

GITHUB ACTIONS – Proyecto final

Para el proyecto final realizamos en Python los métodos que utilizaría una calculadora y los introducimos en el fichero `my_python_code.py`.

```
my_python_code.py X
test-github-actions > my_python_code.py > sumar
1 def sumar(a, b):
2     return a + b
3 def restar(a, b):
4     return a - b
5 def multiplicar(a, b):
6     return a * b
```

También realizamos unos ficheros con los test de estos métodos.

```
test_sumar.py X
test-github-actions > test_sumar.py > ...
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
```

```
test_restar.py X
test-github-actions > test_restar.py > ...
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
```

```
test_multiplicar.py X
test-github-actions > test_multiplicar.py > ...
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 *requirements.txt* incluimos las librerías y versiones que vamos a utilizar más adelante en GitHub Actions. En este caso, únicamente utilizamos *pytest*.

```
≡ requirements.txt X
test-github-actions > ≡ requirements.txt
1   pytest==6.2.2
```

Por último, en el directorio *.github/workflows* incluimos un fichero *main.yml* que ofrece la información de los pasos a seguir para los GitHub Actions.

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

Ahora, en cuanto realizamos una acción *push* o *pull request* se ejecutará este script que instala Python, las dependencias especificadas en *requirements.txt* y por último ejecuta los test.

El resultado de los tests lo podemos ver en el apartado *Actions* en GitHub.

The screenshot shows the GitHub Actions interface for a workflow named 'Python Unit Tests'. The workflow is currently running, and the 'build' job is the active step. The job status is 'succeeded' and it was completed '1 hour ago in 10s'. The 'build' job contains several steps: 'Set up job', 'Run actions/checkout@v2', 'Set up Python', 'Install dependencies', and 'Run tests'. The 'Run tests' step is expanded, showing the execution of 'pytest .'. The output of the tests is displayed, showing three tests passing: 'test_multiplicar.py' (138ms), 'test_restar.py' (60ms), and 'test_sumar.py' (100ms). The final result is '3 passed in 0.02s'.