

HERRAMIENTAS DE TESTING EN PYTHON

¡BIENVENIDOS!

Soy Juan Benito Pacheco

Tech Lead

 <https://www.linkedin.com/in/juan-benito-pacheco-rubio/>

ÍNDICE

- **Introducción**
- **Unit Testing - Pytest**
- **Property Based Testing - Hypothesis**
- **Contract Testing - Schematessis**
- **Load Testing - Locust**
- **Práctica**

INTRODUCCIÓN

- El testing es una parte fundamental de cualquier aplicación, que permite validar el correcto funcionamiento de la misma de forma automática.

UNIT TESTING - PYTHON

El testing unitario se encarga de validar el comportamiento aislado de los componentes de nuestro código.

Pytest es una librería muy popular que permite crear tests unitarios de forma muy sencilla.

```
# test_sum.py
@pytest.mark.parametrize("test_input,expected", [(2,3,5), (3,4,7), (5,6,11)])
def test_sum_is_working(a, b, result):
    assert sum(a,b) == result
```

PROPERTY BASED TESTING - HYPOTHESIS

Se basa en la creación de tests válidos para cualquier combinación de parámetros que se le pueda pasar a la función de testing.

```
@given(op1=st.integers(), op2=st.integers())  
def test_calculate_is_working(op1: int, op2: int):  
    assert sum([op1, op2]) == op1 + op2
```

CONTRACT TESTING - SCHEMATESIS

Se basa en la utilización de API contract para comprobar que la implementación del mismo se realiza de forma correcta.

```
schemathesis run http://localhost:8000/openapi.json
```

LOAD TESTING - LOCUST

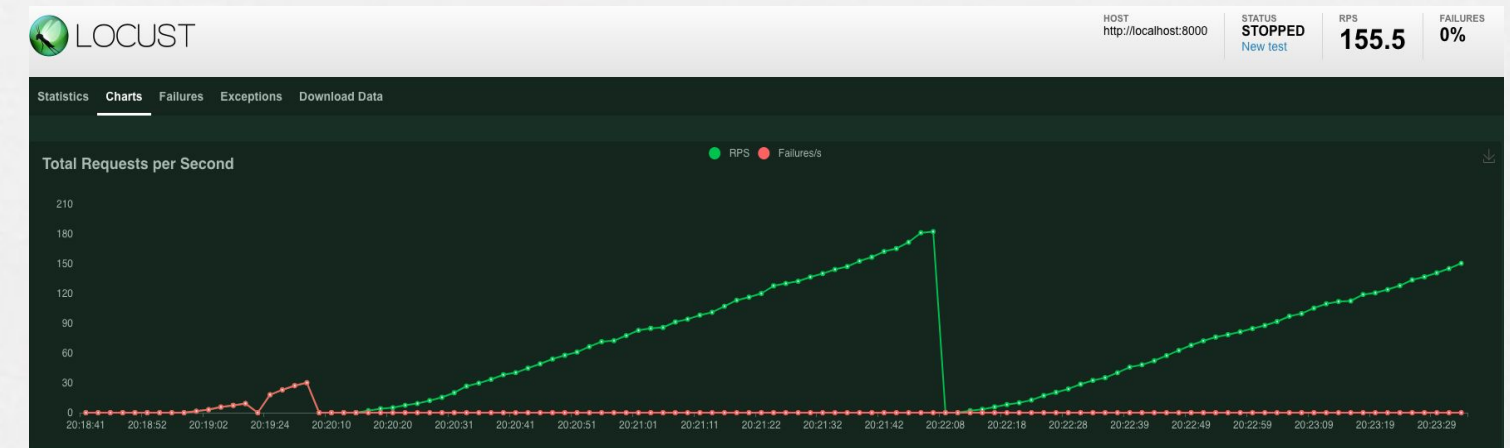
Se basa en la generación de varios usuarios virtuales para generar carga sobre nuestra aplicación, y comprobar cómo se comporta.

LOAD TESTING - LOCUST

```
class QuickstartUser(HttpUser):
    wait_time = between(1, 2.5)

    @task
    def hello_world(self):
        self.client.get("/hello_world")

    def on_start(self):
        self.client.post("/login", json={"username": "foo",
        "password": "bar"})
```



```
[2021-08-07 20:26:05,425] MacBook-Pro-de-Juan.local/INFO/locust.main: Starting Locust 2.0.0
```

Name	# reqs	# fails	Avg	Min	Max	Median	req/s	failures/s
Aggregated	0	0(0.00%)	0	0	0	0	0.00	0.00

```
[2021-08-07 20:26:05,426] MacBook-Pro-de-Juan.local/INFO/locust.runners: Ramping to 100 users at a rate of 10.00 per second
```

Name	# reqs	# fails	Avg	Min	Max	Median	req/s	failures/s
POST /maths/calculate	58	0(0.00%)	12	2	34	6	0.00	0.00
POST /maths/expression	13	0(0.00%)	9	3	33	6	0.00	0.00
Aggregated	71	0(0.00%)	11	2	34	6	0.00	0.00

PRÁCTICA

Repositorio:

https://github.com/juanbenitopr/taller_testing_tools_python