

```

int opc;

char uno = 's', dos = 'S';

bool canConvert = false;

double[,] matrizLlena1;

double[,] matrizLlena2;

double[,] matrizLlenaResultado;

StreamReader archivoAbierto;

string linea;

StreamWriter archivoNuevo;

archivoNuevo = File.AppendText("prueba.txt");

do
{
    do
    {
        Console.Clear();

        Console.WriteLine("<<<<<<<< Programa que realice las siguientes operaciones con  
Matrices: Suma, Resta y Multiplicación de Matrices. >>>>>>>>>>>>> ");

        Console.WriteLine("1.- Suma");

        Console.WriteLine("2.- Resta ");

        Console.WriteLine("3.-Multiplicación ");

        Console.WriteLine("4.- Mostrar Datos Almacenados en Archivo de Matrices");

        Console.WriteLine("5.- Salir del Programa");

        opc = int.Parse(Console.ReadLine());

        Console.WriteLine(" ");

        switch (opc)
        {
            case 1:
                Console.Clear();

                do

```

```

{
    Console.Clear();
    Console.WriteLine("la Matriz A ");
        archivoNuevo.WriteLine("la Matriz A ");
        matrizllena1 = llenarmatriz(archivoNuevo);
    Console.WriteLine("la Matriz B ");
        archivoNuevo.WriteLine(" ");
        archivoNuevo.WriteLine("la Matriz B ");
        matrizllena2 = llenarmatriz(archivoNuevo);
        canConvert = validaciondematrices(matrizllena1, matrizllena2);
} while (canConvert != true);
Console.Clear();
matrizllenaresultado = sumallenarmatriz(matrizllena1, matrizllena2, archivoNuevo);
resultado(matrizllenaresultado, archivoNuevo);
    Console.WriteLine(" ");
    Console.WriteLine("quieres reguresar al menu s/ n ");
dos = char.ToUpper(char.Parse(Console.ReadLine()));
if (dos== 'N')
{
    Console.WriteLine("gracias por usar este progama hasta luego");
    System.Threading.Thread.Sleep(2000);
    uno = 'N';
}
archivoNuevo.Close();
break;
case 2:
    Console.Clear();
do
{

```

```

        Console.Clear();

        Console.WriteLine("la Matriz A ");
        archivoNuevo.WriteLine("la Matriz A ");
        matrizllena1 = llenarmatriz(archivoNuevo);

        Console.WriteLine("la Matriz B ");
        archivoNuevo.WriteLine(" ");
        archivoNuevo.WriteLine("la Matriz B ");
        matrizllena2 = llenarmatriz(archivoNuevo);

        canConvert = validaciondematrices(matrizllena1, matrizllena2);
    } while (canConvert != true);

    Console.Clear();

    matrizllenaresultado = restallarmatriz(matrizllena1, matrizllena2,
archivoNuevo);

    resultado(matrizllenaresultado, archivoNuevo);

    Console.WriteLine(" ");

    Console.WriteLine("quieres regresar al menu s/ n ");
    dos = char.ToUpper(char.Parse(Console.ReadLine()));

    if (dos == 'N')
    {
        Console.WriteLine("gracias por usar este programa hasta luego");
        System.Threading.Thread.Sleep(2000);

        uno = 'N';
    }

    archivoNuevo.Close();

    break;

case 3:

    Console.Clear();

    do
    {

```

```

        Console.Clear();

        Console.WriteLine("la Matriz A ");
        archivoNuevo.WriteLine("la Matriz A ");
        matrizllena1 = llenarmatriz(archivoNuevo);

        Console.WriteLine("la Matriz B ");
        archivoNuevo.WriteLine(" ");
        archivoNuevo.WriteLine("la Matriz B ");
        matrizllena2 = llenarmatriz(archivoNuevo);

        canConvert = validaciondematrices(matrizllena1, matrizllena2);
    } while (canConvert != true);

    Console.Clear();

    matrizllenaresultado = multillenarmatriz(matrizllena1, matrizllena2,
archivoNuevo);

    resultado(matrizllenaresultado, archivoNuevo);

    Console.WriteLine(" ");

    Console.WriteLine("quieres reguresar al menu s/ n ");
    dos = char.ToUpper(char.Parse(Console.ReadLine()));

    if (dos == 'N')
    {
        Console.WriteLine("gracias por usar este progama hasta luego");

        System.Threading.Thread.Sleep(2000);

        uno = 'N';
    }

    archivoNuevo.Close();

    break;

case 4:

    try
    {
        archivoAbierto = File.OpenText("prueba.txt");

```

```

do
{
    linea = archivoAbierto.ReadLine();

    Console.WriteLine(linea);
} while (linea != null);
if (linea != null)
{
    Console.WriteLine("el Archivo  esta vacio");
}
archivoAbierto.Close();
}
catch (Exception exp)
{
    Console.WriteLine("Hay un error con el Archivo aun no esta cerrado o no se a
creado");

    Console.WriteLine(exp.Message);
}

Console.WriteLine("quieres reguesar al menu s/ n ");
dos = char.ToUpper(char.Parse(Console.ReadLine()));
if (dos == 'N')
{
    Console.WriteLine("gracias por usar este progama hasta luego");
    System.Threading.Thread.Sleep(2000);
    uno = 'N';
}

break;
case 5:

```

```

        Console.WriteLine("gracias por usar este progama hasta luego");

        System.Threading.Thread.Sleep(2000);

        uno = 'N';

        break;

    default:

        Console.WriteLine("error");

        break;

    }

    } while (dos == 'S');

    System.Threading.Thread.Sleep(1000);

} while (uno == 'S');

}

//

//operaciones

//

public static double[,] llenarmatriz(StreamWriter archivoNuevo)
{
    int n1, n2;

    n1 = validacion();

    n2 = validacion();

    double[,] matriz = new double[n1, n2];

    for (int f = 0; f < matriz.GetLength(0); f++)
    {
        archivoNuevo.WriteLine(" ");

        for (int c = 0; c < matriz.GetLength(1); c++)
        {
            Console.Write("Introduce los numeros {0} ,{1} : ", f, c);

```

```

        matriz[f, c] = double.Parse(Console.ReadLine());

        archivoNuevo.Write(matriz[f, c]+" ");
    }
}

return (matriz);
}

public static double[,] sumallenarmatriz(double[,] matrizllena1, double[,] matrizllena2,
StreamWriter archivoNuevo)
{
    double[,] matriz = new double[matrizllena1.GetLength(0), matrizllena1.GetLength(1)];

    for (int f = 0; f < matrizllena1.GetLength(0); f++)
    {
        archivoNuevo.WriteLine(" ");
        for (int c = 0; c < matrizllena1.GetLength(1); c++)
        {
            archivoNuevo.Write("(" + matrizllena1[f, c] + "+" + matrizllena2[f, c] + ")");
            matriz[f, c] = matrizllena1[f, c] + matrizllena2[f, c];
        }
    }

    return (matriz);
}

public static double[,] restallenarmatriz(double[,] matrizllena1, double[,] matrizllena2,
StreamWriter archivoNuevo)
{
    double[,] matriz = new double[matrizllena1.GetLength(0), matrizllena1.GetLength(1)];

    for (int f = 0; f < matrizllena1.GetLength(0); f++)
    {
        archivoNuevo.WriteLine(" ");
    }
}

```

```

        for (int c = 0; c < matrizllena1.GetLength(1); c++)
        {
            archivoNuevo.Write("(" + matrizllena1[f, c] + "-" + matrizllena2[f, c] + ")");
            matriz[f, c] = matrizllena1[f, c] - matrizllena2[f, c];
        }
    }
    return (matriz);
}

public static double[,] multillenarmatriz(double[,] matrizllena1, double[,] matrizllena2,
StreamWriter archivoNuevo)
{
    double[,] matriz = new double[matrizllena1.GetLength(0), matrizllena1.GetLength(1)];
    for (int f = 0; f < matrizllena1.GetLength(0); f++)
    {
        for (int c = 0; c < matrizllena2.GetLength(1); c++)
        {
            matriz[f, c]=0;
            archivoNuevo.WriteLine(" ");
            for (int z = 0; z < matrizllena2.GetLength(0); z++)
            {
                archivoNuevo.Write("(" + matrizllena1[f, c] + "*" + matrizllena2[f, c] + "+" + matriz[f,
c] + ")");
                matriz[f, c] = matrizllena1[f, z] * matrizllena1[z, c]+ matriz[f, c];
            }
        }
    }
    return (matriz);
}

public static void resultado(double[,] matrizllena1, StreamWriter archivoNuevo)
{

```



```

Console.WriteLine("el resultado de la Matriz A y Matriz B es :");
archivoNuevo.WriteLine(" ");
archivoNuevo.WriteLine("el resultad de la Matriz A y Matriz B es : ");
for (int f = 0; f < matrizllena1.GetLength(0); f++)
{
    archivoNuevo.WriteLine(" ");
    Console.WriteLine( );
    for (int c = 0; c < matrizllena1.GetLength(1); c++)
    {
        archivoNuevo.Write(matrizllena1[f, c] + " ");
        Console.Write(matrizllena1[f, c] + " ");
    }
}
Console.ReadKey();
}

public static int validacion ()
{
    string cadenaEvaluar;
    bool canConvert;
    int cal;
    do
    {
        Console.WriteLine("introducce el tamaño:");
        cadenaEvaluar = Console.ReadLine();
        canConvert = int.TryParse(cadenaEvaluar, out cal);
        if (canConvert == true)
        {

        }
    }
}

```

```

        else
        {
            Console.WriteLine("Introduca un numero valido Incorrectamente");
        }
    } while (canConvert != true);
    return (cal);
}

public static bool validaciondematrices (double[,] matrizllena1, double[,] matrizllena2)
{
    bool canConvert;

    if (matrizllena1.GetLength(0) == matrizllena2.GetLength(0) && matrizllena1.GetLength(1) ==
matrizllena2.GetLength(1))
    {
        canConvert = true;
    }
    else
    {
        Console.WriteLine("incorrecto las matrices son diferentes ");
        canConvert = false;
    }

    System.Threading.Thread.Sleep(1000);

    return (canConvert);
}

public static bool validaciondematricesmulti(double[,] matrizllena1, double[,] matrizllena2)
{
    bool canConvert;

    if (matrizllena1.GetLength(0) == matrizllena2.GetLength(1) )
    {
        canConvert = true;
    }
}

```

```
else
{
    Console.WriteLine("incorrecto las matrices son diferentes ");
    canConvert = false;
}
System.Threading.Thread.Sleep(1000);
return (canConvert);
}
```