Actividad Clase 7: **Mejora continua y herramientas CI/CD**

Objetivo:

Aplicar conceptos de integración y entrega continua utilizando GitHub Actions y simular un flujo básico de CI/CD.

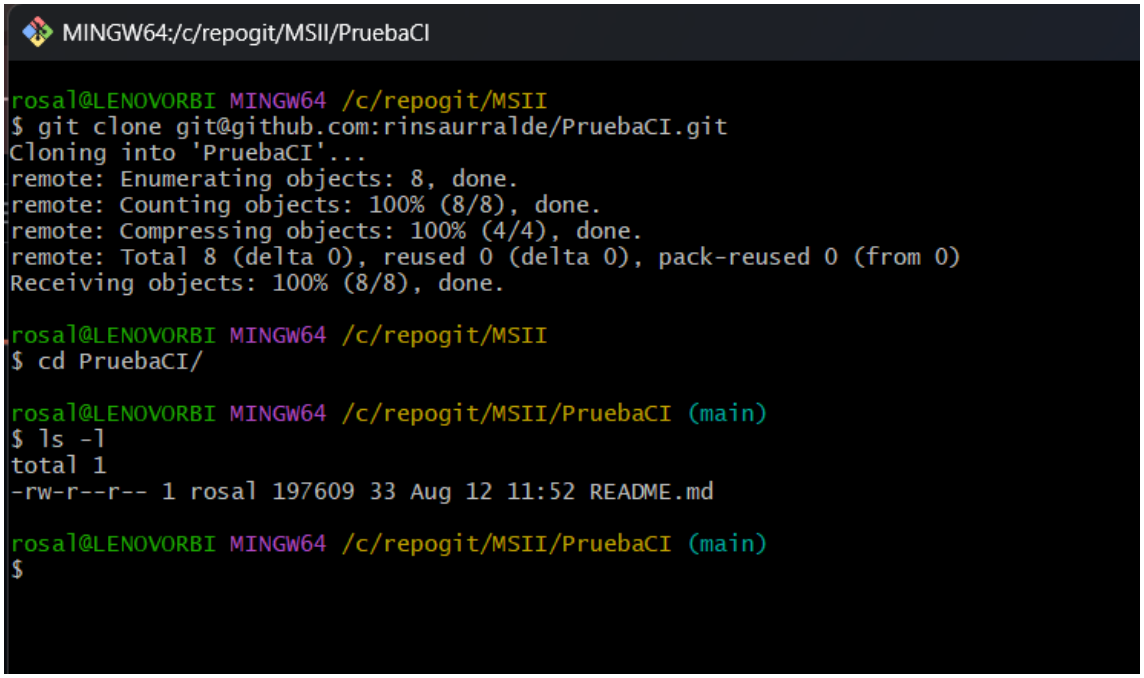
Ejercicio 1: Simulación de un pipeline básico con GitHub Actions

Requisitos previos:

- Tener cuenta en GitHub
- Tener instalado Python 3
- Tener un repositorio nuevo en GitHub

Pasos:

1. Crear un repositorio llamado PruebaCI_apellido.
2. Clonar el repositorio localmente.



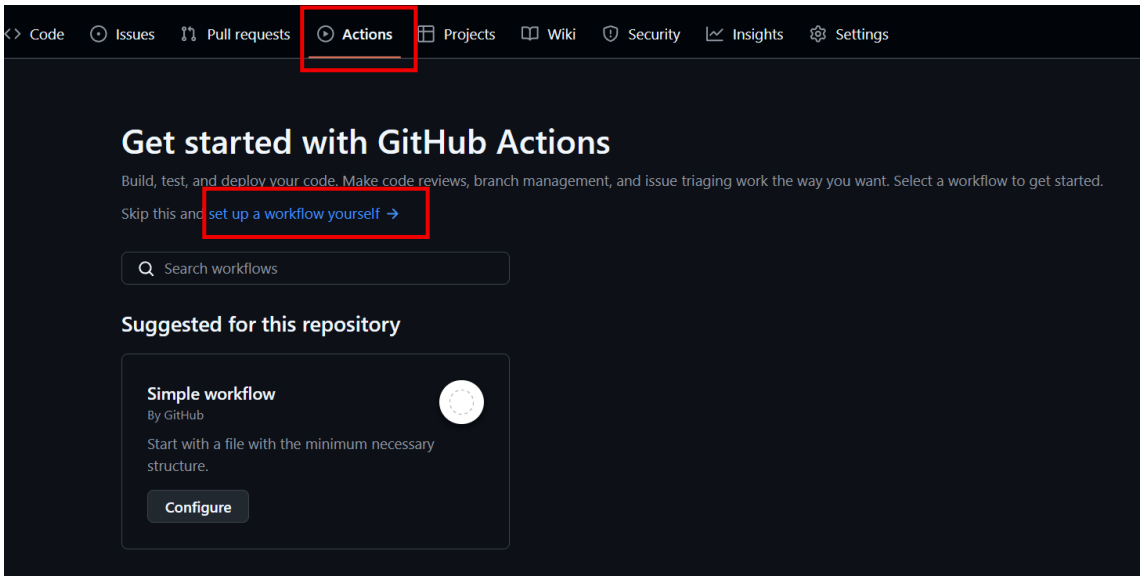
3. Dentro del repositorio local, crear un archivo app.py con una función simple en Python:

```
# app.py
def suma(a, b):
    return a + b
```

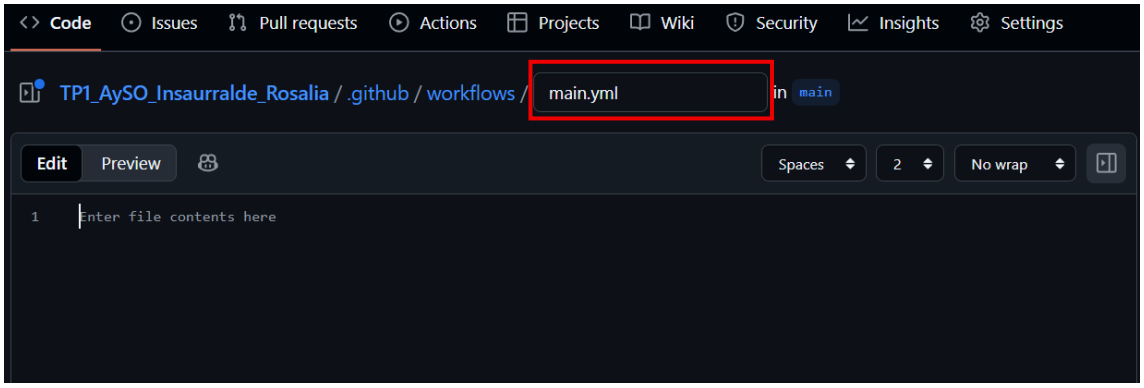
4. **Crear un archivo de prueba test_app.py:**

```
# test_app.py
from app import suma

def test_suma():
    assert suma(2, 3) == 5
```

5. **Crear la estructura del workflow:**
   En el repositorio remoto, agregar el archivo. Seleccionar "Actions" :

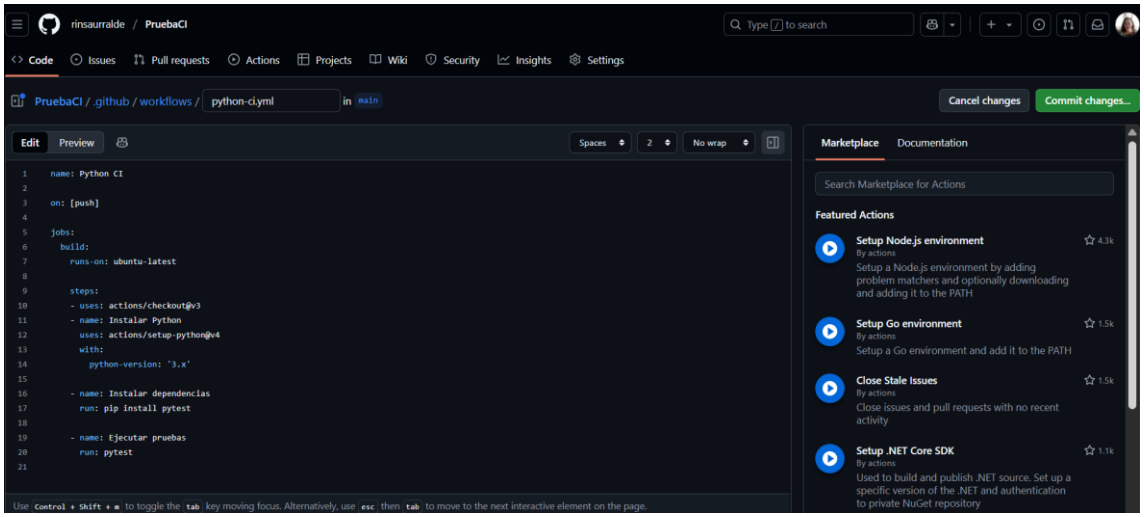


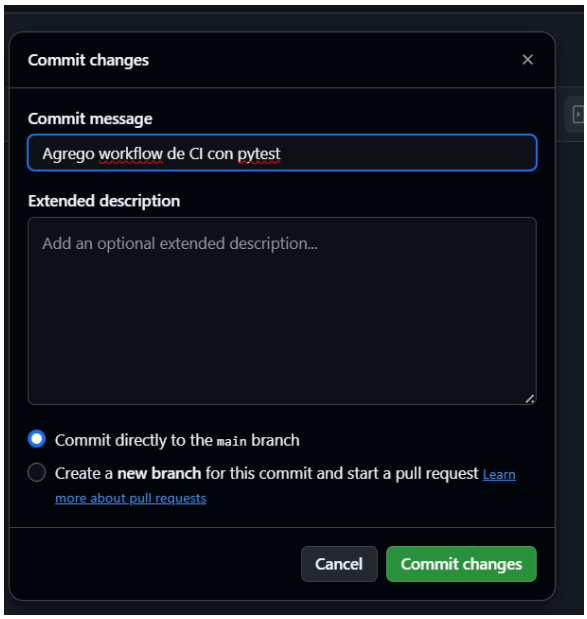
.github/workflows/python-ci.yml



6. **Copiar el siguiente contenido en python-ci.yml:**

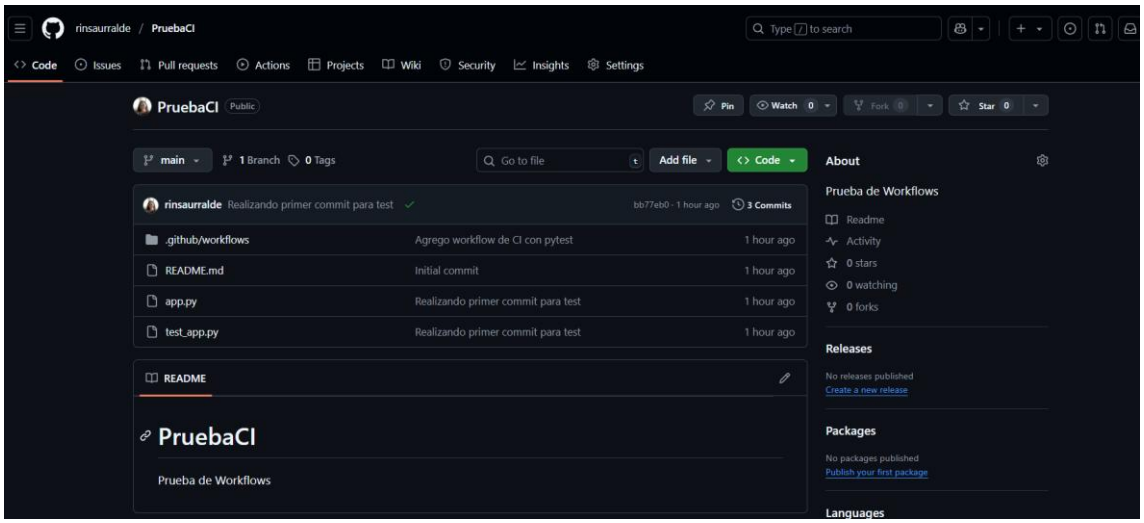
```
name: Python CI

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
    - name: Instalar Python
      uses: actions/setup-python@v4
      with:
        python-version: '3.x'

    - name: Instalar dependencias
      run: pip install pytest

    - name: Ejecutar pruebas
      run: pytest
```

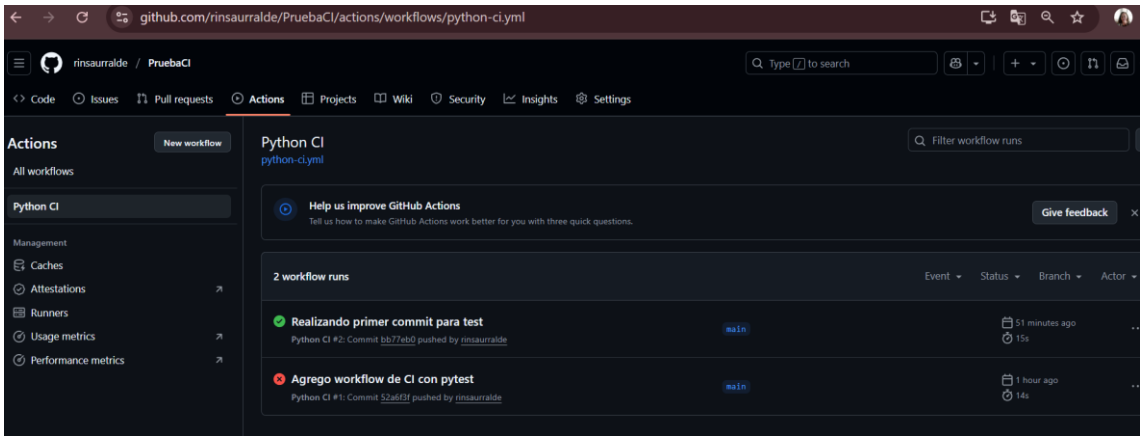




7. Hacer commit y push. Ir a la pestaña Actions del repositorio y observar la ejecución automática del pipeline.

```
rosal@LENOVORBI MINGW64 /c/repogit/MSII/PruebaCI (main)
$ ls -l
total 3
-rw-r--r-- 1 rosal 197609 33 Aug 12 11:52 README.md
-rw-r--r-- 1 rosal 197609 33 Aug 12 11:58 app.py
-rw-r--r-- 1 rosal 197609 67 Aug 12 12:00 test_app.py

rosal@LENOVORBI MINGW64 /c/repogit/MSII/PruebaCI (main)
$ git add .
warning: in the working copy of 'app.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'test_app.py', LF will be replaced by CRLF the next time Git touches it

rosal@LENOVORBI MINGW64 /c/repogit/MSII/PruebaCI (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   app.py
        new file:   test_app.py


rosal@LENOVORBI MINGW64 /c/repogit/MSII/PruebaCI (main)
$ git commit  -m "Realizando primer commit para test"
[main bb77eb0] Realizando primer commit para test
 2 files changed, 7 insertions(+)
 create mode 100644 app.py
 create mode 100644 test_app.py

rosal@LENOVORBI MINGW64 /c/repogit/MSII/PruebaCI (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 452 bytes | 226.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:rinsaurralde/PruebaCI.git
   52a6f3f..bb77eb0  main -> main

rosal@LENOVORBI MINGW64 /c/repogit/MSII/PruebaCI (main)
$
```





Ejercicio 2: Análisis de mejora continua

1. Analizar qué pasaría si la función tuviera un error.
2. Modificar la función suma() para que devuelva incorrectamente a - b y observar cómo falla el pipeline.
3. Corregir el error y volver a hacer push.