

Taller 1 – David Barros:

Teoría:

- ¿Qué es un ciclo infinito? Escriba un ejemplo:

RTA:

Consiste de un bucle que no tiene una condición de salida, lo que significa que continuará ejecutándose indefinidamente. Este tipo de bucles pueden causar problemas graves, ya que el programa nunca avanza más allá de esa parte del código.

Un ejemplo podría ser el siguiente:



```
while True:  
print("Este es un ciclo infinito")
```



En esta situación, el bucle *"while True"* siempre se evalúa como verdadero, por lo que el programa imprimirá *"Este es un ciclo infinito"* una y otra vez sin detenerse. La mayor amenaza de un bucle infinito es que se consume recursos de manera infinita y esto puede descontrolar a una aplicación totalmente.

- ¿Cuál es la diferencia entre la longitud de un array y la de una cadena?:

RTA:

La principal diferencia radica en el tipo de dato que están representando. La longitud de un array tiene que ver con el número de elementos en ese mismo elemento, mientras que la longitud de una cadena se refiere al número de caracteres en la misma.

Además, la longitud de una cadena puede cambiar dinámicamente, mientras que la de un array generalmente es fija.

Ejemplo de un array:



```
int[] miArray = { 1, 2, 3, 4, 5 };  
int longitudArray = miArray.Length; // longitudArray sería 5
```



Ejemplo de una cadena:



```
string miCadena = "Hola, mundo!";  
int longitudCadena = miCadena.Length; // longitudCadena sería 12
```



Ejercicios:

Realice un programa que:

- 1. Recorra una matriz nxm y reemplace los valores negativos por un número positivo, como muestra el ejemplo:

-1	2	-3
6	-8	9
-3	4	-5

 ->

1	2	3
6	8	9
3	4	5



```
Program.cs
C# Ejercicio 1

1 using System;
2
3 class Program
4 {
5     static void Main()
6     {
7         // Tamaño de la matriz (puedes ajustarlo según tus necesidades):
8
9         int filas = 3;
10        int columnas = 4;
11
12        // Crear y llenar la matriz con valores de ejemplo:
13
14        int[,] matriz = new int[,]
15        {
16            { 1, -2, 3, -4 },
17            { 5, 6, -7, 8 },
18            { -9, 10, -11, 12 }
19        };
20
21        // Número positivo con el que reemplazaremos los valores negativos:
22
23        int numeroPositivo = 999;
24
25        // Mostrar la matriz original:
26
27        Console.WriteLine("Matriz Original:");
28        ImprimirMatriz(matriz);
29
30        // Recorrer la matriz y reemplazar los valores negativos:
31
32        for (int i = 0; i < filas; i++)
33        {
34            for (int j = 0; j < columnas; j++)
35            {
36                if (matriz[i, j] < 0)
37                {
38                    matriz[i, j] = numeroPositivo;
39                }
40            }
41        }
42
43        // Mostrar la matriz después de reemplazar los valores negativos:
44
45        Console.WriteLine("\nMatriz con Valores Negativos Reemplazados:");
46        ImprimirMatriz(matriz);
47
48        // Método para imprimir la matriz:
49
50        static void ImprimirMatriz(int[,] matriz)
51        {
52            for (int i = 0; i < matriz.GetLength(0); i++)
53            {
54                for (int j = 0; j < matriz.GetLength(1); j++)
55                {
56                    Console.Write(matriz[i, j] + "\t");
57                }
58                Console.WriteLine();
59            }
60        }
61    }
62 }
```

```
Consola de depuración de Microsoft Visual Studio

Matriz Original:
1      -2      3      -4
5       6      -7      8
-9     10     -11     12

Matriz con Valores Negativos Reemplazados:
1     999      3     999
5       6     999      8
999    10     999     12

E:\Semestre 2024\Scripting\Taller 1\Ejercicio 1\Ejercicio 1\bin\Debug\net6.0\Ejercicio 1.exe (proceso 10312) se cerró co
n el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```



- 3. Dada una matriz de cadenas de longitud $n \times m$, cree una cadena concatenando sus celdas, recorriendo la matriz de izquierda a derecha y de arriba hacia abajo.



```
Ejercicio 2
1 using System;
2
3 class Program
4 {
5     static void Main()
6     {
7         // Tamaño de la matriz (puedes ajustarlo según tus necesidades)
8         int filas = 3;
9         int columnas = 4;
10
11         // Crear y llenar la matriz con valores de ejemplo
12         string[,] matriz = new string[filas, columnas];
13         {
14             { "A", "B", "C", "D" },
15             { "E", "F", "G", "H" },
16             { "I", "J", "K", "L" }
17         };
18
19         // Concatenar las celdas de la matriz
20         string resultado = ConcatenarMatriz(matriz);
21
22         // Mostrar el resultado
23         Console.WriteLine("Matriz Concatenada:");
24         Console.WriteLine(resultado);
25     }
26
27     // Método para concatenar las celdas de la matriz
28     static string ConcatenarMatriz(string[,] matriz)
29     {
30         int filas = matriz.GetLength(0);
31         int columnas = matriz.GetLength(1);
32
33         // Usar un StringBuilder para eficiencia en la concatenación de cadenas
34         System.Text.StringBuilder resultadoBuilder = new System.Text.StringBuilder();
35
36         for (int i = 0; i < filas; i++)
37         {
38             for (int j = 0; j < columnas; j++)
39             {
40                 resultadoBuilder.Append(matriz[i, j]);
41             }
42         }
43
44         // Devolver la cadena resultante
45         return resultadoBuilder.ToString();
46     }
47 }
48
```

```
Consola de depuración de Microsoft Visual Studio
Matriz Concatenada:
ABCDEFGHIJKL

E:\Semestre 2024\Scripting\Taller 1\Ejercicio 1\Ejercicio 2\Ejercicio 2\bin\Debug\net6.0\Ejercicio 2.exe (proceso 4148)
se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```



- 6. Usando un ciclo do-while calcule la suma de los primeros n números impares.



```
Program.cs  X
Ejercicio 3
1  using System;
2
3  0 referencias
4  class Program
5  {
6      0 referencias
7      static void Main()
8      {
9          // Solicitar al usuario ingresar el valor de n
10         Console.Write("Ingrese el valor de n: ");
11         int n = int.Parse(Console.ReadLine());
12
13         // Calcular la suma de los primeros n números impares usando un ciclo do-while
14         int suma = CalcularSumaImpares(n);
15
16         // Mostrar el resultado
17         Console.WriteLine($"La suma de los primeros {n} números impares es: {suma}");
18     }
19
20     // Método para calcular la suma de los primeros n números impares
21     1 referencia
22     static int CalcularSumaImpares(int n)
23     {
24         int suma = 0;
25         int contador = 1; // Inicializar con el primer número impar
26
27         do
28         {
29             suma += contador;
30             contador += 2; // Mover al siguiente número impar
31             n--; // Decrementar n para controlar la cantidad de números impares a sumar
32         } while (n > 0);
33
34         return suma;
35     }
36 }
```

```
Consola de depuración de Microsoft Visual Studio
- □ X
Ingrese el valor de n: 6
La suma de los primeros 6 números impares es: 36

E:\Semestre 2024\Scripting\Taller 1\Ejercicios\Ejercicio 3\Ejercicio 3\bin\Debug\net6.0\Ejercicio 3.exe (proceso 16672)
se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```





- 15. Realizar una función que calcule si una cadena es palindroma.



```
Program.cs  + X
C# Ejercicio 4
1  using System;
2
3  0 referencias
4  class Program
5  {
6      0 referencias
7      static void Main()
8      {
9          // Solicitar al usuario ingresar una cadena
10         Console.WriteLine("Ingrese una cadena: ");
11         string input = Console.ReadLine();
12
13         // Verificar si la cadena es un palindromo
14         bool esPalindromo = EsPalindromo(input);
15
16         // Mostrar el resultado
17         if (esPalindromo)
18         {
19             Console.WriteLine($"La cadena '{input}' es un palíndromo.");
20         }
21         else
22         {
23             Console.WriteLine($"La cadena '{input}' no es un palindromo.");
24         }
25     }
26
27     // Función para verificar si una cadena es un palindromo
28     1 referencia
29     static bool EsPalindromo(string cadena)
30     {
31         // Convertir la cadena a minúsculas y quitar espacios en blanco
32         cadena = cadena.ToLower().Replace(" ", "");
33
34         int longitud = cadena.Length;
35
36         // Verificar si la cadena es un palindromo comparándola con su inversa
37         for (int i = 0; i < longitud / 2; i++)
38         {
39             if (cadena[i] != cadena[longitud - 1 - i])
40             {
41                 return false; // No es un palindromo
42             }
43         }
44
45         return true; // Es un palindromo
46     }
47 }
48
49
```

```
Consola de depuración de Microsoft Visual Studio
Ingrese una cadena: anitalavalatina
La cadena 'anitalavalatina' es un palindromo.

E:\Semestre 2024\Scripting\Taller 1\Ejercicios\Ejercicio 4\Ejercicio 4\bin\Debug\net6.0\Ejercicio 4.exe (proceso 3840) s
e cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```



- 19. Realice una función que invierta un array de números enteros



```
Program.cs  + x
C# Ejercicio 5

0 referencias
class Program
{
    0 referencias
    static void Main()
    {
        // Ejemplo de un array de números enteros
        int[] numeros = { 1, 2, 3, 4, 5 };

        // Mostrar el array original:
        Console.WriteLine("Array Original:");
        ImprimirArray(numeros);

        // Invertir el array utilizando la función InvertirArray:
        InvertirArray(numeros);

        // Mostrar el array después de invertirlo
        Console.WriteLine("\nArray Invertido:");
        ImprimirArray(numeros);
    }

    // Función para invertir un array de números enteros:

    1 referencia
    static void InvertirArray(int[] array)
    {
        int longitud = array.Length;
        int i = 0;
        int j = longitud - 1;

        // Intercambiar elementos desde los extremos hacia el centro:
        while (i < j)
        {
            int temp = array[i];
            array[i] = array[j];
            array[j] = temp;

            i++;
            j--;
        }
    }

    // Método para imprimir un array:

    2 referencias
    static void ImprimirArray(int[] array)
    {
        foreach (var numero in array)
        {
            Console.Write(numero + " ");
        }
        Console.WriteLine();
    }
}
```

```
Consola de depuración de Microsoft Visual Studio

Array Original:
1 2 3 4 5

Array Invertido:
5 4 3 2 1

E:\Semestre 2024\Scripting\Taller 1\Ejercicios\Ejercicio 5\Ejercicio 5\bin\Debug\net6.0\Ejercicio 5.exe (proceso 11796)
se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```