

Tema 8: Ficheros

- ☐ Conceptos básicos.
- ☐ La clase StreamWriter.
- ☐ La clase StreamReader.



Conceptos básicos

- ❑ Para poder trabajar con ficheros deberemos utilizar el espacio de nombres System.IO
- ❑ **System.IO:** Las clases incluidas en System.IO, nos van a permitir trabajar con archivos de texto y binarios así como la creación y manipulación de los archivos y directorios.
- ❑ **Objetos Stream:** Un objeto Stream representa un flujo o corriente de datos, es decir, un conjunto de información guardada en formato de texto o binario, que podremos leer y escribir sobre un soporte físico.
- ❑ En función del tipo de operación utilizaremos un Stream de escritura o de lectura.

La clase StreamWriter

- ❑ **La clase StreamWriter:** Un objeto StreamWriter realiza operaciones de escritura de texto sobre un archivo.
- ❑ El proceso de escritura de datos comprende los siguientes pasos:
 - ✓ Instanciar un objeto de esta clase mediante alguno de los constructores disponibles, creando un nuevo archivo para escribir datos abriendo uno existente.
 - ✓ Escritura de texto mediante los métodos WriteLine() y Write().
 - ✓ El primero escribe el texto pasado como parámetro, y añade los caracteres especiales de retorno de carro y nueva línea.
 - ✓ El segundo escribe el texto pasado y deja el puntero de escritura a partir del último carácter escrito, con lo que no produce un cambio automático de línea. Debemos utilizar la propiedad NewLine para introducir manualmente un salto de línea.
 - ✓ Cierre del Stream con el método Close(). Esta acción vuelca el contenido del búfer del objeto en el archivo.

La clase StreamWriter (II)

❑ Si el archivo existe podemos usarlo para añadir al final o para sobrescribir completamente:

'abre el archivo y se sitúa al final del texto para añadir
`swEscritor = New StreamWriter("c:\prueba.txt", True)`

'se elimina el contenido previo del archivo
`swEscritor = New StreamWriter("c:\prueba.txt", False)`

La clase StreamWriter (III)

```
Imports System.IO
Module Module1
    Sub Main()
        Dim swEscrivor As StreamWriter
        ' creamos un stream de escritura, y al mismo tiempo un nuevo archivo para escribir texto
        swEscrivor = New StreamWriter("c:\prueba.txt")

        ' escribir líneas
        swEscrivor.WriteLine("esta es la primera línea")
        swEscrivor.WriteLine("segunda línea de texto")

        ' ahora escribimos texto pero sin provocar un salto de línea
        swEscrivor.Write("línea ")
        swEscrivor.Write("tercera")

        ' esto introduce el salto de línea
        swEscrivor.Write(swEscrivor.NewLine)
        swEscrivor.WriteLine("con esta línea cerramos")

        ' cerrar el stream y el archivo asociado
        swEscrivor.Close()
        Console.WriteLine("Fichero creado. Pulsa enter ....")
        Console.ReadLine()
    End Sub
End Module
```

La clase StreamReader

- ❑ **La clase StreamReader:** Un objeto StreamReader realiza operaciones de lectura de texto sobre un archivo.
- ❑ El proceso que debemos llevar a cabo para leer el contenido de un Stream de lectura es muy similar al de escritura.
- ❑ Los métodos de lectura son ReadLine(), que devuelve una línea del archivo; y ReadToEnd(), que devuelve el resto del contenido del archivo, desde el punto en el que se encontrara el Stream al realizar la última lectura

La clase StreamReader (II)

```
Imports System.IO
Module Module1
    Sub Main()
        Dim srLector As StreamReader = New StreamReader("c:\prueba.txt")
        ' leer línea a línea mediante un bucle

        Dim Linea As String
        Dim ContadorLin As Integer = 1
        Linea = srLector.ReadLine()
        While Not (Linea Is Nothing)

            Console.WriteLine("Línea: {0} - Contenido: {1}", ContadorLin, Linea)
            ContadorLin += 1
            Linea = srLector.ReadLine()
        End While
        Console.ReadLine()
    End Sub
End Module
```