

## 9.4. Lectura de directorios

Si queremos analizar el contenido de un directorio, podemos emplear las clases `Directory` y `DirectoryInfo`.

La clase `Directory` contiene métodos para crear un directorio (`CreateDirectory`), borrarlo (`Delete`), moverlo (`Move`), comprobar si existe (`Exists`), etc. Por ejemplo, podríamos crear un directorio con:

```
// Ejemplo_09_04a.cs
// Crear un directorio
// Introducción a C#, por Nacho Cabanes

using System.IO;

class Ejemplo_09_04a
{
    static void Main()
    {
        string miDirectorio = @"d:\datos";
        if (!Directory.Exists(miDirectorio))
            Directory.CreateDirectory(miDirectorio);
    }
}
```

(la clase `Directory` está declarada en el espacio de nombres `System.IO`, por lo que deberemos añadirlo entre los "using" de nuestro programa).

También tenemos un método "GetFiles" que nos permite obtener la lista de ficheros que contiene un directorio. Así, podríamos listar todo el contenido de una carpeta con:

```
// Ejemplo_09_04b.cs
// Lista de ficheros en un directorio
// Introducción a C#, por Nacho Cabanes

using System;
using System.IO;

class Ejemplo_09_04b
{
    static void Main()
    {
        string miDirectorio = @"c:\\";
        string[] listaFicheros;

        listaFicheros = Directory.GetFiles(miDirectorio);
        foreach (string fichero in listaFicheros)
            Console.WriteLine(fichero);
    }
}
```

```
}
```

Se puede añadir un segundo parámetro a "GetFiles", que sería el patrón que deben seguir los nombres de los ficheros que buscamos, como "\*.txt". Por ejemplo, podríamos saber todos los fuentes de C# (\*.cs) de la carpeta actual (".") con:

```
// Ejemplo_09_04c.cs
// Lista de ficheros en un directorio
// Introducción a C#, por Nacho Cabanes

using System;
using System.IO;

class Ejemplo_09_04c
{
    static void Main()
    {
        string[] listaFicheros = Directory.GetFiles(".", "*.cs");
        foreach(string fichero in listaFicheros)
            Console.WriteLine(fichero);
    }
}
```

Si necesitamos más detalles, la clase DirectoryInfo permite obtener información sobre fechas de creación, modificación y acceso, y, de forma análoga, FileInfo nos permite conseguir información similar sobre un fichero. Podríamos usar estas dos clases para ampliar el ejemplo anterior, y que no sólo muestre el nombre de cada fichero, sino otros detalles adicionales como el tamaño y la fecha de creación:

```
// Ejemplo_09_04d.cs
// Lista detallada de ficheros en un directorio
// Introducción a C#, por Nacho Cabanes

using System;
using System.IO;

class Ejemplo_09_04d
{
    static void Main()
    {
        string miDirectorio = @"c:\";
        DirectoryInfo dir = new DirectoryInfo(miDirectorio);
        FileInfo[] infoFicheros = dir.GetFiles();
        foreach (FileInfo infoUnFich in infoFicheros)
        {
            Console.WriteLine("{0}, de tamaño {1}, creado {2}",
                infoUnFich.Name,
                infoUnFich.Length,
                infoUnFich.CreationTime);
        }
    }
}
```

```
}
}
```

que escribiría cosas como

```
hiberfil.sys, de tamaño 6775930880, creado 15/07/2013 17:48:07
```

También se puede comprobar los valores de los **"atributos"** de un fichero, que permiten saber detalles como si está oculto, si es de sólo lectura, si está pendiente de copia de seguridad... Como es algo más complejo y menos frecuente, vamos a ver sólo un ejemplo de su forma de uso: se debe comparar el valor de la propiedad "Attributes" con los posibles valores de la enumeración "FileAttributes", usando una multiplicación bit a bit (un producto lógico, con el operador "&"):

```
bool oculto =
    (infoUnFich.Attributes & FileAttributes.Hidden) == FileAttributes.Hidden;
```

### Ejercicios propuestos:

**(9.4.1)** Crea un programa que muestre en pantalla el contenido de un fichero de texto, cuyo nombre escoja el usuario. Si el usuario no sabe el nombre, podrá pulsar "Intro" y se le mostrará la lista de ficheros existentes en el directorio actual, para luego volver a preguntarle el nombre del fichero.

**(9.4.2)** Crea un programa que cree un fichero de texto a partir del contenido de todos los ficheros de texto existentes en la carpeta actual.

**(9.4.3)** Crea un programa que permita "pasear" por la carpeta actual, al estilo del antiguo "Comandante Norton": mostrará la lista de ficheros y subdirectorios de la carpeta actual, y permitirá al usuario moverse hacia arriba o abajo dentro de la lista usando las flechas del cursor. El elemento seleccionado se mostrará en color distinto del resto.

**(9.4.4)** Mejora el ejercicio 9.4.3 para que muestre directorios (en primer lugar) y ficheros (a continuación), y permita entrar a un directorio si se pulsa Intro sobre él (en ese momento, se actualizará la lista de ficheros y directorios, para mostrar el contenido del directorio al que se ha accedido).

**(9.4.5)** Mejora el ejercicio 9.4.4 para que contenga dos paneles, uno al lado del otro, cada uno de los cuales podrá estar mostrando el contenido de un directorio distinto. Si se pulsa el "tabulador", cambiará el panel activo.

**(9.4.6)** Mejora el ejercicio 9.4.5, para que se pueda "seleccionar un fichero" pulsando "Espacio" o "Insert". Los ficheros seleccionados se mostrarán en un color distinto. Se podrán deseleccionar volviendo a pulsar "Espacio" o "Insert". Si se

pulsa F5, los ficheros seleccionados en la carpeta actual del panel actual se copiarán a la carpeta del otro panel.

**(9.4.7)** Mejora el ejercicio 9.4.6, para que mientras se están copiando ficheros, el programa muestre una "barra de progreso" de color amarillo, que indicará el porcentaje de ficheros que ya se han copiado.