Programas de Aplicación III

Tema 2. Elementos del lenguaje (Parte 3)

Luis Rodríguez Baena y María Dorrego Luxán

Universidad Pontificia de Salamanca (campus Madrid)
Facultad de Informática

Salida de mensajes por pantalla (I)

- Cuadros de mensajes (MsgBox).
  - Proporcionan un método para la salida de mensajes de alerta por pantalla.
  - Devuelve un entero que es posible capturar.

```
MsgBox mensaje[,tipoMensaje][, título]
```

- *mensaje*
  - Una expresión de cadena con el texto del mensaje.
    - ✓ Se pueden intercalar retornos de carro (`chr(10)` o `vbCrLf`) en la expresión para que ocupe varias líneas.

Salida de mensajes por pantalla (II)

- *tipoMensaje*
  - Una expresión numérica con el tipo de cuadro de diálogo (*véase tabla en la transparencia siguiente*).
    - ✓ El valor 0 es el tipo de cuadro por omisión.
    - ✓ Los valores de 1 a 5 indican el tipo y número de botones.
    - ✓ Los valores de 16 a 64 indican el tipo de icono.
    - ✓ Los valores 256 a 768 indican el botón por omisión.
    - ✓ El valor 4096 indica la modalidad del cuadro.
  - Los valores de los distintos grupos se pueden combinar sumándolos.

    ```
    vbQuestion+vbYesNo
    ```

- *título*.
  - Expresión de cadena con el título de la ventana.

Salida de mensajes por pantalla (III)

Constante	Valor	Descripción
VbOKOnly	0	Muestra solamente el botón Aceptar.
VbOKCancel	1	Muestra los botones Aceptar y Cancelar.
VbAbortRetryIgnore	2	Muestra los botones Anular, Reintentar e Ignorar.
VbYesNoCancel	3	Muestra los botones Sí, No y Cancelar.
VbYesNo	4	Muestra los botones Sí y No.
VbRetryCancel	5	Muestra los botones Reintentar y Cancelar.
VbCritical	16	Muestra el icono de mensaje crítico.
VbQuestion	32	Muestra el icono de pregunta de advertencia.
VbExclamation	48	Muestra el icono de mensaje de advertencia.
VbInformation	64	Muestra el icono de mensaje de información.
vbDefaultButton1	0	El primer botón es el predeterminado.
vbDefaultButton2	256	El segundo botón es el predeterminado.
vbDefaultButton3	512	El tercer botón es el predeterminado.
vbDefaultButton4	768	El cuarto botón es el predeterminado.
VbApplicationModal	0	Aplicación modal; el usuario debe responder al cuadro de mensajes antes de poder seguir trabajando en la aplicación actual.
VbSystemModal	4096	Sistema modal; se suspenden todas las aplicaciones hasta que el usuario responda al cuadro de mensajes.

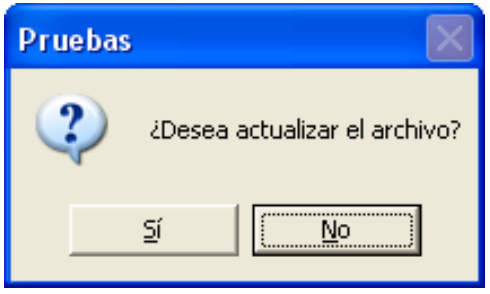
Salida de mensajes por pantalla (IV)

- Valor de retorno.
  - MsgBox puede trabajar como una función devolviendo un valor de retorno de tipo entero.
  - Valores devueltos:

Constante	Valor	Descripción
VbOK	1	Aceptar
VbCancel	2	Cancelar
VbAbort	3	Anular
VbRetry	4	Reintentar
VbIgnore	5	Ignorar
VbYes	6	Sí
VbNo	7	No

Salida de mensajes por pantalla (V)

- Ejemplo:
  - Controlar que botón se ha pulsado en un cuadro de diálogo de tipo pregunta con dos botones Si y No (No es la opción por omisión).

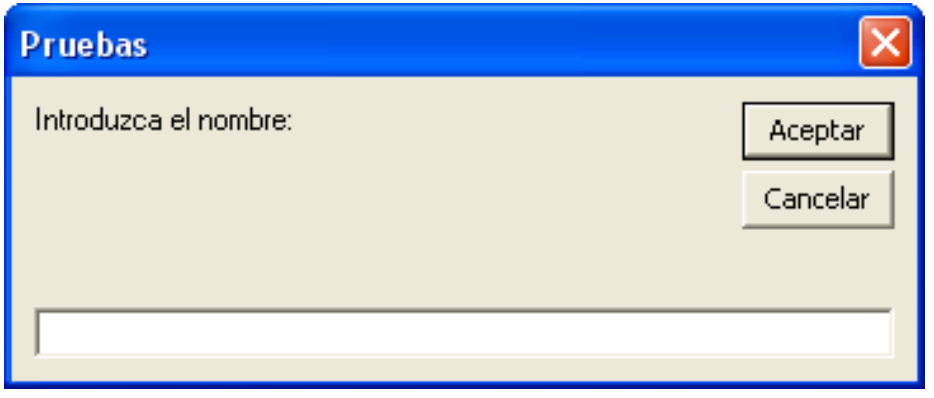



```
If MsgBox("¿Desea actualizar el archivo?", _
      vbQuestion + vbYesNo + vbDefaultButton2, _
      "Pruebas") = vbYes Then
  'Acciones del botón Si
Else
  'Acciones del botón No
End If
```

Entrada de datos: InputBox

- Función InputBox
  - Muestra un cuadro de diálogo con una caja de texto en la que el usuario puede introducir información.

    `InputBox(`*`mensaje`*`[, `*`título`*`][, `*`valorOmisión`*`])`
  - Devuelve un dato de tipo `String`.



Archivos secuenciales (I)

- **Todos los archivos en VB se deben utilizar como archivos auxiliares, no siendo lo más adecuado para almacenar información estructurada. En ese caso se deben utilizar bases de datos.**
- Archivos de texto cuya información puede o no estar estructurada.
- Instrucción `Open`.
  - Activa operaciones de E/S sobre un archivo.

    `Open` *`NombreArchivo`* `[For` *`modo`*`] As [#]`*`NúmeroArchivo`*
    - ✓ *`NombreArchivo`*. Expresión de cadena con el identificador de archivo.
    - ✓ *`Modo`*. Puede contener los valores:
      - × `Input`. Para lectura.
      - × `Output`. Para escritura (coloca el puntero al comienzo del archivo).
      - × `Append`. Para escritura (coloca el puntero al final del archivo).
    - ✓ *`NúmeroArchivo`*. Número entre 1 y 511 que identifica al archivo.
- Instrucción Close.
  - Desactiva las operaciones de E/S sobre uno o todos los archivo.

    `Close [[#]NúmeroArchivo]…`

Archivos secuenciales (II)

- Lectura de datos.
  - Función `Input`.

    `Input[$](n,[#]`*`NúmeroArchivo`*`)`
    - ✓ Lee de un archivo identificado por *`NúmeroArchivo`* el número de bytes (`n`) especificado.

```
'Lee todos los caracteres de Archivo.txt
Open "c:\Archivo.txt" For Input As 1
texto = Input(FileLen("c:\Archivo.txt"), 1)
Close 1
```

  - Intrucción `Input #`.

    `Input #`*`NúmeroArchivo`*`,`*`ListaVariables`*
    - ✓ Leer del archivo especificado e introduce los datos en las variables.
    - ✓ Cada dato debe estar separado por comas.
    - ✓ Los datos de tipo cadena deben estar encerrados entre comillas.
    - ✓ No lee las comillas ni caracteres de fin de línea y retorno de carro.
  - Instrucción `Line Input #`.

    `Line Input #`*`NúmeroArchivo`*`, `*`variable`*
    - ✓ Lee todos los caracteres del archivo especificado hasta el fin de línea.

Archivos secuenciales (III)

- Escritura de datos.
  - Instrucción `Print #`.

    `Print #`*`NúmeroArchivo,expresiones`*
    - ✓ Escribe en el archivo especificado las expresiones que aparecen a continuación.
    - ✓ Las expresiones se deben separar por punto y coma (;) para que aparezcan una a continuación de otra.
    - ✓ Inserta al final un carácter de fin de línea y retorno de carro.
  - Instrucción `Write`.

    `Write #NúmeroArchivo,expresiones`
    - ✓ Escribre las expresiones en el archivo especificado.
    - ✓ Las expresiones deben ir separadas por comas.
    - ✓ En el archivo, escribe las expresiones separadas por comas y las cadenas por comillas dobles (para que puedan ser leidas por `Input #`).

Archivos directos (I)

- Archivos de acceso aleatorio con registros de la misma longitud.
- Instrucción `Open`.

  `Open` *`NombreArchivo`* `[For Random] [Access` *`acceso`*`] As [#]`*`NúmeroArchivo`* `Len =` *`LongitudRegistro`*

  - *`acceso`* puede tomar los siguientes valores.
    - ✓ `Read` (sólo lectura).
    - ✓ `Write` (sólo escritura)
    - ✓ *`Read Write`* (lectura y escritura, opción por omisión).
  - *`LongitudRegistro`* indica el número de bytes del registro.
- Instrucción `Close`.

Archivos directos (II)

- Lectura de información. Instrucción `Get`.

  `Get [#] NúmeroArchivo,[NúmeroRegistro],NombreVariable`

  - *`NúmeroRegistro`* indica el registro a leer. Si se omite realiza la lectura del siguiente registro.
  - *`NombreVariable`*, es la variable donde se introducirá la información.
    - ✓ Su longitud debe coincidir con la especificada en la cláusula `Len` del `Open`.
    - ✓ Puede ser una variable estándar o un tipo definido por el usuario (registro).

- Escritura. Instrucción `Put`.

  `Put [#]NumeroArchivo,[NúmeroRegistro],NombreVariable`

- Posicionamiento. Instrucción `Seek`.

  `Seek [#]NúmeroArchivo, posición`

  - Establece la posición del registro de la siguiente operación de E/S.

Archivos binarios

- Permite el acceso a un archivo con cualquier tipo de información.
- Instrucción `Open`.

  `Open` *NombreArchivo* `[For Binary] As [#]`*NúmeroArchivo*
- Instrucción `Close`.
- Lectura de información. Instrucción `Get`.

  `Get [#]` *NúmeroArchivo*`,[`*byte*`],`*NombreVariable*
  - *byte* indica el byte a partir del cual se va a leer. Si se omite realiza la lectura del siguiente registro.
- Escritura. Instrucción `Put`.

  `Put [#]`*NumeroArchivo*`,[`*byte*`],`*NombreVariable*
- Posicionamiento. Instrucción `Seek`.

  `Seek [#]`*NúmeroArchivo*`,` *posición*
  - Establece la posición del byte de la siguiente operación de E/S.

Otras instrucciones y funciones de archivo

Función / Instrucción	Observaciones
EOF(*n*)	Verdadero si se ha llegado al final del archivo *n*.
Dir(*espArchivo*)	Devuelve una cadena con el primer archivo que cumpla la especificación.
FileCopy *origen,destino*	Copia el archivo *origen* en *destino.*
FileLen(*esp.archivo*)	Devuelve el número de bytes del archivo. *esp.Archivo* es una expresión de cadena
FreeFile()	Devuelve el primer número de archivo libre. Se utiliza en combinación con Open para asignar números de archivo.
Loc(*n*)	Devuelve la última posición de lectura o escritura en un archivo abierto. En archivos binarios devuelve el último byte, en los aleatorios el último registro.
Lof(*n*)	Devuelve la longitud de un archivo abierto (FileLen lo hace en un archivo cerrado).
Seek(*n*)	Devuelve la posición de la siguiente operación de lectura o escritura. En archivos aleatorios devuelve el número de registro, en el resto el número de byte.