

Universidad de Los Andes

Facultad de Ciencia Económicas y Sociales

Escuela de Estadística

## **PRÁCTICAS DE COMPUTACIÓN II CON VISUAL BASIC.**

Autor: Angel Alberto Zambrano

Presentado como trabajo de ascenso para optar a la categoría de Agregado

Febrero-2008

**UNIVERSIDAD DE LOS ANDES**  
**FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES**  
**ESCUELA DE ESTADÍSTICA**  
**MÉRIDA-VENEZUELA**

**ACTA-VEREDICTO**

Nosotros, miembros del Jurado designado por el Ilustre Consejo de la Facultad de Ciencias Económicas y Sociales en su sesión N°07 de fecha 12-03-2008, para conocer y decidir acerca de la Credencial de Mérito para el Ascenso del Profesor: **ÁNGEL A. ZAMBRANO**, Cédula de Identidad N°. **V- 8.027.708** a la Categoría de Profesora **AGREGADO**, conforme al Artículo N°. 164 numeral 2 del Estatuto de Personal Docente y de Investigación de la Universidad de los Andes, reunidos en la sede del Instituto de Investigaciones Económicas y Sociales (I.I.E.S.), el día 17 de abril de 2008 a las 10:00 a.m., una vez revisado, discutido y realizada la correspondiente exposición pública del trabajo titulado: **"PRÁCTICAS DE COMPUTACIÓN II CON VISUAL BASIC"**, y por cuanto consideramos que éste cumple con todos los requisitos exigidos para un Trabajo de Ascenso, acordamos impartir su aprobación para los fines a los cuales ha sido propuesto.

En Mérida a los diecisiete (17) días del mes de abril de dos mil ocho (2008).



Prof. Héctor Mata Brito  
Coordinador del Jurado





Prof. Gerardo Colmenares  
Miembro del jurado



Prof. Israel Ramírez  
Miembro del Jurado



Anexo: Ejemplar de trabajo en referencia.

## Introducción.

El presente trabajo pretende servir de ayuda a los estudiantes de Computación II en la Licenciatura de Estadística, que se dicta en la Facultad de Ciencias Económicas y Sociales de la Universidad de Los Andes.

Dicha asignatura se dicta en dos sesiones semanales de dos horas c/u: una sesión en aula donde se desarrolla la parte teórica-conceptual de la programación, así como también el desarrollo de algoritmos para la solución de problemas planteados, y la otra en el laboratorio de computación, donde se implementan los algoritmos en un lenguaje de programación de uso general. Durante los últimos 8 semestres este lenguaje ha sido el Visual Basic, al principio en su versión 5.0 y ahora en la versión 6.0

Durante el semestre se realizan entre 10 a 12 prácticas, que le dan al estudiante las habilidades necesarias para implementar un proyecto que deben entregar al final del curso.

El objetivo principal de este trabajo es diseñar e implementar un conjunto de prácticas dirigidas que cubren el programa de la Asignatura Computación II. El mismo se divide en doce (12) prácticas y un anexo, en el que se desarrolla un pequeño manual de usuario del Visual Basic 6.0.

Cada práctica esta diseñada de forma tal que el estudiante, siguiendo paso a paso la misma, sea capaz de desarrollar una pequeña aplicación correspondiente a un tópico de la asignatura o un objeto del lenguaje Visual Basic.

El formato de las prácticas es el siguiente: primero se presenta los *objetivos* que se pretenden alcanzar, entre estos está el desarrollo de una aplicación; luego se presenta una sección denominada “*Antes de comenzar*” donde se desarrolla el Análisis Entrada-Proceso-Salida (EPS) de la aplicación, en el cual se presentan los requerimientos de entrada, proceso y salida necesarios para la implementación de la aplicación, y por último la sección denominada “*Actividades a realizar*”, donde está descrito los pasos que el estudiante deben seguir para la realización de la práctica.

Estas prácticas van desde el uso del ambiente de desarrollo del Visual Basic, la descripción de los diferentes controles de programación del Visual Basic, pasando por las estructuras de programación, hasta llegar al uso de archivos como medio de almacenamiento masivo.

# Índice:

<b>INTRODUCCIÓN</b> .....	<b>I</b>
<b>ÍNDICE</b> : .....	<b>III</b>
<b>ÍNDICE DE FIGURAS</b> :.....	<b>VI</b>
<b>ÍNDICE DE TABLAS</b> :.....	<b>VIII</b>
<b>GUÍA DE USUARIO</b> : .....	<b>X</b>
<b>PRÁCTICA 1. CONOCIENDO EL VISUAL BASIC 6.0</b> .....	<b>1</b>
OBJETIVOS: .....	1
ACTIVIDADES A REALIZAR.....	1
<b>PRÁCTICA 2. CONTROLES BÁSICOS</b> .....	<b>6</b>
OBJETIVOS: .....	6
ANTES DE COMENZAR .....	6
ACTIVIDADES A REALIZAR.....	7
<b>PRÁCTICA 3. ESTRUCTURAS DE DECISIÓN</b> .....	<b>13</b>
OBJETIVOS: .....	13
ANTES DE COMENZAR.....	13
ACTIVIDADES A REALIZAR.....	14
<b>PRÁCTICA 4. MÁS CONTROLES. VALIDACIÓN DE DATOS I</b> .....	<b>21</b>
OBJETIVOS: .....	21
ANTES DE COMENZAR.....	22
ACTIVIDADES A REALIZAR.....	23
<b>PRÁCTICA 5. ESTRUCTURAS DE REPETICIÓN I</b> .....	<b>32</b>
OBJETIVOS: .....	32
ANTES DE COMENZAR.....	32
ACTIVIDADES A REALIZAR.....	33
<b>PRÁCTICA 6. VALIDACIÓN DE DATOS II. MÁS CONTROLES</b> .....	<b>39</b>
OBJETIVOS: .....	39
ANTES DE COMENZAR.....	39
ACTIVIDADES A REALIZAR.....	40
<b>PRÁCTICA 7. PROCESANDO UNA ENCUESTA</b> .....	<b>50</b>
OBJETIVOS: .....	50
ANTES DE COMENZAR.....	51

ACTIVIDADES A REALIZAR .....	52
<b>PRÁCTICA 8. USANDO VARIOS FORMULARIOS.....</b>	<b>64</b>
OBJETIVOS: .....	64
ACTIVIDADES A REALIZAR:.....	65
<b>PRACTICA 9. USO DEL CONTROL MSFLEXGRID .....</b>	<b>77</b>
OBJETIVOS: .....	77
ANTES DE COMENZAR .....	77
ACTIVIDADES A REALIZAR:.....	79
<b>PRACTICA 10. ORDENAMIENTO Y BÚSQUEDA.....</b>	<b>91</b>
OBJETIVOS .....	91
ANTES DE COMENZAR .....	91
ACTIVIDADES A REALIZAR:.....	92
<b>PRACTICA 11. MANEJO DE ARCHIVOS DE TEXTO.....</b>	<b>103</b>
OBJETIVOS .....	103
ANTES DE COMENZAR .....	103
ACTIVIDADES A REALIZAR:.....	104
<b>PRACTICA 12. MANEJO DE ARCHIVOS DE TIPO (REGISTROS).....</b>	<b>117</b>
OBJETIVOS .....	117
ANTES DE COMENZAR .....	117
ACTIVIDADES A REALIZAR:.....	118
<b>ANEXO: INTRODUCCIÓN AL VISUAL BASIC .....</b>	<b>124</b>
EL LENGUAJE BASIC .....	125
PROGRAMACIÓN ORIENTADA A OBJETOS .....	127
ELEMENTOS BÁSICOS DE LA POO .....	127
CARACTERÍSTICAS DE LA POO .....	129
LOS OBJETOS DE VISUAL BASIC .....	130
CONOCIENDO EL VISUAL BASIC 6.0 .....	130
UN PROYECTO EN VISUAL BASIC. ....	141
DISEÑO DE UN PROGRAMA EN VISUAL BASIC.....	143
EL FORMULARIO.....	144
CONTROLES BÁSICOS.....	145
EL LENGUAJE .....	148
BIBLIOGRAFÍA: .....	162



## Índice de figuras:

FIGURA 1-1. FORMULARIO CON LAS DOS ETIQUETAS Y UN BOTÓN DE COMANDO .....	3
FIGURA 1-2. VENTANA DE CÓDIGO .....	4
FIGURA 2-1. FORMULARIO DE LA APLICACIÓN CALCULAR NOTA PROMEDIO.....	9
FIGURA 2-2. RESULTADO PARA LOS DATOS DE LA PRÁCTICA .....	12
FIGURA 3-1. FORMULARIO DE LA APLICACIÓN SUELDO .....	15
FIGURA 3-2. RESULTADOS PARA LOS DATOS PLANTEADOS.....	20
FIGURA 4-1. FORMULARIO DE LA APLICACIÓN “SERVICIOS” .....	26
FIGURA 4-2. RESULTADOS PARA LOS DATOS DE LA PRÁCTICA .....	31
FIGURA 5-1. FORMULARIO DE LA APLICACIÓN BINOMIAL .....	35
FIGURA 6-1. FORMULARIO DE LA APLICACIÓN FIGURAS .....	41
FIGURA 7-1. FORMULARIO DE LA APLICACIÓN “ENCUESTA ESTUDIANTIL FACES” .....	53
FIGURA 7-2. INSERTANDO UN MÓDULO AL PROYECTO .....	56
FIGURA 8-1. FORMULARIO DE LA APLICACIÓN “ENCUESTA ELECCIONES” .....	66
FIGURA 8-2. INSERTAR EL SEGUNDO FORMULARIO A LA APLICACIÓN .....	69
FIGURA 8-3. FORMULARIO 2 DE LA APLICACIÓN .....	70
FIGURA 9-1. FORMULARIO 1 DE LA APLICACIÓN “ENCUESTA”. .....	80
FIGURA 9-2. FORMULARIO 2 DE LA APLICACIÓN “ENCUESTA”. .....	86
FIGURA 10-1. FORMULARIO 1 DE LA APLICACIÓN “AGENDA TELEFÓNICA”. .....	93
FIGURA 10-2. FORMULARIO 2 DE LA APLICACIÓN “AGENDA TELEFÓNICA”. .....	100
FIGURA 11-1. FORMULARIO 1 DE LA APLICACIÓN “ARCHIVO DE TEXTO”. .....	105
FIGURA 11-2. VENTANA EDITOR DE MENÚS. ....	107
FIGURA 11-3. MENÚ CREADO CON LA OPCIÓN ARCHIVO ABIERTA .....	109
FIGURA 11-4. FORMULARIO 2 - RESULTADOS .....	117
FIGURA 11-5. CUADRO DE DIÁLOGO AGREGAR FORMULARIO .....	118
FIGURA 11-6. FORMULARIO ACERCA DE MODIFICADO PARA LA PRÁCTICA 11.....	118
FIGURA 12-1. FORMULARIO 1 DE LA APLICACIÓN “BIBLIOTECA”. .....	119
FIGURA A-1. VENTANA NUEVO PROYECTO. ....	131
FIGURA A-2. ENTORNO INTEGRADO DE DESARROLLO DE MICROSOFT VISUAL BASIC 6.0 .....	132
FIGURA A-3. LA BARRA DE HERRAMIENTAS ESTÁNDAR .....	133
FIGURA A-4. CAJA DE HERRAMIENTAS .....	135
FIGURA A-5. VENTANA FORMULARIO. ....	137
FIGURA A-6. VENTANA EXPLORADOR DE PROYECTO. ....	138



<b>FIGURA A-7. VENTANA DE PROPIEDADES. ....</b>	<b>138</b>
<b>FIGURA A-8. VENTANA POSICIÓN DEL FORMULARIO. ....</b>	<b>139</b>
<b>FIGURA A-9. VENTANA EDITOR DE CÓDIGO. ....</b>	<b>140</b>
<b>FIGURA A-10. VENTANA OPCIONES. ....</b>	<b>150</b>

## Índice de tablas:

TABLA 1-1. PROPIEDADES DEL FORMULARIO DE LA PRÁCTICA 1.....	2
TABLA 1-2. PROPIEDADES DE LA ETIQUETA 1 DEL FORMULARIO. ....	3
TABLA 2-1. ANÁLISIS EPS DE LA APLICACIÓN 1.....	6
TABLA 2-2. PROPIEDADES DEL FORMULARIO DE LA PRÁCTICA 2.....	7
TABLA 2-3. PROPIEDADES DE LO CONTROLES DEL FORMULARIO DE LA PRÁCTICA 2.....	9
TABLA 3-1. ANÁLISIS EPS DE LA APLICACIÓN SUELDO .....	13
TABLA 3-2. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN SUELDO .....	15
TABLA 3-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO DE LA APLICACIÓN SUELDO.....	17
TABLA 4-1. ANÁLISIS EPS DE LA APLICACIÓN CONSUMO SERVICIOS .....	22
TABLA 4-2. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN “SERVICIOS” .....	23
TABLA 4-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO DE LA APLICACIÓN “SERVICIOS” .....	25
TABLA 4-4. PROPIEDADES DEL CONTROL SHAPE .....	25
TABLA 4-5. PROPIEDADES DEL CONTROL IMAGE.....	26
TABLA 4-6. PROPIEDADES DEL CONTROL COMBOBOX .....	27
TABLA 4-7. PROPIEDADES DE LOS CONTROLES LINE.....	27
TABLA 5-1. ANÁLISIS EPS DE LA APLICACIÓN BINOMIAL .....	32
TABLA 5-2. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN BINOMIAL .....	33
TABLA 5-3. PROPIEDADES DE LOS CONTROLES DE LA APLICACIÓN BINOMIAL .....	35
TABLA 6-1. ANÁLISIS EPS DE LA APLICACIÓN FIGURAS.....	39
TABLA 6-2. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN FIGURAS .....	40
TABLA 6-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO DE LA APLICACIÓN FIGURAS .....	44
TABLA 7-1. ANÁLISIS EPS DE LA APLICACIÓN “ENCUESTA ESTUDIANTIL FACES” .....	51
TABLA 7-2. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN “ENCUESTA ESTUDIANTIL FACES” .....	52
TABLA 7-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO DE LA APLICACIÓN “ENCUESTA ESTUDIANTIL FACES” .....	56
TABLA 8-1. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN “ENCUESTA ELECCIONES” .....	65
TABLA 8-2. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO DE LA APLICACIÓN “ENCUESTA ELECCIONES” .....	69
TABLA 8-3. PROPIEDADES DEL FORMULARIO 2 DE LA APLICACIÓN “ENCUESTA ELECCIONES” .....	69
TABLA 8-4. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO 2 DE LA APLICACIÓN .....	70
TABLA 9-1. ANÁLISIS EPS DE LA APLICACIÓN “ENCUESTA”.....	77
TABLA 9-2. PROPIEDADES DEL FORMULARIO 1 DE LA APLICACIÓN “ENCUESTA”. .....	79
TABLA 9-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO1 DE LA APLICACIÓN “ENCUESTA”. .....	81
TABLA 9-4. PROPIEDADES DEL FORMULARIO 2 DE LA APLICACIÓN “ENCUESTA”. .....	86

TABLA 9-5. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO 2 DE LA APLICACIÓN “ENCUESTA” .....	86
TABLA 10-1. ANÁLISIS EPS DE LA APLICACIÓN “AGENDA TELEFÓNICA” .....	91
TABLA 10-2. PROPIEDADES DEL FORMULARIO 1 DE LA APLICACIÓN “AGENDA TELEFÓNICA” .....	92
TABLA 10-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO1 DE LA APLICACIÓN “AGENDA TELEFÓNICA” .....	95
TABLA 10-4. PROPIEDADES DEL FORMULARIO 2 DE LA APLICACIÓN “AGENDA TELEFÓNICA” .....	99
TABLA 10-5. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO 2 DE LA APLICACIÓN “AGENDA TELEFÓNICA” .....	99
TABLA 11-1. ANÁLISIS EPS DE LA APLICACIÓN “ARCHIVO DE TEXTO” .....	103
TABLA 11-2. PROPIEDADES DEL FORMULARIO 1 DE LA APLICACIÓN “ARCHIVO DE TEXTO” .....	104
TABLA 11-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO1 DE LA APLICACIÓN “ARCHIVO DE TEXTO” .....	106
TABLA 11-4. PROPIEDADES DEL FORMULARIO 2 DE LA APLICACIÓN “ARCHIVO DE TEXTO” .....	116
TABLA 11-5. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO 2 DE LA APLICACIÓN “ARCHIVO DE TEXTO” .....	116
TABLA 12-1. ANÁLISIS EPS DE LA APLICACIÓN “BIBLIOTECA” .....	117
TABLA 12-2. PROPIEDADES DEL FORMULARIO DE LA APLICACIÓN “BIBLIOTECA” .....	118
TABLA 12-3. PROPIEDADES DE LOS CONTROLES DEL FORMULARIO DE LA APLICACIÓN “BIBLIOTECA” .....	121
TABLA A-1. DESCRIPCIÓN DE LOS CONTROLES DE LA CAJA DE HERRAMIENTAS .....	136
TABLA A-2. PRINCIPALES PROPIEDADES DE UN FORMULARIO .....	144
TABLA A-3. PRINCIPALES EVENTOS DE UN FORMULARIO .....	144
TABLA A-4. PRINCIPALES MÉTODOS DE UN FORMULARIO .....	145
TABLA A-5. PRINCIPALES PROPIEDADES DE UNA ETIQUETA .....	146
TABLA A-6. PRINCIPALES PROPIEDADES DE UN CUADRO DE TEXTO .....	146
TABLA A-7. PRINCIPALES EVENTOS DE UN CUADRO DE TEXTO .....	147
TABLA A-8. PRINCIPALES MÉTODOS DE UN CUADRO DE TEXTO .....	147
TABLA A-9. PRINCIPALES PROPIEDADES DE UN BOTÓN DE COMANDO .....	148
TABLA A-10. PRINCIPALES EVENTOS DE UN BOTÓN DE COMANDO .....	148
TABLA A-11. TIPOS DE DATOS .....	151
TABLA A-12. OPERADORES ARITMÉTICOS .....	156
TABLA A-13. OPERADORES DE RELACIÓN .....	156
TABLA A-14. OPERADORES LÓGICOS .....	157
TABLA A-15. OPERADORES PARA MANEJO DE CARACTERES .....	157

## Guía de Usuario:

A pesar de que ésta guía está diseñada para los estudiantes de la asignatura Computación II, puede llegar a ser utilizada por otros usuarios. Por lo que es conveniente antes de realizar la primera práctica, dedicar unas líneas para dar algunos conceptos necesarios para el mejor aprovechamiento y entendimiento de las prácticas.

### **Convenciones tipográficas:**

El texto está escrito en Arial 12 Normal, en ocasiones se escriben palabras o frases “*entrecorilladas y en cursiva*” para indicar comandos, opciones de menú o teclas que deben presionarse, el texto dentro de las tablas está en Arial 10 y por último el código de programa de los botones de comandos y los módulos de programación está en `COURIER NEW 10`.

### **Visual Basic:**

Visual Basic es una herramienta de diseño de aplicaciones (programas) bajo el ambiente Windows, en la que éstas se desarrollan en gran parte a partir del diseño de una interfaz gráfica. En una aplicación de Visual Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfaz gráfica.

La forma en que éste tipo de herramientas interactúa con el usuario es a través de un entorno de desarrollo integrado. Un entorno de desarrollo integrado o en inglés *Integrated Development Environment* (“IDE”) es un programa o software compuesto por un conjunto de herramientas mediante las cuales el programador o desarrollador puede editar, compilar, ejecutar y hacer pruebas de la aplicación que está construyendo. Para mayor información a partir de la página 133 del anexo se describe el IDE del Visual Basic.

Un IDE es un entorno de programación que ha sido enpaquetado como un programa de aplicación, es decir, contiene un editor de código, un compilador, un depurador y

un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic por ejemplo puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para MS Word, MS Excel o cualquier otra.

### **Los principales controles de Visual Basic:**

Un control es un objeto de programación que se coloca sobre el formulario para crear la interfaz gráfica de usuario de la aplicación que se está construyendo. A continuación se describen los tres controles que se utilizan con más frecuencia en las aplicaciones:

#### **ETIQUETAS (LABEL)**

Es un control que se usa para mostrar texto que el usuario final de la aplicación no podrá cambiar directamente, aunque pueden ser cambiados o modificados en tiempo de ejecución. Por lo general las etiquetas se usan para identificar y/o documentar otros controles.

#### **CUADROS DE TEXTO (TEXTBOX)**

También denominados control de edición de texto, es un control que se usa para introducir información (datos) en tiempo de ejecución, permite que el usuario final pueda introducir los datos necesarios para el funcionamiento del programa.

#### **BOTON DE COMANDO (COMMANDBUTTON)**

Un botón de comando es control parecido a un interruptor, mediante el cual se le da ordenes a la aplicación, es decir, permite ejecutar una acción o un proceso de la aplicación.

### **Propiedades:**

Las propiedades representan las características de los objetos. Hay propiedades particulares, como *Caption* que la posee los botones de comandos y, genéricas como *Name* que la poseen todos los objetos.

Las propiedades de un objeto pueden ser cambiadas o modificadas tanto en diseño como en ejecución. En modo de diseño se utiliza la ventana de propiedades que proporciona el IDE del Visual Basic, aunque algunas se pueden cambiar usando el ratón directamente sobre los objetos. En modo de ejecución se utiliza el siguiente formato para acceder a las propiedades de un objeto: "*nombre\_objeto.propiedad*".


## Práctica 1. Conociendo el Visual Basic 6.0.

### **Objetivos:**

Al finalizar esta práctica el estudiante será capaz de:


1. Identificar los diferentes elementos que conforman el Entorno Integrado de Desarrollo (IDE) de Microsoft Visual Basic 6.0.
2. Diferenciar el modo de diseño y el de ejecución.
3. Identificar y establecer las principales propiedades de un Formulario.
4. Añadir los controles de etiquetas y botones de comandos a un formulario.
5. Identificar y establecer las principales propiedades de los controles.
6. Trabajar con procedimientos de eventos.
7. Guardar un proyecto e identificar los diferentes archivos que se generan.

### **Actividades a realizar.**

1. Ejecute el Visual Basic 6.0. Para hacerlo ubique el icono  , en Inicio-Todos los Programas-Microsoft Visual Basic 6.0.
2. Seleccione “Exe estándar” en el cuadro de dialogo “Nuevo Proyecto”
3. Una vez cargado el Visual Basic 6.0 se mostrará el Entorno Integrado de Desarrollo., el cual puede observar en el anexo de estas prácticas.
4. Identifique en la pantalla del Entorno Integrado de Desarrollo los diferentes elementos que la componen y que se describen detalladamente en el anexo, a la izquierda la “Caja de herramientas” que contienen los diferentes controles que se usaran para crear las aplicaciones, en el centro el “Formulario” en modo de diseño y hacia la derecha las ventanas “Explorador de proyecto”, “Propiedades” y “Posición del formulario”
5. Estudie las propiedades del formulario y verifique las siguientes :

Propiedad	Descripción	Valor
Caption	Texto que se muestra en la barra de Título del Formulario	Form1
Font	Fuente de escritura (Tipo, tamaño y estilo)	Ms Sans Serif, 8, Normal
Height	Alto en píxel del formulario	3600
Left	Distancia en píxel desde el borde izquierdo de la pantalla hasta el formulario	0
Top	Distancia en píxel desde el borde superior de la pantalla hasta el formulario	0
Width	Ancho en píxel del formulario	4800

Tabla 1-1. Propiedades del formulario de la práctica 1.

6. Modifique el tamaño del formulario alargándolo o encogiéndolo por las esquinas del mismo y verifique las propiedades Height y Width del formulario.
7. Cambie la posición del formulario en la pantalla seleccionándolo en la Ventana Posición del formulario y verifique las propiedades Left y Top.
8. Fije manualmente las propiedades Caption="Mi Primer Programa en VB", Height=5000, Width=7500, Left=500 y Top=500 (modificando los valores en la caja de propiedades). Observe el tamaño que toma y la posición del mismo.
9. Seleccione en la caja de Herramientas el control Label  (Etiqueta), e inserte una etiqueta en el formulario usando la función Arrastrar del ratón. Es decir, coloque el cursor en un punto del formulario y presione el botón izquierdo del ratón y sin soltarlo muévalo hasta el punto final que desee y ahí lo suelta. (Ver figura 1-1).
10. Estudie las propiedades de la etiqueta, en particular las que se muestran en la tabla 1-2 y coloque los valores que se sugieren.

En la figura 1-1 se muestra el formulario con la etiqueta después de fijar los valores de las propiedades.

Propiedad	Descripción	Valor
Alignment	Alineación del texto dentro de la etiqueta (Izquierda, Derecha, Centro)	2 - Center
Appearance	Apariencia del control (Plana o 3D)	1 - 3D
BackColor	Color del fondo de la etiqueta	Seleccionelo a su gusto en la paleta.
BackStyle	Estilo del fondo (Opaco o transparente)	1 - Opaque



BorderStyle	Estilo del borde (sin borde o simple)	1 – Fixed Single
Caption	Texto que se muestra en la etiqueta	Bienvenidos a mi Primer Programa en VB
Font	Fuente de escritura (Tipo, tamaño y estilo)	Comic Sans MS, 22, Negrita
Height	Alto en píxel de la etiqueta	1250
Left	Distancia en píxel desde el borde izquierdo del formulario hasta la etiqueta	200
Top	Distancia en píxel desde el borde superior del formulario hasta la etiqueta	200
Width	Ancho en píxel de la etiqueta	7000

Tabla 1-2. Propiedades de la etiqueta 1 del formulario.

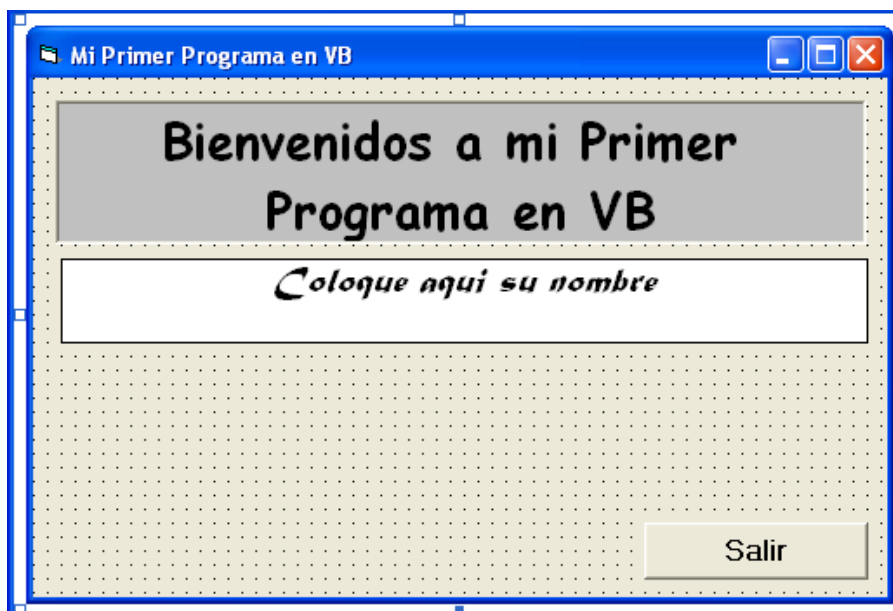






Figura 1-1. Formulario con las dos etiquetas y un botón de comando

11. Ejecute la aplicación presionando el botón Inicio en la Barra de Herramientas

. Se ha creado la primera aplicación bajo Windows con VB, observe que en este momento no puede cambiar las propiedades. Para salir al modo de diseño presione el botón de salir de “Mi Primer Programa en VB”  o al botón Terminar del IDE .

12. Inserte una nueva etiqueta y modifique las propiedades de tal manera que quede como en la figura 1-1.

13. Vuelva a ejecutar la aplicación y después regrese al modo de diseño.

14. Seleccione el control CommandButton  (Botón de Comando) e inserte uno en la parte inferior derecha del formulario (ver figura 1-1).

15. Cambie la propiedad Caption por *Salir*.

16. Ejecute la aplicación y presione el botón salir. ¿Que sucedió? ¡¡¡Nada!!! No se ha programado todavía.

Como se ha dicho antes Visual Basic emplea un modelo de programación manejada por eventos. Los eventos son las acciones del usuario sobre un programa, tales como, hacer clic sobre un botón, el hacer doble clic sobre un archivo 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 puntero del ratón.

Para que el botón que se acaba de crear funcione se tiene que programar, es decir, se tiene que escribir el código de lo que se quiere que el programa haga al pulsarlo. Para esto, haga doble clic sobre el botón (en modo de diseño) y se mostrará la ventana de código, es ahí donde se va a programar el evento (ver figura 5). Lo que aparece es el encabezado para programar el evento clic del objeto Command1. Coloque en medio de las dos líneas la Instrucción *End*.

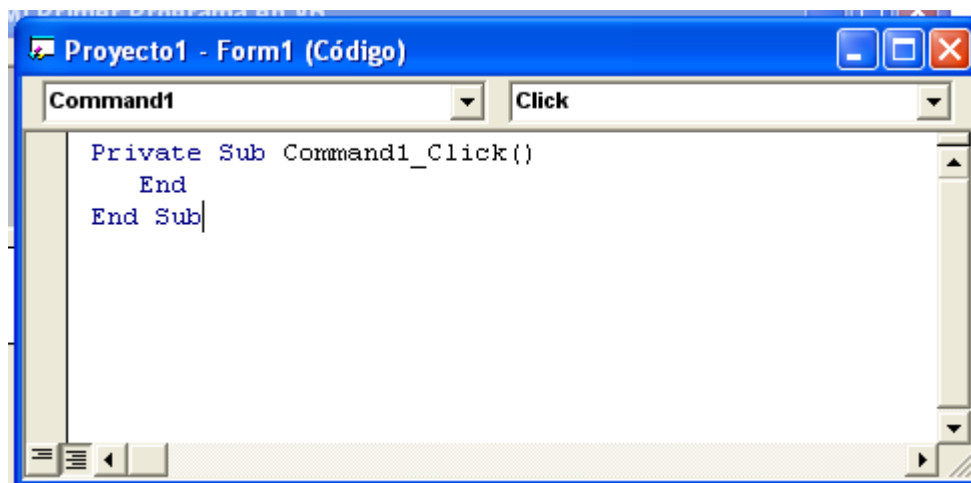





Figura 1-2. Ventana de Código



Ejecute el programa y pulse el botón **Salir**. En este momento estamos en modo de diseño y por lo tanto hemos programado nuestro primer evento.

17. Vamos ahora a almacenar el proyecto, para lo cual pulse Archivo – Guardar proyecto y se mostrará el cuadro de dialogo Guardar archivo como para guardar el formulario, en el cuadro de texto Guardar en seleccione Mis documentos y luego cree un carpeta pulsando el botón Crear nueva Carpeta  con su nombre-practica 1 (Ejemplo: Pedro Perez-Practica 1) y presione el botón Guardar.


Después de esto se mostrará el cuadro de dialogo Guardar proyecto como para guardar el proyecto y presione el botón Guardar.

Verifique en el directorio que acabamos de crear que están los siguientes archivos:

-  Proyecto1 , es este el archivo que debemos abrir para cargar el proyecto; y
-  Form1 (formulario),

18. Presione la flecha del botón Insertar formulario  (en la barra de herramientas) y seleccione en el menú contextual  Módulo , para crear un modulo de programa presione Abrir en el cuadro de dialogo Agregar módulo. Observe que el explorador de proyecto aparece un nuevo elemento.

19. Vuelva a guardar el proyecto y se mostrará el cuadro de dialogo “Guardar

archivo como” para guardar el módulo, este tendrá extensión .bas  Module1.bas .

## Práctica 2. Controles Básicos.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

8. Añadir controles a un formulario (Etiquetas, cuadro de texto y botones de comando)
9. Establecer las propiedades a los controles.
10. Programar eventos de los controles, en especial el evento click de los botones de comando.
11. Crear las siguientes aplicaciones:
  - Calcular la nota promedio de un estudiante conociendo las notas de los tres exámenes.
  - Dada la temperatura en °Centígrados calcular su equivalente en °Kelvin y °Fahrenheit.

### Antes de comenzar

Realicemos el Análisis E-P-S de la Aplicación 1.

#### Entrada.

Dato	Identificador	Tipo	Restricciones
Nombre del Estudiante	Nombre	Cadena	-
Nota del examen 1	Nota1	Entero	[0,20]
Nota del examen 2	Nota2	Entero	[0,20]
Nota del examen 3	Nota3	Entero	[0,20]

Tabla 2-1. Análisis EPS de la aplicación 1.

#### Proceso

Calcular la Nota promedio como la suma de las tres notas dividido entre 3

$$NotaP = \frac{Nota1 + Nota2 + Nota3}{3}$$

## Salida

La nota promedio

Para la interfaz de usuario se usaran los siguientes controles:

- 7 etiquetas (label)
- 4 cajas de texto (TextBox)
- 3 botones de comando (CommandButton)

### Actividades a realizar.

1. Ejecute el Visual Basic 6.0, seleccione “Exe estándar” en el cuadro de dialogo “Nuevo Proyecto”.
2. Coloque las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Práctica # 2. Computación II – Aplicación 1
BackColor	Escójaló en la paleta de colores
Left	3000
Top	2000
Height	6000
Width	8000

Tabla 2-2. Propiedades del formulario de la práctica 2.

3. Agregue al formulario 7 etiquetas (Label), 4 cuadros de texto (TextBox), 3 botones (CommandButton) y fije las propiedades a cada uno de ellos que se muestran en la tabla 2-3.

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Calculo de la Nota Promedio de un Estudiante
	Width, Height, Top y Left	6850, 500, 200, 500
	Alignment	2 - Center
	BorderStyle	1 – Fixed Single
	Font	Arial, Negrita cursiva, 15
Etiqueta 2	(Nombre)	Label2
	Caption	Datos de Entrada:
	Width, Height, Top y Left	2500, 400, 900, 250
	Font	Arial, Negrita, 11
	Alignment	0 – Left Justify

Etiqueta 3	(Nombre)	Label3
	Caption	Nombre del Estudiante:
	Width, Height, Top y Left	2700, 300, 1400, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 4	(Nombre)	Label4
	Caption	Nota del examen 1:
	Width, Height, Top y Left	2700, 300, 2000, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 5	(Nombre)	Label5
	Caption	Nota del examen 2:
	Width, Height, Top y Left	2700, 300, 2600, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 6	(Nombre)	Label6
	Caption	Nota del examen 3:
	Width, Height, Top y Left	2700, 300, 3200, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 7	(Nombre)	Label7
	Caption	Resultados:
	Width, Height, Top y Left	6800, 600, 4000, 250
	Font	Arial, Normal, 12
	BorderStyle	1 – Fixed Single
	Alignment	0 – Left Justify
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	4500, 400, 1400, 3100
	Font	Arial, Normal, 11
Cuadro de texto 2	(Nombre)	Text2
	Text	
	Width, Height, Top y Left	800, 400, 2000, 3100
	Font	Arial, Normal, 11
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	800, 400, 2600, 3100
	Font	Arial, Normal, 11
Cuadro de texto 4	(Nombre)	Text4
	Text	
	Width, Height, Top y Left	800, 400, 3200, 3100
	Font	Arial, Normal, 11

Botón 1	(Nombre)	Command1
	Caption	&Calcular
	Width, Height, Top y Left	1400, 400, 5000 y 2700
Botón 2	(Nombre)	Command2
	Caption	&Limpiar
	Width, Height, Top y Left	1400, 400, 5000 y 4400
Botón 3	(Nombre)	Command3
	Caption	&Salir
	Width, Height, Top y Left	1400, 400, 5000 y 6100

Tabla 2-3. Propiedades de los controles del formulario de la práctica 2.

El formulario debe quedar aproximadamente como se muestra en la figura 2-1.

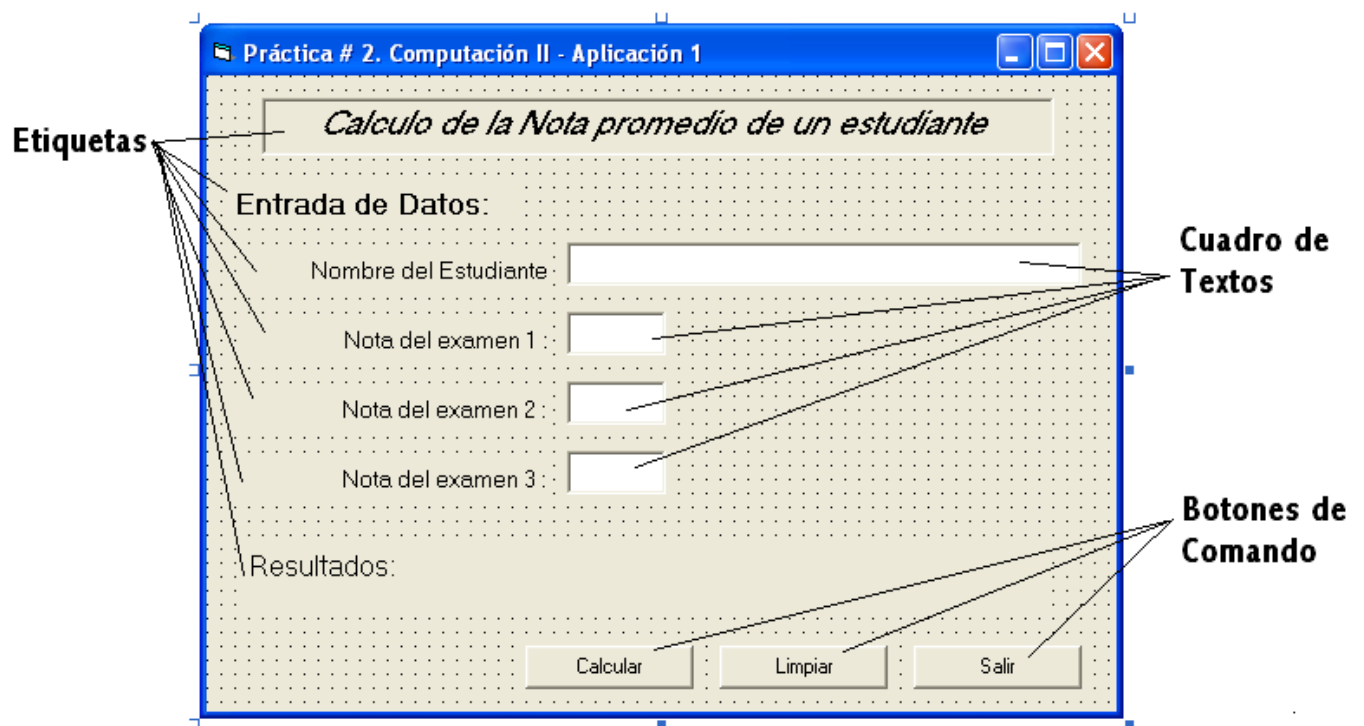



Figura 2-1. Formulario de la Aplicación Calcular Nota promedio

- Se tiene lista la interfase de usuario de la aplicación, y se puede ahora a almacenar el proyecto igual como se hizo en la práctica anterior. Recuerde crear una carpeta para guardar el proyecto, la cual podría ser: Practica 2 – Comp II.

5. Se invoca al editor de código presionando ver código  en la parte superior izquierda del explorador de proyecto y cuando aparezca la ventana escriba la siguiente instrucción:

```
Option Explicit
```

Esta instrucción es para obligar a declarar todas las variables.

6. Ahora se va a programar el evento click() de cada uno de los tres botones de ordenes de nuestra aplicación. El evento click sucede cuando el usuario pulsa, en modo de ejecución, el control con el botón izquierdo del ratón. Para programar este evento haga “doble clic” sobre el botón Calcular en modo de diseño. Al hacer esto aparece la ventana de código con el encabezado y fin del procedimiento que contendrá al evento. Termine de programar el evento para el botón 1 (Calcular):

```
Private Sub Command1_Click()  
    ' Declaración de Variables  
    Dim Nombre As String  
    Dim Nota1 As Integer  
    Dim Nota2 As Integer  
    Dim Nota3 As Integer  
    Dim Nota_P As Single  
  
    ' Lectura de datos  
    Nombre = Text1.Text  
    Nota1 = Text2.Text  
    Nota2 = Text3.Text  
    Nota3 = Text4.Text  
  
    ' Calculo de la nota Propedio  
    Nota_P = (Nota1 + Nota2 + Nota3) / 3  
  
    ' Mostrar Resultado  
    Label7.Caption = "Nota Promedio = " & Nota_P  
  
End Sub
```



7. Una vez terminado de programar el evento click() del botón *Calcular*, cierre la ventana de código y haga “doble clic” al botón *Limpiar*, para programarlo:

```
Private Sub Command2_Click()  
    Text1.Text = ""  
    Text2.Text = ""  
    Text3.Text = ""  
    Text4.Text = ""  
    Label7.Caption = "Resultados"  
End Sub
```

8. Realice el mismo procedimiento para programar el botón Salir:

```
Private Sub Command3_Click()  
    End  
End Sub
```

9. Vuelva a almacenar la aplicación y el formulario. Es una buena practica guardar de vez en cuando los archivos que contienen los datos y programas de lo que se está realizando para evitar pérdida de información sobre todo por fallas eléctricas.

10. Ejecute la aplicación (modo ejecución) y verifique que funcione introduciendo los siguientes datos:

Nombre: Pedro Pérez

Nota 1: 12

Nota 2: 14

Nota 3: 17

El resultado debe ser 14,333333; como se muestra en la figura 2-2

Práctica # 2. Computación II - Aplicación 1

*Calculo de la Nota promedio de un estudiante*

Entrada de Datos:

Nombre del Estudiante : Pedro Pérez

Nota del examen 1 : 12

Nota del examen 2 : 14

Nota del examen 3 : 17

Nota Promedio = 14,33333

Calcular    Limpiar    Salir

Figura 2-2. Resultado para los datos de la práctica

11. Queda de ejercicio para el estudiante la realización de la aplicación 2 (Cambio de temperatura).

Para transformar una temperatura de °C a °K y a °F se utilizan las siguientes relaciones:

$$^{\circ}\text{K} = ^{\circ}\text{C} + 273$$

$$^{\circ}\text{F} = ^{\circ}\text{C} * 9 / 5 + 32$$

## Práctica 3. Estructuras de Decisión.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Usar los controles básicos de un formulario con sus respectivas propiedades para crear la interfaz de usuario.
2. Aplicar la operación de concatenación de cadenas de caracteres.
3. Usar las estructuras de decisión en la programación de los eventos.
4. Crear las siguientes aplicaciones:
  - Calcular el sueldo que recibirá un vendedor cuya remuneración básica depende del tiempo en la empresa (500 Bs. si tiene menos de 10 meses y 550 Bs. si tiene 10 o mas), más una comisión por las ventas mensuales realizadas (.3% de las ventas si vende hasta 5000 y 3.5% si vende mas de 5000) y le descuentan 1% por Ahorro Habitacional, 2% por Seguro Social y 5% para Caja de Ahorros.
  - Calcular las raíces de la ecuación de segundo grado.

### Antes de comenzar.

A continuación se presenta el Análisis Entrada-Proceso-Salida (EPS) para la aplicación 1.

### Análisis E-P-S.

#### Entrada.

Dato	Identificador	Tipo	Restricciones
Tiempo en meses en la empresa	TM	Entero	>0
Total Ventas mensuales	Ventas	Real	>0
Nombre del Vendedor	Nombre	Alfanumérico	-

Tabla 3-1. Análisis EPS de la aplicación Sueldo

**Proceso**

- Calcular la Remuneración Básica

$$RemBas = \begin{cases} 200000 & \text{Si } TM < 10 \\ 250000 & \text{Si } TM \geq 10 \end{cases}$$

- Calcular la Comisión por Ventas,  $Comision = \begin{cases} 3\%Ventas & \text{Si } Ventas \leq 5000 \\ 3.5\%Ventas & \text{Si } Ventas > 5000 \end{cases}$

- Calcular el Total de Ingresos

$$Total\_I = RemBas + Comisi3n$$

- Calcular el Descuento por Ahorro Habitacional

$$AH = 1\% \text{ del Total\_I}$$

- Calcular el Descuento por Seguro Social

$$SS = 2\% \text{ del Total\_I}$$

- Calcular el Descuento por Caja de Ahorros

$$CA = 5\% \text{ del Total\_I}$$

- Calcular Total de Descuentos

$$Total\_Des = AH + SS + CA$$

- Calcular Neto a Cobrar

$$Neto\_Cobrar = Total\_I - Total\_Des$$

**Salida:**

Imprimir cada uno de los conceptos que forman parte del cálculo del Sueldo Neto

Para la interfaz de usuario se usarán los siguientes controles:

- 6 etiquetas (label)
- 3 cajas de texto (TextBox)
- 3 botones de comando (CommandButton)

**Actividades a realizar.**

1. Ejecute el Visual Basic 6.0, seleccione “Exe estándar” en el cuadro de dialogo “Nuevo Proyecto”.

2. Coloque las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Práctica # 3. Computación II – Aplicación 1
BackColor	Escójaló en la paleta de colores
Left	3000
Top	2500
Height	7300
Width	8350

Tabla 3-2. Propiedades del formulario de la aplicación Sueldo

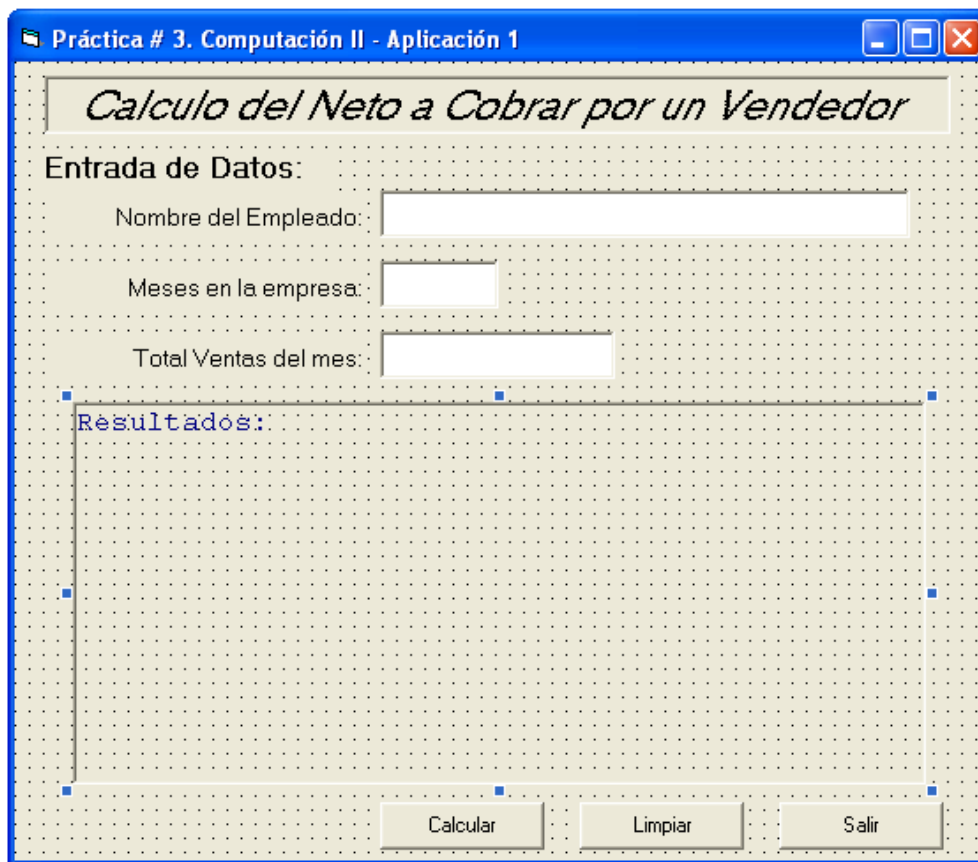



Figura 3-1. Formulario de la Aplicación Sueldo

3. Agregue al formulario 6 etiquetas (Label), 3 cuadros de texto (TextBox), 3 botones (CommandButton). La interfaz de usuario será aproximadamente como se muestra en la figura 3-1. Fije las siguientes propiedades a los controles:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Calculo del Neto a Cobrar por un vendedor
	Width, Height, Top y Left	7700, 500, 120, 250
	Alignment	2 – Center
	BorderStyle	1 – Fixed Single
	Font	Arial, Negrita cursiva, 15
Etiqueta 2	(Nombre)	Label2
	Caption	Datos de Entrada:
	Width, Height, Top y Left	2500, 400, 750, 250
	Font	Arial, Negrita, 11
	Alignment	0 – Left Justify
Etiqueta 3	(Nombre)	Label3
	Caption	Nombre del Vendedor:
	Width, Height, Top y Left	2700, 300, 1200, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 4	(Nombre)	Label4
	Caption	Meses en la empresa:
	Width, Height, Top y Left	2700, 300, 1800, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 5	(Nombre)	Label5
	Caption	Total ventas del mes:
	Width, Height, Top y Left	2700, 300, 2400, 250
	Font	Arial, Normal, 9
	Alignment	1 – Right Justify
Etiqueta 6	(Nombre)	Label6
	Caption	Resultados:
	Width, Height, Top y Left	7250, 3250, 2900, 500
	Font	Courier New, Normal, 12
	BorderStyle	1 – Fixed Single
	Forecolor	&H00800000&
	Alignment	0 – Left Justify
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	4500, 400, 1100, 3100
	TabIndex	1
	Font	Arial, Normal, 11
Cuadro de texto 2	(Nombre)	Text2
	Text	

	Width, Height, Top y Left	1000, 400, 1700, 3100
	TabIndex	2
	Font	Arial, Normal, 11
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	2000, 400, 2300, 3100
	TabIndex	3
	Font	Arial, Normal, 11
Botón 1	(Nombre)	Command1
	Caption	&Calcular
	TabIndex	4
	Width, Height, Top y Left	1400, 400, 6300 y 3100
Botón 2	(Nombre)	Command2
	Caption	&Limpiar
	TabIndex	5
	Width, Height, Top y Left	1400, 400, 6300 y 4800
Botón 3	(Nombre)	Command3
	Caption	&Salir
	TabIndex	6
	Width, Height, Top y Left	1400, 400, 6300 y 6500

Tabla 3-3. Propiedades de los controles del formulario de la aplicación Sueldo

- Almacene el proyecto, siguiendo el mismo procedimiento de las prácticas anteriores. Recuerde crear una carpeta para guardar el proyecto, la cual podría ser: Práctica 3 – Comp II.
- Invoquemos al editor de código presionando ver código  en la parte superior izquierda del explorador de proyecto y cuando aparezca la ventana escriba la siguiente instrucción:

```
Option Explicit
Dim CR As String
```

Se debe recordar que con la primera instrucción se obliga a declarar todas las variables. La variable CR que se declara a nivel del formulario es para usarla como el carácter “*Enter*” para separar cada línea de salida que se mostrará en la etiqueta 6. Esta será asignada en el evento “*load*” del formulario, el cual

programaremos a continuación. Salga de la ventana de código y haga doble clic sobre el formulario para invocar al procedimiento form\_load():

```
Private Sub Form_Load()  
    CR = Chr(10) & Chr(13)  
End Sub
```

6. Ahora se programan los tres botones de comando. Recuerde que para invocar el editor de código se debe hacer doble clic en el respectivo botón en modo de diseño:

```
Private Sub Command1_Click()  
    ' Declaración de las variables  
    Dim Nombre As String  
    Dim TM As Integer          ' Tiempo en Meses en la empresa  
    Dim Ventas As Single  
    Dim Comision As Single  
    Dim RemBas As Single  
    Dim Total_Ing As Single  
    Dim AH As Single          ' Ahorro Habitacional  
    Dim SS As Single          ' Seguro Social  
    Dim CA As Single          ' Caja de Ahorros  
    Dim Total_Des As Single  
    Dim Neto_Cobrar As Single  
    Dim Cad As String  
  
    ' Lectura de Datos  
    Nombre = Text1.Text  
    TM = Text2.Text  
    Ventas = Text3.Text  
  
    ' Calcular los distintos conceptos que forman el pago del vendedor  
    If TM < 10 Then  
        RemBas = 500  
    Else  
        RemBas = 550  
    End If  
  
    If Ventas <= 5000 Then  
        Comision = 0.03 * Ventas  
    Else  
        Comision = 0.035 * Ventas  
    End If
```



```

    Total_Ing = RemBas + Comision
    AH = 0.01 * Total_Ing
    SS = 0.02 * Total_Ing
    CA = 0.05 * Total_Ing
    Total_Des = AH + SS + CA
    Neto_Cobrar = Total_Ing - Total_Des

    ' Mostrar los resultados en la etiqueta 6 utilizando el concepto
del
    ' caracter
    ' Enter y la concatenación de cadenas de caracteres
    ' La salida se irá almacenando en una variable de tipo alfanumerico
    ' llamada CAD y luego se mostrará en la etiqueta.

    Cad = "Recibo de Pago" & CR & "-----" & CR & "Nombre del
Empleado: " & Nombre
    Cad = Cad & CR & "Remuneración Básica: " & RemBas & CR & "Comisión
por Ventas: " & Comision
    Cad = Cad & CR & "Total Ingresos: " & Total_Ing & CR & "Ahorro
Habitacional: " & AH
    Cad = Cad & CR & "Seguro Social: " & SS & CR & "Caja de Ahorros: "
& CA
    Cad = Cad & CR & "Total Descuentos: " & Total_Des & CR & "Neto a
Cobrar: " & Neto_Cobrar

    Label6.Caption = Cad

End Sub

Private Sub Command2_Click()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Label6.Caption = "Resultados"
End Sub

Private Sub Command3_Click()
    End
End Sub

```

7. Una vez programados los botones, guarde los cambios, ejecute la aplicación con algunos datos y verifique los resultados con los siguientes datos

Nombre del Empleado: Juan Antonio Díaz

Tiempo en la empresa: 8

Total Ventas del mes: 8567

Los resultados se muestran en la figura 3-2

Práctica # 3. Computación II - Aplicación 1

*Calculo del Neto a Cobrar por un Vendedor*

Entrada de Datos:

Nombre del Empleado: Juan Antonio Díaz

Meses en la empresa: 8

Total Ventas del mes: 8567

Recibo de Pago

-----

Nombre del Empleado: Juan Antonio Díaz  
Remuneración Básica: 500  
Comisión por Ventas: 299,845  
Total Ingresos: 799,845  
Ahorro Habitacional: 7,99845  
Seguro Social: 15,9969  
Caja de Ahorros: 39,99225  
Total Descuentos: 63,9876  
Neto a Cobrar: 735,8574

Calcular    Limpiar    Salir

Figura 3-2. Resultados para los datos planteados.

8. Para completar la práctica realice la Aplicación 2 (Solución de la ecuación de segundo grado).

## Práctica 4. Más controles. Validación de datos I.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Usar los controles ComboBox (Cuadro combinado de texto), Picture (Contenedor de imágenes), Shape (formas geométricas) y Line (líneas) con sus respectivas propiedades en el diseño de la interfaz de usuario.
2. Usar la estructura de programación selección múltiple (CASE OF) en las aplicaciones.
3. Usar las estructuras de decisión para la validación de datos de entrada.
4. Usar la función MsgBox para mostrar mensajes (errores) a los usuarios.
5. Crear la siguiente aplicación:

Se desea un programa que calcule el total a facturar una compañía eléctrica a un cliente por el consumo de Electricidad, Gas y Aseo Urbano, según las siguientes tarifas:

- La electricidad se cobra como una tarifa básica por un consumo mínimo más un consumo extra:
  - La tarifa básica es de la siguiente forma:
    - 1,80 Bs. por los primeros 100 KWH para la Residencial 1
    - 3,00 Bs. por los primeros 150 KWH para la Residencial 2
    - 12,00 Bs. por los primeros 300 KWH Comercial
  - El consumo extra es 0,02, 0,03 y 0,05 Bs. cada KWH extra que se consuma respectivamente.
- El gas lo cobran a 3,00 Bs. cada m<sup>3</sup> para la Residencial (1 y 2) y 4,20 Bs. cada m<sup>3</sup> para la Comercial.
- El Aseo Urbano cuesta 2,20, 4,50 y 18,00 Bs. respectivamente según el tipo de tarifa.

**Antes de comenzar.****Análisis E-P-S.****Entrada.**

Dato	Identificador	Tipo	Restricciones
Cuenta Cliente	Nombre	Cadena de caracteres	-
Tipo de Usuario	Tipo	Entero	1 = Residencial 1 2 = Residencial 2 3 = Comercial
Cantidad de KWH	KWH	Entero	>0
Cantidad de m <sup>3</sup> de gas	M3	Entero	>0

Tabla 4-1. Análisis EPS de la aplicación consumo servicios

**Proceso**

- Dependiendo del Tipo de Usuario Calcular Tarifa Básica, Costo KWH extra, KWH Extras, GAS y ASEO.

$RB = 1,80$ por los primeros 100 KWH 0,02 Bs cada KWH extra $GAS = 3,00$ cada M <sup>3</sup> $ASEO = 1,80$	} Si es Residencial 1	
$RB = 3,20$ por los primeros 150 KWH 0,03 Bs cada KWH extra $GAS = 3,00$ cada M <sup>3</sup> $ASEO = 4,50$		} Si es Residencial 2
$RB = 12,00$ por los primeros 300 KWH 0,05 Bs cada KWH extra $GAS = 4,20$ cada M <sup>3</sup> $ASEO = 18,00$		

- Pago Extra= KWH extras \* Costo KWH extra
- Elect = Tarifa Basica + PagoExtra
- Total = Elect + Gas + Aseo

**Salida:**

Nombre, Pago por electricidad, Pago por Aseo y Pago por Gas

Para la interfaz de usuario se usaran los siguientes controles:

- 7 etiquetas (label)
- 3 cajas de texto (TextBox)
- 3 botones de comando (CommandButton)
- 1 cuadro combinado de texto (ComboBox)
- 1 figura geométricas (Shape)
- 3 líneas (Line)
- 1 Contenedor de gráficos (Picture)

**Actividades a realizar.**

1. Ejecute el Visual Basic 6.0, seleccione “*Exe estándar*” en el cuadro de dialogo “*Nuevo Proyecto*”.
2. Coloque las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Práctica # 4. Computación I
BackColor	&H00C0C0C0&
StarUpPosition	1 - CenterOwner
BorderStyle	1 – Fixed Single

Tabla 4-2. Propiedades del formulario de la aplicación “Servicios”

3. Agregue al formulario 7 etiquetas, 3 cuadros de texto, 3 botones aproximadamente como se muestra en la figura 4-1:
4. Fije las siguientes propiedades a los controles:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Compañía de Electricidad de la Ciudad
	Width, Height, Top y Left	6200, 1000, 250, 200
	Alignment	2 – Center
	BorderStyle	1 – Fixed Single
	BackStyle	0 – Transparent
	Font	Ms Sans Serif, Normal, 18

Etiqueta 2	(Nombre)	Label2
	Caption	Sistema de Facturación de Consumo de Electricidad y otros Servicios Públicos
	Width, Height, Top y Left	5550, 860, 1450, 450
	Alignment	2 – Center
	BorderStyle	1 – Fixed Single
	BackStyle	0 - Transparent
	Font	Ms Sans Serif, Normal, 14
Etiqueta 3	(Nombre)	Label3
	Caption	Nro de Cuenta :
	Width, Height, Top y Left	2500, 500, 2800, 120
	Font	Arial, Negrita, 10
	Alignment	1 – Right Justify
Etiqueta 4	(Nombre)	Label4
	Caption	Tipo de Usuario :
	Width, Height, Top y Left	2500, 500, 3500, 120
	Font	Arial, Negrita, 10
	Alignment	1 – Right Justify
Etiqueta 5	(Nombre)	Label5
	Caption	Consumo Electricidad (KWH) :
	Width, Height, Top y Left	2500, 500, 4200, 120
	Font	Arial, Negrita, 10
	Alignment	1 – Right Justify
Etiqueta 6	(Nombre)	Label6
	Caption	Consumo gas (m3) :
	Width, Height, Top y Left	2500, 500, 4900, 120
	Font	Arial, Negrita, 10
	Alignment	1 – Right Justify
Etiqueta 7	(Nombre)	Label7
	Caption	Resultados:
	Width, Height, Top y Left	3700, 2800, 2700, 5050
	Font	Courier New, Normal, 10
	BorderStyle	1 – Fixed Single
	Forecolor	&H00800000&
	Alignment	0 – Left Justify
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	2100, 500, 2800, 2700
	TabIndex	1
	Font	Arial, Normal, 11
Cuadro de texto 2	(Nombre)	Text2
	Text	

	Width, Height, Top y Left	1500, 500, 4200, 2700
	TabIndex	3
	Font	Arial, Normal, 11
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	1500, 500, 4900, 2700
	TabIndex	4
Botón 1	Font	Arial, Normal, 11
	(Nombre)	Command1
	Caption	&Calcular
	TabIndex	5
	Font	Century Gothic, Negrita, 11
Botón 2	Width, Height, Top y Left	1600, 500, 5800, 3000
	(Nombre)	Command2
	Caption	&Limpiar
	TabIndex	6
	Font	Century Gothic, Negrita, 11
Botón 3	Width, Height, Top y Left	1600, 500, 5800, 5000
	(Nombre)	Command3
	Caption	&Salir
	TabIndex	7
	Font	Century Gothic, Negrita, 11
	Width, Height, Top y Left	1600, 500, 5800, 7000

Tabla 4-3. Propiedades de los controles del formulario de la aplicación "Servicios"

5. Agregue al formulario una figura geométrica (Shape), con este control podemos agregar rectángulos, cuadrados, óvalos, círculos, rectángulos redondeados y cuadrados redondeados (propiedad shape). Con el botón derecho del ratón seleccione la figura que acaba de insertar y ejecute *enviar al fondo*. Fije las siguientes propiedades:

.Control	Propiedad	Valor
Figura 1	(Nombre)	Shape1
	BackColor	&H00FF8080&
	BackStyle	1 - Opaque
	BorderColor	&H00FF0000&
	FillStyle	1 - Transparent
	Shape	2 - Oval
	Width, Height, Top y Left	6500, 2600, 0, 0

Tabla 4-4. Propiedades del control Shape

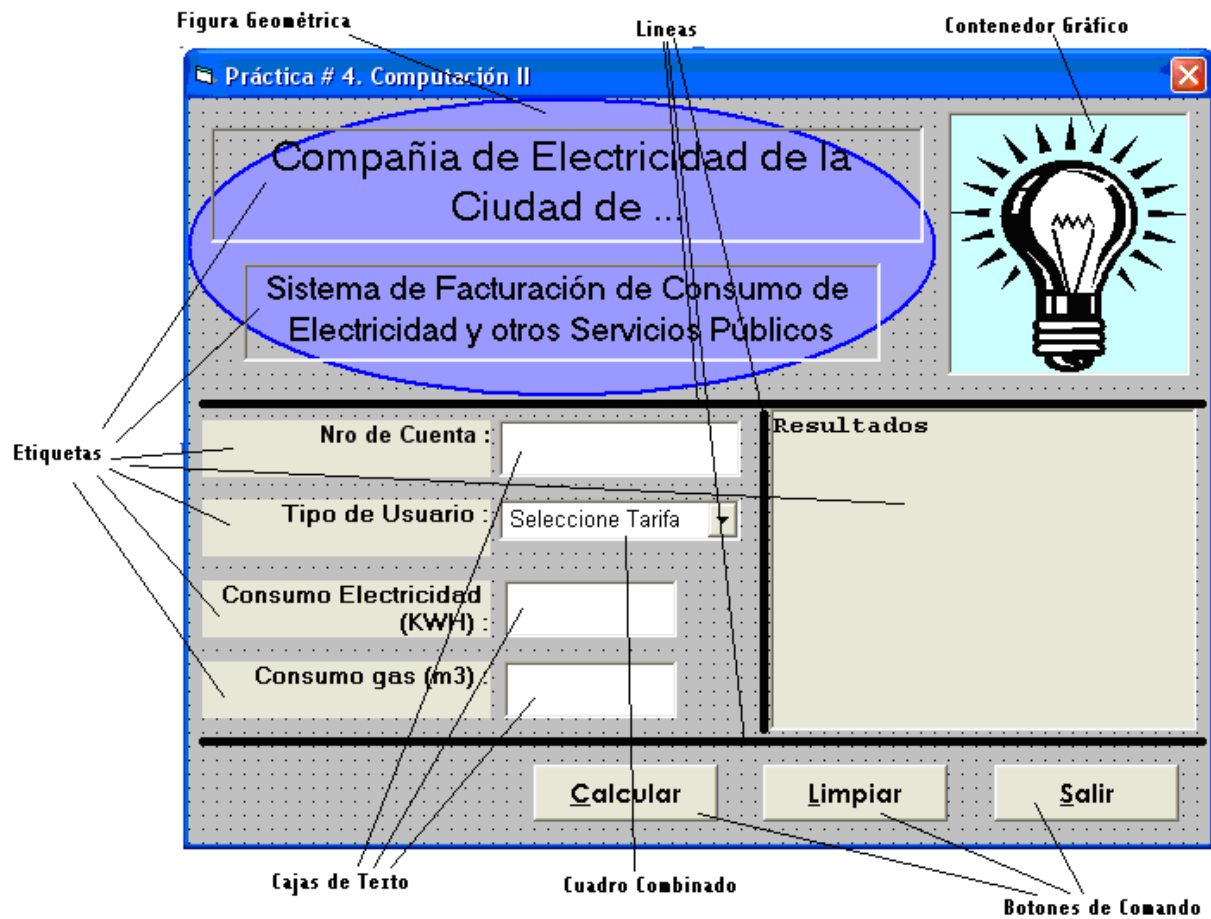


Figura 4-1. Formulario de la Aplicación “Servicios”

6. Agregue al formulario un contenedor de imagen (Picture). Para colocar la imagen en el contenedor gráfico fije la propiedad Picture y localice el archivo gráfico deseado en el cuadro de dialogo Abrir. Fije las siguientes propiedades:

Control	Propiedad	Valor
Imagen 1	(Nombre)	Picture1
	BackColor	&H00FFFC0&
	Picture	bombillo.wmf
	Width, Height, Top y Left	2100, 2300, 120, 6600

Tabla 4-5. Propiedades del control Image

La figura puede ser descargada de la siguiente dirección:  
<http://webdelprofesor.ula.ve/economia/angelz/imagenes/bombillo.wmf>



Agregue al formulario un cuadro combinado de texto (ComboBox) y fije las siguientes propiedades:

Control	Propiedad	Valor
Cuadro combinado 1	(Nombre)	Combo1
	List	Residencial 1 Residencial 2 Comercial
	Text	Seleccione tipo de usuario
	Width, Height, Top y Left	2100, 260, 3500, 2700

Tabla 4-6. Propiedades del control ComboBox

Agregue al formulario 3 controles líneas (Line) y fije las siguientes propiedades:

Control	Propiedad	Valor
Línea 1	(Nombre)	Line1
	BorderColor	&H80000008&
	BorderStyle	1 – Solid
	BorderWidth	5
	DrawMode	13- Copy pen
	X1, X2, Y1, Y2	120, 8800, 2650, 2650
Línea 2	(Nombre)	Line2
	BorderColor	&H80000008&
	BorderStyle	1 – Solid
	BorderWidth	5
	DrawMode	13- Copy pen
	X1, X2, Y1, Y2	120, 8800, 5600, 5600
Línea 3	(Nombre)	Line3
	BorderColor	&H80000008&
	BorderStyle	1 – Solid
	BorderWidth	5
	DrawMode	13- Copy pen
	X1, X2, Y1, Y2	5000, 5000, 2750, 5500

Tabla 4-7. Propiedades de los controles Line

Almacene el proyecto siguiendo el mismo procedimiento de las prácticas anteriores. Recuerda crear una carpeta para guardar el proyecto, la cual podría ser: Practica 4 – Comp II.

invoque al editor de código y coloque la instrucción “*Option Explicit*” al código de la aplicación

```
Option Explicit
Dim CR As String
```

Programa el evento "*load*" del formulario y los eventos "*click*" de los tres botones de ordenes tal y como aparece a continuación.

```
Private Sub Form_Load()
    CR = Chr(13) & Chr(10)
End Sub

Private Sub Command1_Click()
    Dim Cuenta As String * 15
    Dim KWH As Integer
    Dim M3 As Single
    Dim Elect As Single, Gas As Single, Aseo As Single
    Dim Total As Single, TarifaBas As Single, Min As Single
    Dim CostoExt As Single, PagoExtra As Single
    Dim Cad As String

    Dim Usuario As Integer

    ' Lectura y Validación de los datos

    ' El número de cuenta
    If Text1.Text = "" Then
        MsgBox "Error... El número de cuenta no puede estar vacío",
vbOKOnly, "Error"
        Text1.SetFocus
        Exit Sub
    Else
        Cuenta = Text1.Text
    End If

    ' El tipo de usuario
    If Comb1.ListIndex = -1 Then
        MsgBox "Error... Seleccione uno de los usuarios", vbOKOnly, "Error"
        Comb1.SetFocus
        Exit Sub
    Else
        Usuario = Comb1.ListIndex
    End If

    ' El número de Kilovatios-hora
    If Not IsNumeric(Text2.Text) Then
```

```
    MsgBox "Error... El número de KWH debe ser numérico", vbOKOnly,
    "Error"
    Text2.SetFocus
    Exit Sub
Else
    KWH = Text2.Text
    If KWH <= 0 Then
        MsgBox "Error... El número de KWH debe ser mayor que cero",
        vbOKOnly, "Error"
        Text2.SetFocus
        Exit Sub
    End If
End If

' Los metros cúbicos de gas
If Not IsNumeric(Text3.Text) Then
    MsgBox "Error... El número de m.c. de gas debe ser numérico",
    vbOKOnly, "Error"
    Text3.SetFocus
    Exit Sub
Else
    M3 = Text3.Text
    If KWH <= 0 Then
        MsgBox "Error... El número de m.c. de gas debe ser mayor que
        cero", vbOKOnly, "Error"
        Text3.SetFocus
        Exit Sub
    End If
End If

Select Case Usuario
    Case 0
        TarifaBas = 1.8
        Min = 100
        CostoExt = 0.02
        Gas = 3# * M3
        Aseo = 1.8
    Case 1
        TarifaBas = 3.2
        Min = 150
        CostoExt = 0.03
        Gas = 3# * M3
        Aseo = 4.5
    Case 2
```

```
TarifaBas = 12#
Min = 300
CostoExt = 0.05
Gas = 4.2 * M3
Aseo = 18#
End Select

' Verifica si hay consumo extra
If KWH > Min Then
    PagoExtra = (KWH - Min) * CostoExt
Else
    PagoExtra = 0
End If

Elect = TarifaBas + PagoExtra
Total = Elect + Gas + Aseo

' Coloca los resultados en una cadena de caracteres
Cad = "FACTURA" & CR & "-----" & CR
Cad = Cad & "Cuenta : " & Cuenta & CR & "Elect. : " & Elect & CR
Cad = Cad & "Gas      : " & Gas      & CR & "Aseo   : " & Aseo & CR
Cad = Cad & "Total   : " & Total

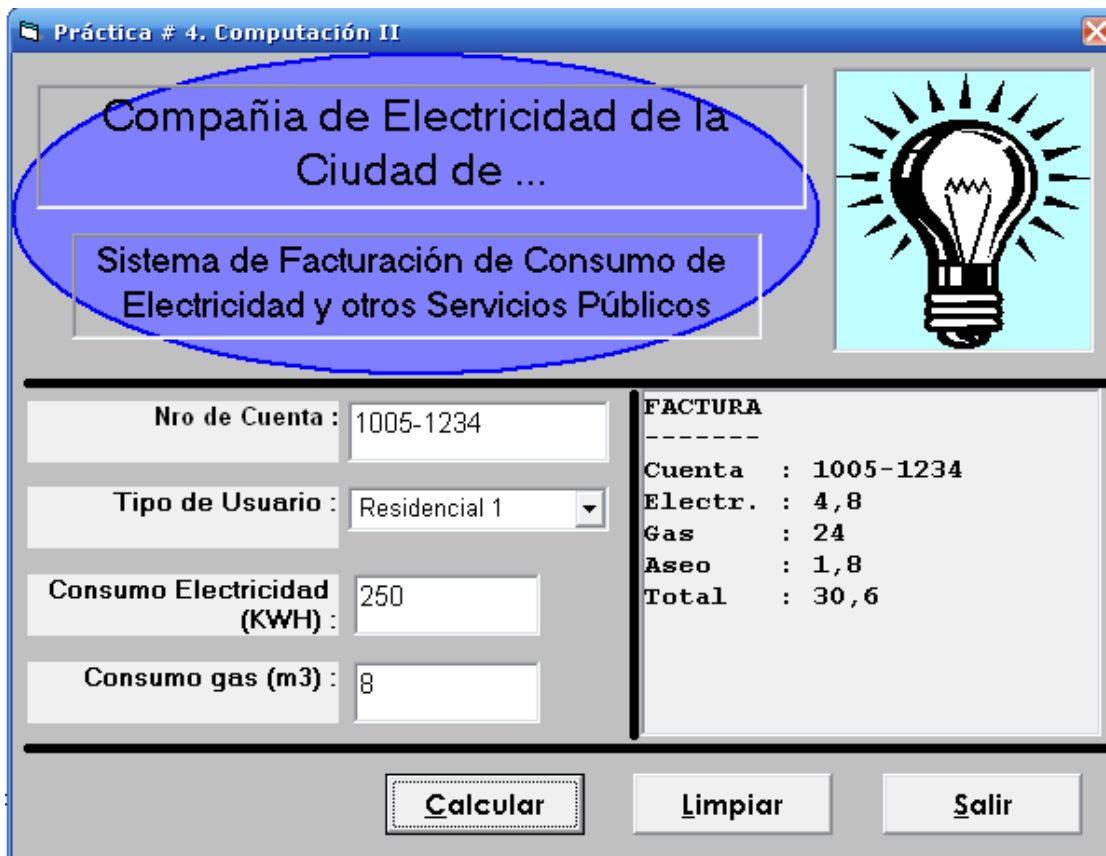
' Muestra los resultados
Label7.Caption = Cad

End Sub

Private Sub Command2_Click()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Combol.ListIndex = -1
    Combol.Text = "Seleccione Tipo Usuario"
End Sub

Private Sub Command3_Click()
    End
End Sub
```


Ejecute la aplicación y verifique que los resultados con los datos que se muestran en la figura 4-2:



Práctica # 4. Computación II

Compañía de Electricidad de la Ciudad de ...

Sistema de Facturación de Consumo de Electricidad y otros Servicios Públicos



Nro de Cuenta :	1005-1234	<b>FACTURA</b>
Tipo de Usuario :	Residencial 1	-----
Consumo Electricidad (KWH) :	250	Cuenta : 1005-1234
Consumo gas (m3) :	8	Electr. : 4,8
		Gas : 24
		Aseo : 1,8
		Total : 30,6

Calcular      Limpiar      Salir

Figura 4-2. Resultados para los datos de la práctica

## Práctica 5. Estructuras de Repetición I.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Usar la estructura de repetición Indexada (Repita para) en la programación de aplicaciones
2. Crear y usar módulos de programas y funciones definidas por el usuario en las aplicaciones.
3. Utilizar los formatos de datos numéricos para mostrar tablas.
4. Crear las siguientes aplicaciones:
  - a. Crear la tabla de probabilidad para la variable aleatoria binomial con n y p dados.
  - b. Crear una tabla que contenga para cada uno de los primeros “n” enteros su cuadrado, cubo y raíz cuadrada

### Antes de comenzar.

#### Análisis E-P-S.

#### Entrada.

Dato	Identificador	Tipo	Restricciones
Número de ensayos	N	Entero	>0
Probabilidad de éxito	P	Real	(0,1)

Tabla 5-1. Análisis EPS de la aplicación binomial

#### Proceso

- 1 La función de probabilidad de la distribución binomial está dada por:

$$P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}; \quad \text{para } x = 0, 1, 2, \dots, n$$

- 2 Para cada valor de x entre 0 y n calcular la probabilidad

**Salida:**

Imprimir cada valor de X con su respectiva probabilidad.

Para la interfaz de usuario se usaran los siguientes controles:

- 4 etiquetas (Label)
- 2 cajas de texto (TextBox)
- 3 botones de comando (CommandButton)
- 3 líneas (Line)

**Actividades a realizar.**

1. Ejecute el Visual Basic 6.0, seleccione “Exe estándar” en el cuadro de dialogo “Nuevo Proyecto”.
2. Coloque las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Práctica # 5. Computación II – Aplicación 1
BackColor	Escójalos en la paleta de colores
Left	2500
Top	850
Height	7000
Width	9750

Tabla 5-2. Propiedades del formulario de la aplicación binomial

3. Agregue al formulario 4 etiquetas (Label), 2 cuadros de texto (TextBox), 3 botones (CommandButton) y 3 líneas (Line). Fije las siguientes propiedades tal y como se muestra en la figura 5-1

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Tabla de Probabilidad Binomial
	Width, Height, Top y Left	8400, 615, 240, 600
	Alignment	2 – Center
	BorderStyle	1 – Fixed Single
	Font	Arial, Negrita cursiva, 18
Etiqueta 2	(Nombre)	Label2
	Caption	Número de Ensayos

	Width, Height, Top y Left	1660, 735, 2150, 480
	Font	Arial, Negrita, 12
	Alignment	1 – Right Justify
Etiqueta 3	(Nombre)	Label3
	Caption	Probabilidad de Exito
	Width, Height, Top y Left	2025, 735, 3500, 240
	Font	Arial, Normal, 12
	Alignment	1 – Right Justify
Etiqueta 4	(Nombre)	Label4
	Caption	Resultados
	Width, Height, Top y Left	5500, 4300, 1080, 4300
	Font	Courier New, Normal, 9
	Alignment	0 – Left Justify
Caja de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	1100, 750, 2150, 2400
	TabIndex	1
	Font	Arial, Normal, 24
Caja de texto 2	(Nombre)	Text2
	Text	
	Width, Height, Top y Left	1500, 750, 3500, 2400
	TabIndex	2
	Font	Arial, Normal, 24
Botón de comando 1	(Nombre)	Command1
	Caption	&Calcular
	TabIndex	3
	Font	MS Sans Serif, Negrita, 12
	Width, Height, Top y Left	1500, 500, 5900 y 4000
Botón de comando 2	(Nombre)	Command2
	Caption	&Limpiar
	TabIndex	4
	Font	MS Sans Serif, Negrita, 12
	Width, Height, Top y Left	1500, 500, 5900 y 6000
Botón de comando 3	(Nombre)	Command3
	Caption	&Salir
	TabIndex	5
	Font	MS Sans Serif, Negrita, 12
	Width, Height, Top y Left	1500, 500, 5900 y 8000
Línea 1	(Nombre)	Line1
	BorderColor	&H80000008&
	BorderStyle	1 – Solid



	BorderWidth	5
	DrawMode	13- Copy pen
	X1, X2, Y1, Y2	120, 9500, 960, 960
Línea 2	(Nombre)	Line2
	BorderColor	&H80000008&
	BorderStyle	1 – Solid
	BorderWidth	5
	DrawMode	13- Copy pen
	X1, X2, Y1, Y2	4080, 4080, 5400, 1080
Línea 3	(Nombre)	Line3
	BorderColor	&H80000008&
	BorderStyle	1 – Solid
	BorderWidth	5
	DrawMode	13- Copy pen
	X1, X2, Y1, Y2	120, 9480, 5640, 5640

Tabla 5-3. Propiedades de los controles de la aplicación binomial

The screenshot shows a graphical user interface for a binomial probability application. The window title is "Práctica # 5. Computación II - Aplicación 1". The main area is titled "Tabla de Probabilidad Binomial". It features two input fields on the left: "Número de ensayos :" and "Probabilidad de Exito :". To the right of these fields is a large, empty rectangular area labeled "Resultados". At the bottom of the window, there are three buttons: "Calcular", "Limpiar", and "Salir".

Figura 5-1. Formulario de la Aplicación binomial

- Almacene el proyecto siguiendo el mismo procedimiento de las prácticas anteriores. Recuerde crear una carpeta para guardarlo, la cual podría ser: Práctica 5 – Comp II.
- Invoque al editor de código y agregue al código de la aplicación la instrucción “*Option Explicit*” y la declaración de la variable utilizada para representar el salto de línea en las etiquetas.

```
Option Explicit
Dim CR As String
```

- Se va a insertar un módulo de programa a nuestro proyecto, en el cual se definirá la función factorial y la función combinatoria necesarias para calcular la probabilidad binomial. Para hacer esto seleccione “*Agregar módulo*” en el menú “*Proyecto*” y se mostrará el editor de código, en el cual agregaremos lo siguiente:

```
' Módulo de programa

' Declaración de la función Factorial()
'
Public Function Factorial(N As Integer) As Long
    Dim F As Long

    F = 1
    For i = 1 To N
        F = F * i
    Next i
    Factorial = F

End Function

' Declaración de la función Combinatoria()
'
Public Function Combinatoria(N As Integer, M As Integer) As Integer

    Combinatoria = Factorial(N) \ (Factorial(M) * Factorial(N - M))

End Function
```

- Almacena de nuevo el proyecto para que el módulo que se acaba de crear, no

se pierda. Compruebe que lo guarda en la misma carpeta.

8. Programe el evento "*load*" del formulario y los eventos "*click*" de los tres botones de órdenes tal y como aparece a continuación.

```
Private Sub Form_Load()  
    CR = Chr(13) & Chr(10)  
End Sub  
' Botón CALCULAR  
'  
Private Sub Command1_Click()  
    Dim N As Integer  
    Dim x As Integer  
    Dim P As Single, Prob As Single, Acum As Single  
    Dim Cad As String  
  
    ' Lectura y validación de datos  
  
    If Not IsNumeric(Text1.Text) Then  
        MsgBox "Error... Debe ser numerico", vbOKOnly, "Error"  
        Text1.SetFocus  
        Exit Sub  
    Else  
        N = Text1.Text  
        If N <= 0 Then  
            MsgBox "Error... N Debe ser mayor que cero", vbOKOnly, "Error"  
            Text1.SetFocus  
            Exit Sub  
        End If  
    End If  
  
    If Not IsNumeric(Text2.Text) Then  
        MsgBox "Error... La probabilidad Debe ser numerico", vbOKOnly,  
        "Error"  
        Text2.SetFocus  
        Exit Sub  
    Else  
        P = Text2.Text  
        If (P <= 0) Or (P > 1) Then  
            MsgBox "Error... Debe estar entre 0 y 1", vbOKOnly, "Error"  
            Text2.SetFocus  
            Exit Sub  
        End If
```

```

End If

Cad = " x | P(X=x) | P(X<=x) " & cr & "-----"
-----" & cr
Acum = 0
For x = 0 To N
    Prob = Combinatoria(N, x) * P ^ x * (1 - P) ^ (N - x)
    Acum = Acum + Prob
    Cad = Cad & Format(x, " 0# | ") & Format(Prob, "0.#0000000 | ")
    Cad = Cad & Format(Acum, "0.#0000000") & cr
Next x
Label4.Caption = Cad

End Sub

' Botón LIMPIAR
'
Private Sub Command2_Click()
    Text1.Text = ""
    Text2.Text = ""
    Label4.Caption = "Resultados"
End Sub

' Botón SALIR
'
Private Sub Command3_Click()
    End
End Sub

```

9. Almacene el proyecto y ejecute el programa.
10. Ejecute el programa y compruebe que la validación de datos funciona introduciendo datos erróneos para  $n$  y  $p$ .
11. Como complemento de la práctica realice la segunda aplicación planteada (Crear una tabla que contenga para cada uno de los primeros “ $n$ ” enteros su cuadrado, cubo y raíz cuadrada).

## Práctica 6. Validación de Datos II. Más Controles.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Validar los datos de acuerdo a las especificaciones de entrada de los mismos usando las estructuras de decisión y el evento LostFocus de los cuadros de texto.
2. Utilizar la estructura de decisión SELECT CASE (en caso de)
3. Utilizar los controles Frame y OptionButton.
4. Trabajar con matrices de controles.
5. Construir una aplicación que calcule el área y el perímetro de un círculo, rectángulo, triángulo rectángulo o un trapecio.

### Antes de comenzar.

#### Análisis E-P-S.

#### Entrada.

Dato	Identificador	Tipo	Restricciones
Figura deseada	Fig	Entero	0=círculo 1=rectángulo 2=triángulo rectángulo 4=trapecio
Radio del círculo	R	Real	> 0
Base del rectángulo, triángulo o base mayor del trapecio	B1	Real	> 0
Base menor del trapecio	B2	Real	> 0
Altura del rectángulo, triángulo o trapecio	H	Real	> 0

Tabla 6-1. Análisis EPS de la aplicación figuras

#### Proceso

En caso de que la figura seleccionada es el círculo:

$$\text{Área}=\text{Pi}*\text{R}^2$$

$$\text{Perímetro}=2*\text{Pi}*\text{R}$$

En caso de que la figura seleccionada es el rectángulo:

$$\text{Área}=\text{B1}*\text{H}$$

$$\text{Perímetro}=2*(\text{B1}+\text{H})$$

En caso de que la figura seleccionada es el triángulo rectángulo:

$$\text{Área}=\text{B1}*\text{H}/2$$

$$\text{Perímetro}=\text{B1}+\text{H}+\text{raíz}(\text{B1}^2+\text{H}^2)$$

En caso de que la figura seleccionada es el triangulo rectángulo:

$$\text{Área}=(\text{B2}-\text{B1})*\text{H}/2$$

$$\text{Perímetro}=\text{B1}+\text{B2}+2*\text{raíz}(((\text{B2}-\text{B1})/2)^2+\text{H}^2)$$

Salida:

Mostrar la figura seleccionada, los valores de los parámetros de entrada, el área y el perímetro de la misma.

### **Actividades a realizar.**

1. Ejecute el Visual Basic 6.0, seleccione “*Exe estándar*” en el cuadro de diálogo “*Nuevo Proyecto*”.
2. Coloque las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Práctica # 6. Computación II
Forecolor	A su gusto
Height	7000
StartPosition	2 - CenterScreen
Width	10500

Tabla 6-2. Propiedades del formulario de la aplicación figuras

3. Agregue al formulario 6 etiquetas (Label), 4 cuadros de texto (TextBox), 3 botones (CommandButton), un cuadro de imagen (Image) y un contenedor de

controles (Frame) sobre el cual va a colocar 4 botones de opciones (OptionButton), aproximadamente como se muestra en la figura 6-1.

4. Descargue las imágenes a usar en el proyecto (circulo, rectángulo, triangulo y trapecio) y almacénelas en la carpeta del proyecto:

<http://webdelprofesor.ula.ve/economia/angelz/imagenes/circulo.jpg>

<http://webdelprofesor.ula.ve/economia/angelz/imagenes/rectangulo.jpg>

<http://webdelprofesor.ula.ve/economia/angelz/imagenes/triangulo.jpg>

<http://webdelprofesor.ula.ve/economia/angelz/imagenes/trapecio.jpg>

Práctica # 6. Computación II

**Área y Perímetro de un círculo, un rectángulo, un triángulo rectángulo o un trapecio**

Figura Geométrica

Círculo

Rectángulo

Triángulo Rectángulo

Trapecio

Radio :

Base :

Base mayor :

Altura :

Resultados:

Figura 6-1. Formulario de la Aplicación figuras

5. Fije las siguientes propiedades a los controles:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label0
	Caption	Área y Perímetro de un círculo, un rectángulo, un triángulo rectángulo o un trapecio
	Width, Height, Top y Left	8100, 830, 120 y 950
	Alignment	2 - Center
	BorderStyle	1 – Fixed Single
	Font	Arial, Negrita cursiva, 16
Etiqueta 2	(Nombre)	Label1
	Caption	Radio :
	Width, Height, Top y Left	1100, 250, 1500 y 3550
	Font	Arial, Normal, 9
	Visible	False
Etiqueta 3	(Nombre)	Label2
	Caption	Base :
	Width, Height, Top y Left	1100, 250, 2000 y 3550
	Font	Arial, Normal, 9
	Visible	False
Etiqueta 4	(Nombre)	Label3
	Caption	Base mayor :
	Width, Height, Top y Left	1100, 250, 2500 y 3550
	Font	Arial, Normal, 9
	Visible	False
Etiqueta 5	(Nombre)	Label4
	Caption	Altura :
	Width, Height, Top y Left	1100, 250, 3000 y 3550
	Font	Arial, Normal, 9
	Visible	False
Etiqueta 6	(Nombre)	Label5
	Caption	Resultados :
	Width, Height, Top y Left	5800, 1900, 3700 y 300
	Font	Arial, Normal, 11
	BorderStyle	1 – Fixed Single
	Visible	True
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	1250, 350, 1500 y 4800
	Font	Arial, Normal, 10
	Visible	False
Cuadro de texto 2	(Nombre)	Text2
	Text	




	Width, Height, Top y Left	1250, 350, 2000 y 4800
	Font	Arial, Normal, 10
	Visible	False
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	1250, 350, 2500 y 4800
	Font	Arial, Normal, 10
	Visible	False
Cuadro de texto 4	(Nombre)	Text4
	Text	
	Width, Height, Top y Left	1250, 350, 3000 y 4800
	Font	Arial, Normal, 10
	Visible	False
Botón 1	(Nombre)	Command1
	Caption	&Calcular rea y perímetro
	Width, Height, Top y Left	1750, 500, 5800 y 4000
Botón 2	(Nombre)	Command2
	Caption	&Limpiar
	Width, Height, Top y Left	1750, 500, 5800 y 6000
Botón 3	(Nombre)	Command3
	Caption	&Salir
	Width, Height, Top y Left	1750, 500, 5800 y 8000
Imagen 1	(Nombre)	Image1
	Width, Height, Top y Left	3500, 3200, 1200 y 6300
Contenedor 1	(Nombre)	Frame1
	Caption	Figura Geométrica
	Width, Height, Top y Left	2800, 1900, 1300 y 250
Opción 1	(Nombre)	Option1
	Caption	Circulo
	Width, Height, Top y Left	2200, 250, 350 y 250
	Value	False
	Index	0
Opción 2	(Nombre)	Option1
	Caption	Rectángulo
	Width, Height, Top y Left	2200, 250, 700 y 250
	Value	False
	Index	1
Opción 3	(Nombre)	Option1
	Caption	Triangulo Rectángulo
	Width, Height, Top y Left	2200, 250, 1050 y 250
	Value	False

	Index	2
Opción 4	(Nombre)	Option1
	Caption	Trapecio
	Width, Height, Top y Left	2200, 250, 1400 y 250
	Value	False
	Index	3

Tabla 6-3. Propiedades de los controles del formulario de la aplicación figuras

Notas:

- Las propiedades Width, Height, Top y Left de los botones de opciones es con respecto al control Frame1.
  - Los cuatro botones de opciones tienen el mismo nombre pero cambia la propiedad Index. Esto significa que son una matriz de controles (array) y que tiene cuatro elementos: Option(0) es el círculo, Option(1) es el rectángulo, etc.
6. Ya se tiene listo la interfase de usuario de la aplicación. Se procede ahora a almacenar el proyecto igual como se ha hecho en las prácticas anteriores. Recuerde crear una carpeta para guardar el proyecto, la cual podría ser: Practica 6 – Comp II.
7. Invoque al editor de código presionando ver código  en la parte superior derecha del explorador de proyecto y escriba las siguientes instrucciones:

```
Option Explicit
Dim CR As String
Const PI = 3.141516
```

La primera es para obligar a declarar todas las variables, la segunda es una variable global en la cual se almacenará el carácter de Retorno de carro y la tercera es la declaración de Pi como una constante.

8. A continuación escriba el módulo limpiar, el cual se utilizará para inicializar los controles de entrada de datos.

```
Private Sub Limpiar()
    Label1.Visible = False
```

```
Text1.Visible = False
Text1.Text = ""
Label2.Visible = False
Text2.Visible = False
Text2.Text = ""
Label3.Visible = False
Text3.Visible = False
Text3.Text = ""
Label4.Visible = False
Text4.Visible = False
Text4.Text = ""
Image1.Picture = Form1.Icon
Label5.Caption = "RESULTADOS"
End Sub
```

9. Se va a programar el evento Load del formulario. Este sucede en el momento en que se está cargando el formulario. Este evento es adecuado para inicializar variables, propiedades y objetos.

```
Private Sub Form_Load()
    Dim i As Integer

    For i = 0 To 3
        Option1(i).Value = False
    Next i

    Limpiar
    CR = Chr(10) & Chr(13)
End Sub
```

10. A continuación se programa el evento Click() de los botones de opciones, el cual sucede cuando se presiona con el ratón alguno de ellos. Dependiendo de la figura seleccionada, se mostrará los cuadros de texto necesarios para leer los parámetros de la figura geométrica seleccionada.

```
Private Sub Option1_Click(Index As Integer)
    Limpiar
    Select Case Index
        Case 0
            Label1.Visible = True
            Text1.Visible = True
```

```
        Image1.Picture = LoadPicture("circulo.jpg")
        Text1.SetFocus
    Case 1
        Label2.Visible = True
        Label2.Caption = "Base :"
        Text2.Visible = True
        Label4.Visible = True
        Text4.Visible = True
        Image1.Picture = LoadPicture("rectangulo.jpg")
        Text2.SetFocus
    Case 2
        Label2.Visible = True
        Label2.Caption = "Base :"
        Text2.Visible = True
        Label4.Visible = True
        Text4.Visible = True
        Image1.Picture = LoadPicture("triangulo.jpg")
        Text2.SetFocus
    Case 3
        Label2.Visible = True
        Label2.Caption = "Base menor:"
        Text2.Visible = True
        Label3.Visible = True
        Text3.Visible = True
        Label4.Visible = True
        Text4.Visible = True
        Image1.Picture = LoadPicture("trapecio.jpg")
        Text2.SetFocus
End Select
End Sub
```

11. Se hace ahora la validación de datos. Para evitar que el usuario ingrese caracteres no válidos para el tipo de dato. Para lo cual vamos a programar el evento LostFocus, el cual sucede cuando un control pierde el foco, es decir se selecciona otro.

```
Private Sub Text1_LostFocus()
    If Text1.Text <> "" Then
        If Not IsNumeric(Text1.Text) Then
            MsgBox "Debe ser numérico", vbOKOnly, "Error de datos"
            Text1.Text = ""
            Text1.SetFocus
        End If
    End If
End Sub
```

```

Else
    If Text1.Text <= 0 Then
        MsgBox "Debe ser mayor que cero", vbOKOnly, "Error de datos"
        Text1.Text = ""
        Text1.SetFocus
    End If
End If
End If
End Sub

```

**Nota:** Solo está reflejado para el cuadro de texto que representa el radio. El estudiante tendrá que programar este evento para los otros 3 cuadros de texto.

12. Ya validados los datos se va a programar evento click() de los 3 botones de ordenes:

```

Private Sub Command1_Click() ' Calcular área y perímetro

    Dim r As Single, b1 As Single, b2 As Single, h As Single
    Dim area As Single, perim As Single

    If Option1(0) Then ' Se selecciono el circulo
        If Text1.Text = "" Then
            Text1.SetFocus
            Exit Sub
        Else
            r = Text1.Text
            area = PI * r ^ 2
            perim = 2 * PI * r
            Label5 = "RESULTADOS" & CR & CR & "CIRCULO" & CR & "Radio = " &
r & CR
            Label5 = Label5 & "Area = " & area & CR & "Perímetro = " & perim
        End If
    ElseIf Option1(1) Then ' Se selecciono el rectangulo
        If Text2.Text = "" Then
            Text2.SetFocus
            Exit Sub
        ElseIf Text4.Text = "" Then
            Text4.SetFocus
            Exit Sub
        Else
            b1 = Text2.Text
            h = Text4.Text

```

```
        area = b1 * h
        perim = 2 * (b1 + h)
        Label5 = "RESULTADOS" & CR & CR & "RECTANGULO" & CR & "Base = "
& b1 & "    Altura = " & h & CR
        Label5 = Label5 & "Area = " & area & CR & "Perímetro = " & perim
    End If
    ElseIf Option1(2) Then          ' Se selecciono el triangulo
rectangulo
        If Text2.Text = "" Then
            Text2.SetFocus
            Exit Sub
        ElseIf Text4.Text = "" Then
            Text4.SetFocus
            Exit Sub
        Else
            b1 = Text2.Text
            h = Text4.Text
            area = b1 * h / 2
            perim = b1 + h + Sqr(b1 ^ 2 + h ^ 2)
            Label5 = "RESULTADOS" & CR & CR & "TRIANGULO RECTANGULO" & CR &
"Base = " & b1 & "    Altura = " & h & CR
            Label5 = Label5 & "Area = " & area & CR & "Perímetro = " & perim
        End If
    ElseIf Option1(3) Then          ' Se selecciono el trapecio
        If Text2.Text = "" Then
            Text2.SetFocus
            Exit Sub
        ElseIf Text3.Text = "" Then
            Text3.SetFocus
            Exit Sub
        ElseIf Text4.Text = "" Then
            Text4.SetFocus
            Exit Sub
        Else
            b1 = Text2.Text
            b2 = Text3.Text
            h = Text4.Text
            area = (b2 - b1) * h / 2
            perim = b1 + b2 + 2 * Sqr(((b2 - b1) / 2) ^ 2 + h ^ 2)
            Label5 = "RESULTADOS" & CR & CR & "TRAPECIO" & CR
            Label5 = Label5 & "Base menor = " & b1 & "    Base mayor = " & b2
& "    Altura = " & h & CR
            Label5 = Label5 & "Area = " & area & CR & "Perímetro = " & perim
        End If
    End If
```

```
End If

End Sub

Private Sub Command2_Click() ' Limpiar o inicializar los controles de
entrada de datos
    Dim i As Integer

    For i = 0 To 3
        Option1(i).Value = False
    Next i

    Limpiar
End Sub

Private Sub Command3_Click() ' Salir
    End
End Sub
```

13. Vuelva a guardar el proyecto.

14. Ejecute el programa con algunos datos de prueba y compruebe que funcione bien.

15. Como complemento de la práctica haga la validación de datos de la aplicación de la practica 4 (servicios publicos).

## Práctica 7. Procesando una encuesta.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Utilizar el control CheckBox.
2. Usar variables como contadores y acumuladores.
3. Construir una aplicación que permita procesar una encuesta.

La siguiente encuesta se aplicó a una muestra de N estudiantes de la FACES:

Sexo : (M / F)	Escuela : (A / C / E / S)
Trabaja : (S / N)	Usa el servicio de comedor : (S / N)
Número de Asignaturas inscritas en el semestre B- 2007: _____	
Retiró asignaturas en el B-2007: (S / N)	Número de Asignaturas: _____
Gastos promedio mensuales para su manutención en la ciudad: _____ Bs	

Se desea realizar un programa que procese la encuesta y calcule los siguientes estadísticos:

- a) Porcentaje de estudiantes masculinos y femeninos
- b) Porcentaje de estudiantes por escuela
- c) Porcentaje de estudiantes que trabajan
- d) Porcentaje de estudiantes que usan el servicio de comedor
- e) Porcentaje de estudiantes que retiraron asignatura en el B-2005
- f) Mínimo, máximo, promedio y varianza para la variable GASTO.

Realice validación de datos.



**Antes de comenzar.****Análisis E-P-S.****Entrada.**

Dato	Identificador	Tipo	Restricciones
Sexo del estudiante	Sexo	Entero	0=masculino, 1=femenino
Escuela a la que pertenece	Esc	Entero	0=Adm, 1=Con, 2=Eco, 3=Est
Trabaja	Trab	Lógica	-
Usa el servicio de comedor	UsaC	Lógica	-
Número de asignaturas inscritas en el semestre B-2005	NAI	Entero	> 0
Retiró Asignaturas en el B-2005	Ret	Lógica	-
Cuántas Retiró?	NAR	Entero	>=0
Gastos promedio para manutención en la ciudad	Gasto	Real	>0

Tabla 7-1. Análisis EPS de la aplicación "Encuesta Estudiantil FACES"

**Proceso**

- Para calcular los porcentajes se deben considerar varios contadores dependiendo del valor de la variable y luego dividirlo entre N
  - Si Sexo = masculino Sumar 1 a NMas
  - Si Sexo = femenino Sumar 1 a NFem
  
  - Si Esc = Adm Sumar 1 a NAdm
  - Si Esc = Con Sumar 1 a NCon
  - Si Esc = Eco Sumar 1 a NEco
  - Si Esc = Est Sumar 1 a NEst
  
  - Si Trabaja Sumar 1 a NTra
  - Si Usa comedor Sumar 1 a NUsa

Si Retiró Asignaturas Sumar 1 a NRet

- Para calcular el mínimo y máximo se debe comparar cada valor con el anterior e ir intercambiando con el último encontrado
- Para calcular el promedio se suma todos los valores GASTO y se divide entre N

Salida:

Mostrar los diferentes porcentajes y los estadísticos de la variable gastos

### **Actividades a realizar.**

1. Ejecute el Visual Basic 6.0, seleccione “Exe estándar” en el cuadro de dialogo “Nuevo Proyecto”.
2. Establezca las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Práctica # 7. Computación II
Forecolor	A su gusto
Height	7000
StartPosition	2 - CenterScreen
Width	10500

Tabla 7-2. Propiedades del formulario de la aplicación “Encuesta Estudiantil FACES”

3. Coloque los siguientes controles sobre el formulario aproximadamente como se muestra en la figura 7-1:
  - 5 etiquetas (Label)
  - 3 cuadros de texto (TextBox)
  - 3 botones de ordenes (CommandButton)
  - 2 contenedores de controles (Frame)
  - 6 botones de opciones (OptionButton)
  - 3 cuadros de verificación (CheckBox)

Recuerde dibujar los botones de opción (option button) sobre el respectivo contenedor de controles (frame) de tal manera que queden asociados a él.

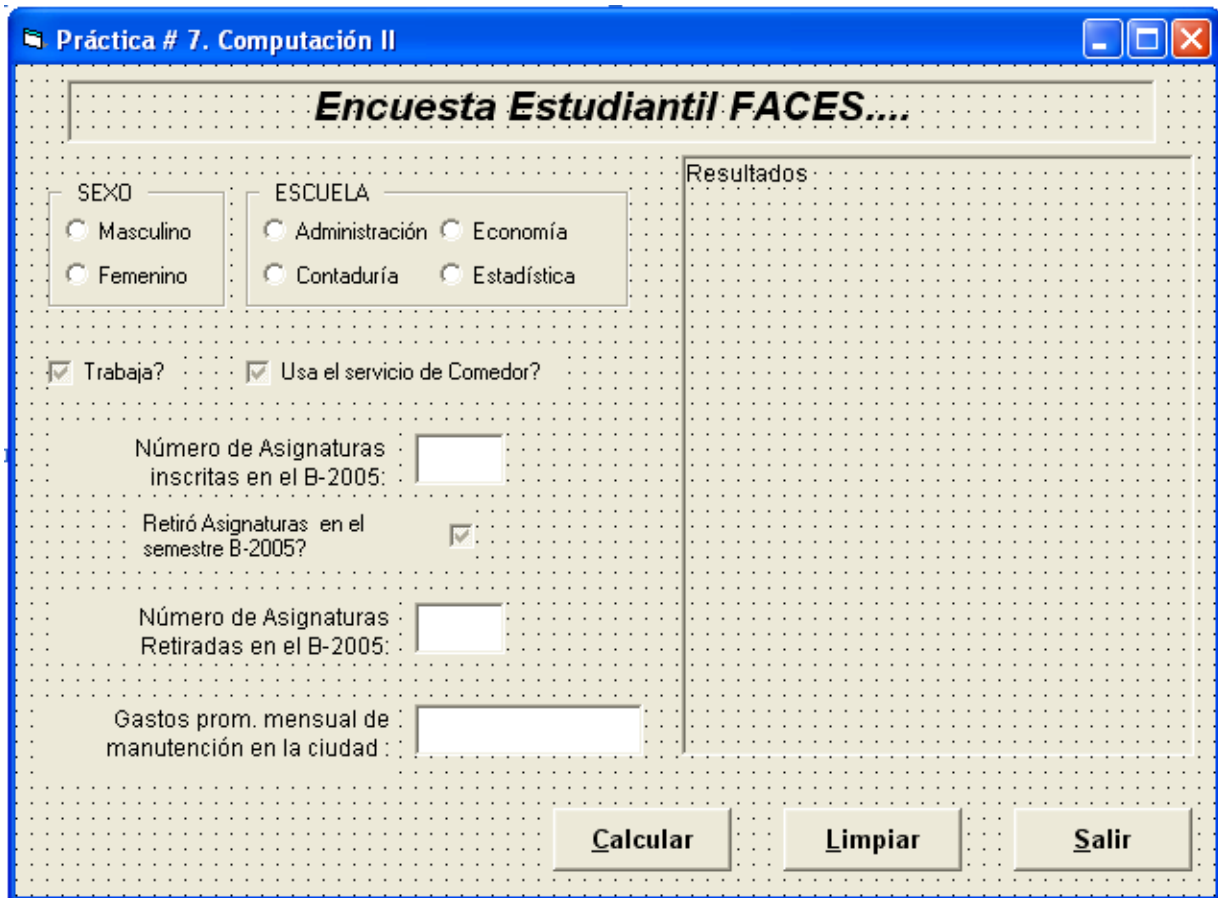


Figura 7-1. Formulario de la Aplicación “Encuesta Estudiantil FACES”

4. Establezca las siguientes propiedades a los controles:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Encuesta Estudiantil FACES....
	Width, Height, Top y Left	8500, 500, 120, 400
	Alignment	2 - Center
	BorderStyle	1 – Fixed Single
	Font	Arial, Negrita cursiva, 16
Etiqueta 2	(Nombre)	Label2
	Caption	Número de Asignaturas Inscritas en el B-2005 :
	Width, Height, Top y Left	2550, 420, 2800, 400
	Font	Arial, Normal, 9
Etiqueta 3	(Nombre)	Label3

	Caption	Número de Asignaturas retiradas en el B-2005 :
	Width, Height, Top y Left	2550, 420, 4200, 400
	Font	Arial, Normal, 9
Etiqueta 4	(Nombre)	Label4
	Caption	Gastos prom. mensual de manutención en la ciudad:
	Width, Height, Top y Left	2550, 420, 5000, 400
	Font	Arial, Normal, 9
Etiqueta 5	(Nombre)	Label5
	Caption	Resultados
	Width, Height, Top y Left	4000, 4700, 700, 5200
	Font	Arial, Normal, 9
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	700, 400, 2880, 3200
	Font	Arial, Normal, 10
Cuadro de texto 2	(Nombre)	Text2
	Text	700, 400, 4200, 3200
	Width, Height, Top y Left	
	Font	Arial, Normal, 10
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	1785, 400, 5000, 3200
	Font	Arial, Normal, 10
Botón 1	(Nombre)	Command1
	Caption	&Calcular
	Width, Height, Top y Left	1400, 500, 5800, 4200
	Font	Arial, Negrita, 10
Botón 2	(Nombre)	Command2
	Caption	&Limpiar
	Width, Height, Top y Left	Arial, Negrita, 10
	Font	1400, 500, 5800, 6000
Botón 3	(Nombre)	Command3
	Caption	&Salir
	Width, Height, Top y Left	1400, 500, 5800, 7800
	Font	Arial, Negrita, 10
Contenedor 1	(Nombre)	Frame1
	Caption	SEXO
	Width, Height, Top y Left	1400, 1000, 900, 250
Contenedor 2	(Nombre)	Frame1
	Caption	ESCUELA
	Width, Height, Top y Left	1400, 1000, 900, 1800
Opción 1	(Nombre)	Option1

	Caption	Masculino
	Width, Height, Top y Left	1100, 300, 250, 120
	Value	False
	Index	0
Opción 2	(Nombre)	Option1
	Caption	Femenino
	Width, Height, Top y Left	1100, 300, 600, 120
	Value	False
	Index	1
Opción 3	(Nombre)	Option2
	Caption	Aministración
	Width, Height, Top y Left	1400, 300, 250, 120
	Value	False
	Index	0
Opción 4	(Nombre)	Option2
	Caption	Contaduría
	Width, Height, Top y Left	1400, 300, 600, 120
	Value	False
	Index	1
Opción 5	(Nombre)	Option2
	Caption	Economía
	Width, Height, Top y Left	1400, 300, 250, 1500
	Value	False
	Index	2
Opción 6	(Nombre)	Option2
	Caption	Estadística
	Width, Height, Top y Left	1400, 300, 600, 1500
	Value	False
	Index	3
Verificación 1	(Nombre)	Check1
	Caption	Trabaja?
	Alignment	0 – Left Justify
	Width, Height, Top y Left	1000, 400, 2200, 250
	Value	2
Verificación 2	(Nombre)	Check2
	Caption	Usa el servicio de Comedor?
	Alignment	0 – Left Justify
	Width, Height, Top y Left	2500, 400, 2200, 250
	Value	2
Verificación 3	(Nombre)	Check3
	Caption	Retiró Asignaturas en el B-2005?
	Alignment	1 – Right Justify

	Width, Height, Top y Left	2500, 400, 2900, 1800
	Value	2

Tabla 7-3. Propiedades de los controles del formulario de la aplicación “Encuesta Estudiantil FACES”

#### Notas:

- Las propiedades de posición de los botones de opción es con respecto al control Frame.
  - Los botones de opción 1 y 2 que están contenidos en el frame 1 forman una matriz de controles con dos elementos. Lo mismo sucede con los botones de opción 3, 4, 5 y 6 los cuales están contenidos en el frame 2.
5. Inserte un módulo al proyecto en el cual se va a declarar las variables globales necesarias para realizar el procesamiento estadístico:

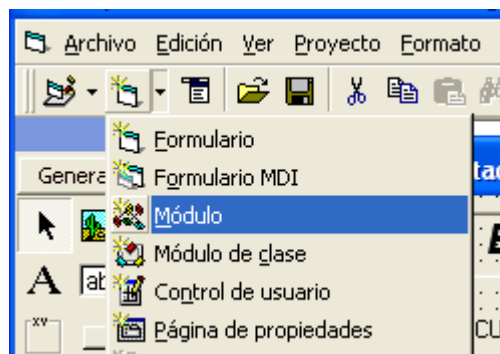


Figura 7-2. Insertando un módulo al proyecto

6. En la ventana del módulo agregue las siguientes declaraciones:

```
' Declaración de variables Globales
```

```
Global Enter As String
```

```
Global N As Integer, Mas As Integer, Fem As Integer
```

```
Global Adm As Integer, Con As Integer, Eco As Integer, Est As Integer
```

```
Global Tra As Integer, Com As Integer, Retiro As Integer
```

```
Global Min As Single, Max As Single, S As Single, S2 As Single
```

7. Cierre la ventana del módulo y agregue los siguientes procedimientos en la ventana de código, los cuales tienen que ver con la inicialización de las variables y controles:

```
Option Explicit

Private Sub Inicializar_Controles()
    '
    ' Inicializar Controles
    '
    Option1(0).Value = False
    Option1(1).Value = False
    Option2(0).Value = False
    Option2(1).Value = False
    Option2(2).Value = False
    Option2(3).Value = False
    Check1.Value = 2
    Check2.Value = 2
    Check3.Value = 2
    Text2.Enabled = False
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
End Sub

Private Sub Inicializar_Variables()
    '
    ' Inicializar Variables Globales
    '
    N = 0
    Mas = 0
    Fem = 0
    Adm = 0
    Con = 0
    Eco = 0
    Est = 0
    Tra = 0
    Com = 0
    Retiro = 0

    Min = 1E+21
    Max = 0
    S = 0
    S2 = 0
End Sub

Private Sub Form_Load()
```

```
Inicializar_Controles
Inicializar_Variables
Enter = Chr(10) & Chr(13)
End Sub
```

8. A continuación, programe los procedimientos de evento “change” de los cuadros de texto, los cuales tienen que ver con la validación de los datos que se introducen en las cajas de texto (variables, Número de materias inscritas, Cuantas materias retiró y los gastos de manutención):

```
Private Sub Text1_Change()
    If Text1.Text <> "" Then
        If Not IsNumeric(Text1.Text) Then
            MsgBox "Debe ser numérico", vbOKOnly, "Error de datos"
            Text1.Text = ""
            Text1.SetFocus
        Else
            If Text1.Text <= 0 Then
                MsgBox "Debe ser mayor que cero", vbOKOnly, "Error de datos"
                Text1.Text = ""
                Text1.SetFocus
            End If
        End If
    End If
End Sub
```

```
Private Sub Text2_Change()
    If Text2.Text <> "" Then
        If Not IsNumeric(Text2.Text) Then
            MsgBox "Debe ser numérico", vbOKOnly, "Error de datos"
            Text2.Text = ""
            Text2.SetFocus
        Else
            If Text2.Text <= 0 Then
                MsgBox "Debe ser mayor que cero", vbOKOnly, "Error de datos"
                Text2.Text = ""
                Text2.SetFocus
            End If
        End If
    End If
End Sub
```



```

End Sub

Private Sub Text3_Change()
    If Text3.Text <> "" Then
        If Not IsNumeric(Text3.Text) Then
            MsgBox "Debe ser numérico", vbOKOnly, "Error de datos"
            Text3.Text = ""
            Text3.SetFocus
        Else
            If Text3.Text <= 0 Then
                MsgBox "Debe ser mayor que cero", vbOKOnly, "Error de datos"
                Text3.Text = ""
                Text3.SetFocus
            End If
        End If
    End If
End Sub

End Sub

```

9. A continuación programa el procedimiento de evento “click” del cuadro de verificación 3. En caso de que se seleccione (si retiró asignaturas), se activa el cuadro de texto de la variable Cuantas retiró:

```

Private Sub Check3_Click()
    If Check3.Value = 1 Then
        Text2.Enabled = True
        Text2.SetFocus
    Else
        Text1.Enabled = False
    End If
End Sub

```

10. Ahora programe los procedimientos de evento “click” para los tres botones de comando:

```

Private Sub Command1_Click()
    Dim NMI As Integer, NMSem As Integer, Gasto As Single
    Dim PM As Single, Pf As Single, PAdm As Single, PCon As Single
    Dim PEco As Single, PEst As Single, PTra As Single, PCom As Single,
    PRet As Single
    Dim Pro As Single, Vari As Single

```

```
Dim cad As String

' Lectura y Validación de los datos

If (Option1(0).Value + Option1(1).Value) = 0 Then
    MsgBox "Error... Seleccione un valor para la variable SEXO",
vbOKOnly, "Error"
    Exit Sub
End If

If (Option2(0).Value + Option2(1).Value + Option2(2).Value +
Option2(3).Value) = 0 Then
    MsgBox "Error... Seleccione un valor para la variable ESCUELA",
vbOKOnly, "Error"
    Exit Sub
End If

If Check1.Value = 2 Then
    MsgBox "Error... Seleccione un valor para la variable TRABAJA",
vbOKOnly, "Error"
    Exit Sub
End If

If Check2.Value = 2 Then
    MsgBox "Error... Seleccione un valor para la variable USA COMEDOR",
vbOKOnly, "Error"
    Exit Sub
End If

If Check3.Value = 2 Then
    MsgBox "Error... Seleccione un valor para la variable CURSÓ
INTENSIVO", vbOKOnly, "Error"
    Exit Sub
End If

If Check3.Value = 1 Then
    If Text1.Text = "" Then
        Text1.SetFocus
        Exit Sub
    Else
        NMI = Text1.Text
    End If
End If

If Text2.Text = "" Then
```

```
Text2.SetFocus
Exit Sub
Else
  NMSEM = Text2.Text
End If

If Text3.Text = "" Then
  Text3.SetFocus
  Exit Sub
Else
  Gasto = Text3.Text
End If

'
' Procesamiento de la Encuesta
'
N = N + 1
If Option1(0) Then
  Mas = Mas + 1
Else
  Fem = Fem + 1
End If

If Option2(0) Then
  Adm = Adm + 1
Else
  If Option2(1) Then
    Con = Con + 1
  Else
    If Option2(2) Then
      Eco = Eco + 1
    Else
      Est = Est + 1
    End If
  End If
End If

If Check1 Then
  Tra = Tra + 1
End If

If Check2 Then
  Com = Com + 1
End If
```

```
If Check3 Then
  Retiro = Retiro + 1
End If

' Estadísticas Básicas

If Gasto < Min Then
  Min = Gasto
End If

If Gasto > Max Then
  Max = Gasto
End If

S = S + Gasto
S2 = S2 + Gasto ^ 2

PM = Mas / N * 100
Pf = Fem / N * 100
PAdm = Adm / N * 100
PCon = Con / N * 100
PEco = Eco / N * 100
PEst = Est / N * 100
PTra = Tra / N * 100
PCom = Com / N * 100
PRet = Retiro / N * 100

Pro = S / N
If N > 1 Then
  Vari = (S2 - N * Pro ^ 2) / (N - 1)
End If

'
' Presentación de Resultados
'
cad = "Resultados" & Enter & Enter
cad = cad & "Porcentaje Masculinos = " & PM & "%" & Enter
cad = cad & "Porcentaje Femeninos = " & Pf & "%" & Enter
cad = cad & "Porcentaje Administración = " & PAdm & "%" & Enter
cad = cad & "Porcentaje Contaduría = " & PCon & "%" & Enter
cad = cad & "Porcentaje Economía = " & PEco & "%" & Enter
cad = cad & "Porcentaje Estadística = " & PEst & "%" & Enter
```

```
cad = cad & "Porcentaje que trabajan = " & PTra & "%" & Enter
cad = cad & "Porcentaje usa comedor = " & PCom & "%" & Enter
cad = cad & "Porcentaje De Estudiantes con retiro = " & PRet & "%" &
Enter

Label5.Caption = cad & Enter

cad = ""
cad = cad & "Mínimo   " & Min & Enter
cad = cad & "Máximo   " & Max & Enter
cad = cad & "Promedio " & Pro & Enter
cad = cad & "Varianza  " & Vari

Label5.Caption = Label5.Caption & cad

Inicializar_Controles

End Sub

Private Sub Command2_Click()
    Inicializar_Controles
    Inicializar_Variables
    Label5.Caption = ""

End Sub

Private Sub Command3_Click()
    End
End
```

11. Guarde el proyecto recordando que debe crear una carpeta exclusiva para todos los archivos que se generan.
12. Ejecute la aplicación y compruebe el funcionamiento con algunos datos.

## Práctica 8. Usando varios formularios.

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Usar varios formularios en una aplicación.
2. Usar los control Shape y ComboBox.
3. Usar arreglos para almacenar información recogida de los controles y luego procesar estos datos en lote.

Realizar un programa que procese la siguiente encuesta utilizando arreglos para almacenar los datos de la misma:

- Sexo : (Masculino / Femenino)
- Nivel Educativo : (Primara / Secundaria / Universitaria)
- Ocupación : (Trabaja / Estudia / Desempleado)
- Votará en las venideras elecciones regionales : (Si / No)
- Candidato a gobernador : XXXX  
YYY  
ZZZZ  
WWW
- Candidato a alcalde: AAAA  
BBBB  
CCCC  
DDDD

Se desea que Usted realice un programa que procese la encuesta y calcule:

- a) Porcentaje de votantes masculinos y femeninos

- b) Porcentaje de votantes por nivel educativo
- c) Porcentaje de votantes por ocupación
- d) Porcentaje de votantes que votaran en las elecciones
- e) Candidato a gobernador favorecido en las encuestas
- f) Candidato a alcalde favorecido en las encuestas

### **Actividades a realizar:**

1. Ejecute el Visual Basic 6.0, seleccione “*Exe estándar*” en el cuadro de dialogo “*Nuevo Proyecto*”.
2. Establezca las siguientes propiedades al formulario:

<b>Propiedad</b>	<b>Valor</b>
Caption	Practica 8. Encuesta
Forecolor	A su gusto
Height	6200
StartPosition	2 - CenterScreen
Width	8200

Tabla 8-1. Propiedades del formulario de la aplicación “Encuesta elecciones”

3. Coloque los siguientes controles sobre el formulario aproximadamente como se muestra en la figura 8-1:
  - 7 etiquetas (Label)
  - 1 rectángulo (Shape)
  - 2 botones de opciones (OptionButton)
  - 4 botones de ordenes (CommandButton)
  - 1 caja de verificación (CheckBox)
  - 4 listas combinadas (ComboBox)

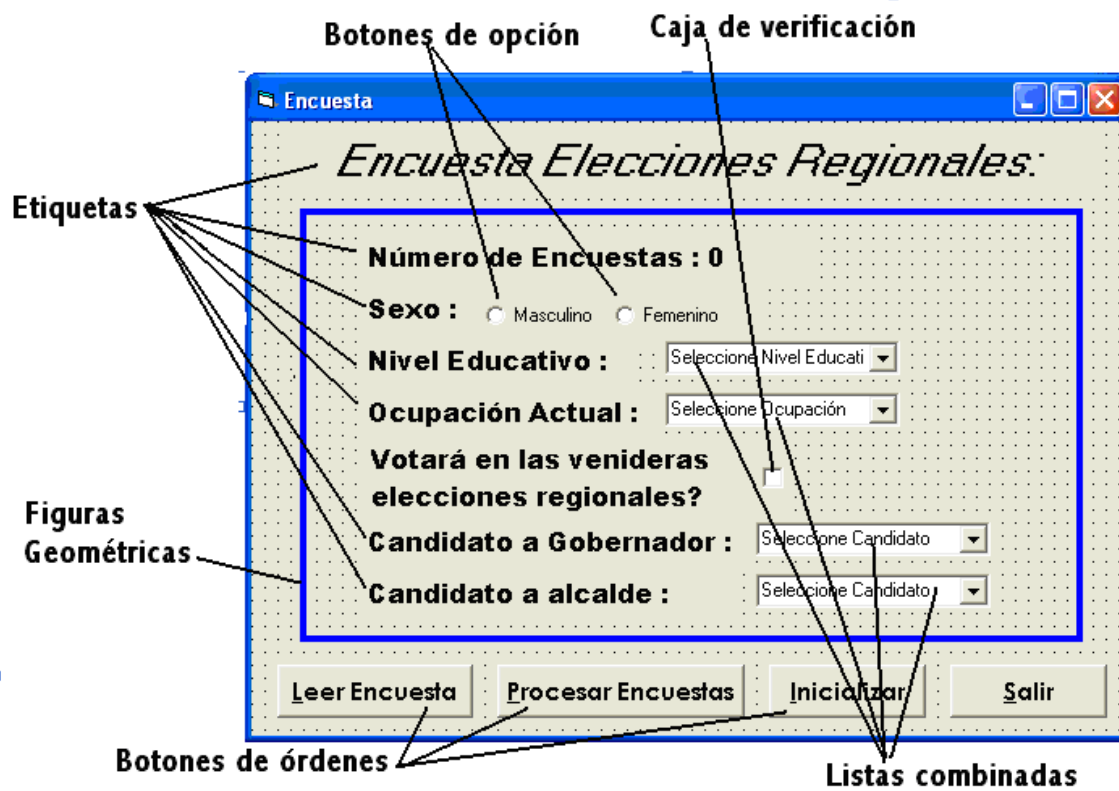


Figura 8-1. Formulario de la aplicación “Encuesta elecciones”

4. Establezca las siguientes propiedades a los controles:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Encuesta Elecciones Regionales:
	Width, Height, Top y Left	7500, 500, 120, 360
	Alignment	2 - Center
	Font	Arial, Negrita cursiva, 18
Etiqueta 2	(Nombre)	Label2
	Caption	Número de Encuestas:
	Width, Height, Top y Left	4100, 375, 1080, 1080
	Alignment	0 - Left
	Font	Arial Black, Normal, 12
Etiqueta 3	(Nombre)	Label3
	Caption	Sexo
	Width, Height, Top y Left	1100, 375, 1560, 1080
	Alignment	0 - Left
	Font	Arial Black, Normal, 12
Etiqueta 4	(Nombre)	Label4



	Caption	Nivel Educativo:
	Width, Height, Top y Left	2400, 375, 2040, 1080
	Alignment	0 - Left
	Font	Arial Black, Normal, 12
Etiqueta 5	(Nombre)	Label5
	Caption	Ocupación Actual:
	Width, Height, Top y Left	2600, 375, 2520, 1080
	Alignment	0 - Left
Etiqueta 6	Font	Arial Black, Normal, 12
	(Nombre)	Label6
	Caption	Candidato a Gobernador:
	Width, Height, Top y Left	3500, 375, 3720, 1080
Etiqueta 7	Alignment	0 - Left
	Font	Arial Black, Normal, 12
	(Nombre)	Label7
	Caption	Candidato a Alcalde:
Rectángulo	Width, Height, Top y Left	3400, 375, 4200, 1080
	Alignment	0 - Left
	Font	Arial Black, Normal, 12
	(Nombre)	Shape1
	Shape	0 - Rectangle
	Width, Height, Top y Left	7200, 4000, 850, 500
Opción 1	BorderColor	Escojalo
	BorderStyle	1 - Solid
	BorderWidth	4
	(Nombre)	Option1
Opción 2	Caption	Masculino
	Width, Height, Top y Left	1300, 255, 1700, 2000
	Value	False
	(Nombre)	Option2
Botón 1	Caption	Femenino
	Width, Height, Top y Left	1300, 255, 1700, 3300
	Value	False
	(Nombre)	Command1
Botón 2	Caption	&Leer Encuesta
	Width, Height, Top y Left	1800, 500, 5000, 250
	Font	Arial, Negrita, 10
	(Nombre)	Command2
Botón 2	Caption	&Procesar Encuesta
	Width, Height, Top y Left	Arial, Negrita, 10
	Enabled	False
	Font	2300, 500, 5000, 2300

Botón 3	(Nombre)	Command3
	Caption	&Inicializar
	Width, Height, Top y Left	1500, 500, 5000, 4800
	Font	Arial, Negrita, 10
Botón 4	(Nombre)	Command4
	Caption	&Salir
	Width, Height, Top y Left	1500, 500, 5000, 6500
	Font	Arial, Negrita, 10
Verificación 1	(Nombre)	Check1
	Caption	Votará en las venideras elecciones regionales?
	Alignment	1 – Right Justify
	Width, Height, Top y Left	3900, 600, 3000, 1080
	Font	Arial Black, Normal, 12
	Value	2
Lista combinada 1	(Nombre)	Combo1
	Text	Seleccione Nivel Educativo
	Width, Height, Top y Left	2200, 325, 2000, 3800
	List	Primaria Secundaria Universitaria
Lista combinada 2	(Nombre)	Combo2
	Text	Seleccione Ocupación
	Width, Height, Top y Left	2200, 325, 2500, 3800
	List	Trabaja Estudia Desempleado
Lista combinada 3	(Nombre)	Combo3
	Text	Seleccione Candidato
	Width, Height, Top y Left	2200, 325, 3700, 4600
	List	XXXXX YYYYY ZZZZZ WWWWW TTTTT

Lista combinada 4	(Nombre)	Combo4
	Text	Seleccione Candidato
	Width, Height, Top y Left	2200, 325, 4200, 4600
	List	AAAAA BBBBB CCCCC DDDDD EEEEEE

Tabla 8-2. Propiedades de los controles del formulario de la aplicación "Encuesta elecciones"

5. Agregue un nuevo formulario a la aplicación:

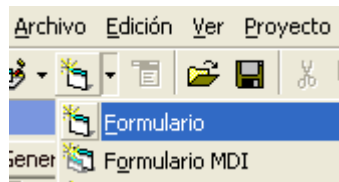


Figura 8-2. Insertar el segundo Formulario a la aplicación

6. Establezca las siguientes propiedades al formulario 2:

Propiedad	Valor
Caption	Resultados Practica 8
Forecolor	A su gusto de la paleta de colores
Height	6500
StartPosition	2 - CenterScreen
Width	8200

Tabla 8-3. Propiedades del formulario 2 de la aplicación "Encuesta elecciones"

7. Coloque los siguientes controles sobre el formulario 2 aproximadamente como se muestra en la figura 8-3:

- 2 etiquetas (Label)
- 1 botón de orden (CommandButton)

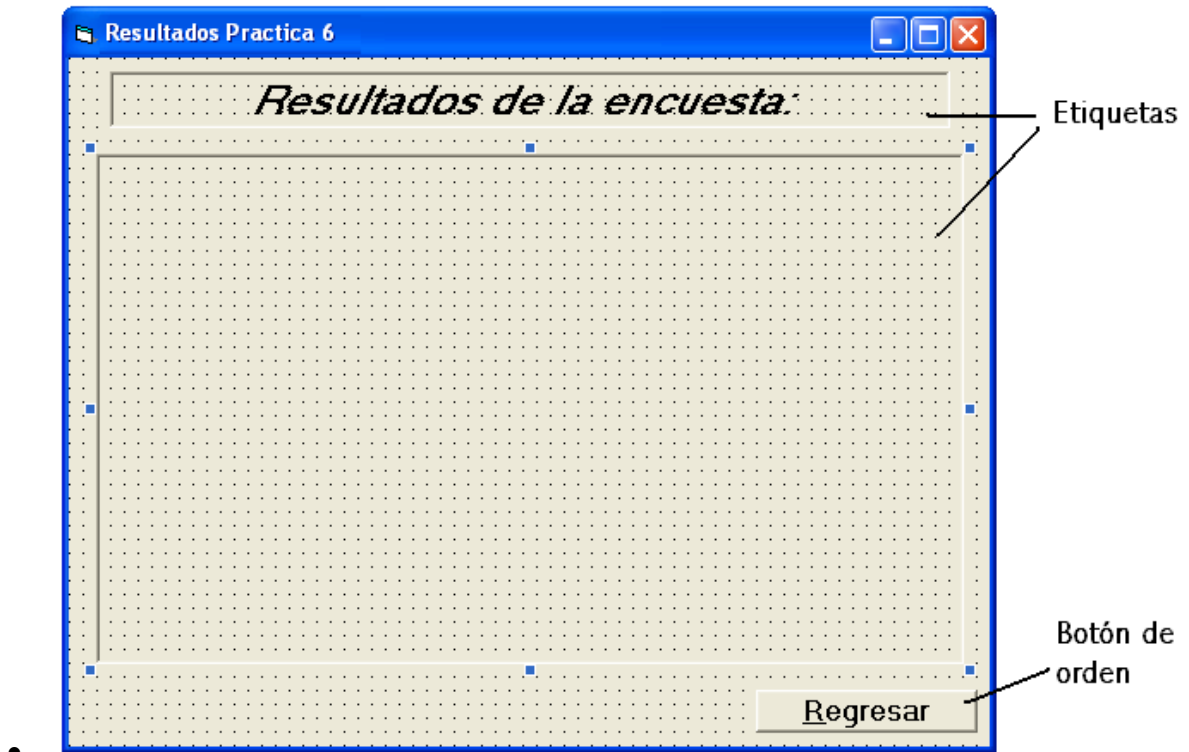


Figura 8-3. Formulario 2 de la aplicación

8. Establezca las siguientes propiedades a los controles:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Resultados de la Encuesta
	Width, Height, Top y Left	7300, 500, 120, 350
	Alignment	2 - Center
	Font	Arial, Negrita cursiva, 18
Etiqueta 2	(Nombre)	Label2
	Caption	
	Width, Height, Top y Left	7600, 4500, 850, 250
	Alignment	0 - Left
	Font	Arial, Normal, 11
	Appearance	1 - 3D
Botón 1	(Nombre)	Command1
	Caption	&Regresar
	Width, Height, Top y Left	2000, 375, 5500, 6000
	Font	Arial, Negrita, 10

Tabla 8-4. Propiedades de los controles del Formulario 2 de la aplicación

9. Inserte un módulo al proyecto en el SE VAN a declarar las variables globales necesarias para realizar el procesamiento estadístico.

10. En la ventana del módulo agregue las siguientes declaraciones:

```
Option Explicit
Option Base 1

'
' Declaración de las variables ARREGLOS que se usaran a nivel global
' para almacenar los datos de la encuesta
'
Global N As Integer
Global Sexo(100) As Byte
Global Nivel(100) As Byte
Global Ocupa(100) As Byte
Global Vota(100) As Byte
Global Gober(100) As Byte
Global Alcalde(100) As Byte

Public Sub Procesar_Encuesta()
    Dim Enter As String
    Dim I As Integer
    Dim Cad As String
    Dim PMasc As Single, PFem As Single
    Dim PNivel(3) As Single
    Dim POcupa(3) As Single
    Dim PVota As Single
    Dim PGober(5) As Single
    Dim Palcalde(5) As Single

    '
    ' Procesamiento de cada encuesta
    '

    Form2.Label2.Caption = ""

    For I = 1 To N
        If Sexo(I) = 0 Then
            PMasc = PMasc + 1
        Else
            PFem = PFem + 1
        End If
        PNivel(Nivel(I) + 1) = PNivel(Nivel(I) + 1) + 1
        POcupa(Ocupa(I) + 1) = POcupa(Ocupa(I) + 1) + 1
    Next I
End Sub
```

```

    If Vota(I) = 1 Then
        PVota = PVota + 1
        PGober(Gober(I) + 1) = PGober(Gober(I) + 1) + 1
        Palcalde(Alcalde(I) + 1) = Palcalde(Alcalde(I) + 1) + 1
    End If
Next I

PMasc = PMasc / N * 100
PFem = PFem / N * 100
For I = 1 To 3
    PNivel(I) = PNivel(I) / N * 100
    POcupa(I) = POcupa(I) / N * 100
Next I
For I = 1 To 5
    PGober(I) = PGober(I) / PVota * 100
    Palcalde(I) = Palcalde(I) / PVota * 100
Next I
PVota = PVota / N * 100

Enter = Chr(10) & Chr(13)

'
' Mostrar resultados en la etiqueta
'
Cad = "Porcentaje Masculinos = " & PMasc & "%" & Enter
Cad = Cad & "Porcentaje Femenino = " & PFem & "%" & Enter & Enter

Cad = Cad & "Porcentaje Primaria = " & PNivel(1) & "%" & Enter
Cad = Cad & "Porcentaje Secundaria = " & PNivel(2) & "%" & Enter
Cad = Cad & "Porcentaje Universitaria = " & PNivel(3) & "%" & Enter & Enter

Form2.Label2.Caption = Cad

Cad = "Porcentaje que Trabaja = " & POcupa(1) & "%" & Enter
Cad = Cad & "Porcentaje que Estudia = " & POcupa(2) & "%" & Enter
Cad = Cad & "Porcentaje Desempleado = " & POcupa(3) & "%" & Enter & Enter

Cad = "Porcentaje Que votaran = " & PVota & "%" & Enter & Enter

Form2.Label2.Caption = Form2.Label2.Caption & Cad

Cad = "Porcentaje cand. Gober XXXXX = " & PGober(1) & "%" & Enter
Cad = Cad & "Porcentaje cand. Gober YYYYY = " & PGober(2) & "%" & Enter
Cad = Cad & "Porcentaje cand. Gober ZZZZZ = " & PGober(3) & "%" & Enter
Cad = Cad & "Porcentaje cand. Gober WWWW = " & PGober(4) & "%" & Enter
Cad = Cad & "Porcentaje cand. Gober TTTTT = " & PGober(5) & "%" & Enter &
Enter

```

```

Form2.Label2.Caption = Form2.Label2.Caption & Cad

Cad = "Porcentaje cand. Alcalde AAAAA = " & Palcalde(1) & "%" & Enter
Cad = Cad & "Porcentaje cand. Alcalde BBBBB = " & Palcalde(2) & "%" & Enter
Cad = Cad & "Porcentaje cand. Alcalde CCCCC = " & Palcalde(3) & "%" & Enter
Cad = Cad & "Porcentaje cand. Alcalde DDDDD = " & Palcalde(4) & "%" & Enter
Cad = Cad & "Porcentaje cand. Alcalde EEEEE = " & Palcalde(5) & "%" & Enter

Form2.Label2.Caption = Form2.Label2.Caption & Cad

End Sub

```

Cierre la ventana del módulo y agregue los siguientes procedimientos en la ventana de código (cuando invoque algún evento de un control hágalo dando doble clic en el respectivo control :

```

' Este procdimiento se utiliza para habilitar o deshabilitar los combos
que
' seleccionan a los candidatos dependiendo si Activa o no el CHECK1
Private Sub Check1_Click()
    If Check1 Then
        Combo3.Enabled = True
        Combo4.Enabled = True
    Else
        Combo3.Enabled = False
        Combo4.Enabled = False
    End If
End Sub

'
' Botón Salir
'
Private Sub Command1_Click()
    End
End Sub

'
' Botón Leer encuesta
'
Private Sub Command2_Click()
    '
    ' VALIDACIÓN de datos
    '

```

```
If Option1.Value + Option2.Value = 0 Then
    MsgBox "Error, seleccione sexo", vbOKOnly, "Error"
    Exit Sub
End If

If Combo1.ListIndex = -1 Then
    MsgBox "Error, seleccione Nivel Educativo", vbOKOnly, "Error"
    Exit Sub
End If

If Combo2.ListIndex = -1 Then
    MsgBox "Error, seleccione Ocupación", vbOKOnly, "Error"
    Exit Sub
End If

If Check1.Value = 2 Then
    MsgBox "Error, seleccione Si va a votar", vbOKOnly, "Error"
    Exit Sub
End If

If Check1 Then
    If Combo3.ListIndex = -1 Then
        MsgBox "Error, seleccione candidato a gobernador", vbOKOnly,
"Error"
        Exit Sub
    End If

    If Combo2.ListIndex = -1 Then
        MsgBox "Error, seleccione candidato a alcalde", vbOKOnly, "Error"
        Exit Sub
    End If
End If

'
' Guardar los datos en la siguiente posición de los arreglos
'
If N = 100 Then
    MsgBox "Error, Arreglos llenos", vbOKOnly, "Error"
    Exit Sub
Else
    Command2.Enabled = True
    N = N + 1
    If Option1 Then
        Sexo(N) = 0
    End If
End If
```



```
    Else
        Sexo(N) = 1
    End If
    Nivel(N) = Combo1.ListIndex
    Ocupa(N) = Combo2.ListIndex
    Vota(N) = Check1.Value
    If Check1 Then
        Gober(N) = Combo3.ListIndex
        Alcalde(N) = Combo4.ListIndex
    End If
End If

'
' Inicializar controles
'

Option1.Value = False
Option2.Value = False
Combo1.ListIndex = -1
Combo2.ListIndex = -1
Combo3.ListIndex = -1
Combo4.ListIndex = -1
Check1.Value = 2
Combo3.Enabled = False
Combo4.Enabled = False

End Sub

'
' Botón Procesar encuestas
'

Private Sub Command3_Click()
    Form1.Enabled = False
    Form2.Enabled = True
    Form2.Visible = True
    Call Procesar_Encuesta
End Sub

'
' Botón Inicializar
'

Private Sub Command4_Click()
    N = 0
    Label2.Caption = "Número de Encuestas : 0"
    Command2.Enabled = False
```

```
End Sub

Private Sub Form_Load()
    N = 0
    Option1.Value = False
    Option2.Value = False
    Combo1.ListIndex = -1
    Combo2.ListIndex = -1
    Combo3.ListIndex = -1
    Combo4.ListIndex = -1
    Check1.Value = 2
    Combo3.Enabled = False
    Combo4.Enabled = False
End Sub
```

En el formulario 2 inserte el siguiente código (evento click() del botón Regresar) para regresar al formulario 1:

```
'
' Botón regresar
'
Private Sub Command1_Click()
    Form1.Enabled = True
    Form2.Enabled = False
    Form2.Visible = False
End Sub
```

11. Guarde el proyecto recordando que debe crear una carpeta exclusiva para todos los archivos que se generan.
12. Ejecute la aplicación y verifique el funcionamiento de los dos formularios y los resultados.

## Practica 9. Uso del control MSFlexGrid

### Objetivos:

Al finalizar esta práctica el estudiante será capaz de:

1. Usar el control MSFlexGrid para mostrar y procesar una matriz.
2. Usar los Arreglos para guardar información
3. Procesar una encuesta para generar estadísticas básicas y calcular el modelo de regresión
4. Crear la siguiente aplicación:

Se desea que Usted realice un programa que procese la encuesta y:

- a) Calcule las Estadísticas Básicas (Media, Varianza, Mínimo y Máximo)
- b) Estime los parámetros de la Recta de Regresión entre el Peso y la Estatura

Encuesta
<ul style="list-style-type: none"> <li>• Edad: _____</li> <li>• Peso: _____</li> <li>• Estatura: _____</li> </ul>

### Antes de comenzar

Realizar el Análisis E-P-S de la Aplicación Encuesta.

### Entrada.

Dato	Identificador	Tipo	Restricciones
Edad del entrevistado	Edad	Real	>15
Peso del entrevistado	Peso	Real	>0
Estatura del entrevistado	Estatura	real	>0

Tabla 9-1. Análisis EPS de la aplicación "Encuesta".

## Proceso

- Cada vez que se lee un dato se incrementa el número de entrevistados (N)
- Para calcular el promedio se llevan tres acumuladores para sumar los datos de cada uno de los entrevistados

*Media :*

$$SEdad = \sum_{i=1}^n Edad(i)$$

$$SPeso = \sum_{i=1}^n Peso(i)$$

$$SEstatura = \sum_{i=1}^n Estatura(i)$$

$$MEdad = SEdad / N$$

$$MPeso = SPeso / N$$

$$MEstatura = SEstatura / N$$

*Varianza :*

$$S2Edad = \sum_{i=1}^n Edad(i)^2$$

$$S2Peso = \sum_{i=1}^n Peso(i)^2$$

$$S2Estatura = \sum_{i=1}^n Estatura(i)^2$$

$$VEdad = \frac{S2Edad - N * MEdad}{n - 1}$$

$$VPeso = \frac{S2Peso - N * MPeso}{n - 1}$$

$$VEstatura = \frac{S2Estatura - N * MEstatura}{n - 1}$$

Para el mínimo (y el máximo) se recorre el arreglo y se va comparando cada valor con el mínimo (máximo) y en caso de encontrar otro que sea menor (mayor) se cambia.

Regresión

$$SPesoEst = \sum_{i=1}^n Estatura(i) * Peso(i)$$

$$Beta1 = \frac{N * SPesoEst - SEstatura * Speso}{N * S2Estatura - SEstatura^2}$$

$$Beta0 = MPeso - Beta1 * MEstatura$$

## Salida

La media, la varianza, el mínimo y el máximo para cada variable

Los parámetros de la recta de regresión

Para la interfaz de usuario se usaran los siguientes controles:

- 3 etiquetas (label)
- 3 cajas de texto (TextBox)
- 4 botones de comando (CommandButton)
- 1 Cuadrícula Flexible (MSFlexGrid)

## Actividades a realizar:

9. Ejecute el Visual Basic 6.0, seleccione “Exe estándar” en el cuadro de dialogo “Nuevo Proyecto”.
10. Establezca las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Practica 9. Ms FlexGrid
Forecolor	A su gusto
Height	6200
StartPosition	2 - CenterScreen
Width	8600

Tabla 9-2. Propiedades del formulario 1 de la aplicación “Encuesta”.

Este formulario contiene en la parte izquierda los cuadros de texto para recoger los datos y guardarlos en tres arreglos (Edad, Peso y Estatura) y en la parte derecha el control MsFlexGrid que se utilizará para mostrar los datos que se han ido leyendo.

11. Coloque los siguientes controles sobre el formulario aproximadamente como se muestra en la figura 9-1:


- 3 etiquetas (Label)
- 3 cuadros de texto (TextBox)
- 4 botones de comando (CommandButton)
- 1 cuadrícula flexible de datos (MsFlexGrid-)



Figura 9-1. Formulario 1 de la aplicación “Encuesta”.

12. En la tabla 9-3 se presenta las propiedades de los controles que se agregaron al formulario:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Edad:
	Width, Height, Top y Left	2175, 375, 960, 240
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Etiqueta 2	(Nombre)	Label2
	Caption	Peso:

	Width, Height, Top y Left	2175, 375, 1800, 240
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Etiqueta 3	(Nombre)	Label3
	Caption	Estatura:
	Width, Height, Top y Left	2175, 375, 2640, 240
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	1250, 625, 840, 2600
	Font	Arial, Normal, 10
Cuadro de texto 2	(Nombre)	Text2
	Text	1250, 625, 1680, 2600
	Width, Height, Top y Left	
	Font	Arial, Normal, 10
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	1250, 625, 2520, 2600
	Font	Arial, Normal, 10
Botón 1	(Nombre)	Command1
	Caption	&Leer Dato
	Width, Height, Top y Left	1650, 500, 5000, 200
	Font	Arial, Negrita, 10
Botón 2	(Nombre)	Command2
	Caption	&Procesar
	Width, Height, Top y Left	1650, 500, 5000, 2200
	Font	Arial, Negrita, 14
Botón 3	(Nombre)	Command3
	Caption	&Inicializar
	Width, Height, Top y Left	1650, 500, 5000, 4200
	Font	Arial, Negrita, 14
Botón 4	(Nombre)	Command4
	Caption	&Salir
	Width, Height, Top y Left	1650, 500, 5000, 6200
	Font	Arial, Negrita, 14
Cuadrícula 1	(Nombre)	MsFlexGrid1
	Width, Height, Top y Left	4200, 4800, 100, 4200
	Font	Arial, Negrita, 14

Tabla 9-3. Propiedades de los controles del formulario1 de la aplicación “Encuesta”.

13. Inserte el Código del Formulario 1. Este código contiene, en primer lugar la inicialización de los controles de lectura, después una función para validar datos y luego el código de los 4 botones de comando.

```
Option Explicit

Public Sub Inicializar_controles()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
End Sub

' Función para validar los datos

Public Function Validar() As Boolean
    Dim X As Single

    Validar = True
    If Not IsNumeric(Text1.Text) Then
        MsgBox "Error... La Edad es numerica", vbOKOnly, "Error"
        Text1.SetFocus
        Validar = False
        Exit Function
    Else
        X = Text1.Text
        If X < 15 Then
            MsgBox "Error... La Edad es mayor o igual a 15", vbOKOnly,
"Error"
            Text1.SetFocus
            Validar = False
            Exit Function
        End If
    End If

    If Not IsNumeric(Text2.Text) Then
        MsgBox "Error... El Peso es numerico", vbOKOnly, "Error"
        Text2.SetFocus
        Validar = False
        Exit Function
    Else
        X = Text2.Text
        If X < 40 Then
```



```
        MsgBox "Error... El Peso es mayor o igual a 40", vbOKOnly,
"Error"
        Text2.SetFocus
        Validar = False
        Exit Function
    End If
End If

If Not IsNumeric(Text3.Text) Then
    MsgBox "Error... La Estatura es numerica", vbOKOnly, "Error"
    Text3.SetFocus
    Validar = False
    Exit Function
Else
    X = Text2.Text
    If X < 1.4 Then
        MsgBox "Error... La estatura es mayor o igual a 1.40",
vbOKOnly, "Error"
        Text3.SetFocus
        Validar = False
        Exit Function
    End If
End If

End Function

'
' Leer los datos y guardarlos en los arreglos

Private Sub Command1_Click()

    If Validar() Then
        ' Pasar los datos de cuadro de texto para los ARREGLOS
        N = N + 1
        If N = 1 Then
            Command3.Enabled = False
        Else
            If N = 2 Then
                Command2.Enabled = True
            End If
        End If
    End If

    Edad(N) = Text1.Text
    Peso(N) = Text2.Text
```

```
    Estatura(N) = Text3.Text

    ' Mostrar los datos en el MSFlexGrid
    MSFlexGrid1.Rows = N + 1
    MSFlexGrid1.Row = N
    MSFlexGrid1.Col = 0
    MSFlexGrid1.Text = N
    MSFlexGrid1.Col = 1
    MSFlexGrid1.Text = Edad(N)
    MSFlexGrid1.Col = 2
    MSFlexGrid1.Text = Peso(N)
    MSFlexGrid1.Col = 3
    MSFlexGrid1.Text = Estatura(N)

    Call Inicializar_controles
End If

End Sub

'
' Procesar
Private Sub Command2_Click()
    Call Estadisticas
    Form1.Enabled = False
    Form2.Visible = True
End Sub

'
' Inicializar los datos
Private Sub Command3_Click()
    N = 0
    MSFlexGrid1.Rows = 1

End Sub

'
' Salir
Private Sub Command4_Click()
    End
End Sub

Private Sub Form_Load()
    Dim i As Integer
```

```
' Fijar número de filas y columnas
MSFlexGrid1.Cols = 4
MSFlexGrid1.Rows = 2

' Fijar número de filas y columnas fijas o encabezados
MSFlexGrid1.FixedCols = 1
MSFlexGrid1.FixedRows = 1

' Tamaño (ancho) total de cada columna
MSFlexGrid1.Width = 4200
MSFlexGrid1.ColWidth(0) = 400
For i = 1 To 3
    MSFlexGrid1.ColWidth(i) = 1200
    MSFlexGrid1.ColAlignment(i) = 3      ' Alineación
Next i

' Encabezados de cada columna
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = " N° "
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = " Edad "
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = " Peso "
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = " Estatura "

Call Inicializar_controles

Command2.Enabled = False
Command3.Enabled = False

N = 0
End Sub
```

14. Almacene la aplicación tal y como se ha hecho en las prácticas anteriores.
15. Inserte un nuevo FORMULARIO (Form2). En este formulario se presentan los resultados obtenidos del programa (ver figura 9-2)
16. Establezca las siguientes propiedades al formulario 2:

Propiedad	Valor
Caption	Resultados de la Encuesta
Forecolor	A su gusto
Height	6400
StartPosition	2 - CenterScreen
Width	8500

Tabla 9-4. Propiedades del formulario 2 de la aplicación "Encuesta".



Figura 9-2. Formulario 2 de la aplicación "Encuesta".

17. Coloque sobre este formulario una etiqueta para mostrar los resultados y un Botón de ordenes para regresar con las siguientes propiedades:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Resultados:
	Width, Height, Top y Left	8100, 4500, 750, 120
	Alignment	0 – Left Justify
Botón 1	(Nombre)	Command1
	Caption	&Regresar
	Width, Height, Top y Left	2000, 400, 5400, 6000
	Font	Arial, Negrita, 10

Tabla 9-5. Propiedades de los controles del formulario 2 de la aplicación "Encuesta".

Observe que la fuente de la etiqueta 2 es la Courier New, la cual es igualmente espaciada, lo que permite que la salida se vea en forma tabular.

18. Coloque el Código del Formulario 2 para activar el formulario 1 una vez que se desee salir de los resultados.

```
Private Sub Command1_Click()  
    Form1.Enabled = True  
    Form2.Visible = False  
End Sub
```

19. Inserte un Módulo.

Aquí se declaran las variables globales: número de personas y los arreglos Edad, Peso y Estatura; y el procedimiento Estadísticas (botón procesar),

20. Coloque en el Módulo el siguiente Código:

```
Option Explicit  
  
Global N As Integer      ' Número de personas  
  
Global Edad(100) As Single  
Global Peso(100) As Single  
Global Estatura(100) As Single  
  
Public Sub Estadisticas()  
    Dim MEdad As Single, MPeso As Single, MEstatura As Single  
    Dim VEdad As Single, VPeso As Single, VEstatura As Single  
    Dim S2Edad As Single, S2Peso As Single, S2Estatura As Single  
    Dim MiEdad As Single, MiPeso As Single, MiEstatura As Single  
    Dim MaEdad As Single, MaPeso As Single, MaEstatura As Single  
    Dim I As Integer  
    Dim SPesEst As Single  
    Dim Beta1 As Single, Beta0 As Single  
    Dim Cad As String, Ent As String  
  
    Ent = Chr(10) & Chr(13)  
  
    '  
    ' Inicializar las sumas, Mínimos y Máximos  
    MEdad = 0  
    MPeso = 0
```

```
MEstatura = 0
SPesEst = 0
S2Edad = 0
S2Peso = 0
S2Estatura = 0
MiEdad = 1E+16
MiPeso = 1E+16
MiEstatura = 1E+16
MaEdad = -10
MaPeso = -10
MaEstatura = -10

For I = 1 To N
  ' Sumas
  MEdad = MEdad + Edad(I)
  MPeso = MPeso + Peso(I)
  MEstatura = MEstatura + Estatura(I)
  SPesEst = SPesEst + Peso(I) * Estatura(I)

  ' Sumas al cuadrado
  S2Edad = S2Edad + Edad(I) ^ 2
  S2Peso = S2Peso + Peso(I) ^ 2
  S2Estatura = S2Estatura + Estatura(I) ^ 2

  ' Mínimos y Máximos

  If MiEdad > Edad(I) Then
    MiEdad = Edad(I)
  End If
  If MiPeso > Peso(I) Then
    MiPeso = Peso(I)
  End If
  If MiEstatura > Estatura(I) Then
    MiEstatura = Estatura(I)
  End If
  If MaEdad < Edad(I) Then
    MaEdad = Edad(I)
  End If
  If MaPeso < Peso(I) Then
    MaPeso = Peso(I)
  End If
  If MaEstatura < Estatura(I) Then
```

```

        MaEstatura = Estatura(I)
    End If
Next I

'
' Coeficientes de Regresión
Beta1 = (N * SPesEst - MEstatura * MPeso) / (N * S2Estatura -
MEstatura ^ 2)
Beta0 = MPeso / N - Beta1 * MEstatura / N

'
' Medias y Varianzas
MEdad = MEdad / N
VEdad = (S2Edad - N * MEdad ^ 2) / (N - 1)
MPeso = MPeso / N
VPeso = (S2Peso - N * MPeso ^ 2) / (N - 1)
MEstatura = MEstatura / N
VEstatura = (S2Estatura - N * MEstatura ^ 2) / (N - 1)

' Aquí se muestra los resultados en la etiqueta 2 del formulario 2

Cad = "; Variable ; Mínimo ; Máximo ; Media ; Varianza ;" &
Ent
Cad = Cad & "-----+-----+-----+-----+-----"
---" & Ent
Cad = Cad & "; Edad ;" & Format(MiEdad, "#####") & " ;" &
Format(MaEdad, "#####") & " ;"
Cad = Cad & Format(MEedad, "#####") & " ;" & Format(VEdad,
"#####") & " ;" & Ent
Cad = Cad & "; Peso ;" & Format(MiPeso, "#####") & " ;" &
Format(MaPeso, "#####") & " ;"
Cad = Cad & Format(MPeso, "#####") & " ;" & Format(VPeso,
"#####") & " ;" & Ent
Cad = Cad & "; Estatura ;" & Format(MiEstatura, "#####") & " ;" &
Format(MaEstatura, "#####") & " ;"
Cad = Cad & Format(MEstatura, "#####") & " ;" & Format(VEstatura,
"#####") & " ;" & Ent
Cad = Cad & "-----+-----+-----+-----+-----"
---" & Ent
Form2.Label2.Caption = Cad & Ent & Ent

Cad = "Recta de Regresión" & Ent
Cad = Cad & "Peso = " & Format(Beta1, "#####") & " * Estatura + "
& Format(Beta0, "#####")

```

```
Form2.Label2.Caption = Form2.Label2.Caption & Cad & Ent  
End Sub
```

21. Almacene de nuevo la aplicación

22. Ejecute la aplicación y verifique su funcionamiento



## Practica 10. Ordenamiento y Búsqueda.

### Objetivos

Al finalizar esta práctica el estudiante será capaz de:

1. Usar el tipo de dato Registro (Declaración Type)
2. Realizar un módulo de ordenamiento y búsqueda secuencial de datos almacenados en arreglos
3. Realizar una aplicación que permita almacenar una pequeña agenda telefónica usando arreglos de registros para almacenar los datos. La misma debe verificar si una persona ya existe (si tiene el mismo número de cédula) y ordenarla ya sea por cedula o por apellido-nombre.

### Antes de comenzar

Realicemos el Análisis E-P-S de la Aplicación Encuesta.

#### Entrada.

Dato	Identificador	Tipo	Restricciones
Cédula del contacto	Cedula	Cadena de caracteres	<> ""
Apellidos del contacto	Apellido	Cadena de caracteres	<> ""
Nombres del contacto	Nombre	Cadena de caracteres	<> ""
Teléfono del contacto	Telefono	Cadena de caracteres	<> ""

Tabla 10-1. Análisis EPS de la aplicación "Agenda Telefónica".

#### Proceso.

Esta aplicación almacena cada contacto de la agenda en un arreglo de registros, verificando que este no exista utilizando un algoritmo de búsqueda secuencial. Si el contacto no existe lo anexa al final de la agenda.

Para ordenar la agenda se utiliza el algoritmo de la burbuja, el cual consiste en revisar la lista desde el primer elemento hasta el último y si consigue dos elementos consecutivos que no están ordenados los intercambia.

#### Salida.

A medida que se va introduciendo los contactos de la agenda se va mostrando en la cuadrícula (Ms FlexGrid)

Para la interfaz de usuario se utilizarán dos formularios. En el primero se colocan 5 etiquetas, 4 cajas de texto, 4 botones de comando y una cuadrícula de datos. En el segundo un contenedor de controles, 2 botones de opción y un botón de comando.

### **Actividades a realizar:**


1. Ejecute el Visual Basic 6.0, seleccione “*Exe estándar*” en el cuadro de dialogo “*Nuevo Proyecto*”.
2. Establezca las siguientes propiedades al formulario:

<b>Propiedad</b>	<b>Valor</b>
Caption	Agenda de Telefonos
Forecolor	A su gusto
Height	7100
StartPosition	2 - CenterScreen
Width	11300

Tabla 10-2. Propiedades del formulario 1 de la aplicación “Agenda Telefónica”.

Este formulario contiene en la parte izquierda los cuadros de texto para recoger los datos de cada contacto que se guardan en un arreglo de registros y en la parte derecha el control MsFlexGrid que se utilizará para mostrar los datos a medida que se van leyendo.

3. Coloque los siguientes controles sobre el formulario aproximadamente como se muestra en la figura 10-1:
  - 5 etiquetas (Label)
  - 4 cuadros de texto (TextBox)
  - 4 botones de ordenes (CommandButton):
    - *Leer Registro*: Toma y valida los datos de los cuadros de texto, los guarda en un arreglo de registro y los muestra en el control MsFlexGrid

- *Ordenar Lista:* Ordena la Agenda ya sea por Cédula o por Apellido-Nombre y los muestra en el control MsFlexGrid
- *Inicializar Lista:* Inicializa el arreglo y el control MsFlexGrid
- *Salir*
- 1 cuadrícula flexible de datos (MsFlexGrid-)

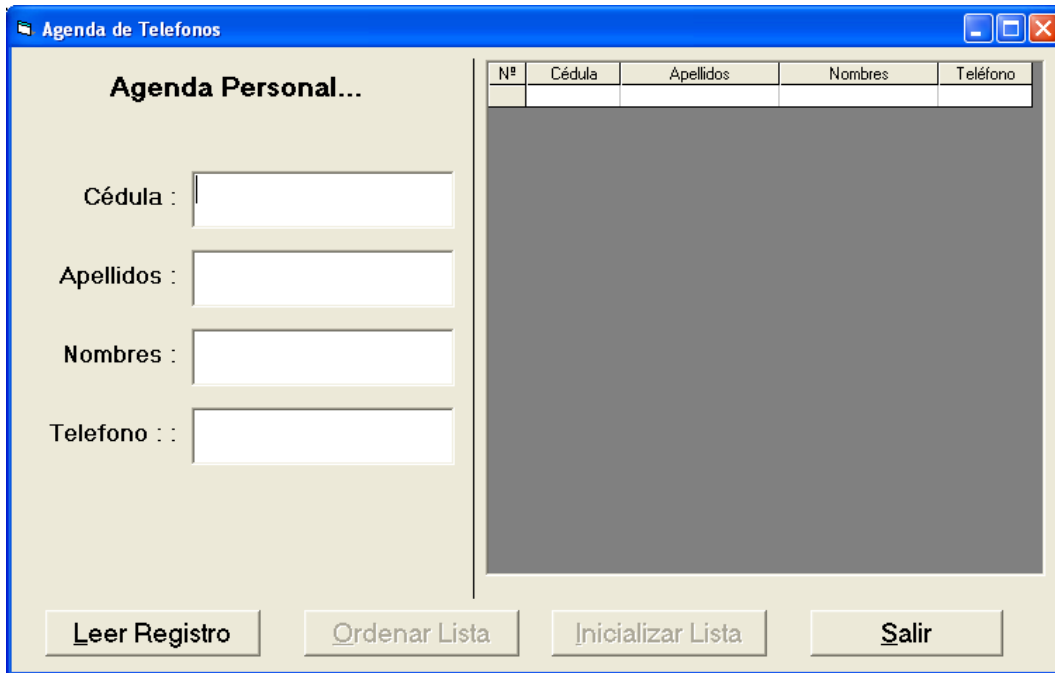


Figura 10-1. Formulario 1 de la aplicación “Agenda Telefónica”.

4. En la tabla 10-3 se presenta las propiedades de los controles que se agregaron para que queden como está en la figura 10-1:

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Agenda Personal...
	Width, Height, Top y Left	4575, 495, 240, 120
	Alignment	1 – Center
	Font	Ms Sans Serif, Negrita, 14
Etiqueta 2	(Nombre)	Label2
	Caption	Cédula:
	Width, Height, Top y Left	1650, 375, 1140, 120
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Etiqueta 3	(Nombre)	Label3

	Caption	Apellidos:
	Width, Height, Top y Left	1650, 375, 2280, 120
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Etiqueta 4	(Nombre)	Label4
	Caption	Nombres:
	Width, Height, Top y Left	1650, 375, 3120, 120
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Etiqueta 5	(Nombre)	Label4
	Caption	Telefono:
	Width, Height, Top y Left	1650, 375, 3960, 120
	Alignment	1 – Right Justify
	Font	Ms Sans Serif, Normal, 14
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	2800, 625, 1920, 920
	Font	Arial, Normal, 12
Cuadro de texto 2	(Nombre)	Text2
	Text	2800, 625, 2160, 920
	Width, Height, Top y Left	
	Font	Arial, Normal, 12
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	2800, 625, 3000, 920
	Font	Arial, Normal, 12
Cuadro de texto 4	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	2800, 625, 3840, 920
	Font	Arial, Normal, 12
Botón 1	(Nombre)	Command1
	Caption	&Leer Registro
	Width, Height, Top y Left	2295, 500, 6000, 360
	Font	Arial, Negrita, 12
Botón 2	(Nombre)	Command4
	Caption	&Ordenar Lista
	Width, Height, Top y Left	2295, 500, 6000, 3120
	Font	Arial, Negrita, 12
Botón 3	(Nombre)	Command3
	Caption	&Inicializar
	Width, Height, Top y Left	2295, 500, 6000, 5760
	Font	Arial, Negrita, 12

Botón 4	(Nombre)	Command4
	Caption	&Salir
	Width, Height, Top y Left	2295, 500, 6000, 8520
	Font	Arial, Negrita, 12
Cuadrícula 1	(Nombre)	MsFlexGrid1
	Width, Height, Top y Left	4200, 4800, 100, 4200
	Font	Arial, Negrita, 8

Tabla 10-3. Propiedades de los controles del formulario1 de la aplicación “Agenda telefónica”.

5. Inserte el siguiente código al Formulario 1. Este código contiene, en primer lugar la inicialización de los controles de lectura, después una función para validar datos que la debe programar como complemento de las prácticas de validación de datos.

```
Option Explicit
Dim Buscar As Boolean

' Procedimiento para inicializar los controles (TextBox) que se
utilizan
' Capturar datos
'
Public Sub Inicializar_controles()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
End Sub

'
' Función para validar los datos
'
Public Function Validar() As Boolean
    Dim X As Single

    ' Recuerden que aquí deben validar los datos de los controles
dependiendo
    ' del tipo de datos. También lo pueden hacer utilizando el evento
LostFocus
    ' de las cajas de texto

    Validar = True

End Function
```

6. El siguiente código de programa se refiere al evento LostFocus() del cuadro de texto 1, el cual se activa cuando éste pierde el foco. Es aquí donde se realiza la búsqueda y verificación de la cédula que se está leyendo. La variable Buscar es una bandera que se hace verdadera si el texto del cuadro de texto 1 está en la lista y falsa si no está.

```
Private Sub Text1_LostFocus()  
Dim i As Integer  
Dim CedB As String * 10  
  
    ' Buscar la cédula que acabamos de leer para verificar si ya existe  
    Buscar = False  
    CedB = Text1.Text  
  
    If Text1.Text <> "" Then  
        For i = 1 To N  
            If Agenda(i).Cédula = CedB Then  
                Buscar = True  
            End If  
        Next i  
    End If  
    If Buscar Then  
        MsgBox "Cédula ya existe", vbOKOnly, "Error....."  
        Text1.SetFocus  
    End If  
End Sub
```

7. El siguiente código de programa se refiere al evento clic() del botón 1, es aquí donde se lee el registro y se guarda en el arreglo de registro.

```
' Leer los datos y guardarlos en los arreglos  
'  
Private Sub Command1_Click()  
  
    If Not Buscar Then  
        If Validar() Then  
  
            N = N + 1  
            If N = 1 Then  
                Command3.Enabled = True  
            Else  
                If N = 2 Then
```

```
        Command2.Enabled = True
    End If
End If

' Pasar los datos de cuadro de texto para el ARREGLO
Agenda(N).Cédula = Text1.Text
Agenda(N).Apellido = Text2.Text
Agenda(N).Nombre = Text3.Text
Agenda(N).Telefono = Text4.Text

' Mostrar los datos en el MSFlexGrid
'
MSFlexGrid1.Rows = N + 1
MSFlexGrid1.Row = N
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = N
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = Text1.Text
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = Text2.Text
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = Text3.Text
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = Text4.Text

Call Inicializar_controles
Text1.SetFocus
End If
End If

End Sub
```

8. A continuación se muestra el código de programa de los botones 2, 3 y 4 (Ordenar la lista, Inicializar la lista y Salir):

```
'
' Ordenar la lista
Private Sub Command2_Click()
    Form1.Enabled = False
    Form2.Visible = True
End Sub

'
' Inicializar la lista
Private Sub Command3_Click()
```

```
N = 0
MSFlexGrid1.Rows = 1

End Sub

'
' Salir
Private Sub Command4_Click()
    End
End Sub
```

9. A continuación se muestra el código de programa del evento FormLoad(), en el cual se inicializa la cuadrícula de datos con los encabezados de columnas:

```
Private Sub Form_Load()
Dim i As Integer

' Fijar número de filas y columnas
MSFlexGrid1.Cols = 5
MSFlexGrid1.Rows = 2

' Fijar número de filas y columnas fijas o encabezados
MSFlexGrid1.FixedCols = 1
MSFlexGrid1.FixedRows = 1

' Tamaño (ancho) total de cada columna
MSFlexGrid1.Width = 6000
MSFlexGrid1.ColWidth(0) = 400
MSFlexGrid1.ColWidth(1) = 1000
MSFlexGrid1.ColWidth(2) = 1700
MSFlexGrid1.ColWidth(3) = 1700
MSFlexGrid1.ColWidth(4) = 1000

Call Encabezados

Call Inicializar_controles

Command2.Enabled = False
Command3.Enabled = False

N = 0

End Sub
```



10. Inserte un formulario a la aplicación. En este formulario se presentan las opciones del criterio de ordenamiento. Una vez seleccionado el criterio, ordena la agenda y la muestra en el control MsFlexGrid del formulario 1. Fije las propiedades mostradas en la tabla 10-4:

Propiedad	Valor
Caption	Ordenar
Forecolor	A su gusto
Height	3200
StartPosition	2 - CenterScreen
Width	5000

Tabla 10-4. Propiedades del formulario 2 de la aplicación "Agenda Telefónica".

11. Inserte en este formulario un contenedor de controles (Frame) y sobre él, dos botones de opción; así como también, un botón de comando. Fije las propiedades que se muestran en la tabla 10-5 de tal forma que se vea como la figura 10-2.

Control	Propiedad	Valor
Contenedor 1	(Nombre)	Frame1
	Caption	Criterio para ordenar
	Width, Height, Top y Left	3620, 1220, 600, 720
Botón de Opción 1	(Nombre)	Option1
	Caption	Ordenar por Cédula
	Width, Height, Top y Left	3140, 255, 360, 240
	Alignment	0 – Left Justify
	Font	Ms Sans Serif, Normal, 10
	Index	0
Botón de Opción 2	(Nombre)	Option1
	Caption	Ordenar por Cédula
	Width, Height, Top y Left	3140, 255, 840, 240
	Alignment	0 – Left Justify
	Font	Ms Sans Serif, Normal, 10
	Index	1
Botón 1	(Nombre)	Command1
	Caption	&Aceptar
	Width, Height, Top y Left	1450, 375, 2160, 3000
	Font	Arial, Negrita, 10

Tabla 10-5. Propiedades de los controles del formulario 2 de la aplicación "Agenda Telefónica".

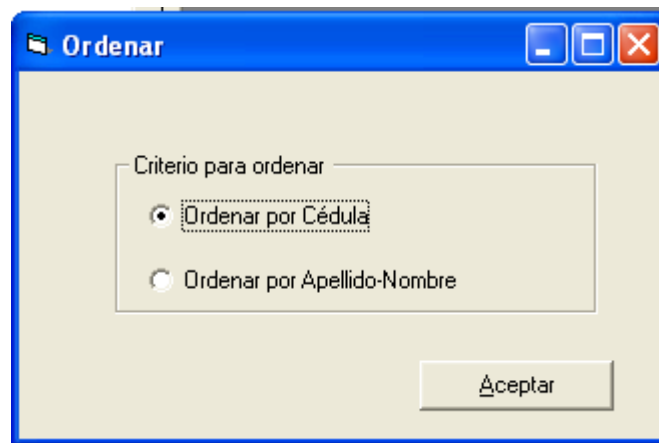


Figura 10-2. Formulario 2 de la aplicación “Agenda Telefónica”.

## 12. Inserte el siguiente código en el formulario 2:

```

Private Sub Command1_Click()
    Dim aux As Tipo_Agenda
    Dim menor As Integer, i As Integer, j As Integer

    If Option1(0) Then
        ' Ordenar por Cédula
        For i = 1 To N - 1
            menor = i
            For j = i + 1 To N
                If Agenda(menor).Cédula > Agenda(j).Cédula Then
                    menor = j
                End If
            Next j
            If i <> menor Then
                aux = Agenda(i)
                Agenda(i) = Agenda(menor)
                Agenda(menor) = aux
            End If
        Next i
    Else
        ' Ordenar por Apellido - Nombre
        For i = 1 To N - 1
            menor = i
            For j = i + 1 To N
                If Agenda(menor).Apellido & Agenda(menor).Nombre >
                Agenda(j).Apellido & Agenda(j).Nombre Then
                    menor = j
                End If
            Next j
    
```

```

        If i <> menor Then
            aux = Agenda(i)
            Agenda(i) = Agenda(menor)
            Agenda(menor) = aux
        End If
    Next i
End If

Call Encabezados

Form1.MSFlexGrid1.Clear
For i = 1 To N
    Form1.MSFlexGrid1.Row = i
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.Text = i
    Form1.MSFlexGrid1.Col = 1
    Form1.MSFlexGrid1.Text = Agenda(i).Cédula
    Form1.MSFlexGrid1.Col = 2
    Form1.MSFlexGrid1.Text = Agenda(i).Apellido
    Form1.MSFlexGrid1.Col = 3
    Form1.MSFlexGrid1.Text = Agenda(i).Nombre
    Form1.MSFlexGrid1.Col = 4
    Form1.MSFlexGrid1.Text = Agenda(i).Telefono
Next i

Form2.Visible = False
Form1.Enabled = True
Form1.SetFocus
End Sub

```

13. Inserte un módulo a la aplicación e inserte el siguiente código de programa. En primer lugar, se declara el tipo registro (Tipo\_Agenda), mediante la declaración Type; luego las variables globales: número de personas y el arreglo de registro Agenda; y por último el procedimiento Encabezados del control MsFlexGrid,

```

Option Explicit

Global N As Integer      ' Número de personas

Type Tipo_Agenda
    Cédula As String * 10
    Apellido As String * 20
    Nombre As String * 20

```

```
        Telefono As String * 15
End Type

Global Agenda(100) As Tipo_Agenda

Public Sub Encabezados()
Dim i As Integer

    For i = 0 To 4
        Form1.MSFlexGrid1.ColAlignment(i) = 3        ' Alineación
    Next i

    ' Encabezados de cada columna
    Form1.MSFlexGrid1.Row = 0
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.Text = " N° "
    Form1.MSFlexGrid1.Col = 1
    Form1.MSFlexGrid1.Text = " Cédula "
    Form1.MSFlexGrid1.Col = 2
    Form1.MSFlexGrid1.Text = " Apellidos "
    Form1.MSFlexGrid1.Col = 3
    Form1.MSFlexGrid1.Text = " Nombres "
    Form1.MSFlexGrid1.Col = 4
    Form1.MSFlexGrid1.Text = " Teléfono "
End Sub
```

14. Almacene de nuevo la aplicación

15. Ejecute la aplicación y verifique su funcionamiento

## Practica 11. Manejo de Archivos de texto.

### Objetivos

Al finalizar esta práctica el estudiante será capaz de:

1. Manejar archivos de texto para guardar y recuperar información en lote.
2. Crear un sistema de menús para la aplicación
3. Agregar un formulario *Acerca de...* a la aplicación.
4. Crear una aplicación que lea los datos de una encuesta (Edad, Peso y Estatura) y las muestre en un control MSFlexGrid, a su vez permita guardar y recuperar los datos en un archivo tipo texto. La aplicación contendrá un sistema de menú para la manipulación del archivo (cargar y guardar), así como para procesar los datos (estadísticas básicas y regresión)

### Antes de comenzar

Realicemos el Análisis E-P-S de la Aplicación Encuesta.

#### Entrada.

Dato	Identificador	Tipo	Restricciones
Edad del entrevistado	Edad	Real	>15
Peso del entrevistado	Peso	Real	>0
Estatura del entrevistado	Estatura	real	>0

Tabla 11-1. Análisis EPS de la aplicación "Archivo de texto".

#### Proceso

- Al comienzo del programa, éste cargará los datos almacenados en el archivo de texto (Datos.dat) y los mostrará en la cuadrícula de datos (MsFlexGrid)
- Cada vez que se lee un registro de datos se incrementa el número de entrevistados (N) y se muestra en la cuadrícula de datos

Para calcular las Estadísticas Básicas y la Regresión se sigue el procedimiento descrito en la práctica 9.

## Salida

La media, la varianza, el mínimo y el máximo para cada variable

Los parámetros de la recta de regresión

La interfaz de usuario de la aplicación está formada por tres formularios. El primero contiene 5 etiquetas, 3 cuadros de texto, dos líneas (line), un botón de comando, un control MSFlexGrid y un sistema de menús. El segundo contiene dos etiquetas para mostrar los resultados. Y el tercero es un formulario del tipo Acerca de...

### **Actividades a realizar:**

1. Ejecute el Visual Basic 6.0, seleccione “*Exe estándar*” en el cuadro de dialogo “*Nuevo Proyecto*”.
2. Establezca las siguientes propiedades al formulario:


Propiedad	Valor
Caption	Practica 11. Manejo de archivos tipo texto
Forecolor	A su gusto
Height	6300
StartPosition	2 - CenterScreen
Width	9100

Tabla 11-2. Propiedades del formulario 1 de la aplicación “Archivo de texto”.

Este formulario contiene en la parte superior el sistema de menú (Archivo, Procesar y Ayuda) para los diferentes procesos de la aplicación, en la parte izquierda los cuadros de texto para recoger los datos, en la parte derecha el control MsFlexGrid que se utilizará para mostrar los datos que se han ido leyendo. Observe que solo tiene un botón de comando: Leer Encuesta, el cual toma y valida los datos de los cuadros de texto.

3. Coloque sobre el formulario los siguientes controles y fije las propiedades mostradas en la tabla 11-3, para que quede aproximadamente como se muestra en la figura 11-1.

- 5 etiquetas (Label)

- 3 cuadros de texto (TextBox)
- 1 botón de comando (CommandButton)
- 2 líneas (Line)
- 1 cuadrícula flexible de datos (MsFlexGrid )

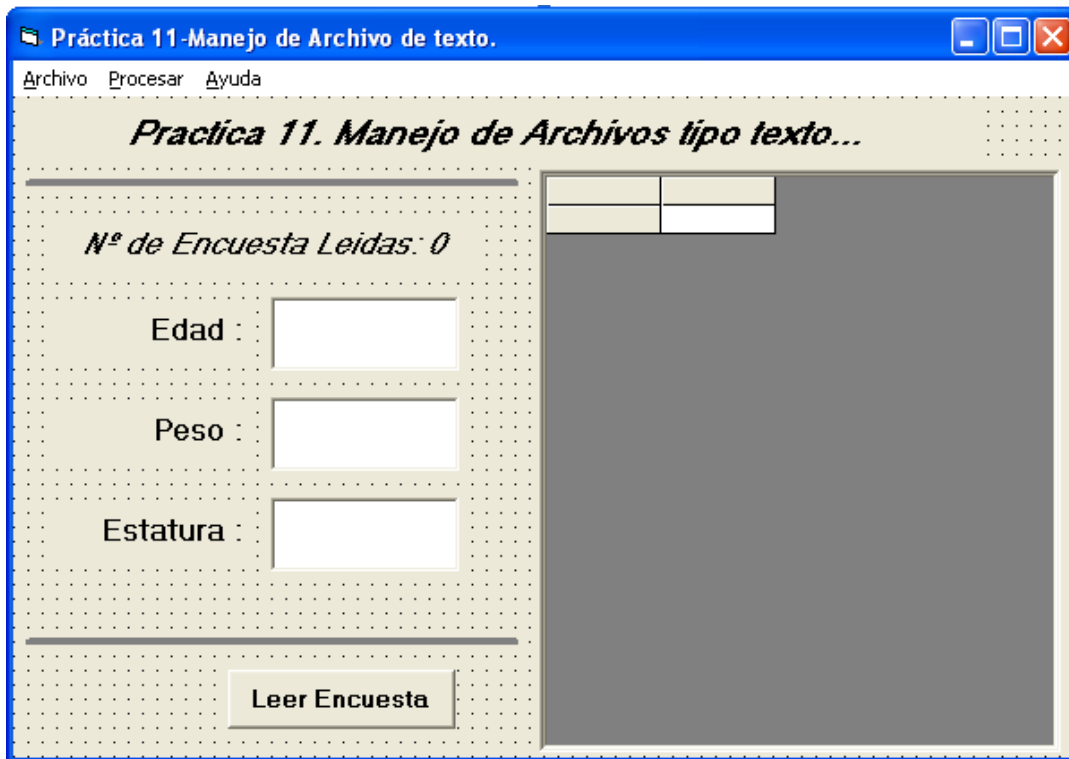


Figura 11-1. Formulario 1 de la aplicación “Archivo de texto”.

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Practica 11. Manejo de Archivos tipo texto...
	Width, Height, Top y Left	8000, 375, 120, 120
	Alignment	2 – Center
	Font	MS Sans Serif , Negrita cursiva, 14
Etiqueta 2	(Nombre)	Label2
	Caption	Nº de Encuesta Leidas: 0
	Width, Height, Top y Left	3500, 375, 1080, 360
	Font	MS Sans Serif , Negrita cursiva, 12
Etiqueta 3	(Nombre)	Label3
	Caption	Edad
	Width, Height, Top y Left	1575, 375, 1800, 360

	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12
Etiqueta 4	(Nombre)	Label4
	Caption	Peso
	Width, Height, Top y Left	1575, 375, 2640, 360
	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12
Etiqueta 5	(Nombre)	Label5
	Caption	Estatura
	Width, Height, Top y Left	1575, 375, 3480, 360
	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	1575, 615, 1680, 2160
	Font	MS Sans Serif, Normal, 12
Cuadro de texto 2	(Nombre)	Text2
	Text	
	Width, Height, Top y Left	1575, 615, 2520, 2160
	Font	MS Sans Serif, Normal, 12
Cuadro de texto 3	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	1575, 615, 3360, 2160
	Font	MS Sans Serif, Normal, 12
Botón 1	(Nombre)	Command1
	Caption	&Leer Encuesta
	Width, Height, Top y Left	1900, 500, 4800, 1800
Linea 1	(Nombre)	Line1
	BorderWidth	4
	X1, X2, Y1, Y2	120, 420, 720, 720
Linea 2	(Nombre)	Line2
	BorderWidth	4
	X1, X2, Y1, Y2	120, 420, 4500, 4500
Cuadrícula 1	(Nombre)	MsFlexGrid1
	Width, Height, Top y Left	4400, 4900, 600, 4400
	Las demás propiedades se fijaran en tiempo de ejecución	


Tabla 11-3. Propiedades de los controles del formulario1 de la aplicación "Archivo de texto".

4. Se va a crear el sistema de menús con las siguientes opciones principales: *Archivo, Procesar y Ayuda*. La opción *Archivo* tiene cuatro opciones: *Cargar*



*Datos, Guardar Datos, Limpiar Grilla y Salir, la de Procesar dos: Calcular Estadísticas Básicas y Calcular Regresión, y la de Ayuda tiene una opción Acerca de...*

El procedimiento para crear el menú que se muestra en la figura 11-3 es el siguiente:

- Invoque al editor de menú, *Herramientas – Editor de Menús*.
- En el apartado *Caption* introduzca *&Archivo*, el símbolo *&* hace que la letra “A” aparezca subrayado, presione TAB para ir a *Name* y ahí introduzca *MenuArchivo*. Este será el identificador de la opción *Archivo* en caso de necesitarlo a nivel de programación. Presione el botón *Siguiente* para pasar a la próxima opción (Cargar datos).
- Presione la flecha a la derecha  para indicar que esta opción está a nivel jerárquico inferior con respecto a la opción *Archivo*; es decir, que es una sub-opción de la opción *Archivo*. La misma aparecerá indentada con respecto a *Archivo* (ver figura 11-2)

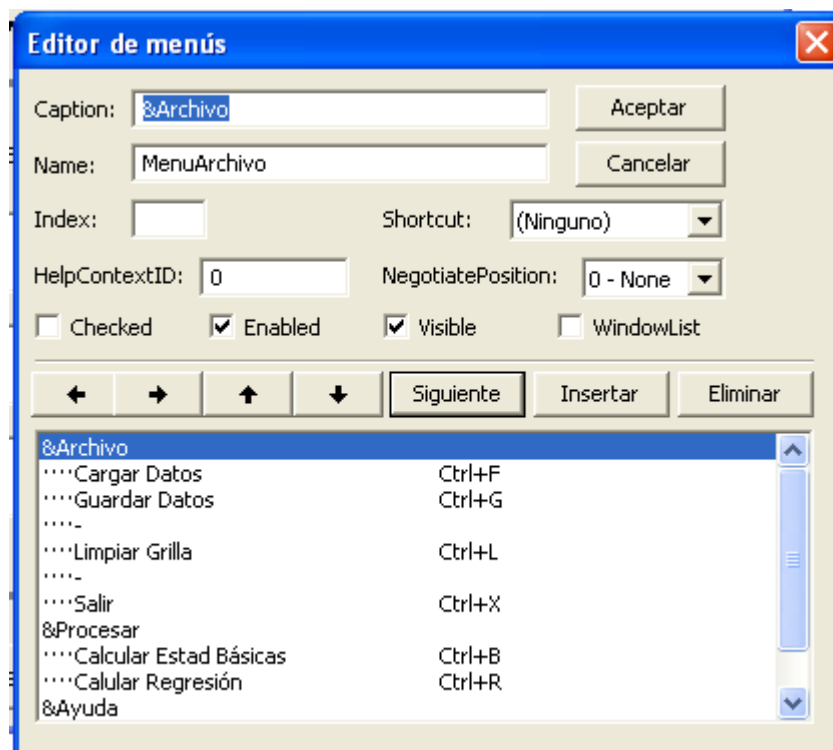






Figura 11-2. Ventana editor de menús.

- d) En el apartado *Caption* introduzca *Cargar Datos*, presione TAB para pasar a *Name* e introduzca *MenuCargar* y en *ShortCut* seleccione Ctrl.+F. Presione el botón *Siguiente* para pasar a la próxima opción (Guardar datos).
- e) En el apartado *Caption* introduzca *Guardar Datos*, presione TAB para pasar a *Name* e introduzca *MenuGuardar* y en *ShortCut* seleccione Ctrl.+G. Presione el botón *Siguiente* para pasar a la próxima opción (separador).
- f) Ahora vamos a crear una barra separadora de menú, para lo cual introduzca en el apartado *Caption* un guión (-), presione TAB para pasar a *Name* e introduzca cualquier identificador, por comodidad escriba *L1*. Presione el botón *Siguiente* para pasar a la próxima opción (Limpiar Grilla).
- g) En el apartado *Caption* introduzca *Limpiar Grilla*, presione TAB para pasar a *Name* e introduzca *MenuLimpiar* y en *ShortCut* seleccione Ctrl.+L. Presione el botón *Siguiente* para pasar a la próxima opción (separador 2).
- h) En el apartado *Caption* introduzca un guión (-), presione TAB para pasar a *Name* e introduzca *L2*. Presione el botón *Siguiente* para pasar a la próxima opción (Salir).
- i) En el apartado *Caption* introduzca *Salir*, presione TAB para pasar a *Name* e introduzca *MenuSalir* y en *ShortCut* seleccione Ctrl.+X. Presione el botón *Siguiente* para pasar a la próxima opción (opción *Procesar* del menú principal).
- j) Presione flecha a la izquierda  para volver al mismo nivel jerárquico de la opción *Archivo*.
- k) En el apartado *Caption* introduzca *&Procesar*, presione TAB para ir a *Name* y ahí introduzca *MenuProcesar*. Presione el botón *Siguiente* para pasar a la próxima opción (Calcular Estadísticas Básicas).
- l) Presione la flecha a la derecha  para indicar que esta opción está a nivel jerárquico inferior con respecto a la opción *Procesar*.

- m) En el apartado *Caption* introduzca *Calcular Estad Básicas*, presione TAB para pasar a *Name* e introduzca *MenuBasicas* y en *ShortCut* seleccione Ctrl.+B. Presione el botón *Siguiente* para pasar a la próxima opción (Guardar datos).
- n) En el apartado *Caption* introduzca *Calcular Regresión*, presione TAB para pasar a *Name* e introduzca *MenuRegresion* y en *ShortCut* seleccione Ctrl.+R. Presione el botón *Siguiente* para pasar a la próxima opción (opción *Ayuda* del menú principal).
- o) Presione flecha a la izquierda  para volver al mismo nivel jerárquico de las opciones Archivo y Procesar.
- p) En el apartado *Caption* introduzca *&Ayudar*, presione TAB para ir a *Name* y ahí introduzca *MenuAyuda*. Presione el botón *Siguiente* para pasar a la próxima opción (Acerca de...).
- q) Presione la flecha a la derecha  para indicar que esta opción está a nivel jerárquico inferior con respecto a la opción *Ayuda*.
- r) En el apartado *Caption* introduzca *Acerca de...*, presione TAB para pasar a *Name* e introduzca *MenuAcerca*.

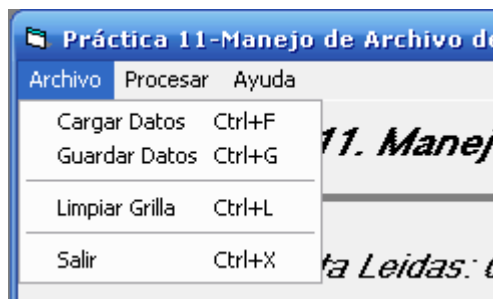


Figura 11-3. Menú creado con la opción Archivo abierta

5. Este sería un buen momento para guardar la aplicación y así evitar pérdida de datos.
6. Inserte el siguiente código de programa al formulario 1. El procedimiento `LimpiarGrilla()` permite inicializar la cuadrícula de datos desde cualquier parte

del programa y LimpiarControles es para Inicializar los cuadros de texto una vez que se haya leído un dato:

```
Option Explicit

Public Sub LimpiarGrilla()
Dim i As Single
    N = 0
    MSFlexGrid1.Cols = 4
    MSFlexGrid1.Rows = 2
    MSFlexGrid1.FixedCols = 1
    MSFlexGrid1.FixedRows = 1
    MSFlexGrid1.Clear
    MSFlexGrid1.Width = 4200
    MSFlexGrid1.ColWidth(0) = 400
    For i = 1 To 3
        MSFlexGrid1.ColWidth(i) = 1200
        MSFlexGrid1.ColAlignment(i) = 3      ' Alineación
    Next i

    ' Encabezados de cada columna
    MSFlexGrid1.Row = 0
    MSFlexGrid1.Col = 0
    MSFlexGrid1.Text = " N° "
    MSFlexGrid1.Col = 1
    MSFlexGrid1.Text = " Edad "
    MSFlexGrid1.Col = 2
    MSFlexGrid1.Text = " Peso "
    MSFlexGrid1.Col = 3
    MSFlexGrid1.Text = " Estatura "
End Sub

Public Sub LimpiarControles()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
End Sub
```

7. Presione el formulario para programar el evento Form\_Load(), el cual permite limpiar la cuadrícula y la carga inicial de los datos guardados en el archivo de texto.

```
Private Sub Form_Load()

    Open "Datos.dat" For Input As #1
    LimpiarGrilla
    While Not EOF(1)
        Input #1, Ed, Pe, Es
        N = N + 1
        MSFlexGrid1.Rows = N + 1
        MSFlexGrid1.Row = N
        MSFlexGrid1.Col = 1
        MSFlexGrid1.Text = Ed
        MSFlexGrid1.Col = 2
        MSFlexGrid1.Text = Pe
        MSFlexGrid1.Col = 3
        MSFlexGrid1.Text = Es
    Wend
    Close #1

    CR = Chr(10) & Chr(13)
End Sub
```

8. Para programar el botón Leer Encuesta haga doble clic sobre él e inserte el siguiente código:

```
Private Sub Command1_Click()
    If Validar Then
        N = N + 1
        MSFlexGrid1.Rows = N + 1
        MSFlexGrid1.Row = N
        MSFlexGrid1.Col = 0
        MSFlexGrid1.Text = N
        MSFlexGrid1.Col = 1
        MSFlexGrid1.Text = Text1.Text
        MSFlexGrid1.Col = 2
        MSFlexGrid1.Text = Text2.Text
        MSFlexGrid1.Col = 3
        MSFlexGrid1.Text = Text3.Text
        Call LimpiarControles
    End If

End Sub

' Para validar los datos introducidos
```

```
'  
Public Function Validar() As Boolean  
'  
' Tome la function Validar de la práctica 9 y coloquela aquí  
'  
  
End Function
```

9. Seleccione la opción Archivo – Cargar datos para invocar el editor y programar el evento MenuCargar\_Click insertando el siguiente código:

```
Private Sub MenuCargar_Click()  
    Dim Ed As Integer  
    Dim Pe As Single  
    Dim Es As Single  
  
    Open "Datos.dat" For Input As #1  
    LimpiarGrilla  
    While Not EOF(1)  
        Input #1, Ed, Pe, Es  
        N = N + 1  
        MSFlexGrid1.Rows = N + 1  
        MSFlexGrid1.Row = N  
        MSFlexGrid1.Col = 1  
        MSFlexGrid1.Text = Ed  
        MSFlexGrid1.Col = 2  
        MSFlexGrid1.Text = Pe  
        MSFlexGrid1.Col = 3  
        MSFlexGrid1.Text = Es  
    Wend  
    Close #1
```

```
End Sub
```

10. Para programar el evento MenuGuardar\_Click seleccione la opción Archivo – Guardar Datos e inserte el siguiente código:

```
Private Sub MenuGuardar_Click()  
    Dim Ed As Integer  
    Dim Pe As Single  
    Dim Es As Single  
    Dim i As Integer
```

```

Open "Datos.dat" For Output As #1
For i = 1 To N
    MSFlexGrid1.Row = i
    MSFlexGrid1.Col = 1
    Ed = MSFlexGrid1.Text
    MSFlexGrid1.Col = 2
    Pe = MSFlexGrid1.Text
    MSFlexGrid1.Col = 3
    Es = MSFlexGrid1.Text
    Write #1, Ed, Pe, Es
Next i
Close #1
End Sub

```

11. Para programar el evento `MenuLimpiar_Click` seleccione la opción Archivo – Limpiar Grilla e inserte el siguiente código:

```

Private Sub MenuLimpiar_Click()
    Call LimpiarGrilla
End Sub

```

12. Para programar el evento `MenuSalir_Click` seleccione la opción Archivo – Salir e inserte el siguiente código:

```

Private Sub MenuSalir_Click()
    End
End Sub

```

13. Para programar el evento `MenuBasicas_Click` seleccione la opción Procesar – Calcular Estad Básicas e inserte el siguiente código:

```

Private Sub MenuBasicas_Click()
    Dim C As String
    Dim Ed As Integer, SEd As Single, S2Ed As Single, Med As Single, VED
    As Single
    Dim Pe As Single, SPe As Single, S2Pe As Single, MPe As Single, VPe
    As Single
    Dim Es As Single, SEs As Single, S2Es As Single, MEs As Single, VEs
    As Single
    Dim i As Integer

    If N > 1 Then
        Form1.Enabled = False
        SEd = 0: S2Ed = 0

```

```

SPE = 0: S2Pe = 0
SEs = 0: S2Es = 0

For i = 1 To N
    MSFlexGrid1.Row = i
    MSFlexGrid1.Col = 1
    Ed = MSFlexGrid1.Text
    SEd = SEd + Ed
    S2Ed = S2Ed + Ed ^ 2

    MSFlexGrid1.Col = 2
    Pe = MSFlexGrid1.Text
    SPE = SPE + Pe
    S2Pe = S2Pe + Pe ^ 2

    MSFlexGrid1.Col = 3
    Es = MSFlexGrid1.Text
    SEs = SEs + Es
    S2Es = S2Es + Es ^ 2
Next i

Med = SEd / N: VEd = (S2Ed - N * Med ^ 2) / (N - 1)
MPe = SPE / N: VPe = (S2Pe - N * MPe ^ 2) / (N - 1)
MEs = SEs / N: VEs = (S2Es - N * MEs ^ 2) / (N - 1)

C = "Variable Edad" & CR
C = C & "    Media = " & Med & CR & "Varianza = " & VEd & CR & CR
C = C & "Variable Peso" & CR
C = C & "    Media = " & MPe & CR & "Varianza = " & VPe & CR & CR
C = C & "Variable Estatura" & CR
C = C & "    Media = " & MEs & CR & "Varianza = " & VEs

Form2.Label1.Caption = "Estadísticas Básicas"
Form2.Label2.Caption = C
Form2.Show
End If
End Sub

```

14. Para programar el evento MenuRegre\_Click seleccione la opción Procesar – Calcular Regresión e inserte el siguiente código:

```

'
' Calculo de Beta0 y Beta1 en un modelo de regresión en el que la

```



```

' Estatura (X) explica al Peso (Y)
Private Sub MenuRegre_Click()
    Dim C As String
    Dim Es As Single, SEs As Single, S2Es As Single ' X, suma de X y
Suma X^2
    Dim Pe As Single, SPe As Single ' Y, suma de Y
    Dim SEsPe ' Suma de X*Y
    Dim B0 As Single, B1 As Single
    Dim i As Integer

    If N > 1 Then
        Form1.Enabled = False

        SPe = 0
        SEs = 0: S2Es = 0
        SEsPe = 0

        For i = 1 To N
            MSFlexGrid1.Row = i

            MSFlexGrid1.Col = 2
            Pe = MSFlexGrid1.Text ' Y
            SPe = SPe + Pe ' Suma de Y

            MSFlexGrid1.Col = 3
            Es = MSFlexGrid1.Text ' X
            SEs = SEs + Es ' Suma de X
            S2Es = S2Es + Es ^ 2 ' Suma de X^2

            SEsPe = SEsPe + Es * Pe ' Suma de X*Y

        Next i

        B1 = (N * SEsPe - SEs * SPe) / (N * S2Es - SEs ^ 2)
        B0 = (SPe - B1 * SEs) / N

        C = "Parametros de la Regresión entre la Estatura y el Peso" &
CR
        C = C & " Beta0 = " & B0 & CR
        C = C & " Beta1 = " & B1 & CR & CR
        C = C & "Peso = " & B0 & " + " & B1 & "*Estatura"

        Form2.Label1.Caption = "Regresión entre el peso y la estatura"
        Form2.Label2.Caption = C
    End If
End Sub

```

```

    Form2.Show
End If
End Sub

```

15. Agregue otro formulario a la aplicación y fije las propiedades mostradas en la tabla 11-4.

Propiedad	Valor
Caption	Resultados-Práctica 11
Forecolor	A su gusto
Height	6300
StartPosition	2 - CenterScreen
Width	8200

Tabla 11-4. Propiedades del formulario 2 de la aplicación "Archivo de texto".

16. Agregue al formulario dos etiquetas para mostrar los resultados de las opciones *Calcular Estadísticas Básicas* y *Calcular Regresión* y un botón para regresar al formulario 1. Fije las propiedades a estos controles (Tabla 11-5). Las etiquetas de este formulario se llenan con los resultados cuando se invocan los eventos *MenuBasicas\_Click* y *MenuRegre\_Click*.

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	
	Width, Height, Top y Left	7700, 500, 120, 120
	Alignment	2 - Center
	Border Style	1 – Fixed Single
	Font	MS Sans Serif , Negrita cursiva, 18
Etiqueta 2	(Nombre)	Label2
	Caption	7700, 3800, 960,120
	Width, Height, Top y Left	3500, 375, 1080, 360
	Border Style	1 – Fixed Single
	Font	MS Sans Serif , Normal, 10
Botón 4	(Nombre)	Command1
	Caption	&Regresar
	Width, Height, Top y Left	1600, 500, 4900, 6200
	Font	Arial, Negrita, 9

Tabla 11-5. Propiedades de los controles del formulario 2 de la aplicación "Archivo de texto".

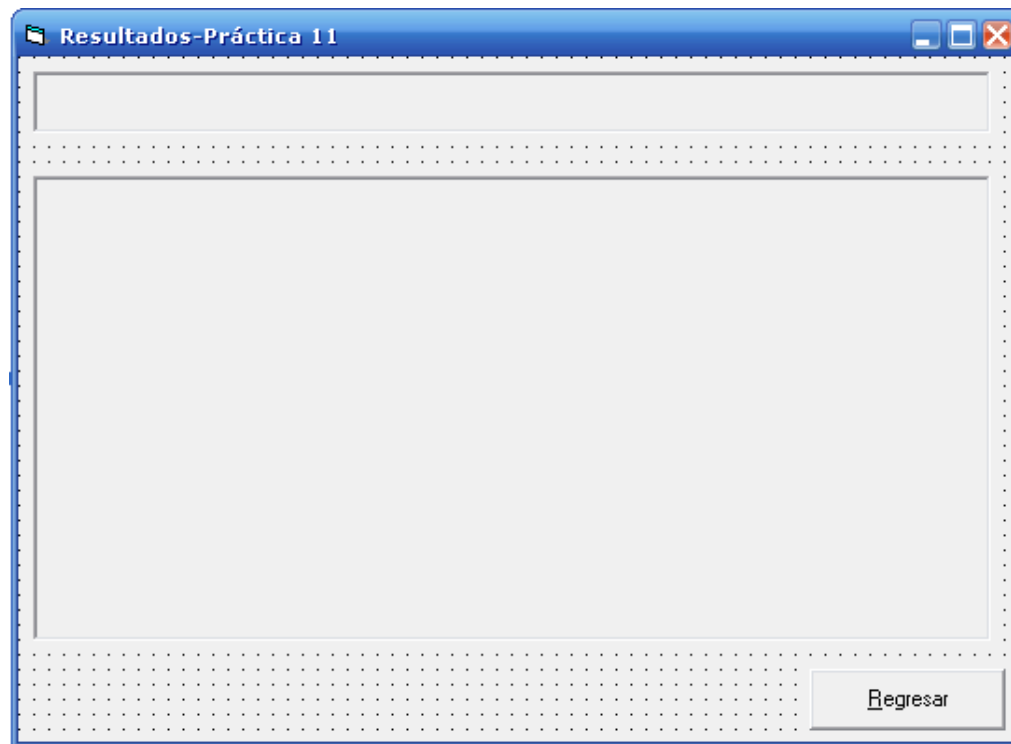


Figura 11-4. Formulario 2 - Resultados

17. Agregue el siguiente código de programa al formulario 2

```
Private Sub Command1_Click()  
    Form1.Enabled = True  
    Unload Me  
End Sub
```

18. Agregue un módulo a la aplicación, en el cual se declaran el tipo registro (Declaración Type) y las variables globales: número de personas y la variable CR (Enter).

```
Option Explicit  
  
Global N As Integer      ' Número de personas  
  
Global CR As String
```

19. Para insertar un formulario *Acerca de...* a la aplicación Seleccione Agregar Formulario y en el cuadro de diálogo Agregar Formulario seleccione Cuadro de diálogo *Acerca de...*

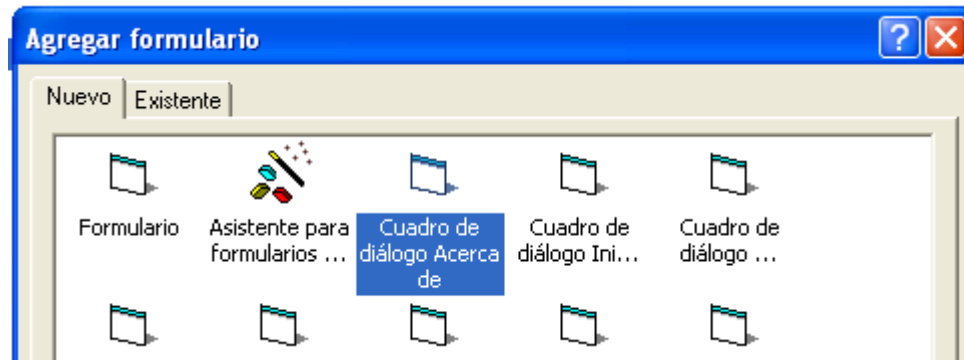


Figura 11-5. Cuadro de diálogo Agregar formulario

20. Modifique los textos de las etiquetas del formulario *Acerca de ...* a los requerimientos de la aplicación.

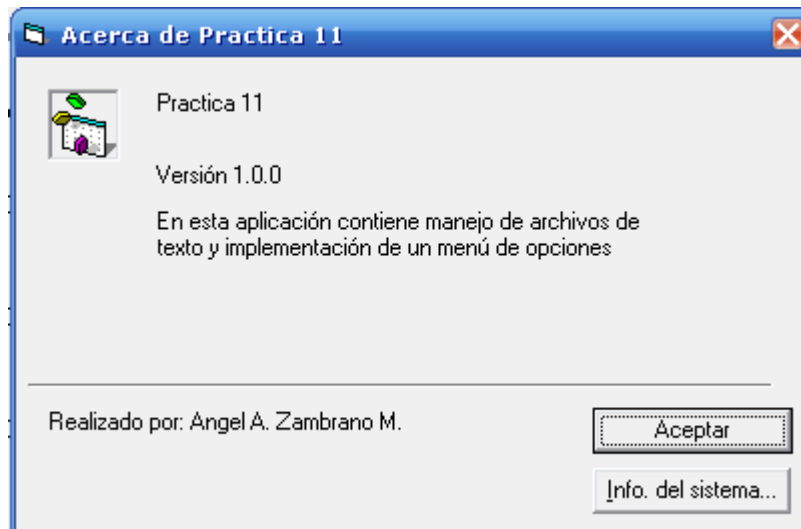


Figura 11-6. Formulario Acerca de modificado para la práctica 11

21. Solo falta programar el evento `MenuAcerca_Click` seleccione la opción Ayuda – Acerca de... en el formulario 1 e inserte el siguiente código:

```
Private Sub MenuAcerca_Click()
    Form1.Enabled = False
    frmAbout.Show
End Sub
```

22. En el formulario 3 (Acerca de) haga doble clic en el botón *Aceptar* y re programe el evento Clic

```
Private Sub cmdOK_Click()  
    Form1.Enabled = True  
    Unload Me  
End Sub
```

23. Almacene de nuevo la aplicación

24. Ejecute la aplicación y verifique los resultados. Si se produce un error (archivo no existe) vaya a la carpeta donde almacenó la aplicación y cree ahí un archivo vacío con el “block de notas” con el nombre “Datos.dat”.

## Practica 12. Manejo de Archivos de tipo (registros).

### Objetivos

Al finalizar esta práctica el estudiante será capaz de:

1. Usar registros para leer y manipular información.
2. Usar archivos de tipo para almacenar y recuperar información por registros.
3. Crear una aplicación que permita guardar y recuperar los datos de un libro (Cota, Autor, Título y Editorial).

### Antes de comenzar

Realicemos el Análisis E-P-S de la Aplicación Encuesta.

### Entrada.

Dato	Identificador	Tipo	Restricciones
Cota del Libro	Cota	Alfanumérico	<> ""
Autor(es) del Libro	Autor	Alfanumérico	<> ""
Título del Libro	Título	Alfanumérico	<> ""
Editorial del Libro	Editorial	Alfanumérico	<> ""

Tabla 12-1. Análisis EPS de la aplicación "Biblioteca".

### Proceso

- Para guardar un registro en el archivo se introducen los datos y se realiza una búsqueda secuencial del campo clave del registro (COTA) sobre todo el archivo. Si éste se encuentra (existe) se muestra un mensaje de error en caso contrario se anexa el registro al final del archivo.
- Para recuperar un registro del archivo se introduce la cota (clave del registro), la cual se busca secuencialmente en todo el archivo y si se encuentra se muestra el registro en los cuadro de texto y en caso contrario se muestra un mensaje de error.

- Para la búsqueda secuencial se abre el archivo y luego se va leyendo uno a uno los registros hasta conseguir el deseado o hasta que llegue al final del archivo.

### Salida

El registro de datos.

Para la interfaz de usuario se usaran los siguientes controles:

- 6 etiquetas (Label)
- 4 cajas de texto (TextBox)
- 3 botones de comando (CommandButton)

### Actividades a realizar:

1. Ejecute el Visual Basic 6.0, seleccione “*Exe estándar*” en el cuadro de dialogo “*Nuevo Proyecto*”.
2. Establezca las siguientes propiedades al formulario:

Propiedad	Valor
Caption	Practica 12. Archivos de tipo
ForeColor	A su gusto
Height	5600
StartPosition	2 - CenterScreen
Width	8100

Tabla 12-2. Propiedades del formulario de la aplicación “Biblioteca”.

3. Coloque sobre el formulario las seis etiquetas, los cuatro cuadro de texto y los tres botones de comando y fije las propiedades contenidas en la tabla 12-3, de tal manera que se vea aproximadamente como se muestra en la figura 12-1:

Figura 12-1. Formulario 1 de la aplicación “Biblioteca”.

Control	Propiedad	Valor
Etiqueta 1	(Nombre)	Label1
	Caption	Sistema de Información de la Biblioteca Nacional
	Width, Height, Top y Left	7300, 375, 120, 250
	Alignment	2 – Center
	Font	MS Sans Serif , Negrita cursiva, 14
Etiqueta 2	(Nombre)	Label2
	Caption	Sub-Sistema de Información de Libros
	Alignment	2 – Center
	Width, Height, Top y Left	7300, 375, 600, 250
	Font	MS Sans Serif , Negrita cursiva, 12
Etiqueta 3	(Nombre)	Label3
	Caption	Cota:
	Width, Height, Top y Left	1100, 260, 1450, 600
	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12



Etiqueta 4	(Nombre)	Label4
	Caption	Título:
	Width, Height, Top y Left	1100, 260, 2050, 600
	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12
Etiqueta 5	(Nombre)	Label5
	Caption	Autor:
	Width, Height, Top y Left	1100, 260, 2650, 600
	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12
Etiqueta 6	(Nombre)	Label5
	Caption	Editorial:
	Width, Height, Top y Left	1000, 260, 3250, 600
	Alignment	1 – Right Justify
	Font	MS Sans Serif , Negrita, 12
Cuadro de texto 1	(Nombre)	Text1
	Text	
	Width, Height, Top y Left	2000, 375, 1400, 1700
	Font	MS Sans Serif, Normal, 12
Cuadro de texto 2	(Nombre)	Text2
	Text	
	Width, Height, Top y Left	5300, 375, 2000, 1700
	Font	MS Sans Serif, Normal, 12
Cuadro de texto 3	(Nombre)	Text3
	Text	

	Width, Height, Top y Left	5300, 375, 2600, 1700
	Font	MS Sans Serif, Normal, 12
Cuadro de texto 4	(Nombre)	Text3
	Text	
	Width, Height, Top y Left	5300, 375, 3200, 1700
	Font	MS Sans Serif, Normal, 12
Botón 1	(Nombre)	Command1
	Caption	&ANEXAR
	Width, Height, Top y Left	1800, 500, 4450, 1600
Botón 2	(Nombre)	Command2
	Caption	&CONSULTAR
	Width, Height, Top y Left	1800, 500, 4450, 3800
Botón 3	(Nombre)	Command3
	Caption	&SALIR
	Width, Height, Top y Left	1800, 500, 4450, 6000

Tabla 12-3. Propiedades de los controles del formulario de la aplicación "Biblioteca".

4. Inserte el siguiente código de programa en el formulario, en primer lugar el botón Anexar y luego Consultar.

```
Option Explicit

Private Sub Command1_Click()
    Dim RLibro As Tipo_Libro
    Dim CotaB As String * 15
    Dim Enc As Boolean

    ' Captura y VALIDACION de los Campos
    CotaB = Text1.Text

    ' Buscar el libro en el archivo

    Enc = False
```

```
Open "PRUEBA.DAT" For Random As #1 Len = Len(RLibro)
While (Not Enc) And (Not EOF(1))
  Get #1, , RLibro
  If RLibro.Cota = CotaB Then
    Enc = True
  End If
Wend

If Enc Then
  MsgBox "Cota ya Existe. Verifique", vbOKOnly, "Error"
  Text1.SetFocus
Else
  ' Llenar el Registro
  RLibro.Cota = CotaB
  RLibro.Titulo = Text2.Text
  RLibro.Autor = Text3.Text
  RLibro.Editorial = Text4.Text
  Put #1, , RLibro
End If
Close #1
End Sub

Private Sub Command2_Click()
  Dim RLibro As Tipo_Libro
  Dim CotaB As String * 15
  Dim Enc As Boolean

  ' Captura y VALIDACION de los Campos
  CotaB = Text1.Text

  ' Buscar el libro en el archivo

  Enc = False
  Open "PRUEBA.DAT" For Random As #1 Len = Len(RLibro)
  While (Not Enc) And (Not EOF(1))
    Get #1, , RLibro
    If RLibro.Cota = CotaB Then
      Enc = True
    End If
  Wend

  If Not Enc Then
    MsgBox "Cota no Existe. Verifique", vbOKOnly, "Error"
```

```
        Text1.SetFocus
    Else
        ' Llenar el Registro
        Text2.Text = RLibro.Titulo
        Text3.Text = RLibro.Autor
        Text4.Text = RLibro.Editorial
    End If
    Close #1

End Sub

Private Sub Command3_Click()
    End
End Sub
```

5. Inserte el siguiente un módulo al proyecto en inserte el siguiente código para declarar el Tipo Registro Libro.

```
' Declaración del registro de Datos
Type Tipo_Libro
    Cota As String * 15
    Titulo As String * 30
    Autor As String * 20
    Editorial As String * 20
End Type
```

6. Almacene la aplicación tal y como se ha hecho en las prácticas anteriores.
7. Ejecute la aplicación y verifique su funcionamiento
8. Como complemento de la práctica agregue un botón realizar un listado de todos los libros contenidos en el archivo.

## **ANEXO: Introducción al Visual Basic**

## ***El Lenguaje BASIC***

El lenguaje de programación BASIC (Beginner's All purpose Symbolic Instruction Code) nació en el año 1964 como una herramienta destinada a principiantes, buscando una forma sencilla de realizar programas, empleando un lenguaje casi igual al usado en la vida cotidiana (inglés), y con instrucciones muy sencillas y escasas. Para el año de su nacimiento, este lenguaje cubría casi todas las necesidades para la ejecución de programas. Los autores fueron John G. Kemeny y Thomas E. Kurtz.

La evolución del BASIC por los años 70 fue escasa, dado el auge que tomaron en aquella época lenguajes de alto nivel como el FORTRAN y el COBOL. En 1978 se definió una norma para unificar los BASIC's existentes creándose la normativa BASIC STANDARD.

Con la aparición a mediados de los 80 de los primeros PC, dedicados principalmente al usuario doméstico, el BASIC resurgió como lenguaje de programación, muchos de estos PC lo usaban como único sistema operativo (Sinclair, Spectrum, Amstrad)

Con la popularización del PC, salieron varias versiones del BASIC que funcionaban en este tipo de computadores (BASICA, GW-BASIC), pero estas versiones del BASIC no hicieron otra cosa que terminar de matar este lenguaje. Los programadores profesionales no llegaron a utilizarlo, debido a sus desventajas con respecto a otros lenguajes de programación (PASCAL, C, CLIPPER), con estas versiones llegó incluso a perder crédito entre los profesionales de la informática.

Tal fue ese abandono por parte de los usuarios, que la aparición del QBASIC de Microsoft, una versión bastante potente del BASIC que corregía casi todos los defectos anteriores, pasó prácticamente inadvertida. Esta versión del popular BASIC era ya un lenguaje estructurado, lo que permitía crear programas modulares, mediante subrutinas y módulos, capaz de crear programas ya

competitivos con otros lenguajes de alto nivel. Sin embargo llegó tarde, pues los entornos MS-DOS estaban ya superados por el entorno gráfico Windows.

Sin embargo algo había en el BASIC que tentaba a superarse: su gran sencillez de manejo. Si a esto se le añade el entorno gráfico Windows, el aprovechamiento al máximo de las posibilidades de Windows en cuanto a intercambio de información, de sus librerías, de sus controladores, manejo de bases de datos, etc. el producto resultante puede ser algo que satisfaga todas las necesidades de programación en el entorno Windows. La suma de todos estos detalles es VISUAL BASIC , un lenguaje que conserva del BASIC de los años 80 prácticamente solo su nombre y su sencillez, y su aceptación a nivel profesional hizo borrar por fin el "mal nombre" asociado a la palabra BASIC.

No se pueden comparar sus prestaciones con otros lenguajes cuando deseamos llegar al fondo de la máquina y controlar uno a uno sus registros, no es ese el fin perseguido con Visual Basic y si se quiere llegar a esas precisiones será necesario utilizar otro lenguaje que permita bajar el nivel de programación (Visual-C, C++), o realizar librerías (DLLs) que lo hagan. En la mayor parte de las aplicaciones, las herramientas aportadas por Visual Basic son más que suficiente para lograr un programa fácil de realizar y de altas prestaciones.

En conclusión Visual Basic es una herramienta de diseño de aplicaciones para Windows, en la que éstas se desarrollan en gran parte a partir del diseño de una interfase gráfica. En una aplicación Visual Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfase gráfica. La palabra “**Visual**” hace referencia al método que se utiliza para crear la interfaz gráfica de usuario. En lugar de escribir numerosas líneas de código para implementar la interfaz, simplemente se utiliza el ratón para agregar objetos prefabricados en el lugar deseado dentro de la pantalla.

Es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos. Combina ambas tendencias. Ya que no podemos decir que Visual Basic

pertenezca por completo a uno de esos dos tipos de programación, debemos inventar una palabra que la defina: PROGRAMACION VISUAL.

## ***PROGRAMACIÓN ORIENTADA A OBJETOS***

Visual Basic se basa en la programación Orientada a Objetos (POO), la cual es una forma de programación que utiliza objetos (similares a los objetos del mundo real) para la solución de problemas. La POO permite descomponer un problema en bloques relacionados. Cada bloque pasa a ser un objeto autocontenido que posee sus propios datos e instrucciones. De esta manera, la complejidad se reduce y se pueden realizar programas más largos de una manera sencilla, mediante la combinación de varios bloques

## ***ELEMENTOS BÁSICOS DE LA POO***

Los elementos básicos de la programación orientada a objetos son:

- Objetos
- Propiedades
- Métodos
- Eventos
- Mensajes
- Clases

### **Objetos**

Un objeto es un ente o entidad que tiene atributos propios (propiedades) y unas formas de operar sobre ellos (métodos). Por tanto, un objeto contiene variables que especifican su estado y operaciones que definen su comportamiento.

Son ejemplos de objetos en Visual Basic: formularios, botones de comando, cuadros de texto, etiquetas, etc.



## **Propiedades**

Las propiedades representan las características del objeto. Hay propiedades particulares, como Caption que la poseen los botones de comando, por ejemplo, y genéricas como Name que la poseen todos los objetos.

## **Métodos**

Los métodos son funciones asociadas a un objeto; pero a diferencia de los procedimientos no son programadas por el usuario, sino que vienen ya programadas con el lenguaje de programación. Cada tipo de objeto tiene sus propios métodos.

## **Eventos**

Un evento es la capacidad de un objeto de reaccionar cuando ocurre una determinada acción (acción y reacción). Como respuesta a un evento se envía un mensaje y se ejecuta un determinado método (procedimiento).

Cada objeto responde a un conjunto de eventos. Como respuesta a un evento se ejecuta un determinado procedimiento que realiza la acción programada por el usuario para ese evento concreto.

## **Mensajes**

Un mensaje es una llamada a un método, de tal forma que cuando un objeto recibe un mensaje, la respuesta a ese mensaje es ejecutar el procedimiento asociado.

Cuando se ejecuta un programa orientado a objetos, los objetos están constantemente recibiendo, interpretando y respondiendo a mensajes de otros objetos.

## **Clases**

Una clase es una descripción para producir objetos de esa clase o tipo. Es decir, se trata de una generalización de un tipo específico de objetos. En otras palabras,

un objeto es una variable del tipo definido por una clase. Por ejemplo, piense en un molde para hacer pasteles, el molde es la clase y los pasteles los objetos.

## ***CARACTERÍSTICAS DE LA POO***

Las características fundamentales de la programación orientada a objetos son:

- Abstracción
- Encapsulamiento
- Herencia
- Polimorfismo

### **Abstracción**

La abstracción permite no detenerse en los detalles concretos del funcionamiento de las cosas, sino centrarse en los aspectos que realmente nos importan y son útiles en un determinado momento. En cierta medida, se podría decir que es “útese el objeto y olvídense de como funciona en forma interna”.

Por ejemplo, para manejar una computadora no se necesita saber como funcionan sus circuitos electrónicos, en términos de corriente, tensión, etc.

### **Encapsulamiento**

Esta característica permite ver un objeto como una “caja negra” autocontenida en la que se ha metido de alguna manera toda la información que maneja dicho objeto. Esto permite manipular los objetos como unidades básicas, permaneciendo oculta su estructura interna.

### **Herencia**

La herencia es la característica que permite compartir automáticamente propiedades y métodos entre objetos. Es decir, se pueden crear nuevas clases de objetos en base a clases existentes. Más concreto, un objeto puede heredar un conjunto general de propiedades y métodos a las que puede añadir aquellas

características que son específicas suyas. El usuario de Visual Basic no dispone de esta característica.

## **Polimorfismo**

Polimorfismo, cuyo significado es “*muchas formas*”, es la característica que permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la que se realiza la implementación. Esto hace posible que se puede acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando exactamente el mismo medio de acceso.

## **LOS OBJETOS DE VISUAL BASIC**


Los objetos de Visual Basic están divididos en dos tipos: controles y contenedores.

Los controles son un medio gráfico que permiten a los usuarios interactuar con la aplicación para la manipulación de datos y ejecución de tareas, como por ejemplo: las etiquetas, los cuadros de texto, los botones de comando, las casillas de verificación, los botones de opción, los cuadros de lista, los cuadros combinados, entre otros.

Los contenedores son objetos que pueden incluir otros objetos y permiten el acceso a los objetos que contienen. Entre ellos se tiene a los formularios, los marcos y las cuadrículas.

El formulario más los controles constituyen la interfaz de la aplicación.

## **CONOCIENDO EL VISUAL BASIC 6.0**

Para ejecutar el Visual Basic 6, proceda como con cualquier aplicación bajo Windows, haciendo click en el correspondiente Icono . Una vez realizado, se presenta la ventana Nuevo Proyecto la cual se muestra en la figura A-1; en ella se especifica el tipo de proyecto que desea crear. En esta ventana también se puede

abrir un proyecto existente (pestaña *Existente*) o uno de los proyectos abiertos recientemente (pestaña *Recientes*).

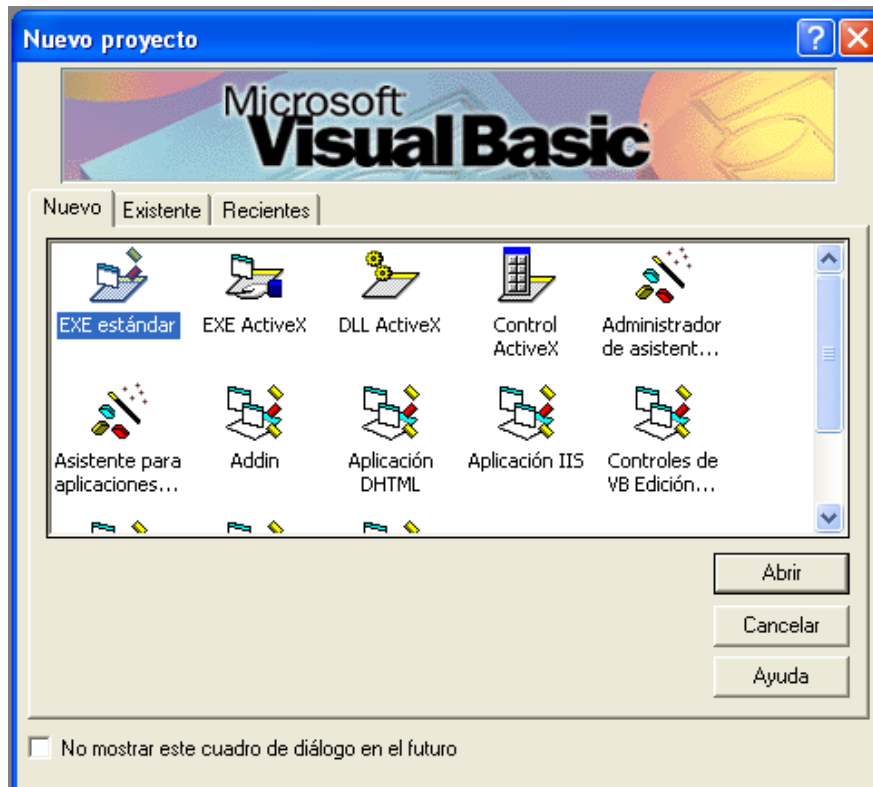


Figura A-1. Ventana Nuevo Proyecto.

Los tipos de proyectos son los siguientes:

- ❖ EXE estándar: genera un proyecto estándar ejecutable bajo el entorno Windows.
- ❖ EXE ActiveX: este tipo de proyecto se diseña para que ejecute tareas de una aplicación con múltiples niveles, los cuales corresponden a los antiguos controles OLE. A éstos los pueden acceder otros programas o ejecutarse con autonomía propia.
- ❖ DLL ActiveX: el proyecto se crea como un programa de ejecución remota como una DLL (Dynamic Link Lador), no tienen autonomía propia pero se ejecutan más rápidamente.
- ❖ Control ActiveX: proyecto para crear controles OCX personalizados para ser

utilizados posteriormente como controles estándar.

- ❖ Asistente para aplicaciones: proyecto para crear el esquema de una aplicación como si se tratara de una plantilla.

Una vez seleccionado el tipo de proyecto, que para nuestro caso por lo general será EXE estándar, se presenta el Entorno Integrado de Desarrollo de Aplicaciones del Visual Basic en modo de diseño (figura A-2).

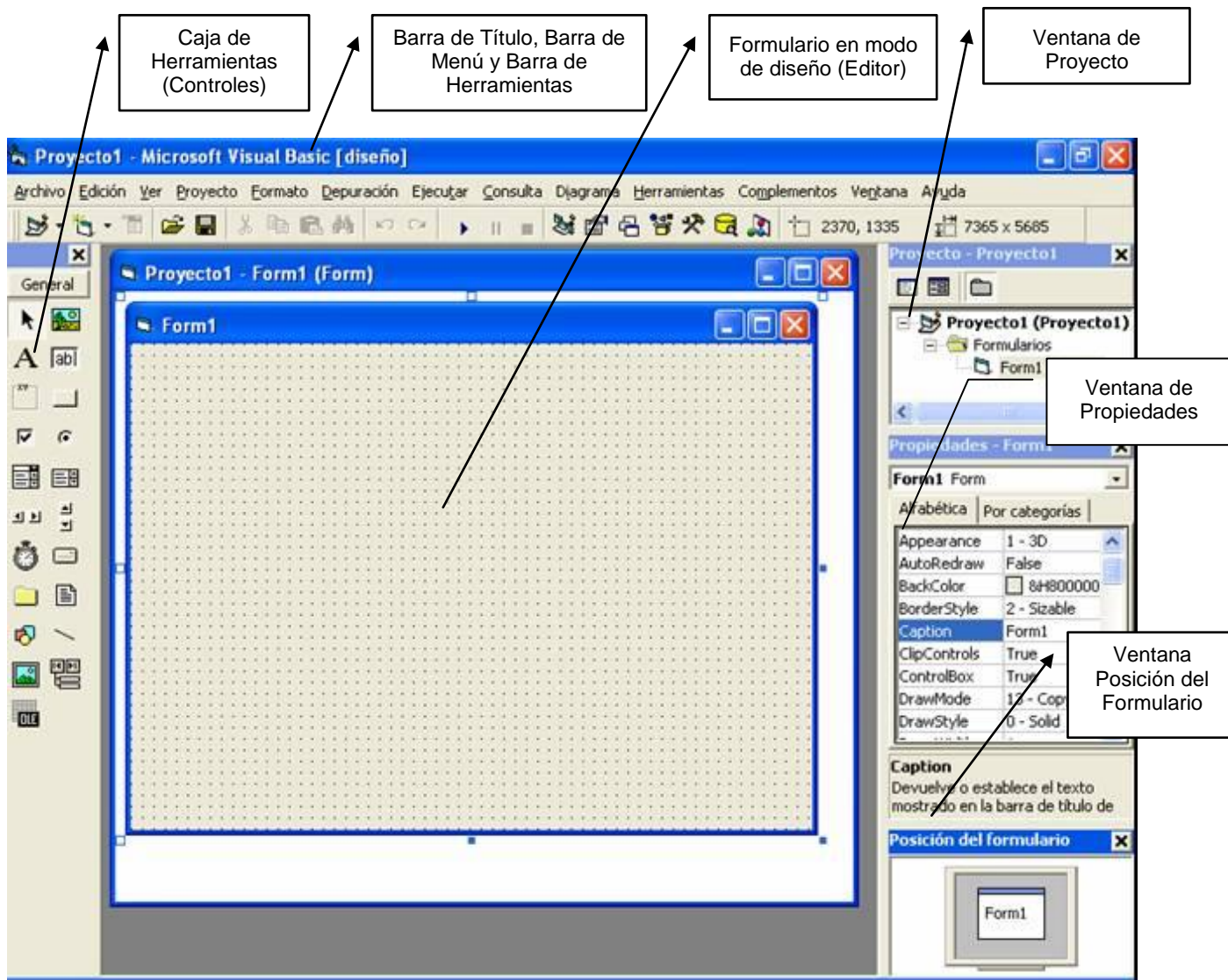


Figura A-2. Entorno Integrado de Desarrollo de Microsoft Visual Basic 6.0

## 1. Menús contextuales:

Contienen accesos directos a acciones que se realizan con frecuencia sobre los objetos. Para abrir un menú contextual, haga clic con el botón secundario del ratón en el objeto que está usando. La lista específica de opciones disponibles en el menú contextual depende de la parte del entorno en la que se hace clic con el botón secundario del ratón.

Por ejemplo, el menú contextual que aparece cuando hace clic con el botón secundario del ratón en el cuadro de herramientas le permite mostrar el cuadro de diálogo Componentes, ocultar el cuadro de herramientas, acoplar o desacoplar el cuadro de herramientas, o agregar una ficha personalizada al cuadro de herramientas.

## 2. Barras de herramientas:

Se encuentra después de la barra de menú (ver figura A-3). Proporcionan un rápido acceso a los comandos usados normalmente en el entorno de programación y que están disponibles en alguna opción del sistema de menú del Visual Basic. Haga clic en un botón de la barra de herramientas para llevar a cabo la acción que representa ese botón.


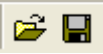
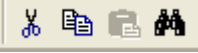



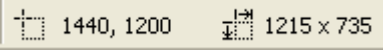
Son cuatro las barras de herramientas: Depuración, Edición, Editor de Formularios y Estándar. Por defecto sólo aparece la barra de herramientas Estándar, pero haciendo click con el botón derecho del ratón sobre la barra de herramientas, aparece un menú contextual con el que se puede seleccionar cualquiera de las barras. Las barras de herramientas se pueden acoplar debajo de la barra de menús o pueden “flotar” si selecciona la barra vertical del borde izquierdo y la arrastra fuera de la barra de menús.

La barra de herramientas Estándar simplifica la opciones mas usadas de los menús Archivo, Edición, Ver y Ejecutar.



Figura A-3. La Barra de Herramientas Estándar

De izquierda a derecha la Barra de Herramientas Estándar contiene los siguientes iconos:

- ❖  Agregar Nuevo Proyecto, Agregar Formulario y Editor de Menú
- ❖  Abrir Proyecto y Guardar Proyecto
- ❖  Los comandos de edición: Cortar, Copiar, Pegar y Buscar
- ❖  Deshacer y Rehacer escribir
- ❖  Los comandos de ejecución: Iniciar Ejecución de Programa, Interrumpir (Pausa) y Terminar Ejecución
- ❖  •Los comandos de ver Explorador de proyectos, Ventana de propiedades, Ventana posición del formulario, Examinador de objetos, Cuadro de herramientas, Ventana de vista de datos y el Manejador de componentes visuales.
- ❖  Posición (Left, Top) del objeto en el formulario o del formulario en la pantalla; y tamaño (ancho y largo) en píxel del objeto o formulario

### 3. Caja de herramientas:

En la parte izquierda del Entorno de Desarrollo de Visual Basic se encuentra la Caja de herramientas, la cual proporciona el conjunto de controles estándar de VB, además de los controles ActiveX y objetos insertables que se han agregado al proyecto. Estos controles se utilizan durante el diseño para crear la interfaz de usuario y establecer a través de ellos los mecanismos que gestionen las tareas de la aplicación.

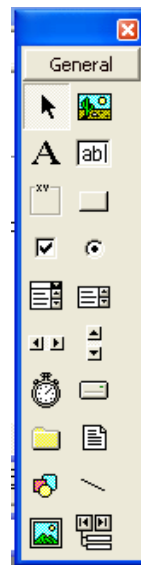










Figura A-4. Caja de Herramientas

Para introducir un control en el formulario simplemente hay que hacer clic con el botón izquierdo del ratón sobre el control deseado y colocarlo en el formulario con la posición y el tamaño deseado. Haciendo doble clic sobre el control es también otra forma de colocar el control en el formulario, quedando éste ubicado en el centro del formulario.

El número de controles que pueden aparecer en la caja de herramientas varían con la configuración del sistema, pero por defecto son 20 y los mismos se describen en la siguiente tabla:

Icono	Descripción
	<u>Puntero</u> : Sirve para seleccionar un control (objeto) sobre el formulario.
	<u>PictureBox</u> : control para mostrar una imagen gráfica.
	<u>Label</u> : muestra un texto como una etiqueta.
	<u>TextBox</u> : caja de texto, a través de la cual el usuario puede ingresar datos
	<u>Frame</u> : es un marco que sirve como contenedor de otros controles, en particular los <u>OptionButton</u>
	<u>CommandButton</u> : son botones de órdenes que activan un proceso cuando se presionan.
	<u>CheckBox</u> : casilla de verificación, los cuales permiten la elección de una varias de las operaciones, opciones o propiedades que representan (no excluyentes)
	<u>OptionButton</u> : botón de opción que permiten la elección o selección de uno y















	sólo uno de los atributos u opciones que representa.
	<u>ComboBox</u> : cuadro combinado de texto, es un cuadro de texto (TextBox) combinado con las características de una lista (ListBox) que contiene los elementos a ser seleccionados.
	<u>ListBox</u> : cuadro de lista, muestra una lista de elementos entre los cuales se puede seleccionar uno o más.
	<u>HscrollBar</u> : barra de desplazamiento horizontal, permite explorar fácilmente una larga lista de elementos basado en la posición horizontal que ocupa en la barra un cuadro (ascensor) que se desplaza de izquierda a derecha.
	<u>VscrollBar</u> : barra de desplazamiento vertical, igual que el anterior pero la barra se desplaza de arriba hacia abajo.
	<u>Timer</u> : es un temporizador o cronómetro que permite ejecutar código a intervalos periódicos de tiempo.
	<u>DriveListBox</u> : cuadro de lista de unidades, es una lista que contiene las unidades de disco conectadas a MiPC, mediante la cual se puede seleccionar o activar una de ellas.
	<u>DirListBox</u> : cuadro de lista de directorios, es una lista que contiene los directorios o carpetas de la unidad de disco activa, mediante la cual se puede seleccionar o activar uno de ellos.
	<u>FileListBox</u> : cuadro de lista de archivos, es una lista que contiene los archivos del directorio o carpeta actual, mediante la cual se puede seleccionar uno o varios de ellos.
	<u>Shape</u> : permite incluir formas geométricas al formulario (rectángulo, cuadrado, elipse, círculo, rectángulo redondeado o un cuadrado redondeado).
	<u>Line</u> : permite incluir una línea horizontal, vertical o diagonal al formulario.
	<u>Image</u> : control para mostrar una imagen gráfica.
	<u>Data</u> : permite acceder los registros de una base de datos enlazando los elementos de la misma a los diferentes controles de un formulario.

Tabla A-1. Descripción de los controles de la Caja de Herramientas

#### 4. Ventana de formulario.

En la parte central está el Formulario (Form) en modo de diseño. Es el área sobre el cual se colocaran los objetos (controles) para diseñar la interfaz de *usuario de la aplicación*. *Al ejecutar el programa (MODO DE EJECUCIÓN)*, el *Form* se convertirá en la ventana de la aplicación donde aparecerán los botones, las cajas de texto, los gráficos, etc

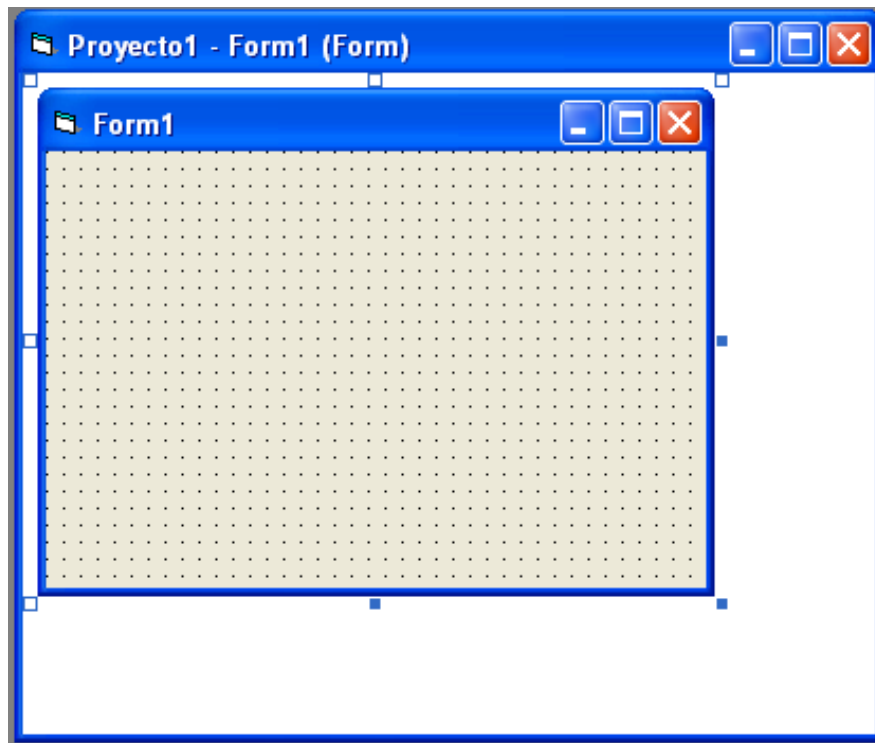


Figura A-5. Ventana formulario.

El conjunto de puntos que aparecen sobre el formulario se llama malla o grilla 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 la malla automáticamente desaparece. En la figura A-5 se observa esta ventana con el fondo y la malla que proporciona VB en forma estándar.

##### 5. Ventana Explorador de proyectos:

En la parte superior del lado derecho se encuentra el *Explorador del proyecto*, que muestra los elementos involucrados en el proyecto, tales como formularios, módulos de programas, controles .ocx, etc. Se trata de una lista jerárquica de los proyectos así como de los elementos contenidos y referenciados por cada uno de los proyectos (figura A-6).

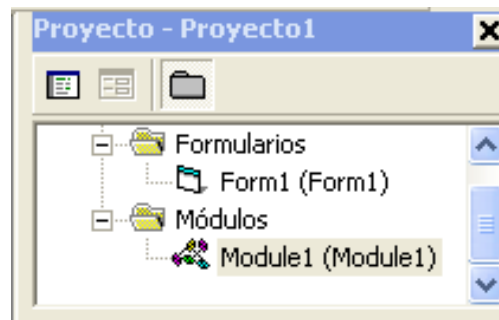


Figura A-6. Ventana Explorador de Proyecto.

## 6. Ventana Propiedades:

Debajo del Explorador del proyecto se encuentra la *Ventana de propiedades* (fig A-7.), en la que se muestran las características o propiedades del objeto seleccionado sobre el formulario o el propio formulario.

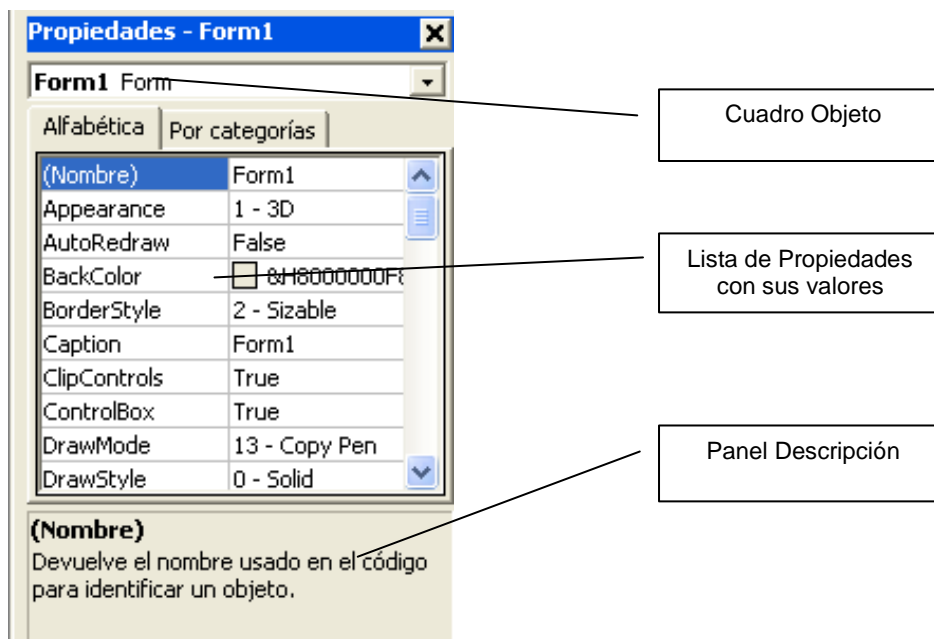


Figura A-7. Ventana de Propiedades.

Una propiedad es una característica de un objeto, como su tamaño, título o color. En esta ventana se enumeran los valores actuales de las propiedades del control o formulario seleccionado. Puede cambiar las propiedades de uno o más objetos en tiempo de diseño seleccionando el nombre de la propiedad y escribiendo o seleccionando un nuevo valor. Cuando se selecciona varios

controles sólo se muestran las propiedades comunes a todos los objetos seleccionados.

### Elementos de la ventana PROPIEDADES:

- Cuadro Objeto: presenta el objeto seleccionado actualmente del formulario activo.
- Ficha lista de propiedades:
  - Ficha Alfabética: presenta por orden alfabético todas las propiedades del objeto seleccionado que se pueden cambiar en tiempo de diseño, así como sus respectivos valores.
  - Ficha Por Categorías: enumera las propiedades del objeto seleccionado clasificadas por categorías, por ejemplo las propiedades Backcolor, Caption y Forecolor pertenecen a la categoría Apariencia.
- Panel Descripción: muestra el tipo de propiedad y una breve descripción de la propiedad seleccionada.

### 7. Ventana Posición del formulario:

La ventana Posición del formulario (fig. A-8) le permite fijar la ubicación que tendrá el formulario en la pantalla cuando se ejecute la aplicación, utilizando una pequeña representación gráfica de la pantalla.

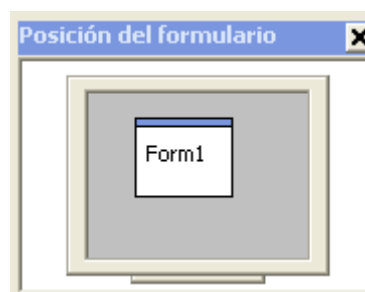



Figura A-8. Ventana Posición del Formulario.

Esta ventana muestra todos los formularios visibles en el entorno. Cuando se coloca el curso en uno de ellos, su forma cambia a . Presionando el botón izquierdo del ratón puede colocar el formulario en la posición deseada.

## 8. Ventana de código:

Funciona como un editor para escribir el código de la aplicación. Se crea una ventana editor de código diferente para cada formulario o módulo del código de la aplicación. En la figura A-9 se muestra esta ventana,

Puede abrir una ventana *Código* desde:

- la ventana Proyecto, seleccionando un formulario o módulo y eligiendo el botón Ver código.
- una ventana Formulario, haciendo doble clic sobre un control o formulario.
- eligiendo Código desde el menú Ver o presionando F7.

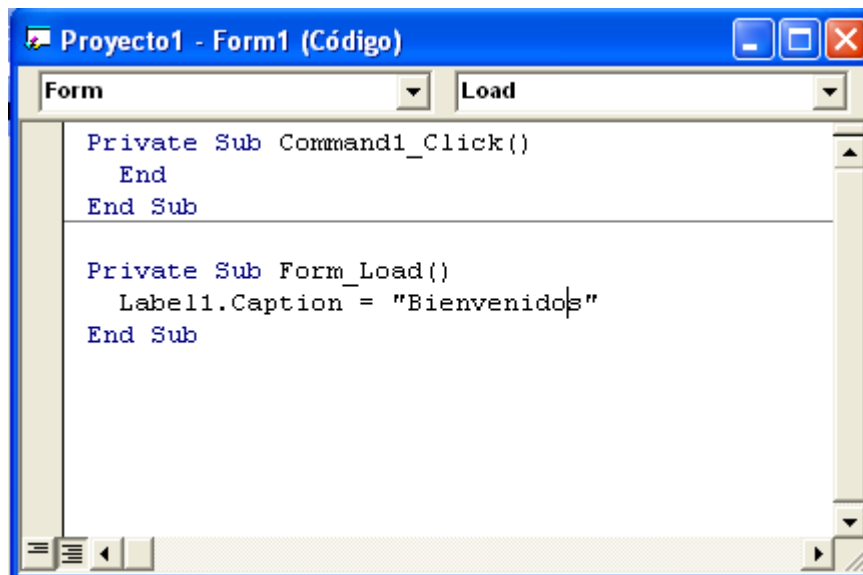


Figura A-9. Ventana Editor de código

Este editor utiliza un código de colores para hacer diferencia entre líneas de código. Así, el código escrito por el usuario aparece en negro, las palabras clave de Basic en azul, los comentarios en verde, los errores en rojo. Esta simple ayuda visual permite detectar y corregir problemas con más facilidad.


En la parte superior de la ventana aparecen dos listas desplegables. La de la izquierda (objeto) corresponde a los distintos controles del formulario. La lista desplegable de la derecha (Procedimiento/Evento) muestra los distintos

procedimientos que se corresponden con el elemento seleccionado en la lista de la izquierda.

Elementos de la ventana CÓDIGO:

- Cuadro Objeto: Es una lista desplegable que muestra el nombre del objeto seleccionado. Haga clic en la flecha situada a la derecha para mostrar una lista de todos los objetos asociados con el formulario, el cual puede seleccionarse desde esta lista y así modificar su código. En la figura A-9 se observa que el objeto seleccionado es el Formulario (Form).
- Cuadro Procedimiento/Evento: Presenta una relación de todos los eventos reconocidos por Visual Basic para un formulario o control mostrado en el cuadro Objeto. Cuando seleccione un evento, el procedimiento de evento asociado con ese nombre de evento se muestra en la ventana Código. En la figura se observa que el evento seleccionado es el Load.

### 9. Examinador de objetos:

Enumera los objetos disponibles que puede usar en su proyecto y le proporciona una manera rápida de desplazarse a través del código. Puede usar el Examinador de objetos para explorar objetos en Visual Basic y otras aplicaciones, ver qué métodos y propiedades están disponibles para esos objetos, y pegar código de procedimientos en su aplicación. Esta ventana no está visible por defecto, para abrirla vaya a Ver – Examinador de Objetos o presione el botón  en la barra de Herramientas.

## ***UN PROYECTO EN VISUAL BASIC.***

Un proyecto en Visual Basic es la forma en que se almacena una aplicación diseñada con Visual Basic, está compuesta por una o varias interfaces de usuario, estas la conforman los formularios (archivos .frm), dentro de los cuales se sitúan los controles. Estos controles son archivos .ocx que también forman parte del proyecto.

Además de los formularios, forman también parte del proyecto el conjunto de procedimientos y funciones que responden a los eventos que se generan sobre los controles situados en la interfaz de usuario en tiempo de ejecución, así como también, todos los módulos que son precisos para otras tareas, como puede ser las de validación de datos (archivos .bas).

Existe otro conjunto de archivos que se deben incluir como un sistema de ayuda a la aplicación, mediante los cuales se le suministra información sobre como utilizar la aplicación, es decir, las diversas tareas que lo conforma (archivos .hlp, .txt o .rtf).

Lo más usual es que la aplicación maneje datos en forma de archivos o de base de datos (archivos .dat, .mdb o cualquier otro formato).

Con frecuencia se insertan en los proyectos imágenes, iconos, gráficos, música y sonidos para facilitar las tareas de selección y darle adorno a la misma (archivos .wfm, .bmp, .jpg, .mid, .wav, .mp3, etc).

En resumen un proyecto en Visual Basic está formado por un conjunto de archivos (datos y objetos asociados con el mismo), los cuales almacenan los diferentes objetos que conforman la aplicación, así como también información sobre las opciones del entorno. Estas especificaciones se almacenan en un archivo con extensión .vbp que se actualiza cada vez que se guarda el proyecto.

## ***DISEÑO DE UN PROGRAMA EN VISUAL BASIC***

La creación de un programa bajo Visual Basic lleva los siguientes pasos:

**Análisis.** Es el estudio de las necesidades que han dado origen a la creación de ese programa. Es lo que se llama *Análisis* de la aplicación. Es la primera fase que debe tener siempre un programa y es también la más olvidada entre los programadores. Una aplicación no se inicia con el teclado, sino sobre un papel.

**Creación de una interfase de usuario.** Esta interfase será la principal vía de comunicación hombre máquina, tanto para salida de datos como para entrada. Será necesario partir de una o varias ventanas - *Formularios* - a las que le iremos añadiendo los controles necesarios.

**Definición de las propiedades de los controles** – Se dará la forma, posición, y todas las características necesarias a los controles que hayamos colocado en ese formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.

**Generación del código asociado a los eventos que ocurran a estos controles.** A la respuesta a estos eventos (click, doble click, una tecla pulsada, etc.) le llamamos Procedimiento, y deberá generarse de acuerdo a las necesidades del programa.

**Generación del código del programa.** Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo, VB ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Estos Procedimientos no responden a un evento acaecido a un control o formulario, sino que responden a un evento producido durante la ejecución del programa.



## ***EL FORMULARIO.***

Es el objeto principal del Visual Basic, sobre el se colocan los diferentes controles para formar la interfaz de usuario. Las principales propiedades, eventos y métodos de un formulario se muestran en las tablas A-2, A-3 y A-4.

<b>Propiedad</b>	<b>Descripción</b>
Name	nombre usado para identificar el formulario
BackColor	Devuelve o establece el color de fondo del formulario
BorderStyle	Devuelve o establece el estilo del borde del formulario
Caption	Texto en la barra de título del formulario
ControlBox	True/False. Determina si el formulario tiene o no el cuadro de control
Enable	True/False. Determina si está habilitado para responder a las acciones del usuario
Font	Determina la fuente que se utiliza para el formulario y todos sus objetos
ForeColor	Devuelve o establece el color del texto del formulario y todos sus objetos
Height	Devuelve o establece el alto en píxel del formulario
Icon	Icono que se muestra cuando el formulario está minimizado
Left	Devuelve o establece la distancia en píxel desde el borde izquierdo de la pantalla hasta el borde izquierdo del formulario
MaxButton	True/False. Determina si el formulario tiene o no el botón Maximizar
MinButton	True/False. Determina si el formulario tiene o no el botón Minimizar
Top	Devuelve o establece la distancia en píxel desde el borde superior de la pantalla hasta el borde superior del formulario
Visible	True/False. Determina si está el formulario se muestra o no
Width	Devuelve o establece el ancho en píxel del formulario
WindowState	Estado inicial del formulario (normal, maximizado o minimizado)

Tabla A-2. Principales propiedades de un formulario

<b>Evento</b>	<b>Descripción</b>
Activate	Ocurre cuando el formulario se convierte en la ventana activa
Click	Ocurre cuando el usuario hace clic sobre el formulario
DbClick	Ocurre cuando el usuario hace dos veces seguidas clic sobre el formulario
Deactivate	Ocurre cuando el formulario deja de ser la ventana activa
Load	Ocurre cuando se carga un formulario
Unload	Ocurre cuando un formulario está a punto de descargarse

Tabla A-3. Principales eventos de un formulario

Método	Descripción
Hide	Oculto el formulario
Refresh	Fuerza a que se vuelva a dibujar el control sobre el formulario.
SetFocus	Sitúa el enfoque sobre la caja de texto
Show	Hace visible el formulario

Tabla A-4. Principales métodos de un formulario

## CONTROLES BÁSICOS.

Los controles básicos se refiere a aquellos que utilizamos con mas frecuencia en el diseño de nuestras aplicaciones y que están entre los 20 estándares de la caja de herramientas. A continuación se describirá los tres principales controles usados para crear las aplicaciones, para cada uno de ellos se dará una descripción de las principales propiedades, eventos y métodos.

### 1. ETIQUETAS (LABEL).

Es un control que se usa para mostrar texto que el usuario no podrá cambiar directamente, aunque pueden ser cambiados en modo de ejecución. También puede usar una etiqueta para identificar a otro control. En la tabla A-5 se muestran las principales propiedades del control etiqueta (Label)

Propiedad	Descripción
Name	Nombre usado para identificar el objeto
Aligment	Devuelve o establece la alineación del texto en la etiqueta
Appearance	Devuelve o establece si la etiqueta se dibuja con efecto 3D
Autosize	Determina si la etiqueta se ajusta automáticamente para mostrar todo el contenido
BackColor	Devuelve o establece el color de fondo usado para mostrar el texto
BackStyle	Indica si el fondo de la etiqueta es transparente u opaco
BorderStyle	Devuelve o establece el estilo del borde de la etiqueta
Caption	Devuelve o establece el texto que se muestra en la etiqueta
Font	Determina la fuente que se utiliza para el texto de la etiqueta (tipo, estilo y tamaño)
ForeColor	Devuelve o establece el color del texto de la etiqueta
Height	Devuelve o establece el alto en píxel de la etiqueta
Index	Devuelve o establece el número que identifica un control en una matriz de controles
Left	Devuelve o establece la distancia en píxel desde el borde izquierdo del formulario hasta el borde izquierdo de la etiqueta
Top	Devuelve o establece la distancia en píxel desde el borde superior del formulario hasta el borde superior de la etiqueta

Visible	Determina si la etiqueta se muestra o no en tiempo de ejecución
Width	Devuelve o establece el ancho en píxel de la etiqueta

Tabla A-5. Principales propiedades de una etiqueta

## 2. CUADROS DE TEXTO (TEXTBOX).

También denominado control de campo de edición o control de edición de texto, es un control que se usa para introducir información en tiempo de ejecución (entrada de datos). En las tablas A-6, A-7 y A-8 se muestran respectivamente las principales propiedades, eventos y métodos del control cuadro de texto (TextBox).

Propiedad	Descripción
Name	Nombre o identificador usado para identificar el objeto
Aligment	Devuelve o establece la alineación del texto en el cuadro de texto
Appearance	Devuelve o establece si la etiqueta se dibuja con efecto 3D
BackColor	Devuelve o establece el color de fondo usado para mostrar el texto
BorderStyle	Devuelve o establece el estilo del borde del cuadro de texto
Enable	Determina si el cuadro de texto responde o no a eventos generados por el usuario
Font	Determina la fuente que se utiliza para el cuadro de texto (tipo, estilo y tamaño)
ForeColor	Devuelve o establece el color del texto del cuadro de texto
Height	Devuelve o establece el alto en píxel del cuadro de texto
Index	Devuelve o establece el número que identifica un control en una matriz de controles
Left	Devuelve o establece la distancia en píxel desde el borde izquierdo del formulario hasta el borde izquierdo del cuadro de texto
Multiline	Devuelve o establece si el control acepta o no múltiples líneas de texto
TabIndex	Devuelve o establece el orden de tabulación del cuadro de texto dentro del formulario
ScrollBars	Devuelve o establece un valor que indica si la caja de texto tiene barras de desplazamiento horizontal o vertical. Puede tomar los valores 0- None (sin barra), 1- Horizontal (barra horizontal), 2- Vertical (barra vertical) y 3- Both (ambas barras).
Text	Devuelve o establece el texto contenido en el control
ToolTipText	Devuelve o establece el texto que se muestra cuando el ratón se sitúa sobre el control
Top	Devuelve o establece la distancia en píxel desde el borde superior del formulario hasta el borde superior del cuadro de texto
Width	Devuelve o establece el ancho en píxel del cuadro de texto

Tabla A-6. Principales propiedades de un cuadro de texto

Evento	Descripción
Change	Ocurre cuando se modifica el texto en el área de edición de la caja de texto
Click	Ocurre cuando el usuario presiona y suelta el botón izquierdo del ratón sobre la caja de texto
DbIcClick	Ocurre cuando el usuario hace dos veces seguidas clic sobre la caja de texto
GotFocus	Ocurre cuando la caja de texto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiar el enfoque en el código mediante el método SetFocus.
LostFocus	Ocurre cuando el control pierde el enfoque.

Tabla A-7. Principales eventos de un cuadro de texto

Método	Descripción
Refresh	Fuerza a que se vuelva a dibujar el control sobre el formulario.
SetFocus	Sitúa el enfoque sobre la caja de texto

Tabla A-8. Principales métodos de un cuadro de texto

### 3. BOTON DE COMANDO (COMMANDBUTTON).

Un botón de comando es un control que permite ejecutar una acción o un proceso de la aplicación. En las tablas A-9 y A-10 se muestran las principales propiedades y eventos asociados con el control botón de comando (CommandButton)

Propiedad	Descripción
Name	nombre o identificador usado para identificar el objeto
Appearance	Devuelve o establece si la botón se dibuja con efecto 3D
BackColor	Devuelve o establece el color de fondo usado para mostrar el texto
Cancel	Indica si el botón es el botón de Cancelar del formulario, es decir, que botón se activa cuando se presiona la tecla "Escape"
Caption	Devuelve o establece el texto que aparece en el botón
Default	Establece que botón es el botón predeterminado del formulario, es decir, que botón se activa cuando se presiona la tecla "Enter"
Enable	Determina si el botón de comando responde o no a eventos generados por el usuario
Font	Determina la fuente que se utiliza para el texto que se muestra en el botón de comando (tipo, estilo y tamaño)
Height	Devuelve o establece el alto en píxel del botón de comando
Index	Devuelve o establece el número que identifica el control en una matriz de controles
Left	Devuelve o establece la distancia en píxel desde el borde izquierdo del formulario hasta el borde izquierdo del botón de comando
Picture	Devuelve o establece el gráfico que se muestra en el botón en caso de que el estilo del mismo sea gráfico
Style	Devuelve o establece el estilo del botón: estándar o gráfico
TabIndex	Devuelve o establece el orden de tabulación del cuadro de texto dentro del formulario
ToolTipText	Devuelve o establece el texto que se muestra cuando el ratón se sitúa sobre el control

Top	Devuelve o establece la distancia en píxel desde el borde superior del formulario hasta el borde superior del cuadro de texto
Visible	Devuelve o establece si el botón de comando está visible o no.
Width	Devuelve o establece el ancho en píxel del cuadro de texto

Tabla A-9. Principales propiedades de un botón de comando

Evento	Descripción
Change	Ocurre cuando se modifica el texto en el área de edición de la caja de texto
Click	Ocurre cuando el usuario presiona y suelta el botón izquierdo del ratón sobre la caja de texto
DbIcClick	Ocurre cuando el usuario hace dos veces seguidas clic sobre la caja de texto
GotFocus	Ocurre cuando la caja de texto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiar el enfoque en el código mediante el método SetFocus.
LostFocus	Ocurre cuando el control pierde el enfoque.

Tabla A-10. Principales eventos de un botón de comando

## ***EL LENGUAJE***

Visual Basic es un lenguaje de programación basado en el lenguaje BASIC, al cual se le incorpora la funcionalidad de la programación orientada a objetos.

El código de Visual Basic se almacena en módulos, donde cada uno está subdividido en distintas secciones, una para cada objeto del módulo. Cada sección de código puede contener uno o más procedimientos, formados por declaraciones de constantes y variables, expresiones, sentencias de control y llamadas a procedimientos y/o funciones.

En esta parte del anexo vamos a analizar los diferentes elementos que intervienen al momento de realizar el código para una aplicación.

## **IDENTIFICADORES**

Los identificadores son nombres dados a los elementos de una aplicación, tales como constantes, variables, procedimientos, funciones, objetos, etc. Un identificador es una secuencia de caracteres que puede ser de hasta 255 caracteres. Para la construcción de identificadores debemos tener presente las siguientes reglas:

- Deben comenzar con una letra y no puede contener espacios en blanco.
- Letras, dígitos y caracteres subrayados están permitidos después del primer carácter.
- No se puede utilizar una palabra reservada como identificador. Una palabra reservada tiene un significado especial para Visual Basic. Por ejemplo INTEGER, el cual es un tipo de dato.

## CONSTANTES

Una constante almacena un dato cuyo valor no cambia durante la ejecución de un programa. Para declarar una constante utilice la siguiente sintaxis:

```
Const NOMBRE_CONSTANTE [As TIPO] = VALOR
```

Si no se declara el tipo de constante (utilizando As TIPO) se asigna a la constante el tipo de dato más apropiado a su valor.

El valor de una constante puede ser numérico, alfanumérico, carácter o de tipo fecha y hora. Por ejemplo:

```
Const MAXIT = 25
Const PI As Double = 3.141592
Const CADENA As String = "Visual Basic es fácil"
Const FECHA_POR_DEFECTO = #01/01/99#
```

## VARIABLES

Una variable almacena un dato cuyo valor puede cambiar durante la ejecución de un programa.

Por defecto Visual Basic no exige que se declaren las variables, es decir que se le diga al sistema cuales variables y el tipo de las variables que se utilizaran en el programa.

Mediante las declaraciones, el programa sabe de que tipo de dato se trata y por tanto cómo debe trabajar con él. En otros lenguajes de programación distintos de Basic, es necesario realizar esta declaración antes de introducir una variable.

Esta particularidad de no necesitar declarar las variables hace que sea sencillo introducir una variable nueva. Sin embargo acarrea el gran peligro de escribir mal el nombre de las variables y por lo tanto de utilizar valores erróneos o nulos para los datos.

El peligro de no declarar las variables no termina en esa posibilidad de error. Cuando no declaramos el tipo de variables, Visual Basic supone que la variable es de un tipo que soporta cualquier tipo de datos: una variable tipo **Variant**. En ese tipo podrá considerar cualquier tipo de variable, pero a costa de ocupar mucho mas espacio en la memoria que si la hubiese declarado con el tipo correcto para el dato que se va a almacenar en ella.

### Instrucción OPTION EXPLICIT.

Obliga a declarar previamente las variables que se vayan a usar. Esta declaración debe ponerla al comienzo de la sección de declaraciones de cada formulario y módulo que contenga su aplicación. Pero para evitar un olvido basta que lo programe en las opciones del VB, haciendo click en *Herramientas | Opciones* de la barra de menú del VB y le aparecerá la ventana Opciones (figura A-10), donde debe marcar las casilla *Requerir declaración de variables*. A partir de ahora, será VB quien ponga la sentencia Option Explicit en el lugar correcto.

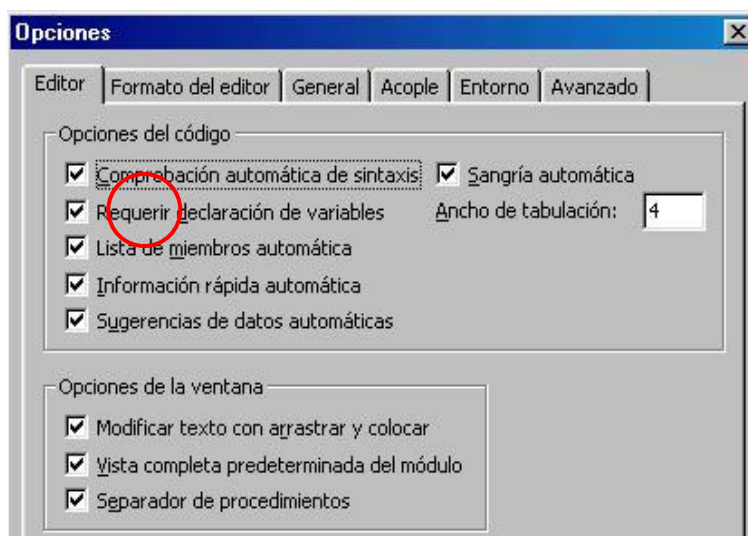


Figura A-10. Ventana Opciones.

## TIPOS DE VARIABLES

En la tabla A-11 se muestra los diferentes tipos de datos que pueden tomar las variables.

TIPO DATO	DE	TAMAÑO EN BYTES	COMENTARIOS
Booleana		(2)	Admite los valores 0 para True (verdadero) y 1 para False (falso)
Byte		(1)	Números enteros, en el rango de 0 a 255
Integer		(2)	Números enteros en el rango de -32768 a 32767
Long		(4)	Números enteros en el rango de -2147483648 a 2147483647
Single		(4)	Punto flotante, simple precisión
Doble		(8)	Punto flotante, doble precisión.
Currency		(8)	Entero, con punto decimal fijo (Típico de monedas)
String		(*)	Cadenas de caracteres alfanuméricas de longitud variable o fija
Date		(8)	Fechas
Object		(4)	Referencia a objetos
Variant		(**)	Otros tipos de datos

Tabla A-11. Tipos de datos

(\*) Una variable tipo String ocupa el mismo número de bytes que caracteres tenga la cadena.

(\*\*) Una variable tipo Variant ocupa 16 bytes si se trata de un número y 22 bytes + longitud de la cadena si se trata de un dato tipo cadena de caracteres.

Una cadena de caracteres se refiere a un dato que puede contener letras, dígitos o cualquier carácter especial incluidos los espacios, sin embargo, hay ocasiones en que un dato que pareciera ser numérico como por ejemplo el número de Cédula de Identidad, un código postal, el número de una calle o el piso de un edificio se manejan como cadenas de caracteres. Estas variables, aunque su contenido sean números, se deben declarar como cadenas de caracteres (string), ya que en realidad el número no representa una cantidad numérica, sino una etiqueta.

Las variables booleanas (True/False) pueden en muchos casos sustituirse por una variable del tipo Byte. Si ese datos True / False se va a introducir en una base de datos o en archivo en el disco, puede ser mas prudente poner 0 en vez de False y 1 en vez de True.



Una variable byte ocupa muy poco, simplemente 1 byte como su nombre indica. Pero no puede contener números mayores de 255 ni números negativos.

Cada vez que declare una variable numérica piense en los valores que puede tener, sobre todo cuando esa variable va a ser el resultado de una operación matemática. Recuerde el escaso margen de una variable tipo Integer ( de -32768 a 32767).

Si la aplicación va a manipular monedas, piense en la forma de expresar los números decimales y el número de ellos permitidos, así como el redondeo. La variable correcta para este caso es Currency, pero Currency le añade automáticamente el tipo de moneda de cada país. Los datos del tipo de moneda los toma del sistema operativo, es decir, depende de cómo está configurado el computador. Por lo que no se extrañe si los datos tipo moneda los expresa en euros o en dólares.

Otro tipo de variable es Date. Este tipo de variable representa una fecha o una hora. Pero debe ser objeto de un estudio un poco más extenso para tener claro que es una fecha para Visual Basic, como las trata y como se pueden comparar.

### **Ámbito de las variables.**

Denominamos ámbito de una variable al área o parte del programa donde esa variable está declarada y puede usarse.

Un programa Visual Basic tiene uno o varios formularios. Cada formulario tiene varios controles. Tanto el formulario como cada uno de sus controles tienen una parte del programa, justamente la parte relacionada con cada uno de los eventos que pueden suceder bien al formulario o a los controles. A estas partes las habíamos llamado Procedimientos. Podemos tener procedimientos que no estén relacionados con ningún evento ocurrido al formulario o a sus controles. (Los Procedimientos que iremos insertando a lo largo de la aplicación).

Aparte de formularios y controles, un programa puede tener Módulos, y en cada uno de los módulos podemos insertar cuantos Procedimientos y Funciones

queramos. Si se declara una variable dentro de un procedimiento o Función, esa variable es interna del Procedimiento o Función donde se declaró. El procedimiento puede estar en un Formulario (Cualquier procedimiento de un control o un procedimiento creado por nosotros) o en un Módulo (En este caso, el procedimiento solo puede ser creado por nosotros).

En un Formulario, una variable puede declararse de dos formas: *Privada* o *Pública*. Para declarar una variable a nivel de formulario debe hacerse en la sección de declaraciones, la cual está en la ventana de código *Objeto = General, Proc. = Declaraciones*. Si se declara *Privada*, esa variable se puede ver en todo el formulario, (es decir, en todos los procedimientos de todos los controles del formulario y en los Procedimientos que pudiésemos insertar en ese formulario), pero no sale de dicho formulario. Si se declara como *Pública*, esa variable puede verse por todo el formulario, de la misma forma que lo haría declarada como *Privada*, y además puede ser usada desde otro Formulario o Módulo, accediéndola con el nombre del Formulario, seguido del nombre de la variable (*Formulario.Variable*)

En un Módulo una variable puede declararse como *Privada*, con lo que no saldrá de ese Módulo, o *Pública*, pudiendo en este caso usarse en todo el programa. Cuando se declara una variable como pública en un Módulo, basta referirse a ella por su nombre, sin citar el nombre del Módulo donde se declaró.

### **Tipos de declaración de variables.**

- **Sentencia DIM:** Es la forma mas común de declarar una variable como Privada. Puede emplearse en un Procedimiento, Función, Formulario o Módulo. La sintaxis es de la siguiente forma:

**Dim nombrevariable As tipo**

Declarando una variable con la sentencia DIM, en un formulario, función, procedimiento o módulo, el ámbito de la variable será el explicado anteriormente para una variable declarada como Privada. Es decir, esa

variable no sale del formulario, procedimiento ó módulo donde se declaró. Cada vez que entremos al formulario, procedimiento o módulo, esa variable tomará el valor cero (si es numérica) o nulo (si es string).

- *Sentencia PRIVATE:* Es la forma de declarar una variable como Privada. Puede emplearse solamente en la sección de declaraciones de un Formulario o Módulo. La sintaxis es de la siguiente forma:

**Private** nombrevariable **As** Tipovariable

Declarando una variable mediante la sentencia PRIVATE en un Formulario o Módulo, esa variable puede usarse en todo ese Formulario o Módulo (En todos sus Procedimientos y Funciones), pero NO fuera del Formulario o Módulo donde se declaró.

La sentencia Private no puede usarse en un procedimiento o función.

- *Sentencia PUBLIC:* Es la forma de declarar una variable como Pública. Puede emplearse solamente en la sección de declaraciones de un Formulario o Módulo. La sintaxis es de la siguiente forma:

**Public** nombrevariable **As** Tipovariable

Declarando una variable de esta forma en la sección de declaraciones de un Módulo, esa variable puede usarse en cualquier parte del programa citándola simplemente por su nombre.

Si se declara de esta forma en la sección de declaraciones de un Formulario, esa variable puede usarse en toda el programa. Para nombrarla, si estamos en el Formulario donde se declaró basta con citarla por su nombre. Si no estamos en ese Formulario, habrá que citarla por el nombre del Formulario, seguido del nombre de la variable, separados por un punto:

NombreFormulario.Nombrevariable

En un Módulo puede usarse también la sentencia Global en vez de Public :

- Sentencia GLOBAL: Declara una variable que es válida en todo el programa. La sintaxis es:

**Global** nombrevariable **As** tipovvariable

La sentencia Global sólo puede usarse en el apartado de declaraciones de un Módulo. *Esta sentencia se mantiene por compatibilidad con versiones anteriores de VB. No es recomendable usarla.*

- Sentencia STATIC: Como se dijo anteriormente, una variable declarada en un procedimiento pierde su valor al salir de él. Lo peor es que una vez que el programa vuelva a entrar en ese procedimiento, la variable estará puesta a cero. Afortunadamente, esto último tiene solución. Si declarásemos una variable en un procedimiento o función, como estática, esa variable, aunque no la podremos utilizar fuera de ese procedimiento o función, cuando volvamos a él conservará el valor que tenía cuando lo abandonamos. Esta declaración como estática se realiza mediante la instrucción Static

**Static** nombrevariable **As** tipovvariable

El nombre de una variable puede ser tan largo como queramos. hasta un máximo de 40 caracteres. Es indiferente usar mayúscula ó minúsculas. El nombre de una variable siempre debe comenzar por una letra.

No hay problema por utilizar variables largas. Al compilar el programa no se lleva el nombre, es decir, no le va a ocupar mas espacio. Utilice siempre nombres que le definan la variable con algún sentido gramatical. Es muy útil a la hora de acordarse como se llaman.

## OPERADORES

Los operadores son símbolos que indican las diferentes operaciones que se efectuaran sobre los datos. Los operandos son los datos (constantes y/o variables) sobre los que actúa los operadores para producir un determinado resultado. Los operadores se clasifican en:

- Operadores aritméticos, los cuales se usan con datos numéricos y producen un resultado numérico. En la tabla A-12 se muestra estos operadores y el orden de en que están en la misma indica el orden de ejecución de ellos.
- Operadores de relación, los cuales se usan para comparar (en cuanto al orden) datos numéricos o alfanuméricos, produciendo un resultado booleano (verdadero o falso). En la tabla A-13 se muestran estos operadores, los cuales tienen igual orden de ejecución.
- Operadores lógicos, se usan para comparar valores booleanos produciendo un resultado booleano. En la tabla A-14 se muestran los operadores lógicos.
- Operadores para manejo de caracteres (tabla A-15)

Operador	Significado
^	Exponenciación
-	Menos unario
* /	Multiplicación y división
\	División entera
Mod	Resto de una división entera
+ -	Suma y resta

Tabla A-12. Operadores aritméticos

Operador	Significado
=	Igual que
<>	Diferente que
<	Menor que
>	Mayor que
<=	Menor o igual que
=>	Mayor o igual que

Tabla A-13. Operadores de relación

Operador	Significado
Not	Negación
And	Conjunción
Or	Disyunción inclusiva
Xor	Disyunción exclusiva
Eqv	Equivalencia (opuesto a Xor)
Imp	Implicación (falso si primer operando verdadero y segundo operando falso)

Tabla A-14. Operadores lógicos

Operador	Significado
&	Concatenación
Like	Compara dos cadenas de caracteres

Tabla A-15. Operadores para manejo de caracteres

## SENTENCIAS

Una sentencia o instrucción es una línea de código que indica una o más operaciones a realizar. Una línea puede incluir varias sentencias, separadas unas de otras por dos puntos, aunque lo ideal es utilizar una línea para cada sentencia.

Por ejemplo:

```
Dim CANTIDAD As Integer, PRECIO As Double, TOTAL As Double
CANTIDAD = 30 : PRECIO = 1.5
TOTAL = CANTIDAD * PRECIO
```

Una sentencia Visual Basic puede escribirse en más de una línea física utilizando el carácter de continuación de línea (un espacio en blanco seguido del carácter de subrayado).

## SENTENCIAS ENTRADA Y SALIDA DE DATOS

Anteriormente describimos los principales controles de Visual Basic que se usan para entrada y salida de datos (etiquetas y cuadros de texto). Otra posibilidad de proporcionar datos o de visualizarlos durante la ejecución de una aplicación es utilizando cajas de diálogos predefinidas. Debido a que esta es la forma más simple de realizar la E/S.

## Entrada de datos

La entrada de datos permite proporcionar valores durante la ejecución de una aplicación. Estos valores son ingresados generalmente mediante el teclado y asignados a variables del programa. En Visual Basic una forma de ingresar datos a la aplicación es utilizando la caja de diálogo predefinida provista por la función `InputBox`. Su sintaxis es la siguiente:

```
VARIABLE = InputBox(MENSAJE [, TÍTULO] [, PREDETERMINADO])
```

Donde `MENSAJE` es el mensaje que indica al usuario el tipo de información que debe ingresar. Por ejemplo, las sentencias:

```
Dim EDAD As Integer  
EDAD = InputBox("Ingrese su edad?")
```

El resto de los parámetros son opcionales. `TÍTULO` es el texto que se muestra en la barra de título del cuadro de diálogo, si se omite `TÍTULO`, el nombre de la aplicación es la que se muestra en la barra de título. `PREDETERMINADO` es una expresión de cadena que aparece en el cuadro de texto como respuesta predeterminada si no se proporciona ningún otro texto. Si se omite `PREDETERMINADO` el cuadro de texto se muestra vacío.

## Salida de datos

La salida de datos generalmente consiste en tomar la información de las variables almacenadas en memoria y mostrarla en pantalla. Para realizar la operación de salida de datos en Visual Basic, podemos utilizar el procedimiento `MsgBox`, cuya sintaxis es como sigue:

```
MsgBox MENSAJE [, ICONO, TÍTULO]
```

`MENSAJE` es La cadena que se muestra como salida en la caja de diálogo. `ICONO` es se refiere al estilo de icono que se va ha utilizar y `TÍTULO` es un texto que se mostrará en la barra de título de la caja de diálogo.

## ESTRUCTURAS DE CONTROL

Las estructuras de control permiten tomar decisiones (sentencias de decisión) y realizar un proceso repetidas veces (sentencias de repetición). Para ello Visual Basic dispone de las siguientes estructuras:

- Decisión: If . . . Then . . . Else
- Decisión multiple: If . . . Then . . . Elseif
- Seleccionador: Select Case
- Repetición para o repetición indexada: For . . . Next
- Repetición mientras: While . . . . Wend
- Repetición hasta Do . . . Loop

### Sentencia If . . . Then . . . Else

Este tipo de estructura evalúa una determinada condición y en función a ello ejecuta uno de entre dos posibles grupos de sentencias. Su sintaxis es la siguiente:

```
If CONDICIÓN Then
    SENTENCIAS_VERDADERAS
Else
    SENTENCIAS_FALSAS
End If
```

### Sem|ntencia If . . . Then . . . Elseif

Este tipo de estructura se utiliza para elegir una de entre múltiples alternativas. Su sintaxis es como sigue:

```
If CONDICIÓN_1 Then
    SENTENCIAS_1
ElseIf CONDICIÓN_2 Then
    SENTENCIAS_2
.
.
```



```
Else
    SENTENCIAS_FALSAS
End If
```

### **Sentencia Select Case**

Esta estructura es una alternativa a la estructura `If . . . Then . . . Elseif,` cuando lo que necesita es comparar la misma expresión con diferentes valores.

Su sintaxis es la siguiente:

```
Select Case EXPRESIÓN_TEST
    Case EXPRESIÓN_1
        SENTENCIAS_1
    Case EXPRESIÓN_2
        SENTENCIAS_2
    .
Case Else
    SENTENCIAS_FALSAS
End Select
```

### **Estructura For . . . Next**

Esta estructura es utilizada para ejecutar un bucle un número determinado de veces. El número de iteraciones deberá ser conocido de antemano. Su sintaxis es la que se presenta a continuación:

```
For CONTADOR = INICIO To FINAL [Step INCREMENTO/DECREMENTO]
    SENTENCIAS_REPETITIVAS
[Exit For]
Next
```

### **Estructura While . . . Wend**

La estructura `While . . . Wend` es aquella en la que el número de iteraciones no se conoce por anticipado y el cuerpo del bucle se repite mientras se cumple una determinada condición. Su sintaxis es la siguiente:

```
While CONDICIÓN
SENTENCIAS_REPETITIVAS
Wend
```

### **Sentencia Do . . . Loop**

Esta estructura ejecuta un bucle mientras una condición dada sea cierta, o hasta que una condición dada sea cierta. La condición puede ser verificada antes o después de ejecutarse el cuerpo del bucle. Su sintaxis es:

```
Do [While/Until] CONDICIÓN
    SENTENCIAS_REPETITIVAS
[Exit Do]
Loop
```

### **O la siguiente**

```
Do
    SENTENCIAS_REPETITIVAS
[Exit Do]
Loop [While/Until] CONDICIÓN
```

Esta estructura (como se puede observar en ambos formatos) permite realizar varias estructuras diferentes dependiendo si la condición de terminación está al principio o al final del cuerpo del bucle.

**Bibliografía:**

- Luis Joyanes Aguilar – Antonio Muñoz Clemente. “Microsoft Visual Basic 6.0”. Mc Graw Hill. 1999.
- Carlos M. Rodríguez Bucarelly - Pablo A. Rodríguez Bucarelly. “VISUAL BASIC 6.0 Programación Orienta a Objetos”. Primera Edición.
- Marco Antonio Tiznado Santana. “El Camino fácil a Visual Basic 6.0”. Mc Graw Hill. 2000.
- Ayuda en línea del Visual Basic 6.0