

Clase 9

Jueves, 28 Septiembre 2023

[Indice](#)

Concatenación de strings

- **FUNCIÓN:** Devuelve la concatenación de dos strings.

```
,
public static string Concat(string a, string b)
{
    return a + b;
},
```

- **FUNCIÓN:** devuelve una serie de números.

```
,
public static string Concat(int number)
{
    string result = "0";
    for(int i = 0; i < number; i++)
    {
        result += "," + i;
    }
    return result;
},
```

- **FUNCIÓN:** devuelve una serie de números.

```
,
public static string Concat(int number)
{
    string result = "0";
    int multiplicador = 1;
    for(int i = 0; i < number; i++)
    {
        multiplicador *= 2;
        result += "," + multiplicador;
    }
    return result;
},
```

- **FUNCIÓN:** realiza distintas concatenaciones con variables.

```
,
public static void Concatenate(string text1, string text2, string text3)
{
    Console.WriteLine("Frase1 , 2 y 3");
    string result1 = (text1 + ", " + text2 + ", " + text3);

    Console.WriteLine("Frase1 y 2");
    string result2 = ("{"0} , {1}", text1, text2);

    Console.WriteLine("Frase2 y 3");
    string result3 = ("{"text2} , {"text3}");
}
,
```

-

```
,
public static void SingularText(string text1)
{
    Console.WriteLine("@Todo este texto es literal / & + text1. Se consigue con @");
    Console.WriteLine("El texto que inicia con \"\$\" permite introducir variables dentro del formato");
    Console.WriteLine("Este texto escapa \"con barras inclinadas\" (\\)\");
    Console.WriteLine("El parametro \"\\r\\n\" sirve para mover el cursor a la izquierda del todo");
    Console.WriteLine("El parametro \"\\n\\n\" sirve para cambiar de linea");
    Console.WriteLine("El parametro \"\\b\" crea un espacio");
    Console.WriteLine("El parametro \"\\t\" crea un tabulado");
}
,
```

-

```
,
public static void StringReplace(string text, string word, string replace)
{
    Console.WriteLine("El metodo \"string.Replace(texto, alternative)\" reemplaza el 'texto' por un 'alternative'");
    string result = text.Replace(word, replace);
    Console.WriteLine(result);
}
,
```

-

```
,
public static void StringIndexOf(string text)
{
    Console.WriteLine("El metodo \"string.IndexOf(text)\" nos indica la posicion del string 'text'");
    int result = text.IndexOf("=");
    Console.WriteLine(text);
    Console.WriteLine("El = ocupa la posicion numero: {0}", result);
}
,
```

- ```

 ,
 public static void StringSubstring(string text, int index1, int index2)
 {
 Console.WriteLine("El metodo \"string.Substring(index1, index2)\" devuelve el texto que se encuentra entre index1 y index2")
 Console.WriteLine(text);
 Console.WriteLine(text.Substring(index1, index2));
 Console.WriteLine(text.Substring(index1));
 }
 ,

```
- ```

      ,
      public static void StringRemove(string text)
      {
          Console.WriteLine(text);

          string result = text.Remove(5);
          Console.WriteLine(result);

          result = text.Remove(5, 7);
          Console.WriteLine(result);
      }
      ,
      
```
- ```

 ,
 public static void StringChangeCase(string text)
 {
 Console.WriteLine(text);
 Console.WriteLine("Puedes devolver todo el texto en minusculas: " + text.ToLower());
 Console.WriteLine("Puedes devolver todo el texto en mayusculas: " + text.ToUpper());
 }
 ,

```
- ```

      ,
      public static void StringTrim(string text)
      {
          Console.WriteLine("La funcion \"text.Trim()\" elimina los espacios en blanco antes y despues de los caracteres");
      }
      ,
      
```

- **Notas:**

- Es recomendable empezar los **FOR** siempre desde *int i = 0*
- No se deben pervertir los valores que toma *i* en el cuerpo del **FOR**
- El tipo 'string' no soporta acumular el operador con *= o /=