

## EJERCICIO 2

Crear módulo `matricesT.py` e implemente las siguientes funciones:

```
from typing import List, TypeVar, Callable

# Definimos un tipo para el contenido de la matriz
T = TypeVar('T')

# Tipo para la matriz
MatrizT = List[List[T]]

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

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

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

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

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

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

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

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

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

def aplicarOperacion(matriz4: MatrizT, matriz3: MatrizT, operacion:Callable[[MatrizT, MatrizT],
MatrizT]) -> MatrizT:
    """ Aplicar una operación matricial recibida como parámetro """
    pass
```

Se facilita el módulo `matricesT_test.py` :

```
from matricesT import *
from fractions import Fraction

def imprimirMatriz(matriz):
    for fila in matriz:
        print(' '.join(str(elemento) for elemento in fila))

if __name__ == '__main__':

    matriz1 = crearMatrizHomogenea(3, 3)

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

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

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

    matriz2 = crearMatriz(datos1)
    matriz3 = crearMatriz(datos2)
```

```

matriz4 = crearMatriz(datos3)

print("\nMatriz 1:")
imprimirMatriz(matriz1)

print("\nMatriz 2:")
imprimirMatriz(matriz2)

print("\nMatriz 3:")
imprimirMatriz(matriz3)

print("\nMatriz 4:")
imprimirMatriz(matriz4)

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

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

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

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

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

print("\n ** Sumar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **")
matriz_suma2 = aplicarOperacion(matriz3, matriz4, sumarMatrices)
imprimirMatriz(matriz_suma2)

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

matriz_resta2 = aplicarOperacion(matriz3, matriz4, restarMatrices)
imprimirMatriz(matriz_resta2)

print("\n ** Multiplicar Matriz 3 y Matriz 4, usando parámetro de tipo Callable **")
matriz_multiplicada2 = aplicarOperacion(matriz3, matriz4, multiplicarMatrices)

imprimirMatriz(matriz_multiplicada2)

```

## Resultados

Matriz 1:

```

0 0 0
0 0 0
0 0 0

```

Matriz 2:

```

1 2 3
4 5 6
7 8 9

```

Matriz 3:

```

2 5/2 3
7/2 4 9/2
1/2 1 3/2

```

Matriz 4:

```

7/3 8/3 3
1/3 2/3 1
4/3 5/3 2

```

Suma de Matriz 2 y Matriz 1:

```

1 2 3
4 5 6
7 8 9

```

Resta de Matriz 2 y Matriz 1:

```

1 2 3
4 5 6
7 8 9

```

Multiplicación de Matriz 2 y Matriz 1:

```

0 0 0
0 0 0

```

0 0 0

Transpuesta de Matriz 2:

1 4 7  
2 5 8  
3 6 9

Traza de la Matriz2: 15

**\*\* Sumar Matriz 3 y Matriz 4, usando parámetro de tipo Callable \*\***

13/3 31/6 6  
23/6 14/3 11/2  
11/6 8/3 7/2

**\*\* Restar Matriz 3 y Matriz 4, usando parámetro de tipo Callable \*\***

-1/3 -1/6 0  
19/6 10/3 7/2  
-5/6 -2/3 -1/2

**\*\* Multiplicar Matriz 3 y Matriz 4, usando parámetro de tipo Callable \*\***

19/2 12 29/2  
31/2 39/2 47/2  
7/2 9/2 11/2