

ARCHIVOS

Un Archivo es un conjunto de bytes almacenados (de forma permanente) en un dispositivo periférico, tal como un disco duro, una memoria flash, un CD-ROM.

Diferencia entre trabajar con datos en	
Memoria	Archivos en el Disco duro
Al apagar el equipo los datos en la memoria se borran	Al apagar los datos persisten, podrán ser leídos y modificados la próxima vez
El acceso a los datos es muy rápido (nseg)	El acceso a los datos es lento (mseg). Para ahorrar tiempo se usa un 'buffer'

Justificación del uso de un 'buffer' para escribir en un dispositivo.

Es más rápido escribir 1 vez 500 bytes
que escribir 500 veces 1 byte en un disco duro.

Por eso, cuando se escribe en un periférico no se escribe directamente, sino en un 'buffer', que es un trozo de memoria que almacena temporalmente los datos con destino a un dispositivo.

Así, cuando escribimos un dato, el S.O. lo hace en el 'buffer' del dispositivo correspondiente, a la espera de que lleguen más datos con destino a ese dispositivo. Cuando se llena el 'buffer' (o al terminar de usar ese dispositivo) se envían todos los datos del 'buffer' de golpe.

Tipos de ficheros según su contenido. Se suele distinguir entre:

- Ficheros de texto: La información se corresponde a caracteres (en algún tipo de código: ASCII, ASCII extendido, ISO 8859-1 (ISO-Latin-1), Unicode, UTF-8 ...)
- Ficheros binarios: Contiene información de cualquier tipo almacenada en binario. No se puede saber en principio cuantos bytes o bits es un dato, o si todos los datos son iguales.

Tipos de ficheros según su acceso:

- Ficheros de acceso secuencial: Para acceder a un dato, hay que hacerlo en un orden preestablecido. Es decir, para leer el dato en la posición 8 he de empezar en la posición 1 y moverme hasta la 8 (como en una cinta de cassette. Observa cómo se tardará muy poco en llegar al dato 8 pero mucho en llegar al dato 77678)
- Ficheros de acceso aleatorio: Se puede acceder a cualquier dato en cualquier orden y en prácticamente el mismo tiempo esté donde esté

Trabajar con ficheros en VB.NET

1.- Crear un objeto de la clase `FileStream` y asociarlo con un archivo externo.

`Dim fichero1 As New FileStream(Nombre_fichero, Modo, Acceso)`

- **Nombre_fichero** es un string con el nombre del fichero, que puede incluir el path completo. Si no, se presupone que está en el directorio que esté en uso en ese momento.

- **Modo:**
- | | |
|---------------------|---|
| CreateNew | Crear un nuevo fichero. Si el fichero existe, se lanzará una excepción del tipo IOException . |
| Create | Crear un nuevo fichero. Si el fichero existe, será sobrescrito. |
| Open | Abrir un fichero existente. Si el fichero no existe, se lanzará una excepción del tipo FileNotFoundException . |
| OpenOrCreate | Abrir un fichero si existe; si no, se crea un nuevo fichero. |
| Truncate | Abrir un fichero existente. Una vez abierto, el fichero será truncado a cero bytes de longitud. |
| Append | Abrir un fichero para añadir datos al final del mismo si existe, o crear un nuevo fichero si no existe. |
- Acceso:
- | | |
|------------------|--|
| Read | Permite acceder al fichero para realizar operaciones de lectura. |
| ReadWrite | Permite acceder al fichero para realizar operaciones de lectura y escritura. |
| Write | Permite acceder al fichero para realizar operaciones de escritura. |

2.- Crear un objeto tipo `BinaryReader` o `BinaryWriter` para facilitar las operaciones de lectura y/o escritura.

`Dim lector_binario As New BinaryReader(fichero1)`

3.- Leer (o Escribir) un dato del fichero utilizando el método del BinaryReader (o BinaryWriter) que se corresponda con el tipo de dato a leer (o escribir)

```
Dim n as Double
n = lector_binario.ReadDouble()
```

4.- Al terminar cerrar el acceso.

```
lector_binario.Close()
fichero1.Close()
```



Programa en el que se almacena el número más grande introducido hasta el momento en un TextBox.

Imports System.IO ' Hacemos esto para poder escribir FileStream en vez de System.IO.FileStream todo el rato
Public Class Form1

```
-----
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
    ' Antes de cargar el Formulario leemos el fichero.
    Dim nombre_completo As String =
        System.Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) _
        + "\" + "record.dat"
    If (File.Exists(nombre_completo)) Then ' Si el fichero existe leer el número
        Dim fichero1 As FileStream = Nothing ' Solo declararlo, lo construiremos en un bloque TRY por si da problemas
        Dim lector_binario As BinaryReader = Nothing
        Try
            fichero1 = New FileStream(nombre_completo, FileMode.Open, FileAccess.Read)
            lector_binario = New BinaryReader(fichero1)
            Labell.Tag = lector_binario.ReadDouble()
            Labell.Text = "El record hasta el momento es " & Labell.Tag
        Catch ex As Exception
            MessageBox.Show("Se produjo un error al leer: " & ex.Message)
        Finally
            If lector_binario IsNot Nothing Then lector_binario.Close()
            If fichero1 IsNot Nothing Then fichero1.Close() ' Si el fichero se abrió,
            ' aunque luego haya una excepción se cerrará
        End Try
    Else ' Si el fichero no existe asignar cero
        MessageBox.Show("Fichero NO encontrado. Se asigna 0")
        Labell.Tag = 0
        Labell.Text = "El record hasta el momento es " & Labell.Tag
    End If
End Sub
-----
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles Button1.Click
    ' Boton PROBAR NUMERO
    If Convert.ToDouble(TextBox1.Text) > Convert.ToDouble(Labell.Tag) Then
        Labell.Tag = TextBox1.Text
        Labell.Text = "El record hasta el momento es " & Labell.Tag
    End If
End Sub
-----
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    ' Boton Salir, tambien actua cuando se cierra en la X
    Me.Close()
End Sub
-----
Private Sub saliendo(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Me.FormClosing
    Dim nombre_completo As String =
        System.Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)
        + "\" + "record.dat"
    Dim fichero1 as FileStream = Nothing
    Dim escritor_binario as BinaryWriter = Nothing
    Try
        fichero1 = New FileStream(nombre_completo, FileMode.Create, FileAccess.Write)
        escritor_binario = New BinaryWriter(fichero1)
        escritor_binario.Write(Convert.ToDouble(Labell.Tag))
    Catch ex As Exception
        MessageBox.Show("Se produjo un error al guardar: " & ex.Message)
    Finally
        If escritor_binario IsNot Nothing Then escritor_binario.Close()
        If fichero1 IsNot Nothing Then fichero1.Close()
    End Try
End Sub
End Class
```