

# PROYECTO FINAL

## Github Actions

Primero se crea el programa en Python. En nuestro caso se realizará una calculadora con las siguientes funciones:

```
1  def sumar(a, b):
2      return a + b
3
4  def restar(a, b):
5      return a - b
6
7  def multiplicar(a, b):
8      return a * b
```

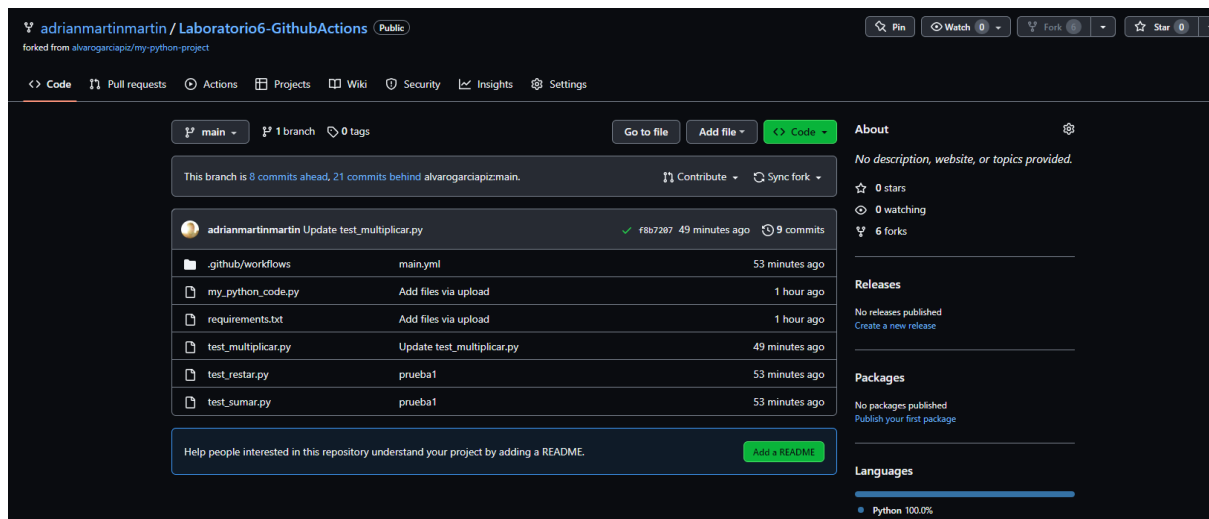
Para la parte de *testing* se realizan los siguientes *scripts*:

```
1  from my_python_code import sumar
2  def test_sumar():
3      assert sumar(2, 3) == 5
4      assert sumar(0, 0) == 0
5      assert sumar(-1, 1) == 0
```

```
1  from my_python_code import restar
2  def test_restar():
3      assert restar(5, 2) == 3
4      assert restar(0, 0) == 0
5      assert restar(-1, 1) == -2
```

```
1  from my_python_code import multiplicar
2  def test_multiplicar():
3      assert multiplicar(2, 3) == 6
4      assert multiplicar(0, 0) == 0
5      assert multiplicar(-1, 1) == -1
```

En el archivo *requirements.txt* especificamos la versión de *pytest*.



Dentro de *Actions* el archivo *main.yml* será:

```
1  name: Python Unit Tests
2  on:
3    push:
4    pull_request:
5  jobs:
6    build:
7      runs-on: ubuntu-latest
8      steps:
9        - uses: actions/checkout@v2
10       - name: Set up Python
11         uses: actions/setup-python@v2
12         with:
13           python-version: '3.9'
14       - name: Install dependencies
15         run: |
16           python -m pip install --upgrade pip
17           pip install -r requirements.txt
18           pip install pytest
19       - name: Run tests
20         run: |
21           pytest .
```

En el momento que hagamos un push el programa se ejecutará. En este caso el resultado será exitoso porque en el *testing* hemos indicado que las operaciones sean correctas:

The screenshot shows a GitHub Actions workflow run for 'Proyecto Final Github Actions' on the 'main' branch. The workflow is triggered by a push to the 'main' branch. The run is successful and completed in 12 seconds. The steps are: Set up job (2s), Run actions/checkout@v2 (1s), Set up Python (0s), Install dependencies (7s), Run tests (0s), Post Set up Python (0s), Post Run actions/checkout@v2 (1s), and Complete job (0s). The 'Run tests' step shows the output of a pytest command, indicating that 3 tests passed in 0.02s.

```
1 ► Run pytest .
7 ===== test session starts =====
8 platform linux -- Python 3.9.16, pytest-6.2.2, py-1.11.0, pluggy-0.13.1
9 rootdir: /home/runner/work/Laboratorio6-GithubActions/Laboratorio6-GithubActions
10 collected 3 items
11
12 test_multiplicar.py . [ 33%]
13 test_restar.py . [ 66%]
14 test_sumar.py . [100%]
15
16 ===== 3 passed in 0.02s =====
```

Si ahora modificamos el siguiente *script*:

```
1 from my_python_code import sumar
2 def test_sumar():
3     assert sumar(2, 3) == 8
4     assert sumar(0, 0) == 0
5     assert sumar(-1, 1) == 0
```

Obtendremos un error (2 + 3 no es 8):

The screenshot shows a GitHub Actions workflow run for 'Modificación del Proyecto final de github Actions' on the 'main' branch. The workflow is triggered by a push to the 'main' branch. The run is failed and completed in 24 seconds. The steps are: Set up job (2s), Run actions/checkout@v2 (1s), Set up Python (0s), Install dependencies (7s), Run tests (0s), Post Set up Python (0s), Post Run actions/checkout@v2 (1s), and Complete job (0s). The 'Run tests' step shows the output of a pytest command, indicating that 3 tests failed in 0.02s.

```
1 ► Run pytest .
7 ===== test session starts =====
8 platform linux -- Python 3.9.16, pytest-6.2.2, py-1.11.0, pluggy-0.13.1
9 rootdir: /home/runner/work/Laboratorio6-GithubActions/Laboratorio6-GithubActions
10 collected 3 items
11
12 test_multiplicar.py . [ 33%]
13 test_restar.py . [ 66%]
14 test_sumar.py . [100%]
15
16 ===== 3 passed in 0.02s =====
```

Run tests

```
1  ▶ Run pytest .
7  ===== test session starts =====
8  platform linux -- Python 3.9.16, pytest-6.2.2, py-1.11.0, pluggy-0.13.1
9  rootdir: /home/runner/work/Laboratorio6-GithubActions/Laboratorio6-GithubActions
10 collected 3 items
11
12 test_multiplicar.py . [ 33%]
13 test_restar.py . [ 66%]
14 test_sumar.py F [100%]
15
16 ===== FAILURES =====
17 _____ test_sumar _____
18
19     def test_sumar():
20 >     assert sumar(2, 3) == 8
21 E     assert 5 == 8
22 E     + where 5 = sumar(2, 3)
23
24 test_sumar.py:3: AssertionError
25 ===== short test summary info =====
26 FAILED test_sumar.py::test_sumar - assert 5 == 8
27 ===== 1 failed, 2 passed in 0.03s =====
28 Error: Process completed with exit code 1.
```