

## Antecedentes Históricos

El lenguaje de programación BASIC (Beginner's All Purpose Symbolic Instruction Code), nació en 1964 como una herramienta para principiantes, muy versátil y fácil de utilizar, con una manera sencilla para programar con instrucciones simples tomadas del lenguaje humano como print o input.

Por los años 70, el BASIC se quedó relegado por el apareamiento de lenguajes estructurados como el FORTRAN, COBOL, mientras que el BASIC no manejaba estructuras y la programación era lineal – secuencial con saltos de control a cualquier parte del mismo programa lo que hacía que el programador sea desordenado.

A partir del año 78 el BASIC creó un lenguaje estándar, que venía como parte del Sistema Operativo de la Compañía Microsoft, lo que implicaba que toda persona que tenía un computador personal debía aprender por lo menos los comandos básicos para su utilización.

Sin embargo cuando vino ya el QUICK BASIC, que era el mismo BASIC pero mejorado, estructurado, con herramientas de intercambio de información, creación de ejecutables, etc., los otros lenguajes ya habían tomado ventaja a pesar de ser más complejos en su aprendizaje y utilización.

La aparición de Manejadores de Base de datos por FOX software también presentó un gran problema, relegando aún más al ya abatido QBASIC.

Luego apareció el entorno Windows, con su interfaz gráfica que simplificaba aún más el trabajo en el computador personal pues, ya no había que aprender y escribir un gran número de comandos sino, simplemente con un click del mouse seleccionar y ejecutar.

Dado este entorno, BASIC se sintió tentado a superarse uniendo su gran sencillez de manejo y aprendizaje con la interfaz gráfica de windows, intercambio de información, sus librerías, drivers y controladores, manejo de bases de datos, versatilidad, etc, con lo que nació VISUAL BASIC, que recuperó el terreno perdido y en la actualidad es el lenguaje visual más utilizado, pues se ha convertido en la primera herramienta de desarrollo de aplicaciones en entorno Windows.

Es obligado decir sin embargo, que VISUAL BASIC no puede comparar sus prestaciones con otros lenguajes que pueden llegar al fondo de la máquina, manejar sus registros de memoria, direccionarla o realizar librerías propias (DLL), como lo hace el Visual C, pero, si no nos interesa manejar por nosotros mismos la memoria y dejar que el computador lo haga debemos utilizar VISUAL BASIC y lograremos programas fáciles de hacer y lograr altas prestaciones y confiabilidad.

Actualmente tenemos la versión 6.0 de VISUAL BASIC que forma parte de la familia de MICROSOFT VISUAL ESTUDIO junto con otros lenguajes visuales aunque últimamente, se está expendiendo también la versión VISUAL .NET, para trabajo en especial con redes de internet, pero, tanto en la una como en la otra versión su trabajo es similar.

# GUIA PRACTICA NO.1

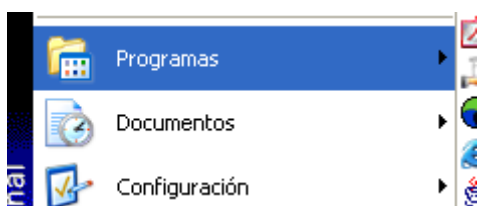
## TEMA: LA VENTANA DE TRABAJO DE VISUAL BASIC.

**Nota:** Recuerde primero que en el computador debe estar ya instalado VISUAL BASIC para poder trabajar, si no lo está, instálelo o busque un técnico especializado para que lo haga:

Ubique el puntero del mouse en el botón de INICIO y haga click, (no importa la versión de Windows que maneje en todos será similar el trabajo).



Cuando se abra el menú contextual (ventana de opciones) ubique el puntero del mouse en la opción PROGRAMAS (o TODOS LOS PROGRAMAS en windows XP)



Vista tradicional



Vista Windows XP

Observe que se abre otra ventana de opciones, busque ahí MICROSOFT VISUAL ESTUDIO y ubique el puntero del mouse allí.

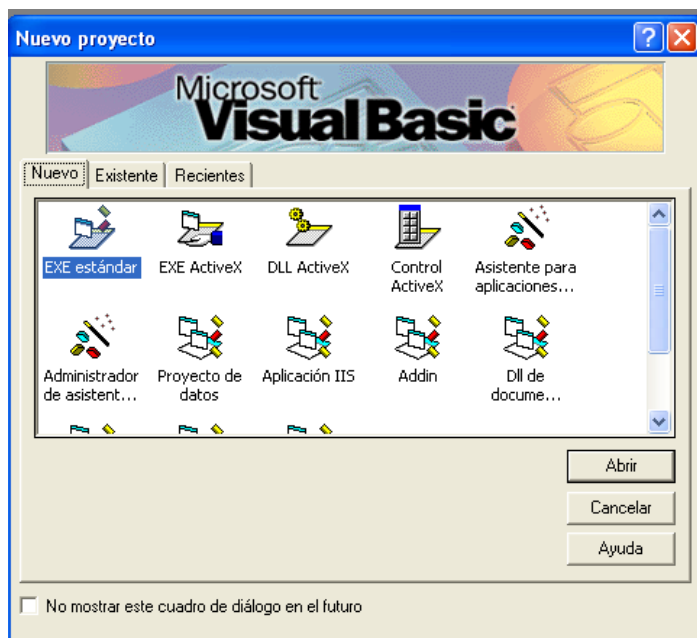


Se obtendrá la siguiente ventana de opciones:



Las opciones aquí presentadas dependerá de cómo haya sido la instalación, es decir, si se ha instalado todo el paquete de lenguajes, sino, aparecerá solo lo que se haya instalado.

Ahí ubique el puntero del mouse de la opción Microsoft Visual Basic 6.0 y de un click, inmediatamente se abrirá una ventana similar a la siguiente:



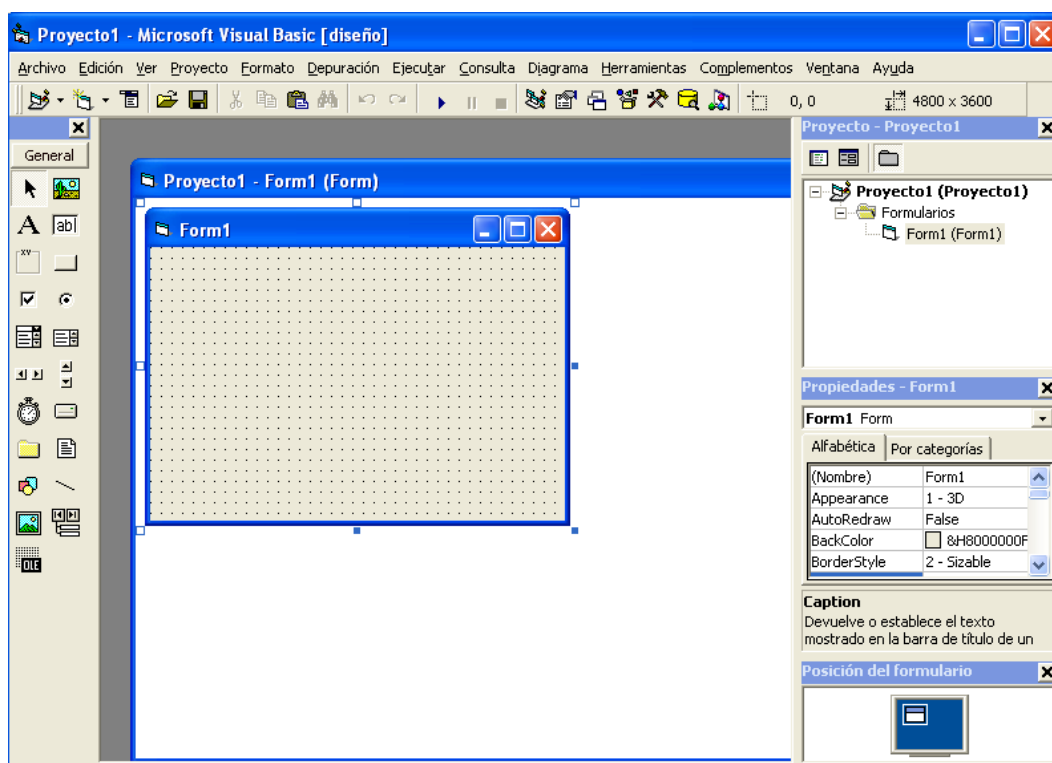
Esta ventana nos permite definir el proyecto que vamos a realizar, así:

- Tres pestañas con NUEVO, EXISTENTE, RECIENTES, que permiten: crear un nuevo proyecto, recuperar uno previamente grabado y uno que se haya trabajado recientemente, respectivamente
- Opciones: EXE estandar, EXE Activex, DLL Activex, etc, permite especificar el tipo de proyecto que vamos a realizar, en este momento, nos basaremos en la primera opción
- Botones: ABRIR, CANCELAR, AYUDA.- Permite, una vez seleccionado el proyecto, abrirlo, cancelar la acción o acceder a un nivel de ayuda que el mismo lenguaje le proporciona, siempre y cuando esté instalado previamente.

Vamos a hacer la selección de la opción haciendo doble clic del mouse en el icono correspondiente o haciendo un click en el ícono y click en la opción abrir



A continuación se abrirá otra ventana similar a la siguiente:



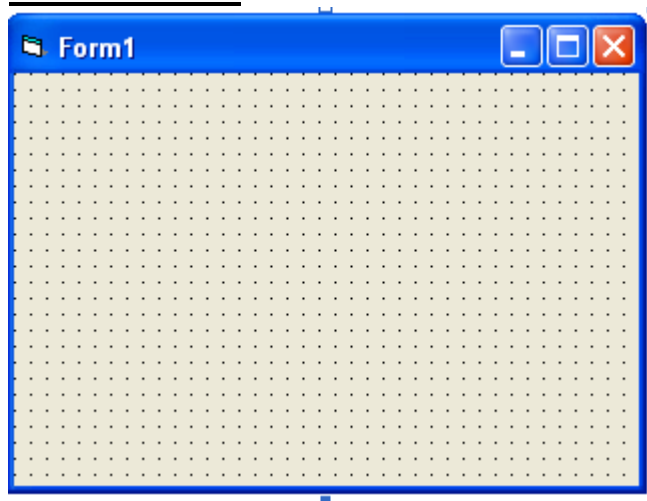
Es la ventana de trabajo de Visual basic, aquí podemos encontrar:



Barra de títulos ----->  
 Barra de Menús----->  
 Barra de herramientas----->

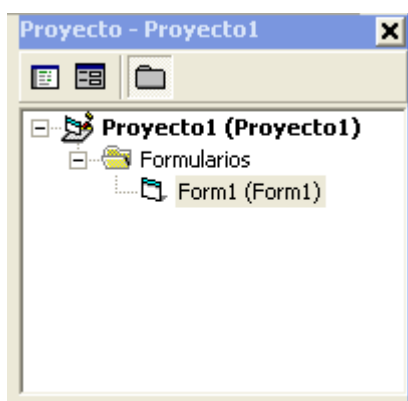
- En la barra de títulos aparecerá el nombre con el que grabemos el proyecto.
- La barra de menús nos permite escoger las opciones referentes al proyecto, que seguiremos tratando conforme se aplique.
- La barra de herramientas.- son opciones generalizadas de Microsoft para todos sus productos algunas son iguales en su trabajo como en Word, Excel, etc., y otras propias del lenguaje que de igual manera seguiremos tratando conforme se aplique.

**El formulario:**



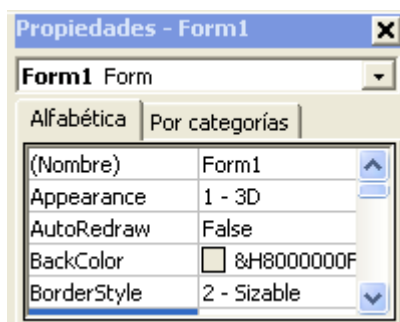
Es la parte en donde se crea la aplicación que será ejecutada en windows, aquí se ubicarán todos los elementos para realizar un programa, igual que otros elementos de Microsoft también tiene iconos de minimizar, maximizar y cerrar, recuerde que: un proyecto puede tener varios formularios. Por defecto aparece la palabra Form1, pero podemos cambiarlo a gusto y necesidad del programador.

### Explorador de proyectos.



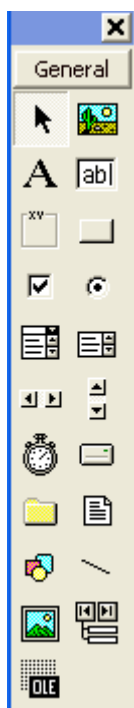
Esta ventana nos permite ver el o los proyectos actuales y todos los formularios vinculados a el, es muy importante cuando se quiere hacer varios formularios en un solo proyecto y permite seleccionar el formulario que se vaya a utilizar.

### Ventana de Propiedades:



Muestra las propiedades del control que se encuentre en el formulario o en el mismo formulario, por ejemplo, si está el formulario activo, la ventana de propiedades presenta las características de él , como por ejemplo, color de fondo, apariencia, etc., estas propiedades cambiarán de acuerdo al control que esté activo en ese momento.

### Cuadro de herramientas:

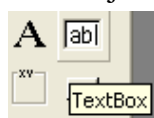


Son los controles que dispone el Visual basic para su utilización, obviamente aquí no se muestran todos porque la ventana no alcanzaría, solo están los más generales o los de utilización más seguida. aquí podemos encontrar Botones de comando, Cajas de texto, Etiquetas, etc, solo basta poner el puntero del mouse en cada una de las opciones (iconos) y el mismo lenguaje le dirá de que control se trata, por ejemplo, si pongo el puntero del mouse sobre la letra A, tendré el siguiente resultado:



Lo mismo sucede con los otros controles, recuerde que cada control tiene propiedades que en algunas pueden coincidir y en otras no.

Aquí van más ejemplos de controles (Llamados también objetos)

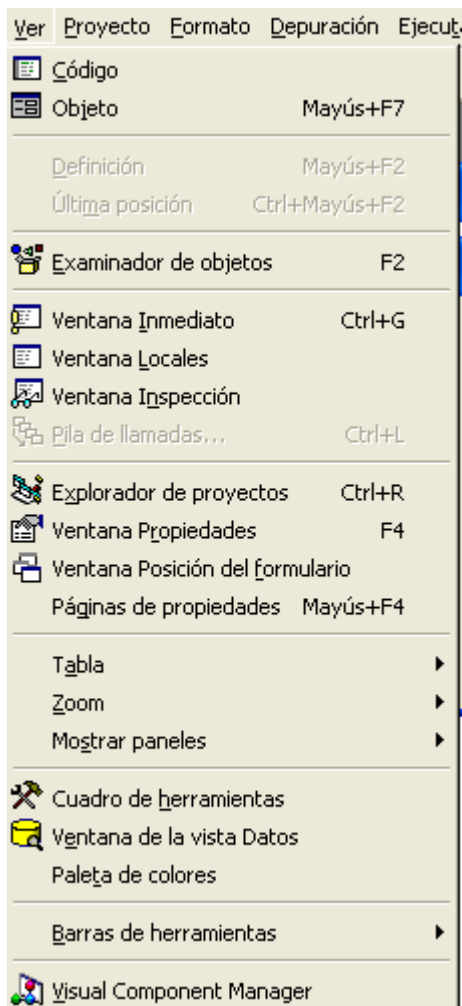


## Ventana Posición del formulario.



Muestra la forma en la que se está diseñando la aplicación y al mismo tiempo muestra en que parte va a ejecutarse el formulario con relación a la pantalla.

Cada uno de estos componentes pueden o no aparecer esto depende de cómo haya sido configurado el Visual basic, puede que ud, no encuentre por ejemplo la ventana de posición del formulario, no se asuste, no hay problema, puede irse a la opción VER de la barra de menús, dar un click allí y se abrirá una ventana, similar a la siguiente:



De un click en la opción Ventana de posición del formulario, y logrará que se aparezca dicha opción en la ventana de trabajo de Visual Basic (VB).

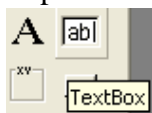
## Guia practica No.2

### TEMA: MANEJO DE PROPIEDADES Y OBJETOS (CONTROLES).

Marco teórico: Conceptos fundamentales para el uso de Visual basic.

#### Objeto.

Un objeto es una entidad que tiene asociado un conjunto de métodos, eventos y propiedades, Hay muchas clases de Objetos, y por lo tanto, puede llegar a haber tantos métodos, eventos y propiedades distintas como objetos diferentes, por ejemplo: Una caja de texto(TextBox), en la cual podemos escribir cualquier cosa es un objeto.



#### Método.

Los métodos son funciones internas de un determinado objeto que permite realizar funciones sobre él o sobre otro objeto. Ejemplo: Form1.Print "Bienvenidos al Visual Basic" es una línea de instrucción (método) que mostrará el mensaje Bienvenidos al Visual Basic cuando se ejecute el formulario.

#### Evento.

Un evento es una acción que realiza un programa, rutina, objeto para que nuestro programa lo ejecute, es decir, depende de las condiciones que existan para que un evento tenga razón de ser. Por ejemplo, ud., está distraído cruzando la calle y no se percató que un carro reanuda su marcha al ponerse el semáforo en verde, el conductor, al verle, pita, ud, reacciona y hace las siguientes consideraciones:

- El carro pita,
- el semáforo está en verde
- está en la calle al paso del carro

Analiza casi imperceptiblemente las condiciones e inmediatamente da algunos pasos a tras para salvar su vida, eso es un evento, es una atención a las condiciones presentadas sobre algo que va a llevar a cabo el protagonista.

#### Formulario.

Un formulario no es más que una ventana. Una ventana en windows de cualquier aplicación, puede haber cuantas ventanas necesite el usuario, pero cadauna tendrá un nombre propio.

#### Propiedades.

Son datos que hacen referencia a un objeto o formulario, es decir, son las cualidades o atributos que tiene algún objeto o control.

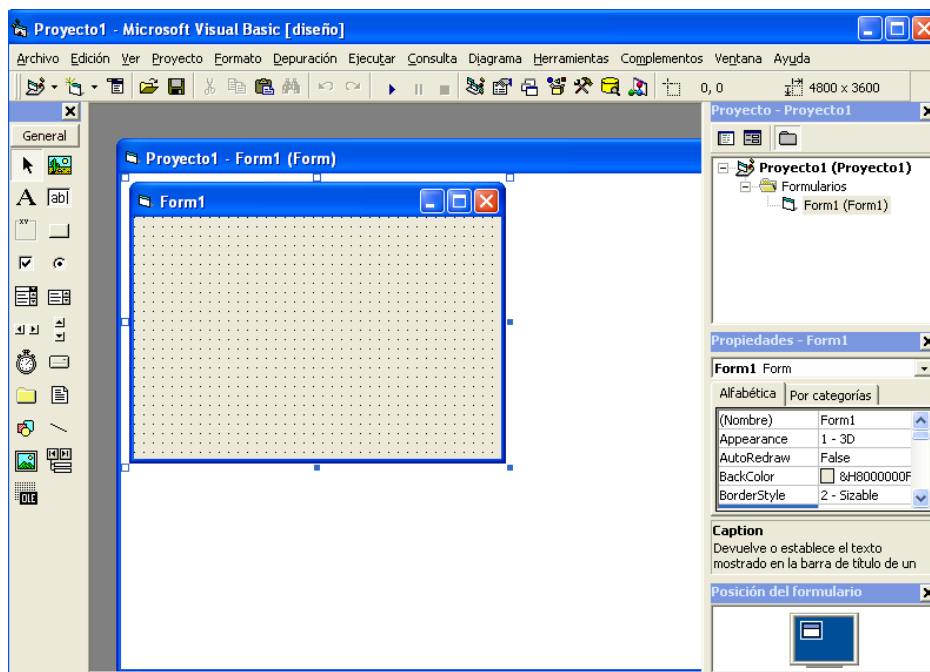
#### Ejemplo general de Objeto – Propiedad – Evento.

Vamos a suponer una silla, ese es un objeto.- el Objeto silla posee algunas propiedades como: color, material, alto, ancho, forma, etc, y si se le moviera es el evento mover

El objeto silla cumple una función determinada, sirve para sentarse, pero el fabricante puede cambiar sus propiedades, unos podrán hacerle negra, otros, café, azul, etc, y se moverá (evento mover) dependiendo de la necesidad de quien quiera sentarse y en el lugar donde quiera sentarse.

**Ahora vamos con la práctica.**

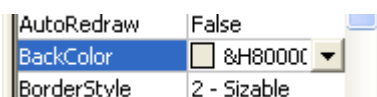
Abra un formulario nuevo en Visual Basic



De un click del mouse en el formulario cuyo título es Form1, observe que todo el rededor se marcará, e inclusive verá unos pequeños cuadraditos en las esquinas y en las partes centrales, eso permite cambiar de tamaño al formulario, de un clic en uno de esos cuadraditos y no suelte la tecla el mouse, mueva a una nueva dirección y suéltelo, observe que ha cambiado de tamaño.

Allí hemos aplicado lo que aprendimos anteriormente, el Objeto es FORMULARIO, el evento es UTILIZAR EL MOUSE en la propiedad ANCHO y LARGO, también podemos, cambiar en la VENTANA DE PROPIEDADES. Por ejemplo: Vamos a cambiar de color de fondo:

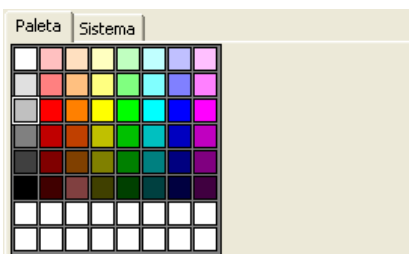
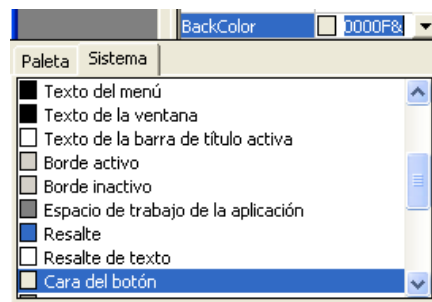
Busque en la VENTANA DE PROPIEDADES la opción: BackColor, haga click allí



observe que se ilumina la propiedad y aparece un triángulo invertido, es un objeto que al hacer click allí presenta algunas opciones, tales como:

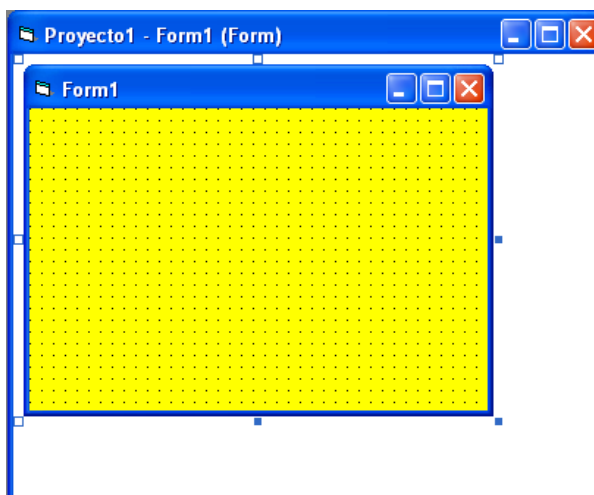


Opciones que permiten cambiar las coloraciones De algunas partes, que iremos aplicando conforme Avancemos en nuestro aprendizaje, observe que En la parte superior presenta dos pestañas, por Defecto aparece activa Sistema, de un click en Paleta, y observará lo siguiente.



Allí elija un color a su gusto y haga click, Por ejemplo, en nuestro caso vamos a escoger el color amarillo.

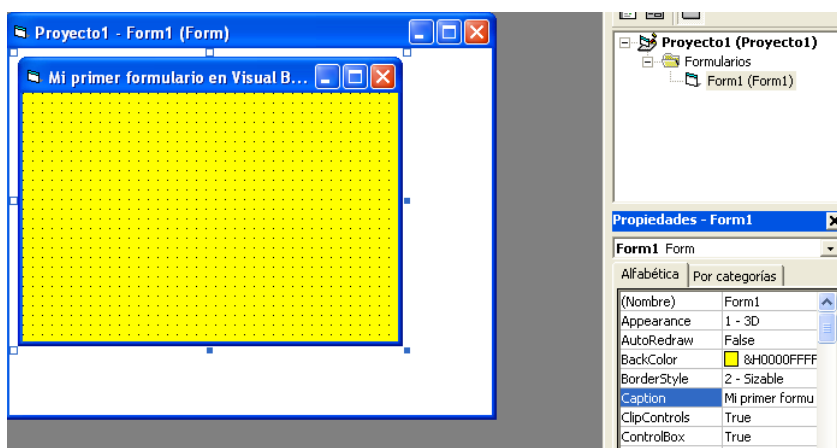
Luego de haber escogido una propiedad diferente para el objeto formulario el resultado será así.



Observe el cambio de Color de fondo del Formulario. Si su manual es en Blanco y negro, no Verá ningún cambio Pero ud., sabrá que si Se realizó.

Otra de las propiedades que tiene el formulario y que también coincide con algunos otros objetos es CAPTION, que permite cambiar el nombre al formulario, es decir, vamos a cambiar el nombre Form1, por otro, entonces, haga click en la propiedad CAPTION, y Digite el siguiente texto “Mi primer formulario en Visual Basic”

El resultado será el siguiente:



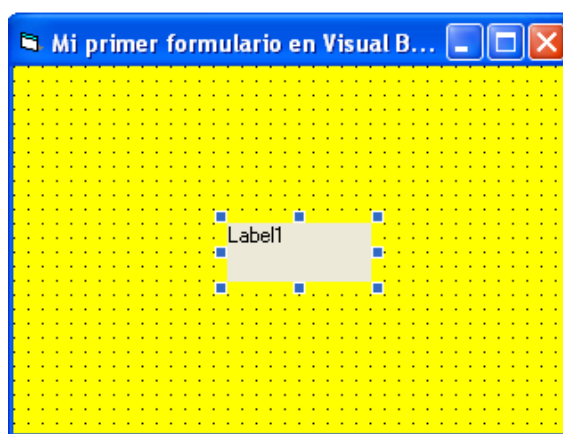
Observe que, mientras está escribiendo en la propiedad CAPTION, simultáneamente aparecerá en el formulario.

### ¿Cómo ubico los objetos (controles) en el formulario??.

Pues, existen dos formas de ubicar los controles:

1. Haciendo doble click en el objeto deseado en la barra de herramientas, por ejemplo, elija el objeto LABEL y haga doble click.
2. Haciendo click en el objeto LABEL, diríjase al formulario, de un click sostenido del mouse en el lugar que ud., desee llévalo hasta el lugar deseado y suelte el botón,

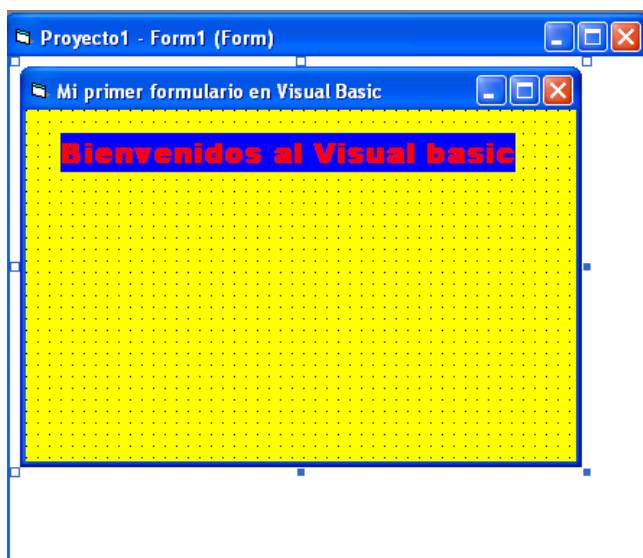
Sea cual fuere el proceso que realice se tendrá un resultado similar al siguiente:



Es el Objeto LABEL, marcado

Vamos a las propiedades a realizar algunos cambios:

- Mueva con el ratón el objeto a la parte superior del formulario y céntrelo (click sostenido y soltar)
- Elija la opción color de fondo (**BackColor**), y cámbiele de color, en nuestro ejemplo, azul.
- Elija la opción **CAPTION** y escriba Bienvenidos al Visual Basic, observe que la visibilidad no es muy buena, pero.
- Elija la opción **ForeColor** y cambie el color de la letra, su procedimiento es igual al BackColor.
- Elija la opción Font de un click en el ícono que contiene tres puntitos y le saldrá un ventana de escogitación, cambie de tipo de letra, tamaño, estilo, etc y de click en aceptar, por ejemplo: tipo: arial; estilo: negritas; tamaño: 14, observe que el texto se perdió en el cuadro de Label,
- Elija AutoSize y haga doble click , observe el resultado, será similar al siguiente:



Si tiene problemas al observar  
El formulario tal como se muestra  
En este resultado, aumente el tamaño

Para este trabajo hemos utilizado las siguientes objetos y propiedades

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
FORM1	CAPTION: FORM1 BACKCOLOR:&H8000000H	CAPTION:MI PRIMER FORM.. BACKCOLOR:&H0000000H
LABEL	BACKCOLOR:&h8000000H CAPTION:LABEL FOREGOLOR: &H80000012& FONT: MS SANS SERIF 8	BACKCOLOR:F0000& CAPTION:BIENVENIDOS AL... FOREGOLOR: &H000000FF& FONT: ARIAL 14 NEGRITA

En las próximas prácticas se hará una especificación de las propiedades como se expresa en el cuadro anterior, con los objetos utilizados en los formularios, sus propiedades originales y las propiedades que serán cambiadas.

**TRABAJO PARA EL ESTUDIANTE:**

Ubique más objetos a su gusto (arrastrándolos de la barra de herramientas al formulario) y cambie las propiedades, investigue que acción realizan cada una de las propiedades.

**¿Cómo hago para salir del Visual Basic?**

Existen dos formas de salir:

1. Dirija el puntero del mouse al menú Archivo y de click en salir.
2. haga click en el ícono cerrar

De las dos maneras el Visual Basic le recordará que no ha grabado su trabajo, de click en No, pues, aún no es tiempo de guardar lo que hacemos, eso lo aprenderemos en la siguiente lección.

## GUIA PRACTICA # 3

### TEMA : MENSAJE Y USO DE OBJETOS

1. Acceda usted al botón INICIO de Windows de un click
2. Busque MICROSOFT VISUAL ESTUDIO 6.0, observe que se abre un menú contextual
3. Ubíquese en MICROSOFT VISUAL BASIC 6.0 O VISUAL BASIC 5.0, de click
4. Observe que se abre una ventana similar a esta:

Es la ventana que nos presenta Algunos tipos de trabajos que Podemos hacer con Visual Basic

Observe que hay tres pestañas:

- Nuevo.- Para crear nuevos proyectos
- Existente.- Los que existen grabados en alguna unidad
- Recientes.- Los que se ha estado Trabajando últimamente

Por defecto (automáticamente) Aparece iluminado EXE estándar, hagamos click en Abrir, observe la siguiente ventana:

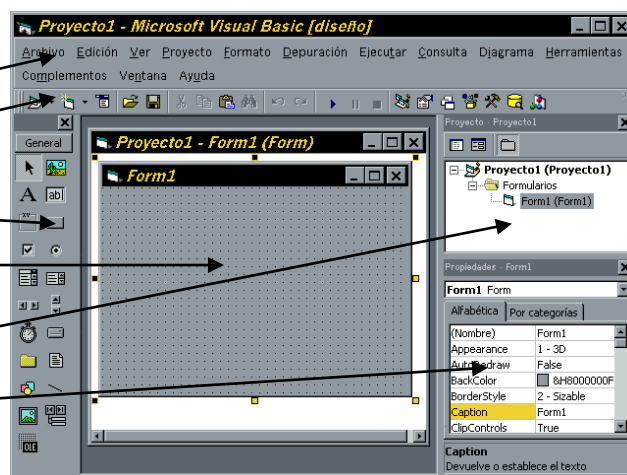


Recuerde que es la ventana de edición de VB, aquí Encontramos:

- Barra de menús
- Barra estándar
- Barra de objetos

Formulario

- Zona de indicadores
- Explorador de proyectos
- Propiedades



Esto cambiará dependiendo de la versión del VB y de lo que esté activado en el menú VER, observe que es lo que ya vimos en una práctica anterior, y está presentado aquí para que se reafirme su conocimiento.

Lo que nos interesa por el momento es el formulario: es allí en donde haremos nuestro primer trabajo con VB, el formulario es lo que se verá en el ambiente Windows como una ventana.

Lo que vamos a hacer es desplegar un mensaje en el formulario: el mensaje sugerido es "MI PRIMER PROGRAMA EN VISUAL BASIC". Pero no como lo hicimos anteriormente, es decir, cambiando la propiedad CAPTION del formulario, sino, ya en el entorno de la programación, o sea, poniendo un código

Para esto de un doble click del ratón en el formulario, seguramente observará lo siguiente:

Aquí existen varias cosas que destacar:

Form.- es la especificación del objeto que se está tratando en ese momento, en este caso el formulario.

Load.- Es el procedimiento que está vinculado a tal objeto (FORM), es decir el evento que va a realizar ese objeto.

Private Sub Form\_Load()

End sub

Es el evento en sí, aparece automáticamente y nos indica que se ejecutará al cargarse el formulario en memoria, es decir, al ejecutarse, allí se debe poner el código o instrucciones del programa.

Si usted tiene un poco de conocimientos previos, sabrá que es una subrutina o un pequeño programa, Private es la especificación del tipo de datos o su alcance, lo cual trataremos más adelante.

Deberá usted digitar lo siguiente:

```
Private Sub Form_Load()
    Form1.Show
    Print "Mi PRIMER PROGRAMA EN VISUAL BASIC"
End sub
```

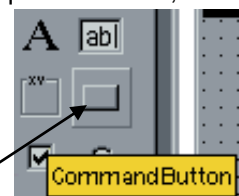
Una vez escrito, ejecute su programa dando un click en el ícono ejecutar o presionando F5, observe el resultado.....



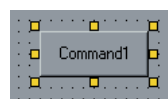
Para regresar a la edición en Visual Basic utilice el icono de Cerrar típico de las ventanas de Windows.

Seguramente estará aún en la ventana de código, salga de allí, igual que el anterior, dando un click en el icono de cierre.

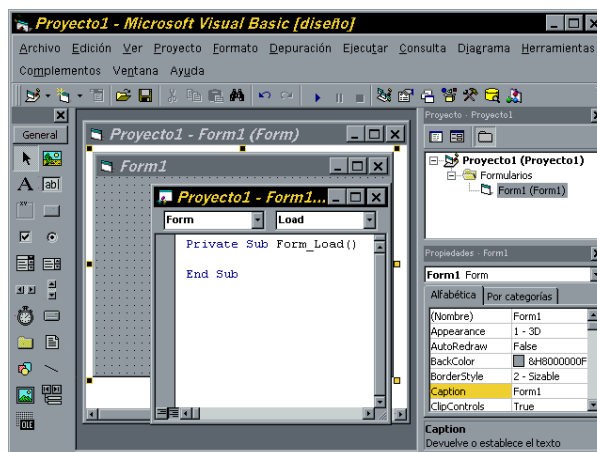
Hasta aquí, nuestro programa bien, pero vamos a mejorarlo. Vamos a ubicar el primer OBJETO en nuestro formulario, con la flecha del ratón ubíquese en la barra de OBJETOS y busque uno que diga COMMAND BUTTON haga un doble click

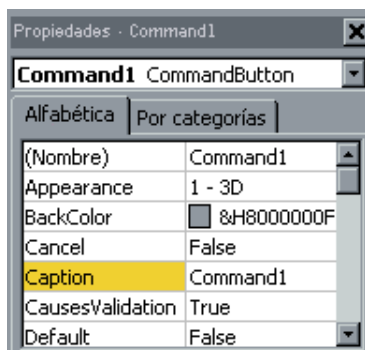


Observe que aparece en el formulario un objeto, así:

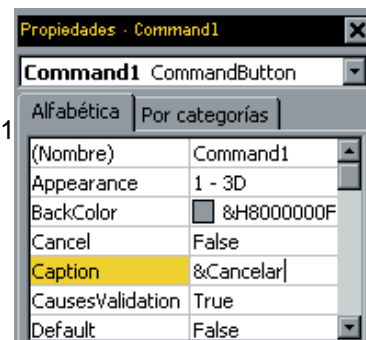


Por defecto aparece con el nombre de Command1, pero, podemos cambiarle de nombre, yéndonos a las propiedades del objeto y buscando la opción CAPTION, así:





De un click en donde está Command1 y Cámbielo por &Cancelar, así ---->



En este momento estamos cambiando las propiedades en TIEMPO DE DISEÑO, es decir, cuando estamos diseñando el formulario.

Note que el ampersand (&) crea una tecla de acceso rápido es decir si usted presiona ALT + C, se realizará una acción en TIEMPO DE EJECUCION, es decir, cuando el programa se esté ejecutando.

Ahora, solo falta añadir código al Objeto CANCELAR, de un doble click allí y digite lo siguiente:

```
Private Sub Command1_Click()
    End
End Sub
```

Private Sub Command1\_Click(), sale automáticamente, y significa que si el usuario hace un click del ratón sobre este botón de comando, se realizará alguna acción.

End, es una instrucción de BASIC que permite terminar un programa

End Sub Finaliza esta parte del programa.

Ejecute su programa y observe lo que obtiene:

Su programa deberá tener esta apariencia:



Ahora bastará con hacer click en el botón Cancelar o ALT + C, para terminar la ejecución de su programa(y regresará al formulario).

También puede cambiar el nombre del formulario, accediendo a la propiedad CAPTION , entonces, de un click en el formulario y busque CAPTION y cámbielo a su gusto, ejecute su programa y observe el resultado.


Recuerde que cada OBJETO tiene sus propiedades, y para cambiar sus propiedades debe dar un click en el OBJETO que usted desea, por ejemplo, de un click en el formulario y pruebe con la propiedad BACKCOLOR

**TRABAJO PARA EL ESTUDIANTE:**

- A través de los pasos aprendidos, ubique el nombre del colegio, su nombre, su curso, especialidad, y otros datos particulares en un formulario, con color amarillo sobre un fondo verde, el nombre del formulario debe ser DATOS PERSONALES, y ubique un botón de cancelar, para terminar su ejecución. Cambie de tipos de letras, tamaños, etc.

## ¿Cómo grabo mi trabajo en Visual Basic?.

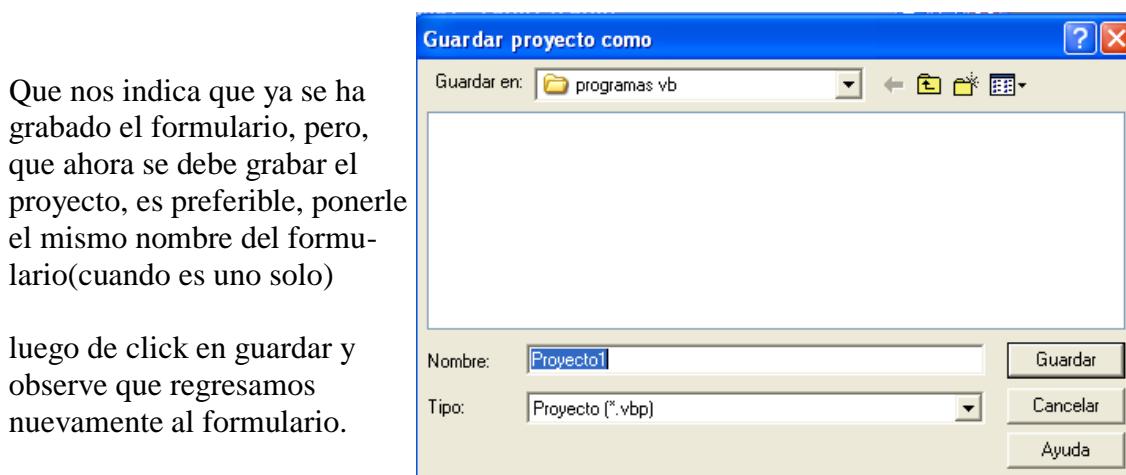
Si se desea grabar el formulario, puede hacer las siguientes acciones:

De un click en el ícono que representa un diskette,  seguramente, observara una pantalla similar a la siguiente:



Es la venta que nos indica que se va a grabar, puede usted seleccionar el lugar, el nombre, y el tipo de trabajo, luego, de click en guardar para grabar, en cancelar para no grabar y regresar al formulario y ayuda si necesita alguna explicación.

Cuando ya haya cambiado lo que usted necesite cambiar y de click en guardar, observe que tendrá otra ventana como esta:



Con este proceso se crearán tres programas que tendrán las extensiones: VBP, VBW y, FRM,(Proyecto, Windows y Formulario respectivamente).

Otra forma para grabar nuestro trabajo es haciendo click en la opción ARCHIVO de la barra de menús, se abrirá una ventana similar a la siguiente:

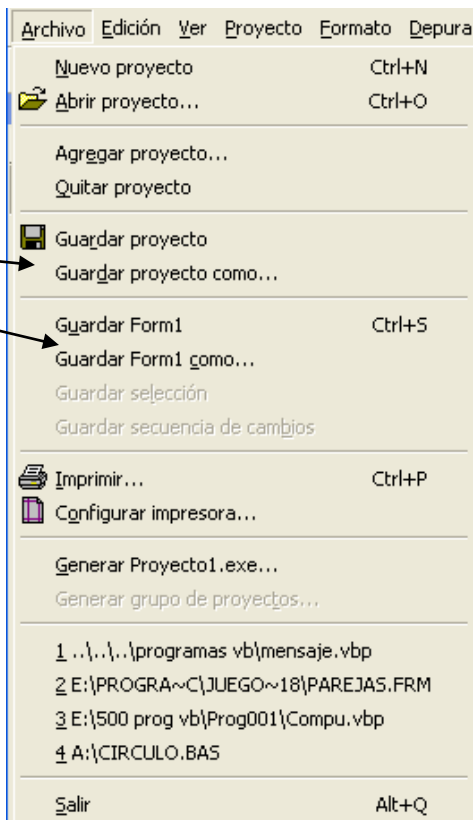
Aquí podemos observar que podemos grabar Solo el formulario o el proyecto en opciones diferentes

En las opciones guardar, se grabará con el nombre existente y en el lugar ya definido anteriormente,

En las opciones guardar **como**, se puede cambiar El nombre y la dirección de grabación.

Al poner GUARDAR PROYECTO, se realizarán Los pasos que ya vimos anteriormente, grabándose Los tres programas.

Al poner GUARDAR FORMULARIO, solo se grabará el formulario, tenga cuidado, pues no se grabará el proyecto y puede darle problemas al tratar de recuperarlo(Abrirlo), aunque, al tratar de salir del VB, el lenguaje le recordará que debe grabar el proyecto.

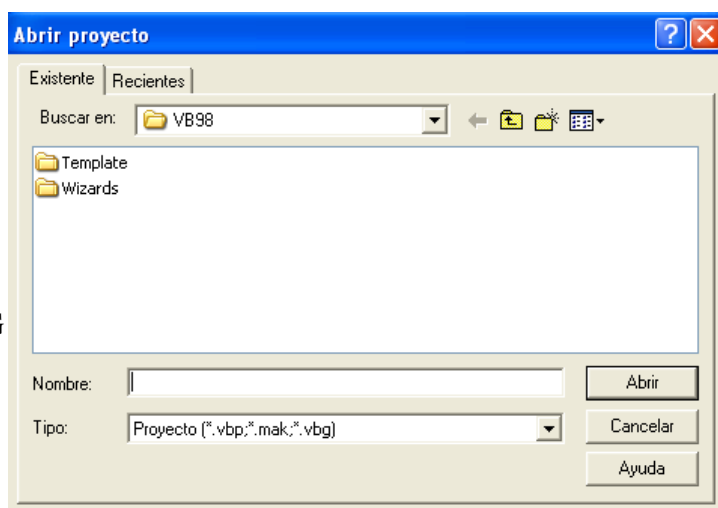


### ¿Cómo recupero un programa ya grabado??

El proceso es similar al que se utiliza en todas las ventanas de Microsoft, al activar visual basic haga click en la pestaña de existentes, busque el directorio en el que está el programa deseado y haga click en abrir, o, si el programa ha sido hecho recientemente lo encontrará en una lista y haga el mismo proceso, por otra parte, también puede hacer click en la opción ARCHIVO de la barra de menú, de click en la opción ABRIR PROYECTO, en cualquiera de los casos, tendrá la siguiente ventana:

Note las pestañas, Existente y reciente,

Observe también que se podrán abrir los trabajos con las extensiones VBP, MAK y VBG



Ahora ya puede guardar sus proyectos (formularios) o recuperarlos cuando desee.



## GUIA PRACTICA # 4

TEMA:

Si bien nuestro programa anterior nos gustó por ser nuestro primer proyecto en VB, ahora aprenderemos a ubicar 2 OBJETOS más, llamados LABEL y TEXTBOX, el primero sirve para poner títulos y el segundo para ubicar textos, se los reconoce por los íconos (de la barra de objetos):

LABEL está identificado con la letra A, y,  
TEXT BOX está identificado con las letras ab.

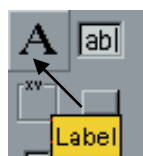


Fig 1

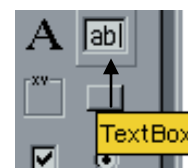
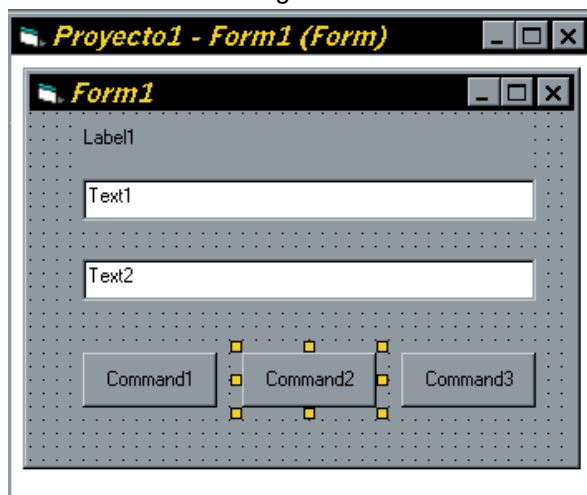


Fig 2

**PRACTICA:**

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique un LABEL haciendo doble click en el ícono correspondiente Fig1
4. Ubique dos Text Box haciendo doble click en el icono correspondiente Fig 2, observe que:
  - Al hacer doble click en TEXT BOX, se crea TEXT1, al hacer doble click en TEXT BOX otra vez, se crea TEXT2. uno encima de otro, no se asuste, puede moverlos a su gusto, como ya lo vimos anteriormente.
5. Ubique tres botones de comando, al igual que la práctica anterior, haciendo doble click en el ícono correspondiente, hasta conseguir COMMAND1, COMMAND2, COMMAND3
6. Su formulario y Objetos deberán tener la siguiente forma:



A continuación vamos a cambiar las PROPIEDADES de los objetos(Utilice las barras de desplazamiento para encontrar la propiedad deseada) , observe con atención el siguiente cuadro:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
LABEL	CAPTION : LABEL1 AUTOSIZE : FALSE FONT: SANS SERIF NORMAL 8	CAPTION : Ingrese su nombre: AUTOSIZE: TRUE FONT: ARIAL NEGRITA 14
TEXT1	TEXT: TEXT1	TEXT: (nada borre todo)
TEXT2	TEXT:TEXT2 APPEARANCE: 1-3D BORDERSTYLE: 1-FIXED STILE BACKCOLOR: &H80000005 VISIBLE: TRUE	TEXT: (nada borre todo) APPEARANCE:0-FLAT BORDERSTYLE: 0-NONE BACKCOLOR: &H80000004 VISIBLE: FALSE
COMMAND1	CAPTION:COMMAND1	CAPTION: &ACEPTAR TEXTO
COMMAND2	CAPTION:COMMAND2	CAPTION: ¿&OTRA VEZ?
COMMAND3	CAPTION:COMMAND3	CAPTION: &CANCELAR

FORM1	CAPTION:FORM1 ICON: (ICONO)	CAPTION: SEGUNDO PROYECTO ICON: MYCOMP (observe la ruta a continuación)
-------	--------------------------------	--

Breve explicación de las propiedades:

**LABEL:**

**Captión:** Título visible en momentos de diseño y ejecución.

**Autosize.**- Tamaño automático en relación al texto (TRUE), asignado por el usuario en tiempo de diseño (FALSE).

**Font.**- Tipo de letra, atributos y tamaño

**TEXT:**

**Text.**- similar al CAPTION, permite ubicar un texto a gusto del programador o pedido de ingreso

**Appearance:** Apariencia del objeto, 3 dimensiones, o normal (0-flat)

**BorderStyle.**- Estilo de borde, simple o nada

**BackColor.**- Color de fondo, generalmente blanco (&H80000005=fondo de ventana), cambiamos al color del formulario(&H80000004=barra de menús)

**Visible.**- Permite ocultar el objeto es decir hacerlo invisible según este activo TRUE o inactivo FALSE.

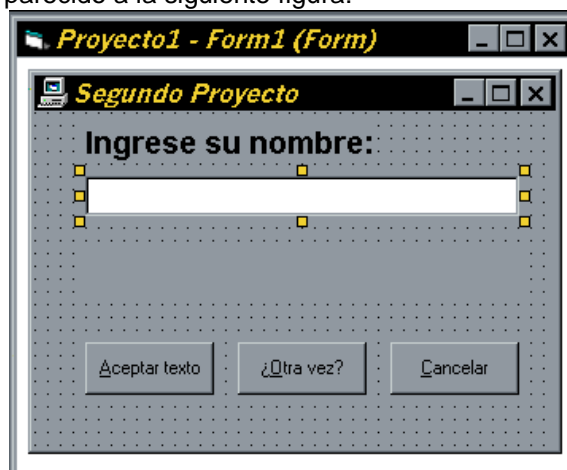
**FORM1:**

**Icon:** Permite cambiar el icono que acompaña al título del formulario, esta puede ser la ruta en su computador:

C:\Archivos de programa\Microsoft Visual Studio\Common\Graphics\icons\win95\Mycomp

O simplemente busque Mycomp.ico es su computador y ubique la ruta, si no lo encuentra, ponga uno cualquiera o déjelo sin icono.

Su formulario debe estar parecido a la siguiente figura:



Ahora solo falta poner el código que permita la ejecución de esta formulario:

- De doble click en el botón **Aceptar texto** y escriba:

```
Private Sub Command1_Click()
    Text2.Visible = True
    Text2.Text = "Hola " & Text1 & " Bienvenido al Visual Basic"
End Sub
```
- De doble click en **¿Otra vez?** Y escriba:

```
Private Sub Command2_Click()
    Text1 = ""
    Text2 = " "
    Form1.Show
End Sub
```

- De doble click en **C**ancelar y escriba  
Private Sub Command3\_Click()  
End  
End Sub

Ahora analicemos cada una de las líneas de código (script):

Al hacer click en el botónr **A**ceptar texto, ocurre lo siguiente:

Text2.Visible = True (Hace visible al objeto TEXT2, cambiando su propiedad en tiempo de Ejecución, es decir cuando se esté ejecutando el formulario).

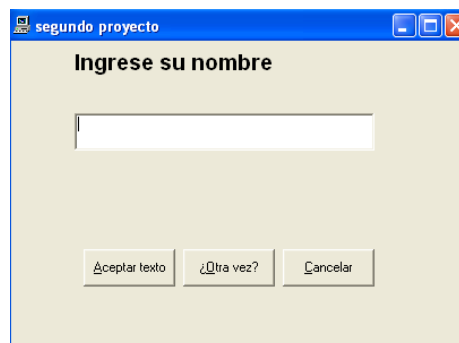
Text2.Text = "Hola " & Text1 & " Bienvenido al Visual Basic"  
En el objeto TEXT2, se ubica el mensaje Hola... Bienvenido al Visual Basic y entre ellos se ubica lo ingresado en TEXT1, observe que el ampersand está uniendo cadenas de caracteres.

Al hacer click en **¿O**tra vez? , ocurre lo siguiente:

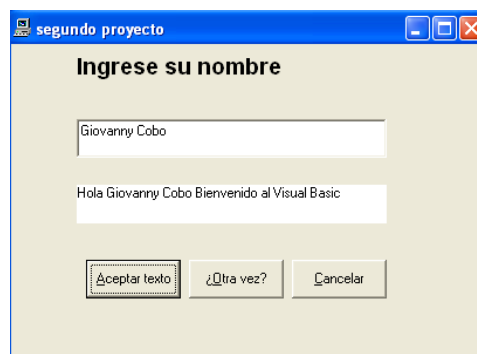
Text1 = "" Limpia lo escrito en TEXT1, ubicando espacios en blanco  
Text2 = " " Limpia lo escrito en TEXT2, ubicando espacios en blanco  
Form1.Show Activa o muestra otra vez el formulario activo.

Ejecute su proyecto y observe el resultado.

La primera ventana que obtendrá será la siguiente:



Cuando ingrese su nombre y haga click en ACEPTAR TEXTO, observará una ventana como la de este ejemplo:



Si hace click en OTRA VEZ?, se borrará todo y Aparecerá la ventana como el inicio,

Al escoger la opción Cancelar, se terminará la ejecución del formulario y regresará a la ventana de edición.

### **TRABAJO PARA EL ESTUDIANTE:**

- Realice el mismo formulario, aumentando varios textos de despliegue, por ejemplo: si ingresa el texto AMBATO, al hacer click en Aceptar texto puede salir, Ciudad Jardín, Tierra de los tres Juanes, Tierra de las flores y frutas, etc.
- Cree ejemplos de los anteriores y puede cambiar algunas propiedades del formulario o de los objetos como colores, tamaños, etc.

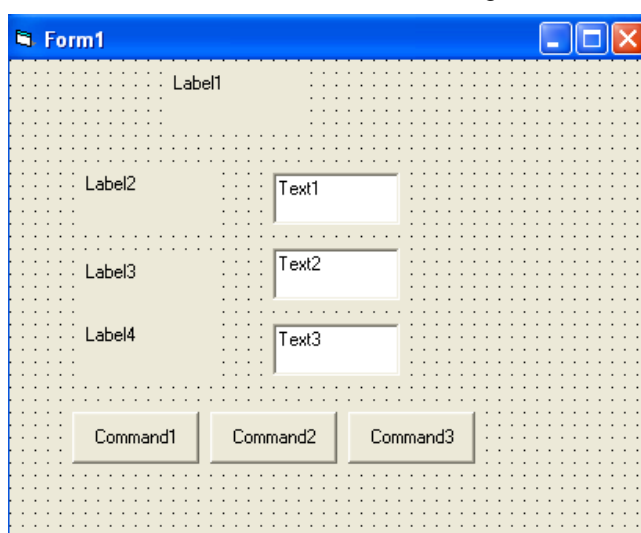
## GUIA PRACTICA # 5

### TEMA: PROGRAMACION TRADICIONAL (SUMA DE DOS NUMEROS):

A pesar que vamos a utilizar variables en este programa, no las vamos a tratar en este momento, sino, en el siguiente práctica.

#### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 4 objetos LABEL haciendo doble click en el ícono correspondiente
4. Ubique 3 objetos TEXT BOX haciendo doble click en el icono correspondiente
5. Ubique tres botones de comando, al igual que la práctica anterior, haciendo doble click en el ícono correspondiente, hasta conseguir COMMAND1, COMMAND2, COMMAND3
6. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



Ahora vamos a cambiar las propiedades de los objetos de la siguiente manera:

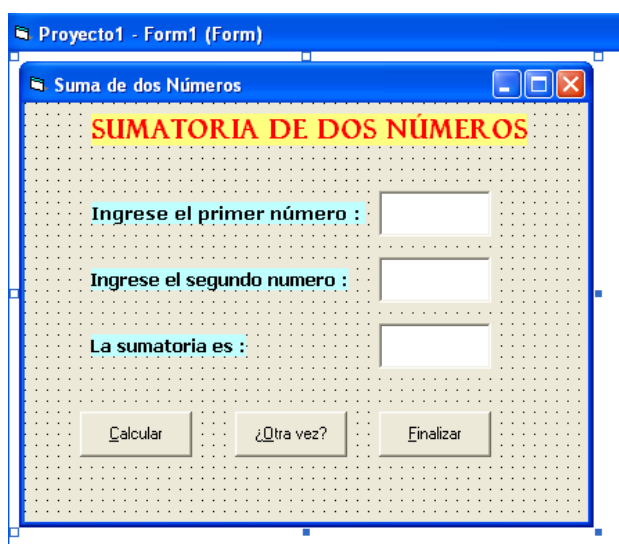
OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption : Form1	Caption : Suma de dos Números Color de fondo y tamaño de formulario a su gusto,
Label1	Caption : Label1 AutoSize : False BackColor : &H80000005 Font : Mssans Serif Normal 8	Caption : Sumatoria de dos Números AutoSize : True BackColor: A su gusto Font: A su gusto
Label2	Caption : Label2 AutoSize : False BackColor : &H80000005 Font : Mssans Serif Normal 8	Caption : Ingrese el primer número AutoSize : True BackColor: A su gusto Font: A su gusto
Label3	Caption : Label3 AutoSize : False BackColor : &H80000005 Font : Mssans Serif Normal 8	Caption : Ingrese el Segundo número AutoSize : True BackColor: A su gusto Font: A su gusto
Label4	Caption : Label4 AutoSize : False	Caption : La sumatoria es : AutoSize : True

	BackColor : &H80000005 Font : Mssans Serif Normal 8	BackColor: A su gusto Font: A su gusto
Text1	Name(Nombre) : Text1 Text: Text1	Name (Nombre): A Text : Nada – Borrar todo
Text2	Name(Nombre) : Text2 Text: Text2	Name (Nombre): B Text : Nada – Borrar todo
Text3	Name(Nombre) : Text3 Text: Text2	Name (Nombre): S Text : Nada – Borrar todo
Command1	Caption : Command1 Nombre : Command1	Caption : &Calcular Nombre : calcular
Command2	Caption : Command2 Nombre : Command2	Caption : ¿&Otra vez? Nombre : otravez
Command3	Caption : Command3 Nombre : Command3	Caption : &Finalizar Nombre : fin

**Recuerde que:**

- Para cambiar las propiedades de un objeto debe seleccionarlo primero, es decir, si se desea cambiar las propiedades de Label1, debo hacer un click en él (objeto), y ahí si proceder al cambio.
- Solo haga un click en el objeto para cambiar sus propiedades, pues, al hacer doble click se irá a la ventana del generador de códigos y esa opción no la vamos a hacer todavía sino después de cambiar las propiedades del objeto
- Cada objeto, entre sus propiedades posee color de fondo, color de letra, tamaño, etc, eso se deja a su gusto, cambie de colores, tamaños, tipos de letras, etc. Hasta que el o los objetos tengan atractivo para el trabajo tanto suyo como para el resto de usuarios.
- En las cajas de texto (Text Box) hemos cambiado el nombre original a las letras A, B y S, que representan las variables que vamos a utilizar.

Su formulario después de haber hecho los cambios pertinentes debe tener una forma similar a la siguiente:



Cada objeto puede tener código, es decir, debemos escoger el objeto que debe tener el código para ubicarlo, en nuestro caso, vamos a hacer doble click en el botón Calcular, y se abrirá la siguiente estructura.

```
Private Sub calcular_Click()

End Sub
```

Observe que se activa una subrutina del objeto de nombre calcular que se ejecutará al hacer click del ratón, ahí ubique la siguiente fórmula  $S=Val(a)+Val(b)$ , es decir:

```
Private Sub calcular_Click()
    S=Val(A)+Val(b)
End Sub
```

Significa que tomará el valor ingresado en la caja de texto llamada A y el valor de la caja de texto llamada B, la sumará y pondrá ese resultado en la variable S, que se mostrará en la caja de texto correspondiente.

La función Val transforma el texto ingresado (TextBox) y lo transforma a valor numérico para poder hacer cálculos, sino, nos dará un resultado diferente es decir, la unión de los números no su sumatoria, porque lo ingresado será tomado como cadena de caracteres, puede ud., hacer la experiencia quitando la función val y ejecutando su programa.

Haga doble click en **¿Otra vez?** , Obtendrá la subrutina correspondiente, allí ubique el siguiente texto.

A = ""	Limpia lo escrito en TEXT1 (A), ubicando espacios en blanco
B = " "	Limpia lo escrito en TEXT2 (B), ubicando espacios en blanco
S = " "	Limpia lo escrito en TEXT2 (S), ubicando espacios en blanco
A.SetFocus	(Ubica el cursor en ese objeto cuando se ejecute el formulario)
Form1.Show	Activa o muestra otra vez el formulario activo.

La subrutina en dicho objeto deberá quedar así:

```
Private Sub otravez_Click()
    A = " "
    B = " "
    S = " "
    A.SetFocus
    Form1.Show
End Sub
```

Por último haga doble click en Finalizar y ubique la instrucción END, así:

```
Private Sub fin_Click()
    End
End Sub
```

Significa que al hacer click en este botón la aplicación (ejecución del formulario) acabará y regresará a la ventana de formulario.

En ejecución el formulario deberá aparecer similar al siguiente:

### **TRABAJO PARA EL ALUMNO:**

Realice los siguientes programas utilizando lo estudiado hasta el momento:

- Multiplicación de tres números
- Las cuatro operaciones básicas con dos números
- Area de un triángulo  $(B*H)/2$
- Valor por la compra: en base al número de productos comprados y al valor de cada producto.
- Del programa anterior, incremente el 12% del IVA y despliegue ese valor y el valor total a pagar.
- Programas con fórmulas, masa, peso, densidad, áreas, perímetros, etc.

Seguramente para esto ud., necesitará la referencia de **operadores aritméticos**, así:

Para la suma utilice el signo +,

Para la resta utilice el signo -,

Para la multiplicación utilice el signo \*, (Asterisco)

Para la división utilice el signo /, (Slage)

### **JERARQUIA, PRIORIDAD O PRECEDENCIA**

En una fórmula se puede utilizar los siguientes parámetros en orden de acción

1. (); []; { }
- 2.- \*; /;
- 3.- +; -

Puede utilizar llaves, corchetes y/o paréntesis para destacar la forma en la que se realizarán las operaciones y su prioridad, por ejemplo, en el área de un triángulo no es lo mismo poner  $AT = b * h / 2$  que  $AT = (b*h)/2$ , en la primera fórmula solo se dividirá  $h$  para 2, lo cual nos puede dar un resultado erróneo y en la fórmula original se debe multiplicar primero  $b * h$  para luego dividirse para 2, es por eso que se debe ubicar entre paréntesis  $(b*h)$  para aclararle al computador que se debe realizar esa multiplicación primero para con ese resultado hacer la división.

Aunque en este programa con paréntesis o no, siempre nos dará igual resultado es mejor que se acostumbre a utilizarlos porque en otros programas más complejos será imprescindible utilizarlos.

## GUIA PRACTICA # 6

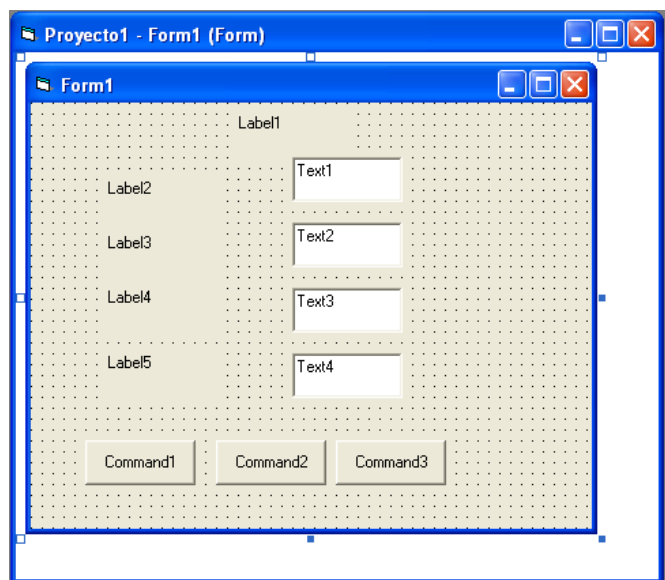
### TEMA: PROGRAMACION TRADICIONAL (TOMA DE DECISIONES):

#### MAYOR DE TRES NUMEROS

En Visual Basic tiene la misma estructura de las decisiones tradicionales utilizadas en cualquier lenguaje común inclusive se puede utilizar operadores lógicos, es por esto que si ud. Amable lector no tiene conocimientos sólidos de lo que son las decisiones, su estructura, su utilización se le sugiere, que trate de aprenderlos (utilice los libros de esta misma colección Qbasic, Pascal, Fox).

#### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 5 objetos LABEL haciendo doble click en el ícono correspondiente
4. Ubique 4 objetos TEXT BOX haciendo doble click en el icono correspondiente
5. Ubique 3 botones de comando, COMMAND BUTTON
6. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



Vamos a hacer el cambio de las propiedades de los objetos, recuerde que los colores, tipos de letras, tamaños, entre otros, ya se quedan a su gusto o criterio, además en LABEL recuerde que debe poner la propiedad AutoSize= true para que se autoajuste el texto, igual en TEXTBOX cambie la propiedad Text y déjela vacía (sin nada escrito), para que al ejecutar el formulario no aparezca nada escrito.

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption : Form1	Caption : Mayor de tres números
Label1	Caption : Label1	Caption : Mayor de tres números
Label2	Caption : Label2	Caption : Ing. Primer número
Label3	Caption : Label3	Caption : Ing. Segundo número
Label4	Caption : Label4	Caption : Ing. Tercer número
Label5	Caption : Label5	Caption : El número mayor es



Text1	Nombre : Text1	Nombre : A
Text2	Nombre : Text2	Nombre : B
Text3	Nombre : Text3	Nombre : C
Text4	Nombre : Text4	Nombre :M
Command1	Caption : Command1 Nombre : Command1	Caption : &Obtener mayor Nombre : obtener
Command2	Caption : Command2 Nombre : Command2	Caption : ¿&Otra vez? Nombre : otravez
Command3	Caption : Command3 Nombre : Command3	Caption : &Finalizar Nombre : fin

Luego se haber hecho los cambios pertinentes, el formulario deberá presentar la siguiente apariencia:



Ahora vamos a poner el código, haga doble click en el botón de comando correspondiente a **Obtener mayor** y ubique el siguiente código:

```
Private Sub obtener_Click()
    If Val(A) > Val(B) Then
        If Val(A) > Val(C) Then
            M = A
        Else
            M = C
        End If
    Else
        If Val(B) > Val(C) Then
            M = B
        Else
            M = C
        End If
    End If
End Sub
```

Note aquí los principios de programación al poner la estructura de decisiones con anidamientos, puede ud., poner el flujograma al lado derecho del código.

Los demás botones deben tener el código ya conocido y tratado en prácticas anteriores, ubíquelos para que pueda ver el funcionamiento correcto del formulario, que más o menos sería así:

Los **operadores relacionales** que se pueden utilizar son:

> Mayor que ; > = Mayor o igual que  
< Menor que ; < = Menor o igual que

Los **operadores lógicos** que se pueden utilizar son:

AND = Y ; OR = O ; NOT = NO

Utilizando los operadores lógicos el código del botón **Obtener mayor** sería el siguiente:

```
Private Sub obtener_Click()
    If Val(A) > Val(B) AND Val(A) > Val(C) Then
        M = A
    Else
        If Val(B) > Val(C) Then
            M = B
        Else
            M = C
        End If
    End If
End Sub
```

Lo cual nos daría el mismo resultado.

### **TRABAJO PARA EL ALUMNO:**

- Obtenga el mayor de 4 números,
- Obtenga El mayor, el menor y el medio de tres números
- Valor por la compra: en base al número de productos comprados y al valor de cada producto si el valor supera los 4 dólares incrementa el 12% IVA despliegue ese valor y el valor total a pagar.
- Obtenga el dígito verificador de la cédula de identidad
- Ingrese un día y un mes y despliegue las cualidades del signo correspondiente, ej: número y día de suerte, piedra, flor, color, afinidades, etc.

## GUIA PRACTICA # 7

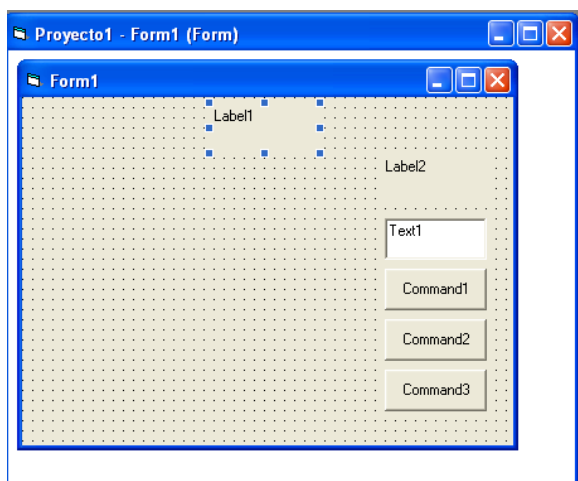
### TEMA: PROGRAMACION TRADICIONAL (BUCLES- CONTADORES):

#### TABLA DE MULTIPLICAR DE UN NUMERO CUALQUIERA

En Visual Basic tiene la misma estructura de las estructuras de control tradicionales utilizadas en cualquier lenguaje común (While...Wend; For...Next) , es por esto que si ud. Amable lector no tiene conocimientos sólidos de lo que son las decisiones, su estructura, su utilización se le sugiere, que trate de aprenderlos (utilice los libros de esta misma colección Qbasic, Pascal, Fox).

#### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 1 TEXTBOX
4. Ubique 2 LABEL
5. Ubique 3 botones de comando(COMMAND BUTTON)
6. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:

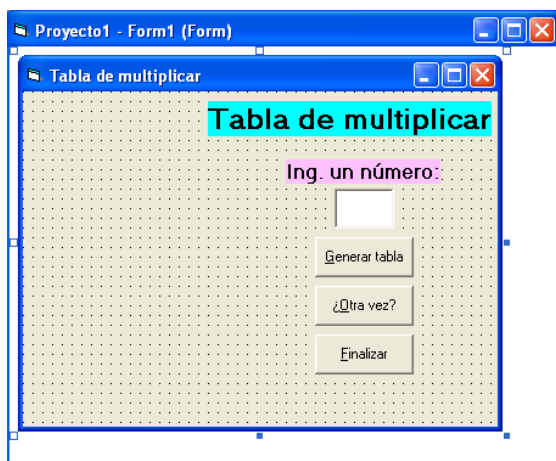


Vamos a hacer el cambio de las propiedades de los objetos, recuerde que los colores, tipos de letras, tamaños, entre otros, ya se quedan a su gusto o criterio, además en LABEL recuerde que debe poner la propiedad AutoSize= true para que se autoajuste el texto, igual en TEXTBOX cambie la propiedad Text y déjela vacía (sin nada escrito), para que al ejecutar el formulario no aparezca nada escrito.

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption : Form1	Caption : Tabla de multiplicar
Label1	Caption : Label1	Caption : Tabla de multiplicar
Label2	Caption : Label2	Caption : Ing. Un número
Command1	Caption : Command1 Nombre : Command1	Caption : &Generar tabla Nombre : generar
Command2	Caption : Command2 Nombre : Command2	Caption : ¿&Otra vez? Nombre : otravez
Command3	Caption : Command3 Nombre : Command3	Caption : &Finalizar Nombre : fin

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Text1	Nombre : Text1	Nombre: A

Hasta ahora su formulario debe lucir similar al siguiente:



Ahora vamos a ubicar el código correspondiente en cada botón.

Haga doble click en la Caja de texto (Text1) y ubique el siguiente texto:

```
Private Sub a_Change()
    a.SetFocus
End Sub
```

Representa al evento que va a realizar la caja de texto llamada a, en relación al cambio que tenga (Change), **a.setfocus** permite poner el cursor en la caja de texto a, cuando se ejecute el formulario.

Haga doble click en GENERAR TABLA y ubique el código de tal manera que tenga este texto:

```
Private Sub generar_Click()
    For i = 1 To 12
        m = i * a
        Print i; " * "; a; " = "; m
    Next i
End Sub
```

Es un bucle automático que comienza desde 1 hasta 12 en la variable I, se realiza la multiplicación tomando el valor del bucle, el valor de la caja de texto y se acumula la multiplicación en m ( $m = i * a$ ), posteriormente, se despliega en formato de tabla (Print i; " \* "; a; " = "; m), y se cierra el bucle repetitivo (Next i)

Haga doble click en ¿OTRA VEZ? Y escriba el código correspondiente, así:

```
Private Sub otravez_Click()
    a=""
    Form1.Cls
End Sub
```

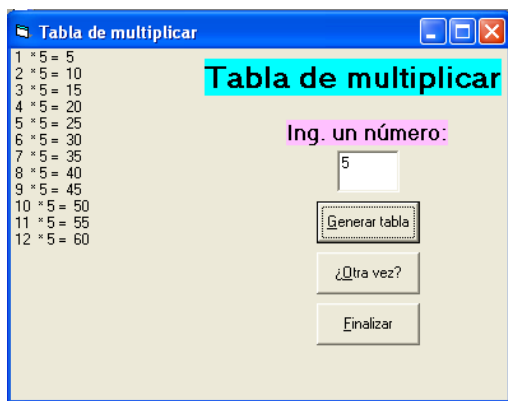
```
Form1.Show
End Sub
```

Aquí hemos aumentado algo nuevo Form1.cls, que permite limpiar (Clear Screen) el formulario para dejarlo en blanco y poder utilizarlo nuevamente.

Y por último haga doble click en Finalizar y escriba el código ya conocido:

```
Private Sub fin_Click()
    End
End Sub
```

Ejecute el formulario y observe el resultado que debe ser parecido al siguiente:



Igual que el for....next, existen otros bucles o contadores, en este caso, no automáticos, sino, condicionales, como el **WHILE...WEND**, es decir, que en el Botón de comando (Generar tabla) podemos ubicar otra programación o código, así:

```
Private Sub generar_Click()
    i = 1
    While i <= 12
        m = i * a
        Print i; " * "; a; " = "; m
        i = i + 1
    Wend
End Sub
```

i está en primer lugar en el evento tomando el valor de 1, la comparación que se hace es Mientras (While) i sea menor o igual a 12 haga lo siguiente,: realice la multiplicación tomando el valor del bucle condicional, el valor de la caja de texto y se acumula la multiplicación en m ( $m = i * a$ ), posteriormente, se despliega en formato de tabla (Print i; " \* "; a; " = "; m), acumula el bucle condicional ( $i=i+1$ ) y se cierra el bucle (Wend).

También podemos hacer lo mismo con el bucle condicional **DO UNTIL...LOOP**, así:

```
Private Sub generar_Click()
    i = 1
    Do Until i > 12
```

```
m = i * a
Print i; " * "; a; " = "; m
i = i + 1
Loop
End Sub
```

Do until significa (Hacer hasta), en este caso hacer hasta que i sea mayor que 12, y se repite lo mismo reemplazando Wend por Loop.

El resultado será el mismo al ejecutar el formulario con cualquiera de los bucles tratados sea automático o condicional.

### **Recuerde que:**

- Un bucle automático no necesita un número de asignación, ni un incremento dentro de él, por eso se llama automático, pues, va incrementando automáticamente 1 cada vez que se cumple hasta completar el rango.
- Un bucle condicional, debe tener un número de asignación inicial, y obligatoriamente un incremento dentro de su estructura, sino, no se cumplirá nunca el bucle.

### **TRABAJO PARA EL ALUMNO:**

- Despliegue en el formulario 20 veces la palabra “hola”
- Despliegue en el formulario N veces su nombre
- Despliegue en el formulario los números secuenciales, al frente el cuadrado y el cubo del mismo número, así.

Numero	cuadrado	cubo
1	1	1
2	4	8
3	9	27
4	16	64

- Genere los 10 múltiplos de un número ingresado por teclado
- Genere la tabla de fibonacci
- Ingrese un número y despliegue si es primo o no
- Genere la tabla de números primos

## GUIA PRACTICA # 8

### TEMA: PROGRAMACION TRADICIONAL (OPCIONES SELECT CASE):

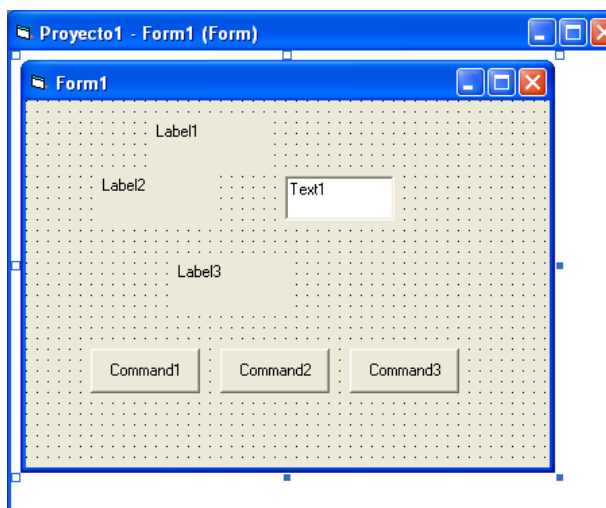
#### En base a ingresar un número muestre el día de la semana al que corresponda

Si bien Visual Basic tiene estructuras de decisión bien formadas y fáciles de utilizar, existe otra opción que a muchos programadores les resulta más sencillo de utilizar, esto es Select case....End select.

Esta estructura permite escoger opciones dependiendo del valor asignado o dado a una variable.

#### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 1 TEXTBOX
4. Ubique 3 LABEL
5. Ubique 3 botones de comando(COMMAND BUTTON)
6. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



Vamos a hacer el cambio de las propiedades, algunas de ellas deben ser a su gusto o necesidad, otras (las nuevas), serán tratadas en el presente texto, así:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption : Form1	Caption : Día de la semana
Label1	Caption : Label1	Caption : Día de la semana
Label2	Caption: Label2	Caption: Ing. Un número de 1 a 7
Label3	Caption: Label3 Visible: True Nombre:Label3	Caption: (borre todo, déjelo vacío) Visible: False Nombre: día
Text1	Nombre : Text1 Text:Text1	Nombre: A Text: (borre todo, déjelo vacío)
Command1	Caption : Command1 Nombre : Command1	Caption : &Día correspondiente Nombre : generar
Command2	Caption : Command2	Caption : ¿&Otra vez?

	Nombre : Command2	Nombre : otra vez
Command3	Caption : Command3 Nombre : Command3	Caption : &Finalizar Nombre : fin

La mayoría de objetos poseen la propiedad `VISIBLE` como `TRUE` o `FALSE`, la primera hace que se observe en el formulario ejecutado, la segunda en cambio lo oculta y se hace invisible en la ejecución, a pesar de estar presente ahí.

Luego de haber realizado los cambios, el formulario debe tener la siguiente apariencia:



Ahora, generemos el código, escriba en los eventos, según los objetos correspondientes, así:

Doble click en la Caja de texto (Text1):

```
Private Sub a_Change()
    a.SetFocus
End Sub
```

Doble click en el botón de comando (Día correspondiente):

```
Private Sub generar_Click()
    Dim y As String
    x = Val(a)
    Select Case x
        Case 1
            y = "LUNES"
        Case 2
            y = "MARTES"
        Case 3
            y = "MIERCOLES"
        Case 4
            y = "JUEVES"
        Case 5
            y = "VIERNES"
        Case 6
            y = "SABADO"
    End Select
End Sub
```



```

Case 7
  y = "DOMINGO"
Case Is > 7
  y = "Error solo son 7 días"
End Select
dia.Caption = "Corresponde a " + y
dia.Visible = True
End Sub

```

Explicación: Dim y as String , permite definir la variable y de tipo carácter , x=Val(a), a pesar de haberlo tratado ya en prácticas anteriores transforma a valor numérico un ingreso de tipo texto(TextBox), Select Case x, permite seleccionar los casos según el valor ingresado, así:, en el caso de que se ingrese el número 1 (Case 1), se asigna la palabra “LUNES” a la variable y, así sucesivamente, por último, se cierra End Select y se cambia la propiedad al objeto de nombre dia(LABEL):

```
dia.Caption = "Corresponde a " + y
```

El label dia tendrá como texto escrito (Caption) mas (+) lo asignado en la variable y según los casos, el más(+) permite unir dos cadenas de caracteres.

```
dia.Visible = True
```

por último, el objeto dia, cambiará la propiedad visible de falso a verdadero, para poder observarlo en el formulario.

Esto que hemos realizado se denomina cambios en **tiempo de ejecución** y lo que hacemos antes de ejecutar el formulario se llama cambios en **tiempo de diseño**.

Doble click en el botón ¿Otra vez?, y realice:

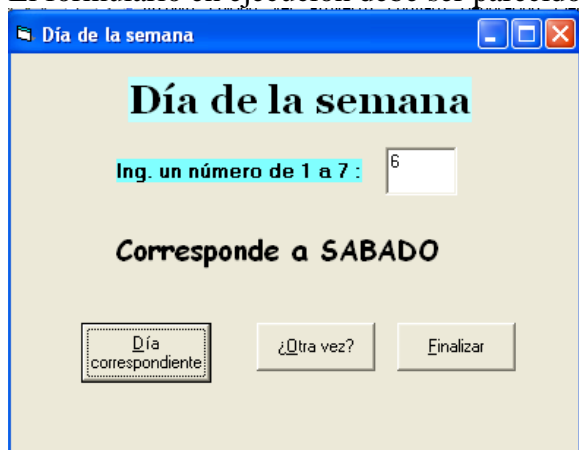
```

Private Sub otravez_Click()
  a = " "
  dia.Visible = False
  Form1.Show
End Sub

```

El botón de finalizar ya es por usted conocido, prepárelo para su funcionamiento

El formulario en ejecución debe ser parecido al siguiente:



## GUIA PRACTICA # 9

### TEMA: INPUTBOX () Y MSGBOX():

En base a ingresar la edad se debe desplegar un mensaje en base a la siguiente consideración:

- Si es menor que 13 años “ES UN NIÑO APENAS”
- Si tiene de 13 a 19 años “ES UN ADOLESCENTE”
- Si tiene de 20 a 30 años “ES UN ADULTO MADURO”
- Si tiene más de 30 años “QUE LINDA EDAD”

**Qué son InputBox() e MsgBox()???**- No son nada más que cuadros de comunicación entre el usuario y el computador que hacen posible una interactividad más atractiva para ingresar y presentar valores.

InputBox (Caja de ingreso).- Permite ingresar valores.

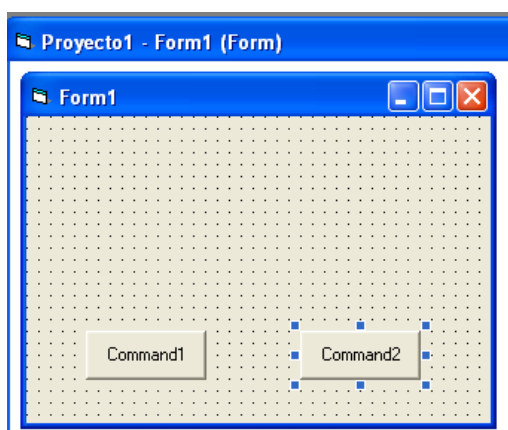
MsgBox (Caja de mensajes).- Permite mostrar resultados o valores

Cuando se utiliza windows o cualquiera de los programas de Office, muchas veces se puede cometer un error, entonces, nos sale una “ventanita” mostrándonos el error o aviso, pues, eso es un MsgBox.

Para comprender como se los coloca o como se les utiliza y cual es su funcionamiento en si, hagamos una practica:

### **PRACTICA:**

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Botones de comando (COMMAND BUTON)
4. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



Ahora cambiaremos las propiedades, así:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Manejo de cajas
Command1	Caption: Command1 Nombre: Command1	Caption: &pedir Edad Nombre: ingedad

Command2	Caption: Command2 Nombre: Command2	Caption: &Finalizar Nombre: fin
----------	---------------------------------------	------------------------------------

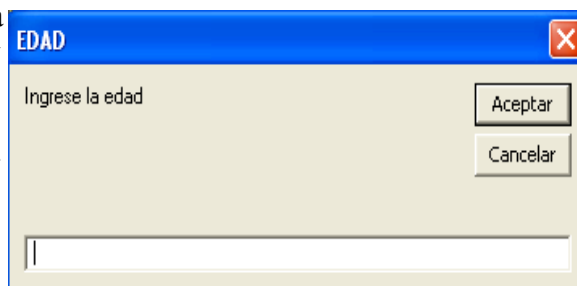
Comencemos a poner el código para los eventos

```
Private Sub ingedad_Click()
    Dim EDAD As Integer
    EDAD = InputBox("Ingrese la edad", "EDAD", , 1000, 1000)
    Select Case EDAD
        Case Is < 13
            x = MsgBox("ES UN NIÑO APENAS", vbInformation)
        Case 13 To 20
            x = MsgBox("ES UN ADOLESCENTE", vbExclamation)
        Case 20 To 39
            x = MsgBox("ES UN ADULTO MADURO", vbCritical)
        Case Else
            x = MsgBox("QUE LINDA EDAD", vbYesNoCancel)
    End Select
End Sub
```

### **Explicación:**

Dim EDAD As Integer.- Define la variable EDAD como entera

EDAD = InputBox("Ingrese la edad", "EDAD", , 1000, 1000).- Permite mostrar un cuadro de ingreso con el mensaje "ingrese la edad", El título de la ventana será EDAD y la ubicación que tendrá en la pantalla será 1000,1000 en filas y columnas, las comas y espacio en blanco permite poner un valor por defecto, el resultado de esto es más o menos así cuando esta siendo ejecutado:

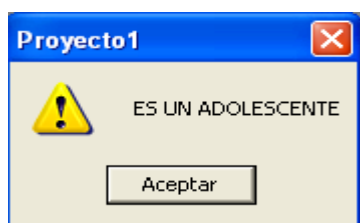


Allí se ingresará un valor y se hará click en Aceptar o cancelar, lo cual devolverá un valor para que sea analizado en el CASE como vimos en la práctica anterior, observe que los casos son diferentes que en el ejercicio anterior, aquí se le toma por rangos o especificaciones.

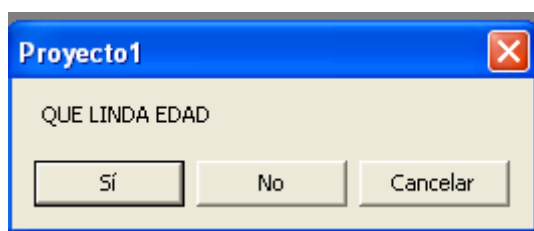
```
x = MsgBox("ES UN NIÑO APENAS", vbInformation)
x = MsgBox("ES UN ADOLESCENTE", vbExclamation)
x = MsgBox("ES UN ADULTO MADURO", vbCritical)
x = MsgBox("QUE LINDA EDAD", vbYesNoCancel)
```

Son una serie de textos que aparecerán en cajas de mensajes, con la cualidad de tener un ícono representativo que se le ubica de acuerdo al criterio o necesidad del programador, esto está especificado, es decir, si se ubica vbInformation, saldrá un ícono de Información, como en los siguientes ejemplos:

La x es la variable que obtendrá el resultado de la acción que se haya ubicado



vbExclamation



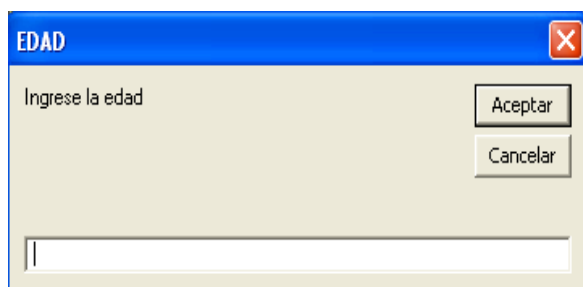
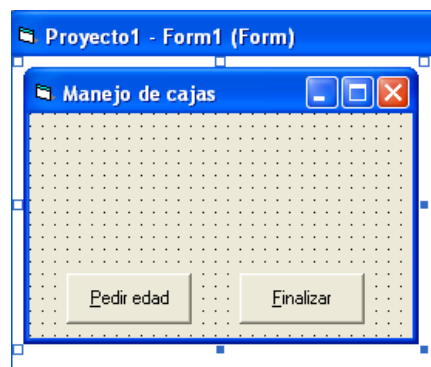
vbYesNoCancel

Recuerde que los botones Aceptar, Si, No, cancelar, nosotros a pesar de no haberlos programado, los hemos ubicado a través de su especificación como en las figuras.

Cuando ejecute su formulario seguramente tendrá apariencia similar a esta:

Observe que existen dos botones, el de Finalizar ud., ya lo puede hacer,

Si hace click en Pedir edad, obtendrá Esta otra ventana:



Es la ventana donde Ud., debe ingresar el número correspondiente a la edad, y, dependiendo del ingreso que se haga se tendrá el resultado correspondiente, como los tratados anteriormente.

### TRABAJO PARA EL ALUMNO:

- Consulte cuantos estilos de botones o íconos para msgbox existen
- Consulte los formatos completos de InputBox() y MsgBox() con ejemplos
- En base al ingreso de un número con un TEXTBOX, despliegue el correspondiente mes del año usando un LABEL para la respuesta..
- En base al ingreso de un número con INPUTBOX, despliegue el correspondiente mes del año usando un MSGBOX para la respuesta..
- Realice un formulario para conocer el signo del zodiaco al que pertenece una persona al ingresar el día y su mes de nacimiento, despliegue además la edad que tiene, utilice TEXTBOX y LABEL o TEXTBOX para las respuestas.
- Realice un formulario para conocer el signo del zodiaco al que pertenece una persona al ingresar el día y su mes de nacimiento, despliegue además la edad que tiene, utilice INPUTBOX y MSGBOX para las respuestas.
- La matrícula de un vehículo se basa en su valor de comercialización o avalúo, así, los carros que cuestan menos de 5000 dólares deben pagar el 10% del costo y reciben un adhesivo celeste, los que cuestan entre 5000 y 10000, pagan el 20% del costo y reciben un adhesivo amarillo, y los que cuestan más de 10000 pagan un 30% del costo y reciben un adhesivo blanco, genere un formulario para que en base al avalúo se de los resultados automáticamente, utilice los objetos que crea conveniente para lograr tal objetivo.

## GUIA PRACTICA # 10

### TEMA: UTILIZACION DE VARIABLES :

Hasta ahora hemos utilizado variables en nuestros proyectos (prácticas), pero aún no las hemos tratado como debe ser, pues, ellas tienen un alcance y pueden ser de varios tipos de acuerdo a necesidad del programador.

Una variable no es más que un espacio de memoria identificado con un nombre, capaz de contener un cierto tipo de datos que puede modificarse tanto en tiempo de diseño como en tiempo de ejecución. Los nombres de las variables deben cumplir con ciertos requisitos:

- Pueden tener hasta 255 caracteres
- Comenzar con un carácter alfabético (sino, sería considerado un valor numérico)
- No pueden tener caracteres especiales(·\$%/#@), pero si un guión de subrayado
- No pueden ser palabras reservadas
- Pueden combinarse con números pero siempre debe comenzar con una letra

Así los nombres: EDAD, E1, EDAD\_1, A, B, AB1 Son variables correctas, en cambio los nombres: 1EDAD, N#1, A 1, DIM , son variables incorrectas que producirán errores.

### Tipos de datos que se puede utilizar en Visual Basic:

Tipo	Subfijo	Tamaño de almacenamiento	Rango
Byte	Ninguno	1 Byte	De 0 a 255
Boolean	Ninguno	2 Bytes	True o False
Integer	%	2 Bytes	-32.768 – 32.768
Long(Integer)	&	4 Bytes	-2.147.483.648 a 2.147.483.647
Single(Precision)	!	4 Bytes	-3.40E38 a 3.40E38
Double(Precision)	#	8 Bytes	-1.80E308 a 1.80E308
Currency	@	8 Bytes	-9.22E14 a 9.22E14
Date	Ninguno	8 Bytes	1 de enero,100- 31 de diciembre, 9999
Object	Ninguno	4 Bytes	Cualquier referencia de objeto
String	\$	1 Byte/Caract	Aprox. 65.500 Max Bytes
Variant	Ninguno	X Necesidad	Determinado por el tipo necesario
Definida usuario	Ninguna	X Necesidad	Determinado por el tipo de elemento

Si ud no ha declarado un variable, Visual basic lo toma como Variant que será general, es decir puede ser tomado como cualquier tipo, según el proceso que tenga que hacer el formulario(código de evento).

Se puede declarar una variable **implícitamente**, es decir, poniendo un subfijo al final de la misma, así: puede indicar que la variable EDAD será entera poniendo un % :  
EDAD%

Entonces el compilador reconocerá la variable y su tipo, si no se ha definido ningún tipo se convertirá en variant.

Otra forma de declarar una variable es **explícitamente**, con la utilización de la instrucción DIM, así:

```
Dim E As Integer
Dim DOB As Double
Dim Mimens, Tumens As String
Dim Mi_Nombre As String * 11
```

En la primera línea la variable E esta definida como entera, en la segunda la variable DOB como doble, pero ponga atención a la tercera línea, que declara dos variables al mismo tiempo, pero, Mimens es de tipo variant, pues no se está especificando el tipo en forma **Explícita**, mientras que Tumens es de tipo carácter, la última línea define la variable Mi\_nombre como una cadena de caracteres de 11 espacios disponibles.

## **Alcance de una variable**

El alcance de una variable se refiere a las áreas del programa en las cuales se puede utilizar o aplicar, aquí existen cuatro niveles en las que se las puede declarar:

- Nivel de procedimiento
- Variables estáticas dentro del procedimiento
- Nivel de forma
- Nivel de módulo estándar(código)

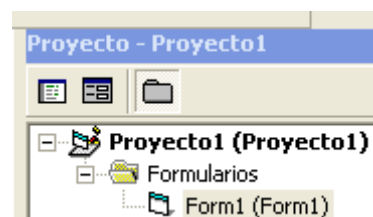
Hasta ahora hemos definido solo variables **locales o privadas**, esto es dentro de un **procedimiento** con sub...end sub(recuerde Dim EDAD As Integer, en la práctica anterior), estas tienen un alcance un tanto “estrecho” pues solo sirven para tal procedimiento y no se les puede utilizar en otros.

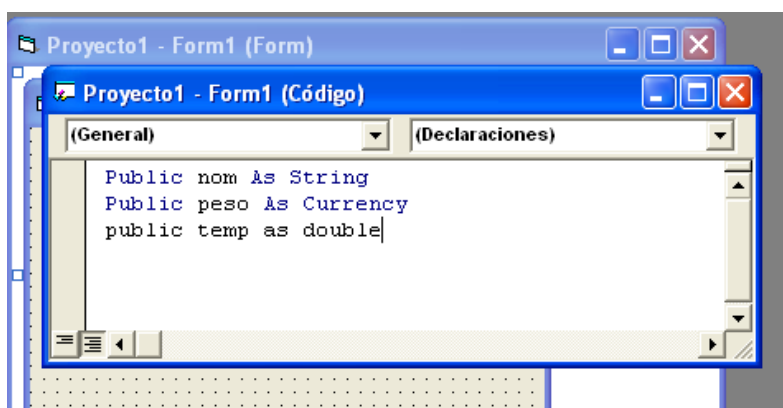
Las variables locales no sobreviven al procedimiento en el cual están definidas, cuando un procedimiento termina todas las variables(locales) son removidas de la memoria, lo cual es bueno porque reserva recursos, pero, lo malo es que son reinicializadas cada vez que se ejecuta el procedimiento, lo cual merma recursos. Para no permitir esta reinicialización se debe utilizar la opción **Static** en vez de **Dim**.

Si se declara una variable a **nivel de forma** se puede utilizar en todos los procedimientos existentes, de esta manera se puede compartir información entre procedimientos, lo mismo sucede con los **módulos de código**, solo que en este último hay que crear un **Módulo** del menú **insert** (presente también en la barra de herramientas), a este tipo de variables también se les conoce como **Globales o públicas** por su alcance.

Se define una variable Global o pública en el mismo formulario, acceda al ícono **ver el código** que se encuentra en el explorador de proyectos:

Observe la siguiente ventana, e implemente las variables tal como lo presenta el siguiente gráfico:

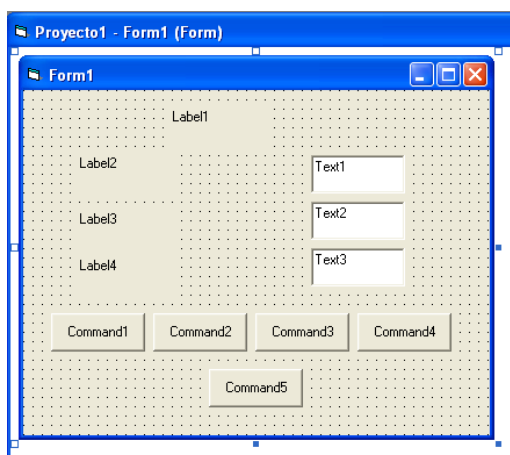




Constituye las declaraciones generales para todo el proyecto, es decir que, si se define variables aquí, se podrán utilizar en todos los procedimientos.

**PRACTICA:**

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 4 LABEL (etiquetas)
4. Ubique 3 TEXTBOX
5. Ubique 5 Botones de comando (COMMAND BUTON)
6. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:

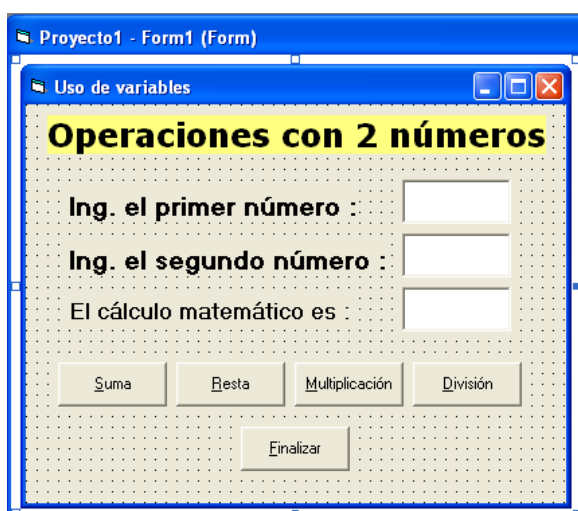


Cambiamos ahora las propiedades de acuerdo a nuestra necesidad o gusto y en base a la siguiente descripción:

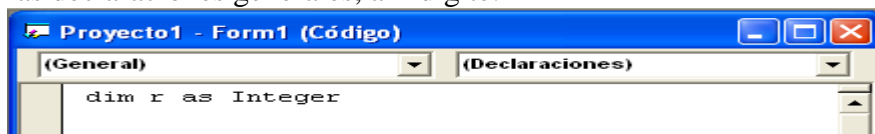
OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Uso de Variables
Label1	Caption: Label1	Caption: Operaciones con 2 numeros
Label2	Caption: Label2	Caption: Ing. Primer número :
Label3	Caption: Label3	Caption: Ing. Segundo número :
Label4	Caption: Label4	Caption: El cálculo matemático es :
Text1	Nombre: Text1 Text:Text1	Nombre: a Text: Vacío, borre todo
Text2	Nombre: Text2 Text:Text2	Nombre: b Text: Vacío, borre todo
Text3	Nombre: Text3 Text:Text3	Nombre: respuesta Text: Vacío, borre todo

Command1	Caption: Command1 Nombre: Command1	Caption. &Suma Nombre: suma
Command2	Caption: Command2 Nombre: Command2	Caption. &Resta Nombre: resta
Command3	Caption: Command3 Nombre: Command3	Caption. &Multiplicación Nombre: multi
Command4	Caption: Command4 Nombre: Command4	Caption. &División Nombre: divi
Command5	Caption: Command5 Nombre: Command5	Caption. &Finalizar Nombre: fin

Una vez cambiadas las propiedades, nuestro formulario debe tener similar apariencia a esta:



Diríjase a la ventana del explorador de proyectos y haga click en ver el código y tendrá las declaraciones generales, allí digite:



Estamos definiendo una variable pública o global que puede ser utilizada en todos los procedimientos. Luego, haga doble click en el botón suma y escriba el código respectivo, así:

```
Private Sub suma_Click()
    r = Val(a) + Val(b)
    respuesta.Text = r
End Sub
```

Doble click en resta y escriba:

```
Private Sub resta_Click()
    r = Val(a) - Val(b)
    respuesta.Text = r
End Sub
```



Lo mismo en los botones de multiplicación y división, así:

```
Private Sub multi_Click()
    r = Val(a) * Val(b)
    respuesta.Text = r
End Sub
```

```
Private Sub divi_Click()
    r = Val(a) / Val(b)
    respuesta.Text = r
End Sub
```

El boton de finalización ud., ya lo sabe, prográmelo.

Ejecute su formulario que debe tener la siguiente apariencia.

Observe que hemos identificado la variable r de la respuesta como general y la hemos utilizado en 4 procedimientos diferentes.

Si desea obtener el resultado con decimales cambie la definición de la variable por BOOLEAN, es decir: Dim r As Boolean en vez de Dim r As Integer y ejecute el formulario para que vea la respuesta.

Puede clicar en los botones para observar el cambio en las operaciones.

### **TRABAJO PARA EL ALUMNO:**

Aplicando variables globales y locales realice los siguientes formularios:

- Multiplicación de tres números
- Las cuatro operaciones básicas con dos números
- Area de un triángulo  $(B*H)/2$
- Valor por la compra: en base al número de productos comprados y al valor de cada producto.
- Del programa anterior, incremente el 12% del IVA y despliegue ese valor y el valor total a pagar.
- Programas con fórmulas, masa, peso, densidad, áreas, perímetros, etc.

## GUIA PRACTICA # 11

### TEMA: BOTONES DE OPCION Y MARCOS:

Al utilizar aplicaciones de windows, seguramente nos hemos topado con formularios que presentan al usuario algunas opciones que pueden activarse al mismo tiempo, por ejemplo, establecer el fondo de escritura de un color y la letra de otro color.

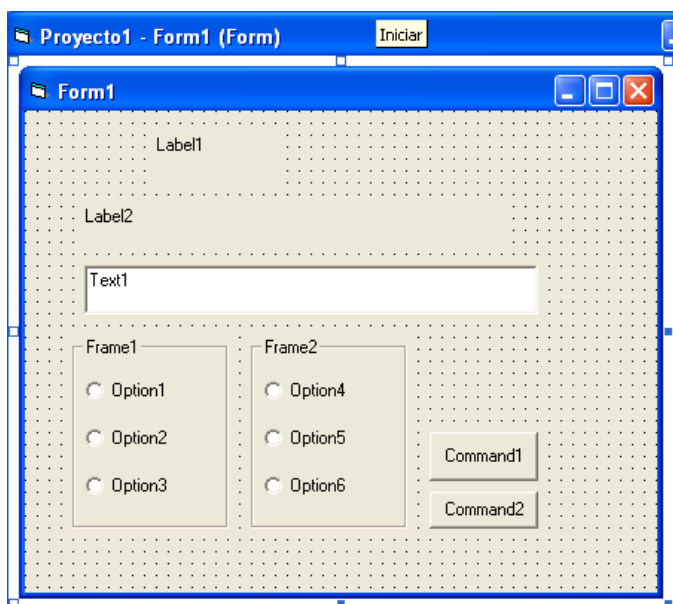
Existe en la barra de herramientas dos íconos importantes los **botones de opción**(Option button), que muestra la opción de activado y desactivado, encendido y apagado, obviamente cuando está activo hará alguna acción y lo mismo cuando esté desactivado; y los **Marcos**, que proporcionan una manera adecuada para agrupar estos controles(botones de opción), los marcos permiten hacer estas agrupaciones y además puede servir para dividir un formulario.

Imagínese que va a necesitar 6 botones de opción y tenerlos individualmente en el formulario sería una verdadera molestia, es preferible encerrarlos en un Marco (Frame), y trabajarlo como todo un conjunto (de una sola).

Entonces, vamos a utilizar botones de opción los cuales al estar seleccionados darán un resultado visible en la pantalla, dentro de un frame(Marco) para trabajarlos como un solo objeto.

### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Label
4. Ubique 1 textBox
5. Ubique 2 FRAMES (Marcos)
6. Ubique 6 Button option 3 en cada frame (de click en option button, váyase al frame y diseñelo ahí dentro de frame)
7. Ubique 2 Botones de comando (COMMAND BUTON)
8. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:

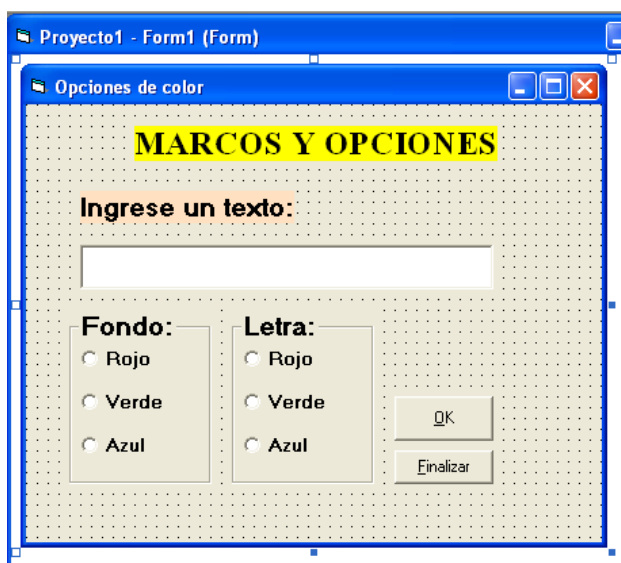


Recuerde que los botones de opción deben estar dentro de los marcos.

Cambiamos ahora las propiedades de acuerdo a nuestra necesidad o gusto y en base a la siguiente descripción:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Opciones de Color
Label1	Caption: Label1	Caption: MARCOS Y OPCIONES
Label2	Caption: Label2	Caption: Ingrese un texto
Text1	Text:Text1 Nombre: Text1	Text: Vacio borre todo Nombre: A
Frame1	Caption: Frame1	Caption: Fondo
Frame2	Caption: Frame2	Caption: Letra
Option1	Caption: Option1 Nombre: frojo	Caption: frojo Nombre: frojo
Option2	Caption: Option2 Nombre: fverde	Caption: fverde Nombre: fverde
Option3	Caption: Option3 Nombre: fazul	Caption: fazul Nombre: fazul
Option4	Caption: Option4 Nombre: trojo	Caption: trojo Nombre: trojo
Option5	Caption: Option5 Nombre: tverde	Caption: tverde Nombre: tverde
Option6	Caption: Option6 Nombre: tazul	Caption: tazul Nombre: tazul
Command1	Caption:Command1 Nombre: Command1	Caption: &OK Nombre: ok
Command2	Caption:Command2 Nombre: Command2	Caption: &Finalizar Nombre: fin

Una vez cambiadas las propiedades, nuestro formulario debe presentar una apariencia similar a la siguiente:



Recuerde que el diseño del formulario en lo que respecta al tipo y tamaño de letra es a su gusto, por ejemplo, de los marcos (frames) y opciones (Option button), son negritas, de 10 o 12 de tamaño, tipo Arial, de igual manera los tamaños de cada uno de los componentes depende de su necesidad o gusto, lo mismo que los colores de letra y fondo de los objetos que son cosas que ud., ya aprendió anteriormente y es por eso que no se les aclara en las propiedades del cuadro anterior.

Ahora, pondremos el código respectivo a los objetos que serán utilizados:

Haga doble click en el botón de OK y escriba el siguiente código:

```
Private Sub ok_Click()
    If frojo Then
        a.BackColor = &HFF8
    ElseIf fverde Then
        a.BackColor = &HC000&
    ElseIf fazul Then
        a.BackColor = &HC00000
    End If

    If trojo Then
        a.ForeColor = &HFF&
    ElseIf tverde Then
        a.ForeColor = &HC000&
    ElseIf tazul Then
        a.ForeColor = &HC00000
    End If
End Sub
```

Observe que la codificación se hará solo en este botón allí se tomarán las decisiones de acuerdo a lo que se haya activado, ejemplo: si está activado rojo el objeto a, cambiará de fondo a &HFF8 (que representa rojo), en:

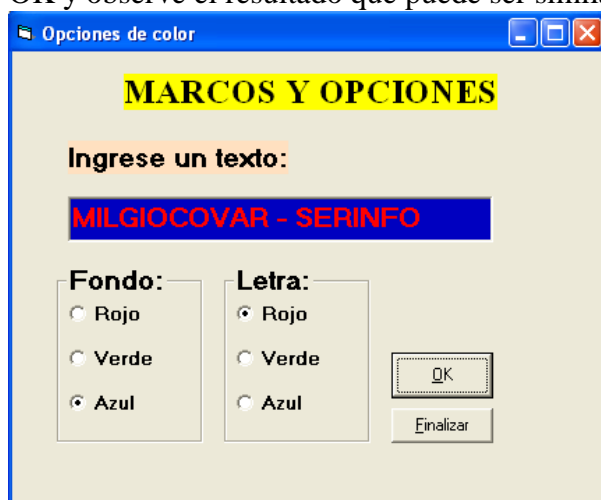
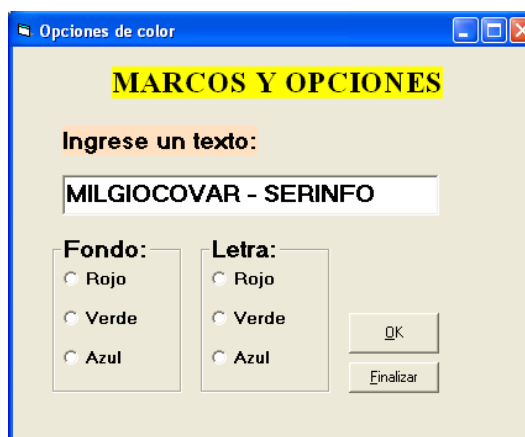
```
If frojo Then
    a.BackColor = &HFF8
```

Lo mismo sucede con las otras opciones.

El botón de finalizar ud., ya puede hacerlo solo.

Ejecute su formulario y seguramente tendrá un resultado similar al siguiente:

Tendrá que ingresar cualquier texto en el objeto Text, luego seleccione las opciones tanto para fondo como para la letra y haga click en OK y observe el resultado que puede ser similar al siguiente:



Aquí están activas las opciones de azul en el fondo y rojo en la letra, observe que solo puede tener un activado a la vez, es decir, en el marco de fondo solo se puede activar uno solo a la vez al igual que en el otro marco, esa es la gran diferencia con las casillas de verificación que es el tema de nuestra siguiente práctica.

## GUIA PRACTICA # 12

### TEMA: CASILLAS DE VERIFICACION Y MARCOS:

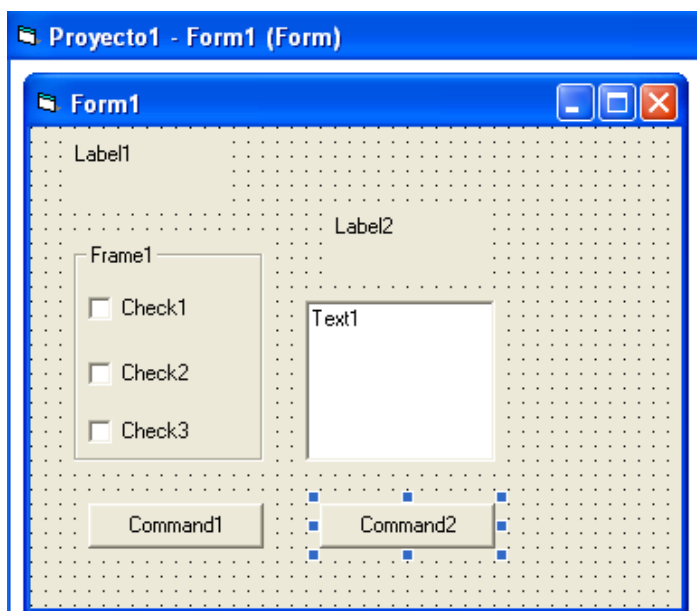
En la práctica anterior, al utilizar los botones de opción solo podíamos escoger una de ellas a la vez, en cambio con las casillas de verificación se puede activar una o varias, dependiendo de la necesidad del programador o usuario.

Pero...¿Qué son las casillas de verificación?... (Check Box) muestra una marca cuando está seleccionado (parecido a option button), y obviamente vacío cuando no lo está, de esta manera el usuario puede seleccionar algunas opciones no solo una, por ejemplo: cuando se trabaja con algunas opciones de windows le dan la posibilidad de “Habilitar” el color, el sonido, el modem, el ratón, entre otros.

Igual que option button es preferible trabajar con CheckBox si es más de uno dentro de un frame para agruparlo y hacerlo más comprensible y fácil.

### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Label
4. Ubique 1 textBox
5. Ubique 1 FRAMES (Marcos)
6. Ubique 3 CcheckBox (de click en CheckBox, váyase al frame y diseñelo ahí dentro de frame)
7. Ubique 2 Botones de comando (COMMAND BUTON)
8. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



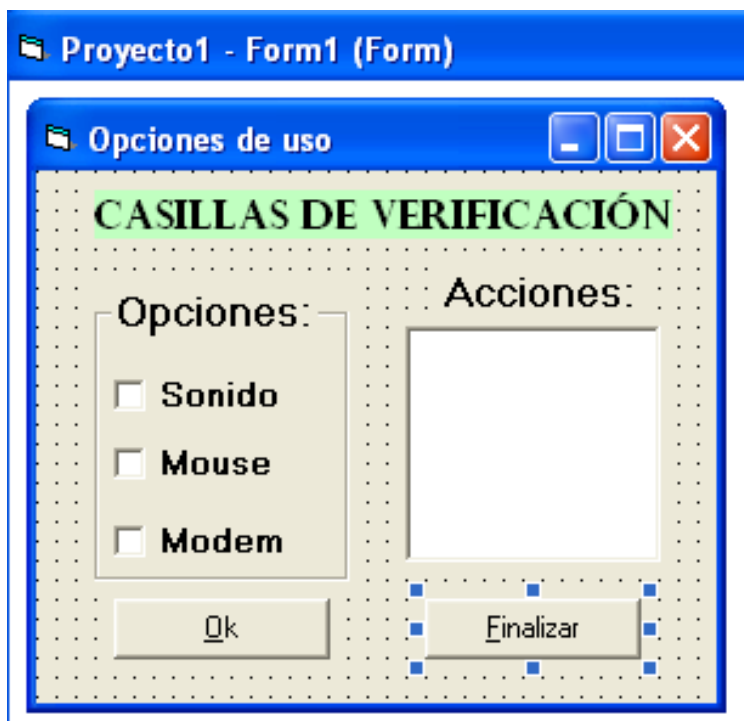
Importante: Recuerde que las casillas de verificación (CheckBox) deben estar dentro de los Marcos (Frames)

Cambiamos las propiedades de acuerdo a nuestra necesidad o gusto y en base a la siguiente descripción:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Opciones de uso
Label1	Caption: Label1	Caption: Casillas de verificación
Label2	Caption: Label2	Caption: Acciones
Frame1	Caption: Frame1	Caption: Opciones
Check1	Caption: Check1 Nombre: Check1	Caption: Sonido Nombre:monido
Check2	Caption: Check2 Nombre: Check2	Caption: Mouse Nombre:mouse
Check3	Caption: Check3 Nombre: Check3	Caption: Modem Nombre:modem
Command1	Caption:Command1 Nombre: Command1	Caption: &OK Nombre: ok
Command2	Caption:Command2 Nombre: Command2	Caption: &Finalizar Nombre: ok

Para el cuadro de texto TEXTBOX, déjelo en blanco y póngale de nombre a, pero, además cambie la propiedad **Multiline** a true, lo que permite que la caja de texto no admita una sola línea sino, varias.

Una vez cambiadas las propiedades, nuestro formulario deberá presentar un aspecto similar al siguiente:



Ahora vamos a adicionar el código respectivo a los botones necesarios. Haga doble click en el botón de OK y escriba lo siguiente:

```
Private Sub ok_Click()

    Dim texto As String

    If sonido = Checked Then
        texto = texto & " Sonido activado "
    Else
        texto = texto & " Sonido desactivado "
    End If

    If mouse = Checked Then
        texto = texto & VBCRLF & " Mouse activado "
    Else
        texto = texto & VBCRLF & " Mouse desactivado "
    End If

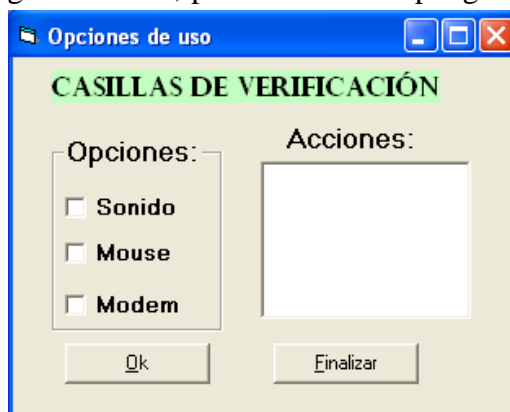
    If modem = Checked Then
        texto = texto & VBCRLF & " Modem activado "
    Else
        texto = texto & VBCRLF & " Modem desactivado "
    End If

    a = texto

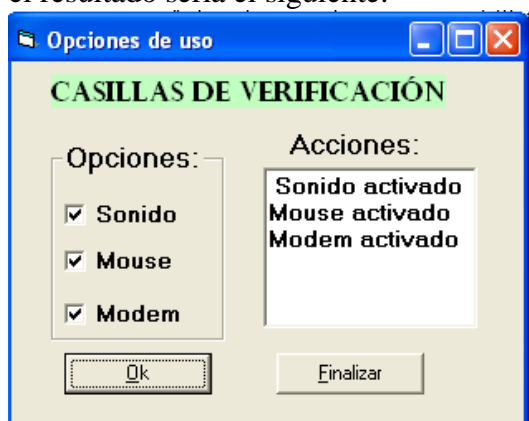
End Sub
```

Importante: Primero dimensionamos la variable texto como string (Caracter), luego realizamos decisiones para comprobar cuales de ellos están activados y se ubica el mensaje correspondiente (recuerde que el carácter & permite unir dos cadenas de caracteres), y CRLF permite hacer un salto a la siguiente línea, por último se despliega el texto a través de a (caja de texto multilínea).

Al ejecutar el formulario inicialmente Aparecerá de la siguiente manera:



Active las que desea y ponga OK para ver el resultado, en este caso, por ejemplo: si activamos las tres opciones, el resultado sería el siguiente:



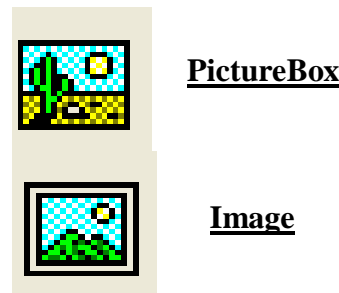
He aquí la diferencia entre botones de opción y casillas de verificación, en estas últimas no solo puede ser una opción la escogida sino algunas o todas, pero, hay que reconocer el tipo de acción que van a realizar para emplearlas de manera adecuada.

# GUIA PRACTICA # 13

## TEMA: USO DE PICTUREBOX E IMAGE:

Uno de los aspectos más agradables que puede presentar cualquier lenguaje visual es la capacidad de utilizar gráficos(sean dibujos o fotos) en él, cambiarles de tamaño, moverlos, hacerlos desaparecer o cambiarlos y otras propiedades más.

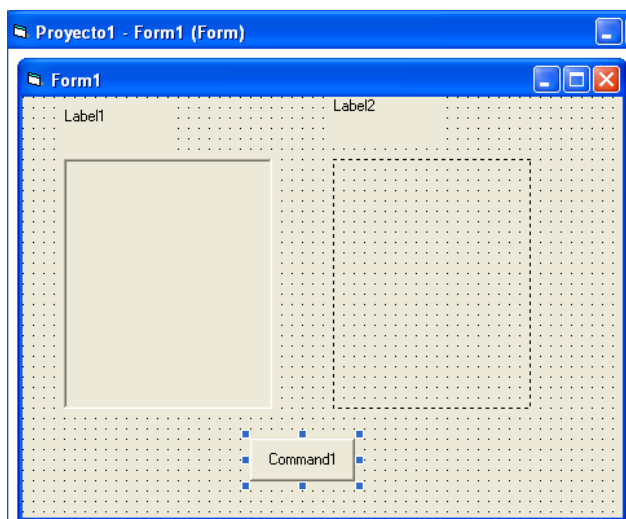
¿Qué es un PictureBox e Image?.- pues son dos de los controles más utilizados para desplegar gráficos en Visual Basic, aunque éste último (Image) utiliza menos recursos del sistema que el primero, pero Picture Box, tiene más propiedades lo que le hace más versátil para el trabajo. Todo depende de la utilidad que le quiera dar el programador o cual será su función dentro del formulario.



Ud. Puede poner cualquier gráfico o foto escaneada en estos dos objetos siempre y cuando tenga las extensiones: BMP, JPG, GIF, DIB, EMF, CUR, ICO, WMF.

### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Label
4. Ubique 1 Picture Box
5. Ubique 1 Image
6. Ubique 1 Command Button
7. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



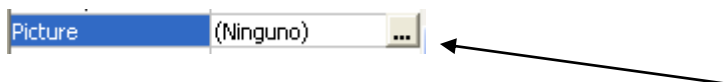
Ahora vamos a cambiar las propiedades de los Objetos en base a la siguiente descripción:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Gráficos en Visual Basic
Label1	Caption:Label1	Caption: PictureBox

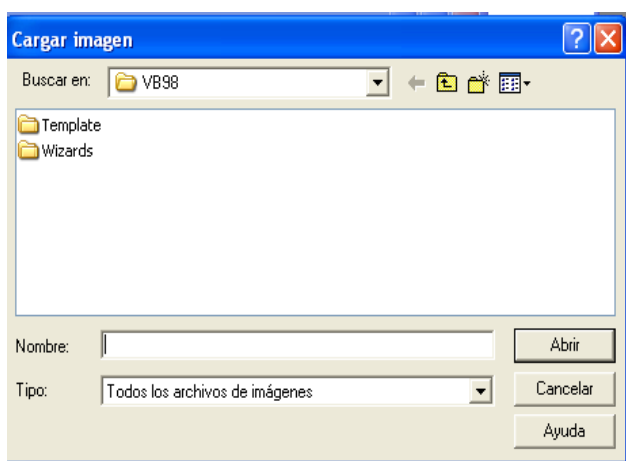


Label2	Caption: Label2	Caption: Image
Command1	Caption: Command1 Nombre: Command1	Caption: &Finalizar Nombre: fin


Para el Objeto **PICTURE BOX**, una vez hecho click en este objeto, diríjase a la propiedad picture y encontrará este resultado:



Observe la palabra ninguno y al extremo derecho tres puntos, significa el acceso al explorador, haga click ahí y obtendrá un resultado similar al siguiente:




Es la ventana de Cargar imagen, que es igual a las ventanas de windows o de office, observe la opción Buscar en.... Es el lugar en donde se encuentra la imagen (carpeta o directorio), Nombre y tipo de imagen. Si ud., tiene experiencia en windows esto no será problema. Si sabe donde se encuentran imágenes en el computador o desea poner una a su gusto y conoce su localización, búsquela y colóquela

En este lugar no existen imágenes, entonces vamos a proceder a buscar una. Por lo tanto haga click en  Observará otra ventana, allí haga doble click en **COMMON**, doble

click en **GRAPHICS**, doble click en **ICONS**, doble click en **COMPUTER**, busque **PC01** y haga doble click. entonces que regresará al formulario con la imagen (ícono) cargado ya, diríjase a la propiedad **Autosize** y cambié por true, observe el resultado.



El tamaño del PictureBox se ajusto automáticamente al tamaño del ícono

Ahora haga click en el objeto **Image**, diríjase a la ventana de propiedades y busque Picture, aparecerá lo mismo que ya vimos antes, de click en los tres puntos y aparecerá una ventana activa con la última carpeta abierta, salga de allí haciendo click en  hasta que regrese a **GRAPHICS**, allí busque **METAFILE** y haga doble click, Haga también doble click en **BUSSINESS**, Busque **PCOMPUTR**, haga doble click, entonces regresará al formulario con la imagen cargada ya, pero....sorpresa, la imagen es demasiado grande, busque la propiedad **stretch** y active **True**, en ese momento se ajustará la imagen al tamaño que ud., le de. Cambie de tamaño y vea el resultado.

Ejecute su formulario poniendo primero el código para el botón de finalizar y observe que será similar al siguiente:




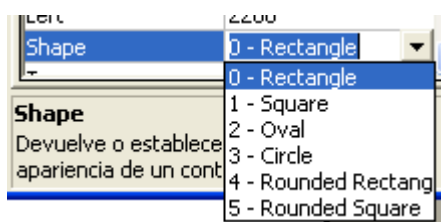
La propiedad Picture está presente en la mayoría de los objetos de Visual Basic, y su trabajo es igual, eso implica que podemos poner imágenes en todo lugar que nos de esta posibilidad.

También existe la posibilidad de cambiar de gráfico pero en tiempo de ejecución con el uso de la función **LoadPicture** y señalando la ruta en la que se encuentra la imagen a cargar, así por ejemplo:

```
imagen.Picture = LoadPicture("C:\windows\g4.jpg")
```

Tenga cuidado cuando la ruta no es clara por ejemplo cuando las carpetas tienen espacios en blanco en su nombre, puede producirle problemas, entonces es preferible manejar una carpeta especial para los gráficos que se quiera utilizar.

El Objeto **Shape**  (Figura) permite ubicar figuras en el formulario, haga doble click allí y se dirigirá automáticamente a la propiedad Shape, en donde tenemos las siguientes opciones



Permite según la opción trazar un rectángulo, cuadrado, ovalo, círculo, entre otros, en el objeto seleccionado, por ejemplo si se escoge circle dibujará un círculo en el objeto y ud. Puede cambiarle el tamaño a su gusto.

### **TRABAJO PARA EL ALUMNO:**

- Cree un nuevo proyecto(Formulario)
- Ubique un Label, que llevará el título Gráficos
- Ubique un Image llamado imagen y la propiedad Stretch en true
- Ubique 4 botones de comando, llamados dibujo 1, dibujo 2, dibujo 3, Finalizar
- Al ejecutarse el formulario el objeto image debe tener una imagen, pero al hacer click en los botones debe cambiarse esa imagen por otra, el formulario terminará al hacer click en el botón finalizar.

Puede hacer lo mismo utilizando Botones de Opción.

## GUIA PRACTICA # 14

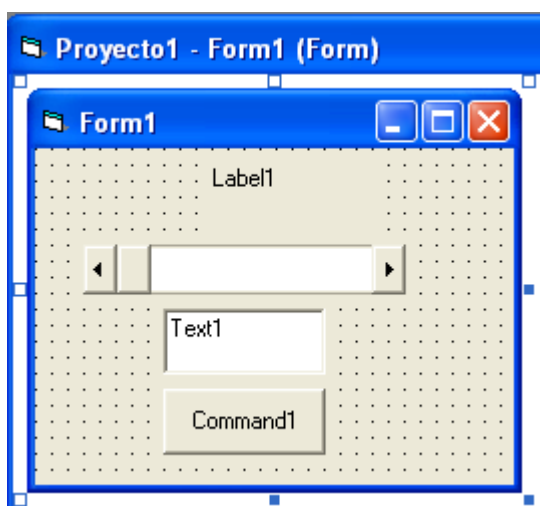
### TEMA: BARRAS DE DESPLAZAMIENTO :

Si ya ha utilizado windows anteriormente sabrá que las barras de desplazamiento son una característica básica de su trabajo, por ejemplo, al buscar las propiedades de un objeto, tiene que utilizar la barra de desplazamiento, para de entre muchas opciones escoger una.

En Visual Basic existe una horizontal(**HscrollBar**)l y una vertical(**VscrollBar**)

#### PRACTICA 14.1 HscrollBar:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 1 Label
4. Ubique 1 TextBox
5. Ubique 1 HscrollBar
6. Ubique 1 Command Button
7. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



Ahora vamos a cambiar las propiedades:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Barra de desplazamiento
Label1	Caption: Label1	Caption: Múltiplos de 3
Text1	Texto: Text1 Nombre: Text1	Text: Vacío borre todo Nombre: a
Command1	Caption:Command1	Caption: &Finalizar
HScroll1	Nombre: Hscroll1 Max: 32767 LargeChange: 1	Nombre: Multiplo Max: 1000 LargeChange: 20

Max representa el valor máximo que va a tener la barra de desplazamiento, LargeChange es el rango de cambios que se dará cuando haga click en el avance de la

barra, en ocasiones hay que aclarar otras propiedades como min (que es el mínimo valor, es decir del que comenzará) en 1 y de igual manera SmallChange en 1 (el cambio mínimo que se debe realizar).

Luego de los cambios realizados nuestro formulario debe quedar similar al siguiente:

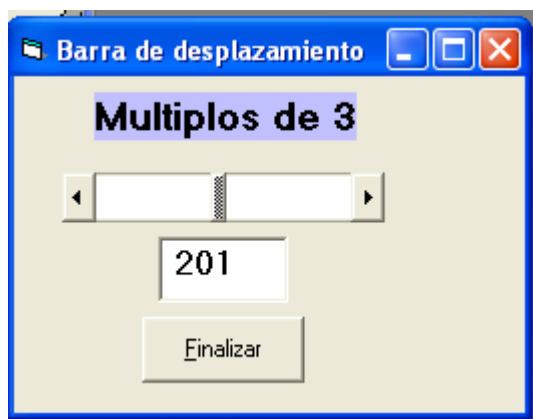


Ahora ubicaremos el código necesario, haga doble click en la barra de desplazamiento (HScroll) y escriba:

```
Private Sub multiplo_Change()
    If multiplo.Value Mod 3 = 0 Then
        a.Text = Str(multiplo.Value)
    End If
End Sub
```

En la decisión se toma el valor de la barra de desplazamiento(MULTIPLLO.VALUE) y se divide (MOD) para 3, si el resultado es 0 (valor exacto es múltiplo), entonces se desplegará en la caja de texto como un carácter por la transformación STR(MULTIPLLO.VALUE) a string o carácter.

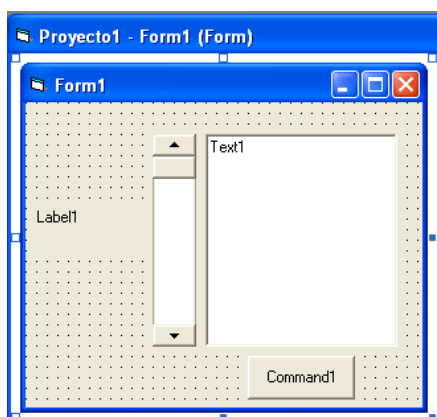
Ejecute su formulario y haga click repetidamente en los extremos de la barra de desplazamiento y vea el resultado parecido al siguiente:



Lo mismo se puede trabajar con la barra de desplazamiento vertical, sin embargo, haremos un ejemplo:

**PRACTICA 14.2 VscrollBar:**

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 1 Label
4. Ubique 1 TextBox
5. Ubique 1 VscrollBar
6. Ubique 1 Command Button
7. Mueva los objetos en todo el formulario hasta obtener la siguiente forma:



Cambiamos las propiedades basándonos en la descripción aquí presentada:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Barra de desplazamiento
Label1	Caption: Label1	Caption: Tamaño
Text1	Texto: Text1 Nombre: text1	Text: Hola Nombre: a
Command1	Caption: Command1	Caption: &Finalizar
HScroll1	Nombre: Vscroll1 Max: 32767 Min: 0 LargeChange: 1	Nombre: tamaño Max: 72 Min: 8 LargeChange: 5

Luego de haber cambiado las propiedades el formulario debe presentarse similar al siguiente:



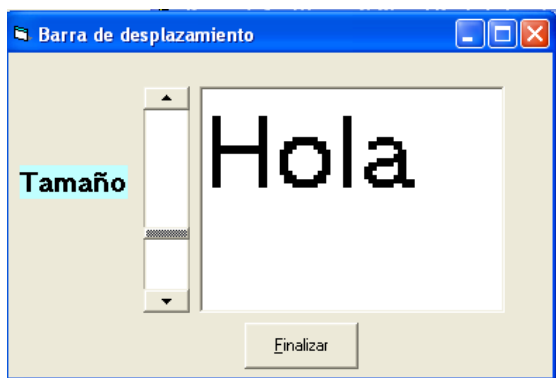
Ahora, adicionemos el código respectivo, haga doble click en la barra de desplazamiento (Vscroll1) y escriba lo siguiente:

```
Private Sub tamaño_Change()  
    a.Font.Size = tamaño.Value  
End Sub
```

Permitirá cambiar en la caja de texto a, el tamaño de la letra(función Font.Size) que sea igual al tamaño asignado o activo en la barra de desplazamiento.(Tamano.value).

El botón de finalizar nuevamente ud., lo puede hacer sin inconvenientes

Al final, al ejecutar el formulario debe tener una apariencia similar a la siguiente:



### **TRABAJO PARA EL ALUMNO:**

- Convierta las temperaturas de centígrados a fahrenheit utilizando barras de desplazamiento.
- Convierta las temperaturas de fahrenheit a centígrados utilizando barras de desplazamiento.
-

## GUIA PRACTICA # 15

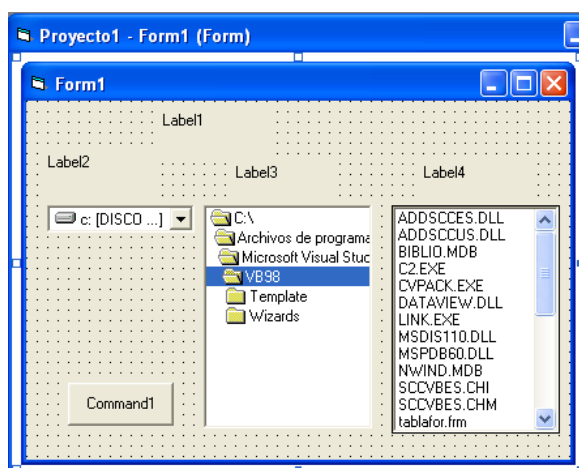
### TEMA: CONTROLES DE LISTA DE UNIDAD - DIRECTORIO Y ARCHIVO :

Muchas veces en windows podemos utilizar el explorador para cambiar de unidad, ver las carpetas existentes, ingresar a una de ellas y ver también su contenido, eso se logra en Visual Basic con los objetos: DriveListBox(Cuadro de lista de drives), DirListBox(Cuadro de lista de directorios o carpetas) y FileListBox(Cuadro de lista de archivos).

El primero, permite obtener una lista de las unidades (drives) existentes en el computador, el segundo lo mismo pero con directorios o carpetas y el tercero con los archivos.

### PRACTICA 15:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 4 Label
4. Ubique 1 DriveListBox
5. Ubique 1 DirListBox
6. Ubique 1 FileListBox
7. Ubique 1 Command Button
8. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:



Observe que al ubicar DriveListBox, DirListBox y FileListBox aparece automáticamente con algún contenido, y es que esa es la ubicación por defecto que nos muestra visual basic, pero, no se preocupe lo podemos cambiar para que sea útil a sus necesidades.

Veamos ahora las propiedades, que deben tener a parte de las ya conocidas:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: MI EXPLORADOR
Label1	Caption: Label1	Caption: MI EXPLORADOR
Label2	Caption: Label2	Caption: Unidad
Label3	Caption: Label3	Caption: Carpeta o directorio

Label4	Caption: Label4	Caption: Archivos
DriveListBox	Nombre: Drive1	Nombre.unidad
DirListBox	Nombre: Dir1	Nombre.directorio
FileListBox	Nombre: File1 MultiSelect: 0-none	Nombre.Archivos MultiSelect: 2-Extended
Command1	Caption. Command1 Nombre: Command1	Caption. &Finalizar Nombre: fin



Una vez cambiadas las propiedades el formulario debe presentar un aspecto similar al del gráfico presentado aquí.

Recuerde que, el contenido de la unidad, directorio y Archivos aparece automáticamente y es por lo que está activo en ese momento en el computador.

Ahora vamos a poner el código correspondiente en los objetos necesarios, así, haga doble click en DriveListBox(Unidad) y escriba:

```
Private Sub Unidad_Change()  
    Directorio.Path = Unidad.Drive  
End Sub
```

Permite seleccionar la unidad (origen) que tendrá DirListBox(directorio) que será igual al Drive seleccionado en la Unidad, es decir, si se escoje una nueva unidad automáticamente las carpetas o directorios cambiarán.

Ahora haga doble click en DirListBox(Directorio) y escriba:

```
Private Sub Directorio_Change()  
    Archivos.Path = Directorio.Path  
End Sub
```

Es lo mismo que el anterior, es decir, si se cambia de directorio, aparecerán en FileListBox(Archivos), los archivos existentes en él.

Solo basta codificar el objeto finalizar y listo, ejecute su formulario y tendrá una respuesta similar a la siguiente:





Haga click en unidad y escoja la que crea conveniente y haga doble click, de igual manera explore las carpetas o directorios y por ende obtendrá los archivos existentes en él.

En el cuadro de archivos salen todos los existentes y es que en la ventana propiedades se encuentra activado **PATTERN** como **\*.\***, pero, si ud., quiere trabajar con ciertos archivos especiales como los de gráficos por ejemplo solo bastará ubicar en PATTERN. La extensión necesaria así: \*.bmp;\*.ico;\*.wmf;\*.gif;\*.jpg, entre otros, pruébelo para que vea la diferencia.

También puede trabajar con diskettes, Cds, DVDs, entre otros, es decir con cada una de las unidades presentes en el computador, con las carpetas, directorio y archivos a gusto y necesidad del programador o usuario.

### **TRABAJO PARA EL ALUMNO:**

- Permita que el formulario busque todos los programas de pascal, fox, c, java, etc
- Permita también que el formulario encuentre solo y exclusivamente los documentos de Word, Excel y Power Point.
- Ubique un diskette y realice la exploración a su contenido poniendo la propiedad Pattern en \*.\*.
- Lo mismo haga en un CD.
- Practique con cada una de las unidades de su computador y con los directorios o carpetas existentes en él

## GUIA PRACTICA # 16

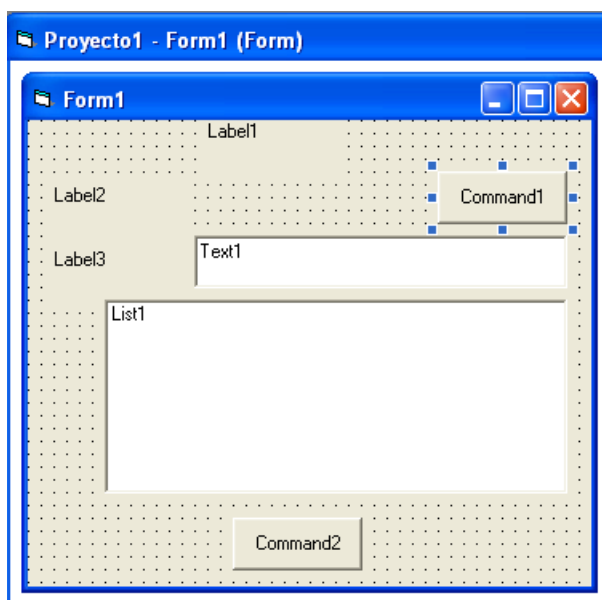
### TEMA: CUADROS DE LISTA (LIST BOX):

Un cuadro de lista es un control que presenta una lista de elementos en la que el usuario puede seleccionar uno o más de dichos elementos.

Por ejemplo, en un sistema de matriculación de alumnos, se puede ingresar nuevos alumnos y adicionarlos a una lista.

### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 3 Label
4. Ubique 1 TextBox
5. Ubique 1 ListBox
6. Ubique 2 Command Button
7. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:

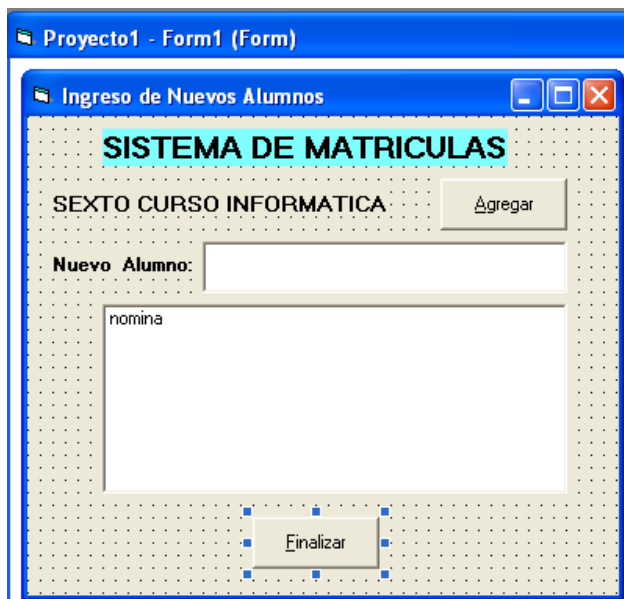


Ahora cambiemos las propiedades de los elementos necesarios, en base a la siguiente tabla y a lo aprendido anteriormente:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Ingreso de Nuevos alumnos
Label1	Caption: Label1	Caption: SISTEMA DE MATRICULAS
Label2	Caption: Label2	Caption: SEXTO CURSO INFORMATICA
Label3	Caption: Label3	Caption: Ing. Nombres nuevo alumno:
Text1	Text: Text1 Nombre: Text1	Text. Vacio (Borre todo) Nombre: Alumno
List1	Nombre: List1	Nombre: Nomina
Command1	Caption: Command1 Nombre: Command1	Caption: &Agregar Nombre: Agregar

Command2	Caption: Command2 Nombre: Command2	Caption: &Finalizar Nombre: fin
----------	---------------------------------------	------------------------------------

Una vez cambiadas las propiedades el formulario debe presentar un aspecto similar al del gráfico presentado aquí

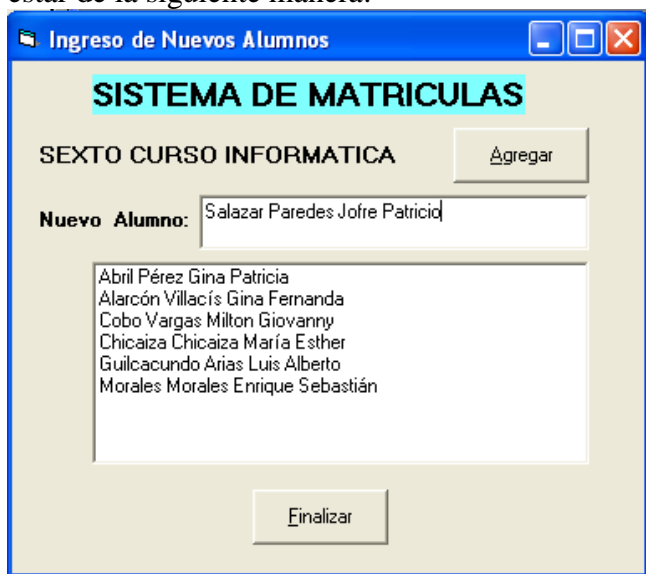


Ahora, vamos a adicionar el código respectivo a los botones necesarios, haga doble click en Agregar y escriba:

```
Private Sub agregar_Click()
    nomina.AddItem Alumno
    Alumno = " "
    Alumno.SetFocus
End Sub
```

nomina.AddItem Alumno, Permite adicionar un item o texto (Additem) del texto ingresado en Alumno al cuadro de lista llamado nómina, luego, limpia el TextBox (Alumno=" ") y envía el cursor al TextBox (Alumno.SetFocus).

El botón de finalizar ya lo puede realizar usted solo. Ejecute el formulario que debe estar de la siguiente manera:



Escriba un nombre, haga click en agregar y observe como dicho nombre pasa al Cuadro de Lista y cada vez que se ingresa un nombre y se hace click en Agregar van poniendo uno tras otro (los nombres), si no alcanza la lista inmediata y automáticamente se ubicará una barra de desplazamiento.

**TRABAJO PARA EL ALUMNO:**

- Realice un formulario similar al presente pero para un consultorio médico en el cual los pacientes según sigan llegando se pongan en una lista como esperando turno de atención.
- Lo mismo puede hacer para una lista de sus amigos, profesores, gustos, música, entre otros ejemplos que usted puede realizar.

## GUIA PRACTICA # 17

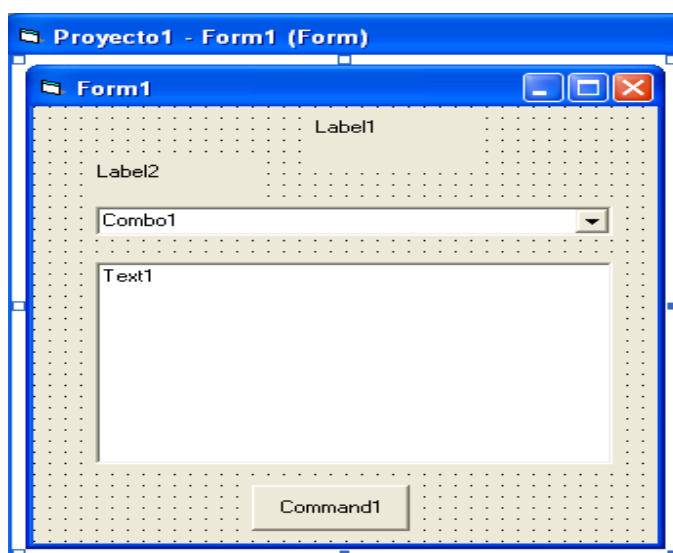
### TEMA: CUADROS COMBINADOS (COMBO BOX):

Un cuadro de lista es una mezcla de un cuadro de texto con una lista. En él no sólo se podrá ingresar un dato sino también seleccionarlo de la lista.

Por ejemplo: si se deseara conocer que cátedras dicta un determinado maestro, al localizar el nombre de un profesor debe salir automáticamente las materias y los cursos.

### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Label
4. Ubique 1 ComboBox
5. Ubique 1 TextBox
6. Ubique 1 Command Button
7. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:



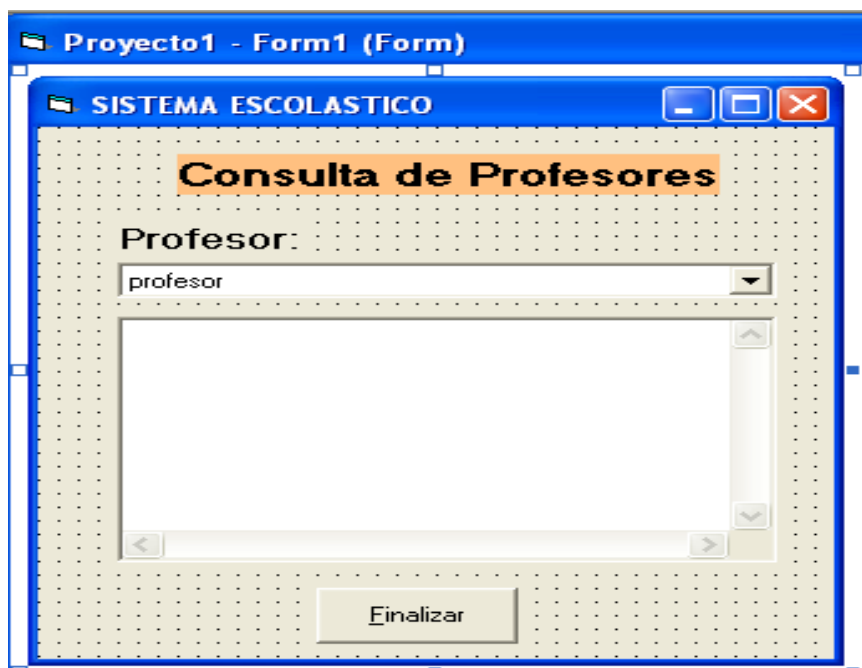
Ahora, pongamos las propiedades en base al siguiente cuadro y a lo que hemos tratado en anteriores capítulos:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: SISTEMA ESCOLASTICO
Label1	Caption: Label1	Caption: Consulta de Profesores
Label2	Caption: Label2	Caption: Profesor
Combo1	Nombre: Combo1 Text: Combo1 Style: 0-Dropdown Combo	Nombre: Profesor Text: vacío – Borre todo Style: 2-Dropdown List
Text1	Nombre: Text1 Text: Text1 ScrollBars: 0-None Multiline: False	Nombre: Materias Text: vacío – Borre todo ScrollBars: 3-Both Multiline: True

Command1	Caption: Command1 Nombre: Command1	Caption. &Finalizar Nombre: fin
----------	---------------------------------------	------------------------------------

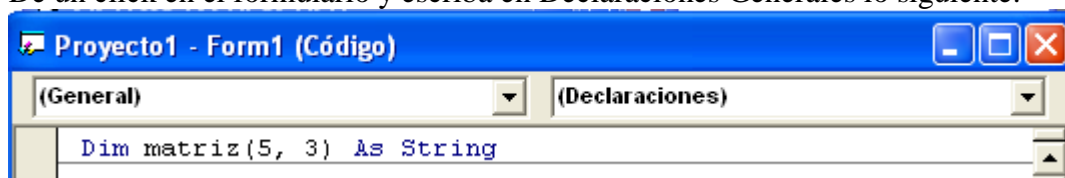
Importante: En text1 se ha ubicado una propiedad nueva llamada ScrollBars con: 3-Both es decir que van a aparecer las barras de desplazamiento tanto horizontal como vertical al ejecutarse el formulario y funcionarán si es que el texto excede al tamaño de la caja de texto.

Una vez cambiadas las propiedades, el formulario debe tener la siguiente apariencia:



Ahora vamos a escribir el código respectivo

De un click en el formulario y escriba en Declaraciones Generales lo siguiente:



Estamos dimensionando una matriz(arreglo bidimensional) cuya variable se llama matriz de 5 filas por 3 columnas y que contendrá valores de tipo carácter (String).

Luego haga doble click en el formulario, note que se abre el evento cargar el formulario (Form\_load) y escriba lo siguiente:

```
Private Sub Form_Load()
    profesor.AddItem "Ing. Giovanni Cobo Vargas"
    profesor.AddItem "Ing. Fabricio valencia"
    profesor.AddItem "Ing. Jofre Salazar"
    profesor.AddItem "Ing. Enrique Morales"
    profesor.AddItem "Lic. Darwin Céspedes"
```

```
matriz(0, 0) = "Programación I"
matriz(1, 0) = "Programación II"
matriz(2, 0) = "Programación III"
matriz(3, 0) = "Programación Orientada a Objetos I"
matriz(4, 0) = "Programación Orientada a Objetos II"
matriz(5, 0) = "Programación VISUAL"
```

```
matriz(0, 1) = "PRIMERO"
matriz(1, 1) = "SEGUNDO"
matriz(2, 1) = "TERCERO"
matriz(3, 1) = "CUARTO"
matriz(4, 1) = "QUINTO"
matriz(5, 1) = "SEXTO"
```

```
matriz(0, 2) = "ING. SISTEMAS"
matriz(1, 2) = "ING. SISTEMAS"
matriz(2, 2) = "ING. SISTEMAS"
matriz(3, 2) = "ING. SISTEMAS"
matriz(4, 2) = "ING. SISTEMAS"
matriz(5, 2) = "ING. SISTEMAS"
```

End Sub

### **Explicación:**

- En el ComboBox (Profesor) se adicionan los ítems (5) con diferentes nombres.
- En matriz fila de 0 a 5 columna 0 se ubica las materias dictadas
- En matriz fila de 0 a 5 columna 1 se ubica los cursos correspondientes
- En matriz fila de 0 a 5 columna 2 se ubica la carrera o especialidad correspondiente

Otra vez, si usted no tiene buenos conocimientos de lo que son arreglos mono y bidimensionales comúnmente conocidos como vectores y matrices en programación le sugerimos que trate de encontrar una guía suficiente para que ud., lo pueda comprender de mejor manera, temática que la puede encontrar en los libros de cursos anteriores que provee el colegio.

Posteriormente, haga doble click en el objeto profesor (ComboBox) y busque el evento click (Observe que le puede salir el evento Change, no se preocupe cámbielo o busque el evento click ) y escriba:

```
Private Sub profesor_Click()
```

```
    Dim PROF As String
```

```
    Dim MAT As String
```

```
    Dim CUR As String
```

```
    Dim ESP As String
```

```
    MAT = "Materia : " & matriz(profesor.ListIndex, 0)
```

```
    CUR = "Nivel  : " & matriz(profesor.ListIndex, 1)
```

```
    ESP = "Carrera : " & matriz(profesor.ListIndex, 2)
```

```
materias = MAT & vbCrLf & CUR & vbCrLf & ESP
```

```
End Sub
```

### Explicación:

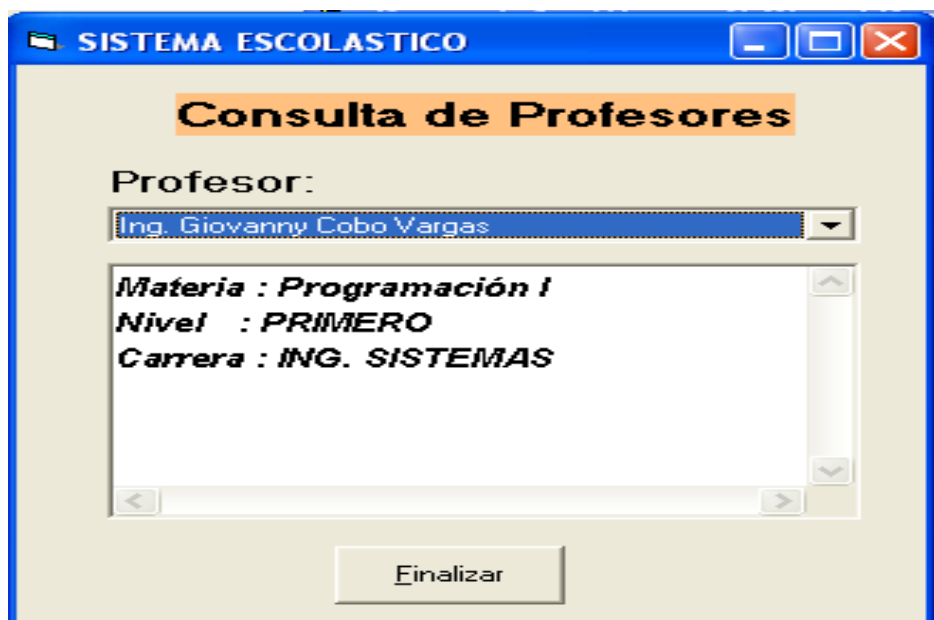
- Se dimensiona(define) las variables PROF, MAT, CUR, ESP de tipo character
- Se especifica el origen de los datos por ejemplo: para MAT vendrá de la Matriz llamada matriz según la posición del profesor se toma el índice de la lista de profesores, según la columna especificada), o sea, si se ha hecho click en el primer item (Ing. Giovanni Cobo Vargas) el ListIndex valdrá 1, entonces:

```
MAT = "Materia : " & matriz(profesor.ListIndex, 0)
      Texto      Y matriz(1,1,0)
```

Si se hace click en el primer item saldrá el mensaje "Materia:" seguido del valor de la matriz(1,1,0), o sea, Programación I, PRIMERO, ING. SISTEMAS.

- Por último, la caja de texto materias, desplegará esos contenidos asignados anteriormente (materias = MAT & vbCrLf & CUR & vbCrLf & ESP), vbCrLf es igual a poner un enter para que cada mensaje salgan en línea diferente.

Ponga código al Botón finalizar y ya está listo el formulario, ejecútelo y deberá tener un formulario similar al siguiente:



### TRABAJO PARA EL ALUMNO:

- Realice un formulario utilizando un Combo Box y matrices para un hospital en el cual se pueda hacer un click en el nombre de un médico y salga la información de su especialidad, dirección domiciliaria y número telefónico.
- Otros formularios a criterios del alumno y/o profesor

## GUIA PRACTICA # 18

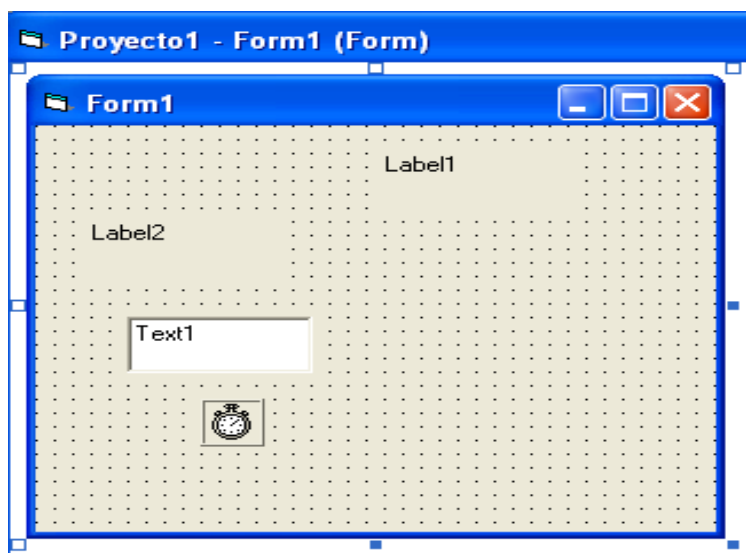
### TEMA: CONTROL DE TIEMPO – CRONOMETRO – RELOJ (TIMER):

Un timer conocido como Cronometro o reloj permite controlar el tiempo transcurrido para una acción determinada realizada por un objeto o evento, este se controla desde el inicio hasta su finalización asignada por el usuario en tiempo de diseño o en tiempo de ejecución.

Por ejemplo: si quisiéramos simular el funcionamiento de un reloj digital como el que aparece en la ventana de tareas de windows sin utilizar ningún bucle o contador.

### PRACTICA:

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Label
4. Ubique 1 Timer
5. Ubique 1 TextBox
6. Ubique 1 Command Button
7. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:



Cambiamos las propiedades según la siguiente descripción:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: Uso de Timer
Label1	Caption: Label1	Caption: RELOJ DIGITAL
Label2	Caption: Label2	Caption: Hora:
Text1	Nombre:Text1 Text : text1	Nombre:Hora Text: Vacío – Borre todo
Timer1	Interval: 0	Interval: 500

Ubique usted un botón de finalizar que ya lo puede hacer en base a lo aprendido.



Una vez que ya se haya cambiado las propiedades, el formulario debe tener la siguiente apariencia (En tiempo de diseño) :



Ahora, ubicaremos el código correspondiente, Haga doble click en el formulario y busque el evento Activate y escriba:

```
Private Sub Form_Activate()  
    hora = Time()  
End Sub
```

Asigna la función Time() que da la hora del sistema a la variable hora (Caja de texto), pero es estática, es decir, no se mueve, se queda fija, puede usted ejecutar el formulario para que vea el resultado, ahora asignaremos la función del timer, haga doble click en Timer1 y escriba:

```
Private Sub Timer1_Timer()  
    If hora <> Time() Then  
        hora = Time()  
    End If  
End Sub
```



**Explicación:** Activa el timer comparando si el valor que se encuentra en la caja de texto(hora) es diferente de la hora actual del computador, entonces asigna automáticamente la nueva hora a la caja de texto. Ejecute su formulario y observe que debe tener apariencia similar a la siguiente:

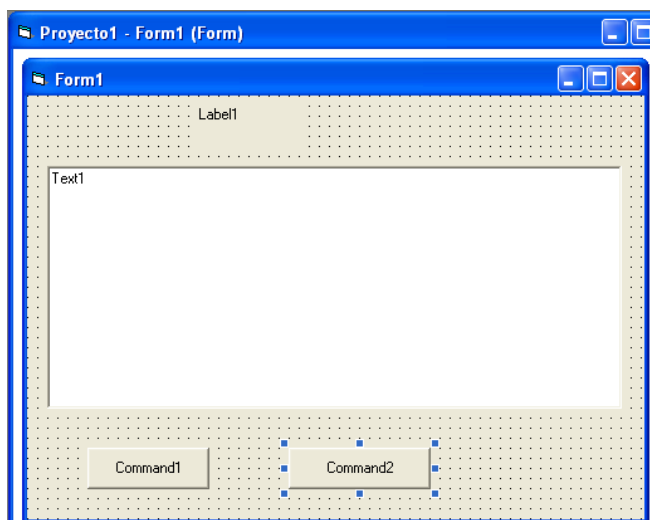
## GUIA PRACTICA # 19

### TEMA: IMPRIMIR COMO EN UN EDITOR DE TEXTOS:

Creo que la mayoría de personas al ver por lo menos un computador sabrá que se está trabajando como en una máquina de escribir tradicional solo que la hoja está en la pantalla y cuando ya no se tenga nada que corregir se imprimirá, pues, eso es justamente lo que vamos a hacer en esta práctica, a presentar el principio de un editor de textos que se pueda imprimir.

### PRACTICA:

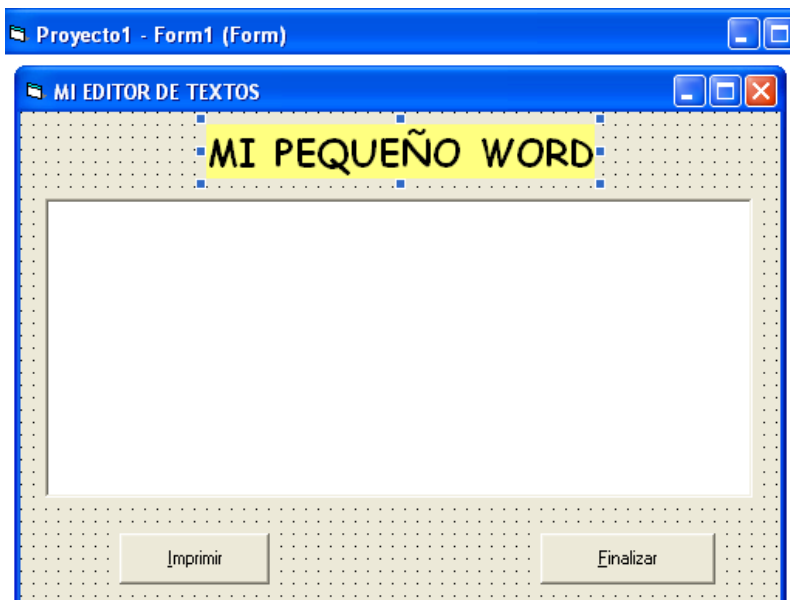
1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 1 Label
4. Ubique 1 TextBox
5. Ubique 2 Command Button
6. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:



A continuación vamos a establecer las propiedades (Recuerde propiedades que ya deben ser puestas a su gusto en base a lo aprendido en capítulos anteriores).

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption: Form1	Caption: MI EDITOR DE TEXTOS
Label1	Caption: Label1	Caption: MI PEQUEÑO WORD
Text1	Text:Text1 Nombre:Text1 Multiline:False	Text. Vacío borre todo Nombre: texto Multiline:True
Command1	Caption:Command1 Nombre:Command1	Caption:&Imprimir Nombre:imprimir
Command2	Caption:Command2 Nombre:Command2	Caption:&Finalizar Nombre:fin

Hasta ahora el formulario debe quedarse de la siguiente manera.



Adicionemos el código correspondiente para nuestro formulario. Haga doble click en el botón imprimir y escriba:

```
Private Sub imprimir_Click()
```

```
    Dim Ancho, Alto
```

```
    Ancho = Printer.TextWidth(texto) / 8
```

```
    Alto = Printer.TextHeight(texto) / 1
```

```
    Printer.CurrentX = Printer.ScaleWidth / 6 - Ancho
```

```
    Printer.CurrentY = Printer.ScaleHeight / 6 - Alto
```

```
    Printer.Print texto
```

```
    Printer.EndDoc
```

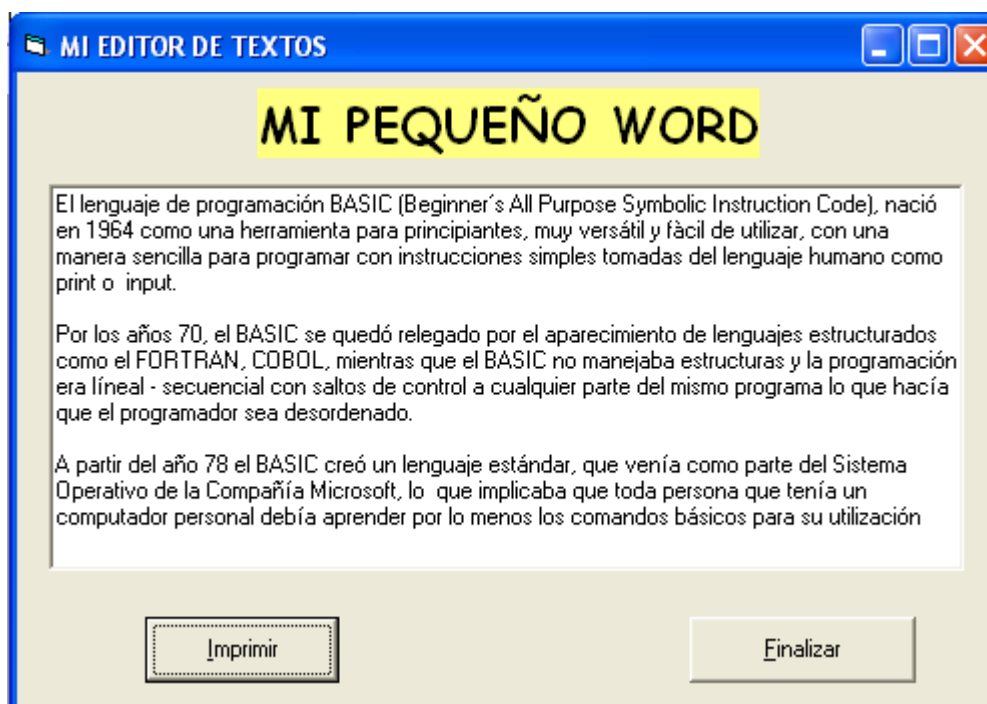
```
End Sub
```

Se dimensiona como variables Alto y Ancho, luego se les da valores a través de la opción Printer.TextWidth(texto), es decir el ancho del texto y en la siguiente línea el largo o extensión.

Printer.CurrentX = Printer.ScaleWidth y Printer.CurrentY = Printer.ScaleHeight, permiten imprimir el texto en base al ancho y alto, los números permiten establecer ubicaciones para la impresión, pruebe cambiandoles de valor y vea el resultado.

Por último, La función Printer.Print texto, permite enviar la información a la impresora, y Printer.EndDoc, finaliza tal impresión y el control regresa a la pantalla.

Programo el botón de finalizar y ejecute su formulario, que seguramente sera similar al siguiente(note que, ya se ha escrito una gran cantidad de palabras en su interior):



La impresión que se obtendrá de este documento no será perfecta puesto que aún nos faltan una serie de controles para que lo hagan profesional, como por ejemplo, negritas, centrados, subrayados, tamaños y tipos de letras, colores y más, pero, con este ejemplo nos basta para tener una idea de lo que es la creación de un editor de textos o procesador de palabras.

En un principio al trabajar con Word por ejemplo, se nos viene a la idea que debe ser “una monstruosidad” hacer algo así, pero, como ve usted, al trabajar con lenguajes visuales no se hace tan complicado, solo se necesita conocimiento y práctica que le invitamos a realizar.

### **TRABAJO PARA EL ALUMNO:**

- En el mismo formulario anterior, ubique un nuevo botón de comando que le permita realizar un nuevo texto.
- Cambie de colores al fondo del texto y póngale color también a la letra
- Cambie también el tamaño de la letra y el tipo.

## GUIA PRACTICA # 20

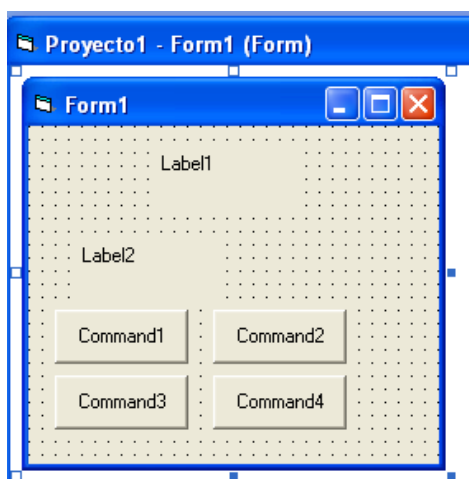
### TEMA: MENU DE OPCIONES:

Hasta ahora hemos realizado formularios simples, pero, todos ellos podemos enlazarlos en uno solo, a esto se le llama menú de opciones, es decir, haremos un formulario principal que enlace o “Llame” a otros formularios, que se ejecutarán y regresarán al formulario principal.

Por ejemplo: haremos un formulario principal al que llamaremos menú (sin tilde), y allí mismo haremos otros tres formularios llamados suma, imágenes y explorador, cada uno de ellos hará su trabajo propio, el menú tendrá cuatro botones que enlazarán a los otros formularios y en cada formulario habrá a más de lo necesario un botón de regresar que le permitirá volver al menú. Entonces, manos a la obra:

### PRACTICA :

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 2 Label
4. Ubique 4 Command Button
5. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:



Pongamos las propiedades de los objetos, según la descripción a continuación:

OBJETO	PROPIEDAD ORIGINAL	PROPIEDAD NUEVA
Form1	Caption:Form1 Nombre:Form1	Caption: Menú de opciones Nombre: menú
Label1	Caption: Label1	Caption: Menú de opciones
Label2	Caption: Label2	Caption: Haga click en la opción deseada
Command1	Caption:Command1 Nombre: Command1	Caption. &Sumatoria Nombre. sumatoria
Command2	Caption:Command2 Nombre: Command2	Caption. &Imagenes Nombre. imagen
Command3	Caption:Command3 Nombre: Command3	Caption. &Explorar Nombre. explorar

Command4	Caption:Command4 Nombre: Command4	Caption. &Finalizar Nombre. fin
----------	--------------------------------------	------------------------------------

Después de haber cambiado las propiedades, observe el formulario que debe tener la siguiente apariencia:



Haga doble click en Sumatoria y ponga el siguiente código:

```
Private Sub sumatoria_Click()
    suma.Show
End Sub
```

Suma.Show, permite activar el formulario llamado suma y mostrarlo en la pantalla, para activar este botón en tiempo de ejecución, el formulario debe existir sino, nos dará un error.

Haga lo mismo con los siguientes botones(haciendo doble click en ellos), así:

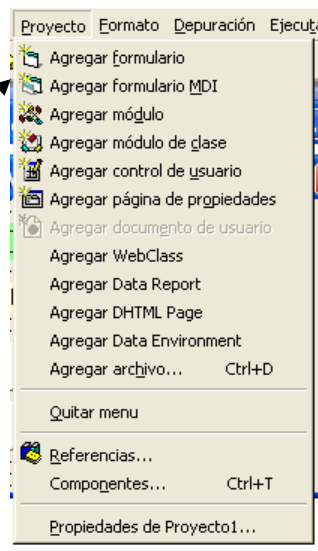
```
Private Sub explorar_Click()
    explorador.Show
End Sub
```

```
Private Sub imagenes_Click()
    fotos.Show
End Sub
```

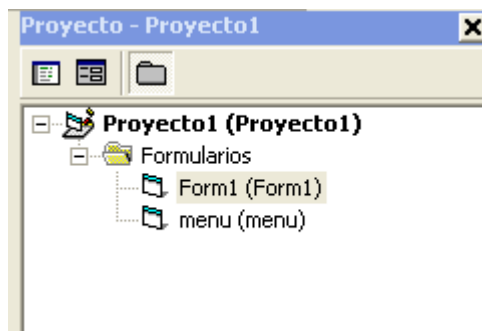
Ahora diríjase a la **barra de menús** y haga click en **proyecto**, obtendrá la siguiente ventana:

Allí haga click en Agregar formulario

Y observe una ventana similar a la primera que expone Visual Basic, allí escoja...Formulario y observe como se ubica un nuevo formulario encima del anterior.

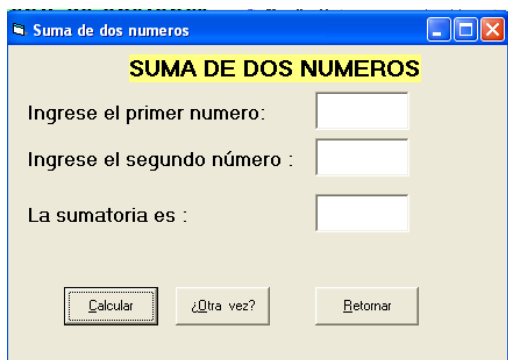


Observe también la ventana de proyectos, y encontrará dos formularios dentro del proyecto el uno llamado menu, porque ya lo definimos anteriormente y el nuevo llamado Form1.

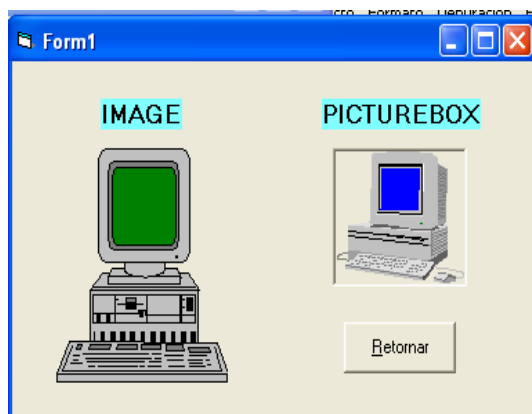


Al formulario nuevo cámbiele de nombre y póngale, suma (en vez de Form1), observe el cambio.

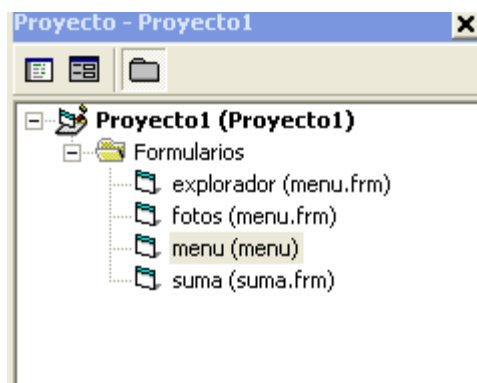
Ahora diseñe el formulario de la suma de dos números visto anteriormente



luego adicione otro formulario con el nombre fotos y otro llamado explorador, de ma misma manera como creamos el anterior y diseñe los formularios correspondientes:



De tal manera que en la ventana de proyectos se vea de la siguiente manera:



Observe que no existe el botón de finalizar, sino se a ubicado el mismo botón pero con el título **RETORNAR**.

Allí se debe programar lo siguiente, por ejemplo en el formulario de la suma:

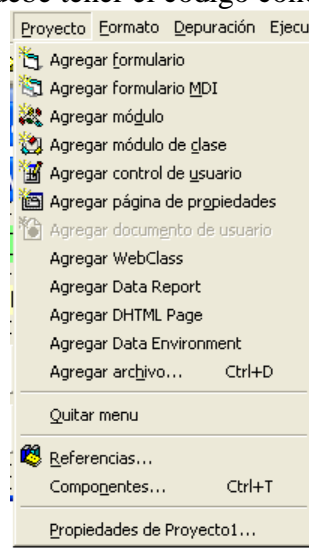
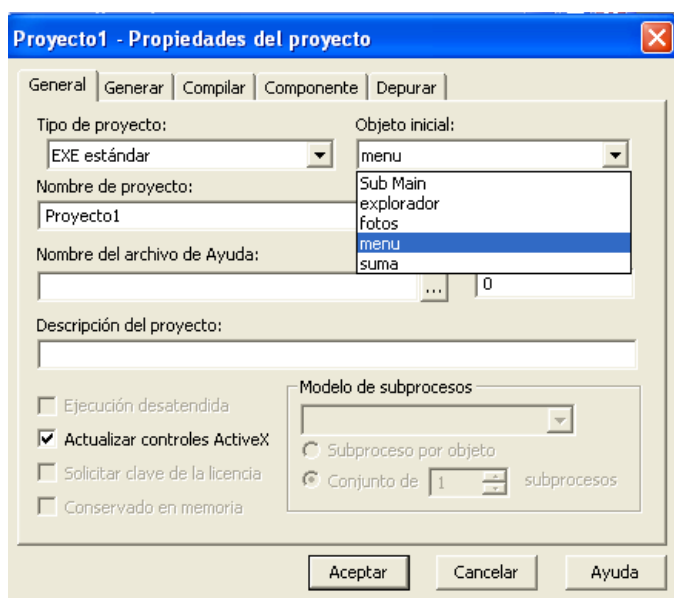
```
Private Sub retornar_Click()
    suma.Hide
    menu.Show
End Sub
```

Suma.hide, permite ocultar el formulario llamado suma y al mismo tiempo activa el formulario llamado menú.

Lo mismo se debe hacer con el resto de formularios activos para cerrarlos y poder regresar al menú principal, en donde, el botón de finalizar debe tener el código conocido para terminar la ejecución del formulario principal.

**NOTA IMPORTANTE:** Cuando se realiza varios formularios en un solo proyecto siempre se queda activo el primer formulario abierto, es decir, cuando ud., desee ejecutarlos siempre se activará el primero que se creo, para evitar esto tenemos que cambiar las propiedades de ejecución que está ubicado en la barra de menús opción proyecto, elija propiedades del proyecto

allí obtendrá una ventana similar a la siguiente:



← Esta ventana proporciona algunas utilidades, lo que nos interesa por el momento es la opción objeto inicial, de un click en el triángulo invertido y aparecerán los nombres de los formularios existentes, elija el que ud., desea y haga click en aceptar.

De esta manera se puede seleccionar el formulario que se activara inicialmente al ejecutar el proyecto.

Obviamente cuando ya estén correctamente funcionando los formularios debe poner el formulario menú, como objeto inicial.



## GUIA PRACTICA # 21

### TEMA: CONTROL DATA - MANEJO DE BASES DE DATOS:

Visual Basic es un potente manejador de bases de datos, es decir, se puede enlazar bases de datos a los formularios y editar, modificar, borrar, grabar registros y además trabajar con otras aplicaciones.

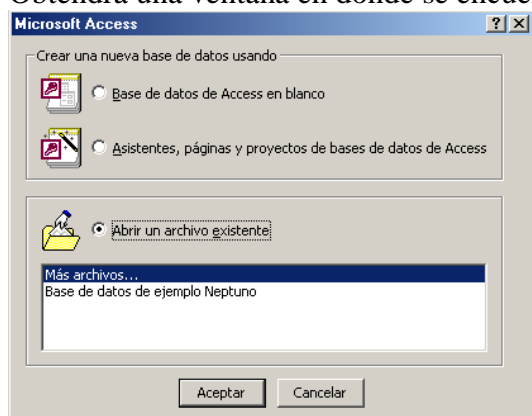
Visual Basic, puede enlazarse con bases de datos de diferentes proveedores tales como Access, Foxpro, Dbase, Paradox, excel, Oracle o de cualquier ODBC.

Para esto haremos una introducción básica a Access que es un gestor y motor de Bases de datos del paquete Office de la misma compañía Microsoft proveedor de Visual Basic.

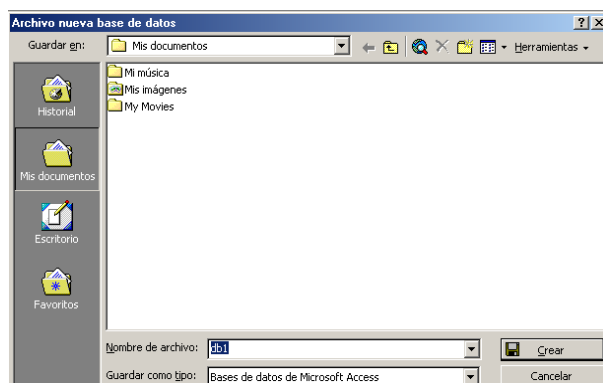
### Crear bases de datos en ACCESS.

En el escritorio de Windows haga click en Inicio, luego busque programas y ubique Microsoft Access y haga click allí.

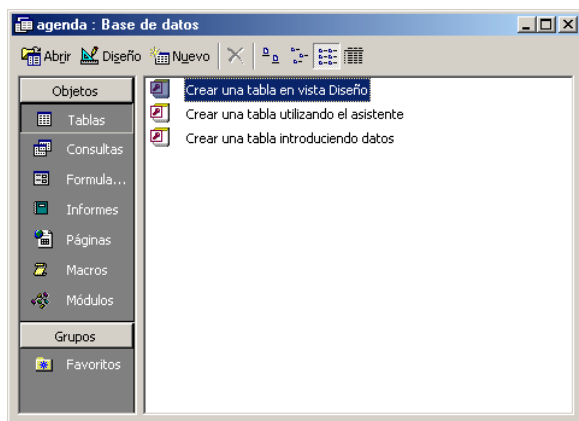
Obtendrá una ventana en donde se encuentran tres opciones:



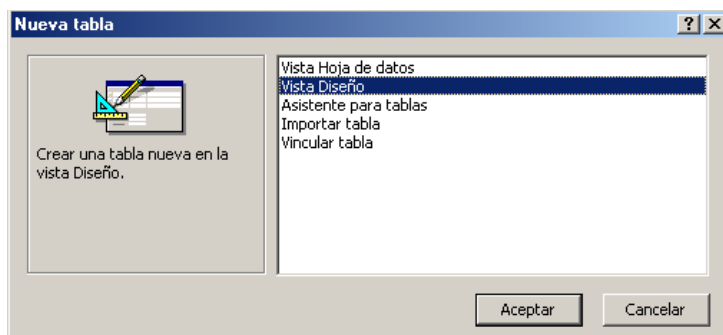
Elija Base de datos de Access en blanco y haga click en aceptar.



Obtiene una ventana en donde se especifica que se guardará en mis la carpeta mis documentos, ubique un nombre adecuado para la base de datos, en este caso digite AGENDA, al hacer click en guardar tendrá otra ventana:



Es la ventana de trabajo de Access, allí se puede crear tablas, formularios, módulos, consultas, entre otras opciones, lo que nos interesa en este momento es lo que se encuentra activo. TABLA, observe que tiene tres opciones y por defecto está activado **Crear una tabla en vista diseño**, haga click en **Nuevo**. Entonces obtendrá la siguiente ventana:



Debemos escoger la opción **Vista diseño**, haga click allí y luego en Aceptar, con lo que obtendremos la siguiente ventana:

Tabla1 : Tabla			
	Nombre del campo	Tipo de datos	
▶			

Es justamente el indicativo de **Vista diseño**. Aquí diseñaremos la tabla que va a tener la estructura de los registros, si usted ya ha visto sistemas en otros lenguajes como en FOX, comprenderá de mejor manera estos conceptos, caso contrario le sugerimos referirse al libro de FOXPRO de esta misma colección y que lo provee el colegio.

En **Nombre del campo**, especificaremos el nombre que llevará la estructura del campo, por ejemplo. Código, Nombres, Dirección, teléfono, entre otros; en **tipo de datos**, determinaremos que tipo de valores se podrán ingresar, así. Numérico, carácter, fecha, etc.

Para ejemplo ubicaremos la siguiente estructura:

Nombre del campo	Tipo de datos
Cedula	Texto
	Texto
	Memo
	Numérico
	Fecha/Hora
	Moneda
	Autonumérico
	Sí/No
	Objeto OLE
	Hipervínculo
	Asistente para búsquedas...

Observe que en **nombre del campo** se ha ubicado: cedula (sin tildes ni espacios en blanco), al poner enter el cursor pasa automáticamente a la columna **Tipo de datos**, y se ubica un triángulo invertido en la parte derecha, en donde podemos hacer click y escoger el tipo de campo deseado, fíjese en el ejemplo y escriba los campos como a continuación se presentan:

Nombre del campo	Tipo de datos
Cedula	Texto
Nombres	Texto
direccion	Texto
telefono	Texto
edad	Texto
estado civil	Texto

Para este ejemplo solo se han escogido campos de tipo textual, los demás campos ahora se pueden pasar por alto, pero esto no implica que si ud., desea que algún campo sea numérico, puede hacerlo.

Al terminar de ubicar los campos, haga click en el icono de guardar(Diskette) o cierre la ventana (X), o Archivo....Guardar y obtendrá la siguiente ventana:

**Guardar como** [?] [X]

Nombre de la tabla:

Tabla1

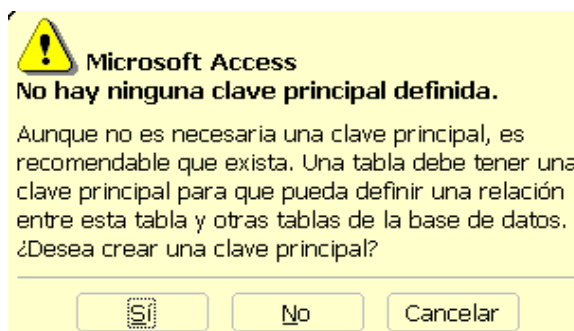
Aceptar

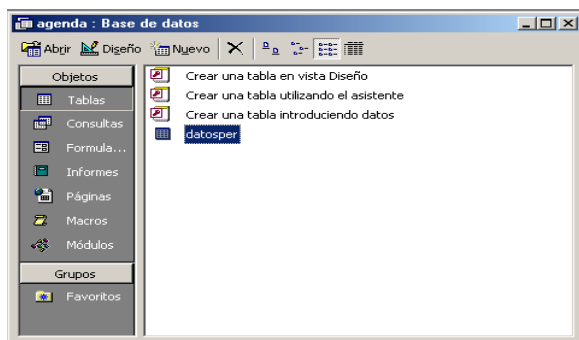
Cancelar

Es el nombre que se le dará a la tabla (pues puede tener varias tablas en la misma base de datos), allí escriba **datosper** y haga click en aceptar.

Observe que el asistente le mostrará un mensaje recordatorio similar al siguiente:

No lo haga caso, por ahora no es de nuestro interés, solo haga click en la opción NO. Y regresará a la ventana inicial en donde podrá ya observar la tabla que hemos creado, así:





Ahora procedamos a llenar de valores esos campos, recuerde que lo que hemos hecho hasta ahora es crear la estructura, por eso vamos a llenar de valores, haga doble click en la tabla **datosper**, y se aparecerá una ventana en la que deberá usted escribir, como por ejemplo:

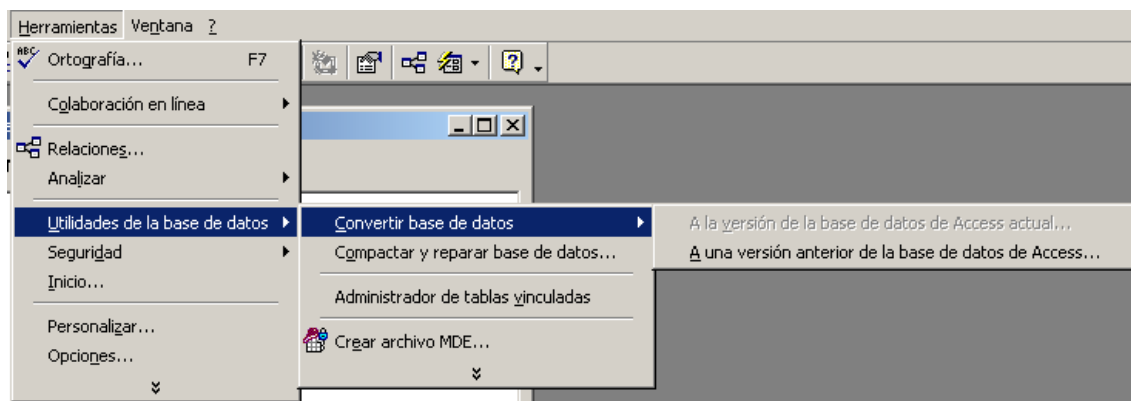
datosper : Tabla						
	Cedula	Nombres	direccion	telefono	edad	estado civil
	1800000000	YOLA CHANG	QUIZ QUIZ	8263523	23	VIUDA
	1811111111	AQUILES CHAI	LA MERCED	099765432	19	CURA
	1822222222	BENITO CAME	LA PRADERA	414243	25	DIVORCIADO
*						

Recuerde que este es un ejemplo, puede usted escribir los valores que desee. No importa si lo que esta escribiendo no alcanza, habrá un desplazamiento automático.

Llene los datos que crea necesario pueden ser como en el ejemplo 3, 4, 5, o más, eso lo dejamos a su gusto.

Por último, vamos a proceder a grabar la base de datos en una versión anterior de access para resguardar nuestro trabajo con seguridad.

Haga click en **herramientas** (Barra de menús), busque la opción **Utilidades de la base de datos**, **Convertir base de datos** y **A una versión anterior de la base de datos de Access** y de click.



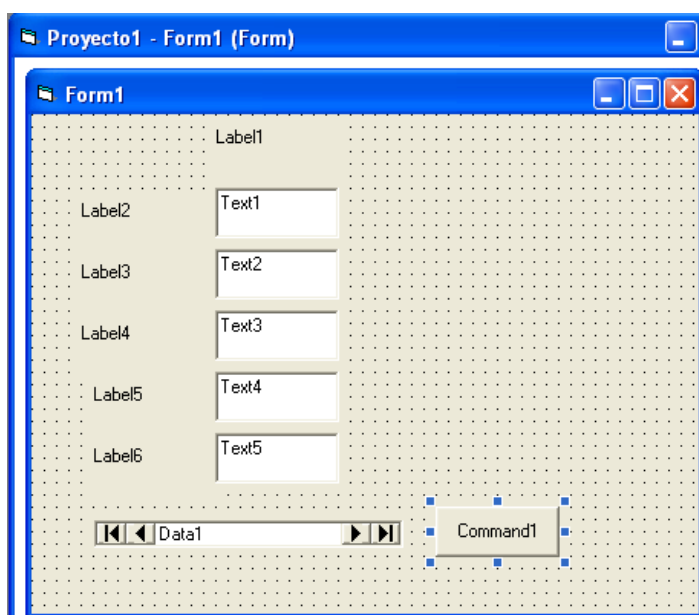
Obtendrá una ventana en la que deberá poner otro nombre para la base de datos versión anterior, en este caso escriba **Agen\_ant**, y haga click en aceptar. Si pone el mismo nombre tendrá un mensaje de error.

Ahora tiene dos bases de datos una en una versión anterior (Agen\_ant) y otra en versión actual(Agenda),

Hemos terminado el trabajo en access por favor ciérrelo y vuelva a windows.

**PRACTICA:**

1. Acceda a Visual Basic
2. Cree un nuevo PROYECTO (FORMULARIO)
3. Ubique 6 Label
4. Ubique 5 Textbox
5. Ubique 1 Data
6. Mueva y dimensione los objetos en todo el formulario hasta obtener la siguiente forma:



Ubiquemos las propiedades correspondientes, en base al siguiente cuadro y a lo aprendido anteriormente:

<b><u>OBJETO</u></b>	<b><u>PROPIEDAD ORIGINAL</u></b>	<b><u>PROPIEDAD NUEVA</u></b>
Form1	Caption: Form1	Caption: Trabajo con Base de datos
Label1	Caption: Label1	Caption: Agenda personal - telefónica
Label2	Captión: Label2 Nombre: Label2	Caption: Cédula Nombre: cedula
Label3	Captión: Label3 Nombre: Label3	Caption: Nombres Nombre: nombres
Label4	Captión: Label4 Nombre: Label4	Caption: Dirección Nombre: dirección
Label5	Captión: Label5 Nombre: Label5	Caption: Teléfono Nombre: telefono

Label6	Captión: Label6 Nombre: Label6	Caption: Estado civil Nombre: estadocivil (sin espacios)
Data	Caption: Data1 DatabaseName: Connect: RecordSource:	Caption: Registros DatabaseName: C:\Mis documentos\agen_ant.mdb Connect: Access RecordSource: Datosper
Text1	Text:Text1 Name: Text1 DataSource: DataField:	Text:Vacio (Borre todo) Name: Cedula Datasource: Data1 DataField: Cedula
Text2	Text:Text2 Name: Text2 DataSource: DataField:	Text:Vacio (Borre todo) Name: Nombres Datasource: Data1 DataField: Nombres
Text3	Text:Text3 Name: Text3 DataSource: DataField:	Text:Vacio (Borre todo) Name: Direccion Datasource: Data1 DataField: Direccion
Text4	Text:Text4 Name: Text4 DataSource: DataField:	Text:Vacio (Borre todo) Name: Telefono Datasource: Data1 DataField: Telefono
Text5	Text:Text5 Name: Text5 DataSource: DataField:	Text:Vacio (Borre todo) Name: Estadocivil Datasource: Data1 DataField: Estado civil
Command1	Caption:Command1 Nombre: Command1	Caption: &Finalizar Nombre: fin

### **Aclaración.**

En el control data ubicamos estas propiedades a más de las conocidas:

DatabaseName: C:\Mis documentos\agen\_ant.mdb

Es el origen de la base de datos sobre la cual se va a trabajar, observe que le saldrá una ventana de explorador para encontrar la base de datos que generalmente está en la carpeta de mis documentos

Connect: Access

Significa que se conectará a una base de datos cuyo origen es Access, aunque puede conectarse a otras como ya se vio anteriormente

RecordSource: Datosper

El origen de los registros está ubicado en la tabla Datosper, que ud., deberá escoger.

Fíjese que al ponerse en estas propiedades se tendrá un triángulo invertido al cual se puede acceder para escoger opciones o ubicaciones.

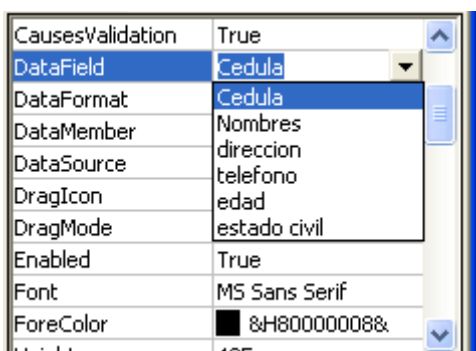
Por otro lado en los TextBox, aparecen opciones diferentes tales como:

Datasource: Data1

Destaca el origen de los datos que en este caso es el control data llamado Data1

DataField: Cedula

Aquí se manifiesta el origen del campo, en este caso Cedula, cuando se ubica en esta propiedad le saldrá un triángulo invertido que al dar click allí obtendrá los campos posibles a utilizar, así:



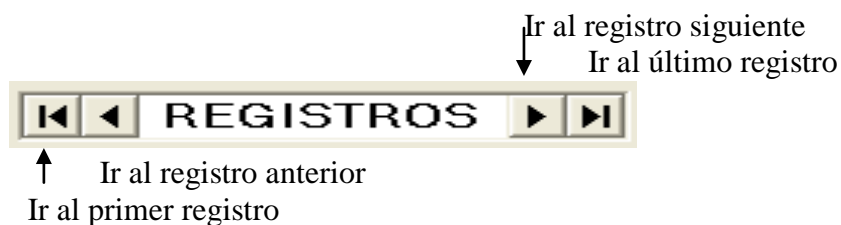
Dependiendo del TextBox escoja el que le corresponda, al final el formulario debe tener la siguiente apariencia.



En este formulario no vamos a adicionar código sino solo al botón de Finalizar, puesto que lo demás saldrá automáticamente. Ejecútelos para que vea el resultado que será similar al siguiente:



Haga click en las barras de desplazamiento de registros para pasar de uno a otro, El control data tiene las siguientes botones en su contexto:



Al dar click en cada una de esas opciones verá como los registros se van desplazando en su formulario.

**TRABAJO PARA EL ALUMNO:**

- Diríjase nuevamente a Access y haga el ingreso de más registros.
- Cree una base de datos en Access llamada Clientes, con los siguientes campos.
  - Código                    Numérico
  - Apellidos                Texto
  - Nombres                 Texto
  - Dirección                Texto
  - Telefono                 Numérico
- Haga un formulario en Visual Basic que enlace esa base de datos y permita trabajar con los registros existentes en él que deben ser mínimo 10.



## ANEXO

### CONTROLES (OBJETOS), PROPIEDADES Y EVENTOS GENERALMENTE MAS UTILIZADOS,

#### Etiqueta

Una etiqueta es un control que permite presentar texto de solo lectura. Se utiliza para mostrar títulos y mensajes que por lo general no cambian. Sin embargo, si lo desea, puede escribir procedimientos que cambien el texto mostrado por una etiqueta como respuesta a eventos en tiempo de ejecución.

Propiedad	Descripción
Alignment	Especifica la forma como se alineará el texto.
AutoSize	Especifica si se cambia automáticamente el tamaño del control para que quepan sus contenidos.
BackColor, ForeColor	Especifica el color de fondo y de primer plano empleado para mostrar el texto.
BackStyle	Especifica si el fondo del control es transparente u opaco.
Caption	Especifica el texto que se muestra en el control.
WordWrap	Especifica si la etiqueta se extiende a más de una línea cuando el texto es amplio.

#### Cuadro de texto

Un cuadro de texto es un control en el que el usuario pueda ingresar datos desde el teclado o visualizar un resultado generado por la aplicación. El valor de un cuadro de texto por lo general se asocia a una variable de memoria.

Propiedad	Descripción
Enabled	Habilita o deshabilita el control, es decir si se puede o no modificar el contenido.
Locked	Permite bloquear el acceso al control en tiempo de ejecución.
MaxLength	Especifica la longitud máxima (en caracteres) que puede escribirse en el control.
MultiLine	Especifica si el control admitirá múltiples líneas.
PasswordChar	Determina si se muestran en el control los caracteres escritos por el usuario o los caracteres de marcador de posición.
ScrollBars	Especifica si se muestran las barras de desplazamiento horizontales o verticales.
SelLength	Especifica el número de caracteres seleccionados.
SelStart	Especifica el inicio del texto seleccionado. Si su valor es 0, el texto seleccionado comienza por el primer carácter.
SelText	Contiene el texto seleccionado.
Text	Contiene el texto introducido en el control. Puede inicializar el contenido en tiempo de diseño.

Los eventos asociados a un cuadro de texto son los siguientes:

Evento	Descripción
Change	Ocurre cuando se cambia el contenido del control utilizando el teclado o el ratón.
KeyPress	Ocurre cuando se pulsa una tecla.

### Botón de comando

Un botón de comando es un control que suele emplearse para ejecutar una acción, como cerrar un formulario, imprimir un informe, etc.

Propiedad	Descripción
Cancel	Especifica si el control responderá a la tecla ESC, por lo general para terminar la aplicación.
Caption	Muestra un título para el control.
Default	Especifica si el control responderá a la tecla ENTRAR, sin importar la ubicación del enfoque.
Enabled	Permite habilitar o deshabilitar el control.
Picture	Especifica una imagen tipo icono para el control.
Style	Especifica la apariencia del control, ya sea estándar (estilo estándar de Windows) o gráfica (con una imagen personalizada).

Los botones de comando suelen responder al evento Click, cuya descripción es:

Evento	Descripción
Click	Ocurre cuando el usuario hace click sobre el control.

### Marco

Un marco es un objeto contenedor que puede utilizarse para agrupar casillas de verificación, botones de opción, botones de comando, etc.

Propiedad	Descripción
Caption	Muestra un título para el control.

### Casilla de verificación

Una casilla de verificación se utiliza para alternar entre dos posibles valores (Activado/Desactivado). Cada casilla de verificación es independiente de las demás ya que cada una tiene su propio nombre (Name). Puede utilizar casillas de verificación en grupos para mostrar múltiples opciones entre las cuales el usuario puede seleccionar todas las que desee a la vez.

Propiedad	Descripción
Caption	Muestra un título para el control.
Value	Devuelve o establece el estado del control, puede ser: Desactivado (0), Activado (1).

Las casillas de verificación suelen responder al evento Click.

### Botón de opción

Un botón de opción se utiliza para alternar entre dos posibles valores (Activado/Desactivado), de manera similar a las casillas de verificación, la diferencia radica en que de un grupo de botones de opción el usuario sólo puede seleccionar un único botón a la vez.

Propiedad	Descripción
Caption	Muestra un título para el control.
Value	Devuelve o establece el estado del control, puede ser: Desactivado (Falso), Activado (True).

Los botones de opción también suelen responder al evento Click.

### Cuadro de lista

Un cuadro de lista es un control que presenta una lista de elementos en la que el usuario puede seleccionar uno o más de dichos elementos. Si el número de elementos supera el número que puede mostrarse, se agregará automáticamente una barra de desplazamiento al control.

Propiedad	Descripción
List	Contiene los elementos de la lista. Puede definirlos en tiempo de diseño.
ListCount	Especifica el número de elementos de la lista. Su valor siempre es uno más que el mayor valor de ListIndex.
ListIndex	Especifica el número de orden que ocupa el elemento seleccionado en la lista. Toma el valor de 0 para el primer elemento, si no ha seleccionado ninguno retorna -1.
MultiSelect	Especifica si se pueden seleccionar varios elementos de la lista.
SelCount	Devuelve el número de elementos seleccionados.
Selected	Especifica si un elemento de la lista está seleccionado.
Sorted	Especifica si los elementos de la lista se ordenan alfabéticamente.
Text	Especifica el valor del elemento seleccionado en ese instante. El valor es tratado como una cadena de caracteres.

Los métodos que se pueden utilizar para un cuadro de lista son los siguientes:

Método	Descripción
AddItem	Agrega un nuevo elemento a la lista.
Clear	Elimina todos los elementos contenidos en la lista.
RemoveItem	Elimina un elemento de la lista.

### Cuadro combinado

Un cuadro combinado es una mezcla de un cuadro de texto con una lista. En él no sólo se podrá ingresar un dato sino también seleccionarlo de la lista. Presenta las mismas propiedades, eventos y métodos que el cuadro de lista a excepción de lo siguiente:

Propiedad	Descripción
Style	Especifica si el control se comporta como un cuadro de texto o como una lista desplegable. Un cuadro de texto permite ingresar valores. Una lista desplegable sólo permite seleccionar valores. El valor predeterminado es 0-vbComboDropDown, e incluye una lista desplegable y un cuadro de texto. El valor 1-vbComboSimple incluye tan solo cuadro de texto. El valor 2-Dropdown List incluye sólo una lista desplegable.

### Cuadro de imagen

Un cuadro de imagen se utiliza para mostrar un archivo de imagen (BMP, GIF o JPG). Recorta el gráfico si el control no es lo bastante grande para mostrar la imagen completa.

Propiedad	Descripción
AutoSize	Especifica si el control ajusta automáticamente su tamaño para mostrar un gráfico completo.
Picture	Especifica el archivo de imagen para el control.

El evento que suele manejarse para este control es MouseMove, cuya descripción es:

Evento	Descripción
MouseMove	Ocurre cuando el usuario mueve el ratón sobre el control.

### Imagen

Un control imagen se utilizará para mostrar un archivo de imagen (BMP, GIF o JPG) de manera similar a un Cuadro de imagen. Sin embargo, este control utiliza menos recursos del sistema y se actualiza con más rapidez que un cuadro de imagen.

Propiedad	Descripción
Picture	Especifica el archivo de imagen para el control.
Stretch	Especifica como se ajusta el tamaño del archivo gráfico para que quepa dentro del control imagen.

Para este control también se suele manejar el evento MouseMove.

### Barras de desplazamiento horizontal

Una barra de desplazamiento horizontal representa un valor entero, la cual tiene un cuadrado que se desplaza a lo largo de la misma para fijar un valor.

La posición más a la izquierda se corresponde con el valor mínimo, la posición más a la derecha con el valor máximo, y cualquier otra posición es un valor entre ellos dos.

Propiedad	Descripción
Max	Especifica el valor máximo que puede tomar la propiedad Value.
Min	Especifica el valor mínimo que puede tomar la propiedad Value.

LargeChange	Especifica en cuanto cambiará el valor de la propiedad Value cuando el usuario haga click antes o después del cuadrado de desplazamiento.
SmallChange	Especifica en cuanto cambiará el valor de la propiedad Value cuando el usuario haga click en alguna de las flechas de los extremos de la barra.
Value	Especifica el valor del control, este se encuentra siempre entre los valores de las propiedades Min y Max.

Las barras de desplazamientos responden a los eventos:

Evento	Descripción
Change	Ocurre inmediatamente después que el cuadrado de desplazamiento ha sido movido.
Scroll	Ocurre mientras el cuadrado de desplazamiento está siendo movido (este evento sólo ocurre cuando el cuadrado es arrastrado).

### Barras de desplazamiento vertical

Una barra de desplazamiento vertical representa un valor entero, la cual tiene un cuadrado que se desplaza a lo largo de la misma para fijar un valor.

La posición más hacia arriba se corresponde con el valor mínimo, la posición más hacia abajo se corresponde con el valor máximo, y cualquier otra posición es un valor entre ellos dos.

Las propiedades y eventos de las barras de desplazamiento horizontal y vertical son análogas.

### Cuadro de lista de unidades

Un cuadro de lista de unidades permite al usuario seleccionar una unidad de disco válida en tiempo de ejecución. Utilice este control para presentar una lista de todas las unidades válidas del sistema.

Propiedad	Descripción
Drive	Especifica la unidad seleccionada en tiempo de ejecución.

Este control responde al evento Change, cuya descripción es la siguiente:

Evento	Descripción
Change	Ocurre cuando el usuario hace click sobre la unidad que desea seleccionar. El evento Change también ocurre si se selecciona la unidad mediante el teclado.

### Cuadro de lista de directorios

Un cuadro de lista de directorios muestra directorios y rutas de acceso en tiempo de ejecución. Utilice este control para mostrar una lista jerárquica de directorios.

Propiedad	Descripción
Path	Especifica la ruta de acceso actual.

Un cuadro de lista de directorios también responde al evento Change.

### Cuadro de lista de archivos

Un cuadro de lista de archivos encuentra y muestra los archivos del directorio especificado por la propiedad Path en tiempo de ejecución. Utilice este control para mostrar una lista de los archivos seleccionados por tipo.

Propiedad	Descripción
Path	Especifica la ruta de acceso actual.
Pattern	Especifica el tipo de archivo que se desea visualizar en el control.
List	Devuelve los elementos contenidos en la parte de lista del control.
ListIndex	Especifica el índice del elemento seleccionado actualmente en el control.

### LA FUNCIÓN MSGBOX

A menudo habrá situaciones en las cuales tendrá que presentar algún mensaje (por ejemplo una pregunta) y en función de la respuesta del usuario se ejecutará una determinada acción.



La función MsgBox() muestra un cuadro de diálogo predefinido y retorna un valor de tipo entero dependiendo del botón seleccionado por el usuario. Su sintaxis es de la siguiente forma:

VALOR = MsgBox(MENSAJE [, BOTÓN + ICONO, TÍTULO])

Donde MENSAJE se refiere a una expresión de cadena que se muestra como mensaje en el cuadro de diálogo.

BOTÓN es una expresión numérica que se refiere al tipo y cantidad de botones a utilizar.

ICONO es una expresión numérica que se refiere al estilo de icono que se va a utilizar.

TÍTULO es un texto que se mostrará en la barra de título de la caja de diálogo.

#### Tipo de botón

Valor	Botón	Nombre
0	Aceptar	vbOKOnly
1	Aceptar y Cancelar	vbOKCancel
2	Anular, Reintentar, Ignorar	vbAbortRetryIgnore
3	Sí, No y Cancelar	vbYesNoCancel

4	Sí y No	vbYesNo
5	Reintentar y Cancelar	vbRetryCancel

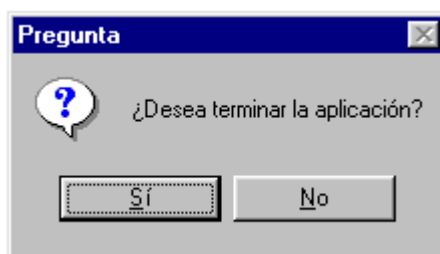
**Tipo de icono**

Valor	Icono	Nombre
16	Mensaje crítico	vbCritical
32	Signo de interrogación	vbQuestion
48	Signo de exclamación	vbExclamation
64	Signo de información	vbInformation

**Valores retornados**

Valor	Botón	Nombre
1	Aceptar	vbOK
2	Cancelar	vbCancel
3	Anular	vbAbort
4	Reintentar	vbRetry
5	Ignorar	vbIgnore
6	Sí	vbYes
7	No	vbNo

Como ejemplo vamos a construir un formulario que presente un botón “Salir”, de tal forma que cuando el usuario pulse dicho botón se presente el siguiente mensaje:



En caso de que el usuario elija la opción “Sí”, la aplicación debe terminar. En caso contrario, es decir si el usuario elige la opción “No”, se debe proseguir con la aplicación.

Para el desarrollo del presente ejemplo, proceda a ubicar sobre el formulario un botón de comando (CmdSalir), a continuación ingresar el siguiente código:

```
Private Sub CmdSalir_Click()
    Dim RESP As Integer
    RESP = MsgBox("Desea terminar la aplicación?", _
        vbQuestion + vbYesNo, "Pregunta")
    If RESP = vbYes Then
        End
    End If
End Sub
```

**FUENTE BIBLIOGRAFICA:**

- HEYMAN, Mark. ESSENTIAL VISUAL BASIC 4  
SAMS Publishing, 1996- Prentice Hall
- CASTILLO, Peralta. Ing. TUTORIAL DE VISUAL BASIC 5  
Autoedición – España 2000
- SERRANO, Jorge. Ing. VISUAL BASIC 4.0  
Autoedición – 1997  
Mvisual.com
- TIZNADO, Carlos. Ing. VISUAL BASIC – PASSWORD II  
McGrawHill – Colombia 1977
- VARIOS. LO MAS IMPORTANTE DE VISUAL BASIC 6  
Edit. Atlántida. Buenos Aires Argentina – 2002