

Esta es la prueba unitaria de Factorial y de Fibonacci

```
# test_calculadora.py

import unittest
from calculadora import calcular_factorial, calcular_fibonacci

class TestCalculos(unittest.TestCase):
    """Clase de pruebas para las funciones de cálculo."""

    # -----
    # PRUEBAS PARA FACTORIAL
    # -----
    def test_factorial_base(self):
        """Comprobar casos base del factorial."""
        self.assertEqual(calcular_factorial(0), 1)
        self.assertEqual(calcular_factorial(1), 1)

    def test_factorial_valor(self):
        """Comprobar un caso normal de factorial."""
        self.assertEqual(calcular_factorial(5), 120) # 5! = 120

    def test_factorial_error(self):
        """Comprobar que lanza error con número negativo."""
        with self.assertRaises(ValueError):
            calcular_factorial(-3)

    # -----
    # PRUEBAS PARA FIBONACCI
    # -----
    def test_fibonacci_base(self):
        """Comprobar casos base de Fibonacci."""
        self.assertEqual(calcular_fibonacci(0), 0)
        self.assertEqual(calcular_fibonacci(1), 1)

    def test_fibonacci_valor(self):
        """Comprobar un caso normal de Fibonacci."""
        self.assertEqual(calcular_fibonacci(6), 8) # Serie:
        0, 1, 1, 2, 3, 5, 8

    def test_fibonacci_error(self):
        """Comprobar que lanza error con número negativo."""
        with self.assertRaises(ValueError):
            calcular_fibonacci(-5)

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

Resultado:

```
Launching unittests with arguments python -m unittest C:\Users\adri\PyCharmMiscProject\test_calculadora.py in C:\Users\adri\PyCharmMiscProject

Ran 6 tests in 0.020s

OK

Process finished with exit code 0
```

Hice otra prueba unitaria de fibonacci:

```
# test_fibo.py

# Importamos la librería de pruebas de software incluida en
Python
import unittest

# Importamos la función fibonacci desde el archivo fibo.py
from fibo import fibonacci

# Creamos una clase de prueba que hereda de unittest.TestCase
class TestFibonacci(unittest.TestCase):
    """Clase de pruebas para la función fibonacci."""

    # Definimos una función de prueba. Los métodos de prueba deben
    empezar por 'test_'
    def test_quinto_numero(self):
        """Comprobar que el quinto número de Fibonacci sea 3."""

        # Llamamos a la función fibonacci con el valor 4
        # (recordemos: F(0)=0, F(1)=1, F(2)=1, F(3)=2, F(4)=3)
        result = fibonacci(4)

        # Comprobamos si el resultado devuelto por la función es
        igual a 3
        # Si no lo es, la prueba fallará
        self.assertEqual(result, 3)

# Este bloque indica que si ejecutamos directamente este archivo,
# Python debe ejecutar las pruebas definidas arriba.
if __name__ == '__main__':
    unittest.main()
```

Resultado:

```
Launching unittests with arguments python -m unittest C:\Users\adri\PyCharmMiscProject\test_fibo.py in C:\Users\adri\PyCharmMiscProject

Ran 1 test in 0.027s

OK

Process finished with exit code 0
```

Codigos:

1 Fibonacci:

```
# fibo.py

# Definimos una función llamada fibonacci que recibe un número entero n.
def fibonacci(n):
    """Devuelve el número de Fibonacci en la posición n."""

    # Si el número es negativo, lanzamos un error porque la sucesión no está
    # definida para negativos.
    if n < 0:
        raise ValueError("El número debe ser positivo")

    # Si n es 0, el valor de Fibonacci es 0 (caso base).
    elif n == 0:
        return 0

    # Si n es 1, el valor de Fibonacci es 1 (segundo caso base).
    elif n == 1:
        return 1

    # Si n es mayor o igual a 2, usamos un bucle para calcular la secuencia.
    else:
        a, b = 0, 1 # 'a' es el número anterior y 'b' el actual (iniciamos con
        # F(0)=0 y F(1)=1)
        # Repetimos el cálculo desde la posición 2 hasta n (inclusive)
        for _ in range(2, n + 1):
            a, b = b, a + b # Avanzamos la secuencia: el nuevo número es la suma
            # de los dos anteriores
        return b # Devolvemos el número que corresponde a la posición n

# Este bloque solo se ejecuta si el archivo se ejecuta directamente (no cuando se
# importa desde otro script)
if __name__ == "__main__":
    # Pedimos al usuario que introduzca la posición que quiere calcular
    limite = int(input("Introduce la posición hasta la que deseas calcular
    Fibonacci: "))

    # Mostramos un mensaje con la posición que se calculará
    print(f"La secuencia de Fibonacci hasta la posición {limite} es:")

    # Usamos un bucle for para mostrar todos los números desde F(0) hasta
    # F(limite)
    for i in range(limite + 1):
        # Llamamos a la función fibonacci para cada número y mostramos el
        # resultado
```

```
print(f"F({i}) = {fibonacci(i)}")
```

2. Calculadora:

```
# calculadora.py

# Importamos el módulo sys para poder salir del programa cuando
# el usuario lo desee
import sys

# -----
# FUNCIÓN: calcular_factorial
# -----
def calcular_factorial(n):
    """Devuelve el factorial de un número entero positivo n."""
    if n < 0:
        # Si el número es negativo, lanzamos un error
        raise ValueError("El número debe ser positivo.")
    elif n == 0 or n == 1:
        # El factorial de 0 y 1 es siempre 1
        return 1
    else:
        # Calculamos el factorial multiplicando todos los enteros
        # desde 1 hasta n
        resultado = 1
        for i in range(1, n + 1):
            resultado *= i
        return resultado

# -----
# FUNCIÓN: calcular_fibonacci
# -----
def calcular_fibonacci(n):
    """Devuelve el número de Fibonacci en la posición n."""
    if n < 0:
        raise ValueError("El número debe ser positivo.")
    elif n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        a, b = 0, 1
        for _ in range(2, n + 1):
            a, b = b, a + b
        return b

# -----
# FUNCIÓN: mostrar_menu
# -----
def mostrar_menu():
```

```

    """Muestra el menú de opciones al usuario."""
    print("\n===== MENÚ DE OPCIONES =====")
    print("1. Calcular un valor de la Serie de Fibonacci")
    print("2. Calcular el Factorial de un número")
    print("3. Salir del programa")
    print("=====")

# -----
# PROGRAMA PRINCIPAL
# -----
if __name__ == "__main__":
    while True:
        mostrar_menu() # Mostramos las opciones
        opcion = input("Elige una opción (1-3): ")

        if opcion == "1":
            # Opción para calcular Fibonacci
            try:
                n = int(input("Introduce un número entero
positivo: "))
                resultado = calcular_fibonacci(n)
                print(f"El número de Fibonacci en la posición {n}
es: {resultado}")
            except ValueError as e:
                print("Error:", e)

        elif opcion == "2":
            # Opción para calcular Factorial
            try:
                n = int(input("Introduce un número entero
positivo: "))
                resultado = calcular_factorial(n)
                print(f"El factorial de {n} es: {resultado}")
            except ValueError as e:
                print("Error:", e)

        elif opcion == "3":
            # Salir del programa
            print("Saliendo del programa... ¡Hasta luego!")
            sys.exit()

        else:
            # Opción inválida
            print("Opción no válida. Por favor, elige 1, 2 o 3.")

```

Resultado:

Fibonacci:

```
C:\Users\adriif\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\adriif\PyCharmMiscProject\fibonacci.py
Introduce la posición hasta la que deseas calcular Fibonacci: 5
La secuencia de Fibonacci hasta la posición 5 es:
F(0) = 0
F(1) = 1
F(2) = 1
F(3) = 2
F(4) = 3
F(5) = 5

Process finished with exit code 0
```

Calculadora:

- Menú

```
C:\Users\adriif\PyCharmMiscProject\.venv\Scripts\python.exe
===== MENÚ DE OPCIONES =====
1. Calcular un valor de la Serie de Fibonacci
2. Calcular el Factorial de un número
3. Salir del programa
=====
Elige una opción (1-3):
```

- Elegimos Fibonacci

```
Elige una opción (1-3): 1
Introduce un número entero positivo: 10
El número de Fibonacci en la posición 10 es: 55

===== MENÚ DE OPCIONES =====
1. Calcular un valor de la Serie de Fibonacci
2. Calcular el Factorial de un número
3. Salir del programa
=====
Elige una opción (1-3):
```

- Elegimos Factorial:

```
Elige una opción (1-3): 2
Introduce un número entero positivo: 4
El factorial de 4 es: 24

===== MENÚ DE OPCIONES =====
1. Calcular un valor de la Serie de Fibonacci
2. Calcular el Factorial de un número
3. Salir del programa
=====
Elige una opción (1-3):
```

- Salir del programa

```
Elige una opción (1-3): 3
Saliendo del programa... ¡Hasta luego!

Process finished with exit code 0
```