

Curso Especialización

Ciberseguridad

Asignatura:

Puesta en Producción Segura

Javier Mosquera González

ÍNDICE

- 1. Script con la función de fibonacci**
- 2. Programa principal para prueba**
- 3. Verificación y pregunta final**

1. Script sucesión de fibonacci (fibonacci.py)

```
1 def fibonacci(n):
2     secuencia = [0, 1]
3
4     for i in range(2, n):
5         siguiente_num = secuencia[-1] + secuencia[-2]
6         secuencia.append(siguiente_num)
7
8     return secuencia
9
10 def main():
11     try:
12         n = int(input("Cuántos términos quieres generar: "))
13
14         if n <= 0:
15             print("Tiene que ser un número entero positivo.")
16         else:
17             resultado = fibonacci(n)
18             print(f"La secuencia de Fibonacci con {n} términos es: {resultado}")
19
20     except ValueError:
21         print("Ingresa un número entero válido.")
22
23 if __name__ == "__main__":
24     main()
```

Ejecución del script (py fibo.py desde cmd)

The screenshot shows a web browser window displaying a course page for "15021482. CSIFC91. MP5023. Posta en producción segura. 2023-2024. (Grupo A)". The page has a blue header with navigation links: Inicio, Os meus cursos, Guía do alumnado, and Incidencias. A sidebar on the left lists various course materials and activities. In the foreground, a terminal window titled "Símbolo del sistema" is open, showing the execution of the Python script "fibonacci.py". The terminal output shows three successful runs with inputs 5, -3, and 7, and one error message for an invalid input.

```
C:\Users\jmosq\Desktop>py fibonacci.py
Cuántos términos quieres generar: 5
La secuencia de Fibonacci con 5 términos es: [0, 1, 1, 2, 3]

C:\Users\jmosq\Desktop>py fibonacci.py
Cuántos términos quieres generar: -3
Tiene que ser un número entero positivo.

C:\Users\jmosq\Desktop>py fibonacci.py
Cuántos términos quieres generar: 7
La secuencia de Fibonacci con 7 términos es: [0, 1, 1, 2, 3, 5, 8]

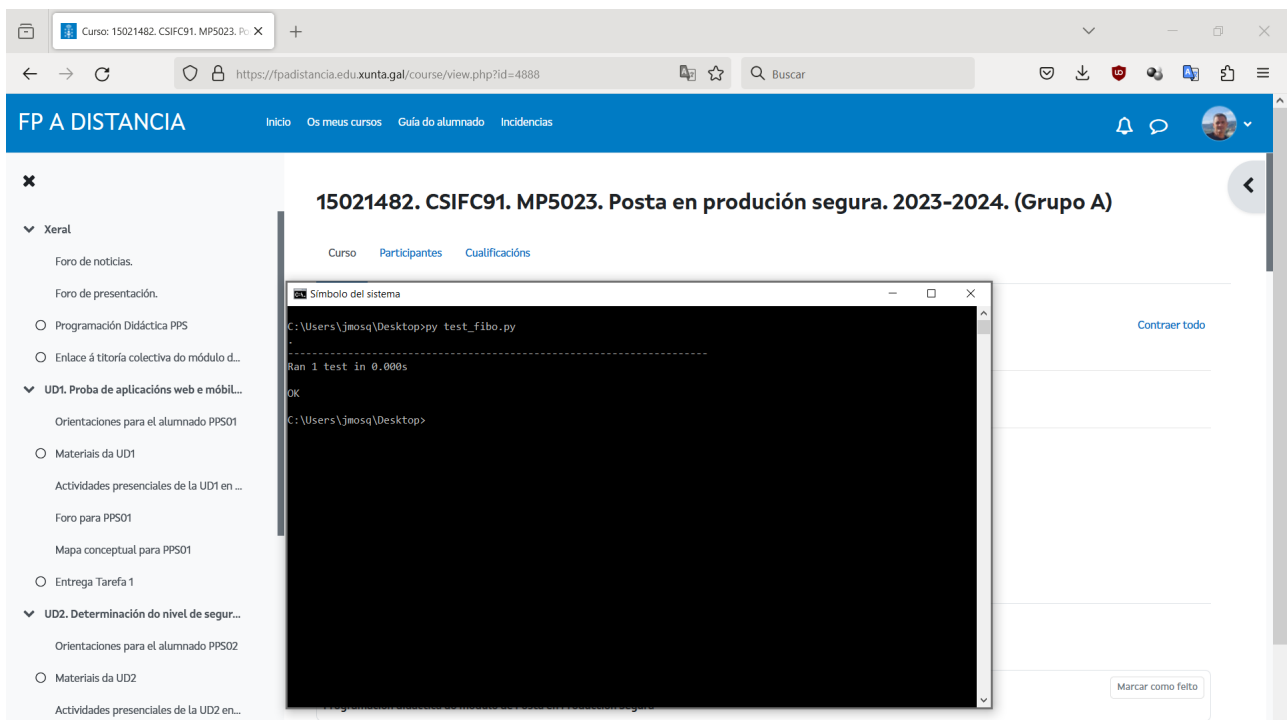
C:\Users\jmosq\Desktop>
```

2. Programa principal para prueba (test_fibo.py)

```
1 # test_fibo.py
2 import unittest
3 from fibo import fibonacci
4
5 class TestFibonacci(unittest.TestCase):
6     def test_quinto_numero(self):
7         resultado = fibonacci(5)
8         self.assertEqual(resultado[-1], 3, "El quinto número de la secuencia de Fibonacci debería ser 3")
9
10 if __name__ == "__main__":
11     unittest.main()
```

3. Verificación y pregunta final

Ejecución (py test_fibo.py)



Hemos realizado una **prueba funcional**, ya que hemos verificado una función emulando una escenario (pasando la quinta posición). Sería una prueba de caja negra.

NOTA: Los archivos *fibo.py* y *test.py* deben de estar en el mismo directorio o en el path de python.