

Las clases Directory y DirectoryInfo

Cuando necesitemos acceder a un directorio, por ejemplo, para saber qué subdirectorios y ficheros contiene o para obtener otro tipo de información, incluso para saber si existe o no, podemos utilizar las clases *Directory* o *DirectoryInfo*.

La primera de ellas: *Directory*, proporciona **métodos estáticos**, es decir, los métodos que contiene siempre estarán **disponibles sin necesidad de crear una nueva instancia de la clase**. Por tanto, podremos usar estos métodos simplemente usando la propia clase.

La segunda: *DirectoryInfo*, proporciona los mismos métodos que *Directory*, pero en lugar de ser métodos estáticos, son **métodos de instancia**, es decir, estos métodos solamente se podrán usar después de haber **creado una instancia (u objeto en memoria) de esta clase**. Cada una de las instancias de *DirectoryInfo*, hará referencia a un directorio en particular, ese directorio se indicará al crear el objeto, aunque también puede ser que esté referenciado al obtenerlo mediante otros métodos que devuelven este tipo de objetos.

Nota.- El directorio asignado al crear la instancia de *DirectoryInfo* no tiene porqué existir. De esa forma podemos comprobar si existe, y en caso de que no exista, crearlo, etc.

Por ejemplo, si queremos saber si un directorio existe, utilizaremos el método *Exists*:

- En el caso de la clase *Directory*, como parámetro le indicaremos el directorio del que queremos comprobar su existencia, sin embargo,
- si usamos un objeto del tipo *DirectoryInfo*, no necesitamos indicar ningún parámetro, ya que el directorio que está asociado con esa clase lo habremos indicado al **crear el objeto**.

En el siguiente código vemos cómo usar estas dos clases:

```
Const dir1 As String = "E:\Pruebas"
Const dir2 As String = "E:\Pruebas2"

Sub Main()
    claseDirectory()
    Console.WriteLine()
    claseDirectoryInfo()
    Console.WriteLine()
    Console.ReadLine()
End Sub

Sub claseDirectory()
    ' Los métodos de la clase Directory son estáticos,
    ' por tanto no necesitamos crear una instancia de esta clase
    '
    ' Comprobar si existe un directorio:
    If System.IO.Directory.Exists(dir1) Then
        Console.WriteLine("El directorio '{0}' SI existe", dir1)
    Else
        Console.WriteLine("El directorio '{0}' NO existe", dir1)
    End If
End Sub
```

```
Sub claseDirectoryInfo()
    ' Los métodos de la clase DirectoryInfo son de instancia,
    ' por tanto tenemos que crear una instancia de la clase
    ' para poder usarla
    '
    Dim di As New System.IO.DirectoryInfo(dir2)
    '
    ' Comprobar si existe un directorio:
    If di.Exists Then
        Console.WriteLine("El directorio '{0}' SI existe", dir2)
    Else
        Console.WriteLine("El directorio '{0}' NO existe", dir2)
    End If
End Sub
```

El resto de métodos de estas dos clases funcionan de forma similar, al menos en el sentido de que si usamos la clase *Directory*, siempre habrá un parámetro que haga referencia al directorio que queremos usar, mientras que con *DirectoryInfo* siempre estará haciendo referencia al directorio usado al crear la instancia.

Nota.- Interesa que quede claro cómo funcionan este tipo de clases, que por otro lado, será similar al de las clases *File* y *FileInfo*, en el sentido de que **la clase con los métodos estáticos siempre necesitará saber a qué elemento del sistema de archivos estamos refiriéndonos, mientras que las clases que debemos instanciar, siempre sabrán con qué elemento trabajarán.**

Como hemos comentado, la clase *DirectoryInfo* siempre hará referencia al directorio utilizado para instanciarla, por tanto algunos de los métodos de la clase *Directory* se convierten en propiedades de esta clase y otros cambian de nombre para adecuarlos mejor a la acción a realizar con ellos. Por ejemplo, en la clase *Directory* tenemos el método *Exists*, que en *DirectoryInfo* es una propiedad. De igual forma, el método *Move* de la clase *Directory* es el método *MoveTo* de *DirectoryInfo*.

En cuanto a los métodos de *Directory* para obtener o asignar la fecha de creación, acceso, etc., en *DirectoryInfo* son propiedades de lectura/escritura. Dicho esto, no vamos a nombrar todos estos miembros de *DirectoryInfo*, ya que en la documentación están bien detallados y no nos resultará difícil saber cómo usarlos. Lo que sí queremos resaltar es que algunos de los métodos de la clase *DirectoryInfo*, concretamente *GetDirectories* y *GetFiles*, no devuelven un array de tipo *String*, sino un array de objetos *DirectoryInfo* o *FileInfo* respectivamente.

También debemos indicar que la clase *DirectoryInfo* tiene ciertas propiedades, como *FullName* o *Name*, que nos dan información del nombre de la ruta completa o del nombre del directorio.