

## Trabajo Práctico

### “Recursividad”

Alumna: Wolanink, Melany Luz Valeria (wolaninkmelany@gmail.com)

## **Tecnicatura Universitaria En Programación**

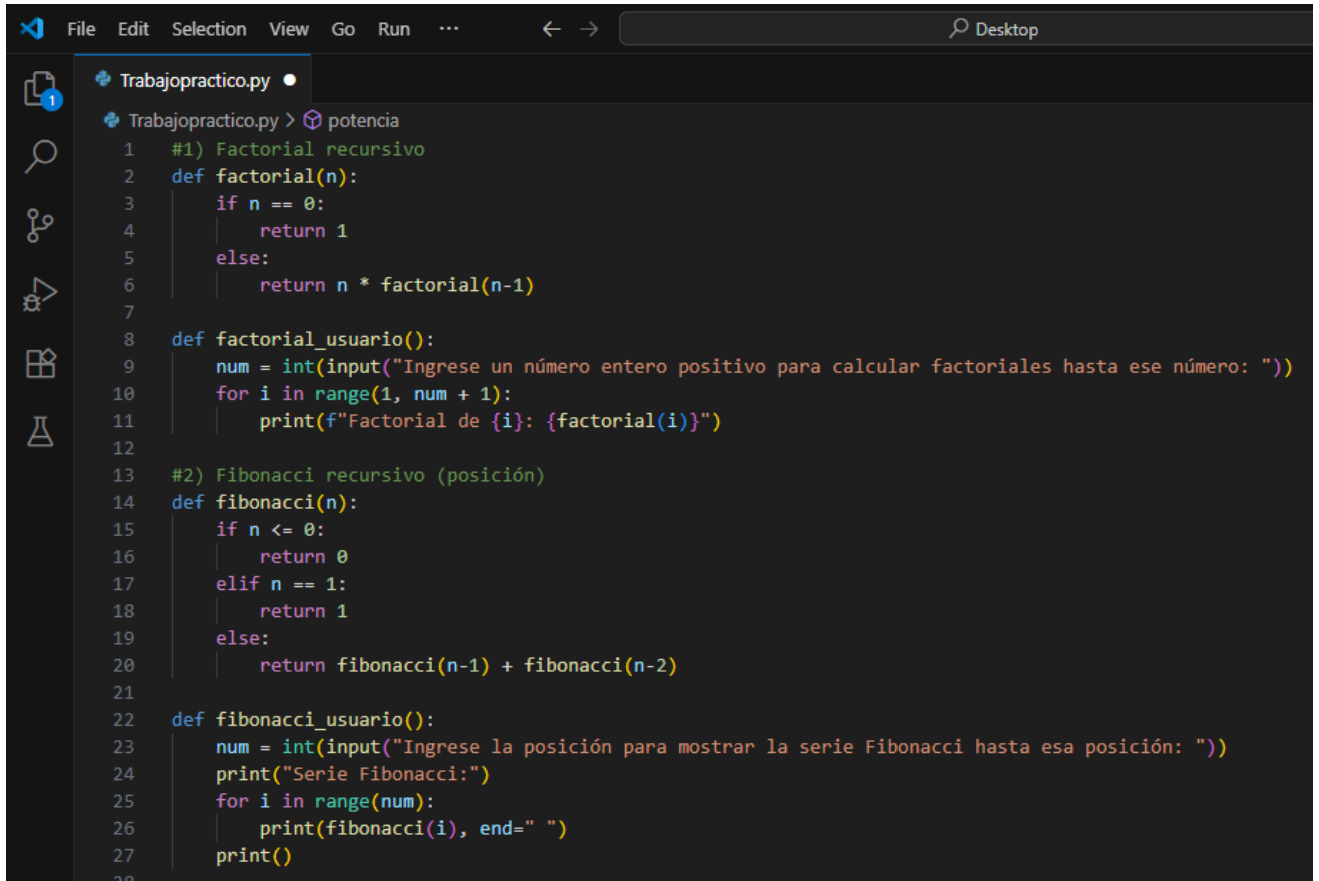
### **Universidad Tecnológica Nacional**

### **Programación I**

Comisión: 23

Docente Titular: Nicolas Quirós

Docente Tutor: Flor Camila Gubiotti



```
Trabajopractico.py
Trabajopractico.py > potencia
1 #1) Factorial recursivo
2 def factorial(n):
3     if n == 0:
4         return 1
5     else:
6         return n * factorial(n-1)
7
8 def factorial_usuario():
9     num = int(input("Ingrese un número entero positivo para calcular factoriales hasta ese número: "))
10    for i in range(1, num + 1):
11        print(f"Factorial de {i}: {factorial(i)}")
12
13 #2) Fibonacci recursivo (posición)
14 def fibonacci(n):
15     if n <= 0:
16         return 0
17     elif n == 1:
18         return 1
19     else:
20         return fibonacci(n-1) + fibonacci(n-2)
21
22 def fibonacci_usuario():
23     num = int(input("Ingrese la posición para mostrar la serie Fibonacci hasta esa posición: "))
24     print("Serie Fibonacci:")
25     for i in range(num):
26         print(fibonacci(i), end=" ")
27     print()
28
```

```

File Edit Selection View Go Run ... Desktop

Trabajopractico.py
Trabajopractico.py > es_palindromo

28
29 #3) Potencia recursiva (base^exponente)
30 def potencia(base, exponente):
31     if exponente == 0:
32         return 1
33     else:
34         return base * potencia(base, exponente - 1)
35
36 def potencia_usuario():
37     base = int(input("Ingrese la base: "))
38     exponente = int(input("Ingrese el exponente: "))
39     print(f"{base} elevado a {exponente} es {potencia(base, exponente)}")
40
41 #4) Decimal a binario recursivo
42 def decimal_a_binario(num):
43     if num == 0:
44         return ""
45     else:
46         return decimal_a_binario(num // 2) + str(num % 2)
47
48 def decimal_a_binario_usuario():
49     num = int(input("Ingrese un número decimal positivo para convertir a binario: "))
50     if num == 0:
51         print("0")
52     else:
53         print(decimal_a_binario(num))
54

```

```

File Edit Selection View Go Run ... Desktop

Trabajopractico.py
Trabajopractico.py > es_palindromo

54
55 #5) Palíndromo recursivo
56 def es_palindromo(palabra):
57     if len(palabra) <= 1:
58         return True
59     elif palabra[0] != palabra[-1]:
60         return False
61     else:
62         return es_palindromo(palabra[1:-1])
63
64 def palindromo_usuario():
65     palabra = input("Ingrese una palabra sin espacios ni tildes para verificar si es palíndromo: ").lower()
66     if es_palindromo(palabra):
67         print("Es palíndromo")
68     else:
69         print("No es palíndromo")
70
71 #6) Suma de dígitos recursiva
72 def suma_digitos(n):
73     if n == 0:
74         return 0
75     else:
76         return n % 10 + suma_digitos(n // 10)
77
78 def suma_digitos_usuario():
79     num = int(input("Ingrese un número entero positivo para sumar sus dígitos: "))
80     print(f"Suma de dígitos: {suma_digitos(num)}")
81

```

```
81
82 #7) Contar bloques pirámide recursivo
83 def contar_bloques(n):
84     if n == 1:
85         return 1
86     else:
87         return n + contar_bloques(n - 1)
88
89 def contar_bloques_usuario():
90     n = int(input("Ingrese el número de bloques en el nivel más bajo de la pirámide: "))
91     print(f"Total de bloques necesarios: {contar_bloques(n)}")
92
93 #8) Contar ocurrencias de un dígito recursivo
94 def contar_digito(numero, digito):
95     if numero == 0:
96         return 0
97     else:
98         return (1 if numero % 10 == digito else 0) + contar_digito(numero // 10, digito)
99
100 def contar_digito_usuario():
101     numero = int(input("Ingrese un número entero positivo: "))
102     digito = int(input("Ingrese un dígito para contar sus apariciones: "))
103     print(f"El dígito {digito} aparece {contar_digito(numero, digito)} veces en el número {numero}")
104
105 #Ejemplo de ejecución de todos (descomentar para probar)
106
107 # factorial_usuario()
108 # fibonacci_usuario()
109 # potencia_usuario()
110 # decimal_a_binario_usuario()
111 # palindromo_usuario()
112 # suma_digitos_usuario()
```