EJERCICIO 1

Crear módulo matrices.py e implemente las siguientes funciones:

```
from typing import List, Callable

# Tipo para la matriz
Matriz = List[List[int]]

def crearMatrizHomogenea(filas:int, columnas:int, valor_inicial:int = 0) -> Matriz:
    """ Crear matriz homogénea, de num. filas x columnas, e inicializada con valor_inicial """
    pass

def crearMatriz(datos: Matriz) -> Matriz:
    """ Crear una matriz a partir del parámetro de entrada """
    pass

def es_cuadrada(matriz: Matriz) -> bool:
    """ Verificar si una matriz es cuadrada """
    pass

def es_simetrica(matriz: Matriz) -> bool:
    """ Verificar si una matriz es cuadrada y es simétrica """
    pass

def sumarMatrices(matriz1: Matriz, matriz2: Matriz) -> Matriz:
    """ Sumar 2 matrices cuadradas del mismo tamaño """
    pass

def restarMatrices(matriz1: Matriz, matriz2: Matriz) -> Matriz:
    """ Restar 2 matrices cuadradas del mismo tamaño """
    pass

def multiplicarMatrices(matriz1: Matriz, matriz2: Matriz) -> Matriz:
    """ Multiplicar 2 matrices cuadradas y compatibles para la multiplicación """
    pass

def trasponerMatriz(matriz: Matriz) -> Matriz:
    """ Obtener traspuesta de la matriz cuadradas indicada """
    pass

def trazaMatriz(matriz: Matriz) -> int:
    """ Calcular La traza de una matriz cuadrada """
    pass

def aplicarOperacion(matriz1: Matriz, matriz2: Matriz, operacion:Callable[[Matriz, Matriz],
Matriz]) -> Matriz:
    """ Aplicar una operación matricial recibida como parámetro """
    pass
```

Se facilita el módulo **matrices_test.py** :

```
from matrices import *

if __name__ == '__main__':

    matriz1 = crearMatrizHomogenea(3, 3, 1)

    print("\nMatriz 1:")
    for fila in matriz1:
        print(fila)

    datos1 = [[1, 2, 3],
              [4, 5, 6],
              [7, 8, 9]]

    datos2 = [[4, 5, 6],
              [7, 8, 9],
              [1, 2, 3]]

    datos3 = [[7, 8, 9],
              [1, 2, 3],
              [4, 5, 6]]

    matriz2 = crearMatriz(datos1)
    matriz3 = crearMatriz(datos2)
    matriz4 = crearMatriz(datos3)

    print("\nMatriz 2:")
    for fila in matriz2:
```

```
        print(fila)

    print("\nMatriz 3:")
    for fila in matriz3:
        print(fila)

    print("\nMatriz 4:")
    for fila in matriz4:
        print(fila)

    # Suma de matrices
    matriz_suma = sumarMatrices(matriz2, matriz1)
    print("\nSuma de Matriz 2 y Matriz 1:")
    for fila in matriz_suma:
        print(fila)

    # Resta de matrices
    matriz_resta = restarMatrices(matriz2, matriz1)
    print("\nResta de Matriz 2 y Matriz 1:")
    for fila in matriz_resta:
        print(fila)

    # Multiplicación de matrices
    matriz_multiplicada = multiplicarMatrices(matriz2, matriz1)
    print("\nMultiplicación de Matriz 2 y Matriz 1:")
    for fila in matriz_multiplicada:
        print(fila)

    # Trasposición de matriz
    matriz_traspuesta = trasponerMatriz(matriz2)
    print("\nTraspuesta de Matriz 2:")
    for fila in matriz_traspuesta:
        print(fila)

    print(f"\nTraza de la Matriz2: {trazaMatriz(matriz2)}")

    print("\n ****** Callable ****** ")

    print("\n ** Sumar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **")
    matriz_suma1 = aplicarOperacion(matriz3, matriz4, sumarMatrices)

    for fila in matriz suma1:
        print(fila)

    print("\n ** Restar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **")

    matriz_resta1 = aplicarOperacion(matriz3, matriz4, restarMatrices)

    for fila in matriz_resta1:
        print(fila)

    print("\n ** Multiplicar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **")
    matriz_multiplicada1 = aplicarOperacion(matriz3, matriz4, multiplicarMatrices)

    for fila in matriz_multiplicada1:
        print(fila)
```

Resultados

```
Matriz 1:
[1, 1, 1]
[1, 1, 1]
[1, 1, 1]

Matriz 2:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

Matriz 3:
[4, 5, 6]
[7, 8, 9]
[1, 2, 3]

Matriz 4:
[7, 8, 9]
[1, 2, 3]
[4, 5, 6]

Suma de Matriz 2 y Matriz 1:
[2, 3, 4]
[5, 6, 7]
[8, 9, 10]
```

```
Resta de Matriz 2 y Matriz 1:
[0, 1, 2]
[3, 4, 5]
[6, 7, 8]

Multiplicación de Matriz 2 y Matriz 1:
[6, 6, 6]
[15, 15, 15]
[24, 24, 24]

Traspuesta de Matriz 2:
[1, 4, 7]
[2, 5, 8]
[3, 6, 9]

Traza de la Matriz2: 15

 ****** Callable ******

 ** Sumar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **
[11, 13, 15]
[8, 10, 12]
[5, 7, 9]

 ** Restar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **
[-3, -3, -3]
[6, 6, 6]
[-3, -3, -3]

 ** Multiplicar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **
[57, 72, 87]
[93, 117, 141]
[21, 27, 33]
```