

## Transposición y Traza

Un analista de datos trabaja con matrices de medidas y necesita frecuentemente transponerlas (convertir filas en columnas) y calcular su traza (suma de la diagonal principal) para ciertos algoritmos de procesamiento.

Historia de usuario Nro.	1	Título:	Transponer matriz y calcular su traza
Descripción	COMO:	Analista de datos	
	QUIERO:	Obtener la matriz transpuesta y la suma de su diagonal principal	
	PARA:	Preparar datos para otros cálculos estadísticos	
Criterios de aceptación	<ul style="list-style-type: none"><li>▪ La matriz es cuadrada de orden n (n filas y n columnas).</li><li>▪ Debe imprimirse primero la matriz transpuesta (cada fila de la transpuesta en una línea).</li><li>▪ Después, imprimir la traza como un número.</li><li>▪ Si <math>n &lt; 1</math> o la entrada no corresponde a una matriz <math>n \times n</math>, imprimir "ERROR".</li></ul>		

Usted es contratado por una empresa de análisis de datos para incorporar esta característica a un programa en Python que cumpla las funcionalidades requeridas por los analistas de datos teniendo como referencia las historias de usuario presentadas previamente.

Entrada	Primera línea: entero $n$ . Sigüientes $n$ líneas: cada una con $n$ enteros separados por espacios.
Salida	Primeras $n$ líneas: filas de la matriz transpuesta. Última línea: traza (suma de $\text{matrix}[i][i]$ ).

## Instrucciones para la calificación automática

Antes de enviar la solución del reto, por favor tenga en cuenta los siguientes aspectos:

- Clase principal: RetoMatriz.
- Método principal: run.
- Leer desde entrada estándar, escribir a salida estándar.
- Sin imprimir textos adicionales.

## Casos de prueba (visibles)

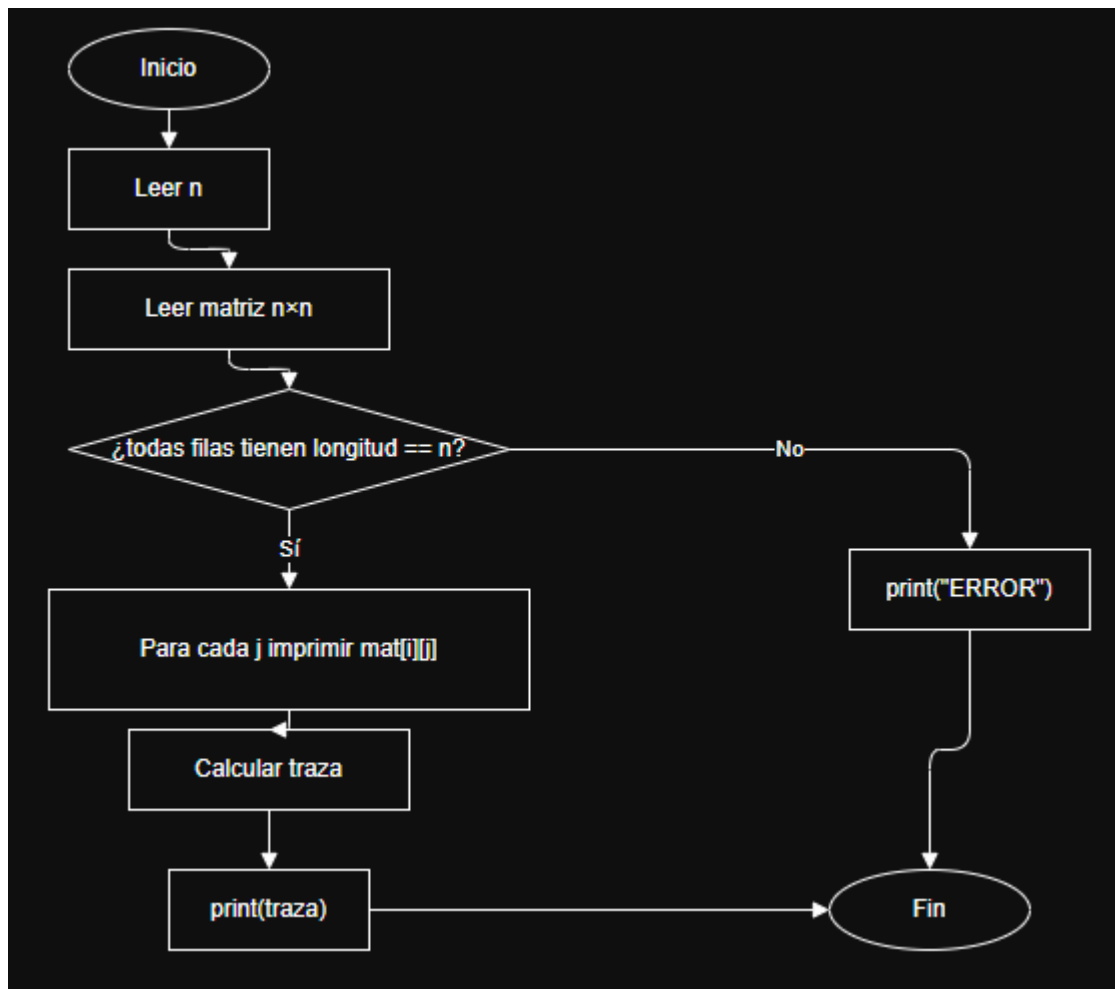
Entradas de ejemplo	3 1 2 3 4 5 6 7 8 9
---------------------	------------------------------

Salida de ejemplo	1 4 7 2 5 8 3 6 9 15
-------------------	-------------------------------

### Casos de prueba (ocultos)

Caso prueba	Entrada	Salida esperada
1	2 5 6 7 8	5 7 6 8 13
2	3 1 2 3 4 5 6	ERROR

### Diseño: Diagrama de flujo



## Código Python :

```
class RetoMatriz:

    def run(self):

        try:

            n = int(input().strip())

            mat = [list(map(int, input().split())) for _ in range(n)]

            if any(len(row) != n for row in mat):

                print("ERROR")

                return

            # Transponer

            for j in range(n):

                print(" ".join(str(mat[i][j]) for i in range(n)))

            # Calcular traza

            traza = sum(mat[i][i] for i in range(n))

            print(traza)

        except Exception:

            print("ERROR")

if __name__ == "__main__":

    RetoMatriz().run()
```

```
if __name__ == "__main__":  
    RetoMatriz().run()  
|
```

⇒ 3  
1 2 3  
4 5 6  
7 8 9  
1 4 7  
2 5 8  
3 6 9  
15

```
if __name__ == "__main__":  
    RetoMatriz().run()  
|
```

⇒ 2  
5 6  
7 8  
5 7  
6 8  
13

```
if __name__ == "__main__":  
    RetoMatriz().run()
```

⇒ 3  
1 2  
3 4  
5 6  
ERROR