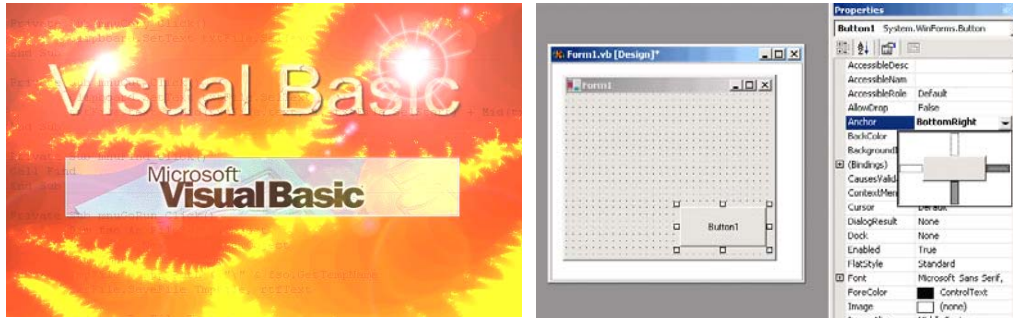


CURSO

Curso Completo de Visual Basic 6.0



Escuela Superior de Ingenieros Industriales

UNIVERSIDAD DE NAVARRA

Javier García de Jalón · José Ignacio Rodríguez

Alfonso Brazález · Patxi Funes · Eduardo Carrasco · Jesús Calleja

2. ENTORNO DE PROGRAMACIÓN VISUAL BASIC 6.0

2.1 INTRODUCCIÓN : ¿QUÉ ES VISUAL BASIC 6.0?

Visual Basic 6.0 es una excelente herramienta de programación que permite crear aplicaciones propias (programas) para *Windows 95/98SE/XP/* o *Windows Vista*. Con ella se puede crear desde una simple calculadora hasta una hoja de cálculo de la talla de *Excel* (en sus primeras versiones...), pasando por un procesador de textos o cualquier otra aplicación que se le ocurra al programador.

Sus aplicaciones en Ingeniería son casi ilimitadas: representación de movimientos mecánicos o de funciones matemáticas, gráficas termodinámicas, simulación de circuitos, etc.

Este programa permite crear ventanas, botones, menús y cualquier otro elemento de *Windows* de una forma fácil e intuitiva. El lenguaje de programación que se utilizará será el *Basic*, que se describirá en el siguiente capítulo.

A continuación se presentarán algunos aspectos del entorno de trabajo de *Visual Basic 6.0*: menús, opciones, herramientas, propiedades, etc.

2.2 EL ENTORNO DE VISUAL BASIC 6.0

Visual Basic 6.0 tiene todos los elementos que caracterizan a los programas de *Windows* e incluso alguno menos habitual. En cualquier caso, el entorno de *Visual Basic 6.0* es muy lógico y natural, y además se puede obtener una descripción de la mayoría de los elementos clicando en ellos para seleccionarlos y pulsando luego la tecla <F1>.

2.2.1 La barra de menús y las barras de herramientas

La *barra de menús* de *Visual Basic 6.0* resulta similar a la de cualquier otra aplicación de *Windows*, tal y como aparece en la **Figura 2.2**.

Bajo dicha barra aparecen las *barras de herramientas*, con una serie de botones que permiten acceder fácilmente a las opciones más importantes de los menús.

En *Visual Basic 6.0* existen cuatro barras de herramientas: *Debug*, *Edit*, *Form Editor* y *Standard*. Por defecto sólo aparece la barra *Standard*, aunque en la **Figura 2.2** se muestran las cuatro. Haciendo Click con el botón derecho sobre cualquiera de las barras de herramientas aparece un menú contextual con el que se puede hacer aparecer y ocultar cualquiera de las barras. Al igual que en otras aplicaciones de *Windows 95/98/XP/VISTA*, también pueden modificarse las barras añadiendo o eliminando botones (opción *Customize*).

En la barra de herramientas *Standard* también se pueden ver a la derecha dos recuadros con números, que representan cuatro propiedades del formulario referentes a su posición y tamaño que se verán más adelante: *Top* y *Left*, que indican la posición de la esquina superior izquierda del formulario, y también *Height* y *Width*, que describen el tamaño del mismo en unas unidades llamadas *twips*, que se corresponden con la vigésima parte de un *punto* (una pulgada tiene 72 puntos y 1440 twips). Las dimensiones de un control aparecen también cuando con el ratón se arrastra sobre el formulario, según se puede ver en la **Figura 2.1**.

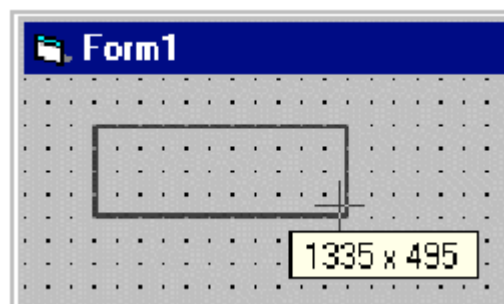


Figura 2.1. Información visual sobre el tamaño de un control.

Los botones de la barra de herramientas **Standard** responden a las funciones más importantes: abrir y/o guardar nuevos proyectos, añadir formularios, hacer visibles las distintas ventanas del entorno de desarrollo, etc. Todos los botones tienen su correspondiente comando en alguno de los menús. Son importantes los botones que permiten arrancar y/o parar la ejecución de un proyecto, pasando de modo diseño a modo de ejecución y viceversa.

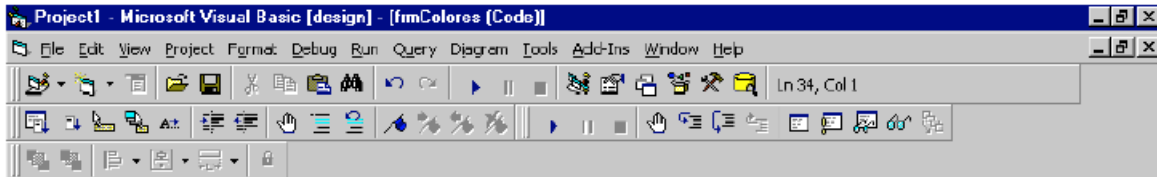


Figura 2.2. Barra de menús y barras de herramientas de Visual Basic 6.0.

El menú **File** tiene pocas novedades. Lo más importante es la distinción entre **proyectos** y todos los demás ficheros. Como ya se ha dicho, un proyecto reúne y organiza todos los ficheros que componen el programa o aplicación (hace la función de una *carpeta* que contuviera *apuntes*).

Estos ficheros pueden ser *formularios*, *módulos*, *clases*, *recursos*, etc. **Visual Basic 6.0** permite tener más de un proyecto abierto simultáneamente, lo cual puede ser útil en ocasiones.

Con el comando **AddProject** se añade un nuevo proyecto en la ventana **Project Manager**. Con los comandos **Open Project** o **New Project** se abre o se crea un nuevo proyecto, pero cerrando el o los proyectos que estuvieran abiertos previamente.

En este menú está el comando **Make ProjectName.exe**, que permite crear ejecutables de los proyectos.

Tampoco el menú **Edit** aporta cambios importantes sobre lo que es habitual. Por el contrario el menú **View**, generalmente de poca utilidad, es bastante propio de **Visual Basic 6.0**. Este menú permite hacer aparecer en pantalla las distintas ventanas del entorno de desarrollo, así como acceder a un formulario o al código relacionado con un control (que también aparece si se clica dos veces en dicho control), y manejar funciones y procedimientos.

El menú **Project** permite añadir distintos tipos de elementos a un proyecto (formularios, módulos, etc.). Con **Project/Properties** se puede elegir el tipo de proyecto y determinar el formulario con el que se arrancará la aplicación (**Startup Object**). Con el comando **Components** se pueden añadir nuevos controles a la Toolbox que aparece a la izquierda de la pantalla.

El menú **Format** contiene opciones para controlar el aspecto de la aplicación (alinear controles, espaciarlos uniformemente, etc.). Los menús **Debug** y **Run** permiten controlar la ejecución de las aplicaciones. Con **Debug** se puede ver en detalle cómo funcionan, ejecutando paso a paso, yendo hasta una línea de código determinada, etc.

Esto es especialmente útil cuando haya que encontrar algunos errores ejecutando paso a paso, o viendo resultados intermedios.

En el menú **Tools** se encuentran los comandos para arrancar el **Menu Editor** (para crear menús, como se verá en el Apartado **Menús**) y para establecer las **opciones** del programa.

En **Tools/Options** se encuentran las opciones relativas al proyecto en el que se trabaja. La lengüeta **Environment** determina las propiedades del entorno del proyecto, como las opciones para **actualizar** o no los ficheros antes de cada ejecución; en **General** se establece lo referente a la retícula o **grid** que aparece en el formulario; **Editor** permite establecer la necesidad de **declarar** todas las variables junto con otras opciones de edición, como si se quieren ver o no todos los procedimientos juntos en la misma ventana, y si se quiere ver una línea separadora entre procedimientos; **Editor Format** permite seleccionar el tipo de letra y los códigos de color con los que aparecen los distintos elementos del código.

La opción **Advanced** hace referencia entre otras cosas a la opción de utilizar **Visual Basic 6.0** en dos formatos SDI (**Single Document Interface**) y MDI (**Multiple Document Interface**).

Por último, la ayuda (siempre imprescindible y en el caso de **Visual Basic 6.0** particularmente bien hecha) que se encuentra en el menú **Help**, se basa fundamentalmente en una clasificación temática ordenada de la información disponible (**Contents**), en una clasificación alfabética de la información (**Index**) y en la búsqueda de información sobre algún tema por el nombre (**Search**).

Como ya se ha mencionado, la tecla <F1> permite una ayuda directa sensible al contexto, esto es dependiente de donde se haya hecho click con el ratón (o de lo que esté seleccionado).

2.2.2 Las herramientas (toolbox)

Es la zona reservada a los **controles** con los que se puede diseñar la pantalla de la aplicación. Estos controles son por ejemplo botones, etiquetas, cajas de texto, zonas gráficas, etc. Para introducir un control en el formulario simplemente hay que clicar en el icono adecuado de la **toolbox** y colocarlo en el formulario con la posición y el tamaño deseado, clicando y arrastrando con el ratón. Clicando dos veces sobre el icono de un control aparece éste en el centro del formulario y se puede modificar su tamaño y/o trasladar con el ratón como se desee.



Figura 2.3. Caja de componentes (**Toolbox**)

El número de controles que pueden aparecer en esta ventana varía con la configuración del sistema. Para introducir nuevos componentes se utiliza el comando **Components** en el menú **Project**, con lo cual se abre el cuadro de diálogo mostrado en la **Figura 2.4**.

2.3 FORMULARIOS (**FORMS**) Y MÓDULOS

Los **formularios** son las zonas de la pantalla sobre las que se diseña el programa y sobre las que se sitúan los controles o herramientas de la **toolbox**.

Al ejecutar el programa, el **form** se convertirá en la ventana de la aplicación, donde aparecerán los botones, el texto, los gráficos, etc.

Para lograr una mejor presentación existe una malla o retícula (**grid**) que permite alinear los controles manualmente de una forma precisa (evitando tener que introducir coordenadas continuamente). Esta malla sólo será visible en el proceso de diseño del programa; al ejecutarlo no se verá. De cualquier forma, se puede desactivar la malla o cambiar sus características en el menú **Tools/Options/General**, cambiando la opción **Align Controls to Grid**.

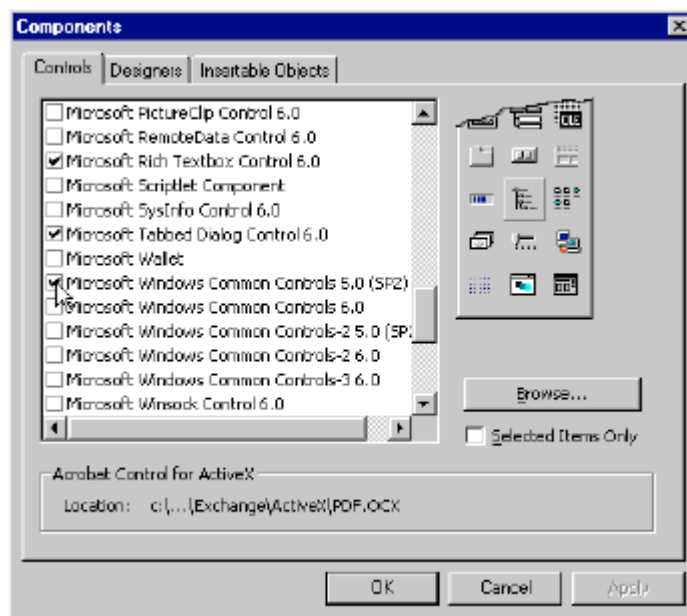


Figura 2.4. Cuadro de diálogo Components.



Exteriormente, los formularios tienen una estructura similar a la de cualquier ventana. Sin embargo, también poseen un código de programación que estará escrito en **Basic**, y que controlará algunos aspectos del formulario, sobre todo en la forma de reaccionar ante las acciones del usuario (eventos). El formulario y los controles en él situados serán el esqueleto o la base del programa.

Una aplicación puede tener varios formularios, pero siempre habrá uno con el que arrancará la aplicación que se determina con *Project/Properties*, en *Startup Objects*.

Resumiendo, cuando se vaya a crear un programa en *Visual Basic 6.0* habrá que seguir dos pasos:

1. **Diseñar y preparar la parte gráfica (formularios, botones, menús, etc.)**
2. **Realizar la programación que gestione la respuesta del programa ante los distintos eventos.**

2.4 LA VENTANA DE PROYECTO (*PROJECT*)

Esta ventana, mostrada en la **Figura 2.5**, permite acceder a los distintos formularios y módulos que componen el proyecto. Desde ella se puede ver el diseño gráfico de dichos formularios (botón *View Object* ) , y también permite editar el código que contienen (botón *View Code* ).

Estos botones están situados en la parte superior de la ventana, debajo de la barra de títulos.

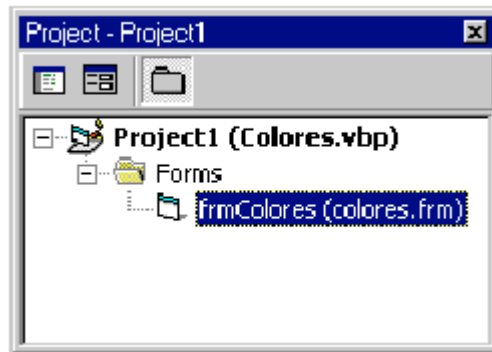


Figura 2.5. Ventana de proyecto.

Los *módulos estándar* (ficheros *.bas) contienen sólo código que, en general, puede ser utilizado por distintos formularios y/o controles del proyecto e incluso por varios proyectos.

Por ejemplo puede prepararse un módulo estándar de funciones matemáticas que sea de utilidad general. Normalmente contienen siempre algunas declaraciones de variables globales o *Public*, que serán accesibles directamente desde todos los formularios.

Los *módulos de clase* (ficheros *.cls) contienen clases definidas por el usuario. Las clases son como formularios o controles complejos, sin interface gráfica de usuario.

Todos los objetos **Visual Basic 6.0** tienen unas propiedades que los definen: su *nombre (name)*, su *etiqueta o título (caption)*, el *texto que contiene (text)*, su *tamaño y posición*, su *color*, si está *activo o no (enabled)*, etc.

La **Figura 2.6** muestra parcialmente las *propiedades* de un formulario. Todas estas propiedades se almacenan dentro de cada control o formulario en forma de *estructura* (similar a las del lenguaje C).

Por tanto, si por ejemplo en algún momento se quiere modificar el nombre de un botón basta con hacerlo en la ventana de propiedades (al diseñar el programa) o en el código en **Basic** (durante la ejecución), mediante el *operador punto (.)*, en la forma:

```
Boton1.Name = "NuevoNombre"
```

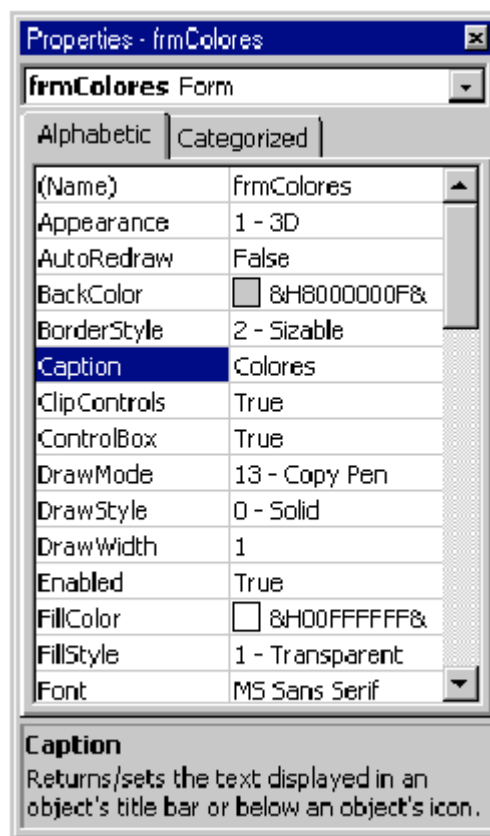


Figura 2.6. Ventana de propiedades.

Para realizar una modificación de las propiedades de un objeto durante el diseño del programa, se activará la ventana de propiedades (con el menú, con el botón de la barra de herramientas o pulsando <F4>). Esta ventana tiene dos lengüetas, que permiten ordenar las propiedades *alfabéticamente* o por *categorías*. Utilizando la forma que sea más cómoda se localizará con ayuda de la barra de desplazamiento la propiedad que se quiera modificar.

Al hacer click sobre ella puede activarse un menú desplegable con las distintas opciones, o bien puede modificarse directamente el valor de la propiedad.

Si esta propiedad tiene sólo unos valores fijos (por ejemplo, los colores), puede abrirse un cuadro de diálogo para elegir un color, o el tamaño y tipo de letra que se desee si se trata de una propiedad *Font*.

La **Figura 2.7** muestra la ventana *FormLayout*, que permite determinar la posición en la que el formulario aparecerá sobre la pantalla cuando se haga visible al ejecutar la aplicación.

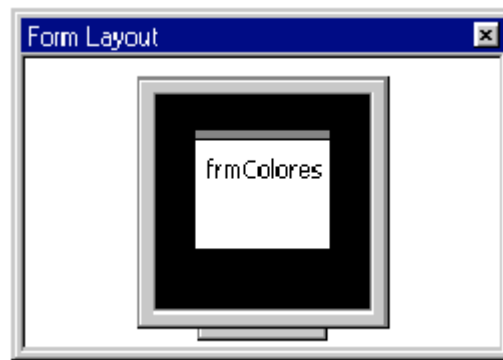


Figura 2.7. Ventana Form Layout.

2.6 CREACIÓN DE PROGRAMAS EJECUTABLES

Una vez finalizada la programación de la nueva aplicación, la siguiente tarea suele consistir en la creación de un programa ejecutable para su distribución e instalación en cuantos ordenadores se desee, incluso aun en ellos no esté instalado *Visual Basic 6.0*.

Para crear un programa ejecutable se utiliza el comando *Make nombreProyecto.exe ...* en el menú *File*. De esta manera se generará un fichero cuya extensión será **.exe*.

Para que este programa funcione en un ordenador solamente se necesita que el fichero *MSVBVM60.DLL* esté instalado en el directorio *c:\Windows\System* o *c:\Winnt\System32*.

En el caso de proyectos más complejos en los que se utilicen muchos controles pueden ser necesarios más ficheros, la mayoría de ellos con extensiones **.ocx*, **.vbx* o **.dll*. Para saber en cada caso cuáles son los ficheros necesarios se puede consultar el fichero **.vbp* que contiene la descripción completa del proyecto.

Casi todos esos ficheros necesarios se instalan automáticamente al instalar el compilador de *Visual Basic 6.0* en el ordenador.

En el caso de el programa se vaya a utilizar en un ordenador en el que no esté instalado *Visual Basic* o de que en el proyecto se hayan utilizado controles comerciales (como los *Crystal Reports* para la creación de informes, los *Sheridan Data Widgets* o los *True DBGrid* de *Apex* para la gestión de bases de datos, etc.), puede resultar interesante construir unos *CDs de instalación* que simplifiquen la tarea de instalar el programa en cualquier ordenador sin tener que ver en cada caso cuáles son los ficheros que faltan. *Visual Basic 6.0* dispone de un *Asistente (Wizard)* que, interactivamente, simplifica enormemente la tarea de creación de CDs de instalación.

Este *Asistente* está en el mismo grupo de programas que *Visual Basic 6.0* y se llama *Package and Deployment Wizard*.

2.7 CÓMO UTILIZAR EL HELP

Visual Basic 6.0 dispone de un *Help* excelente, como la mayoría de las aplicaciones de *Microsoft*. En esta nueva versión la ayuda se ofrece a través de una interface de usuario similar a la de *Internet Explorer*.

Estando seleccionado un control, una propiedad o un formulario, o estando seleccionada una palabra clave en una ventana de código, esta ayuda se puede utilizar *de modo sensible al contexto* pulsando la tecla <F1>.

También se puede ver toda la información disponible de modo general y ordenado por temas con el comando *Help/Contents*.



Figura 2.8. Ayuda de Visual Basic 6.0.

Otra forma de acceder a la ayuda es mediante las opciones del menú **Help**. Así mediante la opción **Index** se puede obtener información sobre muchos términos relacionados con **Visual Basic 6.0**.

Una vez obtenida la ayuda sobre el término solicitado se pueden encontrar temas relacionados con ese término en la opción **See Also**.

En el caso de que se haya solicitado ayuda sobre un determinado tipo de control (en la **Figura 2.9** se ha hecho con los botones de comando), se podría acceder también a la ayuda sobre todos y cada uno de sus propiedades, eventos y métodos con las opciones **Properties**, **Methods** y **Events**, respectivamente.



Figura 2.9. Ayuda sobre el botón de comando.

La solapa **Contents** de la ventana de ayuda sirve para acceder a una pantalla en la que la ayuda está ordenada por temas, la de **Index** sirve para acceder a una pantalla en la que se podrá realizar una búsqueda a partir de un término introducido por el usuario, entre una gran lista de términos ordenados alfabéticamente.

Al teclear las primeras letras del término, la lista de palabras se va desplazando de modo automático en busca de la palabra buscada. El botón **Back** sirve para regresar a la pantalla desde la que se ha llegado a la actual y con el botón **Print** se puede imprimir el contenido de la ayuda.

Continuará.....

Nota de Radacción: El lector puede descargar este capítulo y capítulos anteriores del curso desde la sección **“Soporte Técnico”** en el sitio web de **EduDevices** (www.edudevices.com.ar)