



aprenderaprogramar.com

Instrucción leer e instrucción guardar. Introducción a archivos (ficheros) con Visual Basic (CU00327A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº26 del Curso Visual Basic Nivel I

29

INSTRUCCIÓN LEER E INSTRUCCIÓN GUARDAR. INTRODUCCIÓN A ARCHIVOS CON VB.

Advertencia: una manipulación incorrecta de ficheros puede dar lugar a la pérdida de datos, al malfuncionamiento de programas o a fallos del sistema operativo del ordenador. Es necesario disponer de copias de seguridad de toda información que se considere importante.

Guardar datos a un disco o recuperar los datos previamente guardados son procesos fundamentales en cualquier lenguaje informático. La importancia del almacenamiento de contenidos es obvia: envío de información a otros usuarios, posponer el trabajo varios días o semanas sin tener que introducir manualmente los datos de nuevo, acceso a información almacenada en sistemas remotos, etc. Incluso para desarrollos de software de relativamente corta longitud resulta relevante la gestión de datos, por ahorrar una cantidad de tiempo considerable.

Ficheros y bases de datos constituyen un cuerpo de conocimiento de cierta complejidad y extensión. Por motivos de espacio no vamos a hacer más que una breve introducción a ellos.

Podemos pensar en los ficheros o archivos de forma análoga a lo que sería un archivo físico: un lugar donde hay información almacenada. Ahora bien, eso no nos dice cómo se encuentra esa información: si está ordenada (alfabéticamente, de mayor a menor, en impresos con campos predefinidos), si son datos de un tipo o de varios tipos (numéricos, tipo texto, mezcla de tipos de datos...), si se leen usando alguna clave de interpretación (una partitura no se lee igual que un texto), etc. Todo esto es relevante y por ello normalmente sabremos qué tipo de datos contiene el fichero y cómo están ordenados antes de acceder a él. No es lo mismo guardar "3246" como número tipo *Integer* que como cadena de caracteres. Un tipo *Integer* ocupa 2 *bytes* y tiene una "clave" para su lectura, mientras que como caracteres cada cifra ocupa un *byte* y se lee como si de texto se tratara.

Un asunto a considerar es si la información está contenida en campos de longitud predefinida. Consideremos un archivo donde se almacenan los nombres, números de DNI y año de nacimiento de 3 personas. Podemos considerar diferentes casos:

a) Existen campos de longitud predefinida, que pueden carecer de información por no estar disponible.

Nombre: Alfredo	Nombre: Pedro González Aparicio	Nombre: Ramón González Pérez
DNI: 79555444	DNI:	DNI: 79666444
Año de nacimiento: 1975	Año de nacimiento: 1966	Año de nacimiento: 1981

En el fichero físico la información la guardamos en hojas preimpresas donde cada apartado tiene una longitud disponible para escribir en él. En el fichero informático la información la guardamos en

registros (un registro equivale a una hoja del fichero físico), constando cada registro de 3 campos (campo "Nombre", campo "DNI" y campo "Año de nacimiento"). Los campos serían por ejemplo:

Nombre: tipo texto 25 caracteres de longitud.

DNI: tipo entero largo (*Long*).

Año de nacimiento: tipo entero (*Integer*).

Cada registro ocupa $25 + 4 + 2 = 29$ bytes. Si tenemos tres registros el fichero ocupa $29 * 3 = 87$ bytes, independientemente de que los datos realmente existan o no, ya que el espacio se encuentra disponible para albergar un dato.

b) La información no está en campos de longitud predefinida.

```
Pedro González Aparicio. 1966.  
Alfredo Pérez Pérez. 79555444. 1975.  
Ramón González Pérez. 79666444.  
1981.
```

En el fichero físico la información la guardamos en hojas donde escribimos los datos disponibles para cada persona. En el fichero informático la información la guardamos en líneas (cada línea equivale a la información disponible de una persona) sabiendo que cada línea puede tener información de uno o varios aspectos (nombre, DNI y edad) y una longitud variable.

No sabemos cuántos bytes nos ocupa cada línea ni cuántos bytes nos ocupa el fichero. No sabemos en qué posición se encuentra una información dada porque no tenemos referencias de situación. Ante ello, optamos por hacer una lectura tipo "texto" en la que un número de 8 cifras nos va a requerir 8 bytes frente a los 4 bytes que supondrían como tipo *Long*.

- c) La información no está en campos de longitud predefinida pero se dispone de información referente a su ubicación.

Alfredo Pérez Pérez 79555444 1975	Pedro González Aparicio 1966	Ramón González Pérez 79666444 1981
Página 1 líneas 1 a 3	Página 2 líneas 1 a 2	Página 3 líneas 1 a 3

En el fichero físico la información la guardamos en hojas indexadas, donde cada dato dispone de un localizador. Por ejemplo, en la página 3 línea 3 encontramos el dato correspondiente al año de nacimiento de Ramón González. En el fichero informático la información la guardamos por *bytes*, sabiendo que cada tipo de dato ocupa un número de *bytes* determinado. Así si guardamos *Importe: 3235*, podríamos hacerlo ocupando *11 bytes* desglosados en *9 bytes* (uno por carácter, incluidos los espacios) asociados a "Importe:" como *tipo texto + 2 bytes* (para un tipo *Integer*) asociados al número *3235*.

También podríamos guardarlo todo como tipo texto ocupando *13 bytes*.

Una vez guardada la información, podríamos recuperarla indicándole al ordenador: *Extrae como tipo texto los bytes comprendidos entre 1 y 9 y como número los bytes comprendidos entre 10 y 11*.

Hemos tratado de introducir, a través de la analogía con un fichero físico, enfoques básicos para la comprensión y gestión de ficheros informáticos. La problemática, como se comprenderá, es muy amplia al intervenir numerosos factores (directorios, unidades de disco, sobrescritura de archivos, borrado accidental de datos, etc.) y posibilidades (datos ordenados, parcialmente ordenados o desordenados, diversidad de tipos de datos, tramos de ficheros con registros y campos, etc.).

Nosotros estudiaremos simplícidamente cómo podemos gestionar ficheros con *Visual Basic*.

Próxima entrega: CU00328A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=37&Itemid=61