

## PRUEBAS UNITARIAS

EJ01:

```
PS C:\Users\UsuarioT\Documents\ReposGit\ejercicios-u1-AngelVal06> pytest -v
===== test session starts =====
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0 -- C:\Users\UsuarioT\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\UsuarioT\Documents\ReposGit\ejercicios-u1-AngelVal06
collected 6 items

tests/test_prueba1.py::test_comprobar_numero PASSED [ 16%]
tests/test_prueba1.py::test_comprobar_numero_params[1-1-0] PASSED [ 33%]
tests/test_prueba1.py::test_comprobar_numero_params[0-0-0] PASSED [ 50%]
tests/test_prueba1.py::test_comprobar_numero_params[100--100-100] PASSED [ 66%]
tests/test_prueba1.py::test_comprobar_numero_params[-15--1--1] PASSED [ 83%]
tests/test_prueba1.py::test_comprobar_numero_params[-3-8-8] PASSED [100%]

===== 6 passed in 0.07s =====
PS C:\Users\UsuarioT\Documents\ReposGit\ejercicios-u1-AngelVal06>
```

```
test_comprobar_numero_params[1-1-0]

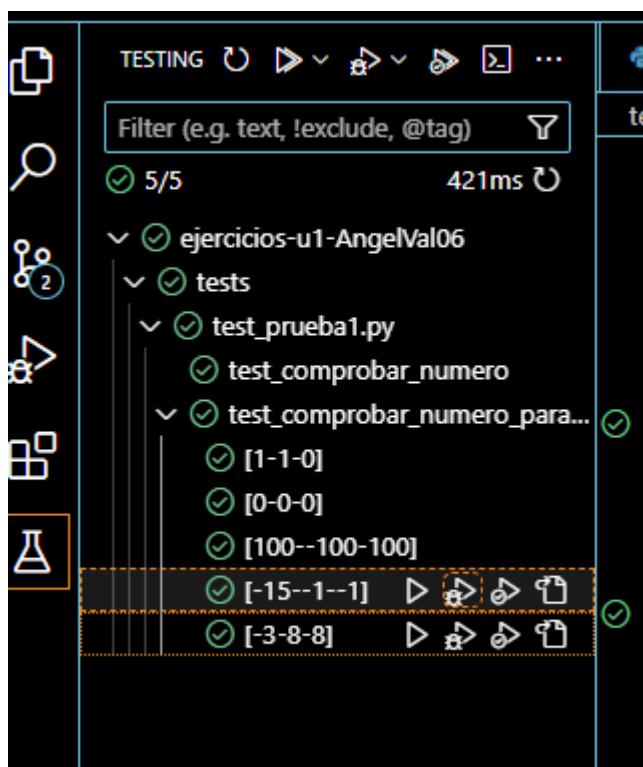
input_n1 = 1, input_n2 = 1, expected = 0

@pytest.mark.parametrize(
    "input_n1, input_n2, expected",
    [
        (1, 1, 0),
        (0, 0, 0),
        (100, -100, 100),
        (-15, -1, -1),
        (-3, 8, 8),
    ]
)
def test_comprobar_numero_params(input_n1, input_n2, expected):
    assert comprobar_numero(input_n1, input_n2) == expected
E     assert 1 == 0
E     + where 1 = comprobar_numero(1, 1)

tests\test_prueba1.py:24: AssertionError

===== short test summary info =====
FAILED tests/test_prueba1.py::test_comprobar_numero - assert 1 == 0
FAILED tests/test_prueba1.py::test_comprobar_numero_params[1-1-0] - assert 1 == 0
===== 2 failed, 4 passed in 0.12s =====
PS C:\Users\UsuarioT\Documents\ReposGit\ejercicios-u1-AngelVal06>
```

EJ02:



✖ 4/5

493ms ↺

▼ ✖ ejercicios-u1-AngelVal06

▼ ✖ tests ▶ ⚙️ 🔍 📄

▼ ✖ test\_prueba1.py

✓ test\_comprobar\_numero

▼ ✖ test\_comprobar\_numero\_para...

✖ [1-1-0]

✓ [0-0-0]

✓ [100--100-100]

✓ [-15--1--1]

✓ [-3-8-8] ▶ ⚙️ 🔍 📄