

PPS01

Tarefa UD01

**Proba Aplicacións web e Aplicacións para dispositivos
móviles, Analizando a Estructura do Código e o seu
Modelo de Execución**

Paula González Seoane

Sumario

Previo.....	3
Enlace Repositorio.....	3
Exercicio 1 – Programa Python fibo.py.....	3
Solución Exercicio 1.....	6
Solución 2 Exercicio 1.....	7
Exercicio 2 – Test unittest.....	8
Solución Exercicio 2.....	9
Exercicio 3 - Verificación de Software e pregunta final.....	10
Solución Exercicio 3.....	10
Bibliografía.....	11

Previo

Para a realización da Tarefa fago uso dunha MV co SO Lubuntu e Visual Code

Como repositorio fago uso dunha conta de Github e o repositorio o deixarei en modo de visualización público de xeito que non haxa problemas na visualización

Enlace Repositorio

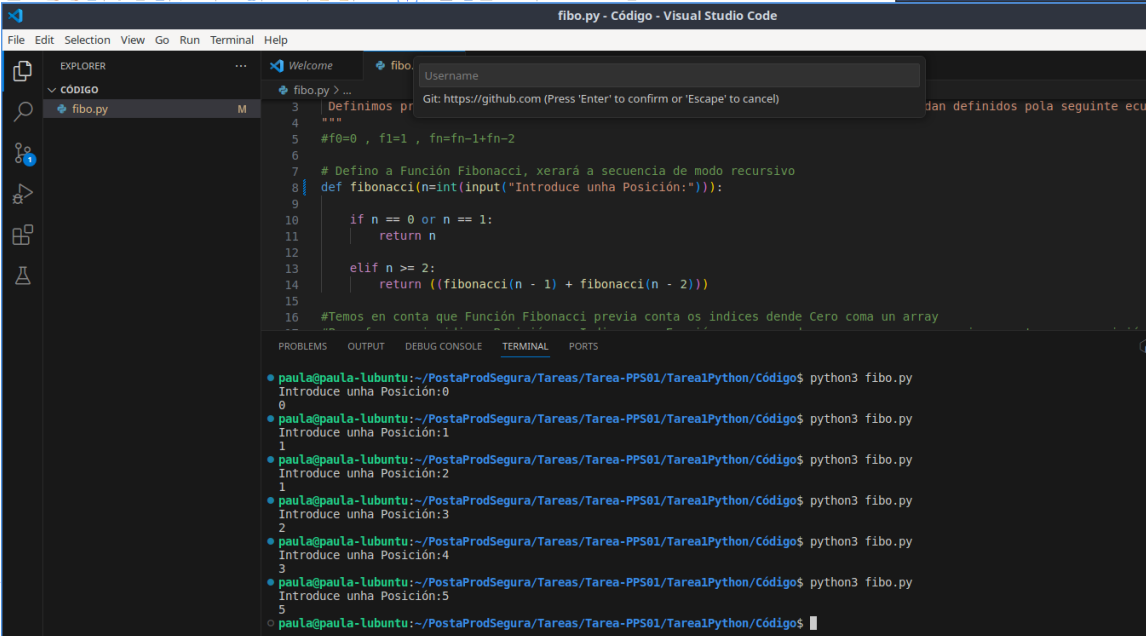
<https://github.com/Paulaa777/Tarea1Python.git>

Exercicio 1 – Programa Python fibo.py

Programa Python Secuencia Fibonacci

- Fago dúas versións do programa

- O principal no que se farán os test baseado na fórmula “clásica” da secuencia de Fibonacci: $f_0=0$, $f_1=1$, $f_n=f_{n-1}+f_{n-2}$.
- Nesta versión fago unha función na que se xenera de forma recursiva a secuencia de Fibonacci, e mostra o número solicitado, pero dado que deste xeito nos mostrará a secuencia coma se temporalmente formase un array comezará a contar dende cero polo que se lle solicito a posición 5 realmente me mostrará a 6
- Como podemos observar na seguinte captura



The screenshot shows the Visual Studio Code editor with the file `fibo.py` open. The code defines a recursive function `fibonacci` that takes an input `n` and returns the `n`-th Fibonacci number. The function uses a recursive approach, calling itself for `n-1` and `n-2`. The terminal output shows the execution of the program for inputs 0 through 5, resulting in the sequence 0, 1, 1, 2, 3, 5.

```
fibonacci.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
  fibo.py

CÓDIGO
  fibo.py
  1 # Definimos la función fibonacci
  2 """
  3 # f0=0 , f1=1 , fn=fn-1+fn-2
  4
  5 # Defino a Función Fibonacci, xerará a secuencia de modo recursivo
  6
  7 def fibonacci(n=int(input("Introduce unha Posición:"))):
  8
  9     if n == 0 or n == 1:
 10         return n
 11
 12     elif n >= 2:
 13         return ((fibonacci(n - 1) + fibonacci(n - 2)))
 14
 15
 16 #Temos en conta que Función Fibonacci previa conta os índices dende Cero coma un array

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:0
0
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:1
1
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:2
1
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:3
2
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:4
3
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:5
5
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$
```

- Resolvo esta cuestión cunha segunda función que garda nun array a secuencia ata o a posición solicitada e mostarei o último número do mesmo
- Como podemos observar na seguinte captura

The screenshot shows the Visual Studio Code editor with a file named `fibo.py` open. The code defines a recursive Fibonacci function and a `position` function that uses it to calculate and return the sequence up to a given position. The terminal at the bottom shows the execution of the script with various inputs and outputs.

```
11     return
12
13     elif n >= 2:
14         return ((fibonacci(n - 1) + fibonacci(n - 2)))
15
16 #Temos en conta que Función Fibonacci previa conta os índices dende Cero coma un array
17 #Para facer coincidir a Posición co Índice creo Función no que gardamos en array a secuencia e mostramos a posición requirida
18 def position(p=int(input("Introduce unha Posición:"))):
19
20     if p <= 0:
21         print("Sorry!!!, La Posición debe ser un Número Mayor que Cero")
22     else:
23         posicion=[fibonacci(n) for n in range(p)]
24         return print ("Para la Posición", p, "el Número es:", posicion[-1])
25
26 #chamamos o programa principal
27 if __name__ == '__main__':
28     position()
```

Terminal Output:

```
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:0
Sorry!!!, La Posición debe ser un Número Mayor que Cero
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:1
Para la Posición 1 el Número es: 0
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:2
Para la Posición 2 el Número es: 1
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:3
Para la Posición 3 el Número es: 1
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:4
Para la Posición 4 el Número es: 2
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$ python3 fibo.py
Introduce unha Posición:5
Para la Posición 5 el Número es: 3
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Código$
```

- Fixeno deste xeito porque me gustou como con esta fórmula clásica nos mostra o número rápidamente pero non souben como resolver máis eficientemente con ella que os índices coincidiran cas posicións

- O segundo, con bucle while que me gusta un pouco menos
 - Cun bucle while ata a posición introducida gardando nun array a secuencia e mostrando a última

The screenshot shows the Visual Studio Code interface with the file explorer on the left, the editor in the center, and the terminal at the bottom. The editor displays the following Python code:

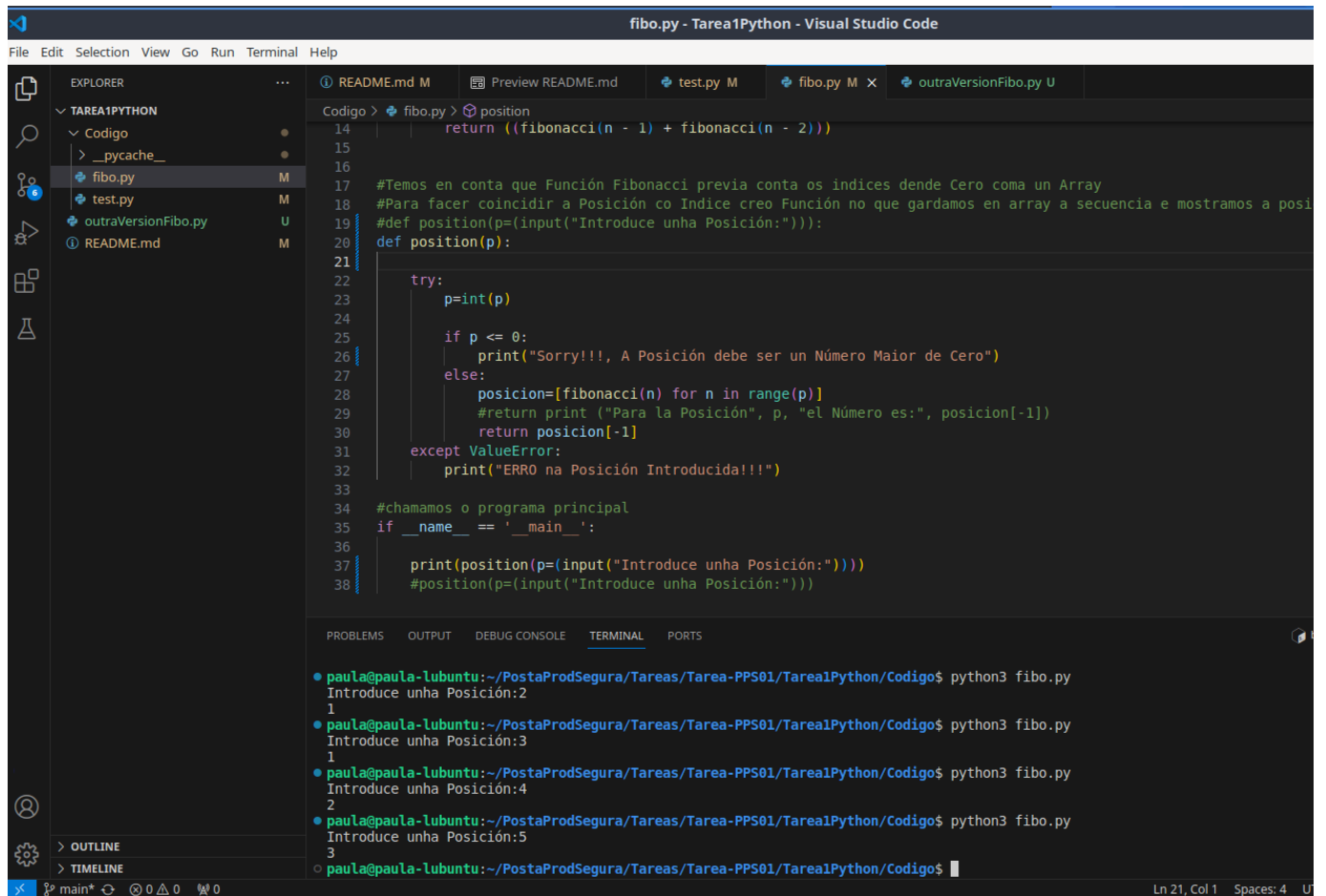
```
1 #Otra versión do programa pedido
2 def fibonacci_prueba(n):
3
4     try:
5         n=int(n)
6
7         if n <= 0:
8             print("Sorry!!!, A Posición debe ser un Número Maior de Cero")
9         else:
10
11             result = []
12             a, b = 0, 1
13
14             while a < n:
15
16                 result.append(a)
17                 a, b = b, a+b
18
19             return result[-1]
20
21     except ValueError:
22         print("ERRO na Posición Introducida!!!")
23
24 if __name__ == '__main__':
25     print(fibonacci_prueba(n=(input("Introduce unha Posición:"))))
26
```

The terminal at the bottom shows the execution of the script with the following output:

```
Introduce unha Posición:1
0
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$ python3 outraVersionFibo.py
Introduce unha Posición:2
1
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$ python3 outraVersionFibo.py
Introduce unha Posición:3
2
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$ python3 outraVersionFibo.py
Introduce unha Posición:5
3
○ paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$
```

Solución Exercicio 1

✓



The screenshot shows the Visual Studio Code editor with a file named `fibonacci.py` open. The code implements a Fibonacci sequence function and a `position` function that takes user input and returns the corresponding Fibonacci number. The code is as follows:

```
14 return ((fibonacci(n - 1) + fibonacci(n - 2)))
15
16
17 #Temos en conta que Función Fibonacci previa conta os índices dende Cero coma un Array
18 #Para facer coincidir a Posición co Índice creo Función no que gardamos en array a secuencia e mostramos a posi
19 #def position(p=(input("Introduce unha Posición:"))):
20 def position(p):
21
22     try:
23         p=int(p)
24         if p <= 0:
25             print("Sorry!!!, A Posición debe ser un Número Maior de Cero")
26         else:
27             posicion=[fibonacci(n) for n in range(p)]
28             #return print ("Para la Posición", p, "el Número es:", posicion[-1])
29             return posicion[-1]
30         except ValueError:
31             print("ERRO na Posición Introducida!!!")
32
33 #chamamos o programa principal
34 if __name__ == '__main__':
35
36     print(position(p=(input("Introduce unha Posición:"))))
37     #position(p=(input("Introduce unha Posición:")))
```

The terminal window at the bottom shows the execution of the program with four test cases:

```
paula@paula-lubuntu: ~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibo.py
Introduce unha Posición:2
1
paula@paula-lubuntu: ~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibo.py
Introduce unha Posición:3
1
paula@paula-lubuntu: ~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibo.py
Introduce unha Posición:4
2
paula@paula-lubuntu: ~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibo.py
Introduce unha Posición:5
3
paula@paula-lubuntu: ~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$
```

Solución 2 Exercicio 1

✓

The screenshot shows the Visual Studio Code interface with the file `outraVersionFibo.py` open. The Explorer sidebar on the left shows the project structure for `TAREA1PYTHON`, including `Codigo`, `__pycache__`, `fibo.py`, `test.py`, `outraVersionFibo.py` (selected), and `README.md`. The main editor displays the following Python code:

```
1 #Otra versión do programa pedido
2 def fibonacci_prueba(n):
3
4     try:
5         n=int(n)
6
7         if n <= 0:
8             print("Sorry!!!, A Posición debe ser un Número Maior de Cero")
9         else:
10
11             result = []
12             a, b = 0, 1
13
14             while a < n:
15
16                 result.append(a)
17                 a, b = b, a+b
18
19             return result[-1]
20
21     except ValueError:
22         print("ERRO na Posición Introducida!!!")
23
24 if __name__ == '__main__':
25     print(fibonacci_prueba(n=(input("Introduce unha Posición:"))))
26
```

The TERMINAL panel at the bottom shows the execution of the script three times, each time with a different input value (1, 2, and 5). The output for each run is the corresponding Fibonacci number (1, 2, and 5 respectively).

```
Introduce unha Posición:1
1
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$ python3 outraVersionFibo.py
Introduce unha Posición:2
2
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$ python3 outraVersionFibo.py
Introduce unha Posición:3
3
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$ python3 outraVersionFibo.py
Introduce unha Posición:5
5
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python$
```

The status bar at the bottom indicates the current cursor position: `Ln 20, Col 10 Spaces: 4`.

Exercicio 2 – Test unittest

Test unittest.TestCase

- Importo o módulo unittest e o programa a testear
- Fago unha clase Test na que implemento unittest.TestCase
- Creo varias funcións
 - **testCinco_Position_Equal** → Testeo con assertEquals a función principal do programa (fibonacci.position), chamando polo programa sen reflexar explícitamente a función non me saían os test. De modo que comprobamos que na posición 5 o resultado é 3
 - **testCinco_Fibonacci_NotEqual** → Testeo con assertNotEqual a función fibonacci do programa (fibonacci.fibonacci), de modo que comprobamos que na posición 5 o resultado non é 3, dado que como expliquei antes con esta función se mostran os índices dende cero polo que a posición 5 neste caso correspondería co resultado 6
 - **testCinco_Fibonacci_Equal** → Testeo con assertEquals a función fibonacci do programa (fibonacci.fibonacci), de modo que comprobamos que na posición 5 o resultado é 5 dado que como expliquei antes con esta función se mostran os índices dende cero polo que a posición 5 neste caso correspondería co resultado 6
 -
 - Incluío funcións setUp, tearDown e doCleanups, que según entendín preparan entorno de probas, execútanse despois de cada proba e limpian o entorno

Solución Exercicio 2

✓

test.py - Tarea1Python - Visual Studio Code

Help

README.md M Preview README.md test.py M X fibo.py M outraVersionFibo.py U

Codigo > test.py > Test

```
4
5
6 class Test(unittest.TestCase):
7
8     #función setup prepara entorno probas
9     def setUp(self) -> None:
10         return super().setUp()
11     #función teardown execútase despois de cada setup
12     def tearDown(self) -> None:
13         return super().tearDown()
14
15     #Proba Unitaria de Verificación > Programa fibo mostra na posición 5 o resultado 3
16     def testCinco_Position_Equal(self):
17
18         #Test assertEquals función position
19         self.assertEqual(fibo.position(5),3,"Test Incorrecto Función POSITION 5 Resultado !=3")
20
21
22     #Proba Unitaria de Verificación > función fibonnacci posición 5 o resultado non é 3
23     def testCinco_Fibonacci_NotEqual(self):
24
25         #Test assertEquals funcion fibonacci position
26         self.assertNotEqual(fibo.fibonacci(5),3,"Test Incorrecto Función POSITION 5 Resultado !=3")
27
28
29     #Proba Unitaria > función fibonnacci posición 5 o resultado es 5
30     def testCinco_Fibonacci_Equal(self):
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 -m unittest test.py -v
testCinco_Fibonacci_Equal (test.Test.testCinco_Fibonacci_Equal) ... ok
testCinco_Fibonacci_NotEqual (test.Test.testCinco_Fibonacci_NotEqual) ... ok
testCinco_Position_Equal (test.Test.testCinco_Position_Equal) ... ok

-----
Ran 3 tests in 0.000s

OK
○ paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$
```

Exercicio 3 - Verificación de Software e pregunta final

Verificación de Software e pregunta final

- As probas saíron ben unha vez o código estaba resolto
- Para isto sempre que estamos a programar facemos mil probas manuais, unitarias, funcionais e de integración
- No caso das probas con unittest diría que son probas unitarias, funcionais e de integración dado que se proban funcións específicas e o programa en sí

Solución Exercicio 3

✓ Captura Verificación Software

The screenshot shows the Visual Studio Code editor with a file named `fibonacci.py` open. The code defines a `fibonacci` function and a `position` function. The `position` function takes user input and prints the corresponding Fibonacci sequence. The terminal at the bottom shows the execution of the program for three different inputs: 5, 6, and 10.

```
def fibonacci(n):
    if n == 0 or n == 1:
        return n
    elif n >= 2:
        return ((fibonacci(n - 1) + fibonacci(n - 2)))

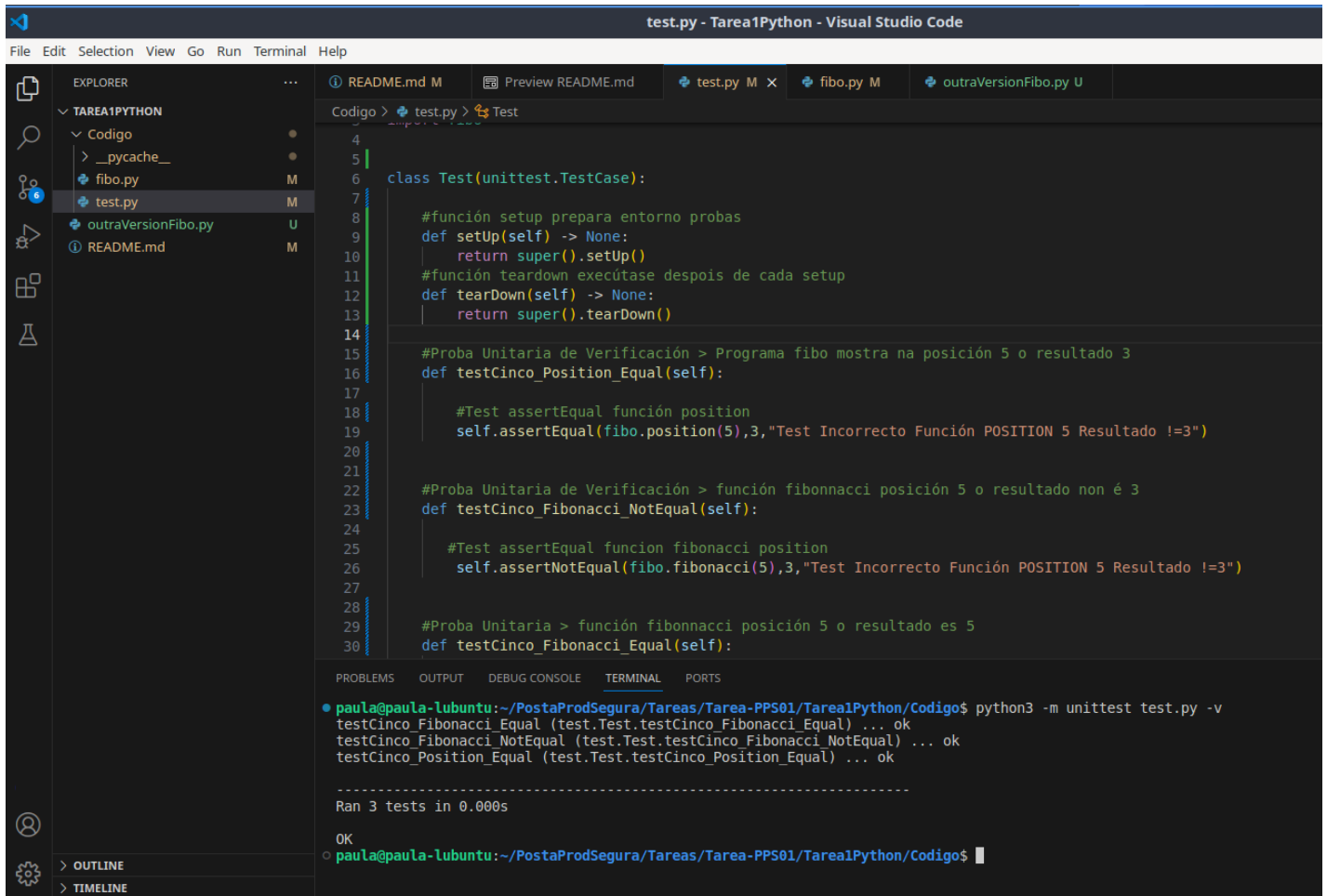
# Temos en conta que Función Fibonacci previa conta os índices dende Cero coma un Array
# Para facer coincidir a Posición co Índice creo Función no que gardamos en array a secuencia e mostramos a p
def position(p=(input("Introduce unha Posición:"))):
    def position(p):
        try:
            p=int(p)
            if p <= 0:
                print("Sorry!!!, A Posición debe ser un Número Maior de Cero")
            else:
                posicion=[fibonacci(n) for n in range(p)]
                #return print ("Para la Posición", p, "el Número es:", posicion[-1])
                return posicion[-1]
        except ValueError:
            print("ERRO na Posición Introducida!!!")

#chamamos o programa principal
```

Terminal Output:

```
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibonacci.py
Introduce unha Posición:5
3
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibonacci.py
Introduce unha Posición:6
5
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 fibonacci.py
Introduce unha Posición:10
34
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$
```

✓ Captura Test



The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'TAREA1PYTHON' with files like 'fibonacci.py', 'test.py', and 'outraVersionFibo.py'. The code editor displays the content of 'test.py', which defines a 'Test' class with three test methods: 'testCinco_Position_Equal', 'testCinco_Fibonacci_NotEqual', and 'testCinco_Fibonacci_Equal'. The terminal shows the command 'python3 -m unittest test.py -v' being executed, resulting in three successful test runs.

```
test.py - Tarea1Python - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
TAREA1PYTHON
├── Codigo
│   ├── __pycache__
│   ├── fibonacci.py
│   ├── test.py
│   └── outraVersionFibo.py
└── README.md

Codigo > test.py
4
5
6 class Test(unittest.TestCase):
7
8     #función setup prepara entorno probas
9     def setUp(self) -> None:
10         return super().setUp()
11     #función teardown executase despois de cada setup
12     def tearDown(self) -> None:
13         return super().tearDown()
14
15     #Proba Unitaria de Verificación > Programa fibo mostra na posición 5 o resultado 3
16     def testCinco_Position_Equal(self):
17
18         #Test assertEquals función position
19         self.assertEqual(fibo.position(5),3,"Test Incorrecto Función POSITION 5 Resultado !=3")
20
21
22     #Proba Unitaria de Verificación > función fibonnacci posición 5 o resultado non é 3
23     def testCinco_Fibonacci_NotEqual(self):
24
25         #Test assertEquals funcion fibonacci position
26         self.assertNotEqual(fibo.fibonacci(5),3,"Test Incorrecto Función POSITION 5 Resultado !=3")
27
28
29     #Proba Unitaria > función fibonnacci posición 5 o resultado es 5
30     def testCinco_Fibonacci_Equal(self):

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$ python3 -m unittest test.py -v
testCinco_Fibonacci_Equal (test.Test.testCinco_Fibonacci_Equal) ... ok
testCinco_Fibonacci_NotEqual (test.Test.testCinco_Fibonacci_NotEqual) ... ok
testCinco_Position_Equal (test.Test.testCinco_Position_Equal) ... ok

Ran 3 tests in 0.000s

OK
paula@paula-lubuntu:~/PostaProdSegura/Tareas/Tarea-PPS01/Tarea1Python/Codigo$
```

Bibliografía

- Apuntes y Recursos PPS01
- <https://docs.python.org/es/3/tutorial/>
- <https://docs.python.org/es/3.9/library/unittest.html#unittest.TestCase.tearDown>
- https://alvarohurtado.es/2020/11/16/que-hace-if-__name__-__main__-en-python/
- <https://learnxinyminutes.com/docs/es-es/python-es/>
- <https://aprendeconalf.es/docencia/python/ejercicios/>