

Tratamiento de Ficheros

- 1. Introducción.....2
- 2. Abrir o visualizar un archivo de texto2
 - 2.1. Cómo abrir un archivo de texto para entrada.....2
- 3. Guardar un archivo de texto.....3

1. Introducción

Administrar documentos electrónicos es una función importante en cualquier empresa. El tipo de documento más básico es el archivo de texto, que está formado por palabras y párrafos no formateados, letras, números y diversos caracteres y símbolos de propósito especial.

Los archivos de texto son distintos de los documentos, que contienen códigos de formato. La mayoría de ellos tendrán extensiones `.txt`, `.ini`, `.log`, `.dat...` etc.

2. Abrir o visualizar un archivo de texto


En las versiones previas a .Net, estas eran las instrucciones a ejecutar:

Función	Descripción
FileOpen	Abre un archivo de texto para entrada o salida
LineInput	Lee una línea de entrada desde el archivo de texto
EOF	Comprueba el final del archivo de texto
FileClose	Cierra el archivo de texto

Todas estas funciones han sido sustituidas por la nueva clase **StreamReader**, que nos permitirá abrir y mostrar el contenido del fichero de texto.

Para poder utilizar esta clase es preciso que la primerísima instrucción de la clase formulario sea: `using System.IO;`



2.1. Cómo abrir un archivo de texto para entrada

OpenFileDialog  Para permitir que el usuario pueda abrir el fichero que desea, utilizaremos este control.

Mediante el cual muestra un explorador de Windows para buscar el fichero deseado, y poder seleccionarlo. Modificaremos sus propiedades:

- **Filter:** Controla los tipos de archivos que se van a mostrar.
Ej. --- Filter: "Documentos de texto (*.txt) | *.txt"
- **ShowDialog:** Muestra el cuadro de diálogo. Es importante que controlemos que el usuario elige algún fichero, y no sale de la ventana cancelando.
`if (this.OpenFile.ShowDialog() == DialogResult.OK)`
- **FileName:** Devuelve la ruta de acceso del fichero escogido por el usuario

Una vez que el usuario pinche a OK abrimos el fichero y lo mostramos en el cuadro de texto


 CIFP VIRGEN DE GRACIA	TEMA 2 Tratamiento de Ficheros	 Página 3 de 4
------------------------------------------------------------------------------------------------------------	-------------------------------------------------	------------------------------------------------------------------------------------------------------

```
private void Abrir()
{
    StreamReader texto;
    //'Si se escogió un fichero
    if (ofdAbrir.ShowDialog() == DialogResult.OK)
    {
        //'Carga del contenido del fichero
        texto = new StreamReader(ofdAbrir.FileName);
        txtTexto.Text = texto.ReadToEnd();

        texto.Close();    // todo lo que se abre se cierra!!!

        //'Activación de los componentes
    }
}
```

3. Guardar un archivo de texto

SaveFileDialog  Para guardar un archivo de texto emplearemos este control, y el usuario elijará el nombre y la carpeta destino.

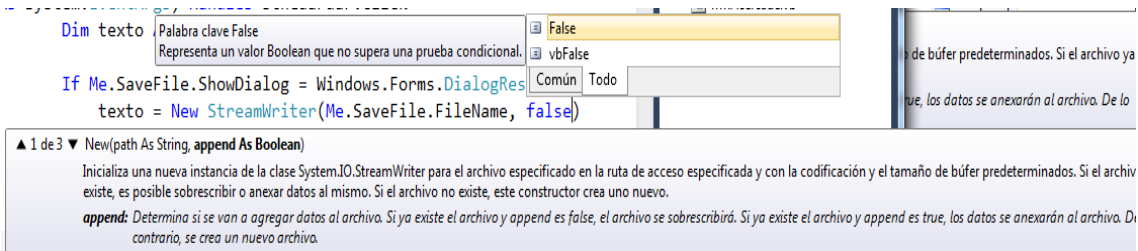
De forma similar mostraremos esta caja y comprobaremos que el usuario presiona Aceptar y no Cancelar... y que elige un nombre para el usuario. También tiene las propiedades **Filter** y **FileName**

En este caso, utilizaremos la clase **StreamWriter** (cuando queramos añadir) o **SaveFile**

```
private void mnGuardar_Click(object sender, EventArgs e)
{
    if (this.sfdGuardar.FileName != "")
    {
        //this.txtTexto.SaveFile(this.sfdGuardar.FileName, RichTextBoxStreamType.PlainText);

        StreamWriter texto = new StreamWriter(this.sfdGuardar.FileName, false); // append = false
        texto.WriteLine(this.txtTexto.Text);
        texto.Close();
    }
    else
        mnGuardarComo_Click(sender, e);
}
```

En algunos casos nos puede interesar que cuando guardemos, no sobrescribamos la información que ya tiene el archivo, sino que lo que queremos es añadirla, en ese caso tendremos que añadir un segundo parámetro: False



Es decir si **Append** es True → El archivo se anexarán, si **Append** es False → Entonces si el archivo existe se sobrescribirá, y si no existe, en cualquiera de los casos, se creará el archivo.

