



Tarefa1

Fibonacci

Daisy Rodriguez Costa

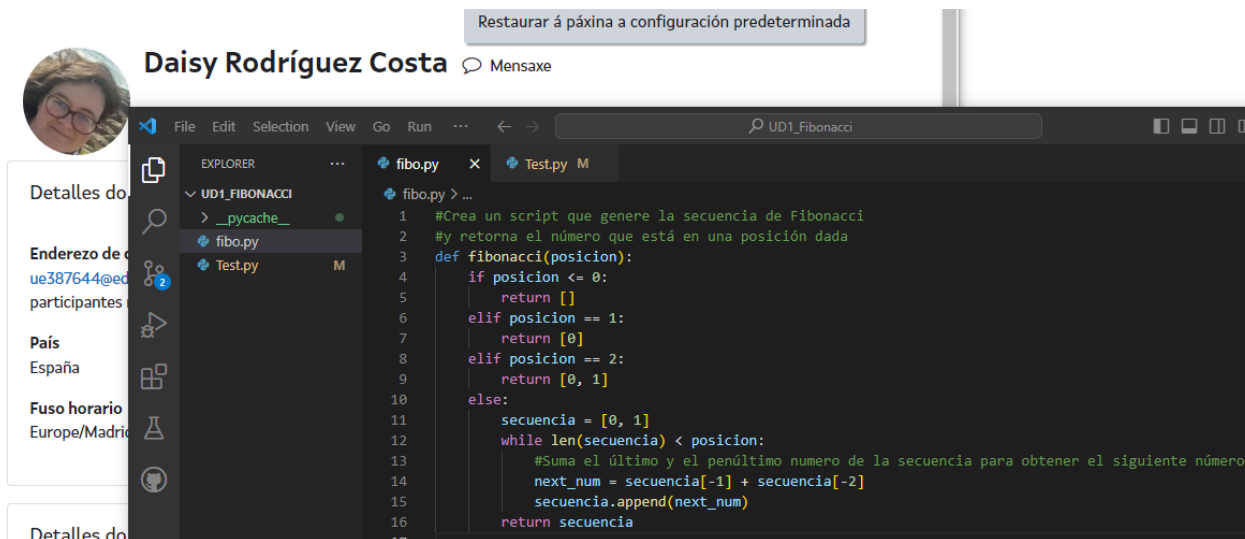


Índice

1.Script que crea a secuencia de Fibonacci.....	2
2. Creación do programa principal.....	2
2.1. Importar librerías	2
2.2. Creación clase.....	2
2.4. Función.....	2
3. Verificación de software y pregunta final	3
¿Que tipo de proba realizamos?	4

1. Script que crea a secuencia de Fibonacci

Creamos un script que se chama `fibonacci.py` no que xera a secuencia de Fibonacci hasta o número ou posición que nos deron. Por exemplo si nos pasaron o número 5, o script xerará a secuencia ata o quinto número, engadindo cada novo número a unha lista para despois sumar os dous últimos para obter os seguintes, ao principio hai que engadir os dous primeiros no código porque non poden ser calculados.



2. Creación do programa principal

Creamos o noso programa principal chamado `Test.py`.

2.1. Importar librerías

A min gústame empezar os scripts importando as librerías que necesito, no caso de que xa sabemos cales van ser, polo que importamos a clase **unittest** e do script **fibonacci** a función **Fibonacci**.

2.2. Creación clase

Creamos unha clase chamada **Test** que é de tipo **unittest.TestCase**.

2.4. Función

Dentro da nosa clase crearemos unha función que nos permitirá chamar a función da creación de secuencia para comprobar que funciona, neste caso chámase **test_secuencia_fibonacci**.

Primeiro pido por consola que se me de a **posición** que queremos verificar e tamén pedimos o **número** que creemos ou sabemos que debe de estar en esta posición, digo creemos porque podemos non saber que número vai nesa posición e facer como si quixéramos adiviñar.

Despois chamo a función **Fibonacci** pasándolle a posición para que me dea a secuencia e meto a lista na variable **resultado**.

Por último, chamo a función **assert.Equal** para que me compare o número da posición dada e o número que eu introducín para saber si é ese ou no. Para aclarar poño resultado -1 porque os arrays empezan a posición en 0 e non en 1, polo que si eu quero poñer que me dea o quinto número, teño que poñerlle 5-1 porque o quinto número corresponde a posición 4 o array.



Daisy Rodríguez Costa Mensaxe

```
def test_secuencia_fibonacci(self):
    # Pedimos por consola para poder hacer distintas pruebas sin modificar el código
    print("Qué posición desea comprobar?:")
    posicion = int(input())
    print("Dame el número que debería estar en esa posición:")
    numero = int(input())
    # Verificamos si el número de la posición indicada es igual al que hemos dado en la secuencia
    resultado = fibonacci(posicion)
    self.assertEqual(resultado[posicion-1], numero)

if __name__ == '__main__':
    unittest.main()
```

3. Verificación de software y pregunta final

Ao executar o script podemos ver que me pide por consola os datos e si os poño correctos terei o test aprobado.

```
PS C:\Users\Daisy\Desktop\Ciberseguridad\Posta_e_Produccion_Segura\Tareas\UD1_Fibonacci\UD1_Fibonacci> & C:\Users\Daisy\AppData\Local\Programs\Python\Python312\python.exe c:/Users/Daisy/Desktop/Ciberseguridad/Posta_e_Produccion_Segura/Tareas/UD1_Fibonacci/UD1_Fibonacci/Test.py
Qué posición desea comprobar?:
5
Dame el número que debería estar en esa posición:
3
-----
Ran 1 test in 7.288s
OK
PS C:\Users\Daisy\Desktop\Ciberseguridad\Posta_e_Produccion_Segura\Tareas\UD1_Fibonacci\UD1_Fibonacci>
```

En cambio, si poño un número que non corresponde esa posición, daramo o test como falido. E ademais dirame que o número esperado era outro e cal foi recibido.

```
PS C:\Users\Daisy\Desktop\Ciberseguridad\Posta_e_Produccion_Segura\Tareas\UD1_Fibonacci\UD1_Fibonacci> & C:\Users\Daisy\AppData\Local\Programs\Python\Python312\python.exe c:/Users/Daisy/Desktop/Ciberseguridad/Posta_e_Produccion_Segura/Tareas/UD1_Fibonacci/UD1_Fibonacci/Test.py
Qué posición desea comprobar?:
5
Dame el número que debería estar en esa posición:
5
-----
FAIL: test_secuencia_fibonacci (_main_.Test.test_secuencia_fibonacci)
Traceback (most recent call last):
  File "c:/Users/Daisy/Desktop/Ciberseguridad/Posta_e_Produccion_Segura\Tareas\UD1_Fibonacci\UD1_Fibonacci\Test.py", line 15, in test_secuencia_fibonacci
    self.assertEqual(resultado[posicion-1], numero)
AssertionError: 3 != 5
-----
Ran 1 test in 2.512s
FAILED (failures=1)
PS C:\Users\Daisy\Desktop\Ciberseguridad\Posta_e_Produccion_Segura\Tareas\UD1_Fibonacci\UD1_Fibonacci>
```

¿Que tipo de proba realizamos?

O tipo de proba realizado neste caso é unha "prueba de unidad" o "prueba unitaria", onde estamos probando unha unidade individual do código, que en este caso é a función **fibonacci**. Estamos verificando se a función produce os resultados esperados para unha entrada específica.

En resumen, creamos un script para xerar a secuencia de Fibonacci e logo escribimos probas unitarias para verificar se o script se comporta como o esperado nun contexto de probas automatizadas.