

Posta En Producción Segura

Tarefa 1 – Sucesión de Fibonacci

Yago Touriño Medrano

Índice

1. Creación de la función Fibonacci
2. Creación del programa para
testeo con unittest.
3. Verificación de software y
pregunta final

Parte 1 - Creación de la función Fibonacci

Para esta parte he creado el archivo “*fibonacci.py*” donde definiré una función llamada “*fibonacci*” para poder calcular el valor del número correspondiente a la posición que se le envía, en la sucesión de Fibonacci.

```
fibonacci.py > ...
1  #Tarea 1 Yago Touriño Medrano
2  #Archivo .py donde está creada la función "fibonacci" perteneciente al primer punto del ejercicio.
3
4  def fibonacci(posicion):
5      num1 = 0
6      num2 = 1
7      if posicion <= 0:
8          return "El número debe ser superior a 0"
9      elif posicion == 1:
10         return 0
11     elif posicion == 2:
12         return 1
13     else:
14         for i in range(posicion - 2):
15             resultado = num2+num1
16             num1 = num2
17             num2 = resultado
18         return resultado
19 #print(fibonacci(5))
```

Primero defino la función “*fibonacci*” la cual recibe como parámetro el número de la posición del cual que se quiere conocer el número correspondiente a esa posición en la sucesión de Fibonacci.

```
def fibonacci(posicion):
```

Después, declaro las variables “*num1*” y “*num2*” y les asigno el valor 0 a ambos.

```
num1 = 0
num2 = 1
```

Luego, compruebo que el valor que se le envía a la función sea superior a 0, que si el valor introducido es “1” devuelva un “0” y que si el valor introducido es 2, devuelva un “1” la función.

```
if posicion <= 0:
    return "El número debe ser superior a 0"
elif posicion == 1:
    return 0
elif posicion == 2:
    return 1
```

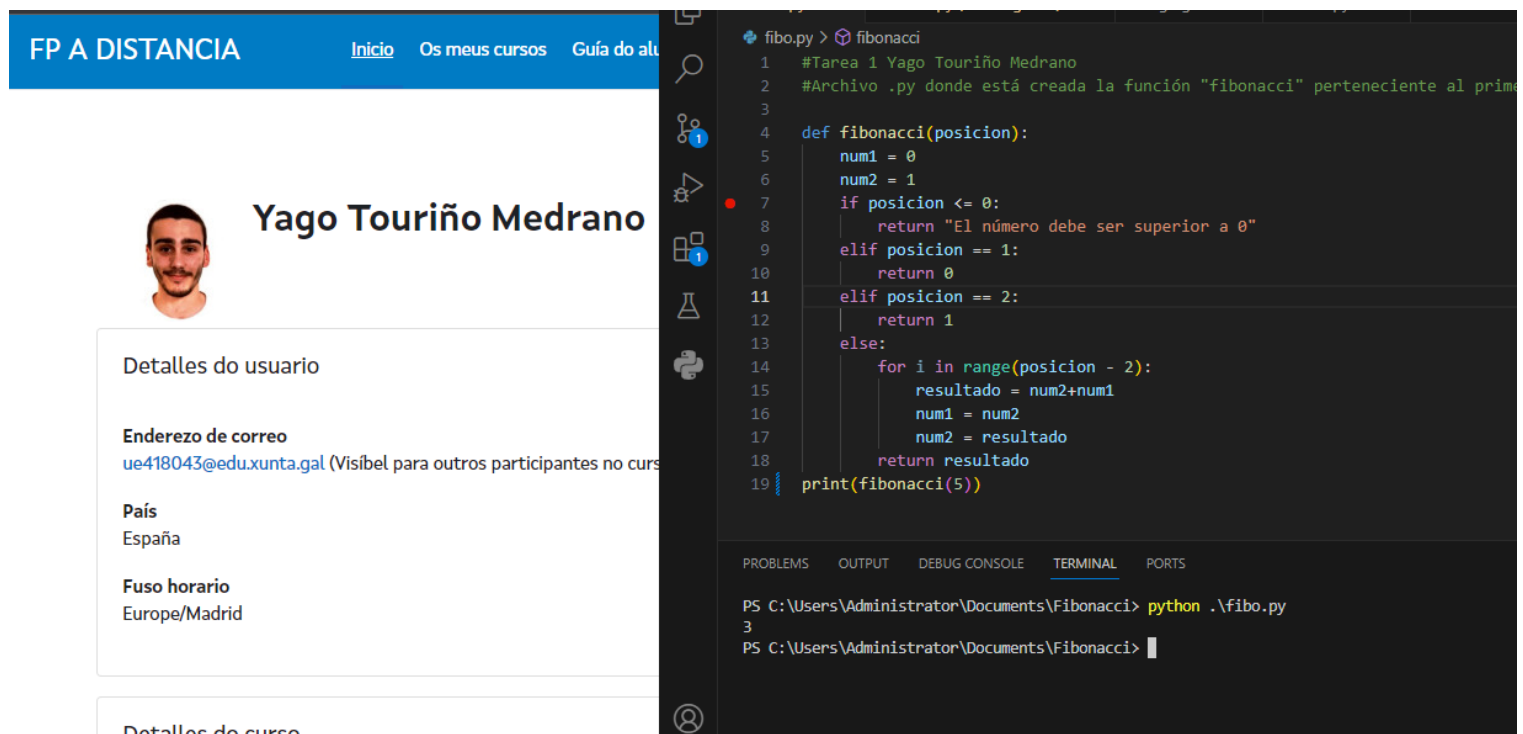
Seguidamente, si no se cumple ninguno de los tres casos anteriores, calculo con un bucle “*for*” la sucesión utilizando la función “*range*” para calcular las rotaciones que tiene que hacer el bucle y que este dé el resultado deseado de la sucesión de Fibonacci.

```
else:
    for i in range(posicion - 2):
        resultado = num2+num1
        num1 = num2
        num2 = resultado
    return resultado
```

Y hago una prueba (en el código la dejo comentada en la línea 19), para verificar que funciona correctamente la función “*fibonacci*”.

```
#print(fibonacci(5))
```

El resultado de ejecutar la función, probando a introducir la quinta posición de la sucesión sería este:



The image is a composite of two screenshots. The left screenshot shows a user profile for 'Yago Touriño Medrano' on a platform called 'FP A DISTANCIA'. The profile includes a blue header with navigation links, a profile picture, and details such as email, country (España), and time zone (Europe/Madrid). The right screenshot shows a code editor with a Python file named 'fibonacci.py'. The code defines a 'fibonacci' function that calculates the nth Fibonacci number. It includes comments in Spanish and a test call to 'print(fibonacci(5))' on line 19. Below the code editor is a terminal window showing the command 'python .\fibonacci.py' being executed, with the output '3'.

FP A DISTANCIA Inicio Os meus cursos Guía do alu

Yago Touriño Medrano

Detalles do usuario

Enderezo de correo
ue418043@edu.xunta.gal (Visíbel para outros participantes no curs

País
España

Fuso horario
Europe/Madrid

Detalles do curso

```
fibonacci.py > fibonacci
1 #Tarea 1 Yago Touriño Medrano
2 #Archivo .py donde está creada la función "fibonacci" perteneciente al primer
3
4 def fibonacci(posicion):
5     num1 = 0
6     num2 = 1
7     if posicion <= 0:
8         return "El número debe ser superior a 0"
9     elif posicion == 1:
10        return 0
11    elif posicion == 2:
12        return 1
13    else:
14        for i in range(posicion - 2):
15            resultado = num2+num1
16            num1 = num2
17            num2 = resultado
18        return resultado
19    print(fibonacci(5))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Administrator\Documents\Fibonacci> python .\fibonacci.py
3
PS C:\Users\Administrator\Documents\Fibonacci>

Parte 2 - Creación del programa para testeo con Unittest

En esta parte, crearé el fichero *“test.py”* donde estará el código correspondiente a la ejecución del test de la función del fichero *“fibonacci.py”*, utilizando la librería *“Unittest”*.

```
test.py > TestFibonacci > test_posicion
1  #Tarea 1 Yago Touriño Medrano
2  #Archivo .py donde se llama a la clase "unittest" y se importa la función "fibonacci" del archivo fibonacci.py
3  #Se crea la clase TestFibonacci para verificar que la posición número "5" de la función es "3"
4
5  import unittest
6  from fibonacci import fibonacci
7  class TestFibonacci(unittest.TestCase):
8      def test_posicion(self):
9          result = fibonacci(5)
10         self.assertEqual(result, 5, "Error, el resultado del test de la función \"fibonacci\" no es el esperado")
```

Primero, importo la librería *“unittest”* para utilizarla en el programa.

```
import unittest
```

Luego, importo la función *“fibonacci”* desde el fichero *“fibonacci.py”*

```
from fibonacci import fibonacci
```

Seguidamente, creo la clase *“TestFibonacci”* de tipo *“TestCase”* para definir dentro la función que ejecutará el test.

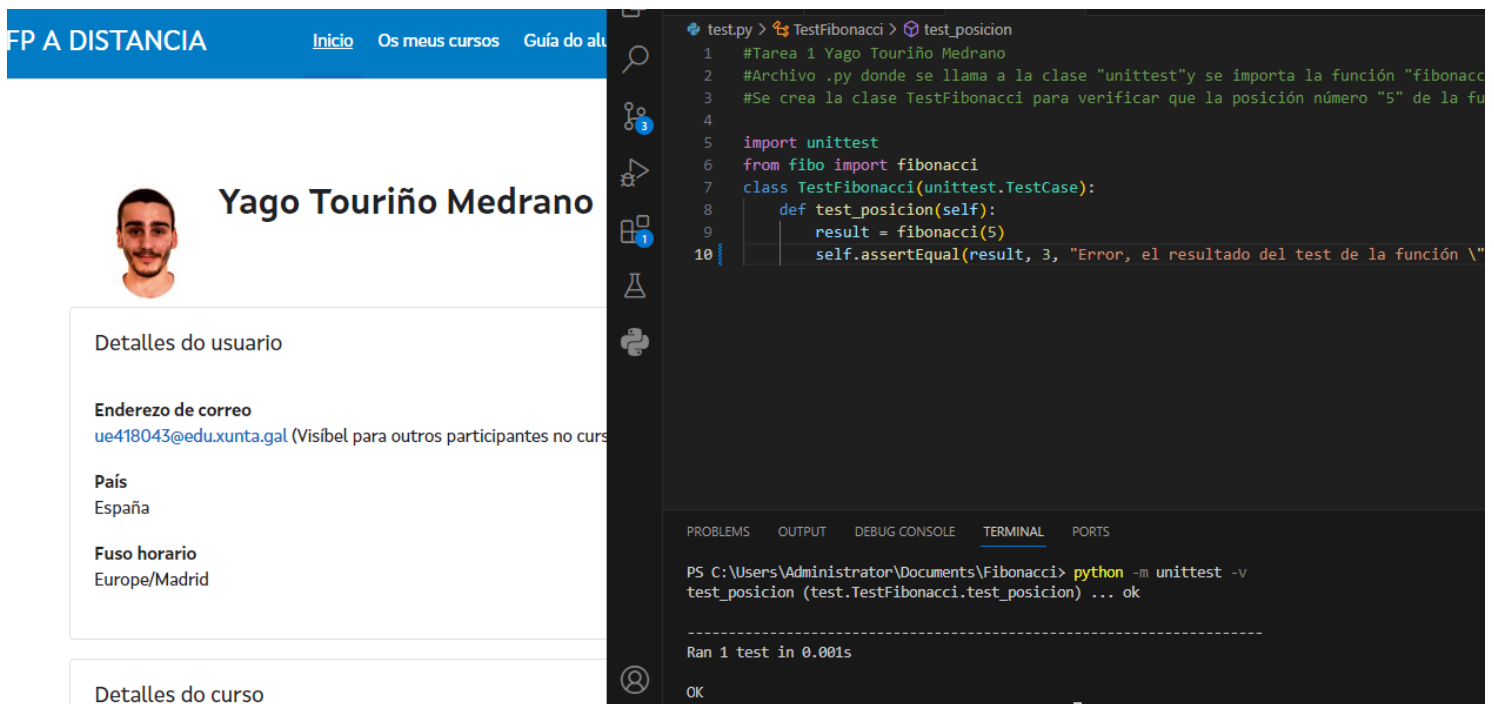
```
class TestFibonacci(unittest.TestCase):
```

Y por último, defino la función “test_posicion”, donde dentro llamo a la función “fibonacci”, le asigno el valor 5 y se lo asigno a la variable “result”. Luego, con “assertEqual”, comparo el valor que devuelve la función en la posición 5, con el resultado esperado que es “3”. En caso de que falle la validación y de un resultado no esperado, devolverá un código de error personalizado.

```
def test_posicion(self):  
    result = fibonacci(5)  
    self.assertEqual(result, 5, "Error, el resultado del test de la función \"fibonacci\" no es el esperado")
```

Parte 3 - Verificación de software y pregunta final

El resultado de la ejecución del test sería este:



The image is a composite screenshot. On the left, it shows a user profile for 'Yago Touriño Medrano' with a profile picture and details like email, country (España), and time zone (Europe/Madrid). On the right, it shows a code editor with a Python script for testing a Fibonacci function. The script defines a class 'TestFibonacci' with a method 'test_posicion' that calls 'fibonacci(5)' and asserts the result is 3. Below the code, a terminal window shows the command 'python -m unittest -v test_posicion (test.TestFibonacci.test_posicion) ... ok' and the output 'Ran 1 test in 0.001s'.

FP A DISTANCIA Inicio Os meus cursos Guía do alu

Yago Touriño Medrano

Detalles do usuario

Enderezo de correo
ue418043@edu.xunta.gal (Visíbel para outros participantes no curs

País
España

Fuso horario
Europe/Madrid

Detalles do curso

```
test.py > TestFibonacci > test_posicion
1 #Tarea 1 Yago Touriño Medrano
2 #Archivo .py donde se llama a la clase "unittest"y se importa la función "fibonacci"
3 #Se crea la clase TestFibonacci para verificar que la posición número "5" de la función "fibonacci"
4
5 import unittest
6 from fibo import fibonacci
7 class TestFibonacci(unittest.TestCase):
8     def test_posicion(self):
9         result = fibonacci(5)
10        self.assertEqual(result, 3, "Error, el resultado del test de la función \\"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Administrator\Documents\Fibonacci> python -m unittest -v test_posicion (test.TestFibonacci.test_posicion) ... ok

Ran 1 test in 0.001s

OK

Respondiendo a la pregunta del ejercicio, el test que se ha realizado, se trata de un test de tipo funcional, en el cual se realiza la comparación de dos valores, uno proporcionado por la función a testear, y otro, el resultado esperado.

Bibliografía

1. <https://manuais.pages.iessanclemente.net/plantillas/cec/pps/ud1/apuntes/>
2. <https://docs.python.org/3/library/unittest.html>
3. <https://docs.python.org/3/library/unittest.html#unittest.TestCase.assertEqual>
4. <https://docs.python.org/3/library/unittest.html#unittest.TestCase.longMessage>
5. https://www.w3schools.com/python/ref_func_range.asp