SIST. Y APLICACIONES INFORMÁTICAS
CURSO: 2009-2010
SEMANA: 18
PROFESOR: Javier Sánchez
TEMAS:Visual Basic .NET

Ejemplo: crearemos un archivo donde introduciremos una serie de datos personales, en concreto nombre, edad y peso, cuando hayamos terminado de introducir estos datos leeremos todos los datos del fichero. Crearemos una clase llamada DatosPersonales donde iremos metiendo todos estos datos solicitados. Otra clase llamada TablaBaseDatos que manejará lasescritura y lectura de datos en un fichero

```
Public Class DatosPersonales
    Private nombre As String
    Private edad As Short
    Private Peso As Single

    Public Sub New(ByVal _nombre As String, ByVal _edad As Int16, ByVal _peso As Single)
        nombre = _nombre
        edad = _edad
        Peso = _peso
    End Sub

    Public Sub Imprimir()
        Console.WriteLine("Nombre: {0} edad :{1}, peso : {2}", nombre, edad, Peso)
    End Sub

    Public Function dameNombre()
        Return nombre
    End Function

End Class


Imports System.IO

Public Class tablabasedatos

    Private mitabla As FileStream
    Private br As BinaryReader
    Private bw As BinaryWriter
    Private numeroRegistros As Integer

    Public Sub leeregistro(ByVal nombretabla As String)

        Dim nombre As String = ""
        Dim edad As Integer
        Dim peso As Single
        Dim Persona As DatosPersonales
        Dim i As Integer

        Try
            br = New BinaryReader(New FileStream(nombretabla, FileMode.Open,
FileAccess.Read))
```

```
            While True
                nombre = br.ReadString
                edad = br.ReadInt16
                peso = br.ReadSingle
                Persona = New DatosPersonales(nombre, edad, peso)
                Persona.Imprimir()
            End While

        Catch e As EndOfStreamException
            Console.WriteLine("Fin de fihero")
        Finally
            If Not br Is Nothing Then br.Close()
        End Try
    End Sub


    Public Sub escriberegistro(ByRef bwaux As BinaryWriter)

        Dim _nombre As String
        Dim _edad As Int16
        Dim _Peso As Single
        Dim resp As Char
        Try
            Console.WriteLine("Introduzca los siguientes datos personales")

            Do
                Console.Write("nombre:     ") : _nombre = Console.ReadLine
                Console.Write("edad  :     ") : _edad = Console.ReadLine
                Console.Write("peso:       ") : _Peso = Console.ReadLine
                bwaux.Write(_nombre)
                bwaux.Write(_edad)
                bwaux.Write(_Peso)

                Console.Write("Desea escribir otro registro?")
                resp = Convert.ToChar(Console.Read)
                Console.ReadLine()

            Loop While (resp = "s"c)
        Finally
            If Not bwaux Is Nothing Then bwaux.Close()
        End Try

    End Sub

End Class

Imports System.IO

Module Module1

    Sub Main()
        Dim resp As Char

        Dim nombrearchivo As String
        Dim tabla As tablabasedatos = New tablabasedatos()
        Dim persona As DatosPersonales
        Dim bw As BinaryWriter
        Dim cont As Integer = 0
```

```
        Console.WriteLine("Introduzca el nombre del archivo")
        nombrearchivo = Console.ReadLine

        bw = New BinaryWriter(New FileStream(nombrearchivo, FileMode.Create,
FileAccess.Write))
        tabla.escriberegistro(bw)
        tabla.leeregistro(nombrearchivo)
    End Sub

End Module
```

Acceso aleatorio

Leer o escribir a partir de una determinada posición dentro del fichero es importante cuando necesitamos modificar alguno de los valores contenidos en el fichero o cuando necesitamos extraer una parte concreta del mismo, sin tener que reescribir todo el fichero.

La clase Stream nos proporcionan métodos que permiten el acceso directo. En un fichero de acceso directo, tenemos métodos que nos permitirán situar el puntero de lectura o escritura del archivo en una posición concreta. Este puntero cuando se abre el fichero tanto para lectura como para escritura siempre se sitúa al comienzo del fichero, excepto cuando lo abrimos para añadir, que se coloca al final. Se nos plantean varios problemas, uno es como diferenciar el final de un registro, sin tener que leer secuencialmente todos los campos de la estructura de datos almacenada y otro problema es a la hora de modificar alguno de los valores contenidos en uno de los registros, tenemos que evitar que el nuevo dato sobrescriba otros campos de la estructura que almacena el fichero. Esto se consigue imponiendo que todos los registros que introducimos en el fichero tengan la misma longitud. En el siguiente ejemplo vamos a seguir utilizando la clase de datos anterior

```
Public Class DatosPersonales
Private nombre As String
Private edad As Short
Private Peso As Single
```

Limitaremos el tamaño del nombre a 20 caracteres y por tanto el tamaño de este registro será de 50 bytes contando que el campo nombre ocupa 20 caracteres Unicode es decir 40 bytes, la edad 2 bytes al ser short y el peso 4 bytes al ser single, que nos dan un total de 46 pero deberemos de dejar algo más de espacio, ya que el método write al escribir un String introduce un campo oculto al comienzo que indica cual es la longitud, esto permitirá al método ReadString saber cuantos bytes tiene que leer. Al introducir un nuevo registro en el archivo deberemos comprobar que no supere este límite. Después deberemos situarnos en la posición adecuada para escribir este registro, si estamos añadiendo registro por el final, deberemos contar cuantos registros tiene el archivo al abrirlo

```
numeroRegistros = Math.Ceiling(fs.Length / longitudregistro)
```

Como es casi seguro que el último registro no ocupe el tamaño fijado, utilizamos Ceiling para redondear por encima.

Para movernos por el archivo utilizaremos el método seek, pero como el flujo binaryReader o BinaryWriter no lo soportan tendremos que recurrir a la propiedad Seek de BaseStream que es flujo que subyace bajo estas dos clases.

```
bw.BaseStream.Seek(numreg * longitudregistro, SeekOrigin.Begin)
```

Y la función nos quedaría de la siguiente forma:

```
Public Sub escriberegistro(ByVal pers As Personas, ByVal numreg As Integer)

    If pers.tamaño +4 > longitudregistro Then
        Console.WriteLine("La longitud del registro es excesiva")
    Else
        bw.BaseStream.Seek(numreg * longitudregistro, SeekOrigin.Begin)
        bw.Write(pers.damenombre)
        bw.Write(pers.dameedad)
        bw.Write(pers.damepeso)
    End If
End Sub
```

Para leer un registro utilizamos el siguiente procedimiento que tiene un parámetro para identificar el número del registro que se desea leer.

```
Public Sub leeregistro(ByVal numreg As Integer)

    Dim nombre As String = ""
    Dim edad As Integer
    Dim peso As Single
    Dim Persona As Personas
    Dim i As Integer

    Try
        'Situas el puntero al comienzo del registro correspondiente
        br.BaseStream.Seek(numreg * longitudregistro, SeekOrigin.Begin)
        nombre = br.ReadString
        edad = br.ReadInt16
        peso = br.ReadSingle
        Persona = New Personas(nombre, edad, peso)
        Persona.Imprimir()

    Catch e As EndOfStreamException
        Console.WriteLine("Fin de fihero")
    End Try
End Sub
```

Para buscar un registro, vamos a buscarlo por su nombre y utilizamos la siguiente función que tiene dos parámetros como son el nombre a buscar y el numero de registros que tiene nuestro archivo. Nos devolverá un entero que será el numero del registro o -1 si no se encuentra el registro en el archivo.

```
Public Function buscar(ByVal nombreaux As String, ByVal fin As Integer) As Integer
    'recorremos todos los los registros comparando el nombre
    Dim cont, i As Integer
    Dim nombre As String
    Dim encontrado As Boolean = False

    For i = 0 To fin
        br.BaseStream.Seek(i * longitudregistro, SeekOrigin.Begin)
        nombre = br.ReadString
        If nombre = nombreaux Then
            encontrado = True
            Exit For
        End If
    Next
    If encontrado Then
        Return i
    Else : Return -1
    End If

End Function
```

El ejemplo completo para manejar un fichero con acceso directo seria:

```
Imports System.IO

Public Class Tabla

    Private fs As FileStream
```

```
Private br As BinaryReader
Private bw As BinaryWriter

Private numeroRegistros As Integer
Private longitudregistro = 50
Private desplazamiento = 4


Public Sub New(ByVal nombrefichero As String)

    fs = New FileStream(nombrefichero, FileMode.OpenOrCreate, FileAccess.ReadWrite)
    bw = New BinaryWriter(fs)
    br = New BinaryReader(fs)
    ' Como es casi seguro que el último registro no ocupe el
    ' tamaño fijado, utilizamos Ceiling para redondear por encima.
    numeroRegistros = Math.Ceiling(fs.Length / longitudregistro)
End Sub

Public Function dame_numero_registros() As Integer
    Return numeroRegistros
End Function

Public Sub cerrartabla()

    If Not br Is Nothing Then br.Close()
    If Not bw Is Nothing Then bw.Close()
    If Not fs Is Nothing Then fs.Close()

End Sub

Public Sub leeregistro(ByVal numreg As Integer)

    Dim nombre As String = ""
    Dim edad As Integer
    Dim peso As Single
    Dim Persona As Personas
    Dim i As Integer

    Try
        'Situas al comienzo del registro correspondiente
        br.BaseStream.Seek(numreg * longitudregistro, SeekOrigin.Begin)
        nombre = br.ReadString
        edad = br.ReadInt16
        peso = br.ReadSingle
        Persona = New Personas(nombre, edad, peso)
        Persona.Imprimir()

    Catch e As EndOfStreamException
        Console.WriteLine("Fin de fihero")
    End Try
End Sub

Public Sub leetodosregistros()
    Dim i As Integer

    For i = 0 To numeroRegistros
        leeregistro(i)
    Next
```

```
    End Sub

    Public Function buscar(ByVal nombreaux As String, ByVal fin As Integer) As Integer
        'recorremos todos los los registros comparando el nombre
        Dim cont, i As Integer
        Dim nombre As String
        Dim encontrado As Boolean = False

        For i = 0 To fin
            br.BaseStream.Seek(i * longitudregistro, SeekOrigin.Begin)
            nombre = br.ReadString
            If nombre = nombreaux Then
                encontrado = True
                Exit For

            End If
        Next
        If encontrado Then
            Return i
        Else : Return -1
        End If
    End Function

    Public Sub escriberegistro(ByVal pers As Personas, ByVal numreg As Integer)

        If pers.tamaño > longitudregistro Then
            Console.WriteLine("La longitud del registro es excesiva")
        Else
            bw.BaseStream.Seek(numreg * longitudregistro, SeekOrigin.Begin)
            bw.Write(pers.damenombre)
            bw.Write(pers.dameedad)
            bw.Write(pers.damepeso)
        End If

    End Sub
End Class

Imports System.IO

Module Module1

    Public Function datospersonales() As Personas
        Dim _nombre As String
        Dim _edad As Int16
        Dim _Peso As Single

        Try
            Console.WriteLine("Introduzca los siguientes datos personales")
            Console.Write("nombre:     ") : _nombre = Console.ReadLine
            Console.Write("edad  :     ") : _edad = Console.ReadLine
            Console.Write("peso:       ") : _Peso = Console.ReadLine
            Return New Personas(_nombre, _edad, _Peso)

        Catch e As Exception
            Console.WriteLine("Datos fuera de rango", e.Message)
        End Try
    End Function
```

```
    Sub Main()

        Dim resp As Char
        Dim nombrearchivo As String
        Dim mitabla As Tabla
        Dim cont, pos, opcion As Integer
        Dim personabuscada As String

        Try
            Console.WriteLine("Introduzca el nombre del archivo")
            nombrearchivo = Console.ReadLine
            mitabla = New Tabla(nombrearchivo)
            cont = mitabla.dame_numero_registros

            Do
                Console.WriteLine()
                Console.WriteLine("1.- Introducir datos personales")
                Console.WriteLine("2.- Listar El fichero")
                Console.WriteLine("3.- Buscar un registro")
                Console.WriteLine("4.- Salir")
                Console.WriteLine()
                Console.Write("Opcion [1-4]: ")
                opcion = Console.ReadLine

                Select Case opcion
                    Case 1 'Escribir registros
                        Do
                            mitabla.escriberegistro(datospersonales, cont)
                            Console.Write("Desea escribir otro registro?")
                            resp = Convert.ToChar(Console.Read)
                            Console.ReadLine()
                            cont += 1

                        Loop While (resp = "s"c)
                    Case 2 ' Mostrar todo el archivo
                        mitabla.leetodosregistros()
                    Case 3 'Buscar un registro concreto
                        pos = -1
                        cont = mitabla.dame_numero_registros

                        Console.WriteLine("Introduzca el nombre de la persona a buscar")
                        personabuscada = Console.ReadLine
                        pos = mitabla.buscar(personabuscada, cont)
                        If pos = -1 Then
                            Console.WriteLine("Registro no encontrado")
                        Else : mitabla.leeregistro(pos)

                        End If
                    Case 4
                        Exit Do
                    Case Else
                        Console.WriteLine("Opcion no válida")
                End Select
            Loop
        Finally
            mitabla.cerrartabla()
        End Try
    End Sub
End Module
```