

**Universidad de Ingenieria y Tecnologia**  
**Departamento de Ciencias de la Computación**

**Ingeniería de Software I**

**Lab VIII: Testing The Basics**  
**2024 - II**

<b>TEAM</b>	Los del fondo
<b>INTEGRANTES</b>	<ul style="list-style-type: none"><li>- José Rodriguez</li><li>- Yesenia Valero</li><li>- Jose Bojorquez</li><li>- Piero Guerrero</li><li>- Aarón Timaná</li><li>- Gonzalo Suárez Torres</li></ul>
<b>Puntos EC</b>	2 ptos

**Background:**

Unit Test

Integration Test

**CASO DE ESTUDIO (level: basic):**

**1. Flzbuzz (UNIT TEST):**

Baje el repo <https://gist.github.com/jaysonrowe/1592775#file-fizzbuzz-py-L3>

- **¿Cuál es el happy path ? ¿Hay algún edge case?**

El happy path del programa se encuentra dentro de la función fizzbuzz, ya que allí se puede analizar el comportamiento que debería satisfacer las peticiones. Uno de los edge case que podemos encontrar es la inserción de un valor flotante en la función principal, aunque se espere un número entero en su lugar. Otro de ellos, sería la inserción del número 0 que es admitido en los 3 casos posibles.

- **Agregue los unit test necesarios para un coverage del 100%**
- **Mostrar coverage:**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
aaron@MacBook-Pro-de-Aaron 1592775 % python3 -m coverage report -m

Name                Stmts   Miss  Cover   Missing
-----
fizzbuzz.py           9      0   100%
test_fizzbuzz.py      65      1    98%    115
-----
TOTAL                  74      1    99%

aaron@MacBook-Pro-de-Aaron 1592775 % python3 -m coverage run -m unittest discover

1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
....F.....
```

<https://github.com/utecsw20242/los-del-fondo/tree/main/lab10/p1>

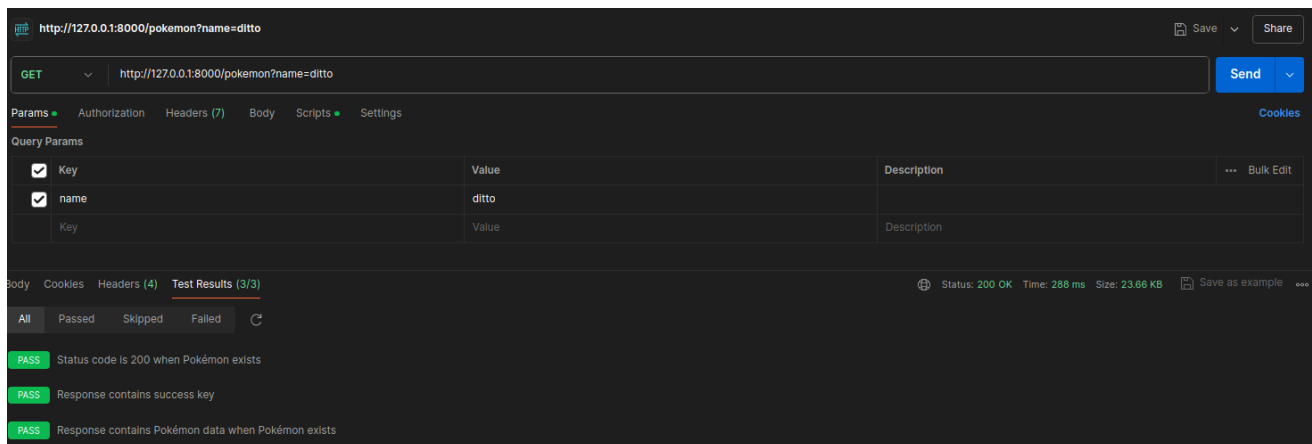
## 2. Pokedex (INTEGRATION TEST + MOCK )

- Haga una api que permita buscar pokemones por nombres de la siguiente API publica: <https://pokeapi.co/> en el lenguaje de su preferencia.
- Elabore un postman que testee esta API. Subir el script a su repo.
  - ¿Qué pasa si su API devuelve 500 ? ¿Cuál es el json body del response ?
  - ¿Qué pasa si su API devuelve 400 ? ¿Cuál es el json body del response ?
  -
- Imagine ahora que los viernes PokeAPI no esta disponible por lo que su equipo de QA le sugiere usar Mockoon para implementar mocks: <https://mockoon.com/>.
  - Mockear 200, 400 y 500 como respuestas de este mock. En base a estas respuestas como responde su API ?

link: <https://github.com/utecsw20242/los-del-fondo/tree/main/lab10/p2>

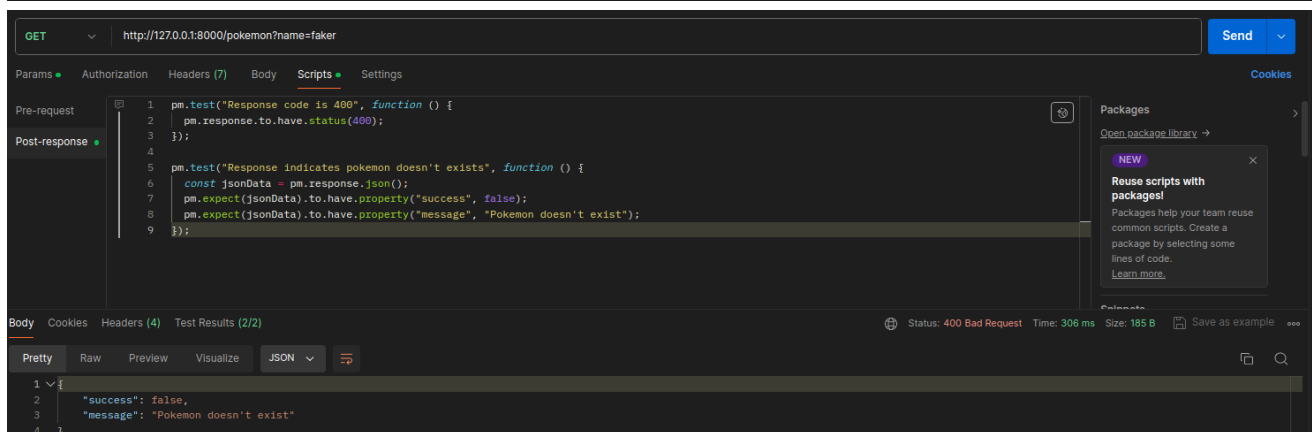
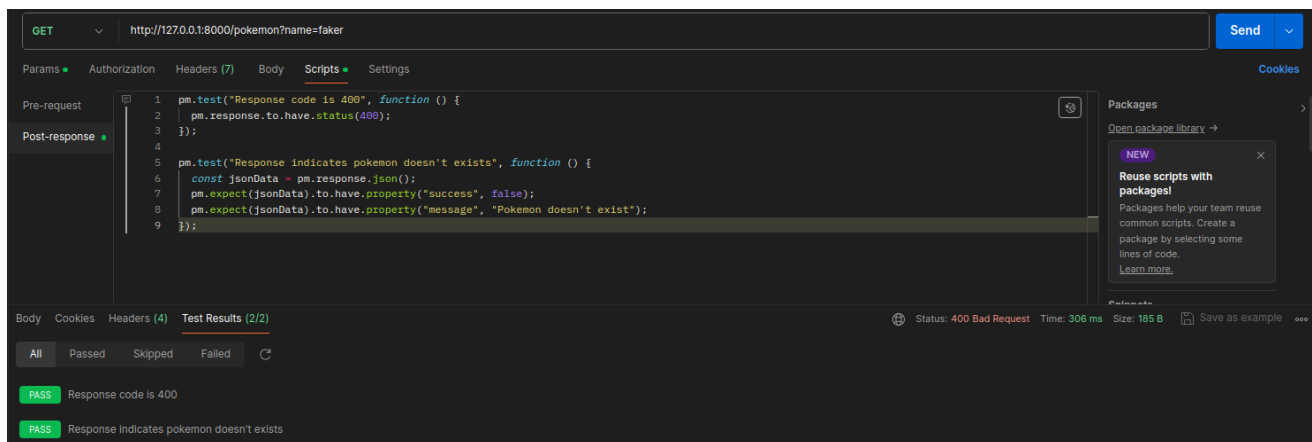
Camino exitoso, 200:

El json regresado tiene 2 valores, success y pokemon. Success es verdad y pokemon tiene todos los datos del pokemon.



Camino faillido, 400:

El json regresado tiene 2 valores, success y pokemon. Success es falso y message es Pokemon doesn't exist.



Camino faillido, 500:

El json regresado tiene 2 valores, success y pokemon. Success es falso y message es Endpoint is not implementedMockoo.

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/poke`. The `Scripts` tab is active, displaying the following code:

```
1 pm.test("Response code is 500", function () {
2   pm.response.to.have.status(500);
3 });
4
5 pm.test("Response indicates pokemon doesn't exists", function () {
6   const jsonData = pm.response.json();
7   pm.expect(jsonData).to.have.property("success", false);
8   pm.expect(jsonData).to.have.property(
9     "message",
10    "Endpoint is not implemented.",
11  );
12 });
```

The `Test Results (2/2)` section shows two passed tests:

- PASS** Response code is 500
- PASS** Response indicates pokemon doesn't exists

At the bottom, the status bar indicates: `Status: 500 Internal Server Error`, `Time: 3 ms`, `Size: 202 B`, and a `Save as example` button.

Mockoon: La API sigue funcionando, solo que ahora funciona exclusivamente cuando el name dado es el id, previamente funcionaba en ambos casos, o sea si lo dado el nombre o el id del pokemon. Esto se debe a que el playground de Mockoon tiene al endpoint de pokemon funcionando exclusivamente con el id.