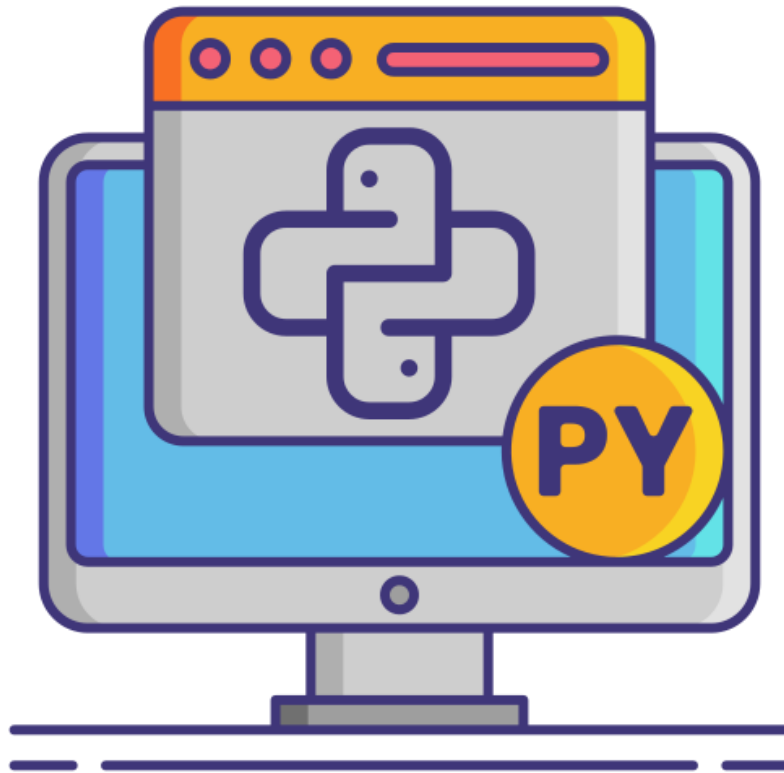


API con Python

Sprint 1 Lab 4



Felipe Izquierdo Romero

Índice

<u>1</u>	<u>Descripción del problema</u>
<u>2</u>	<u>Desarrollo</u>

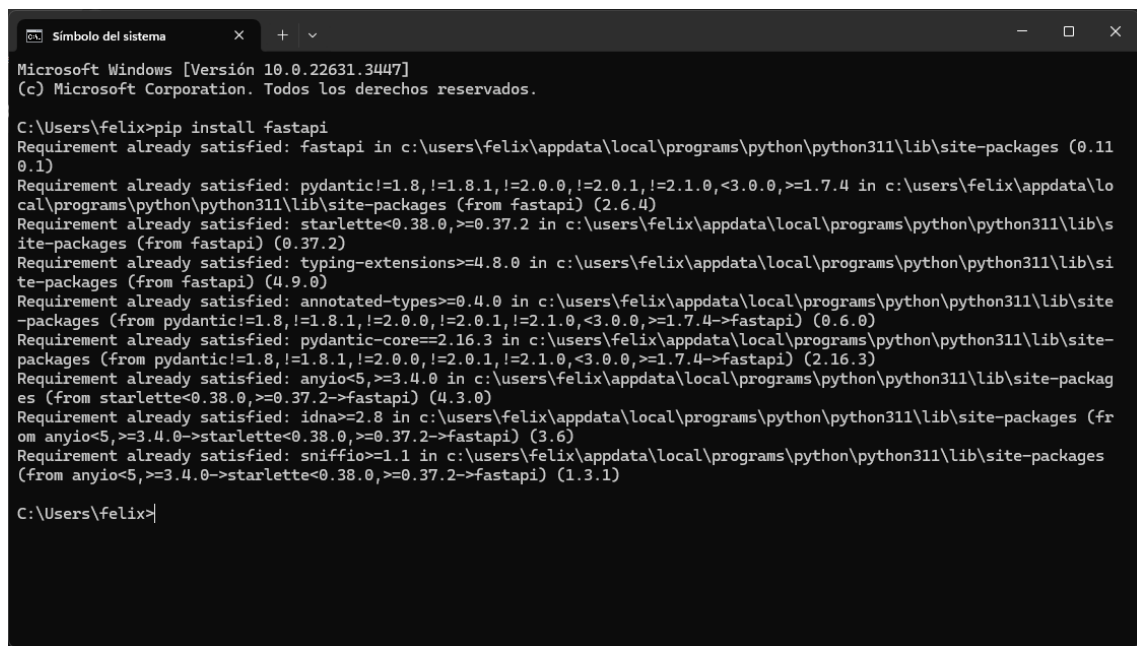
1. Descripción del problema

Para este laboratorio se deberá realizar una API en Python con la librería FastApi con 4 Endpoints por donde acceder. Cada uno de esos Endpoint recibirá 2 operadores y realizará una operación matemática, suma, resta, multiplicación y división.

2. Desarrollo

Para empezar el ejercicio instalaremos la librería FastApi y Uvicorn para crear el servidor con la consola de comandos de Windows “CMD”. Introduciendo el comando

```
pip install <nombre_del_paquete>
```

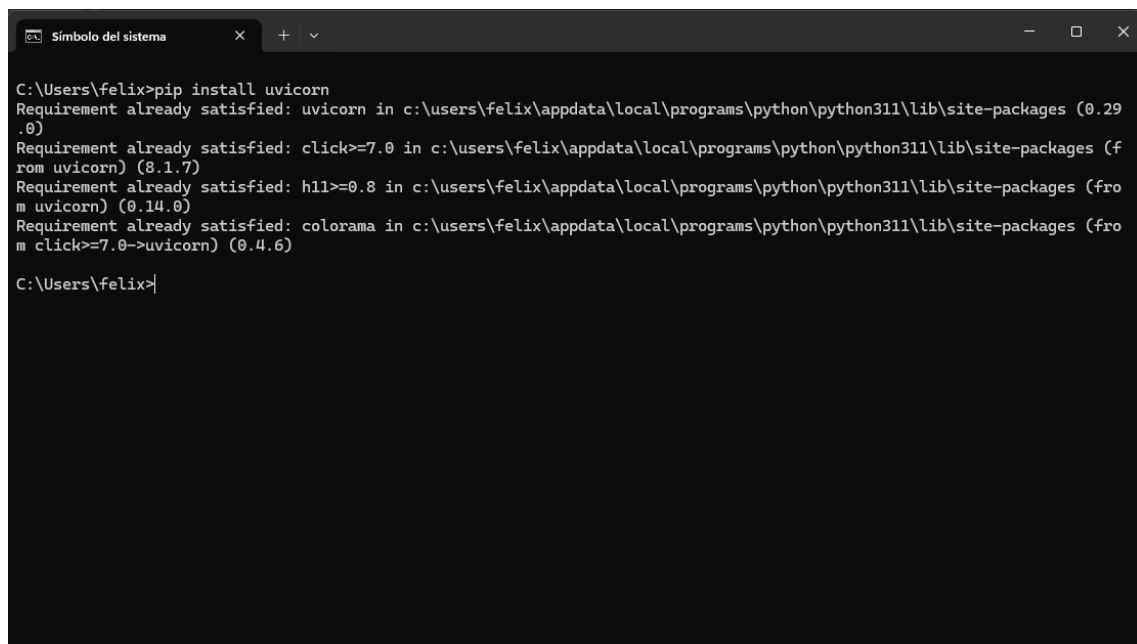


```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.3447]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\felix>pip install fastapi
Requirement already satisfied: fastapi in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (0.110.1)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from fastapi) (2.6.4)
Requirement already satisfied: starlette<0.38.0,>=0.37.2 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from fastapi) (0.37.2)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from fastapi) (4.9.0)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (2.16.3)
Requirement already satisfied: anyio<5,>=3.4.0 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from starlette<0.38.0,>=0.37.2->fastapi) (4.3.0)
Requirement already satisfied: idna>=2.8 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from anyio<5,>=3.4.0->starlette<0.38.0,>=0.37.2->fastapi) (3.6)
Requirement already satisfied: sniffio>=1.1 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from anyio<5,>=3.4.0->starlette<0.38.0,>=0.37.2->fastapi) (1.3.1)

C:\Users\felix>
```

Ilustración 1- Instalación de FastApi



```
Símbolo del sistema

C:\Users\felix>pip install uvicorn
Requirement already satisfied: uvicorn in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (0.29.0)
Requirement already satisfied: click>=7.0 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from uvicorn) (8.1.7)
Requirement already satisfied: h11>=0.8 in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from uvicorn) (0.14.0)
Requirement already satisfied: colorama in c:\users\felix\appdata\local\programs\python\python311\lib\site-packages (from click>=7.0->uvicorn) (0.4.6)

C:\Users\felix>
```

Ilustración 2- Instalación de Uvicorn

En la imagen se puede observar que ya están instalados los recursos necesarios para llevar a cabo el laboratorio.

Para empezar en el modulo de Python donde crearemos la API deberá hacerse la llamada a las librerías importándolas.

```
1 from fastapi import FastAPI
2 from datetime import datetime
```

Ilustración 3- Importar librerías

A continuación, se inicializa la instancia de FastApi

```
3
4 app = FastAPI()
5
```

Ilustración 4- Declaración de la instancia de FastApi

Después se crean los Endpoints escribiendo como acceder al Endpoint y los valores que recibirá.

Un bloque try except para hacer la conversión de string a número entero y un return con la operación realizada según corresponda al endpoint o mensaje de error si no se introdujo un valor correctamente.

```
6 @app.get('/suma/{valor1}/{valor2}')
7 def suma(valor1,valor2):
8     try:
9         return{'resultado' : int(valor1)+int(valor2)}
10    except:
11        return{'error' : 'Los parametros deben de ser numero enteros no decimales ni caracteres'}
```

Ilustración 5- Endpoint suma

```
13 @app.get('/resta/{valor1}/{valor2}')
14 def resta(valor1,valor2):
15     try:
16         return{'resultado' : int(valor1)-int(valor2)}
17    except:
18        return{'error' : 'Los parametros deben de ser numero enteros no decimales ni caracteres'}
```

Ilustración 6- Endpoint resta

```
19
20 @app.get('/multiplicacion/{valor1}/{valor2}')
21 def multiplicacion(valor1,valor2):
22     try:
23         return{'resultado' : int(valor1)*int(valor2)}
24    except:
25        return{'error' : 'Los parametros deben de ser numero enteros no decimales ni caracteres'}
```

Ilustración 7- Endpoint multiplicación

```
27 @app.get('/division/{valor1}/{valor2}')
28 def division(valor1,valor2):
29     try:
30         return{'resultado' : int(valor1)/int(valor2)}
31    except:
32        return{'error' : 'Los parametros deben de ser numero enteros no decimales ni caracteres'}
```

Ilustración 8- Endpoint división

```

6 @app.get('/')
7 def rootAPI():
8     return{'bienvenida':'Bienvenido a la API Calculadora'}
9

```

Ilustración 9- Raiz de la API

Para terminar el laboratorio iniciaremos el servidor desde un terminal de Python con el comando

```
uvicorn <nombre_del_modulo_python>:<nombre_de_instancia> --reload
```

```

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
PS C:\Users\felix\SprintsBootcamp\Sprint_1\Lab_4> uvicorn main:app --reload
INFO:      Will watch for changes in these directories: ['C:\Users\felix\SprintsBootcamp\Sprint_1\Lab_4']
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:      Started reloader process [18884] using StatReload
INFO:      Started server process [8860]
INFO:      Waiting for application startup.
INFO:      Application startup complete.

```

Ilustración 10- Inicio del servidor uvicorn

A continuación en el video se mostrara como acceder a cada Endpoint del servidor local de uvicorn y los datos que recibimos según que parámetros introducamos.

[Link del video](#)