

# **Posta en Producción Segura**

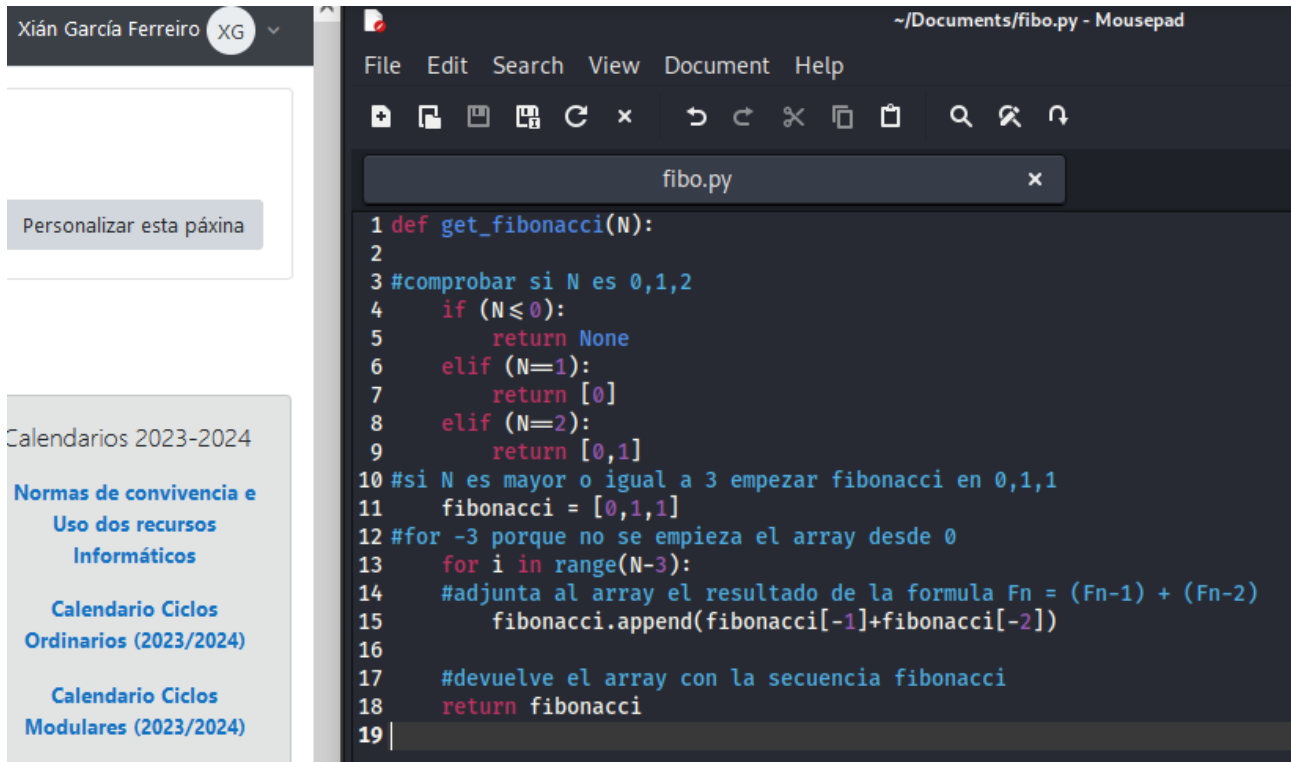
## **UD1**

**Prueba de aplicaciones web y móviles,  
analizando el código y su modelo de  
ejecución. Herramientas básicas.**

**Xián García Ferreiro**

- **Actividades entregables**
- **Resultados de aprendizaje**

1. Crea un script que genere la secuencia de Fibonacci



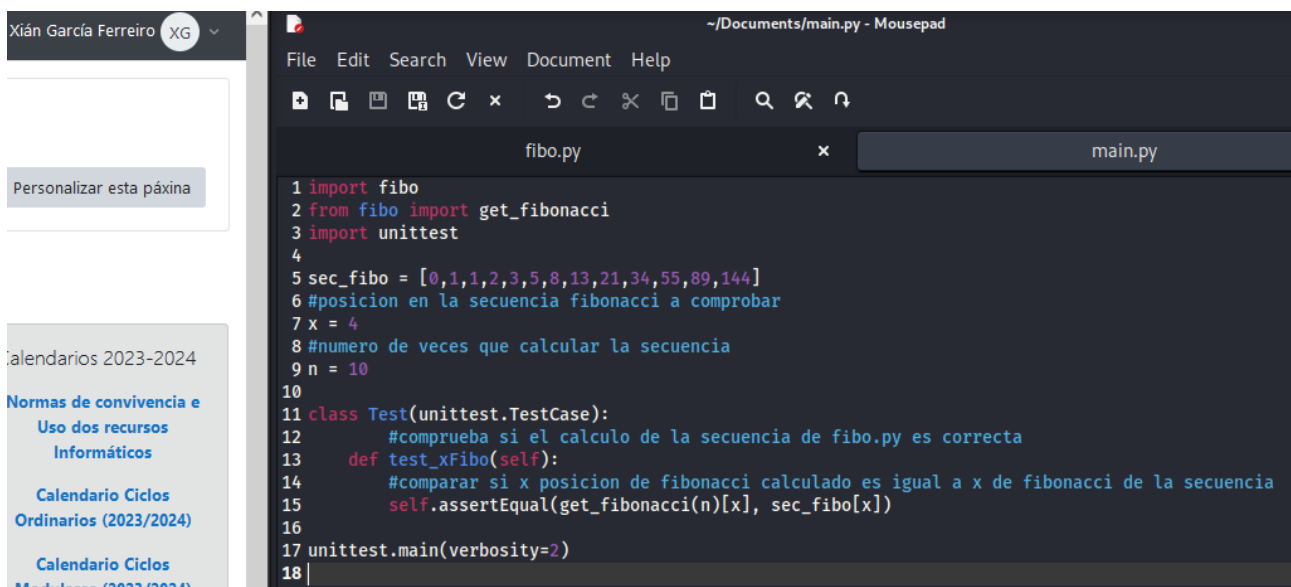
The screenshot shows a web browser on the left with a sidebar containing links like 'Calendarios 2023-2024', 'Normas de convivencia e Uso dos recursos Informáticos', and 'Calendario Ciclos Ordinarios (2023/2024)'. The main content area displays the code for 'fibo.py' in a dark-themed editor. The code defines a function 'get\_fibonacci(N)' that returns the Nth Fibonacci number. It handles base cases for N=0, 1, and 2, and for N>=3, it builds an array from 0 to N-1, updating it with the sum of the last two elements.

```

1 def get_fibonacci(N):
2
3 #comprobar si N es 0,1,2
4     if (N<=0):
5         return None
6     elif (N==1):
7         return [0]
8     elif (N==2):
9         return [0,1]
10 #si N es mayor o igual a 3 empezar fibonacci en 0,1,1
11     fibonacci = [0,1,1]
12 #for -3 porque no se empieza el array desde 0
13     for i in range(N-3):
14         #adjunta al array el resultado de la formula Fn = (Fn-1) + (Fn-2)
15         fibonacci.append(fibonacci[-1]+fibonacci[-2])
16
17     #devuelve el array con la secuencia fibonacci
18     return fibonacci
19

```

2. Creación del programa principal



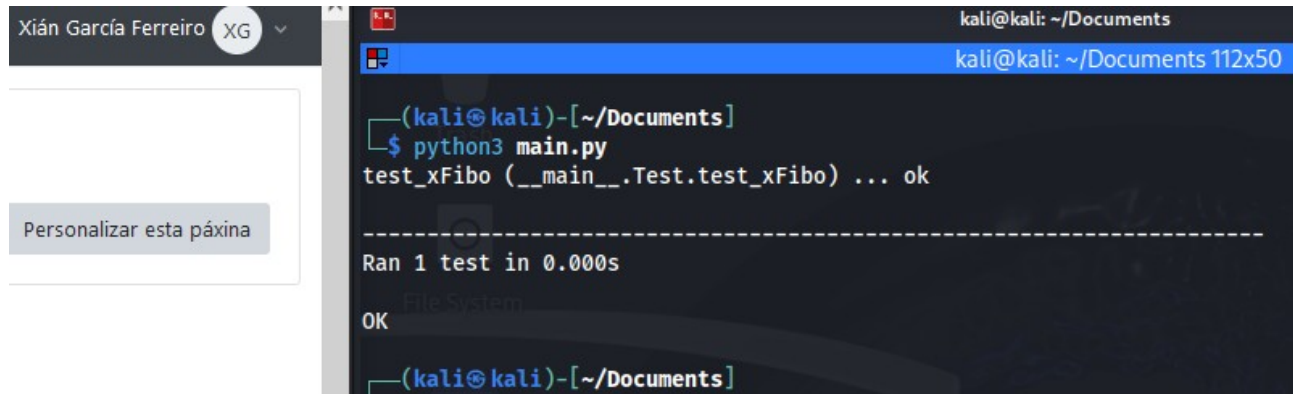
The screenshot shows the same web browser on the left. The code editor on the right now shows 'main.py'. It imports the 'fibo' module and 'unittest'. It defines a list 'sec\_fibo' with the first 14 Fibonacci numbers. It sets 'x' to 4 and 'n' to 10. A test class 'Test' inherits from 'unittest.TestCase' and has a method 'test\_xFibo' that checks if the 4th element of the sequence calculated by 'get\_fibonacci(n)' matches the 4th element of 'sec\_fibo'. Finally, it calls 'unittest.main(verbosity=2)'.

```

1 import fibo
2 from fibo import get_fibonacci
3 import unittest
4
5 sec_fibo = [0,1,1,2,3,5,8,13,21,34,55,89,144]
6 #posicion en la secuencia fibonacci a comprobar
7 x = 4
8 #numero de veces que calcular la secuencia
9 n = 10
10
11 class Test(unittest.TestCase):
12     #comprueba si el calculo de la secuencia de fibo.py es correcta
13     def test_xFibo(self):
14         #comparar si x posicion de fibonacci calculado es igual a x de fibonacci de la secuencia
15         self.assertEqual(get_fibonacci(n)[x], sec_fibo[x])
16
17 unittest.main(verbosity=2)
18

```

### 3. Verificación de software y pregunta final



The image shows a web browser window on the left and a terminal window on the right. The browser window has a header with the name 'Xíán García Ferreiro' and a profile icon 'XG'. Below the header is a button labeled 'Personalizar esta página'. The terminal window is titled 'kali@kali: ~/Documents' and shows the execution of a Python script. The terminal output is as follows:

```
(kali@kali)-[~/Documents]
$ python3 main.py
test_xFibo (__main__.Test.test_xFibo) ... ok

-----
Ran 1 test in 0.000s

OK
(kali@kali)-[~/Documents]
```

- ¿Qué tipo de prueba hemos realizado?
  - Pruebas de integración, unidad

# Anexo: Código fuente

## [Repo Github](#)

fibonacci.py

```
def get_fibonacci(N):

#comprobar si N es 0,1,2
    if (N<=0):
        return None
    elif (N==1):
        return [0]
    elif (N==2):
        return [0,1]
#si N es mayor o igual a 3 empezar fibonacci en 0,1,1
    fibonacci = [0,1,1]
#for -3 porque no se empieza el array desde 0
    for i in range(N-3):
        #adjunta al array el resultado de la formula Fn = (Fn-1) + (Fn-2)
        fibonacci.append(fibonacci[-1]+fibonacci[-2])

#devuelve el array con la secuencia fibonacci
    return fibonacci
```

main.py

```
import fibo
from fibo import get_fibonacci
import unittest

sec_fibo = [0,1,1,2,3,5,8,13,21,34,55,89,144]
#posicion en la secuencia fibonacci a comprobar
x = 4
#numero de veces que calcular la secuencia
n = 10

class Test(unittest.TestCase):
    #comprueba si el calculo de la secuencia de fibo.py es correcta
    def test_xFibo(self):
        #comparar si x posicion de fibonacci calculado es igual a x de
        fibonacci de la secuencia
        self.assertEqual(get_fibonacci(n)[x], sec_fibo[x])

unittest.main(verbosity=2)
```