

# Mock

December 12, 2020

## 1 Mock

Mock es simular el comportamiento de algo. ¿Por qué vamos a necesitar simular? Por ejemplo que para probar una parte de código necesitamos tener funcionando un entorno que es difícil hacer el set up de eso; o si necesitamos usar sistemas externos, o si una operación demora mucho tiempo.

Mock en unittest -> <https://docs.python.org/3/library/unittest.mock.html>

Mock en pytest -> <https://pypi.org/project/pytest-mock/> (pip install pytest-mock)

```
[ ]: import time

def compute(x):
    response = expensive_api_call()
    return response + x

def expensive_api_call():
    time.sleep(1000)
    return 666

def test_compute():
    expected = 124
    actual = compute(1)
    assert expected == actual
```

El objeto más importante que nos brinda mock (de Python) es el objeto MagicMock. Con esto se puede simular:

- simular una constante
- simular un objeto con atributos
- simular una función

### Simular una constante

```
[ ]: # src/calcs.py
CONSTANT = 4

def func():
```

```

    return CONSTANT * 2

# tests/test_calcs.py
from src import calcs
from src.calcs import func

def test_mock_constant(mocker):
    mocker.patch.object(calcs, 'CONSTANT', 10)
    assert 20 == func()

```

```

[3]: def test_slow_function_mocked_api_call(mocker):
    mocker.patch(
        'src.main.api_call',
        return_value=5
    )

    expected = "Transaction sucessfull"
    actual = slow_function()
    assert expected == actual

```

```

[ ]: class Dataset:
    def __init__(self):
        self.data = None

    def load_data(self):
        time.sleep(4)
        self.data = 'slow data'

def test_mocking_class_method(mocker):
    expected = 'xyz'

    def mock_load(self):
        return 'xyz'

    mocker.patch(
        'mock_examples.main.Dataset.load_data',
        mock_load
    )
    actual = slow_dataset()
    assert expected == actual

```

### 1.3 Simular un objeto tipo diccionario

```
[5]: # src/calcs.py
foo = {'hola': 2}

def func_dict():
    return foo['hola']

# tests/test_calcs.py
def test_mock_dict(mock):
    mock.patch.dict('src.calcs.foo', {'chau': 10})
    assert 2 == func_dict()
```

## 2 Ciclo de desarrollo conducido por pruebas TDD