



Medición de Coverage para Python

Andrés Quiroz Valdovinos
737846

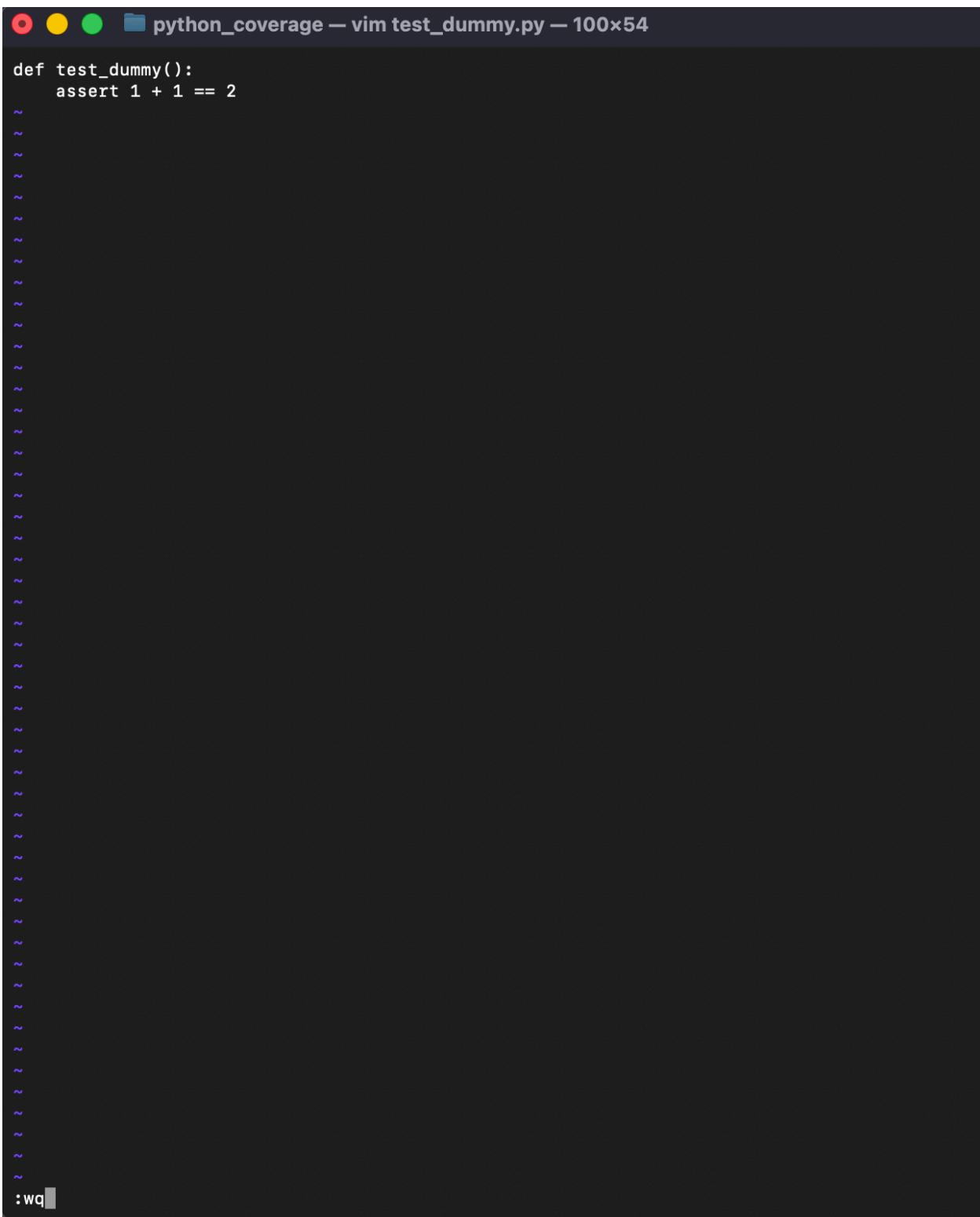
Repositorio completo:
<https://github.com/AndresQuiVal/coverage-python>



NOTA: los archivos .xml de la práctica se encuentran en el repo

Paso 1:

```
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % touch calculator.py
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % touch test_calculator.py
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % touch .coveragerc
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % ls
calculator.py          test_calculator.py
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % ls -la
total 0
drwxr-xr-x  5 andresquirozvaldovinos  staff   160 Nov 12 23:21 .
drwxr-xr-x  5 andresquirozvaldovinos  staff   160 Nov 12 23:21 ..
-rw-r--r--  1 andresquirozvaldovinos  staff     0 Nov 12 23:21 .coveragerc
-rw-r--r--  1 andresquirozvaldovinos  staff     0 Nov 12 23:20 calculator.py
-rw-r--r--  1 andresquirozvaldovinos  staff     0 Nov 12 23:20 test_calculator.py
```



A screenshot of a terminal window titled "python_coverage — vim test_dummy.py — 100x54". The window shows a single line of Python code:

```
def test_dummy():
    assert 1 + 1 == 2
```

The terminal has a dark background with light-colored text. The status bar at the bottom shows the command ":wq".

```
stall --upgrade pip' command.  
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % python3 -m pytest test_dummy.py ]  
===== test session starts =====  
platform darwin -- Python 3.9.6, pytest-8.4.2, pluggy-1.6.0  
rootdir: /Users/andresquirozvaldovinos/Documents/UNIVERSIDAD/COVERAGE/python_coverage  
plugins: html-4.1.1, metadata-3.1.1, anyio-3.7.1, cov-7.0.0  
collected 1 item  
  
test_dummy.py . [100%]  
  
===== 1 passed in 0.01s =====
```

PASO 2


```
python_coverage — vim test_calculator.py — 100x54

import pytest
from calculator import Calculator

@pytest.fixture
def calc():
    return Calculator()

def test_add(calc):
    assert calc.add(2, 3) == 5
    assert calc.add(-1, 1) == 0
    assert calc.add(0, 0) == 0

def test_subtract(calc):
    assert calc.subtract(5, 3) == 2
    assert calc.subtract(3, 5) == -2

def test_multiply(calc):
    assert calc.multiply(2, 3) == 6
    assert calc.multiply(-2, 3) == -6

def test_divide(calc):
    assert calc.divide(6, 2) == 3
    with pytest.raises(ValueError, match="Cannot divide by zero"):
        calc.divide(5, 0)

def test_power(calc):
    assert calc.power(2, 3) == 8
    assert calc.power(5, 0) == 1

def test_is_positive(calc):
    assert calc.is_positive(5) is True
    assert calc.is_positive(-3) is False
    # Nota: No probamos el caso number == 0

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
python_coverage — vim .coveragerc — 100x54

[run]
source = .
omit =
    test_*.py
    */__init__.py

[report]
show_missing = true
precision = 2
~
```

Al ejecutar los tests del calculator:

```
andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % python3 -m pytest --cov=. --cov-report=term-missing --cov-report=html
=====
platform darwin -- Python 3.9.6, pytest-8.4.2, pluggy-1.6.0
rootdir: /Users/andresquirozvaldovinos/Documents/UNIVERSIDAD/COVERAGE/python_coverage
plugins: html-4.1.1, metadata-3.1.1, anyio-3.7.1, cov-7.0.0
collected 7 items

test_calculator.py ..... [ 85%]
test_dummy.py . [100%]

=====
tests coverage =====
----- coverage: platform darwin, python 3.9.6-final-0 -----
Name     Stmts  Miss  Cover  Missing
-----
calculator.py     19      1  94.74%   25
-----
TOTAL          19      1  94.74%
Coverage HTML written to dir htmlcov
=====
7 passed in 0.04s =====
andresquirozvaldovinos@Andress-MacBook-Pro python_coverage %
```

Coverage report .html

Coverage report: 94.74%

Files Functions Classes hide covered

coverage.py v7.10.7, created at 2025-11-12 23:26 -0600

File ▲	statements	missing	excluded	coverage
calculator.py	19	1	0	94.74%
Total	19	1	0	94.74%

coverage.py v7.10.7, created at 2025-11-12 23:26 -0600

análisis rápido con coverage:

```
andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % python3 -m coverage report -m
Name     Stmts  Miss  Cover  Missing
-----
calculator.py    19     1  94.74%   25
-----
TOTAL          19     1  94.74%
```

Ahora la nueva prueba de is_positive

```
import pytest
from calculator import Calculator

@pytest.fixture
def calc():
    return Calculator()

def test_add(calc):
    assert calc.add(2, 3) == 5
    assert calc.add(-1, 1) == 0
    assert calc.add(0, 0) == 0

def test_subtract(calc):
    assert calc.subtract(5, 3) == 2
    assert calc.subtract(3, 5) == -2

def test_multiply(calc):
    assert calc.multiply(2, 3) == 6
    assert calc.multiply(-2, 3) == -6

def test_divide(calc):
    assert calc.divide(6, 2) == 3
    with pytest.raises(ValueError, match="Cannot divide by zero"):
        calc.divide(5, 0)

def test_power(calc):
    assert calc.power(2, 3) == 8
    assert calc.power(5, 0) == 1

def test_is_positive(calc):
    assert calc.is_positive(5) is True
    assert calc.is_positive(-3) is False
    assert calc.is_positive(0) is None # esta es la nueva prueba
```

ahora las pruebas:

```

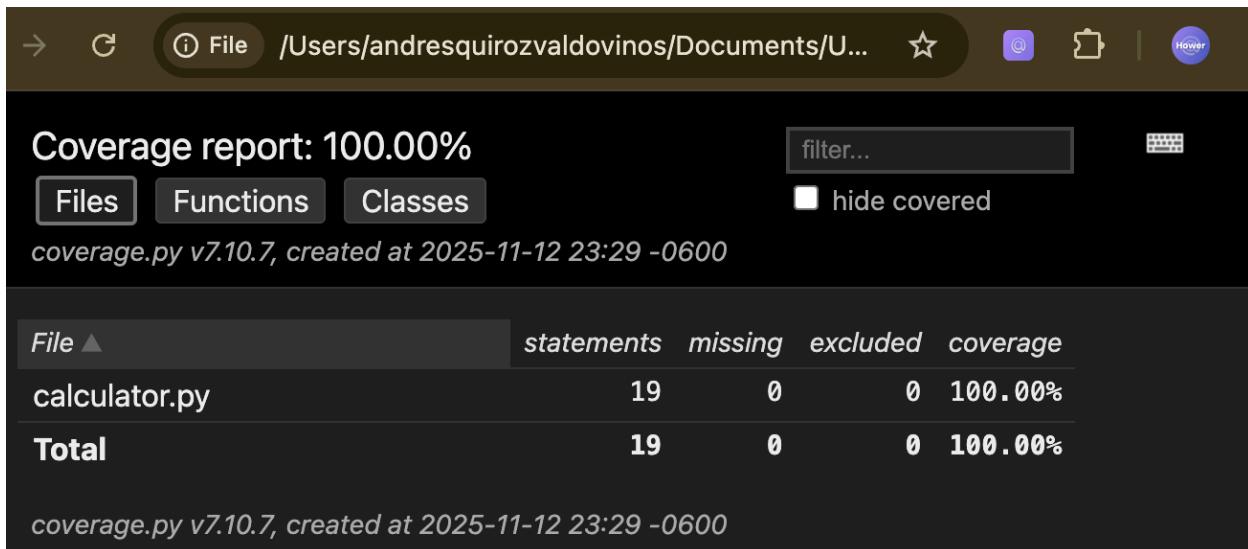
andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % python3 -m pytest --cov=. --cov-report=term-missing --cov-report=html

=====
platform darwin -- Python 3.9.6, pytest-8.4.2, pluggy-1.6.0
rootdir: /Users/andresquirozvaldovinos/Documents/UNIVERSIDAD/COVERAGE/python_coverage
plugins: html-4.1.1, metadata-3.1.1, anyio-3.7.1, cov-7.0.0
collected 7 items

test_calculator.py .....
test_dummy.py .

=====
tests coverage =====
coverage: platform darwin, python 3.9.6-final-0

Name     Stmts  Miss   Cover  Missing
-----
calculator.py    19     0  100.00%
-----
TOTAL          19     0  100.00%
Coverage HTML written to dir htmlcov
===== 7 passed in 0.04s =====
[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % open htmlcov/index.html ]
```



Nuevos tests:

```
python_coverage — vim test_calculator.py — 100x54
```

```
import pytest
from calculator import Calculator

@pytest.fixture
def calc():
    return Calculator()

def test_add(calc):
    assert calc.add(2, 3) == 5
    assert calc.add(-1, 1) == 0
    assert calc.add(0, 0) == 0

def test_subtract(calc):
    assert calc.subtract(5, 3) == 2
    assert calc.subtract(3, 5) == -2

def test_multiply(calc):
    assert calc.multiply(2, 3) == 6
    assert calc.multiply(-2, 3) == -6

def test_divide(calc):
    assert calc.divide(6, 2) == 3
    with pytest.raises(ValueError, match="Cannot divide by zero"):
        calc.divide(5, 0)

def test_power(calc):
    assert calc.power(2, 3) == 8
    assert calc.power(5, 0) == 1

def test_is_positive(calc):
    assert calc.is_positive(5) is True
    assert calc.is_positive(-3) is False
    assert calc.is_positive(0) is None # esta es la nueva prueba

def test_add_decimals(calc):
    assert calc.add(2.5, 3.7) == pytest.approx(6.2)
```

y el nuevo .coveragerc:

```
python_coverage — vim .coveragerc — 100x54

[run]
source = .
omit =
    test_*.py
    */__init__.py

[report]
show_missing = true
precision = 2
exclude_lines =
    pragma: no cover
    if False:
```

y el workflow de github:

```
python_coverage — vim .github/workflows/test.yml — 100x54

name: Run Tests
on: [push]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Set up Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.11'
      - name: Install dependencies
        run: pip install pytest pytest-cov
      - name: Run tests with coverage
        run: pytest --cov=. --cov-report=xml
      - name: Upload coverage
        uses: codecov/codecov-action@v3
```

nuevo test ejecutado:

```

andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % python3 -m pytest --cov=. --cov-report=xml

=====
platform darwin -- Python 3.9.6, pytest-8.4.2, pluggy-1.6.0
rootdir: /Users/andresquirozvaldovinos/Documents/UNIVERSIDAD/COVERAGE/python_coverage
plugins: html-4.1.1, metadata-3.1.1, anyio-3.7.1, cov-7.0.0
collected 8 items

test_calculator.py ..... [ 87%]
test_dummy.py . [100%]

=====
tests coverage =====
coverage: platform darwin, python 3.9.6-final-0 ----

Coverage XML written to file coverage.xml
=====
8 passed in 0.04s =====

```

y el output .xml

```

[andresquirozvaldovinos@Andress-MacBook-Pro python_coverage % cat coverage.xml ] 
<?xml version="1.0" ?>
<coverage version="7.10.7" timestamp="1763012072019" lines-valid="19" lines-covered="19" line-rate="1"
branches-covered="0" branches-valid="0" branch-rate="0" complexity="0">
    <!-- Generated by coverage.py: https://coverage.readthedocs.io/en/7.10.7 -->
    <!-- Based on https://raw.githubusercontent.com/cobertura/web/master/htdocs/xml/coverage-04.
dtd -->
    <sources>
        <source>/Users/andresquirozvaldovinos/Documents/UNIVERSIDAD/COVERAGE/python_coverage
</source>
    </sources>
    <packages>
        <package name=". " line-rate="1" branch-rate="0" complexity="0">
            <classes>
                <class name="calculator.py" filename="calculator.py" complexity="0"
line-rate="1" branch-rate="0">
                    <methods/>
                    <lines>
                        <line number="1" hits="1"/>
                        <line number="2" hits="1"/>
                        <line number="3" hits="1"/>
                        <line number="5" hits="1"/>
                        <line number="6" hits="1"/>
                        <line number="8" hits="1"/>
                        <line number="9" hits="1"/>
                        <line number="11" hits="1"/>
                        <line number="12" hits="1"/>
                        <line number="13" hits="1"/>
                        <line number="14" hits="1"/>
                        <line number="16" hits="1"/>
                        <line number="17" hits="1"/>
                        <line number="19" hits="1"/>
                        <line number="20" hits="1"/>
                        <line number="21" hits="1"/>
                        <line number="22" hits="1"/>
                        <line number="23" hits="1"/>
                        <line number="25" hits="1"/>
                    </lines>
                </class>
            </classes>
        </package>
    </packages>
</coverage>
andresquirozvaldovinos@Andress-MacBook-Pro python_coverage %

```

Conclusión

Sin duda alguna manejar pruebas en python es mucho más sencillo que Java, esto hace que el flujo de trabajo sea más eficiente para los desarrolladores, sin embargo cabe destacar la poca eficiencia del lenguaje mismo impactando en las pruebas y el rendimiento general de las apps. Fuera de eso recalco lo mismo que la tarea pasa, poder probar nuestro código es una de las tareas más importantes para asegurar la entrega segura de una pieza de software al cliente, y que mejor que pruebas eficientes como con ambas librerías analizadas en esta tarea.