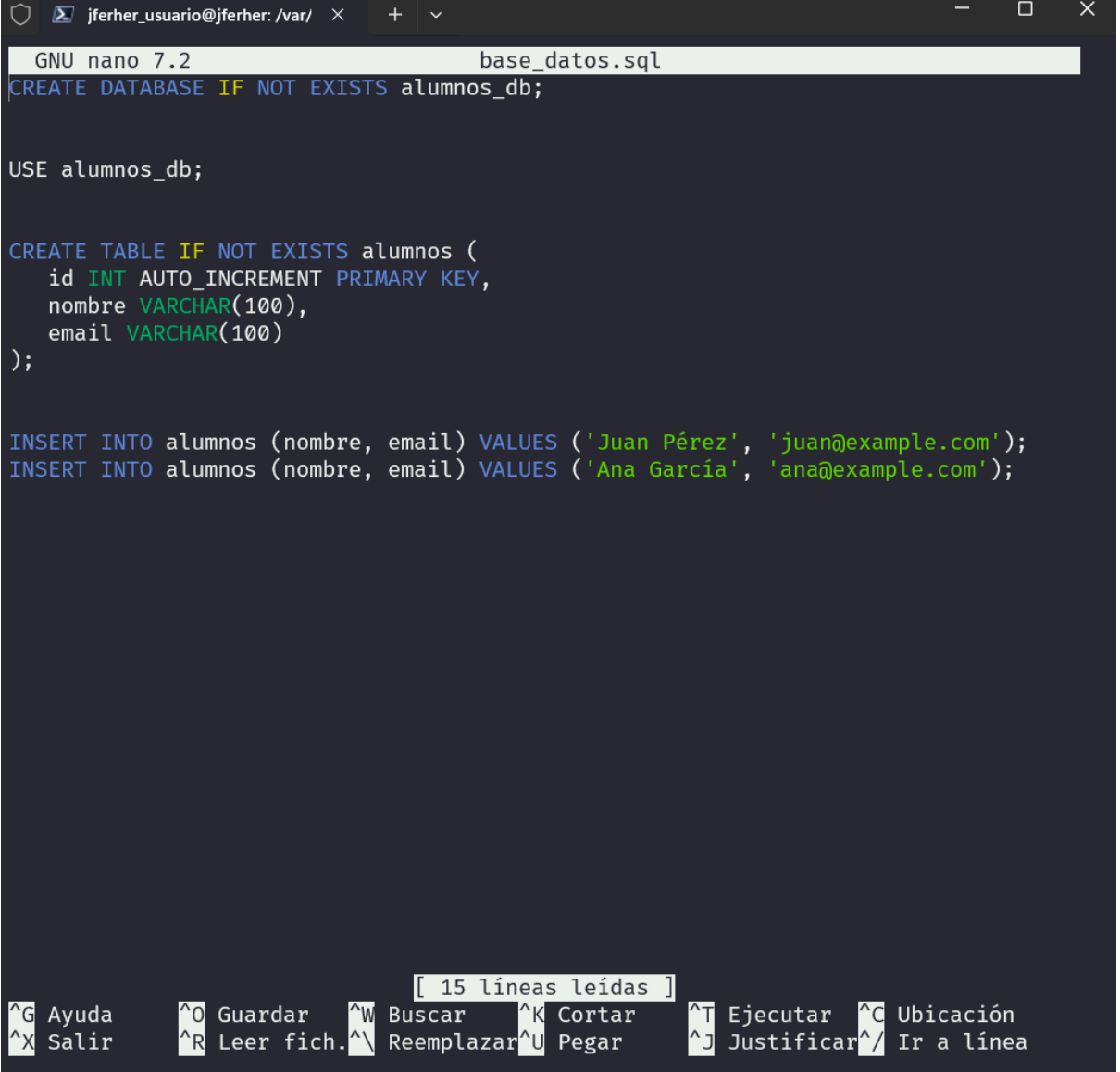
1. Crear un archivo **base_datos.sql** con el siguiente contenido, que define la **estructura** de la **base de datos** y añade algunos **registros**:

```
CREATE DATABASE IF NOT EXISTS alumnos_db;

USE alumnos_db;

CREATE TABLE IF NOT EXISTS alumnos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    email VARCHAR(100)
);

INSERT INTO alumnos (nombre, email) VALUES ('Juan Pérez',
'juan@example.com');
INSERT INTO alumnos (nombre, email) VALUES ('Ana García',
'ana@example.com');
```



```
GNU nano 7.2 base_datos.sql
CREATE DATABASE IF NOT EXISTS alumnos_db;

USE alumnos_db;

CREATE TABLE IF NOT EXISTS alumnos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    email VARCHAR(100)
);

INSERT INTO alumnos (nombre, email) VALUES ('Juan Pérez', 'juan@example.com');
INSERT INTO alumnos (nombre, email) VALUES ('Ana García', 'ana@example.com');
```

[15 líneas leídas]

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^_\ Reemplazar	^U Pegar	^J Justificar	^/ Ir a línea

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo nano base_datos.sql
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

2. Ejecutar el archivo **.sql** en **MariaDB** para crear la **base de datos** y la **tabla** de **alumnos**:

```
mysql -u root -p < base_datos.sql
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ mysql -u root -p < base_datos.sql
Enter password:
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

Entrega:

- Archivo **base_datos.sql**.
- Captura de pantalla del comando ejecutado para **importar la base de datos**.
- Captura de pantalla de la **base de datos** y **tabla** creadas en **MariaDB** con los **registros** añadidos.

Tarea 3: Desarrollo de la aplicación Python.

1. Crear un **directorio** para la aplicación en el servidor **Apache**:

```
sudo mkdir /var/www/html/flaskapp
```

2. Dentro del **directorio**, crear el archivo **app.py** con el siguiente código:

```
#import mysql.connector
import pymysql
```

```
from flask import Flask, render_template

app = Flask(__name__)

# Conexión a la base de datos
#conexion = mysql.connector.connect(
conexion = pymysql.connect(
    host="localhost",
    user="fguardia",
    password="iesabdera", # Añadir la contraseña si es necesario
    database="alumnos_db"
)

@app.route('/')
def index():
    cursor = conexion.cursor()
    cursor.execute("SELECT * FROM alumnos")
    alumnos = cursor.fetchall()
    return render_template('index.html', alumnos=alumnos)

if __name__ == '__main__':
```



```

GNU nano 7.2 app.py
#import mysql.connector
import pymysql

from flask import Flask, render_template

app = Flask(__name__)

# Conexión a la base de datos
#conexion = mysql.connector.connect(
conexion = pymysql.connect(
    host="localhost",
    user="jferher_usuario",
    password="Juan_fher_439", # Añadir la contraseña si es necesario
    database="alumnos_db"
)

@app.route('/')
def index():
    cursor = conexion.cursor()
    cursor.execute("SELECT * FROM alumnos")
    alumnos = cursor.fetchall()
    return render_template('index.html', alumnos=alumnos)

if __name__ == '__main__':

```

[28 líneas leídas]

^G Ayuda	^O Guardar	^W Buscar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^_ Reemplazar	^K Cortar	^J Justificar
		^U Pegar		^/ Ir a línea

```

(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo nano app.py
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$

```

3. Crear la plantilla **HTML** que mostrará la **lista de alumnos** en un **navegador**. Dentro de la carpeta **/var/www/html/flaskapp/templates**, crear el archivo **index.html** con el siguiente contenido:

```

<!DOCTYPE html>
<html lang="es">

```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Lista de Alumnos</title>
</head>
<body>
  <h1>Alumnos registrados</h1>
  <ul>
    {% for alumno in alumnos %}
    <li>ID: {{ alumno[0] }}, Nombre: {{ alumno[1] }}, Email:
{{ alumno[2] }}</li>
    {% endfor %}
  </ul>
</body>
</html>
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo mkdir templates
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ cd templates
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp/templates$ |
```

```

GNU nano 7.2 index.html
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lista de Alumnos</title>
</head>
<body>
  <h1>Alumnos registrados</h1>
  <ul>
    {% for alumno in alumnos %}
    <li>ID: {{ alumno[0] }}, Nombre: {{ alumno[1] }}, Email: {{ alumno[2] }}</li>
    {% endfor %}
  </ul>
</body>
</html>

```

[16 líneas escritas]

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
 ^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^_ Ir a línea

```

(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp/templates$ sudo nano index.html
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp/templates$ |

```

Entrega:

- Código fuente del archivo **app.py**.
- Código fuente del archivo **index.html**.
- Captura de pantalla de la aplicación ejecutándose en el **navegador** con los datos extraídos de la **base de datos**.

Tarea 4: Configuración de Apache para servir la aplicación Python.

1. Dar los **permisos de lectura** necesarios al **usuario de Apache (www-data)** sobre el directorio de trabajo:

```
sudo chown -R www-data:www-data /var/www/html/flaskapp
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo chown -R www-data:www-data /var/www/html/flaskapp
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

```
sudo chmod -R 755 /var/www/html/flaskapp
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo chmod -R 755 /var/www/html/flaskapp
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

2. Crear un archivo **WSGI** en **/var/www/html/flaskapp/flaskapp.wsgi** con el siguiente contenido:

```
import sys
import site

# Ruta del entorno virtual
venv_path = '/var/www/html/flaskapp/.venv'

# Añadir los paquetes del entorno virtual a sys.path
site.addsitedir(f'{venv_path}/lib/python3.11/site-packages')

# Añadir el directorio del proyecto a sys.path
sys.path.insert(0, '/var/www/html/flaskapp')

# Configurar el intérprete de Python para usar el entorno virtual
sys.executable = f'{venv_path}/bin/python'

# Importar y arrancar la aplicación Flask
from app import app as application
```

```

GNU nano 7.2 flaskapp.wsgi
import sys
import site

# Ruta del entorno virtual
venv_path = '/var/www/html/flaskapp/.venv'

# Añadir los paquetes del entorno virtual a sys.path
site.addsitedir(f'{venv_path}/lib/python3.11/site-packages')

# Añadir el directorio del proyecto a sys.path
sys.path.insert(0, '/var/www/html/flaskapp')

# Configurar el intérprete de Python para usar el entorno virtual
sys.executable = f'{venv_path}/bin/python'

# Importar y arrancar la aplicación Flask
from app import app as application

```

[23 líneas leídas]

[^]G Ayuda [^]O Guardar [^]W Buscar [^]K Cortar [^]T Ejecutar [^]C Ubicación
[^]X Salir [^]R Leer fich. [^]\ Reemplazar [^]U Pegar [^]J Justificar [^]/ Ir a línea

Damos **permisos de ejecución** al fichero **flaskapp.wsgi**:

```
sudo chmod +x /var/www/html/flaskapp/flaskapp.wsgi
```

```

(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo nano flaskapp.wsgi
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo chmod +x /var/www/html/flaskapp/flaskapp.wsgi

```

3. Asegúrate de que el **entorno virtual** esté correctamente configurado:

Asegúrate de que has creado correctamente el entorno virtual en **/var/www/html/flaskapp/.venv** con **Python 3.11**:

```
python3.11 -m venv /var/www/html/flaskapp/.venv
```

```
(flaskapp) fguardia@debian12:/var/www/html/flaskapp$ python3.11 -m venv /var/www/html/flaskapp/.venv
```

```
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo python3.11 -m venv /var/www/html/flaskapp/.venv
(flaskapp) jferher_usuario@jferher:/var/www/html/flaskapp$ |
```

4. Instalar **Flask**:

Con el **entorno virtual** activado, puedes proceder a instalar Flask con el siguiente comando:

pip install Flask

```
(.venv) jferher_usuario@jferher:/var/www/html/flaskapp$ sudo su
root@jferher:/var/www/html/flaskapp# source .venv/bin/activate
(.venv) root@jferher:/var/www/html/flaskapp# pip install Flask
Collecting Flask
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
    101.7/101.7 kB 2.0 MB/s eta 0:00:00
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.4-py3-none-any.whl (227 kB)
    227.6/227.6 kB 6.7 MB/s eta 0:00:00
Collecting Jinja2>=3.1.2
  Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
    133.3/133.3 kB 23.5 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting click>=8.1.3
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
    97.9/97.9 kB 12.2 MB/s eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-3.0.1-cp311-cp311-manylinux2_17_x86_64.manylinux2014_x86_64.whl (23 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, Flask
Successfully installed Flask-3.0.3 Jinja2-3.1.4 MarkupSafe-3.0.1 Werkzeug-3.0.4 blinker-1.8.2 click-8.1.7 itsdangerous-2.2.0
(.venv) root@jferher:/var/www/html/flaskapp# |
```

no me dejaba hacerlo sin meterme en sudo su

Para asegurarte de que **Flask** se ha instalado correctamente, puedes ejecutar el siguiente comando dentro del **entorno virtual**:

python3 -m flask --version

```
(.venv) root@jferher:/var/www/html/flaskapp# python3 -m flask --version
Python 3.11.2
Flask 3.0.3
Werkzeug 3.0.4
(.venv) root@jferher:/var/www/html/flaskapp# |
```

5. Configurar **Apache** para servir la aplicación **Flask**. Crear un **archivo de configuración de Apache** en **/etc/apache2/sites-available/flaskapp.conf**:

```
<VirtualHost *:8000>
    ServerName flaskapp.local
    DocumentRoot /var/www/html/flaskapp

    WSGIDaemonProcess flaskapp python-home=/var/www/html/flaskapp/.venv python-path=/var/www/html/flaskapp

    WSGIScriptAlias / /var/www/html/flaskapp/flaskapp.wsgi

    <Directory /var/www/html/flaskapp>
        Require all granted
    </Directory>

    Alias /static /var/www/html/flaskapp/static
    <Directory /var/www/html/flaskapp/static/>
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
root@jferher:/home/jferher_usuario# cd /etc/apache2/sites-available
root@jferher:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf  web1.conf  web2.conf
root@jferher:/etc/apache2/sites-available# nano flaskapp.conf
```

```

GNU nano 7.2 flaskapp.conf *
<VirtualHost *:8000>
    ServerName flaskapp.local
    DocumentRoot /var/www/html/flaskapp
    WSGIDaemonProcess flaskapp python-home=/var/www/html/flaskapp/.venv python-pa
    WSGIScriptAlias / /var/www/html/flaskapp/flaskapp.wsgi

    <Directory /var/www/html/flaskapp>
        Require all granted
    </Directory>

    Alias /static /var/www/html/flaskapp/static
    <Directory /var/www/html/flaskapp/static/>
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

[^]G Ayuda [^]O Guardar [^]W Buscar [^]K Cortar [^]T Ejecutar [^]C Ubicación
[^]X Salir [^]R Leer fich. [^]\ Reemplazar [^]U Pegar [^]J Justificar [^]/ Ir a línea

6. Habilitar el nuevo **sitio** en **Apache**:

```
sudo a2ensite flaskapp
```

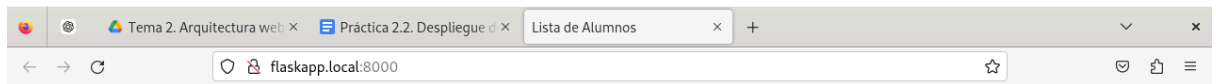
```
sudo systemctl restart apache2
```

```

root@jferher:/etc/apache2/sites-available# sudo a2ensite flaskapp
Enabling site flaskapp.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@jferher:/etc/apache2/sites-available# systemctl restart apache2
root@jferher:/etc/apache2/sites-available#

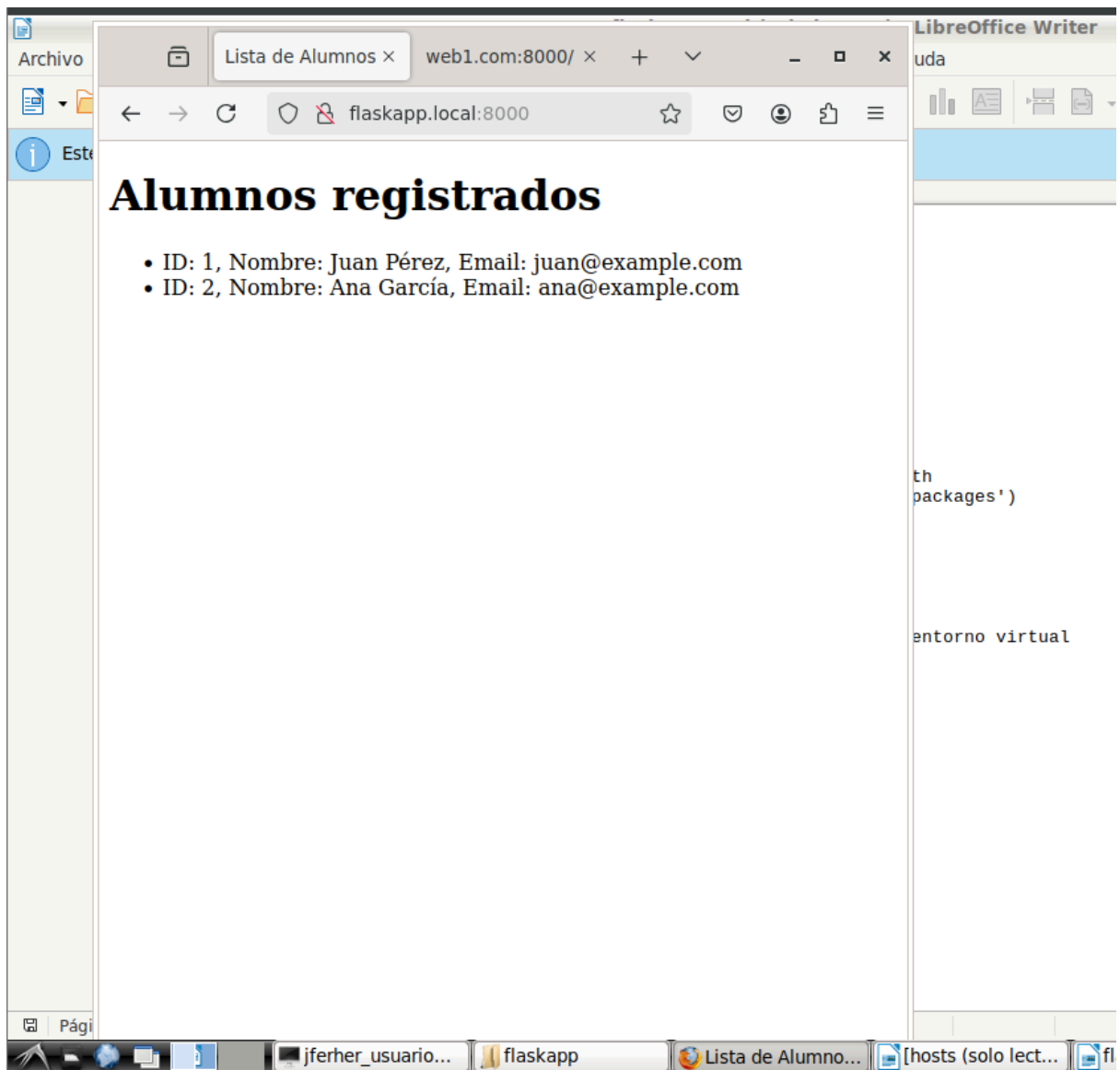
```

<http://flaskapp.local:8000>



Alumnos registrados

- ID: 1, Nombre: Juan Pérez, Email: juan@example.com
- ID: 2, Nombre: Ana García, Email: ana@example.com

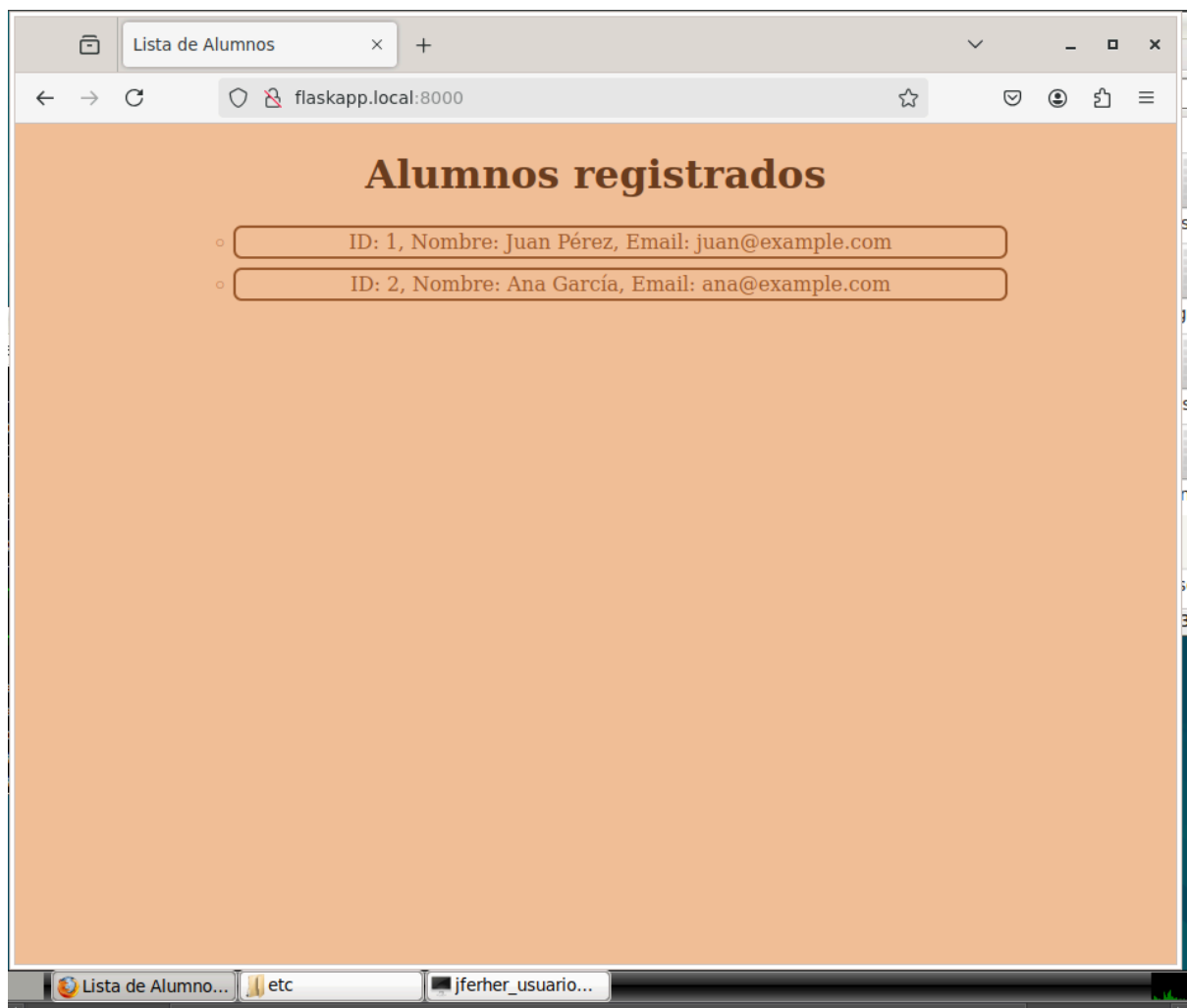


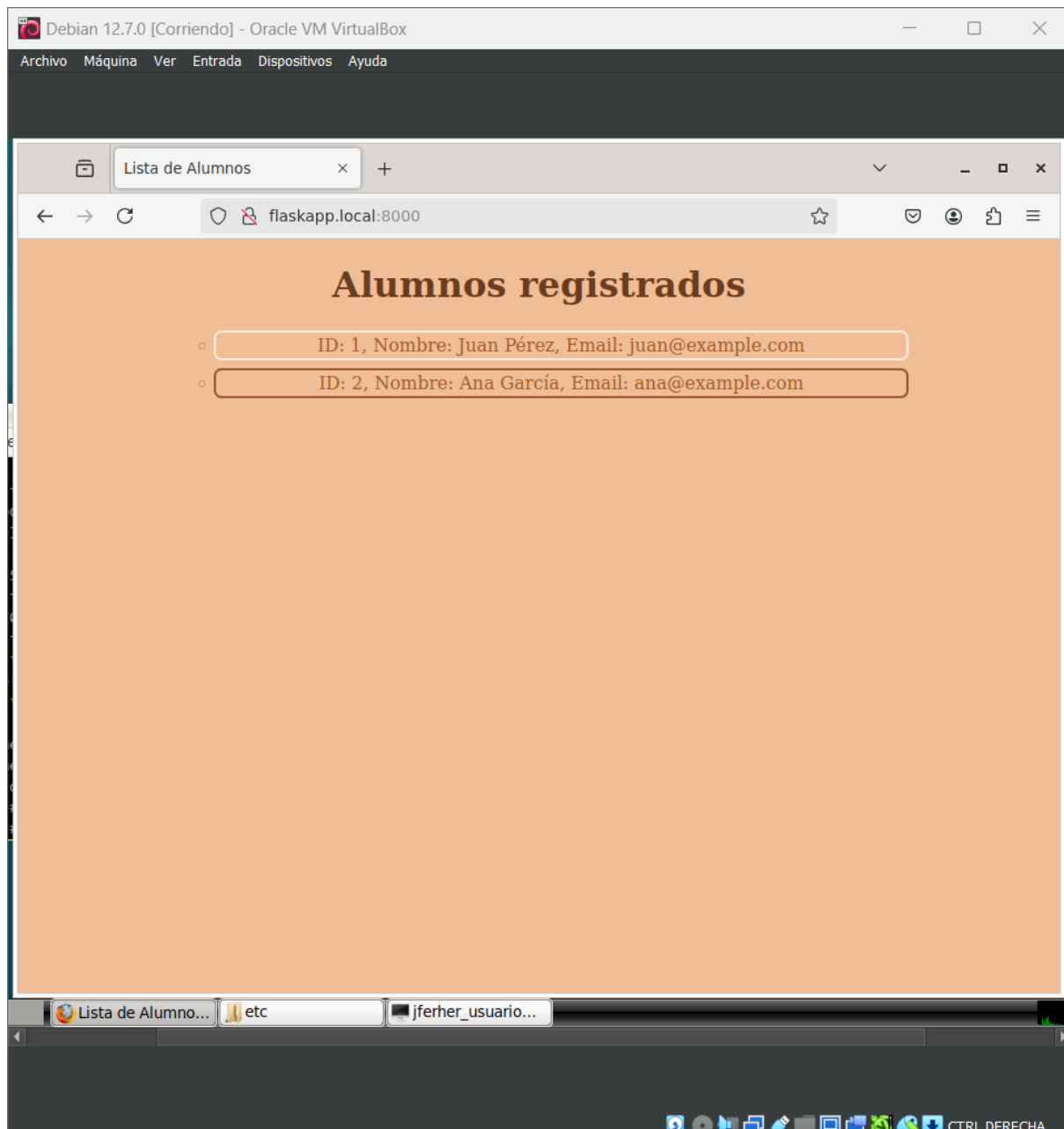
Entrega:

- Captura del archivo **WSGI** y del archivo de configuración de Apache.
- Captura del comando para habilitar el sitio.
- Captura de pantalla mostrando la aplicación en ejecución desde el navegador.

Tarea 5: Mejoras opcionales.

- **Añadir funcionalidad** para insertar nuevos alumnos a través de un formulario web.
- **Implementar funcionalidad** para actualizar o eliminar alumnos desde la página web.
- **Mejorar la apariencia** de la aplicación añadiendo estilos **CSS** o utilizando el framework **Bootstrap**.





5. Rúbrica de evaluación.

Criterio	Excelente (10 puntos)	Aceptable (7-9 puntos)	Insuficiente (4-6 puntos)	No realizado (0-3 puntos)
Instalación del entorno	Instalación completa y sin errores, evidencia clara.	Instalación completa, con pequeños errores corregidos.	Instalación incompleta o con errores significativos.	No se realizó o contiene errores graves.
Creación de la base de datos	Base de datos correctamente creada, datos insertados y visibles.	Base de datos creada, pero con errores menores.	Base de datos incompleta o con datos incorrectos.	No se creó la base de datos.
Desarrollo de la aplicación	Aplicación completa, funcional, y bien estructurada.	Aplicación funcional pero con pequeños errores.	Aplicación incompleta o con errores significativos.	No se desarrolló la aplicación.
Configuración de Apache	Apache correctamente configurado y sirviendo la aplicación.	Configuración realizada, pero con pequeños errores.	Configuración incompleta o con errores importantes.	No se configuró Apache correctamente.
Capturas y entrega de evidencia	Todas las entregas están bien documentadas y completas.	Documentación incompleta o con errores menores.	Documentación insuficiente o con errores graves.	No se entregaron las evidencias solicitadas.

Mejoras opcionales	Mejoras implementadas con éxito y funcionando correctamente.	Mejoras implementadas con pequeños errores.	Mejoras incompletas o no funcionales.	No se implementaron mejoras.
---------------------------	--	---	---------------------------------------	------------------------------

5.1 Rúbrica de evaluación detallada:

Tarea	Descripción de la tarea	Nota máxima acumulada
Tarea 1: Configuración del entorno	Instalación de Python, dependencias, y configuración de mod_wsgi en Apache.	3/10
Tarea 2: Creación de la base de datos	Crear la base de datos y ejecutar el archivo .sql en MariaDB.	5/10
Tarea 3: Desarrollo de la aplicación Python	Desarrollar la aplicación Python que se conecta a la base de datos y muestra una lista de alumnos en una página web.	6/10
Tarea 4: Configuración de Apache	Configurar Apache para servir la aplicación mediante WSGI.	8/10

5.2 Mejoras opcionales necesarias para alcanzar 10/10.

Tarea opcional	Descripción de la mejora	Nota
Funcionalidad de inserción	Implementar un formulario web para insertar nuevos alumnos.	+0.5 puntos
Funcionalidad de actualización	Implementar la posibilidad de actualizar y eliminar alumnos.	+0.5 puntos
Estilo visual mejorado	Aplicar mejoras en el estilo visual de la aplicación con CSS o Bootstrap.	+1 punto

5.3 Escalado de notas.

- **Solo tarea 1 completa: 3/10**
- **Solo hasta tarea 2 completa: 5/10**
- **Solo hasta tarea 3 completa: 6/10**
- **Solo hasta tarea 4 completa: 8/10** (máximo sin mejoras opcionales)
- **Con las tres mejoras opcionales implementadas: 10/10** (solo si completan todas las mejoras).

5.4 Nota final.

- La nota final estará escalada desde **3** hasta un máximo de **8 puntos** si se completan las **tareas básicas**.
- Para alcanzar el **10**, es obligatorio realizar todas las **mejoras opcionales**.