

CAPÍTULO IV: INTRODUCCIÓN A VISUAL-BASIC

4.1 ANTECEDENTES HISTÓRICOS

El primer antecedente de VISUAL-BASIC es la creación del lenguaje BASIC (Beginner's All Purpose Symbolic Instruction Code) en su versión original en el Dartmouth College en el año de 1964. La intención original era crear un lenguaje de programación sencillo muy parecido al lenguaje humano natural en inglés, con instrucciones y procedimientos poco complejos que permitieran al usuario con poca o nula experiencia en programación crear sus propias aplicaciones, al mismo tiempo que se iniciaba en un lenguaje de programación. A pesar de ser un lenguaje sencillo, éste era suficiente en ese entonces para llevar a cabo todas las tareas necesarias para ejecutar aplicaciones.

En un principio, el lenguaje BASIC no evolucionó mucho, ya que fue en los setentas cuando otros lenguajes de programación como FORTRAN y COBOL tenían la mayor atención y las mayores y mejores aplicaciones que el primitivo BASIC de esos años. Para 1978 existían varias versiones del BASIC, por lo que fue necesario unificar todas las variantes de este lenguaje bajo una misma norma, la cual se llamó BASIC STANDARD.

La sencillez del lenguaje hizo que se ganara el desprecio de los programadores profesionales [4.5], ya que lo consideraban un lenguaje para inexpertos y principiantes, fue así que tomaría un poco más de tiempo para que su uso se aceptara y generalizara.

En la primera mitad de la década de los ochenta, gracias al comienzo de la producción en serie de las computadoras personales destinadas al público en general, fue que el BASIC tuvo su resurgimiento como lenguaje para principiantes, que era para lo que originalmente había sido concebido, ya que muchas de estas computadoras personales lo empleaban como sistema operativo.

Conforme la PC se hizo más popular fueron surgiendo nuevas versiones del BASIC, primero fue BASICA, luego GW-BASIC, sin embargo nunca llegaron a ser de importancia ya que presentaban numerosas desventajas con respecto a otros lenguajes de la época como el PASCAL, el C y el CLIPPER, por lo que los programadores profesionales nunca lo utilizaron formalmente, de hecho, esas versiones del BASIC acabaron desvirtuando y desacreditando aún más el lenguaje.

Las razones para ello eran obvias [4.4]:

- No era un lenguaje estructurado.
- No existían herramientas de compilación fiables.
- No disponía de herramientas de intercambio de información.
- No tenía librerías.
- No se podía acceder al interior de la máquina.

En fin, se trataba de una lista larga de desventajas respecto a otros lenguajes de programación.

El abandono que se le dio al lenguaje ocasionó incluso que, cuando Microsoft lanzara una versión muy bien corregida del mismo, llamada Quick-BASIC, ésta pasara sin pena ni gloria, prácticamente ignorada, de no ser porque las últimas versiones del MS-DOS incluían el Q-BASIC que era una versión recortada del Quick-BASIC, como un fichero ejecutable que venía con el sistema operativo y del cual, de hecho, se aprovechaba el editor de texto [4.4]. Además, como en ese entonces las primeras versiones de Windows ya superaban las prestaciones del MS-DOS, de nuevo el Quick-BASIC no tenía la oportunidad de darse mucho a conocer, a pesar de que éste era ya un lenguaje estructurado que, de hecho, permitía ya la creación de programas de alto nivel.

Fue por lo que en un principio fue despreciado, lo que finalmente hizo que se volviera tan popular: su sencillez; aparte de que una vez que se le agregó un entorno gráfico, se hizo posible el aprovechamiento total de los recursos de Windows, resultando así un nuevo producto que actualmente se conoce como VISUAL-BASIC, el cual difiere actualmente mucho del BASIC original y del cual, en realidad, lo único que conserva es su simplicidad. Por el contrario, se trata ahora de un poderoso lenguaje de programación visual que permite la creación de robustos programas de 32 bits en un entorno gráfico que es Windows y que se ha ganado la amplia aceptación de programadores de todos los niveles por las innumerables ventajas con las que cuenta y las que se le han ido encontrando y que han hecho que, hoy en día, Visual Basic sea el lenguaje de programación más popular del mundo [4.1], así como la principal herramienta para el desarrollo de aplicaciones de Windows. Es también gracias al VISUAL-BASIC que Microsoft ha podido expandir su monopolio, ya que este lenguaje es únicamente compatible con Windows.

La última versión que ha salido al mercado de este lenguaje es la 6.0, la cual es, por mucho, la mejor que ha sido lanzada; sin embargo, por su naturaleza, aún resulta insuficiente para llegar al fondo de la máquina y controlar uno a uno sus registros, pero la programación de bajo nivel no es en realidad uno de los propósitos de VISUAL-BASIC; de hecho, las herramientas aportadas por él son más que suficientes para crear programas de manera sencilla y altas prestaciones.

4.2 CARACTERÍSTICAS GENERALES

Las aplicaciones creadas con VISUAL-BASIC (VB) se desarrollan partiendo de una interfaz gráfica con la cual es posible generar, de manera automática, conectividad entre controles y datos de manera fácil y rápida mediante la acción de arrastrar y colocar objetos

sobre formularios. Los programas quedan compuestos, en una parte, por códigos y, en otra, por partes asociadas a los objetos presentes en la interfaz gráfica. Esto hace que el tipo de programación propio de VB sea parcialmente estructurado y parcialmente orientado a objetos.

VISUAL-BASIC es un ambiente de programación integrado, esto quiere decir que se utiliza la misma herramienta para editar, compilar, correr, etc. una aplicación. Los procesos se componen de formularios y controles, los cuales tienen propiedades y están diseñados para llevar a cabo procedimientos cuando ocurre algún evento sobre ellos. Además el programa permite usar procedimientos reutilizables que pueden haber sido creados por otras personas o por uno mismo así como módulos de clase para programación orientada a objetos.

VB puede crear componentes COM (Component Object Module), los cuales constituyen la estructura que utiliza Microsoft para todos sus lenguajes actualmente. Para desarrollar aplicaciones se utilizan componentes y objetos, parte de esos componentes son los controles ActiveX, servidores Active (DLL o EXE), documentos ActiveX, etc.

Cuenta con características para el desarrollo de Internet e Intranet como lo son las aplicaciones DHTML, aplicaciones IIS y soporte para el FTP.

La creación de un programa bajo VISUAL-BASIC requiere de los siguientes pasos:

- La creación de una interfaz de usuario. Esta interfaz constituirá la vía de comunicación con la máquina para la entrada y salida de datos. Se parte de una ventana o formulario a la que se le van añadiendo los controles pertinentes.
- La definición de las propiedades de los controles u objetos que hayan sido colocados en el formulario. Estas propiedades determinan la forma estática de los controles, es decir, su forma y sus funciones.

- La generación de un código asociado a los eventos que ocurran sobre los objetos. A la respuesta a un evento (clic, doble clic, una tecla pulsada, etc.) se le llama procedimiento y deberá desarrollarse de acuerdo al propósito del programa.
- La generación del código del programa. Un programa puede elaborarse simplemente con la programación de los procedimientos que deban acompañar a cada evento sobre un objeto; sin embargo, VB ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en módulos, funciones y procedimientos. Estos procedimientos no responden a un evento acaecido a un objeto, sino que responden a un evento producido durante la ejecución del programa.

VB es extensible gracias a las llamadas al API de Windows, al soporte de terceras partes y a la integración con otras aplicaciones de Windows.

Es utilizado principalmente para aplicaciones de gestión de empresas [4.3], debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, dado que cuenta con varias bibliotecas para el manejo y acceso a bases de datos, entre las que destaca ADO, Jet, ODBC y RDO; además de que existe un gran número de programadores en este lenguaje.

Finalmente, las numerosas bibliotecas DLL facilitan el acceso a muchas funciones del sistema operativo y la integración con otras aplicaciones.

VISUAL-BASIC tiene también sus lenguajes derivados como lo son el VBScript y el Visual-Basic for Applications VBA, el primero es el lenguaje predeterminado para Active Server Pages ASP [4.3] y el segundo es más bien una extensión del mismo lenguaje, el cual permite codificar módulos o macros que se utilizan en los programas de Microsoft Office.

En la actualidad existe una nueva variante del lenguaje que se llama Visual-Basic.NET, el cual es un programa que cuenta ya con funciones equivalentes a C++ y con una orientación a objetos como la de JAVA.

4.3 VERSIONES

Las versiones de VISUAL-BASIC para Windows son muy conocidas, la primera versión: Microsoft VISUAL-BASIC 1.0 para MS-DOS, poco difundida, data de 1991. Era un entorno que, aunque en modo texto, incluía un diseñador de formularios en el que se podían arrastrar y soltar distintos controles. La versión 2.0 salió al mercado en octubre de 1992.

La versión 3.0, que fue la última de 16 bits, fue lanzada en mayo de 1993 e incluía ya una detallada biblioteca de componentes para toda clase de usos. Durante la transición de Windows 3.11 a Windows 95, apareció, en septiembre de 1995, la versión 4.0, la cual podía generar programas de 16 y 32 bits a partir de un mismo código fuente, aunque a costa de un gran aumento en el tamaño de los archivos "runtime" necesarios. Además, se sustituyeron los controles VBX por los nuevos OCX y, por primera vez, se podía compilar a código nativo (aunque no se obtenían grandes mejoras de rendimiento al hacerlo), y no a código intermedio. La versión 5.0 apareció en marzo de 1997 y fue la primera que agregara orientación a objetos al lenguaje. La última versión es la 6.0 y data de agosto de 1998.

4.4 VENTAJAS Y DESVENTAJAS

Ventajas

- VISUAL-BASIC es un lenguaje simple y, por tanto, más fácil de aprender que otros lenguajes más complejos como DELPHI y Power Builder, entre otros.

- Su mayor simplicidad radica en el dibujado de formularios, mediante el arrastre de controles.
- La sintaxis es similar al lenguaje natural humano.
- Es un lenguaje compatible con Microsoft Office y muchas herramientas de 3ª parte.
- Las ramas de Visual Basic X han muerto. Microsoft ha comenzado con su versión .NET, completamente diferente del VISUAL-BASIC original. Visual-Basic.NET compila hacia el mismo código final que C++ y es mucho más potente y alberga muchas más funcionalidades que su versión anterior.
- Es un lenguaje RAD, centrado en conseguir en el menor tiempo posible los resultados deseados.
- Tiene una ligera implementación de la POO.
- Permite el tratamiento de mensajes de Windows.
- Gran parte del trabajo en el diseño de formularios está realizado, gracias a la gran gama de controles incorporados junto al lenguaje, cuyas propiedades y métodos son fáciles de manipular, lo cual, junto con todas las demás características del programa, hace que el desarrollo de aplicaciones se vuelva sumamente rápido.
- Permite crear controles personalizados fácilmente del mismo modo que el diseño de los formularios.
- Puede crear controles ActiveX más fácilmente que si se usa C++.
- Cuenta con un excelente paquete de ayuda.
- Cuenta con herramientas para el desarrollo e integración de Internet/Intranet.
- Se pueden crear servidores fácilmente.
- Es excelente para cálculos intensivos del CPU como por ejemplo operaciones matemáticas.

Desventajas

- El dueño de VB es Microsoft, por lo tanto nadie que no sea del equipo de desarrollo de esta compañía decide la evolución del lenguaje.
- Sólo existe un compilador, llamado igual que el lenguaje.
- Sólo genera ejecutables para Windows.
- La sintaxis es bastante inflexible.
- Los ejecutables generados son relativamente lentos.
- No es adecuado para aplicaciones grandes, multimedia, de oficina, videojuegos, editores gráficos, etc.
- No cuenta con características para programación avanzada.
- No permite generar librerías dinámicas (DLL).
- Sólo permite el uso de funciones de librerías dinámicas (DLL) stdcall.
- Para que los ejecutables que genera funcionen, necesita una DLL llamada MSVBVMxy.DLL: MicroSoft Visual Basic Virtual Machine x.y.
- Algunas funcionalidades están indocumentadas.
- La ligera implementación de la POO no permite sacar el máximo provecho de este modelo de programación.
- No soporta el tratamiento de procesos como parte del lenguaje.
- No maneja excepciones.
- No incluye operadores a nivel de bits.
- No contempla el manejo de memoria dinámica, punteros, arrays, etc. como parte del lenguaje.
- No puede avisar ni advertir cuando están presentes ciertos errores, como sería una inadecuada conversión de tipos.

- El tratamiento de mensajes de Windows es básico e indirecto.
- La gran gama de controles incorporados son, sin embargo, muy generales, lo cual lleva a tener que reprogramar nuevos controles para una necesidad particular de la aplicación.
- Los controles personalizados no mejoran la potencia de la API de Windows y, en algunos casos, acudir a ésta es la única manera de conseguir el control personalizado deseado.
- No tiene la misma funcionalidad que C++ a la hora de obtener características de bajo nivel del sistema operativo.
- La forma de programación que plantea VISUAL-BASIC ha ocasionado que muchos programadores de este lenguaje practiquen malos hábitos, entre los que se encuentran [4.3]:

- Empezar arreglos o *Arrays* en 1.
- La utilización de variables globales.
- Omitir la declaración de variables.
- Trabajar con variables de tipo indefinido (Tipo Variant).
- Escribir código innecesario.
- Escribir código repetido a falta de funciones y procedimientos.
- El uso incorrecto de la API de Windows.
- El uso de goto y etiquetas.
- El uso de controles como simples contenedores de datos.
- El crear dependencia de los controles a la hora de programar.

Finalmente, y a pesar de la larga lista de desventajas presentada, VISUAL-BASIC resulta la herramienta óptima para la realización del programa que constituye esta tesis, ya que para dicho fin, las numerosas ventajas que otorga con respecto a otros lenguajes de

programación, son prácticamente inmejorables para obtener una aplicación vistosa, fácil de utilizar y de entender para cualquier usuario y a la vez potente para los cálculos matemáticos que habrá de realizar para la resolución de los problemas que le serán presentados. Todo lo anterior son objetivos del paquete que ha de crearse en la elaboración de este proyecto, además de que, para un estudiante de ingeniería civil que cuenta con conocimientos básicos de programación, la sencillez y lógica del lenguaje, ayuda a reducir enormemente la curva de aprendizaje del mismo y, por lo tanto, uno puede empezar a programar de forma inmediata prácticamente, aunque no se cuente con conocimientos previos del funcionamiento y sintaxis del lenguaje ni del programa, lo cual reduce, eventualmente, el tiempo en el que uno puede tener una aplicación lista con las mismas características de cualquier programa de Windows con las que todo mundo se encuentra familiarizado.

REFERENCIAS SELECCIONADAS

- 1.1 Ceballos, Fco. Javier. *Curso de programación de Visual Basic 6*. (2000). Alfaomega, México, D.F.
- 1.2 Ceballos, Fco. Javier. *Enciclopedia de Microsoft Visual Basic 6*. (2000). Alfaomega, México, D.F.
- 1.3 http://es.wikipedia.org/wiki/Visual_Basic
- 1.4 <http://www.monografias.com/trabajos2/guiavb/guiavb.shtml>
- 1.5 <http://www.monografias.com/trabajos10/visual/visual.shtml>