

UNIVERSIDA PRIVADA DOMINGO SAVIO



EJERCICIO_CLASE05

MATERIA: PROGRAMACION 2

DOCENTE: ING. JIMMY NATANIEL REQUENA LLORENTTY

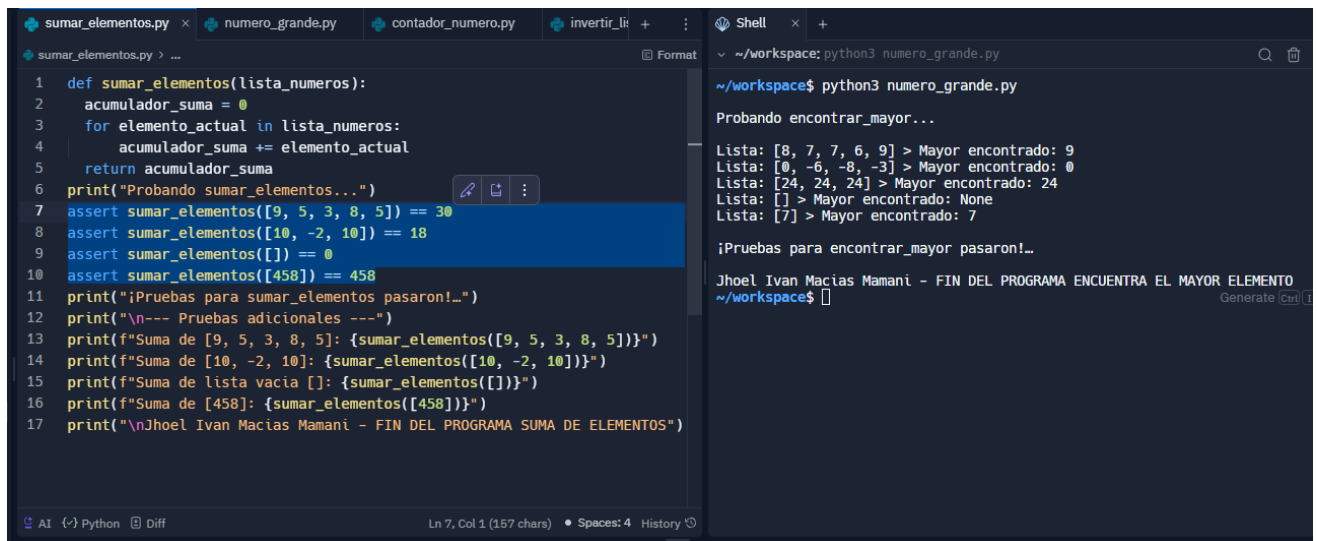
ALUMNA: JHOEL IVAN MACIAS MAMANI

CARRERA: INGENIERIA EN REDES Y TELECOMUNICACIONES

EJERCICIO 1 SUMA DE ELEMENTOS

```
def sumar_elementos(lista_numeros):
    acumulador_suma = 0
    for elemento_actual in lista_numeros:
        acumulador_suma += elemento_actual
    return acumulador_suma

print("Probando sumar_elementos...")
assert sumar_elementos([9, 5, 3, 8, 5]) == 30
assert sumar_elementos([10, -2, 10]) == 18
assert sumar_elementos([]) == 0
assert sumar_elementos([458]) == 458
print("¡Pruebas para sumar_elementos pasaron!...")
print("\n--- Pruebas adicionales ---")
print(f"Suma de [9, 5, 3, 8, 5]: {sumar_elementos([9, 5, 3, 8, 5])}")
print(f"Suma de [10, -2, 10]: {sumar_elementos([10, -2, 10])}")
print(f"Suma de lista vacia []: {sumar_elementos([])}")
print(f"Suma de [458]: {sumar_elementos([458])}")
print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA SUMA DE ELEMENTOS")
```



```
def sumar_elementos(lista_numeros):
    acumulador_suma = 0
    for elemento_actual in lista_numeros:
        acumulador_suma += elemento_actual
    return acumulador_suma

print("Probando sumar_elementos...")
assert sumar_elementos([9, 5, 3, 8, 5]) == 30
assert sumar_elementos([10, -2, 10]) == 18
assert sumar_elementos([]) == 0
assert sumar_elementos([458]) == 458
print("¡Pruebas para sumar_elementos pasaron!...")
print("\n--- Pruebas adicionales ---")
print(f"Suma de [9, 5, 3, 8, 5]: {sumar_elementos([9, 5, 3, 8, 5])}")
print(f"Suma de [10, -2, 10]: {sumar_elementos([10, -2, 10])}")
print(f"Suma de lista vacia []: {sumar_elementos([])}")
print(f"Suma de [458]: {sumar_elementos([458])}")
print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA SUMA DE ELEMENTOS")
```

```
~/workspace$ python3 numero_grande.py
Probando encontrar_mayor...
Lista: [8, 7, 7, 6, 9] > Mayor encontrado: 9
Lista: [0, -6, -8, -3] > Mayor encontrado: 0
Lista: [24, 24, 24] > Mayor encontrado: 24
Lista: [] > Mayor encontrado: None
Lista: [7] > Mayor encontrado: 7
¡Pruebas para encontrar_mayor pasaron!...
Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO
~/workspace$
```

EJERCICIO 2 ENCONTRAR EL MAYOR ELEMENTO

```
def encontrar_mayor(lista_numeros):

    if not lista_numeros:
        return None

    mayor_temporal = lista_numeros[0]

    for elemento_actual in lista_numeros[1:]:
        if elemento_actual > mayor_temporal:
            mayor_temporal = elemento_actual
```

```

return mayor_temporal

print("\nProbando encontrar_mayor...\n")

listas_de_prueba = [
    [8, 7, 7, 6, 9],
    [0, -6, -8, -3],
    [24, 24, 24],
    [],
    [7]
]

for lista in listas_de_prueba:
    resultado = encontrar_mayor(lista)
    print(f"Lista: {lista} > Mayor encontrado: {resultado}")

assert encontrar_mayor([8, 7, 7, 6, 9]) == 9
assert encontrar_mayor([0, -6, -8, -3]) == 0
assert encontrar_mayor([24, 24, 24]) == 24
assert encontrar_mayor([]) == None
assert encontrar_mayor([7]) == 7

print("\niPruebas para encontrar_mayor pasaron!...")

print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO")

```

The screenshot displays a code editor with two panes. The left pane contains the Python code for the 'encontrar_mayor' function and its test cases. The right pane shows the terminal output of running the script.

Code (Left Pane):

```

def encontrar_mayor(lista_numeros):
    if not lista_numeros:
        return None

    mayor_temporal = lista_numeros[0]

    for elemento_actual in lista_numeros[1:]:
        if elemento_actual > mayor_temporal:
            mayor_temporal = elemento_actual

    return mayor_temporal

print("\nProbando encontrar_mayor...\n")

listas_de_prueba = [
    [8, 7, 7, 6, 9],
    [0, -6, -8, -3],
    [24, 24, 24],
    [],
    [7]
]

for lista in listas_de_prueba:
    resultado = encontrar_mayor(lista)
    print(f"Lista: {lista} > Mayor encontrado: {resultado}")

assert encontrar_mayor([8, 7, 7, 6, 9]) == 9
assert encontrar_mayor([0, -6, -8, -3]) == 0
assert encontrar_mayor([24, 24, 24]) == 24
assert encontrar_mayor([]) == None
assert encontrar_mayor([7]) == 7

print("\niPruebas para encontrar_mayor pasaron!...")

print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO")

```

Terminal Output (Right Pane):

```

~/workspace$ python3 numero_grande.py
Probando encontrar_mayor...
Lista: [8, 7, 7, 6, 9] > Mayor encontrado: 9
Lista: [0, -6, -8, -3] > Mayor encontrado: 0
Lista: [24, 24, 24] > Mayor encontrado: 24
Lista: [] > Mayor encontrado: None
Lista: [7] > Mayor encontrado: 7

iPruebas para encontrar_mayor pasaron!...

Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO
~/workspace$ python3 numero_grande.py
Probando encontrar_mayor...
Lista: [8, 7, 7, 6, 9] > Mayor encontrado: 9
Lista: [0, -6, -8, -3] > Mayor encontrado: 0
Lista: [24, 24, 24] > Mayor encontrado: 24
Lista: [] > Mayor encontrado: None
Lista: [7] > Mayor encontrado: 7

iPruebas para encontrar_mayor pasaron!...

Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO
~/workspace$

```

EJERCICIO3 CONTADOR DE NÚMEROS

```
def contar_apariciones(lista, elemento_buscado):
```

```
    contador = 0
```

```
    for elemento in lista:
```

```

    if elemento == elemento_buscado:
        contador += 1
    return contador

print("\nProbando contar_apariciones...\n")

lista_ejemplo = [5, 4, 4, 8, 8, 5, 6, 2, 1, 1, 8, 4]
elementos_a_buscar = [5, 4, 1, 8]

for elemento in elementos_a_buscar:
    resultado = contar_apariciones(lista_ejemplo, elemento)
    print(f"Elemento '{elemento}' aparece {resultado} veces en la lista: {lista_ejemplo}")
print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA QUE CUENTA LAS VECES QUE APARECE UN ELEMENTO")

```

The screenshot shows a code editor with a file named `contador_numero.py`. The code defines a function `contar_apariciones` that counts the occurrences of an element in a list. It then tests this function with a list `lista_ejemplo` and a set of elements `elementos_a_buscar`. The terminal window shows the output of running the program, which includes the function definition, the test results for each element, and a final message from Jhoel Ivan Macias Mamani.

```

1 def contar_apariciones(lista, elemento_buscado):
2
3     contador = 0
4
5     for elemento in lista:
6         if elemento == elemento_buscado:
7             contador += 1
8     return contador
9
10 print("\nProbando contar_apariciones...\n")
11
12 lista_ejemplo = [5, 4, 4, 8, 8, 5, 6, 2, 1, 1, 8, 4]
13 elementos_a_buscar = [5, 4, 1, 8]
14
15 for elemento in elementos_a_buscar:
16     resultado = contar_apariciones(lista_ejemplo, elemento)
17     print(f"Elemento '{elemento}' aparece {resultado} veces en la
18 lista: {lista_ejemplo}")
19
20 print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA QUE CUENTA
21 LAS VECES QUE APARECE UN ELEMENTO")

```

```

Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO
~/workspace$ python3 numero_grande.py

Probando encontrar_mayor...

Lista: [8, 7, 7, 6, 9] > Mayor encontrado: 9
Lista: [0, -6, -8, -3] > Mayor encontrado: 0
Lista: [24, 24, 24] > Mayor encontrado: 24
Lista: [] > Mayor encontrado: None
Lista: [7] > Mayor encontrado: 7

¡Pruebas para encontrar_mayor pasaron!

Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA ENCUENTRA EL MAYOR ELEMENTO
~/workspace$ python3 contador_numero.py

Probando contar_apariciones...

Elemento '5' aparece 2 veces en la lista: [5, 4, 4, 8, 8, 5, 6, 2, 1, 1, 8, 4]
Elemento '4' aparece 3 veces en la lista: [5, 4, 4, 8, 8, 5, 6, 2, 1, 1, 8, 4]
Elemento '1' aparece 2 veces en la lista: [5, 4, 4, 8, 8, 5, 6, 2, 1, 1, 8, 4]
Elemento '8' aparece 3 veces en la lista: [5, 4, 4, 8, 8, 5, 6, 2, 1, 1, 8, 4]

Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA QUE CUENTA LAS VECES QUE APARECE UN ELEMENTO
~/workspace$

```

Ejercicio 4 invertir lista

```

def invertir_lista(lista_original):
    lista_invertida = []

    for i in range(len(lista_original) - 1, -1, -1):
        lista_invertida.append(lista_original[i])

    return lista_invertida

print("\nProbando invertir_lista...\n")

lista_prueba = [1, 2, 3, 4, 5, 6, 7]
lista_resultante = invertir_lista(lista_prueba)

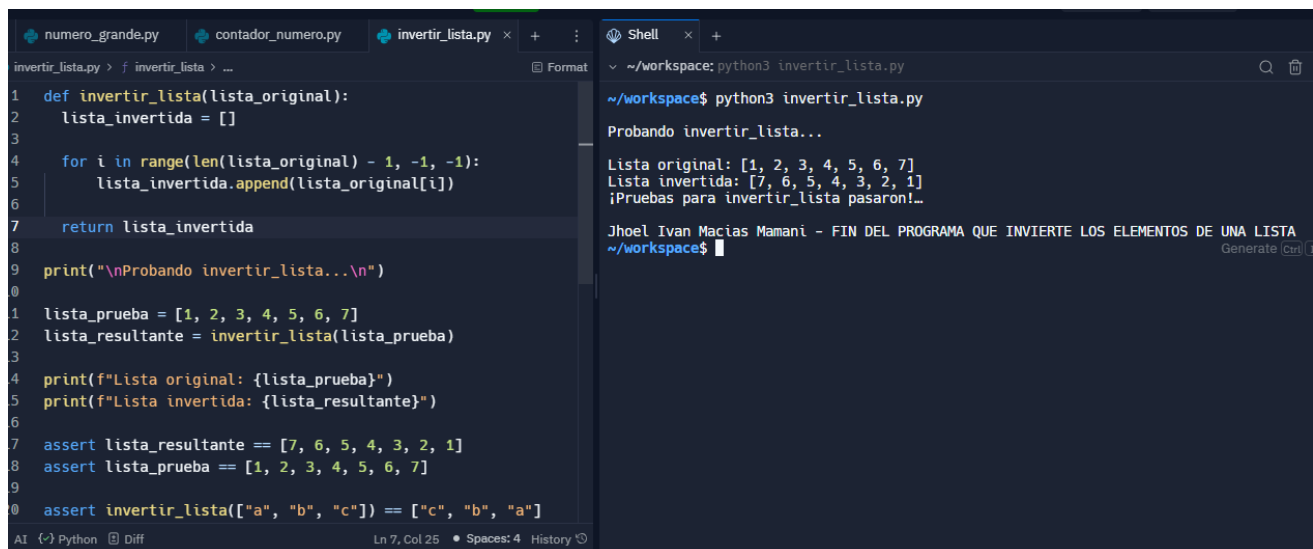
```

```
print(f"Lista original: {lista_prueba}")
print(f"Lista invertida: {lista_resultante}")
```

```
assert lista_resultante == [7, 6, 5, 4, 3, 2, 1]
assert lista_prueba == [1, 2, 3, 4, 5, 6, 7]
```

```
assert invertir_lista(["a", "b", "c"]) == ["c", "b", "a"]
assert invertir_lista([]) == []
```

```
print("¡Pruebas para invertir_lista pasaron!...")
print("\nJhoel Ivan Macias Mamani - FIN DEL PROGRAMA QUE INVIERTE LOS ELEMENTOS DE UNA LISTA")
```



The screenshot shows a code editor with three tabs: 'numero_grande.py', 'contador_numero.py', and 'invertir_lista.py'. The 'invertir_lista.py' tab is active, displaying the following Python code:

```
1 def invertir_lista(lista_original):
2     lista_invertida = []
3
4     for i in range(len(lista_original) - 1, -1, -1):
5         lista_invertida.append(lista_original[i])
6
7     return lista_invertida
8
9 print("\nProbando invertir_lista...\n")
10
11 lista_prueba = [1, 2, 3, 4, 5, 6, 7]
12 lista_resultante = invertir_lista(lista_prueba)
13
14 print(f"Lista original: {lista_prueba}")
15 print(f"Lista invertida: {lista_resultante}")
16
17 assert lista_resultante == [7, 6, 5, 4, 3, 2, 1]
18 assert lista_prueba == [1, 2, 3, 4, 5, 6, 7]
19
20 assert invertir_lista(["a", "b", "c"]) == ["c", "b", "a"]
```

To the right of the code editor is a terminal window titled 'Shell'. It shows the command `python3 invertir_lista.py` being executed. The output of the script is displayed in the terminal:

```
~/workspace$ python3 invertir_lista.py
Probando invertir_lista...
Lista original: [1, 2, 3, 4, 5, 6, 7]
Lista invertida: [7, 6, 5, 4, 3, 2, 1]
¡Pruebas para invertir_lista pasaron!...
Jhoel Ivan Macias Mamani - FIN DEL PROGRAMA QUE INVIERTE LOS ELEMENTOS DE UNA LISTA
~/workspace$
```