

SPRINT 3 – WiseBet

Autor: Javier Cuartero Corredor

Contenido

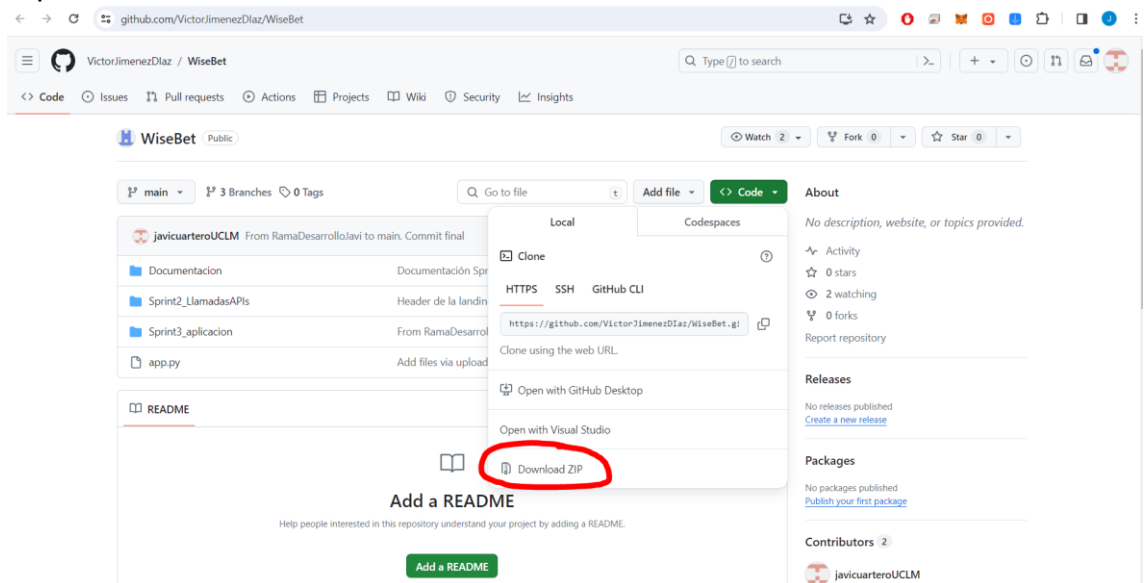
Manual de usuario	1
Credenciales para inicio de sesión	7
Test de integración	8

Manual de usuario

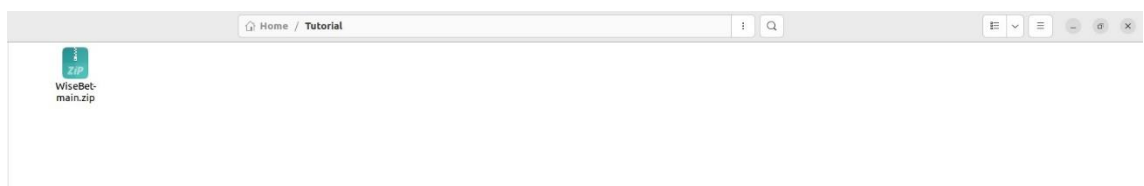
Antes de comenzar, requisito indispensable: tener instalado python3

*Recomendación: ejecutar en sistema operativo Linux/Mac

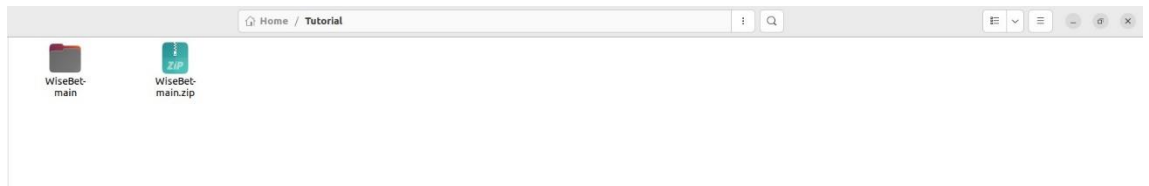
- 1- Accede <https://github.com/VictorJimenezDiaz/WiseBet> y descárgate el repositorio main.



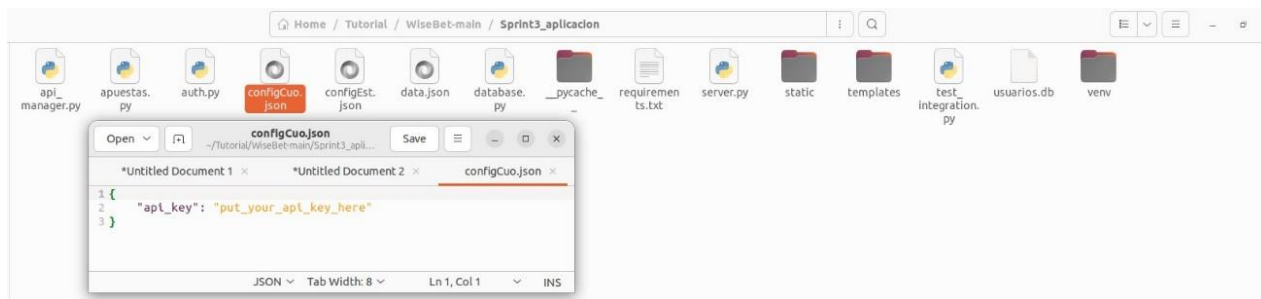
- 2- Una vez descargado guárdalo en la carpeta que tú quieras, en mi caso lo guardo en la carpeta 'Tutorial'.



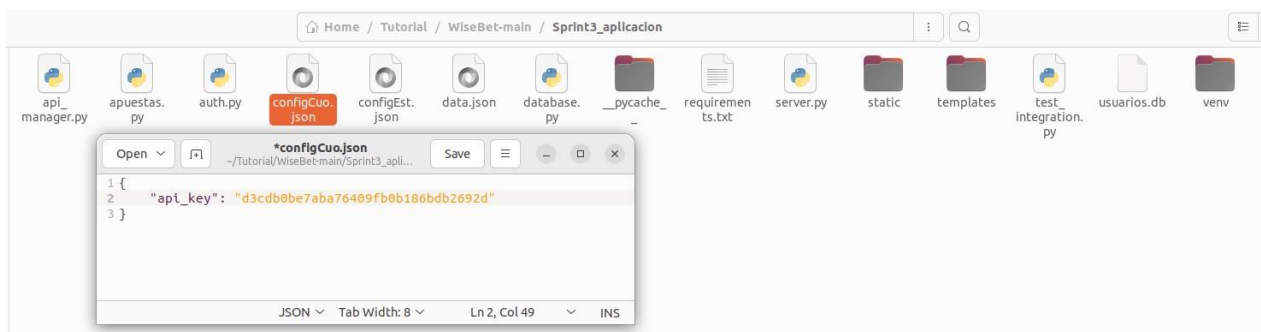
3- Descomprime el archivo



4- Una vez descomprimido, ve al archivo '**configCuo.json**' y ábrelo con tu editor de texto preferido.

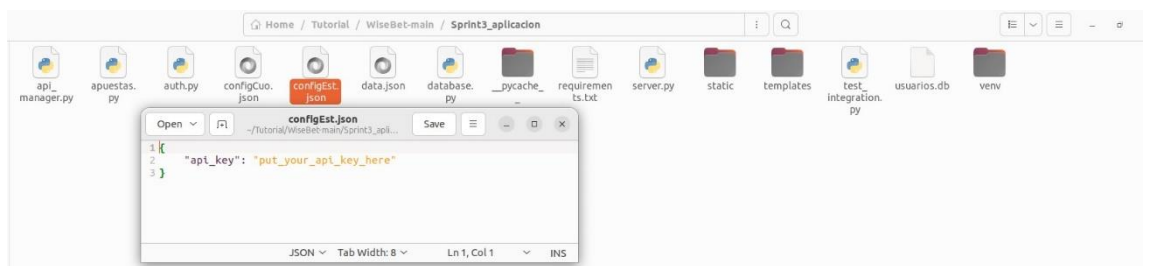


5- Una vez abierto, modifica lo que está entre " por lo siguiente: **d3cdb0be7aba76409fb0b186bdb2692d**



Y guardamos el archivo.

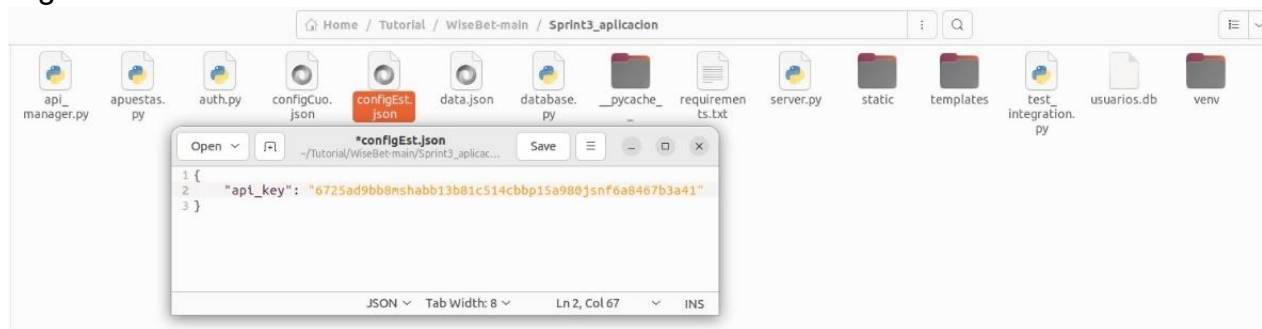
6- Hacemos lo mismo con el archivo '**configEst.json**'



Sustituimos por:

6725ad9bb8mshabb13b81c514cbbp15a980jsnf6a8467b3a41

Y guardamos.

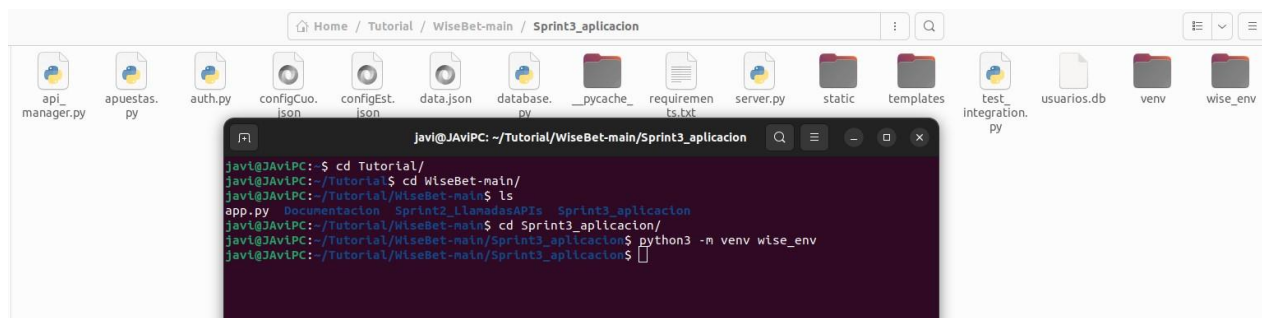


*Lo de poner las apikeys manualmente lo he hecho así para no tener las apikeys directamente en el repositorio github, que es público.

- 7- Una vez modificados y guardados los archivos, abrimos una terminal en la misma ubicación y creamos un entorno virtual.
Para crear el entorno virtual usaremos el comando:

(EN LINUX) \$ python3 -m venv wise_env

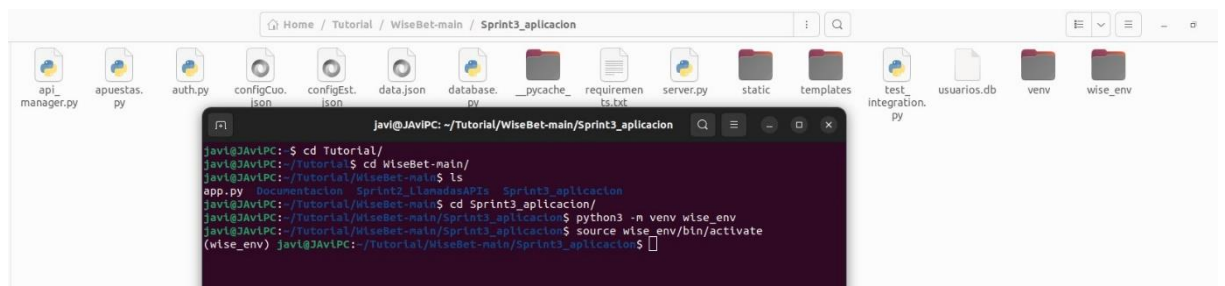
(EN WINDOWS) \$ python -m venv wise_env



- 8- Ya creado el entorno virtual, lo cargamos con

(EN LINUX)\$ python3 -m venv wise_env

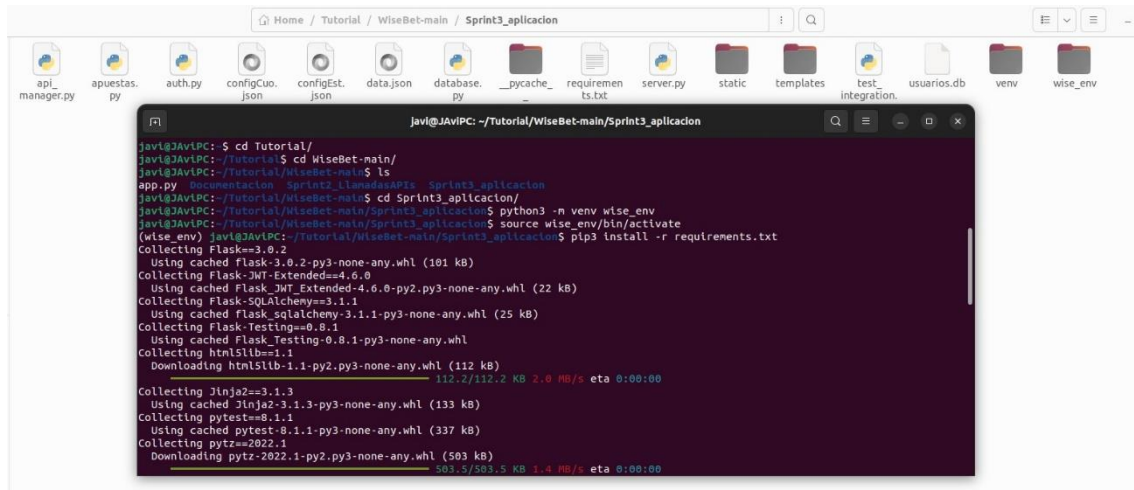
(EN WINDOWS) \$ wise_env\Scripts\activate



9- Instalamos librerías con:

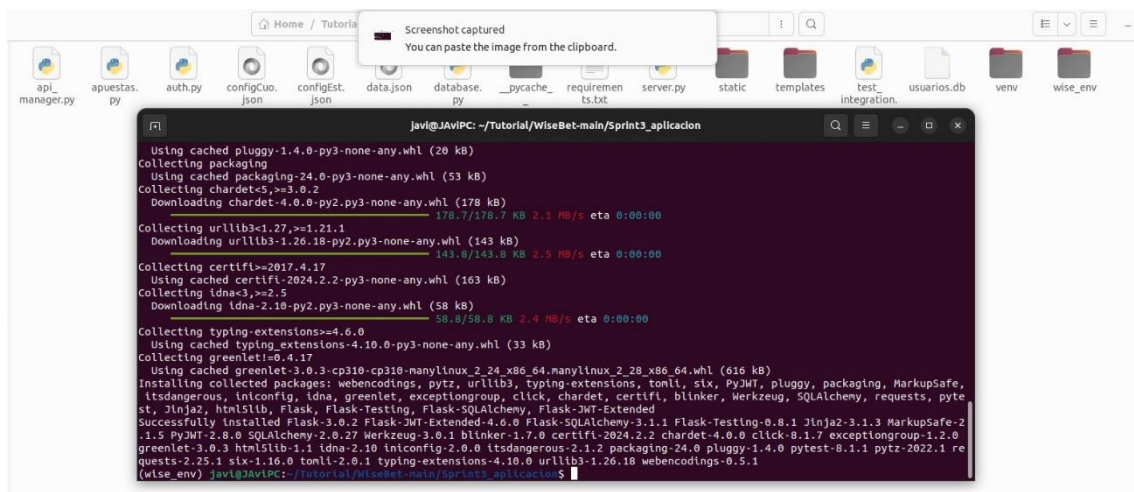
(EN LINUX) \$ pip3 install -r requirements.txt

(EN WINDOWS) \$ pip install -r requirements.txt



A terminal window titled 'Javi@JAVIPC: ~/Tutorial/WiseBet-main/Sprint3_aplicacion' showing the execution of pip3 install -r requirements.txt. The terminal output lists the collection and installation of various packages including Flask, Flask-JWT-Extended, Flask-SQLAlchemy, Flask-Testing, Flask-SQLAlchemy, Jinja2, pytest, and pytz. The progress bar shows the download of Flask==3.0.2 and Jinja2==3.1.3.

```
Javi@JAVIPC: ~/Tutorial/WiseBet-main/Sprint3_aplicacion
javi@JAVIPC:~/Tutorial$ cd WiseBet-main/
javi@JAVIPC:~/Tutorial/WiseBet-main$ ls
app.py  Documentation  Sprint2_llamadasAPIs  Sprint3_aplicacion
javi@JAVIPC:~/Tutorial/WiseBet-main$ cd Sprint3_aplicacion/
javi@JAVIPC:~/Tutorial/WiseBet-main/Sprint3_aplicacion$ python3 -m venv wise_env
javi@JAVIPC:~/Tutorial/WiseBet-main/Sprint3_aplicacion$ source wise_env/bin/activate
(wise_env) javi@JAVIPC:~/Tutorial/WiseBet-main/Sprint3_aplicacion$ pip3 install -r requirements.txt
Collecting Flask==3.0.2
  Using cached flask-3.0.2-py3-none-any.whl (101 kB)
Collecting Flask-JWT-Extended==4.6.0
  Using cached flask_jwt_extended-4.6.0-py2.py3-none-any.whl (22 kB)
Collecting Flask-SQLAlchemy==3.1.1
  Using cached flask_sqlalchemy-3.1.1-py3-none-any.whl (25 kB)
Collecting Flask-Testing==0.8.1
  Using cached flask_testing-0.8.1-py3-none-any.whl
Collecting html5lib==1.1
  Downloading html5lib-1.1-py2.py3-none-any.whl (112 kB)
 112.2/112.2 KB 2.0 MB/s eta 0:00:00
Collecting Jinja2==3.1.3
  Using cached Jinja2-3.1.3-py3-none-any.whl (133 kB)
Collecting pytest==8.1.1
  Using cached pytest-8.1.1-py3-none-any.whl (337 kB)
Collecting pytz==2022.1
  Downloading pytz-2022.1-py2.py3-none-any.whl (503 kB)
 503.5/503.5 KB 1.4 MB/s eta 0:00:00
```



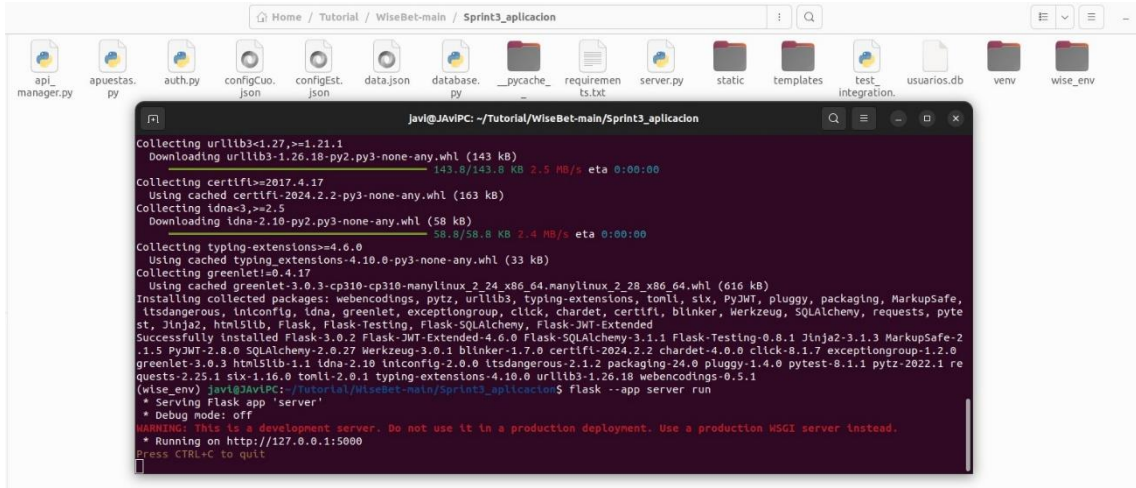
A terminal window titled 'Javi@JAVIPC: ~/Tutorial/WiseBet-main/Sprint3_aplicacion' showing the completion of the pip3 install -r requirements.txt command. The terminal output lists the collection and installation of various packages including pluggy, packaging, chardet, urllib3, certifi, idna, typing-extensions, greenlet, and Werkzeug. The progress bar shows the download of chardet==4.0.0 and urllib3==1.26.18. A 'Screenshot captured' notification is visible in the top right corner.

```
Javi@JAVIPC: ~/Tutorial/WiseBet-main/Sprint3_aplicacion
Using cached pluggy-1.4.0-py3-none-any.whl (20 kB)
Collecting packaging
  Using cached packaging-24.0-py3-none-any.whl (53 kB)
Collecting chardet<5,>=3.0.2
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
 178.7/178.7 KB 2.1 MB/s eta 0:00:00
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.18-py2.py3-none-any.whl (143 kB)
 143.0/143.0 KB 2.5 MB/s eta 0:00:00
Collecting certifi==2017.4.17
  Using cached certifi-2024.2.2-py3-none-any.whl (163 kB)
Collecting idna<3,>=2.5
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
 58.0/58.0 KB 2.4 MB/s eta 0:00:00
Collecting typing-extensions>=4.6.0
  Using cached typing_extensions-4.10.0-py3-none-any.whl (33 kB)
Collecting greenlet==0.4.17
  Using cached greenlet-3.0.3-cp310-cp310-manylinux_2_24_x86_64_manylinux_2_28_x86_64.whl (616 kB)
Installing collected packages: webencodings, pytz, urllib3, typing-extensions, tomli, six, PyJWT, pluggy, packaging, MarkupSafe, itsdangerous, iniconfig, idna, greenlet, exceptiongroup, click, chardet, certifi, blinker, Werkzeug, SQLAlchemy, requests, pyte
st, Jinja2, html5lib, Flask, Flask-Testing, Flask-SQLAlchemy, Flask-JWT-Extended
Successfully installed Flask-3.0.2 Flask-JWT-Extended-4.6.0 Flask-SQLAlchemy-3.1.1 Flask-Testing-0.8.1 Jinja2-3.1.3 MarkupSafe-2
.1.5 PyJWT-2.8.0 SQLAlchemy-2.0.27 Werkzeug-3.0.1 blinker-1.7.0 certifi-2024.2.2 chardet-4.0.0 click-8.1.7 exceptiongroup-1.2.0
greenlet-3.0.3 html5lib-1.1 idna-2.10 iniconfig-2.0.0 itsdangerous-2.1.2 packaging-24.0 pluggy-1.4.0 pytest-8.1.1 pytz-2022.1 re
quests-2.25.1 six-1.16.0 tomli-2.0.1 typing-extensions-4.10.0 urllib3-1.26.18 webencodings-0.5.1
(wise_env) javi@JAVIPC:~/Tutorial/WiseBet-main/Sprint3_aplicacion$
```

10-Ponemos en marcha el servidor flask con el comando

(EN LINUX) \$ flask -app server run --debug

(EN WINDOWS) \$ flask -app server --debug run



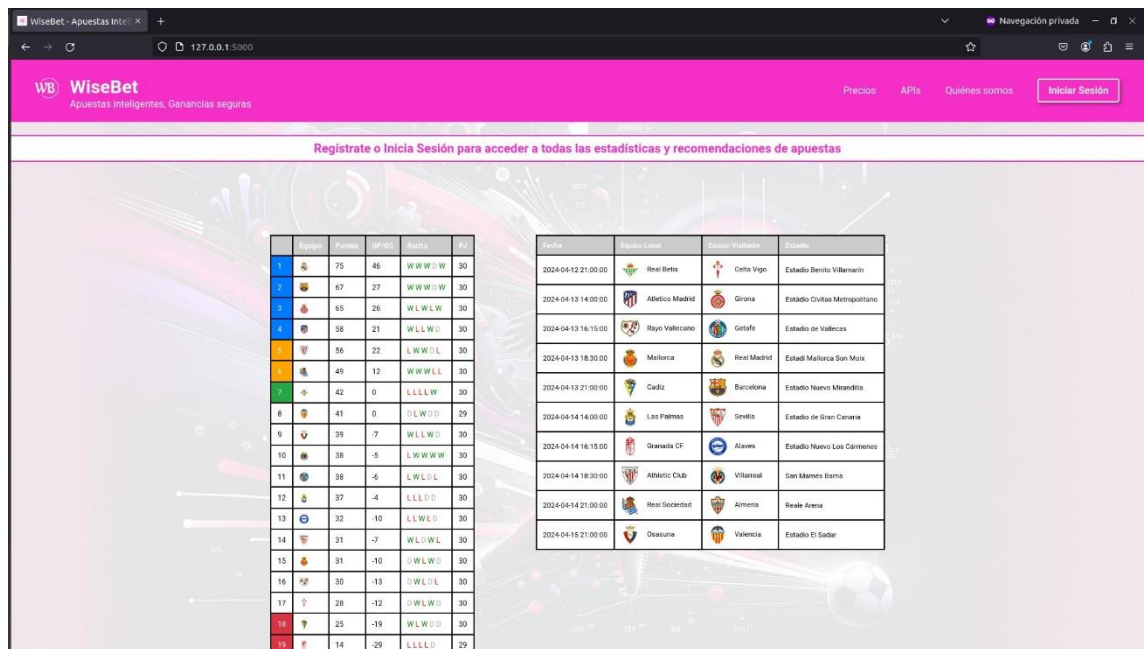
The screenshot shows a Windows file explorer window with the address bar set to 'Home / Tutorial / WiseBet-main / Sprint3_aplicacion'. The file list includes 'api_manager.py', 'apuestas.py', 'auth.py', 'configCuo.json', 'configEst.json', 'data.json', 'database.py', '__pycache__', 'requirements.txt', 'server.py', 'static', 'templates', 'test_integration.py', 'usuarios.db', 'venv', and 'wise_env'. Overlaid on the file explorer is a terminal window titled 'Javi@JAVIPC: ~/Tutorial/WiseBet-main/Sprint3_aplicacion'. The terminal shows the output of the 'flask --app server run --debug' command. It lists the installation of various Python packages including urllib3, certifi, idna, typing-extensions, greenlet, and many others. At the bottom, it states 'Running on http://127.0.0.1:5000' and 'Press CTRL+C to quit'.

11-Abrimos nuestro navegador y accedemos a la dirección:

<http://127.0.0.1/update-data>

Esto es para actualizar los datos de la api en el json de los datos, esto lo he hecho porque teníamos un número muy limitado de peticiones a una de las apis, y lo he hecho de tal forma que cada vez que cargue esta url se actualizan los datos y se escriben en un json, luego el código html lee los datos desde este json y no desde la api directamente. Lo he hecho así porque cada vez que recargaba la página principal para probar algo gastaba peticiones y de esta forma he conseguido reducir el número de peticiones.

12-Ya tenemos nuestra aplicación web funcionando desde la url <http://127.0.0.1/update-data>



13-Para iniciar sesión ver anexo [credenciales](#) al final del documento.

Credenciales para inicio de sesión

-usuario: felix@gmail.com

-contraseña: felix

Test de integración

He implementado un test de integración (**test_integration.py**) que prueba los servicios de iniciar sesión, registrarse y prueba el servicio de obtención de datos de las apis.

```
test_integration.py
home > javi > ISI > WiseBet > WiseBet > Sprint3_aplicacion > test_integration.py > ...

7 def app():
11     _app
13     with _app.app_context():
14         db.drop_all()
15
16 @pytest.fixture
17 def client(app):
18     return app.test_client()
19
20 @pytest.fixture
21 def runner(app):
22     return app.test_cli_runner()
23
24
25 def test_user_registration(client):
26     with client.application.app_context():
27         response = client.post('/register', json={
28             'nombre': 'Test',
29             'apellidos': 'User',
30             'correo': 'test@example.com',
31             'telefono': '1234567890',
32             'password': 'testpassword'
33         })
34         assert response.status_code == 302
35         user = User.query.filter_by(email='test@example.com').first()
36         assert user is not None
37
38 def test_user_login(client):
39     with client.application.app_context():
40         hashed_password = generate_password_hash('testpassword')
41         user = User(username='Test', email='test@example.com', password=hashed_password)
42         db.session.add(user)
43         db.session.commit()
44
45         # Intenta hacer login con el usuario creado
46         response = client.post('/login', json={
47             'correo': 'test@example.com',
48             'password': 'testpassword'
49         })
50         assert response.status_code == 302
51
52
53 def test_data_update(client):
54     response = client.get('/update-data')
55     assert response.status_code == 200
56
57
58
```

RESULTADO DEL TEST

```
javi@JaviPC: ~/ISI/WiseBet/WiseBet/Sprint3_aplicacion$ pytest
===== test session starts =====
platform linux -- Python 3.10.12, pytest-8.1.1, pluggy-1.4.0
rootdir: /home/javi/ISI/WiseBet/WiseBet/Sprint3_aplicacion
collected 3 items

test_integration.py ... [100%]

===== 3 passed in 1.39s =====
```