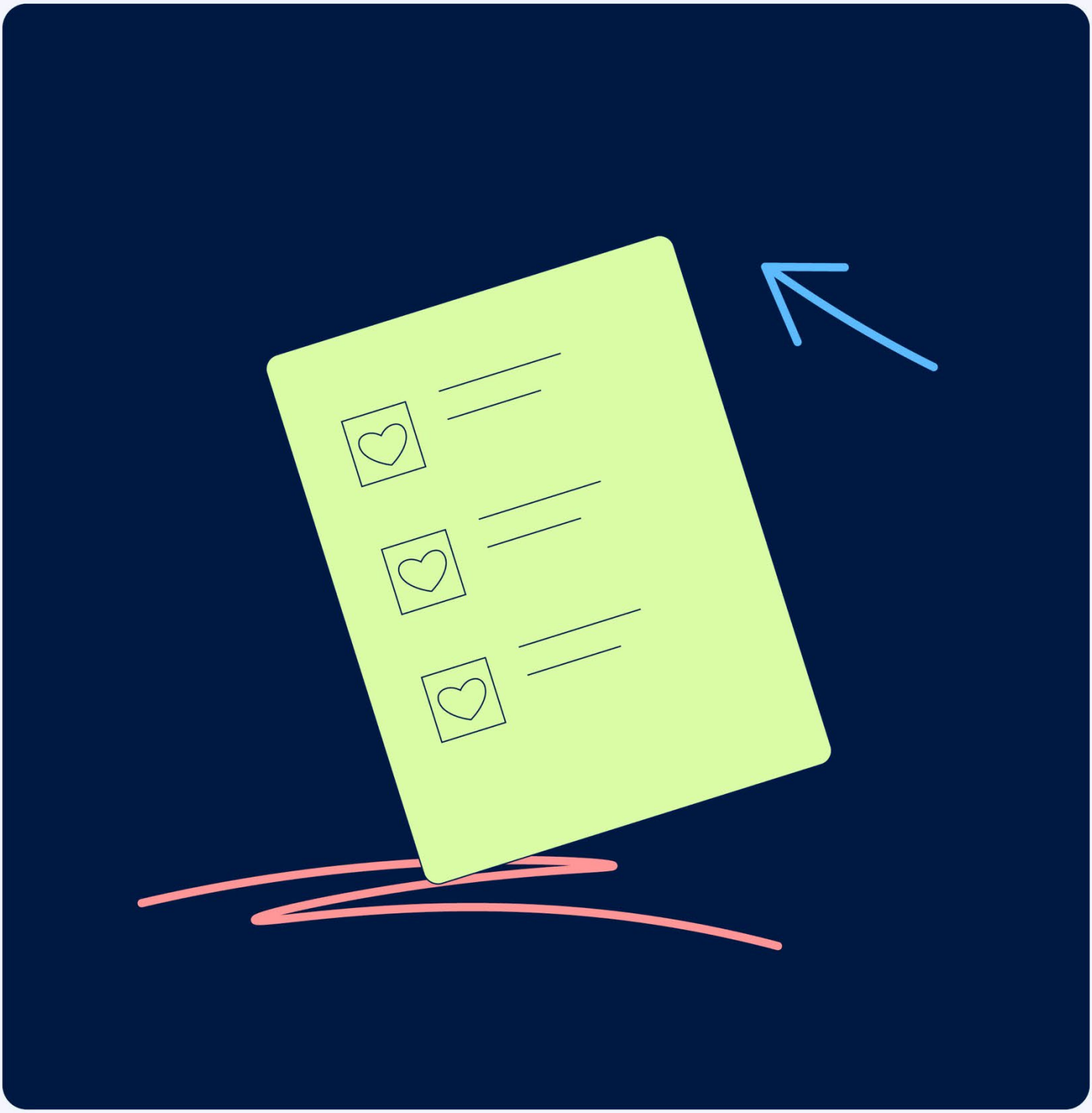

Problema Lab 05

Core Desarrollo

CREA Y DESARROLLA UNA API

La resolución de este ejercicio parte de un script básico (lo encontrarás en este mismo documento) de Python que actualmente solo realiza la operación de suma.

Tu tarea consistirá en expandir dicho script para que incluya las cuatro operaciones matemáticas básicas: suma, resta, multiplicación y división. Adicionalmente, es esencial que sepas y puedas garantizar que todas las funciones implementadas operan de manera correcta, por lo que se deben crear las pruebas unitarias pertinentes para comprobar cada operación. Por último, el mencionado proceso de prueba se deberá automatizar utilizando la plataforma GitHub Actions.

Objetivos de este ejercicio

1. Aplicar los conceptos básicos de programación en Python con el objetivo de expandir un script existente e implementar funciones adicionales.

2. Definir distintas pruebas unitarias, demostrar que se dispone de la habilidad necesaria para poder analizar diferentes escenarios y casos de uso, identificar posibles fallos y, por último, asegurar que cada función del script maneja de forma adecuada las distintas entradas generando así resultados esperados.

3. Implementar GitHub Actions e integrar tanto los conocimientos de desarrollo como pruebas en Python con herramientas de CI/CD. La idea es crear un flujo de trabajo automatizado que garantice la calidad y la funcionalidad del código mediante el uso de pruebas automáticas, este debe operar de forma correcta cada vez que se introduzca un cambio en el repositorio.

Descripción de la actividad

Para desarrollar este ejercicio, deberás seguir las siguientes instrucciones que te enumeramos justo debajo de estas líneas.

Expansión del script:

- utiliza los siguientes scripts básicos que te compartimos a continuación como punto de partida;

suma.py

```
def sumar(a, b):
    return a + b


if __name__ == "__main__":
    print(sumar(5, 3))
```

test_suma.py

```
import unittest

from suma import sumar


class TestSumar(unittest.TestCase):


    def test_sumar(self):

        self.assertEqual(sumar(3, 2), 5)

        self.assertEqual(sumar(-1, 1), 0)

        self.assertEqual(sumar(-1, -1), -2)


if __name__ == '__main__':

    unittest.main()
```

- implementa funciones para la resta, multiplicación y división;

- considera y evalúa el siguiente escenario, dividir por cero, de manera adecuada para evitar errores.

Pruebas unitarias:

- basándote en el ejemplo de prueba unitaria para la función de suma, crea pruebas unitarias para cada una de las nuevas funciones implementadas;
- asegúrate de considerar diferentes escenarios, como números negativos, ceros y valores límite.

Automatización con GitHub Actions:

- crea un repositorio en GitHub y sube tu script;
- implementa un flujo de trabajo de GitHub Actions para ejecutar automáticamente tus pruebas unitarias cada vez que hagas un push al repositorio;
- asegúrate de que todas las pruebas pasen correctamente en GitHub Actions.

Formato de entrega

Envía el código de tu ejercicio, así como un vídeo donde expliques la estructura de tu código y la ejecución del workflow de GitHub actions.

Su utilizas Windows 10 u 11, puedes utilizar la barra de juegos para grabar tu vídeo. Más información en este enlace.

Criterios de corrección

A la hora de autoevaluar tu trabajo y darlo por superado debes comprobar y asegurarte de que has llevado a cabo de forma correcta los siguientes puntos que te compartimos a continuación.

- Correcta implementación de las funciones que tienen que ver con las operaciones matemáticas.
- Cobertura completa con pruebas unitarias, considerando diferentes escenarios.
- Configuración adecuada de GitHub Actions y ejecución exitosa de las pruebas dentro de la plataforma.
- Grabar un vídeo donde se muestre la ejecución del workflow de GitHub Actions.


Qualentum