



---

**Universidad de las Fuerzas Armadas ESPE**

**Departamento:**

**Carrera: Ingeniería en Electrónica y Automatización**

**Taller académico N°: 4**

**Parcial Nro:2**

---

**1. Información General**

- **Asignatura:** Fundamentos de Programación
  - **Apellidos y nombres de los estudiantes:** Alvear Alexander, Campoverde Anthony, Velecela Mateo
  - **NRC:** 20823
  - **Fecha de realización:** 13/06/2025
- 

**2. Objetivo del Taller y Desarrollo**

**Objetivo del Taller:**

Desarrollar habilidades prácticas en el manejo de requisitos funcionales, pseudocódigo en PseInt, aplicando matrices y lógica para operaciones básicas de recorrido, ordenamiento y manipulación de datos, con énfasis en la eficiencia del código.

**Desarrollo:**

Aprendimos a manejar la inversa de las matrices, resolviendo ejercicios prácticos para la aplicación intuitiva de conocimientos. Al final, discutimos los resultados y reforzamos la lógica.

---

**U2 T4 Matrices**

El ejemplo 3.11 es una ampliación del ejemplo 3.10 en el que se reserva espacio adicional para la matriz *mt*, que es donde se almacenará la traspuesta. Una vez introducida la matriz se utilizan dos bucles *for* para asignar al elemento (j,1) de *mt* el valor almacenado en el elemento (i,j) de matriz. Finalmente se imprimen ambas matrices por pantalla.

- **Requisitos Funcionales**

1. **Ingreso de datos en una matriz original:**

- El programa debe permitir al usuario introducir manualmente los elementos de una matriz bidimensional (matriz), especificando filas y columnas.

2. **Reserva de espacio para la matriz traspuesta:**

- El programa debe crear una segunda matriz (mt) con dimensiones invertidas (columnas por filas) para almacenar la traspuesta de la matriz original.

3. **Transposición de la matriz:**

- Utilizar bucles for anidados para recorrer la matriz original y asignar:
  - $mt[j][i] = matriz[i][j]$
- Es decir, el elemento en la fila i y columna j de la matriz original debe colocarse en la fila j y columna i de la matriz traspuesta.

4. **Impresión de la matriz original:**

- El programa debe mostrar por pantalla la matriz original en forma tabular.

5. **Impresión de la matriz traspuesta:**

- El programa debe mostrar por pantalla la matriz traspuesta generada, también en forma tabular.

- **Seudocódigo**

```
#include <stdio.h>
```

```
int main() {
    int filas, columnas;
    int i, j;

    printf("Ingrese el numero de filas: ");
    scanf("%d", &filas);

    printf("Ingrese el numero de columnas: ");
    scanf("%d", &columnas);

    int matriz[filas][columnas];
    int mt[columnas][filas];

    printf("\nIngrese los elementos de la matriz:\n");
    for (i = 0; i < filas; i++) {
        for (j = 0; j < columnas; j++) {
            printf("matriz[%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    for (i = 0; i < filas; i++) {
        for (j = 0; j < columnas; j++) {
            mt[j][i] = matriz[i][j];
        }
    }
}
```

```
}

printf("\nMatriz original:\n");
for (i = 0; i < filas; i++) {
    for (j = 0; j < columnas; j++) {
        printf("%d\t", matriz[i][j]);
    }
    printf("\n");
}

printf("\nMatriz traspuesta:\n");
for (i = 0; i < columnas; i++) {
    for (j = 0; j < filas; j++) {
        printf("%d\t", mt[i][j]);
    }
    printf("\n");
}

return 0;
}
```