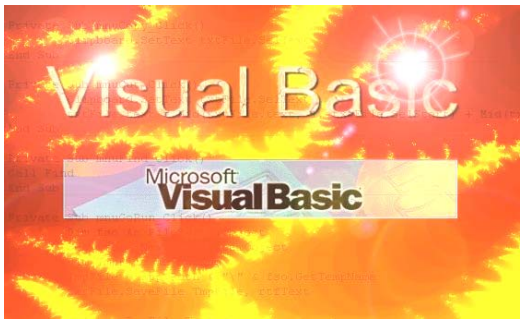
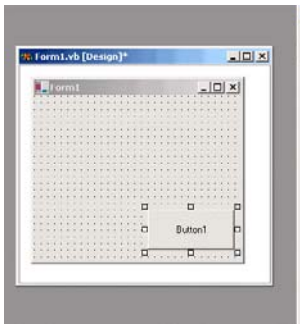
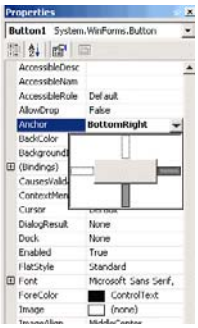
CURSO

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

N. de R.: Este curso completo de Visual Basic 6.0 pretende introducir al lector en el uso de uno de los lenguajes de programación para el mundo "PC" que hoy en día gana cada vez más popularidad entre los programadores por su facilidad y flexibilidad de implemetación. ¡¡Esperamos que sea de vuestro agrado!!

Temario

1. Introducción

1.1 Programas secuenciales, interactivos y orientados a eventos
1.2 Programas para el entorno Windows
1.2.1 Modo de Diseño y Modo de Ejecución
1.2.2 Formularios y Controles
1.2.3 Objetos y Propiedades
1.2.4 Nombres de objetos
1.2.5 Eventos
1.2.6 Métodos
1.2.7 Proyectos y ficheros
1.3 El entorno de programación Visual Basic 6.0
1.4 El Help de Visual Basic 6.0
1.5 Ejemplos
1.5.1 Ejemplo 1.1: Sencillo programa de colores y posiciones
1.5.2 Ejemplo 1.2: Minicalculadora elemental
1.5.3 Ejemplo 1.3: Transformación de unidades de temperatura
1.5.4 Ejemplo 1.4: Colores RGB

2. Entorno de programación Visual Basic 6.0

2.1 Introducción : ¿Qué es Visual Basic 6.0?
2.2 El entorno de Visual Basic 6.0
2.2.1 La barra de menús y las barras de herramientas
2.2.2 Las herramientas (toolbox)
2.3 Formularios (*forms*) y módulos
2.4 La ventana de proyecto (*project*)
2.5 La ventana de propiedades (*Properties*)
2.6 Creación de programas ejecutables
2.7 Cómo utilizar el Help
2.8 Utilización del Code Editor
2.9 Utilización del Debugger
2.9.1 Ejecución controlada de un programa
2.9.2 Ventanas Immediate, Locals y Watches
2.9.3 Otras posibilidades del Debugger

3. Lenguaje BASIC

3.1 Introducción
3.2 Comentarios y otras utilidades en la programación con visual basic
3.3 Proyectos y módulos
3.3.1 Ámbito de las variables y los procedimientos
3.3.1.1 Variables y funciones de ámbito local
3.3.1.2 Variables y funciones de ámbito global
3.4 Variables
3.4.1 Identificadores
3.4.2 Variables y constantes
3.4.3 Nombres de variables
3.4.4 Tipos de datos
3.4.5 Elección del tipo de una variable
3.4.6 Declaración explícita de variables
3.5 Operadores
3.6 Sentencias de control
3.6.1 Sentencia IF ... THEN ... ELSE ...
3.6.2 Sentencia SELECT CASE
3.6.3 Sentencia FOR ... NEXT
3.6.4 Sentencia DO ... LOOP
3.6.5 Sentencia WHILE … WEND
3.6.6 Sentencia FOR EACH ... NEXT
3.7 Algoritmos
3.7.1 Introducción
3.7.2 Representación de algoritmos

3.8 Funciones y Procedimientos
3.8.1 Conceptos generales sobre funciones
3.8.2 Funciones y procedimientos Sub en Visual Basic 6.0
3.8.3 Funciones (function)
3.8.4 Procedimientos Sub
3.8.5 Argumentos por referencia y por valor
3.8.6 Procedimientos recursivos
3.8.7 Procedimientos con argumentos opcionales
3.8.8 Número indeterminado de argumentos
3.8.9 Utilización de argumentos con nombre
3.9 Arrays
3.9.1 Arrays estáticos
3.9.2 Arrays dinámicos
3.10 Estructuras: Sentencia Type
3.11 Funciones para manejo de cadenas de caracteres
3.12 Funciones matemáticas

4. Eventos, Propiedades y Controles

4.1 Eventos
4.1.1 Eventos generales
4.1.1.1 Carga y descarga de formularios
4.1.1.2 Paint
4.1.1.3 El foco (focus)
4.1.1.4 KeyPress, KeyUp y KeyDown
4.1.2 Eventos relacionados con el ratón
4.1.2.1 Click y DblClick
4.1.2.2 MouseDown, MouseUp y MouseMove
4.1.2.3 DragOver y DragDrop
4.2 Propiedades más comunes
4.3 Controles más usuales
4.3.1 Botón de comando (Command Button)
4.3.2 Botones de opción (Option Button)
4.3.3 Botones de selección (Check Box)
4.3.4 Barras de desplazamiento (Scroll Bars)
4.3.5 Etiquetas (Labels)
4.3.6 Cajas de texto (Text Box)
4.3.7 Listas (List Box)
4.3.8 Cajas combinadas (Combo Box)
4.3.9 Controles relacionados con ficheros
4.3.10 Control Timer
4.4 Cajas de diálogo estándar (Controles Common Dialog)
4.4.1 Open/Save Dialog Control
4.4.2 Print Dialog Control
4.4.3 Font Dialog Control
4.4.4 Color Dialog Control
4.5 Formularios múltiples

4.5.1 Formularios y sub-formularios
4.6 Arrays de controles

5. Menús

5.1 Introducción a las posibilidades de los menús
5.2 El editor de menús (Menu Editor)
5.3 Añadir código a los menús
5.4 Arrays de menús
5.5 Ejemplo: Menú para determinar las características de un texto
5.6 Menús contextuales (Popup Menu)

6. Gráficos en Visual Basic 6.0

6.1 Tratamiento del color
6.1.1 Representación hexadecimal de los colores
6.1.2 Acceso a los colores del sistema
6.1.3 Función RGB
6.1.4 Paleta de colores
6.2 Formatos gráficos
6.3 Controles gráficos
Control line
Control shape
6.3.3 Ejemplo 6.1: Uso de los controles line y shape
Control image
6.3.5 Control Picture Box
6.4 Métodos gráficos
6.4.1 Método print
6.4.2 Dibujo de puntos: método Pset
6.4.3 Dibujo de líneas y rectángulos: método line
6.4.4 Dibujo de circunferencias, arcos y elipses: método circle
6.4.5 Otros métodos gráficos
6.5 Sistemas de coordenadas
6.5.1 Método Scale
6.6 Eventos y propiedades relacionadas con gráficos
6.6.1 El evento Paint
6.6.2 La propiedad DrawMode
6.6.3 Planos de dibujo (Layers)
6.6.4 La propiedad AutoRedraw
6.6.5 La propiedad ClipControl
6.7 Ejemplos
6.7.1 Ejemplo 6.1: Gráficos y barras de desplazamiento
6.7.2 Ejemplo 6.2: Representación gráfica de la solución de la ecuación de segundo grado
6.8 Barras de Herramientas (Toolbars)

7. Archivos y Entrada/Salida de Datos

7.1 Cajas de diálogo InputBox y MsgBox
7.2 Método Print
7.2.1 Características generales
7.2.2 Función Format
7.3 Utilización de impresoras
7.3.1 Método PrintForm
7.3.2 Objeto Printer
7.4 Controles FileList, DirList y DriveList
7.5 Tipos de ficheros
7.6 Lectura y escritura en ficheros secuenciales
7.6.1 Apertura y cierre de ficheros
7.6.2 Lectura y escritura de datos
7.6.2.1 Sentencia Input
7.6.2.2 Función Line Input y función Input
7.6.2.3 Función print #
7.6.2.4 Función write #
7.7 Ficheros de acceso aleatorio
7.7.1 Abrir y cerrar archivos de acceso aleatorio
7.7.2 Leer y escribir en una archivo de acceso aleatorio. Funciones Get y Put
7.8 Ficheros de acceso binario

8. ANEXO A: Consideraciones adicionales sobre datos y variables

8.1 Caracteres y código ASCII
8.2 Números enteros
8.3 Números reales
8.3.1 Variables tipo Single
8.3.2 Variables tipo Double
8.4 Sistema binario, octal, decimal y hexadecimal

1. INTRODUCCIÓN

Visual Basic 6.0 es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores de PCs, tanto expertos como novatos. En el caso de los programadores expertos por la facilidad con la que desarrollan aplicaciones complejas en poquísimo tiempo (comparado con lo que cuesta programar en ***Visual C++***, por ejemplo). En el caso de los programadores novatos por el hecho de ver de lo que son capaces a los pocos minutos de empezar su aprendizaje. El precio que hay que pagar por utilizar ***Visual Basic 6.0*** es una menor velocidad o eficiencia en las aplicaciones.

Visual Basic 6.0 es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

Visual Basic 6.0 es también un programa *basado en objetos*, aunque no *orientado a objetos* como ***Visual C++***. La diferencia está en que ***Visual Basic 6.0*** utiliza ***objetos*** con ***propiedades*** y ***métodos***, pero carece de los mecanismos de ***herencia*** y ***polimorfismo*** propios de los verdaderos lenguajes orientados a objetos como ***Java*** y *C++*.

En este primer capítulo se presentarán las características generales de ***Visual Basic 6.0***, junto con algunos ejemplos sencillos que den idea de la potencia del lenguaje y del modo en que se utiliza.

1.1 PROGRAMAS SECUENCIALES, INTERACTIVOS Y ORIENTADOS A EVENTOS

Existen distintos tipos de programas. En los primeros tiempos de los ordenadores los programas eran de tipo ***secuencial*** (también llamados tipo ***batch***) Un programa secuencial es un programa que se arranca, lee los datos que necesita, realiza los cálculos e imprime o guarda en el disco los resultados. De ordinario, mientras un programa secuencial está ejecutándose no necesita ninguna intervención del usuario. A este tipo de programas se les llama también *programas basados u orientados a procedimientos* o *a algoritmos* (*procedural languages*). Este tipo de programas siguen utilizándose ampliamente en la actualidad, pero la difusión de los PCs ha puesto de actualidad otros tipos de programación.

Los programas ***interactivos*** exigen la intervención del usuario en tiempo de ejecución, bien para suministrar datos, bien para indicar al programa lo que debe hacer por medio de menús. Los programas interactivos limitan y orientan la acción del usuario. Un ejemplo de programa interactivo podría ser ***Matlab***.

Por su parte los programas ***orientados a eventos*** son los programas típicos de ***Windows***, tales como ***Netscape***, ***Word***, ***Excel*** y ***PowerPoint***. Cuando uno de estos programas ha arrancado, lo único que hace es quedarse a la espera de las acciones del usuario, que en este caso son llamadas ***eventos***.
El usuario dice si quiere abrir y modificar un fichero existente, o bien comenzar a crear un fichero desde el principio. Estos programas pasan la mayor parte de su tiempo esperando las acciones del usuario (eventos) y respondiendo a ellas. Las acciones que el usuario puede realizar en un momento determinado son variadísimas, y exigen un tipo especial de programación: ***la programación orientada a eventos***. Este tipo de programación es sensiblemente más complicada que la secuencial y la interactiva, pero ***Visual Basic 6.0*** la hace especialmente sencilla y agradable.

1.2 PROGRAMAS PARA EL ENTORNO WINDOWS

Visual Basic 6.0 está orientado a la realización de programas para ***Windows***, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc.

Prácticamente todos los elementos de interacción con el usuario de los que dispone ***Windows 95 / 98 / XP, etc.*** pueden ser programados en ***Visual Basic 6.0*** de un modo extraordinariamente sencillo.

En ocasiones bastan unas pocas operaciones con el ratón y la introducción a través del teclado de algunas sentencias para disponer de aplicaciones con todas las características de ***Windows 95 / 98 / XP / Vista***. En los siguientes apartados se introducirán algunos conceptos de este tipo de programación.

1.2.1 Modo de Diseño y Modo de Ejecución

La aplicación ***Visual Basic*** de ***Microsoft*** puede trabajar de dos modos distintos: en modo de diseño y en modo de ejecución. En ***modo de diseño*** el usuario construye interactivamente la aplicación, colocando ***controles*** en el ***formulario***, definiendo sus ***propiedades***, desarrollando ***funciones*** para gestionar los ***eventos***.

La aplicación se prueba en ***modo de ejecución***. En ese caso el usuario actúa sobre el programa (introduce ***eventos***) y prueba cómo responde el programa. Hay algunas ***propiedades*** de los ***controles*** que deben establecerse en modo de diseño, pero muchas otras pueden cambiarse en tiempo de ejecución desde el programa escrito en ***Visual Basic 6.0***, en la forma en que más adelante se verá.
También hay ***propiedades*** que sólo pueden establecerse en modo de ejecución y que no son visibles en modo de diseño.
Todos estos conceptos –***controles***, ***propiedades***, ***eventos***, etc.- se explican en los apartados siguientes.

1.2.2 Formularios y Controles

Cada uno de los elementos gráficos que pueden formar parte de una aplicación típica de ***Windows 95 / 98 / XP / Vista*** es un tipo de ***control***: los botones, las cajas de diálogo y de texto, las cajas de selección desplegables, los botones de opción y de selección, las barras de desplazamiento horizontales y verticales, los gráficos, los menús, y muchos otros tipos de elementos son controles para ***Visual Basic 6.0***. Cada control debe tener un ***nombre*** a través del cual se puede hacer referencia a él en el programa. ***Visual Basic 6.0*** proporciona nombres *por defecto* que el usuario puede modificar. En el Apartado 1.2.4 se exponen algunas reglas para dar nombres a los distintos controles.

En la terminología de ***Visual Basic 6.0*** se llama ***formulario*** (*form*) a una ventana. Un formulario puede ser considerado como una especie de contenedor para los controles. Una aplicación puede tener varios formularios, pero un único formulario puede ser suficiente para las aplicaciones más sencillas. Los formularios deben también tener un nombre, que puede crearse siguiendo las mismas reglas que para los controles.

1.2.3 Objetos y Propiedades

Los formularios y los distintos tipos de controles son entidades genéricas de las que puede haber varios ejemplares concretos en cada programa. En *programación orientada a objetos* (más bien *basada en objetos*, habría que decir) se llama ***clase*** a estas entidades genéricas, mientras que se llama ***objeto*** a cada ejemplar de una clase determinada. Por ejemplo, en un programa puede haber varios botones, cada uno de los cuales es un ***objeto*** del tipo de control ***command button***, que sería la ***clase***.

Cada formulario y cada tipo de control tienen un conjunto de ***propiedades*** que definen su aspecto gráfico (tamaño, color, posición en la ventana, tipo y tamaño de letra, etc.) y su forma de responder a las acciones del usuario (si está activo o no, por ejemplo). Cada propiedad tiene un ***nombre*** que viene ya definido por el lenguaje.

Por lo general, las propiedades de un ***objeto*** son datos que tienen valores lógicos (*true*, *false*) o numéricos concretos, propios de ese objeto y distintos de las de otros objetos de su clase. Así pues, cada clase, tipo de objeto o control tiene su conjunto de propiedades, y cada objeto o control concreto tiene unos valores determinados para las propiedades de su clase.

Casi todas las propiedades de los objetos pueden establecerse en tiempo de diseño y también -casi siempre- en tiempo de ejecución. En este segundo caso se accede a sus valores por medio de las sentencias del programa, en forma análoga a como se accede a cualquier variable en un lenguaje de programación. Para ciertas propiedades ésta es la única forma de acceder a ellas. Por supuesto ***Visual Basic 6.0*** permite crear distintos tipos de variables, como más adelante se verá.

Se puede acceder a una propiedad de un objeto por medio del ***nombre del objeto*** a que pertenece, seguido de un punto y el ***nombre de la propiedad***, como por ejemplo ***optColor.objName***.

En el siguiente apartado se estudiarán las reglas para dar nombres a los objetos.

1.2.4 Nombres de objetos

En principio cada objeto de ***Visual Basic 6.0*** debe tener un nombre, por medio del cual se hace referencia a dicho objeto. El nombre puede ser el que el usuario desee, e incluso ***Visual Basic 6.0*** proporciona ***nombres por defecto*** para los diversos controles. Estos nombres por defecto hacen referencia al tipo de control y van seguidos de un número que se incrementa a medida que se van introduciendo más controles de ese tipo en el formulario (por ejemplo ***VScroll1***, para una barra de desplazamiento -*scroll bar*- vertical, ***HScroll1***, para una barra horizontal, etc.).

Los ***nombres por defecto no son adecuados*** porque hacen referencia al tipo de control, pero no al uso que de dicho control está haciendo el programador. Por ejemplo, si se utiliza una barra de desplazamiento para introducir una temperatura, conviene que su nombre haga referencia a la palabra ***temperatura***, y así cuando haya que utilizar ese nombre se sabrá exactamente a qué control corresponde. Un nombre adecuado sería por ejemplo ***hsbTemp***, donde las tres primeras letras indican que se trata de una *horizontal scroll bar*, y las restantes (empezando por una mayúscula) que servirá para definir una *temperatura*.

Existe una convención ampliamente aceptada que es la siguiente: *se utilizan siempre tres letras minúsculas que indican el tipo de control, seguidas por otras letras (la primera mayúscula, a modo de separación) libremente escogidas por el usuario, que tienen que hacer referencia al uso que se va a dar a ese control*. La Tabla 1.1 muestra las abreviaturas de los controles más usuales, junto con la nomenclatura inglesa de la que derivan. En este mismo capítulo se verán unos cuantos ejemplos de aplicación de estas reglas para construir nombres.

Abreviatura	Control	Abreviatura	Control
chk	check box	cbo	combo y drop-list box
cmd	command button	dir	dir list box
drv	drive list box	fil	file list box
frm	form	fra	frame
hsb	horizontal scroll bar	img	image
lbl	label	lin	line
lst	list	mnu	menu
opt	option button	pic	picture
shp	shape	txt	text edit box
tmr	timer	vsb	vertical scroll bar

Tabla 1.1. Abreviaturas para los controles más usuales.

1.2.5 Eventos

Ya se ha dicho que las acciones del usuario sobre el programa se llaman ***eventos***. Son eventos típicos el clicar sobre un botón, el hacer doble click sobre el nombre de un fichero para abrirlo, el arrastrar un icono, el pulsar una tecla o combinación de teclas, el elegir una opción de un menú, el escribir en una caja de texto, o simplemente mover el ratón. Más adelante se verán los distintos tipos de eventos reconocidos por ***Windows 95 / 98 / XP / Vista*** y por ***Visual Basic 6.0***.

Cada vez que se produce un evento sobre un determinado tipo de control, ***Visual Basic 6.0*** arranca una determinada ***función*** o ***procedimiento*** que realiza la acción programada por el usuario para ese evento concreto. Estos procedimientos se llaman con un nombre que se forma a partir del nombre del objeto y el nombre del evento, separados por el carácter (_), como por ejemplo ***txtBox_click***, que es el nombre del procedimiento que se ocupará de responder al evento ***click*** en el objeto ***txtBox***.

1.2.6 Métodos

Los ***métodos*** son funciones que también son llamadas desde programa, pero a diferencia de los procedimientos no son programadas por el usuario, sino que vienen ya pre-programadas con el lenguaje. Los métodos realizan tareas típicas, previsibles y comunes para todas las aplicaciones. De ahí que vengan con el lenguaje y que se libere al usuario de la tarea de programarlos. Cada tipo de objeto o de control tiene sus propios métodos.

Por ejemplo, los controles gráficos tienen un método llamado ***line*** que se encarga de dibujar líneas rectas. De la misma forma existe un método llamado ***circle*** que dibuja circunferencias y arcos de circunferencia Es obvio que el dibujar líneas rectas o circunferencias es una tarea común para todos los programadores y que ***Visual Basic 6.0*** da ya resuelta..

1.2.7 Proyectos y ficheros

Cada aplicación que se empieza a desarrollar en ***Visual Basic 6.0*** es un nuevo ***proyecto***. Un proyecto comprende otras componentes más sencillas, como por ejemplo los ***formularios*** (que son las ventanas de la interface de usuario de la nueva aplicación) y los ***módulos*** (que son conjuntos de funciones y procedimientos).

¿Cómo se guarda un proyecto en el disco? Un proyecto se compone siempre de ***varios ficheros*** (al menos de dos) y hay que preocuparse de guardar cada uno de ellos en el directorio adecuado y con el nombre adecuado. Existe siempre un fichero con extensión ****.vbp*** (*Visual Basic Project*) que se crea con el comando ***File/Save Project As***. El fichero del proyecto contiene toda la *información de conjunto*. Además hay que crear *un* ***fichero por cada formulario y por cada módulo*** que tenga el proyecto. Los ficheros de los formularios se crean con ***File/Save File As*** y tienen como extensión ****.frm***. Los ficheros de código o ***módulos*** se guardan también con el comando ***File/Save File As*** y tienen como extensión ****.bas*** si se trata de un módulo estándar o ****.cls*** si se trata de un módulo de clase (***class module***).

Clicando en el botón ***Save*** en la barra de herramientas se actualizan todos los ficheros del proyecto. Si no se habían guardado todavía en el disco, ***Visual Basic 6.0*** abre cajas de diálogo ***Save As*** por cada uno de los ficheros que hay que guardar.

Continuará.....

WWW.EDUDEVICES.COM.AR