

Trabajando con Ficheros

Índice

- 1 - Rutas absolutas y relativas
- 2 - Leer un fichero
 - 2.1 - Tipo StreamReader
 - 2.2 - Tipo File
- 3 - Escribir un fichero
 - 3.1 - Tipo StreamWriter
 - 3.2 - Tipo File

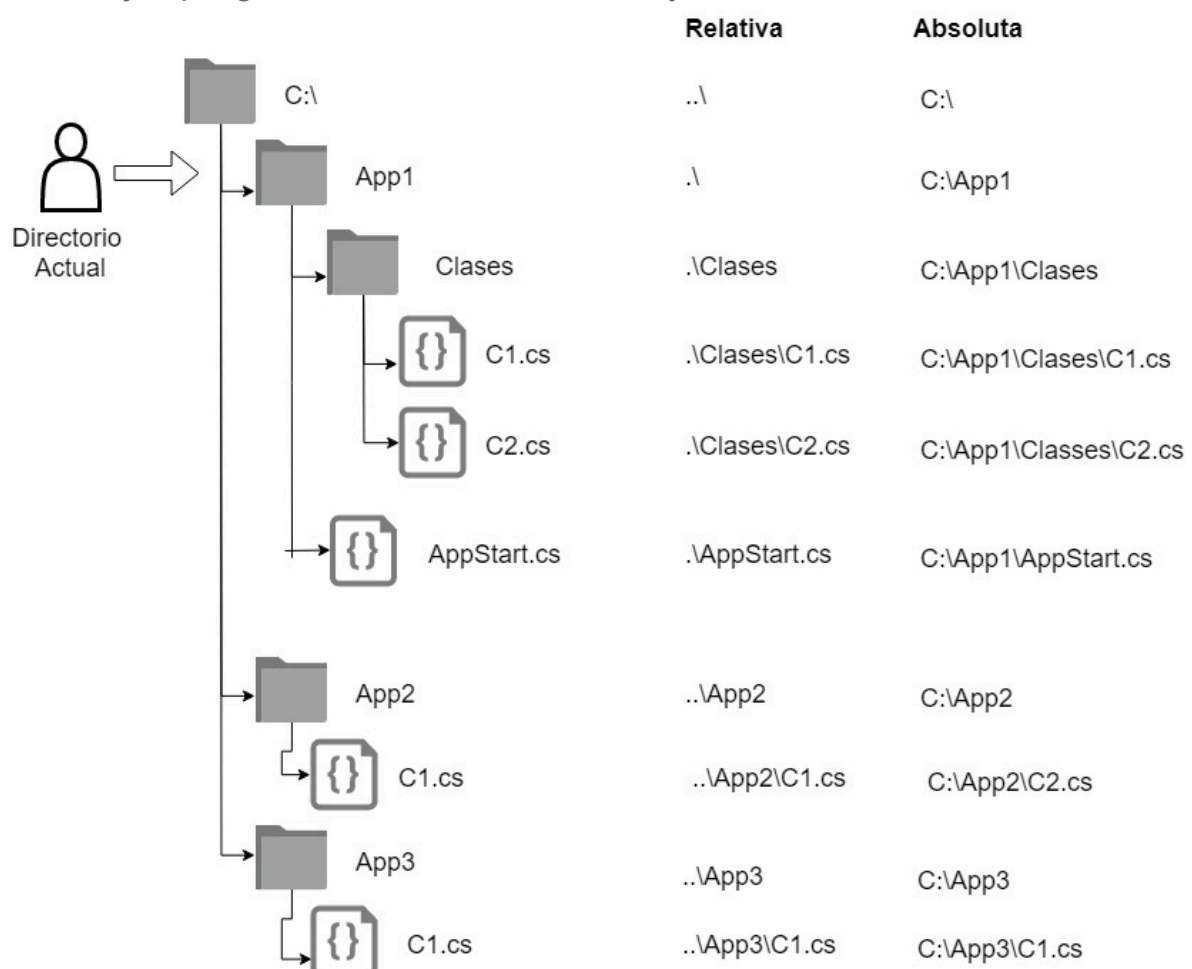
1 - Rutas absolutas y relativas

Lo primero que veremos serán las rutas absolutas y relativas, ya que estas son las que utilizaremos para acceder a los propios ficheros.

La diferencia entre ambas es muy simple:

- En las rutas relativas accedemos en función de la carpeta o directorio en el que nos encontramos.
- Mientras que en las rutas absolutas accedemos basándonos en el directorio raíz del ordenador. Comúnmente `C:\` en Windows o `/` en Linux.

Esto sería un ejemplo gráfico de las rutas relativas y absolutas:



Como vemos a la izquierda nos ubicamos inicialmente en el directorio `C:\App1` por lo que para subir un submenu indicamos `..\` mientras que en las rutas absolutas da igual donde nos coloquemos, ya que siempre inicializamos indicando `C:\`

En el mundo laboral debido a que las aplicaciones pueden estar ubicadas en distintos servidores se utilizan rutas absolutas para ubicar los ficheros.

2 - Leer un fichero

2.1 - Tipo StreamReader

Para leer un fichero ya existente, utilizamos el tipo `StreamReader()` y este a su vez lo pondremos dentro de una sentencia `using`.

Por qué lo instanciamos dentro de la sentencia `using` ?

Porque cuando el bloque de código que crea el `using` se cierra, automáticamente destruye de la memoria todas las variables inicializadas dentro de él. Esto obviamente no es problema para una aplicación muy pequeña, pero cuando tenemos una app muy grande, o que requiere leer varios GB de información, puede llegar a ser un problema muy grande, y hacer fallar nuestro sistema.

`StreamReader` nos trae por defecto un método llamado `.ReadLine()` el cual lee cada línea del documento, y podremos trabajar con él.

```
//Accedemos al fichero
using (StreamReader reader = new StreamReader("c:\ejemplo\fichero.txt"))
{
    string text; //Creamos la variable que contendrá el texto
    while ((text = reader.ReadLine()) != null) //Leemos línea por línea
    {
        Console.WriteLine(text); //Mostramos la información
    }
}
```

2.2 - Tipo File

Otra opción es la clase `File`, la cual nos leerá el fichero de un solo volcado, como se observa en todos mis posts, me preocupa en cierta medida la memoria, así que si el fichero es muy grande quizá no sea la mejor idea.

```
string filereader = File.ReadAllText("c:\ejemplo\fichero.txt");
```

3 - Escribir un fichero

3.1 - Tipo StreamWriter

Para escribir el proceso también es muy sencillo, en este caso utilizaremos el tipo `StreamWriter` y además de ello, deberemos indicar el texto o la lista de texto a escribir.

```
string[] paises = new string[] { "USA", "Inglaterra", "Alemaia" };
using (StreamWriter writer = new StreamWriter("c:\\ejemplo\\fichero.txt"))
{
    foreach (string item in countries)
    {
        writer.WriteLine(item);
    }
}
```

Este método tiene una pequeña pega, y es que, para añadir más adelante una línea de texto, lo que hace por detrás el código es reescribir el fichero entero. Además, si este fichero no existe, nos lo creará.

3.2 - Tipo File

Como en el caso anterior, podemos utilizar el tipo `File` para escribir, en este caso tenemos varias opciones

```
string[] paises = new string[] { "USA", "England", "Germany" };

File.WriteAllText("c:\\ejemplo\\fichero.txt", "China Japon Korea");
File.WriteAllLines("c:\\ejemplo\\fichero.txt", paises);
```

- `File.WriteAllText()` escribirá todo el texto de un solo volcado.
- `File.WriteAllLines()` escribirá línea por línea cada uno de los elementos que mandamos en el `array`.