

Visual Basic 6.0

Prácticas.

Prácticas

Javier García de Jalón · José Ignacio Rodríguez
Alfonso Brazález · Eduardo Carrasco · Jesús Calleja

Índice

Ejercicio 1: Desarrollo de un editor de texto: proyecto MiNotepad.....	1
Ejercicio 2: Introducción de mejoras: tener en cuenta si el texto se ha modificado, y no cerrar la aplicación sin avisar que se puede perder información (proyecto MiNotepad2)	2
Ejercicio 3: Introducción de mejoras: búsqueda de texto (proyecto MiNotepad3)	3

Ejercicio 1: Desarrollo de un editor de texto: proyecto MiNotepad

El ejecutable de este ejercicio está en *Q:\Infor1\Prac14\MiNotepad\MiNotepad.exe*. Crea el directorio *G:\Prac14\MiNotepad* y guarda en él los ficheros de este ejercicio, que se llamarán *MiNotepad.vbp* y *MiNotepad.frm*.

Se trata de crear un procesador de texto sencillo similar a *Notepad*. Este procesador de texto se irá creando progresivamente a lo largo de la práctica. En este primer ejercicio se creará **la versión más sencilla**, que luego se irá progresivamente ampliando.

Desde el punto de vista de los controles este ejemplo tiene solamente un **formulario**, una **caja de texto** que ocupa todo el espacio disponible en el formulario, y unos **menús**.

Es muy importante que la caja de texto ocupe siempre el máximo espacio disponible en el formulario. Para ello se puede utilizar el evento **resize** del formulario.

En este ejercicio se deben poder utilizar los comandos **New**, **Open**, **Save**, **Save As** y **Exit**, de acuerdo con las siguientes descripciones:

- **New**. Este comando inicializa simplemente la caja de texto a la cadena vacía "". Hasta que el fichero se guarde por primera vez en la **barra de título** aparecerá el mensaje "**Untitled – MiNotepad**".
- **Open**. Este comando abre una caja de diálogo –un **Common Dialog Control**¹– para seleccionar un fichero de texto, lo lee y muestra su contenido en la caja de texto. Se establecerá un **filtro** para seleccionar bien los ficheros de texto, bien todos los ficheros ("Texto (*.txt)|*.txt|All Files (*.*)|*..*"). Una vez abierto el fichero, en la barra de título aparecerá un mensaje con el nombre del fichero y el nombre de la aplicación, en la forma "**Fichero.txt – MiNotepad**".
- **Save**. Actualiza el contenido del fichero de acuerdo con el texto actual de la caja de texto. Si el fichero existe, simplemente lo actualiza; si no existe, realiza la misma función que **Save As**.
- **Save As**. Utilizando el mismo **Common Dialog Control** se abre el cuadro de diálogo **Save As** típico de todas las aplicaciones de **Windows**. La extensión por defecto (la que se utiliza si el usuario no pone otra) será ***.txt**. Después de guardar el fichero en el disco, en la barra de título aparecerá el mensaje "**Fichero.txt – MiNotepad**".
- **Exit**. Con este comando la aplicación se termina tal y como esté, es decir sin salvar y sin hacer ningún tipo de averiguación adicional.

¹ Si el **Common Dialog Control** no aparece en el **ToolBox** de **Visual Basic 6.0**, se puede hacer que aparezca por medio del comando **Components** en el menú **Projects** (se llama exactamente **Microsoft Common Dialog Control 6.0**).

Además se deberá poder cambiar el **Font** y el **color de fondo**, utilizando también el **Common Dialog Control**, con los comandos **ShowFont** y **ShowColor** en los eventos **Click** de los ítem **Set Font** y **Set Bgnd Color**, que están en el menú **Edit**. El programa deberá funcionar a este respecto igual que el modelo, incluyendo la posibilidad de efectos especiales (color del texto, tachado, y subrayado).

Ejercicio 2: Introducción de mejoras: tener en cuenta si el texto se ha modificado, y no cerrar la aplicación sin avisar que se puede perder información (proyecto MiNotepad2)

Comienza este ejercicio creando en el directorio **G:\Prac14** una carpeta llamada **MiNotepad2** y copia en ella los ficheros del ejercicio anterior. Después, cámbiales el nombre y haz que se llamen **MiNotepad2.vbp** y **MiNotepad2.frm**. Cuando hayas hecho estos cambios ya podrás abrirlos y seguir trabajando sobre ellos.

El ejecutable de este ejercicio está en el directorio **Q:\Infor1\Prac14\MiNotepad2** y se llama **MiNotepad2.exe**. Cópialo a tu directorio y observa detenidamente cómo funciona.

La principal diferencia de este ejercicio con el anterior es la programación de una serie de medidas de seguridad para no perder información involuntariamente. Se puede perder información siempre que el contenido de la caja de texto esté sin guardar en el disco o siempre que se haya introducido alguna modificación en el texto y no se haya actualizado el fichero de disco. Para establecer estas medidas de seguridad se hará lo siguiente:

- Se creará una variable de tipo **boolean** llamada **Actualizado**. Siempre que el fichero contenga el mismo texto que la caja de texto esta variable valdrá **True**. Cuando la caja de texto haya sufrido algún cambio y el fichero no esté actualizado, dicha variable valdrá **False**. Para simplificar, aunque el cambio en la caja de texto la deje como estaba (por ejemplo, añadir una letra y luego borrarla), la variable **Actualizado** valdrá **False**.
- Algunos editores de texto **indican de un modo visual** que el fichero no está actualizado (no es el caso de **Notepad**). En este programa cuando el fichero esté sin actualizar aparecerá un **asterisco (*)** en última posición de la barra de títulos, justamente al final del nombre de la aplicación. Por ejemplo, con un fichero llamado **Prueba1.txt** la barra de títulos deberá mostrar **“Prueba1.txt – MiNotepad”** si el fichero está actualizado y **“Prueba1.txt – MiNotepad*”** si no lo está. Para conseguir esto habrá que modificar la barra de títulos cada vez que se introduzca un cambio en el texto (poner el asterisco) y cada vez que se guarde el texto en el disco con **Save** o **Save As** (quitar el asterisco).
- La segunda medida de seguridad será pedir confirmación antes de cerrar el programa cuando está en la situación de **Actualizado = False**. La información se puede perder en los supuestos siguientes:
 - **Al cerrar la aplicación.** En este caso lo mejor es utilizar el evento **QueryUnload** de los formularios. El procedimiento para gestionar este evento tiene dos argumentos. Para este ejercicio sólo el primero, llamado **Cancel**, tiene importancia². Si **Cancel** recibe cualquier valor distinto de cero, no se prosigue con el proceso de descarga del formulario. El evento **QueryUnload** deberá sacar un mensaje de confirmación tal como el que se ve en la figura 1. Si se pulsa **Yes** se actualiza el fichero y se sale de la aplicación; si se pulsa **No** se sale de la aplicación sin actualizar el fichero y si se pulsa

² El segundo argumento **unloadMode** sirve para saber de dónde viene la orden de descargar el formulario.

Cancel se vuelve a la aplicación sin hacer nada (argumento **Cancel** del evento **QueryUnload** distinto de cero).

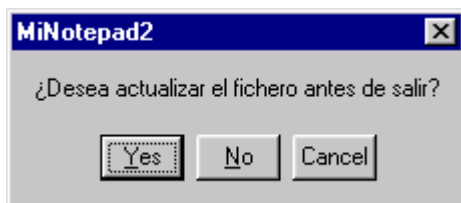


Figura 1.

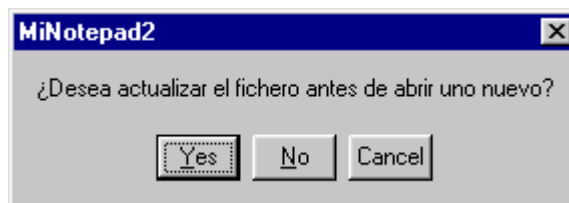


Figura 2.

- Teniendo un fichero abierto y sin actualizar, al abrir un nuevo fichero con **File/Open** o al crear un nuevo texto con **File/New** debe ofrecer la opción de guardar los cambios en el texto anterior. Esta posibilidad debe gestionarse directamente en el evento **Click** de los comandos de menú correspondientes. En este caso aparecerá una caja de mensajes ligeramente diferente, mostrada asimismo en la figura 2.

Ejercicio 3: Introducción de mejoras: búsqueda de texto (proyecto MiNotepad3)

Comienza este ejercicio creando en el directorio **G:\Prac14** una carpeta llamada **MiNotepad3** y copia en ella los ficheros del ejercicio anterior. Después, cámbiales el nombre y haz que se llamen **MiNotepad3.vbp** y **MiNotepad3.frm**. Cuando hayas hecho estos cambios ya podrás abrirlos y seguir trabajando sobre ellos.

El ejecutable de este ejercicio está en el directorio **Q:\Infor1\Prac14\MiNotepad3** y se llama **MiNotepad3.exe**. Cópialo a tu directorio y observa detenidamente cómo funciona.

En primer lugar se trata de encontrar un texto (unas palabras o una frase) dentro del texto contenido en la caja de texto. Al elegir el comando **Find** en el menú **Search** se abre una caja de diálogo (que en realidad es un formulario) tal como la mostrada en la figura 3 (no hace falta buscar en ambas direcciones ni poder distinguir entre mayúsculas y minúsculas. Esas posibilidades aparecen en el formulario con fines únicamente “decorativos”). El usuario tecleará el texto a encontrar y pulsará el botón **Find Next**. En ese momento el programa localiza la primera aparición del texto buscado y lo selecciona para que aparezca visible. Si no se encuentra el texto buscado aparece el mensaje de la figura 4. Si se encuentra y el usuario vuelve a pulsar en **Find Next** busca la siguiente aparición y así hasta llegar al fin del fichero. Al llegar al final del fichero aparece un mensaje que lo indica y que pregunta si se quiere recomenzar la búsqueda desde el principio, como se muestra en la Figura 5.

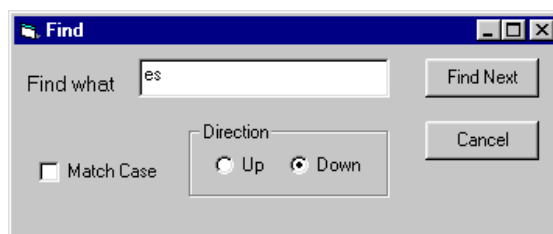


Figura 3.

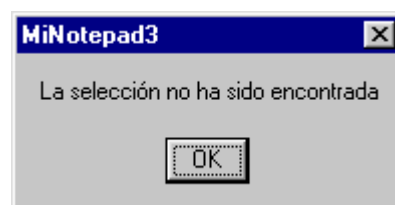


Figura 4.

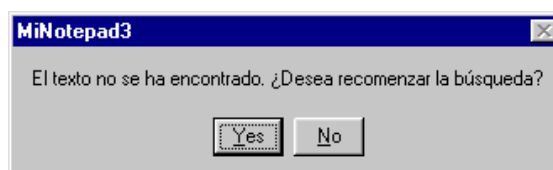


Figura 5.

Este ejercicio es un poco más complicado de lo habitual. Para seleccionar un texto en una caja de texto se debe dar valor a las propiedades **SelStart** y **SelLength** de dicha caja de texto. De todas

formas esto no es suficiente: es necesario poner a **False** la propiedad **HideSelection** de la caja de texto, para que el texto aparezca seleccionado aunque el control de que se trate –la caja de texto- no tenga el **focus**.

Conviene también que el formulario **frmFind** (suponiendo que se llame así el formulario de la Figura 3) de búsqueda de texto siempre esté visible por encima del formulario principal, aunque sea éste el formulario activo. Esto se consigue abriéndolo de una forma especial cuando el usuario ejecuta el comando **Search/Find**. El código del procedimiento que se ejecuta es el siguiente (suponiendo que el formulario principal se llame **Form1**):

```
Private Sub mnuSearchFind_Click()  
    ' abre una ventana que no es modal, pero que está  
    ' siempre delante de form1 (depende de form1)  
    frmFind.Show vbModeless, Form1  
End Sub
```

Por otra parte, es conveniente que el formulario **frmFind** no aparezca en la barra de tareas de **Windows** como si fuera una aplicación más. Ello se consigue modificando la propiedad **ShowInTaskbar** en la forma **frmFind.ShowInTaskbar=False**.