

## Unittest

Es usado para ejecutar una prueba o un conjunto de pruebas.

### A considerar:

- En Unittest, todas las pruebas deben empezar con test
- Cuidado con mayusculas y minusculas
- Unittest siempre con minúscula

Practica aceptada, programa independiente.

```
if __name__=="__main__":  
    unittest.main*()
```

**Test Case.** - unidad mínima en Unittest, puede tener una o más validaciones: Hereda de unittest TestCase class.

**Test Fixture.** - es la preparación y la terminación de todas las pruebas.

setUp

tearDown

**Test Suite.** - una colección o grupo de casos de pruebas (test Cases)

**Test Runner class.** - la parte de unittest que ejecuta las pruebas

**Test report.** - reporte de los resultados de las pruebas

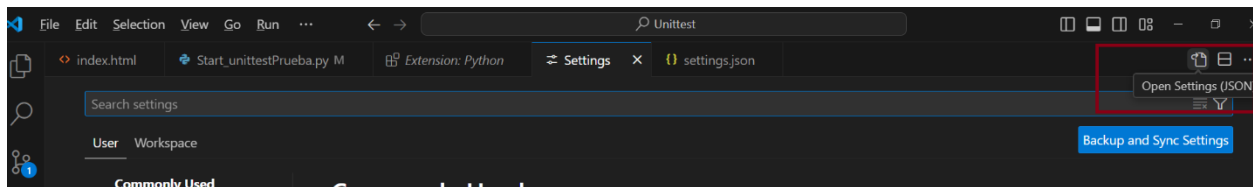
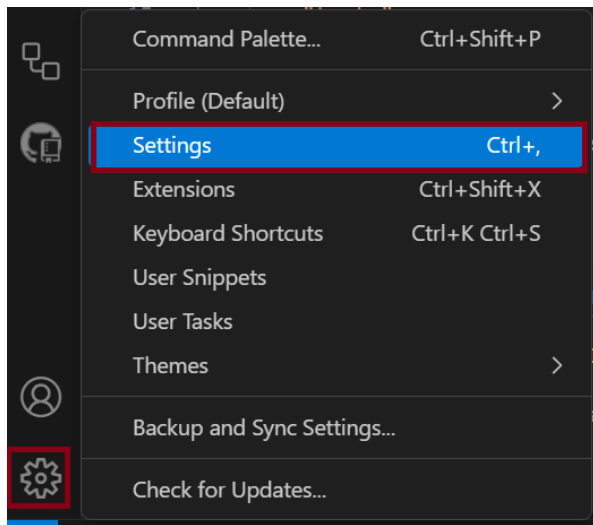
### Herramientas:

Selenium WebDriver, Nose y Pytest

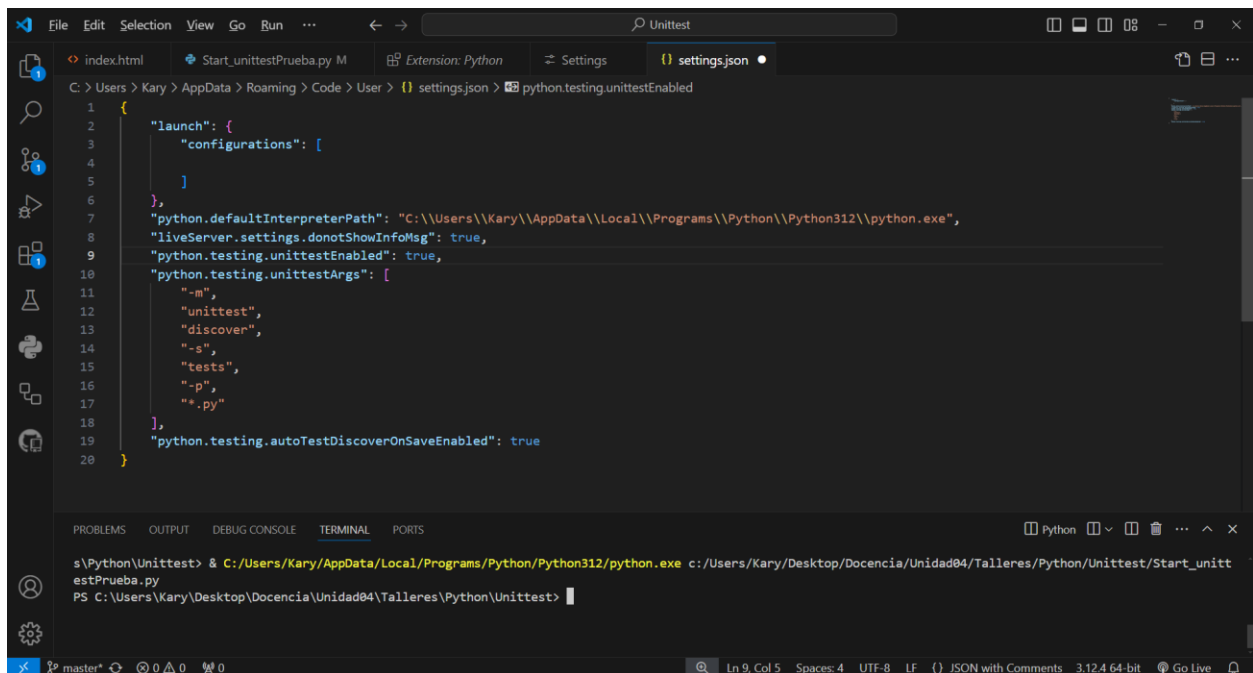
## Laboratorio

1. Crear un repositorio remoto (PaginaPruebas)
2. Clonar el repositorio remoto
3. Desplegar la página **PaginaPruebas** en Page de GitHubActions
4. Habilitar el marco de pruebas en VisualStudioCode

Visual Studio Code -> Engranaje -> Settings -> Open Settings (JSON)



- `python.testing.unittestEnabled`: Habilita el marco de pruebas unittest.
- `python.testing.unittestArgs`: Configura los argumentos para descubrir y ejecutar las pruebas. Aquí se especifica que las pruebas están en el directorio tests y tienen la extensión .py.
- `python.testing.autoTestDiscoverOnSaveEnabled`: Habilita el descubrimiento automático de pruebas al guardar archivos.



- Guardar y reiniciar Visual Studio Code

- Validar a la izquierda el ícono de laboratorio
- Crear directorio **tests** dentro del proyecto

your\_project/

├─ tests/

| └─ test\_start\_unittestPrueba.py

└─ your\_code.py

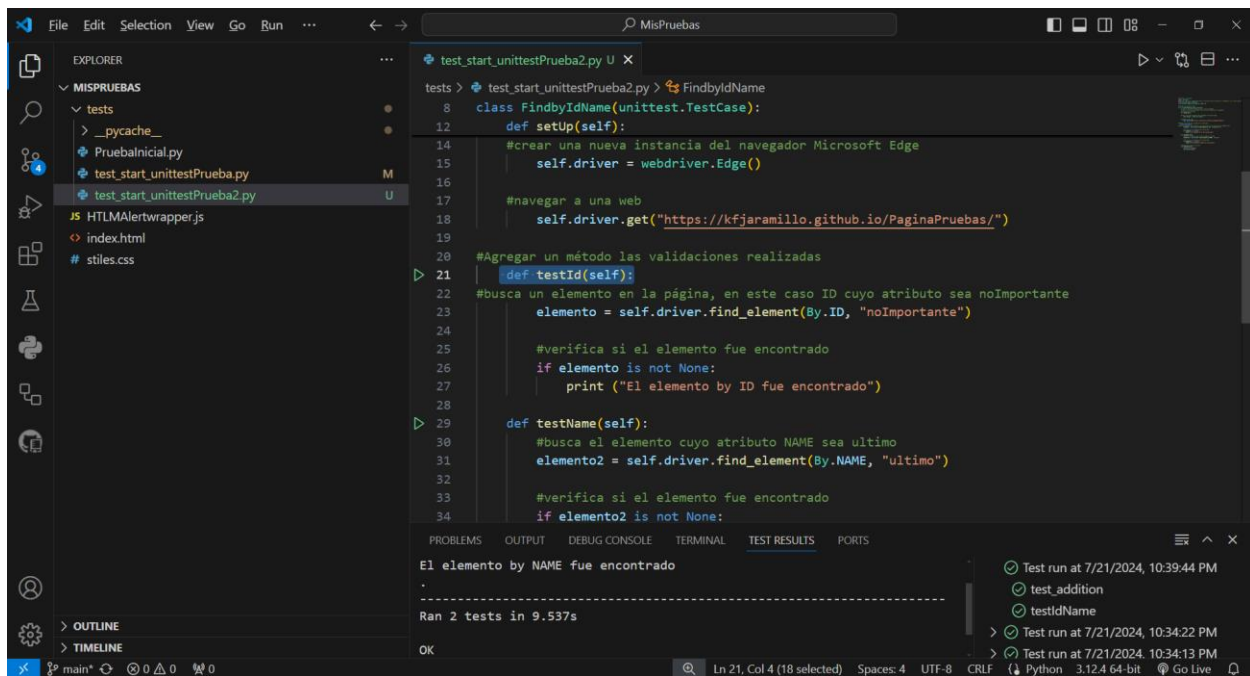
5. Crear caso de prueba **test\_start\_unittestPrueba.py** dentro del directorio **tests**

```

File Edit Selection View Go Run ... Search
Start_unittestPrueba.py X
C:\Users\Kary\Desktop\Docencia\Unidad04\Talleres\Python\Unittest\Start_unittestPrueba.py > ...
1 #convertir a unittest
2 import unittest
3 #importar librerías: webdriver(librería principal para controlar el navegador) y by (localización de elemento)
4 from selenium import webdriver
5 from selenium.webdriver.common.by import By
6
7 #heredar de unittest clase
8 class FindbyIdName(unittest.TestCase):
9     # crear 2 funciones Python inicio y fin de pruebas
10    # metodo de inicio de prueba. Usar driver de Selenium en navegador
11    def setUp(self):
12
13        #crear una nueva instancia del navegador Microsoft Edge
14        self.driver = webdriver.Edge()
15
16        #navegar a una web
17        self.driver.get("https://kfjaramillo.github.io/PaginaPruebas/")
18
19    #Agregar un método las validaciones realizadas
20    def testIdName(self):
21        #busca un elemento en la página, en este caso ID cuyo atributo sea noImportante
22        elemento = self.driver.find_element(By.ID, "noImportante")
23
  
```

6. Ejecutar pruebas
  - a. Dividir pruebas y obtener reportes por cada una

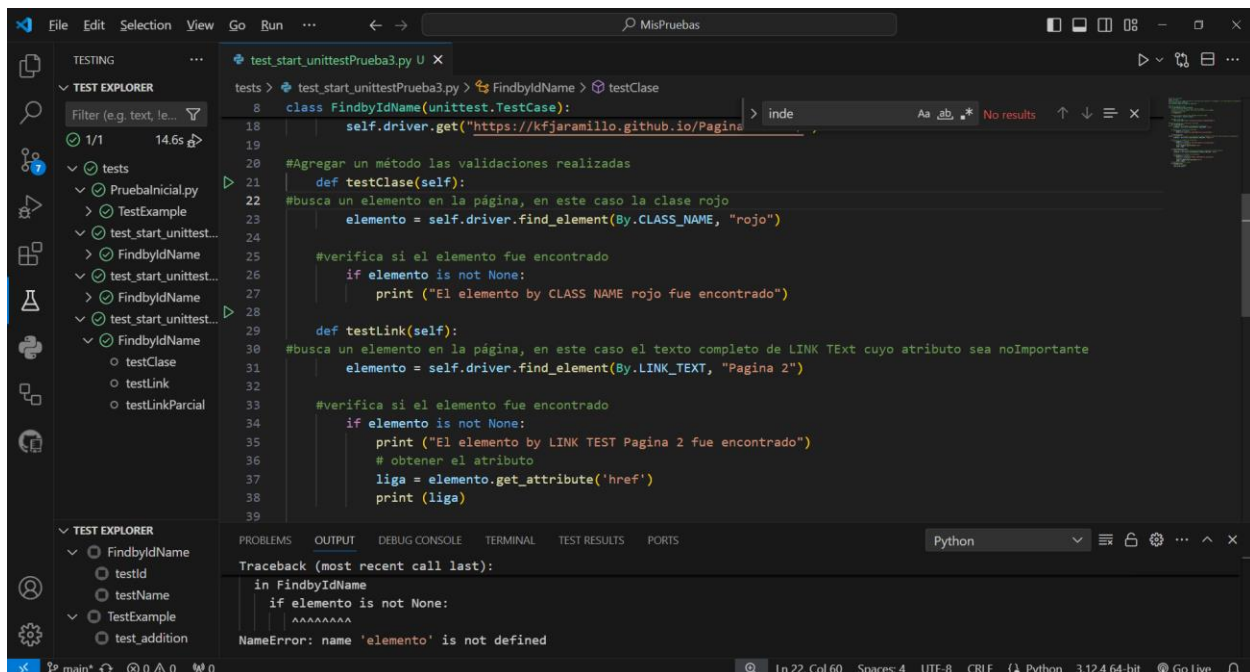
Modificar el archivo test\_start\_unittestPrueba.py para que se ejecute en 2 partes y guardarlo como **test\_start\_unittestPrueba2.py**

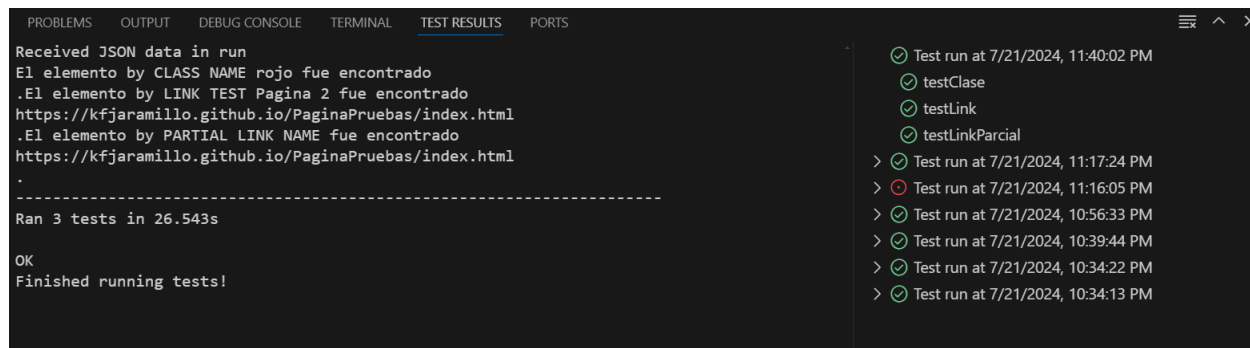


i. Validar como se ejecutan las pruebas en 2 partes.

b. Realizar una prueba de búsqueda por clase, por link y por link parcial

Modificar el archivo test\_start\_unittestPrueba2.py para que se ejecute en 2 partes y guardarlo como test\_start\_unittestPrueba3.py





i. Validar como se ejecutan las pruebas.

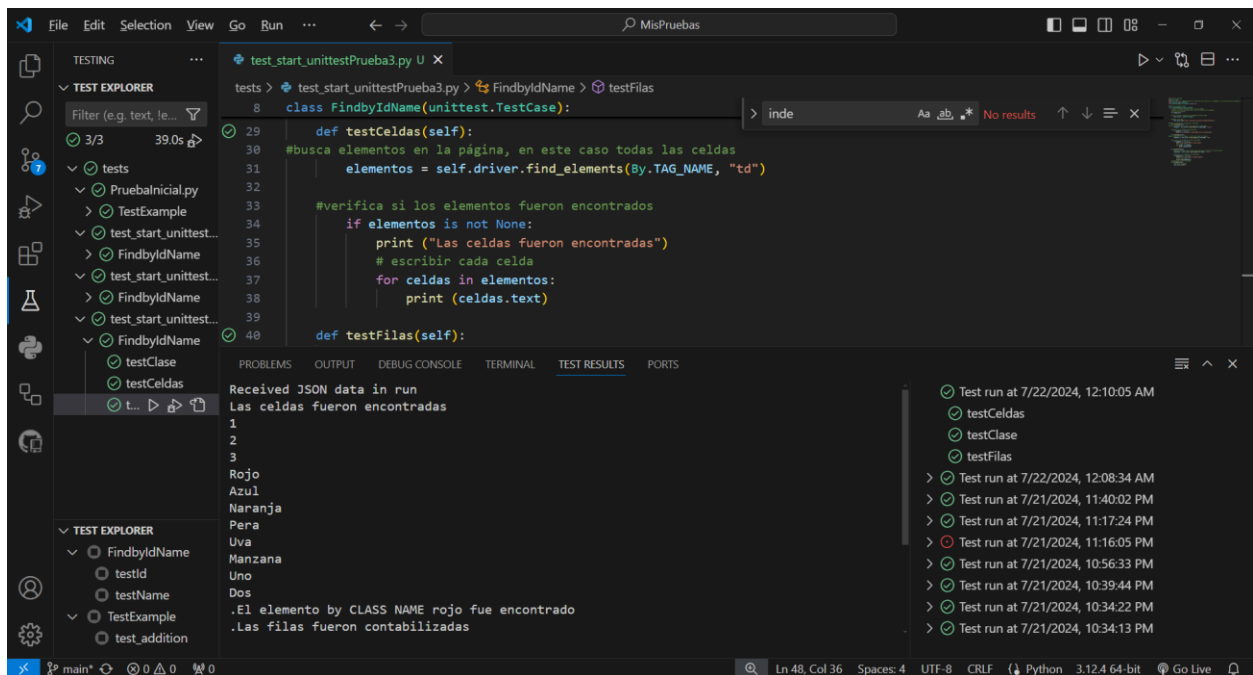
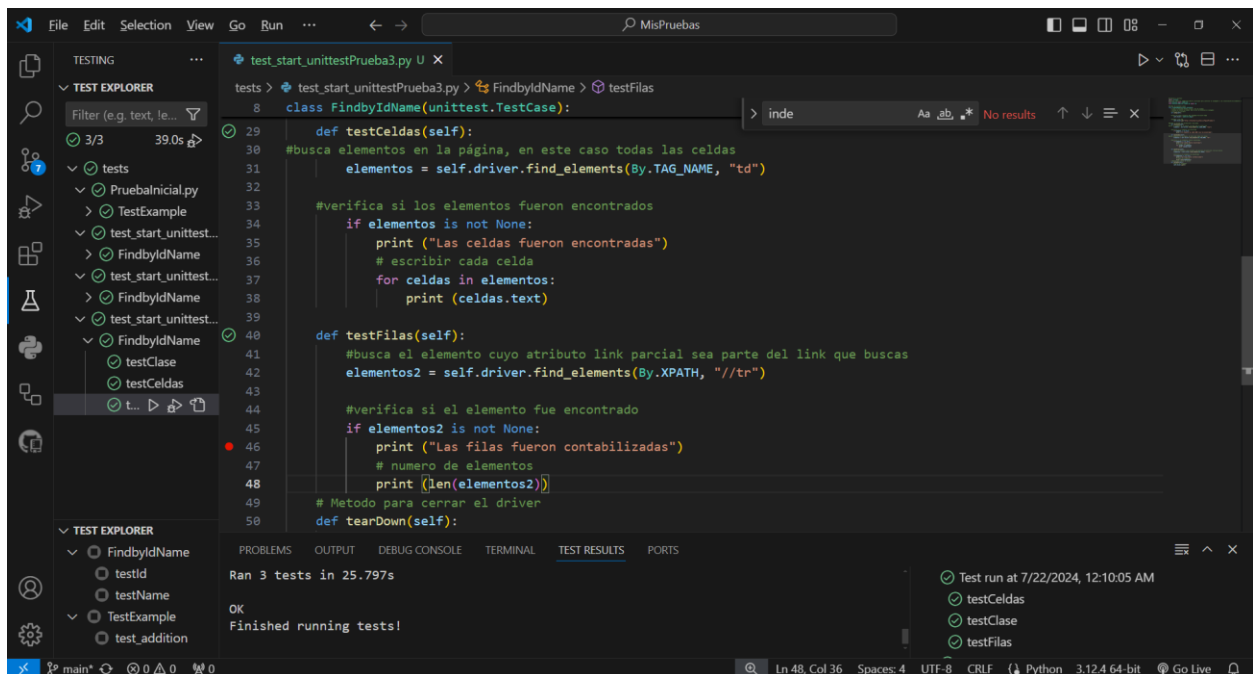
#### Nota:

Los métodos de búsqueda en Selenium, como `find_element_by_class_name`, son herramientas esenciales para localizar y trabajar con elementos en una página web durante las pruebas automatizadas. Estos métodos permiten verificar la presencia de elementos, interactuar con ellos y validar su contenido, lo que es crucial para garantizar que la aplicación web funcione correctamente y proporcione una buena experiencia de usuario. Al usar estos métodos en tus scripts de prueba, puedes automatizar tareas repetitivas y mejorar la eficiencia y cobertura de tus pruebas.

#### c. Encontrar varios elementos `find_elements`:

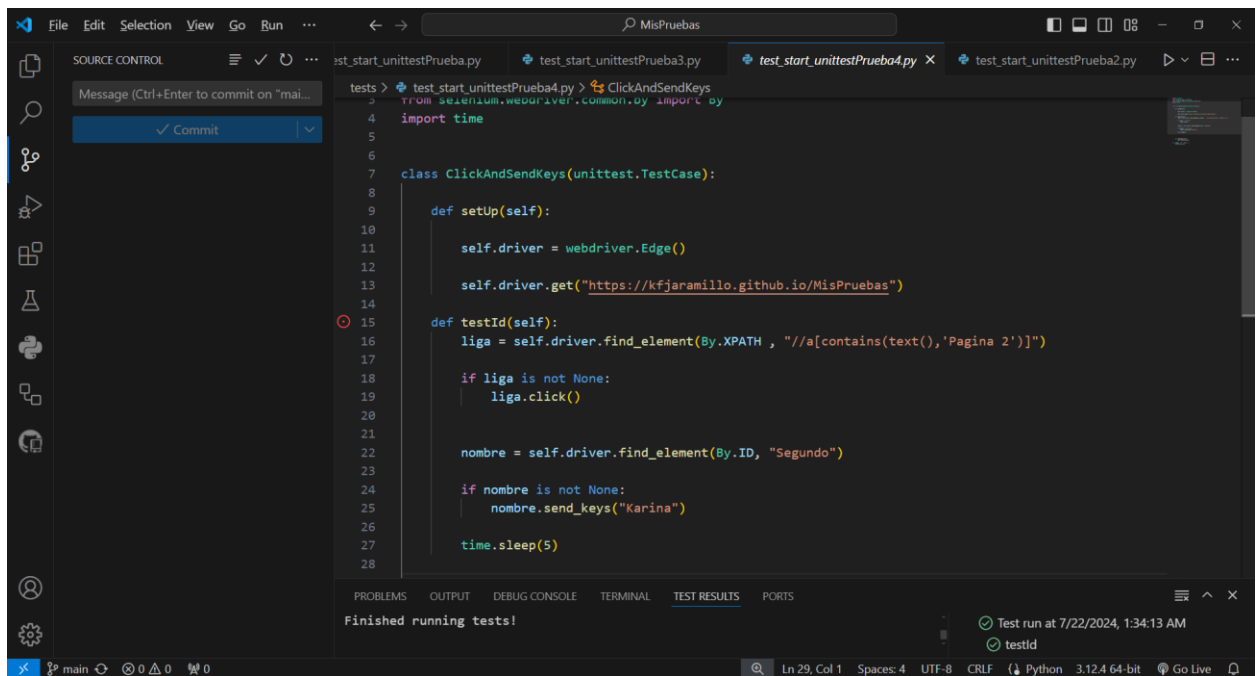
- Name
- Class name
- Css selector
- ID
- Link test
- Partial link test
- Tag name
- Xpath

La diferencia es que devuelve una lista, regresa el texto de cada una de las celdas y cuenta cuantas filas tenemos



#### d. Pruebas con Acciones

- Al dar click en la página principal y escribir en un cuadro de texto



Validar la ejecución

