



EDA

ALUMNO SAMUEL ALEJANDRO ESTRELLA LOPEZ

CUARTO CUATRIMESTRE

PRIMER PARCIAL

P4-MATRICES DE 0'S

GRUPO °D

GITHUB:

[HTTPS://GITHUB.COM/SAMUELESTRELLALOPEZ/MAQUETADOS/BLOB/MAIN/EDA_1/MAT2X2/CONSOLEAPP2/PROGRAM.CS](https://github.com/SAMUELESTRELLALOPEZ/Maquetados/blob/main/EDA_1/MAT2X2/CONSOLEAPP2/PROGRAM.CS)

```

{
    case "a":

        int[,] matrizp = new int[,] {
            {0, 2, 5, 7, 6},
            {0, 0, 0, 3, 8},
            {2, 9, 6, 3, 4},
            {1, 5, 6, 1, 4},
            {0, 9, 2, 5, 0}
        };
        Console.WriteLine("\nMatriz:");
        Mostrarmatriz(matrizp);
        for (int i = 0; i < matrizp.GetLength(0); i++)
        {
            int cerosEnFila = ContarCerosEnFila(matrizp, i);
            Console.WriteLine($"En la fila {i + 1} hay {cerosEnFila} ceros.");
        }
        break;

    case "b":

        Console.Clear();
        int[,] matriz = new int[5, 5];
        Console.WriteLine("Ingrese los valores de la matriz (5x5):");
        leermatriz(matriz);
        Console.WriteLine("\nMatriz:");
        Mostrarmatriz(matriz);

        for (int i = 0; i < matriz.GetLength(0); i++)
        {
            int cerosEnFila = ContarCerosEnFila(matriz, i);
            Console.WriteLine($"En el renglón {i + 1} hay {cerosEnFila} ceros.");
        }
        break;

    case "c":
        if (opcion2 == "c")
        {
            break;
        }
        else
        {
            Console.WriteLine("Opción no válida. Por favor, selecciona una opción válida.");
        }
        break;
}
break;

```

```

case 2:
for (int fila = 0; fila < 3; fila++)
{
    for (int columna = 0; columna < 3; columna++)
    {
        while (true)
        {
            Console.WriteLine($"Ingresa el valor para ({fila} {columna}): ");
            if (fila == columna)
            {
                if (int.TryParse(Console.ReadLine(), out int valor) && valor > 0)
                {
                    agregar(fila, columna, valor, 0);
                    break;
                }
                else
                {
                    Console.WriteLine("El numero de la diagonal debe ser mayor que 0");
                }
            }
            else
            {
                if (int.TryParse(Console.ReadLine(), out int valor))
                {
                    agregar(fila, columna, valor, 0);
                    break;
                }
                else
                {
                    Console.WriteLine("Eso no es un numero");
                }
            }
        }
    }
}

Console.Clear();

Console.WriteLine("Matriz original:");

```

```

static void LeerMatriz(int[,] matrices)
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            Console.Write($"Elemento [{i + 0},{j + 0}]: ");
            string input = Console.ReadLine();

            if (input.Contains(", "))
            {
                Console.WriteLine(";Error! Por favor, ingresa valores válidos.");
                j--;
                continue;
            }

            matrices[i, j] = int.Parse(input);
        }
    }
}

```

2 referencias

```

static void Mostrarmatriz(int[,] matrices)
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            Console.Write(matrices[i, j] + " ");
        }
        Console.WriteLine();
    }
}

```

// Método para contar ceros en una fila de la matriz

2 referencias

```

static int ContarCerosEnFila(int[,] matriz, int fila)
{
    int contador = 0;
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        if (matriz[fila, j] == 0)
        {
            contador++;
        }
    }
    return contador;
}

```

```

public void soloMostrar(int idMatriz, int dimension)
{
    for (int fila = 0; fila < dimension; fila++)
    {
        Console.Write(" ");
        for (int columna = 0; columna < dimension; columna++)
        {
            Console.Write(matrices[idMatriz][fila, columna] + " ");
        }
        Console.WriteLine("");
    }
}

```

1 referencia

```

public void cerodiagonal()
{
    for (int fila = 0; fila < 3; fila++)
    {
        // Paso 1: Obtener el pivote de la fila actual
        int pivote = matrices[0][fila, fila];

        if (pivote == 0)
        {
            Console.WriteLine("Error: El pivote es cero. La matriz es singular.");
            return;
        }

        // Paso 2: Dividir todos los elementos de la fila por el pivote
        for (int columna = 0; columna < 3; columna++)
        {
            matrices[0][fila, columna] /= pivote;
        }

        // Paso 3: Hacer ceros en otras filas
        for (int filas = 0; filas < 3; filas++)
        {
            // Salta la fila actual
            if (filas != fila)
            {
                int factor = matrices[0][filas, fila];

                for (int columna = 0; columna < 3; columna++)
                {
                    matrices[0][filas, columna] -= factor * matrices[0][fila, columna];
                }
            }
        }
    }
}

```

Este es el método de eliminación de Gauss-Jordan

```
1 referencia
public void czerodiagonal()
{
    for (int fila = 0; fila < 3; fila++)
    {
        // Paso 1: Obtener el pivote de la fila actual
        int pivote = matrices[0][fila, fila];

        if (pivote == 0)
        {
            Console.WriteLine("Error: El pivote es cero. La matriz es singular.");
            return;
        }

        // Paso 2: Dividir todos los elementos de la fila por el pivote
        for (int columna = 0; columna < 3; columna++)
        {
            matrices[0][fila, columna] /= pivote;
        }

        // Paso 3: Hacer ceros en otras filas
        for (int filas = 0; filas < 3; filas++)
        {
            // Salta la fila actual
            if (filas != fila)
            {
                int factor = matrices[0][filas, fila];

                for (int columna = 0; columna < 3; columna++)
                {
                    matrices[0][filas, columna] -= factor * matrices[0][fila, columna];
                }
            }
        }

        Console.WriteLine($"Paso {fila + 1}:");
        soloMostrar(0, 3);
    }
}
```

```
C:\Users\salex\Documents\EC  X + v
----Selecciona una opcion:----
1. Calcular ceros
2. matriz cuadrada de números enteros
3. Salir
|

C:\Users\salex\Documents\EC  X + v
----Selecciona una opcion para calcular ceros en filas:----
a. Calcular Matriz predeterminada
b. Ingresar valores de la matriz 5X5
c. salir
|
```

```
C:\Users\salex\Documents\EC  X + v
1 referencia
public void
{
    for (
    {
        Matriz:
        0 2 5 7 6
        0 0 0 3 8
        2 9 6 3 4
        1 5 6 1 4
        0 9 2 5 0
        En la fila 1 hay 1 ceros.
        En la fila 2 hay 3 ceros.
        En la fila 3 hay 0 ceros.
        En la fila 4 hay 0 ceros.
        En la fila 5 hay 2 ceros.
        ----Selecciona una opcion:----
        1. Calcular ceros
        2. matriz cuadrada de números enteros
        3. Salir
        |
    }
}
```

```
C:\Users\sarex\Documents\CL... x 1 0
Matriz original
[ 2  4  6 ]
[ 8 10  2 ]
[ 4  6  8 ]
Paso 1:
[ 1  2  3 ]
[ 0 -6 -22 ]
[ 0 -2  -4 ]
Paso 2:
[ 1  0 -3 ]
[ 0  1  3 ]
[ 0  0  2 ]
Paso 3:
[ 1  0  0 ]
[ 0  1  0 ]
[ 0  0  1 ]
----Selecciona una opcion:----
1. Calcular ceros
2. matriz cuadrada de números enteros
3. Salir
|
encia
ic ve
For (
{
}
```