LatinasInCloud

Primeros pasos con Flask

{ Python Backend > Clase 3 }

Mentores: Brisa Sandoval, Matías Villagrán y Eva Durán

Módulo 3: Primeros pasos con Flask

LatinasInCloud

1. ¿Qué es Flask?
2. Crear un entorno virtual y activarlo
3. Instalación de Flask y creación de App.py
4. Crear una ruta simple “/” con texto de
5. Diferencia entre GET y POST
6. Ruta dinámica “/hola/<nombre>”

Flask

¿Qué es Flask?
Es un microframework para el desarrollo pequeño y mediano de apps.
Microframewok: Conjunto de herramientas de desarrollo que ofrece solo los compomentes eseciales para crear una app.

Minimalista

Librería de python

Ligero

Flexibilidad

Ideal para proyectos pequeños o microservicios

Manejo de peticiones HTTP (GET y POST)

Crear un entorno virtual y activarlo

¿Para qué crear el entorno y activarlo?

Para poder generar los archivos y dependencias dentro de la carpeta raíz del proyecto sin pasar a llevar otros proyectos

LatinasInCloud

Como crearlo y activarlo: en código

El código para crear el proyecto es:
python -m venv venv

Y el código para activarlo es: Windows

venv\Scripts\activate

Y para activarlo en Linux/Mac:

source venv/bin/activate.

Donde:
python: hace un llamado al interprete de python. Es como decirle “Oye, haz esto por mí”.
-m: significa “module”. Le estamos diciendo a python que use un módulo integrado.
venv: es el nombre del módulo que sirve para crear entornos virtuales.
venv: es el nombre de la carpeta donde se guardará el entorno. Aquí podemos colocarle el nombre que deseemos. Suele ser “venv” convencionalmente.

Para asociarlo mejor:
Imagina una mochila con sus distintos compartimentos, si instalas flask en la mochila A,
no afecta a la mochila B. Así evitamos errores, mantenemos el proyecto limpio y ordenado.

LatinasInCloud

Instalación de Flask y creación de App.py

¿Por qué instalar y crear el archivo app.py?

Se instala la librería Flask para utilizar sus componentes de microframework y se crea el archivo que contiene la configuración de proyecto.

Como instalar la librería y crear archivo

El código para instalar la librería:

```
pip install flask
```

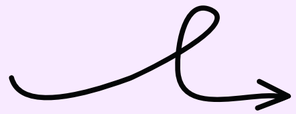
En nuestro VS o Carpeta raíz de proyecto, creamos el archivo “app.py”.

Donde:

pip: instalador de paquetes de python.
install: acción de instalar.
flask: la librería que vamos a instalar.

Para crear el archivo, debes crearlo manualmente en tu carpeta raíz del proyecto (clic derecho->nuevo documento->colocar nombre “app.py” y damos enter.

Dentro del documento nuevo, en nuestro editor de código, escribiremos el siguiente código (Diapositivas siguientes).



Como instalar y crear archivo

Donde:

from flask import flask: Importamos la clase flask desde la librería, es como traer las herramientas que usaremos.

app = flask(__name__): creamos nuestra app web.

(__name__): le dice a flask el nombre del archivo principal que está corriendo. Le ayuda a flask a ubicar las rutas, templates y configuraciones.

@app.route('/'):decorador que define la ruta raíz "/". Es como decir: "cuando alguien visite la página.

principal, muéstrale esto".

def home(): función que se ejecuta al visitar "/".

return "hola, este es mi primer flask": lo que se muestra en el navegador.

if __name__ == '__main__': comprueba si este archivo se está ejecutando directamente (y no importado desde otro lado)

debug=True: es como obtener un asistente que refresca todo por ti cuando haces algún cambio dentro del código.

Una vez ingresado el código, necesitamos echar a correr el servidor. Para echar a correr tu servidor usa el siguiente código:

python app.py

El código para comenzar a utiizar flask:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hola, este es mi primer servidor Flask 🎉"

if __name__ == '__main__':
    app.run(debug=True)
```

Con este código, ya tienes tu primer servidor web corriendo localmente.

Es la base sobre la que construirás páginas y aplicaciones más complejas

Crear una ruta simple “/” con texto de respuesta.

¿Qué es una ruta simple?

Imagina que tienes un restaurante, “/” sería la puerta de entrada hacia tu restaurante. Es decir, la bienvenida o página inicial de tu app web.

En el archivo “app.py”

El código para crear una ruta simple es:

```
@app.route('/')
def home():
    return "Bienvenidas al Bootcamp de Flask 🚀"
```

onde:

/: ruta raíz de la web.

def home(): función que responde cuando alguien entra a esa ruta

return”...”: es la respuesta que muestra la web.

Sirve para ver el resultado concreto de tu servidor corriendo

Diferencia entre GET, POST y Dinámica

GET: Se utiliza pra pedir información, por ejemplo: ver una página.

POST: Se utiliza para enviar información. Ej: Enviar un formulario.

Dinámica: Funciona como un buzón que recibe cartas con distintos nombres.

En el archivo “app.py”

GET

POST

Ruta dinámica

```
@app.route('/datos', methods=['GET'])
def obtener_datos():
  return "Esto es un GET"

@app.route('/enviar', methods=['POST'])
def enviar_datos():
  return "Esto es un POST"

@app.route('/hola/<nombre>')
def saludar(nombre):
  return f"¡Hola, {nombre}!"
```

Sirve para ver el resultado concreto de tu servidor corriendo

En el archivo “app.py”

GET

POST

Ruta dinámica

```
@app.route('/datos', methods=['GET'])
def obtener_datos():
  return "Esto es un GET"

@app.route('/enviar', methods=['POST'])
def enviar_datos():
  return "Esto es un POST"

@app.route('/hola/<nombre>')
def saludar(nombre):
  return f"¡Hola, {nombre}!"
```

Sirve para ver el resultado concreto de tu servidor corriendo

Finalmente, nuestro “app.py” debería estar así: (hasta la sección de la ruta dinámica)

LatinasInCloud

app.py > enviar_datos

```python
from flask import Flask, render_template, request, redirect, url_for

app = Flask(__name__)

@app.route('/')
def inicio():
    return "¡Hola, chiquillas bienvenidas!!!!"

@app.route('/web')
def mostrar_web():
    return render_template("index.html", name="Matías")

@app.route('/datos', methods=['GET'])
def obtener_datos():
    return "Esto es un GET"

@app.route('/enviar', methods=['POST'])
def enviar_datos():
    return "Esto es un POST"

@app.route('/hola/<nombre>')
def saludar(nombre):
    return f"¡Hola, {nombre}!"

@app.route('/tabla')
def tabla():
    personas = ["Pamela", "Angelina", "Natalia", "Valentina", "Sofía"]
    return render_template("tabla.html", personas=personas)


# Ruta para mostrar el formulario
@app.route('/form')
def form():
    return render_template("formulario.html")

```