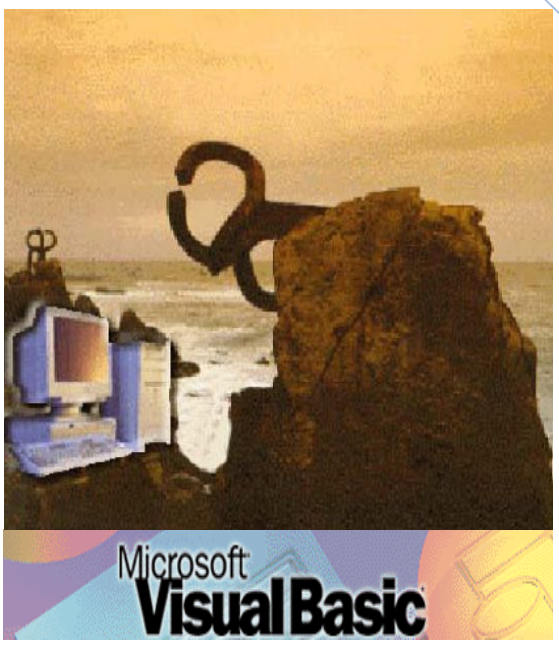


VISUAL BASIC 6.0

MANUAL DEL USUARIO

Visual Basic 6.0 es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores de PCs, tanto expertos como novatos. En el caso de los programadores expertos por la facilidad con la que desarrollan aplicaciones complejas en poquísimo tiempo (comparado con lo que cuesta programar en ***Visual C++***, por ejemplo).

30/11/2008

INDICE

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

UNIDAD I

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

UNIDAD VISUAL BASIC I PROGRAMACIÓN VISUAL ELEMENTOS BÁSICOS

1.- INTRODUCCIÓN VISUAL BASIC

Información y Conocimiento son los dos elementos claves del nuevo milenio, ninguna sociedad podrá alcanzar ni puede ignorar este nuevo esquema, ya las naciones no se miden por su riqueza industrial, ni sus activos físicos, ni por su poder militar, sino por la cantidad de información que produce y consume, así como por la recombinación de información nueva en un conocimiento de grado superior.

Nuevos sistemas de información, tienden a ser cada vez de mayor alcance y complejidad, sobre todo cuando se toman en cuenta las nuevas necesidades de información que demandan las nuevas organizaciones.

Nuevos sistemas de información son costosos en tiempos y recursos, la solución moderna de sistemas de información exigen herramientas y metodología que resuelvan, económica, eficiente y global problemas de información planteados por las organizaciones.

Además el pleno potencial del hardware no es aprovechado plenamente y existe un considerable retraso con el software y sus aplicaciones, generando lo que se conoce como “crisis del software”.

En programación tradicional, modular o estructurada un programa describe una serie de pasos a ser realizados para la solución de un problema, es decir es un algoritmo.

En programación orientada a objetos (OOP) un programa es considerado como un sistema de objetos interactuando entre sí, ambientes de desarrollo visuales facilitan aun más la construcción de programas y solución de problemas, porque permiten abstraer al ingeniero de software de todo el GUI (interfaces gráfica) del problema, que constituye más del 60% del código normal de un programa.

Es decir, en programación modular o estructurada un problema sencillo de información es descompuesto en una serie de módulos (llamados procedimientos o funciones) donde cada uno de ellos realiza una tarea específica, por ejemplo uno de ellos captura los datos, otro resuelve operaciones, etc.

En OOP todo problema aun aquellos sencillos de información, se consideran y resuelven como módulos de código gigante (clases) que contiene todo el código necesario (variables, procedimientos, funciones, interfaces, etc.) para solucionar el problema.

En programación visual (que también es heredera de OOP), la interfaces con el usuario (pantallas) son generadas por el propio compilador y el ingeniero de software solo se concentra en resolver el problema planteado.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Visual Basic es un compilador que permite usar cualquiera de los tres enfoques en la solución de problemas de información que puedan y deban ser resueltos empleando el computador y el lenguaje.

Para propósitos de aprendizaje usaremos el tercer enfoque, es decir programación en ambientes visuales y usando el lenguaje de programación Visual Basic

2.- MODELO DE SOLUCION VISUAL BASIC

En general un problema de información es posible entenderlo, analizarlo y descomponerlo en todos sus componentes o partes que de una u otra manera intervienen tanto en su planteamiento como en su solución.

Una herramienta rápida que nos permite descomponer en partes un problema para su solución, es el llamado modelo de solución, este consiste de una pequeña caja que contiene los tres elementos más básicos en que se puede descomponer cualquier problema sencillo de información, estas tres partes son:

1. LA PRIMERA PARTE son todos los datos que el computador ocupa para resolver el problema, estos datos son almacenados internamente en la memoria del computador en las llamadas variables de entrada.

1. LA SEGUNDA PARTE son todas las operaciones generalmente algebraicas necesarias para solucionar el problema, generalmente esta parte del modelo es una formula (o igualdad matemática, ej. X= y + 5).

1. LA TERCERA PARTE es el resultado o solución del problema que generalmente se obtiene de la parte de operaciones del modelo y dichos datos están almacenados en las llamadas variables de salida.

En resumen para todo problema sencillo de información es necesario plantearse las siguientes preguntas:

Que datos ocupa conocer el computador para resolver el problema y en cuales variables de entrada se van a almacenar?

Que procesos u operaciones debe realizar el computador para resolver el problema planteado ?

Que información o variables de salida se van a desplegar en pantalla para responder al problema planteado originalmente?

Como nota importante no confundir los términos datos, variables e información;

Datos se refiere a información en bruto, no procesada ni catalogada, por ejemplo “Tijuana”, “calle primera # 213”,”15 años”, ” $2,520.00”, etc.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Variables es el nombre de una localidad o dirección interna en la memoria del computador donde se almacenan los datos, ejemplo de variables para los casos del inciso anterior, CIUDAD, DIRECCIÓN, EDAD, SUELDO, ETC.

Información son datos ya procesados que resuelven un problema planteado.

EJEMPLO DE MODELO DE SOLUCIÓN

Construir un modelo de solución que resuelva el problema de calcular el área de un triángulo con la formula área igual a base por altura sobre dos.

Variable(s) de Proceso u Variable(s)

Entrada operación salida

BASE ÁREA = BASE * ALTURA ÁREA

ALTURA 2

PROBLEMA 2.- PROGRAMACION CONVERTIR LA EDAD EN AÑOS DE UNA PERSONA A MESES.

PROBLEMA 3.- CONVERTIR PESOS A DÓLARES.

PROBLEMA 4.- CALCULAR EL ÁREA DE UN CIRCULO CON LA FORMULA

$$AREA = PI * RADIO^2$$

PROBLEMA 5.- EVALUAR LA FUNCIÓN Y = 5X^2 -3X +2

PARA CUALQUIER VALOR DE X.

* Observar para el caso de constantes fijas o conocidas (PI) no se debe dar como dato de entrada su valor, en cambio colocar directamente su valor dentro de la formula, en la parte de operaciones del problema.

* Pero recordar también que existirán problemas sencillos donde:

* No se ocupan entradas o no se ocupan operaciones, pero todos ocupan salida.

* Una formula grande o muy compleja puede ser más segura y fácil de resolver, si es descompuesta y resuelta en partes, juntando al final los parciales para obtener el resultado final.

* Un problema puede tener más de una solución correcta.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

* El problema no esta suficientemente explicado o enunciado, entonces, estudiarlo, analizarlo y construirlo de manera genérica.

TAREAS PROGRAMACION VISUAL BASIC

Construir los modelos de solución de los siguientes problemas:

PROBLEMA 6.- calcular la probabilidad que en los casinos las ruletas tiren el 2 rojo

PROBLEMA 7.- Convertir 125 metros a centímetros (no ocupa entradas)

PROBLEMA 8.- Se calcula que en promedio hay 4.7 nidos en cada árbol en la UABC, también se calcula que en cada nido existen un promedio de 5.8 pájaros, se pide calcular la cantidad total de nidos y de pájaros en los 227 arboles que existen en la UABC. (No ocupa entradas)

PROBLEMA 9.- La gorda Sra. López y sus 8 hijos solo compran una vez al mes su mandado en conocido supermercado, en dicha tienda el kilogramo de frijol cuesta $8.75, el paquete de tortillas cuesta $3.55 y el frasco de café vale $14.25, si solo compran de estos tres productos para su mandado, calcular su gasto total.(problema no claro)

PROBLEMA 10.- Capturar y desplegar los cinco datos mas importantes de un automóvil (no ocupa operaciones)

PROBLEMA 11.- La distancia Tijuana - Ensenada es de 110 Kms, si un automóvil la recorre a una velocidad constante de 30 millas por hora, cuanto tiempo tarda en llegar. (1 milla =1.609 Km.) (Dos maneras correctas de resolverlo).

PROBLEMA 12.- Evaluar la función $y = 3x^2 + 2x - 5$ para cualquier valor de x. (caso normal).

PROBLEMA 13.- Evaluar la función $y = -5x^3 - 3x^2 + 8$ para cuando x vale 4 . (No ocupa entradas).

3.- VISUAL BASIC VARIABLES

Identificadores son conjuntos de letras y/o números que se utilizan para simbolizar todos los elementos que en un programa, son definibles por el usuario (programador o ingeniero de software) del mismo, como son las variables donde se almacenan datos, funciones (pequeños módulos con código), etiquetas, clases, objetos, etc.

Una variable se define como un identificador que se utiliza para almacenar todos los datos generados durante la ejecución de un programa.

Existen ciertas reglas en cuanto a variables:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

* Claras y con referencia directa al problema. * No espacios en blanco, ni símbolos extraños en ellas. * Se pueden usar abreviaturas, pero solo de carácter general. * No deben ser palabras reservadas del lenguaje.

Ejemplos de buenas variables:

Nombre, Edad, SdoDiario, IngMensual, Perímetro, Calif1, etc.

Visual Basic permite variables de hasta 255 caracteres.

4.- VISUAL BASIC DECLARACIÓN Y TIPOS DE DATOS

A toda variable que se use en un programa, se le debe asociar (generalmente al principio del programa) un tipo de dato específico.

Un tipo de dato define todo el posible rango de valores que una variable puede tomar al momento de ejecución del programa y a lo largo de toda la vida útil del propio programa.

Los tipos de datos más comunes en Visual Basic son:

Tipo Rango

BYTE 0-255

INTEGER(%) +-2,147,483,698

SINGLE(!) 3.4E+-38(7 DECIMALES)

DOUBLE(#) 1.8E+308(16 DECIMALES)

CURRENCY 15 DIG IZQ 4 DIG DEECHA

STRING($) 2 BILLONES CHARS

BOOLEAN TRUE, FALSE

DATE FECHA

VARIANT TODOS LOS TIPOS

También toda variable usada en un programa se deberá declarar al principio del programa (luego se indicara donde se hace esto en visual basic), el formato de declaración más sencillo es:

DIM VARIABLE AS TIPO (EJEMPLOS)

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

DIM ALFA AS INTEGER

DIM ALFA AS LONG, BETA AS LONG

DIM ALFA AS INTEGER, NOMBRE AS STRING

DIM CIUDAD AS STRING * 20, ALFA AS DOUBLE

Observar último caso de ciudad, así de esta manera se consigue una string de tamaño definido.

5.- VISUAL BASIC OPERADORES ARITMÉTICOS

Un operador es un símbolo especial que indica al compilador que debe efectuar una operación matemática o lógica.

VISUAL BASIC reconoce los siguientes operadores aritméticos:

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División Flotante
\	División Entera
MOD	Modulo o Residuo

El Operador (/) División Flotante, es el operador de división normal.

El Operador (\) también es división, pero los datos primero, son redondeados al entero mas cercano (byte, integer, long) y al final, se trunca la parte residual.

El Operador (^), nos resuelve dos problemas:

a) Potencias, por ejemplo 3^2; se resuelve como:

dim alfa as integer

alfa = 3 ^ 2

Desplegar alfa ——> Sale 9 en pantalla

b) Raíces solo recordar la ley de exponentes que dice:

Para estos casos:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

<m> root{n}{a^m} = a ^ (m/n) </m>

$\sqrt[3]{5^8}$ = 5^8/3

$\sqrt{9}$ = 9^1/2

En Visual Basic el operador (MOD) devuelve el residuo entero de una división entre enteros, ejemplo;

Dim alfa As Single;

alfa = 23 MOD 4;

Desplegar alfa; ß — El resultado en pantalla es 3

Otro ejemplo;

Alfa = 108 MOD 10;

Desplegar alfa; ß – El resultado en pantalla es 8

TAREAS PROGRAMACION VISUAL BASIC

1. .- $y = 4x^3 - 3x^2 + 2x - 5$

1. .- $y = 5\sqrt[3]{x} + 4x^2 + 6$

1. .- $y = -8\sqrt[5]{x^3} - 3x^3 + 6x^2 - 7$

6.- VISUAL BASIC JERARQUIA DE OPERACIONES

En Visual Basic el problema de no tomar en cuenta la jerarquía de los operadores al plantear y resolver una operación casi siempre conduce a resultados muchas veces equivocados como estos:

Ejemplos: a) 2+ 3* 4 = 20 (incorrecto)

= 14 (correcto)

b) si calif1=60 y calif2=80 entonces si en programa se usa promedio=calif1 + calif2/2 da como resultado promedio = 100

En Visual Baic recordar siempre, que antes de plantear una formula en un programa se deberá evaluar contra el siguiente:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Orden de operaciones:

1.- Paréntesis

2.- Potencias y raíces

3.- Multiplicaciones y divisiones

4.- Sumas y restas

5.- Dos o más de la misma jerarquía u orden, entonces resolver de izquierda a derecha

Nota: Si se quiere alterar el orden normal de operaciones, entonces usar paréntesis.

Nota: Tampoco es bueno usar paréntesis de mas en una operación, esto solo indica que no se evaluó bien la formula, como en el siguiente ejemplo;

Área = (base * altura) / 2

Aqui los paréntesis están de mas, porque por orden de operaciones, multiplicación y división tienen la misma jerarquía y entonces se resuelven de izquierda a derecha, en otras palabras ni que falten paréntesis ni que sobren paréntesis.

7.- PROGRAMACION ORIENTADA OBJETOS VISUAL BASIC

Para nuestro propósito en general, un objeto puede definirse como cualquier ente o entidad física o lógica de información.

En este sentido, todos los elementos materiales o inmateriales pueden clasificarse como objetos.

En particular cualquier objeto considerado, presenta los siguientes tres elementos:

a) Propiedades: Son las características propias de un objeto, estos atributos, son los que permiten diferenciar o individualizar un objeto de otro objeto ya sea de la misma o diferente clase o categoría.

Las propiedades más generales son forma, color, tamaño, peso, etc., pero ya en particular:

Chamarra → Marca, material, precio, color, tamaño, etc.

Alumno → Matricula, nombre, edad, domicilio, etc.

Gato → Raza, nombre, color, edad, etc.

VentanaWindows–>Tamaño, Color, font, etc.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

b) Métodos: Son las conductas propias de la naturaleza del objeto.

Así como las propiedades son el ser (que es) del objeto, los métodos son el hacer (que hacer) del objeto.

Ejemplo de métodos:

Gato —> Maullar(), comer(), correr(), saltar(), etc.

Alumno—> Estudiar(), comer(), asistir clase(), pintear()

Cuaderno–>Esescrito(), esrayado(), esborrado(), etc.

VentanaWindows–> Abrir(), cerrar(), maximizar(), etc….

c) Eventos: Es la relación (de varias maneras) que se puede dar entre dos objetos, ya sean de la misma o diferente clase.

Un evento se manifiesta como un interacción entre dos objetos, en general al momento de la relación, al mismo tiempo se dará una reacción o respuesta por parte de los dos objetos, que se manifiestan como una serie, cadena o conjuntos de métodos propios que se activan. ejemplo:

Evento relación métodos que se activan

gato detecta gata detectar maullar(), correr(), oler()

gato detecta perro detectar bufar(), saltar(), correr()

maestro enseña alumno enseñar pasar lista(), preguntar(), tc

Windows click ratón click maximizar(), cerrar()

Windows dblclk ratón dblclk minimizar(), etc…

Un Programa visual, en Visual BAsic es un conjunto de una o mas formas, donde cada una de ellas, contiene un conjunto de componentes o controles.

Una forma en tiempo de diseño, es una ventana de Windows al momento de la ejecución del programa.

Un componente o propiamente dicho un control, es un objeto que se especializa en una tarea especifica, por ejemplo hay controles especializados en desplegar textos o mensajes, otros controles se especializan en desplegar imágenes o vídeos, otros en manipular directorios o archivos en disco, etc.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Pero en general tanto las formas como los controles, no dejan de ser objetos en programación y por tanto, tienen sus propiedades, métodos y están sujetos a eventos.

8.-IDE COMPILER AMBIENTE VISUAL DE DESARROLLO

Entradas o capturas de datos y salidas o despliegues de información o resultados son de los procesos más comunes en cualquier tipo de problema de información, estos procesos o instrucciones varían de acuerdo a los lenguajes y ambientes de programación a usar.

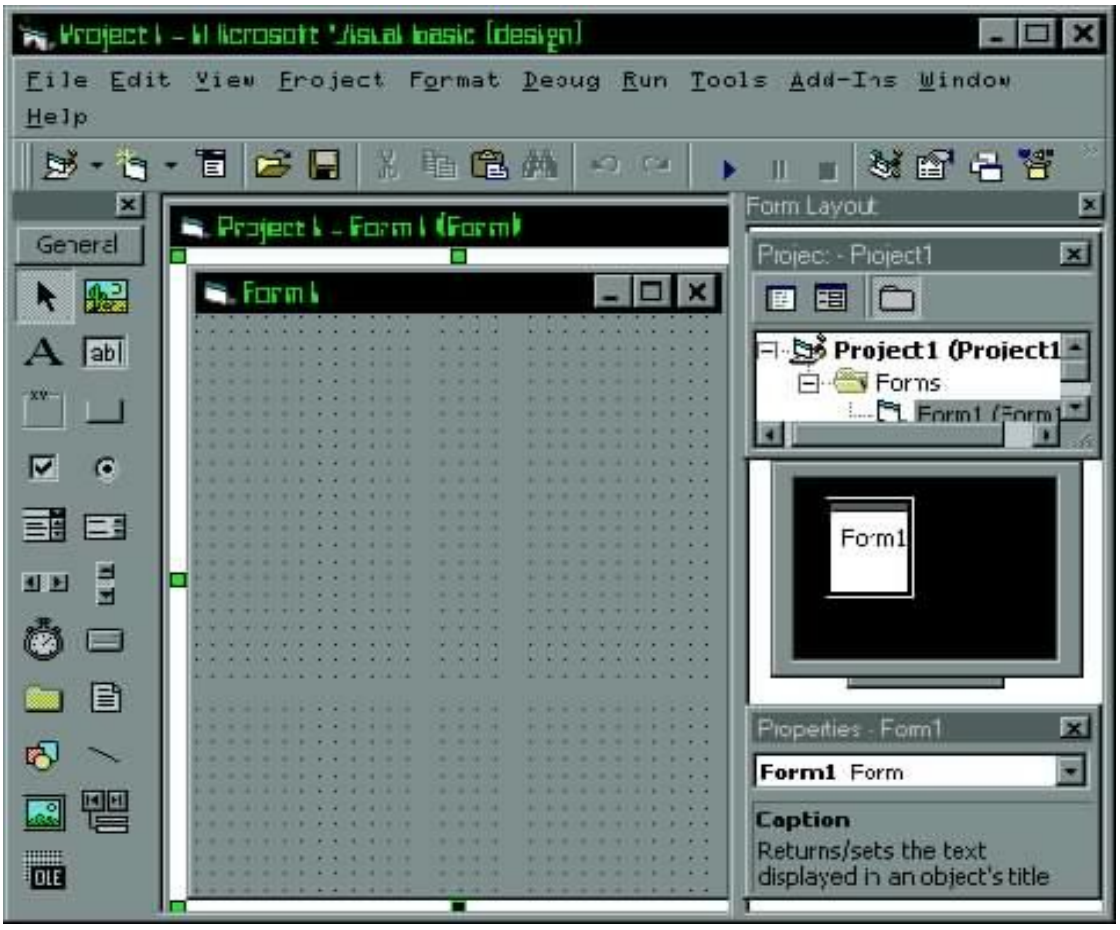
El lenguaje y ambiente de programación a utilizar, es de tipo visual (VISUAL BASIC) y muchos de los problemas asociados a entradas y salidas se encuentran ya resueltos por el propio compilador.

El ambiente de construcción de programas a usar, es el siguiente:

* SOLO CARGARLO EN PANTALLA EJECUTANDO EL VISUAL BASIC, QUE SE

ENCUENTRA EN LA BARRA DE START DE WINDOWS.

Sus elementos básicos son:



Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

1.- Barra de Titulo: Contiene el nombre del programa, tarea o sistema de información que se este desarrollando.

2.- Barra de Menú: Es una barra normal de menús, con opciones de, abrir archivos, cut, paste, imprimir, etc., mas algunas opciones normales para lenguajes de programación, tales como compile, run, etc.

3.-Tool Bar (barra de herramientas): Contiene una serie de iconos, que facilitan algunas de las opciones que están en la barra de menús, por ejemplo el icono de impresora, es el equivalente a la opción, file, print, el icono de start, es el equivalente a la opción run, etc.

4.- Tool Box (caja de herramientas): Contiene los veinte controles que por default Visual Basic proporciona, es de esta caja de herramientas donde se toman los controles y se pasan a la forma que los contendrá, para construir un programa en Windows.

5.-Form Windows: Es la parte principal del ambiente de desarrollo visual de programas (IDE), contiene Form1, que es la ventana principal del programa, o la primera ventana que el usuario observara, al ejecutarse el programa.

6.-Explorador de Proyectos: Es el administrador de el proyecto, recordar que un proyecto completo, son muchas formas, cada forma o ventana con muchos controles, etc. y es en el administrador de proyectos, donde se crean, destruyen formas, controles, etc.

7.-Pagina de Propiedades: Contiene todas las propiedades asociadas a una forma o a un control, es en esta pagina donde se podrá modificar en forma estática una propiedad de alguna forma o un control.

Nota: Para activar el explorador de proyectos o la pagina de propiedades, existen cuatro maneras:

a) Click en la parte superior de la ventanilla

b) Usar la opción de view explorer, en la barra de menús

c) Click en el icono apropiado en la barra de herramientas

d)Click en forma o control, para seleccionarlo, luego click derecho para que aparezca un minimenu y usar opción propiedades.

Nota: Para desactivarla, solo click arriba en la parte superior

8.- Form Layout: Se utiliza para posesionar la forma al momento de la ejecución del programa.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

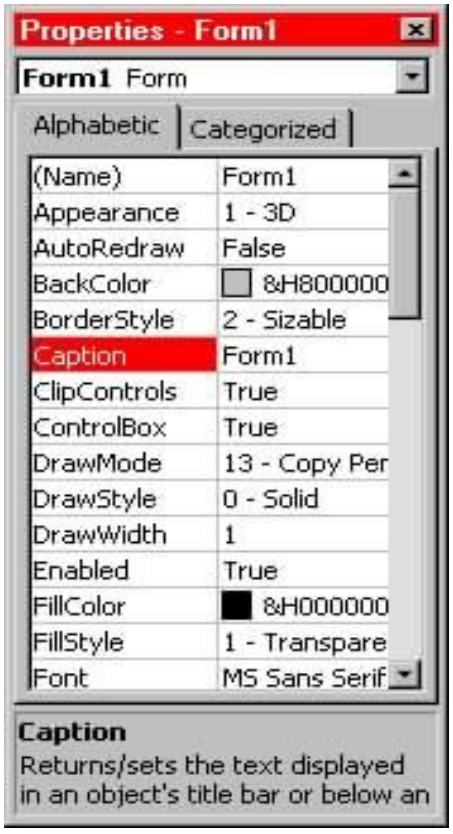
9.- Visual Basic FORM1 ACTIVA O PRINCIPAL

Es sobre esta forma donde se construye el programa y esta forma se convierte en ventana al momento de ejecutarse el programa.

Es decir será la primera ventana que el usuario ve al momento de ejecutarse el programa, su nombre es Form1.

Esta forma o ventana es un objeto de Visual Basic y como todos los objetos de Visual Basic y del universo, la forma o ventana tiene asociados propiedades y eventos.

Propiedades son todas las características particulares que diferencian un objeto de otro objeto, las propiedades o características mas comunes son forma, tamaño, color, etc., para objetos en Visual Basic, estas propiedades se modifican o individualizan usando la pagina de propiedades, que es la parte del programa que las contiene.



* También se pueden modificar las propiedades dentro de un programa, usando instrucciones apropiadas, mismas que llevan el siguiente formato:

nomobjeto.propiedad = nvovalor

Ej.;

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Ni modo, los colores tendrán que darse en hexadecimal, mas adelante se indicara como.

Eventos, son todos aquellos sucesos de carácter externo que afectan o llaman la atención del objeto, para este caso la forma o ventana:

1. Debe tener capacidad de detectar el evento

2. Aun más importante debe tener capacidad de reaccionar y emitir una respuesta, mensaje o conducta apropiada a el evento detectado.

Evento es por ejemplo que otro objeto llamado humano, pulse el objeto tecla ESC, o haga click derecho con el objeto ratón en alguna parte de la ventana , etc. , es en estos casos, cuando la ventana detecta un evento de estos, la propia forma deberá responder de manera apropiada.

Esta respuesta no es automática, sino son la serie de instrucciones del lenguaje (o programa) que los ingenieros de software diseñan(o programan), en otras palabras son los eventos quienes contendrán los programas.

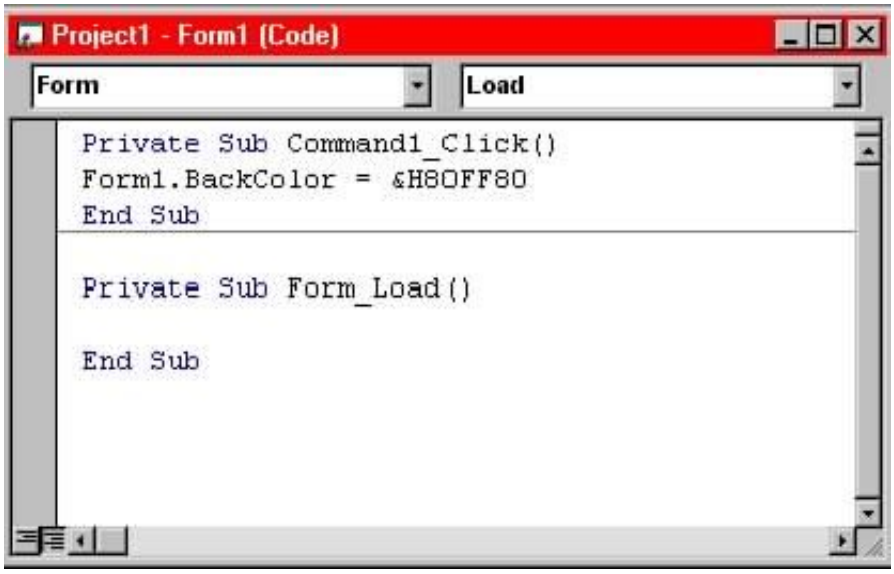
Es la Ventana de Código (Code View), quien contiene todos los eventos que un objeto (forma o control) puede detectar, para activar la ventana de código, también hay varias maneras:

a) DobleClick en forma o control

b) Click en Forma o control y luego click derecho para minimenu y usar opción Code

c)Usar View, Code, en la barra de menús Aparece el siguiente editor de Código:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



La primera ventanilla contiene todas las formas y controles que se están usando en el proyecto y la segunda ventanilla contiene todos los eventos asociados al objeto.

Recordar que para intercambiarse entre las diversas ventanas del IDE de Visual Basic, la de forma, la de propiedades, la de código, etc., se pueden usar las opciones del menú, los iconos del tool bar, clicks en las partes superiores de las ventanas, click en el explorador de proyectos, click derecho para minimenus, la opción Windows, tile, cascade, etc.,

10.- VISUAL BASIC PROGRAMAS, FORMAS Y COMPONENTES

Un programa o problema de información en Visual Basic, no es mas que una o mas formas o ventanas, donde cada una de ellas contiene elementos u objetos especiales llamados controles o componentes, dichos controles Visual Basic los proporciona a través de la caja de herramientas

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

En orden de izquierda a derecha ellos son:

Pointer(no es control se usa para des marcar un control previamente seleccionado), PictureBox, Label, TextBox, Frame, CommandButton, CheckBox, OptionButton, ComboBox, ListBox, HScrollBar, VScrollBar, Timer, DriveListBox, DirListBox, FileListBox, Shape, Line, Image, Data, Ole.

Es decir toda la interfase que se quiera manejar con el usuario del programa, no consiste más que de una colección de estos controles agrupados en una forma o ventana.

Para incorporar un componente a una forma solo basta seleccionarlo con un click en su icono y luego colocar el cursor dentro de la forma en el lugar donde se quiere que aparezca y arrastrando abrir toda el área donde se quiere que se quede, al final hacer click otra vez.

Luego Click en otro componente, colocar cursor donde se quiera que quede, abrir zona donde se queda y al final click, y así sucesivamente.

Para eliminar o borrar controles no deseados, solo click en dicho control para seleccionarlo y tecla DEL.

Recordar que si en Tool Box un control esta seleccionado, se puede deseleccionar usando pointer(el primer icono con una flechita).

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

También los controles son objetos de Visual Basic y como tales también tienen asociados propiedades y eventos, tales como los tiene la forma principal, solo que existen pequeñas variaciones en cuanto a sus propiedades y eventos propios con respecto a Form1.

Recordar además, que es la pagina de propiedades en primera instancia quien permite asociar o modificar propiedades especificas tanto a una forma como a un componente.

Ya en segunda instancia las propiedades de formas y controles se pueden modificar también directamente dentro de un programa, usando instrucciones como las ya descritas en párrafos muy anteriores.

Analizaremos ahora los tres controles más básicos que se usan para construir un programa sencillo en Visual Basic.

11.- VISUAL BASIC CONTROL LABEL 

Este componente se utiliza para desplegar textos o mensajes estáticos dentro de las formas, textos tales como encabezados, solicitud al usuario del programa para que proporcione algún dato o información (edad, dame sueldo, etc.), en cierta forma hace las funciones de printf, cout, writeln, print, display, etc., pero solo cuando se consideran en su parte de mensajes.

También es un objeto en Visual Basic y por tanto tiene asociados sus propias propiedades y eventos, al mismo tiempo como se está usando dentro del objeto form1, muchas propiedades que se definan para el objeto Form1, el objeto Label1 las va a heredar.

Si bien es cierto que el objeto se llama Label, pero cuando se ponen dentro de una forma Visual Basic los va numerando automáticamente, si se ponen tres Labels en Form1, ellos se llaman, simbolizan, procesan o programan como Label1, Label2, Label3.

Es su propiedad Caption, la que lleva el contenido del mensaje que se quiere desplegar en la pantalla, solo click derecho a un lado de la propiedad Caption en la pagina de propiedades, teniendo seleccionada la caja Label1 en la forma y escribir el texto indicado.

12.- CONTROL TEXTBOX 

Este componente es el mas importante componente visual, su función principal es manejar , todos los procesos de entrada y salida (input/output) al programa.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

En resumen de este componente, se necesita entender lo siguiente:

Este componente Text, es el equivalente a las variables en cualquier lenguaje de programación, mas la instrucción de captura o despliegue correspondiente, es decir;

a) En BASIC, Input Edad → Text1

b) En PASCAL, Read(Ciudad) → Text2

c) En C, printf("%d", sueldo) → Text3

d) En C++, cin.get(nombre,30) → Text4

e) En Cobol Display metros → Text5

Observar que en Tool Box se llama TextBox, pero ya dentro de la forma y dentro del código se llama simplemente Text.

Ya aclarada su función, es necesario entender primero, que este componente permite capturar datos y también como en el caso del componente Label desplegar datos, textos, mensajes o resultados de operaciones de ser necesario, usando la propiedad Text del Control Text.

Esta propiedad Text, así como la propiedad Caption en Label, permiten igualarse a muchos procesos básicos, es decir es fácil igualar Text o Caption a un dato, una variable, otro Text u Caption, o una expresión algebraica normal, como en los siguientes ejemplos;

Text1.Text = 5

Label3.Caption = "PATO"

Text4.Text = 3 * 6.2

En principio su valor de default es la palabra Text1, es en su propiedad Text donde se modifica, generalmente al principio de un programa se deja en blanco, y al ejecutarse el programa, el usuario lo llena con los datos solicitados o el programa lo llena con el resultado de las operaciones.

Cuando un usuario lo carga con un dato, recordar que el dato almacenado queda de tipo texto, no importa lo que haya escrito el usuario.

Para resolver el problema de usar datos numéricos se deberán usar las siguientes funciones de conversión de datos:

FUNCIÓN CONVIERTE A

Cbool Boolean

Cbyte Byte

Ccur Currency

Cdate Date

Cdbl Double

Cint Integer

Clng Long

Csng Single

Cstr String

Cvar Variant

Su formato completo es:

Función(dato, var, expresión, control) ej.;

a) Cint(3.1416) —>Convierte a entero y adiós decimal

b) Si Promedio = 45 y CSng(Promedio)—> se despliega 45.00000

c) Text1.Text= Cint(Text2.Text) * 4

Primero lo que el usuario cargue en el control Text2 lo va a convertir a entero, luego lo multiplica por cuatro y al final carga el resultado en el control Text1.

Observar que Text1 no ocupa convertirse a dato numérico, porque el resultado no importa si es texto o es numero.

Solo que mas adelante en el programa se ocupe el valor o dato almacenado en Text1, entonces si se ocupara convertir, por ejemplo:

Text5.Text = CLng(Text1.Text) - CInt(Text3.Text) / 10

Recordar que lo mas importante, es que este Control Text#.Text sustituye a las variables en programas normales, por ejemplo si un problema dice:

dólares = pesos / tipo cambio

queda como:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Text3.Text = CSng(Text1.Text) / CSng(Text2.Text)

Es importante recordar que dos controles cualquiera que tengan propiedades Caption o Text, pueden intercambiar directamente sus datos entre si.

En resumen, este control Text, es el control mas importante y elemental en todo problema que involucre el procesamiento de datos en ambientes visuales, se debe acostumbrar a considerar como una variable normal cualesquiera.

13.- VISUAL BASIC CONTROL COMMANDBUTTON 

Es el control principal de la forma, contiene el código principal del programa y su activación por el usuario provoca que se realicen los principales procesos del problema planteado (aquí es donde se capturan datos, se realizan operaciones, etc.).

De este control se maneja su propiedad Caption para etiquetarlo con la palabra “OK” o “ACEPTAR” o “EXE” , y su evento Click para activarlo, es en dicho evento donde se construye el código del programa.

Recordar que aunque no es un control necesario en los programas, ya que el código se puede asociar o pegar a cualquier evento de cualquier forma, o control del programa, Microsoft ya acostumbro a todos los usuarios al botón OK, de acuerdo, OK.

Este botón también puede activar su evento Click, cuando el usuario presione la tecla <ENTER>, solo poner la propiedad Default en true, en este caso el botón de ordenes, se le conoce como botón de default.

Igualmente puede activar su evento Click cuando el usuario, presione la tecla <ESC>, solo poner la propiedad Cancel en true, a este caso se le conoce como “CANCEL BUTTON”.

También puede mostrar un icono o imagen gráfica, primero poniendo su propiedad style = 1 (gráfico) y luego usando su propiedad picture para buscar una imagen adecuada.

Igualmente también puede usarse como un botón de salida o terminación o exit del programa, solo cargando la instrucción END en su evento click, como en el siguiente ejemplo en visual basic:

Private Sub Command1_Click()

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

End

End Sub

Pero su uso mas importante es contener el código de los procesos u operaciones del problema.

PRACTICA PARA CREAR UN PROGRAMA

Se construye ahora el programa de calcular el área de un triángulo con la formula área igual a base por altura entre dos.

El procedimiento completo para crear y ejecutar el programa es:

1.- Cargar Visual Basic

2.- Al cargarlo ya estará en la pantalla la primera forma (Form1).

3.- Antes de poner el primer control usar la opción, File Save Project As, aparece la siguiente ventana:

Donde se deberá seleccionar primero, el icono de nuevo folder(arriba a la derecha y tiene un folder con rayitos), esto es, para crear un nuevo folder o directorio donde quedara guardado o almacenado el programa, en cuanto se crea el nuevo folder, sobrescribir la palabra “new folder” que aparece, con el nombre que tendrá el directorio donde quedara almacenado el programa, escribir por ejemplo “programa uno” (Windows95 ya permite que los nombres de directorios y sus archivos ya sean mas largos y no necesariamente del tipo antiguo de 8.3), al terminar de sobrescribir, la

palabra “programa uno” apretar tecla <ENTER> y esperar un momento a que se cree el directorio.

Ya creado y renombrado el directorio, observar que en la parte inferior de la ventana el programa ya tiene el nombre de “Form1”, a un lado esta una caja o botón de “OPEN”, mismo que se deberá apretar y después usar en la misma parte un botón llamado “SAVE” para almacenar “Form1” y otra vez usar otro vez botón “SAVE” para almacenar “Project1”.

4.- Ahora ya que se tiene Form1 en pantalla, recordar que se pueden modificar sus propiedades como color, font, etc. usando la pagina de propiedades que esta a un lado de la forma (se sugiere practicar un poco esto), los cambios que se hacen en la pagina de propiedades se van reflejando automáticamente en la forma en pantalla y también en la ventana que el usuario vera al ejecutarse el programa.

5.- Ahora seleccionar con un click el control llamado Label en la barra de herramientas y luego poner el cursor dentro de la forma en el lugar donde se quiera que aparezca el control y manteniendo apretado el click del ratón abrir el espacio donde quedara el control, para que aparezca el control solo liberar el botón del click.

Observar que un control en la forma, que esté seleccionado (esto se puede hacer, usando un click dentro de el componente) se puede arrastrar para cambiarlo de lugar o posición o hacerlo mas pequeño o mas grande.

Para cargar o para que despliegue un texto el componente Label1, solo escribir dicho texto en la cajita que esta a un lado de la propiedad Captión en la pagina de propiedades.

6.-Repetir el procedimiento anterior hasta tener todos los controles en la forma (son cinco labels, tres Text y un Command)

7.- Seleccionar y acomodar ahora el control Command1 en la forma y colocarlo en la esquina inferior derecha, en su propiedad Caption = escribir la palabra “OK”.

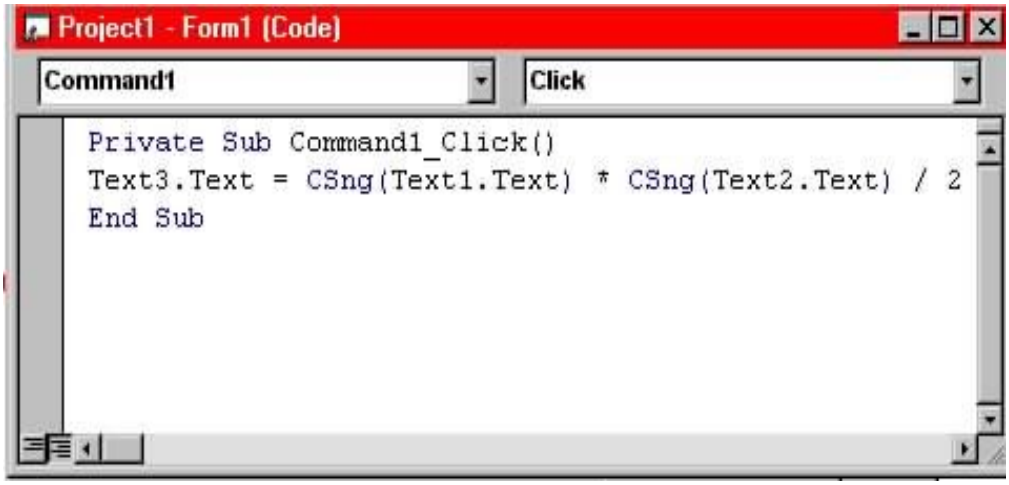
Recordar que este control, es quien contiene todo el código del programa y mas específicamente es su evento Click quien lo contiene y quien además lo activa o ejecuta.

8.-Para añadirle el código existen varias maneras:

a) Hacer un doble click en este control command, para que aparezca la siguiente ventana del editor de código (Code View):

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



Observar que ya viene cargado con el evento Click del control Command1.

Solo escribir las instrucciones dentro del PRIVATE SUB y EL END SUB

El código corresponde a la formula área = base * altura /2

Solo se usan los controles Text y su conversión a datos numéricos, cuando sea apropiado.

b) Otras maneras de activar el editor de código, es:

b.1) usando la opcion view, code de la barra de menús

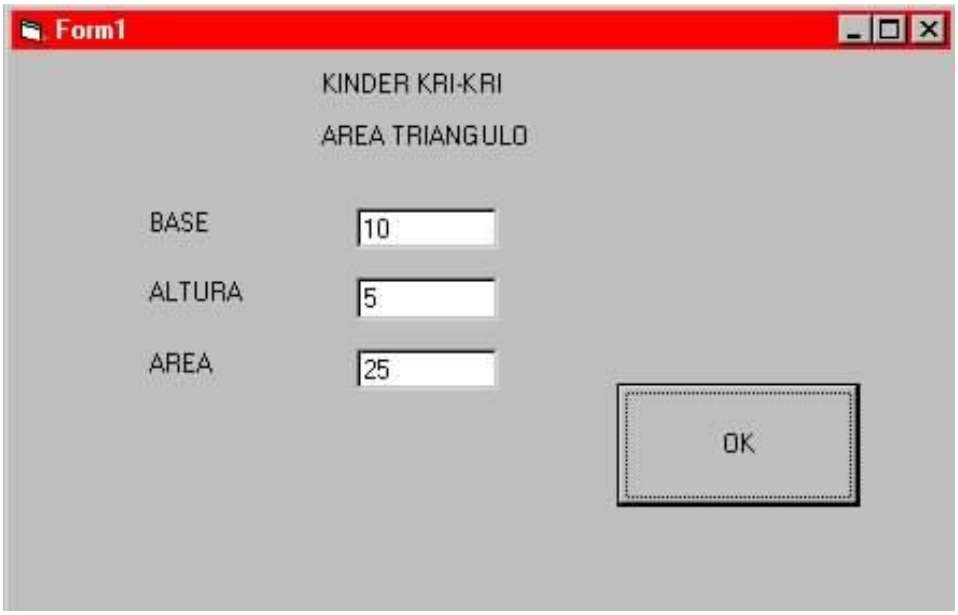
b.2) click en Command1, luego click derecho para minimenu y usar opción code.

9) Usar la opción Run, Start que esta arriba en la barra de menús o también la tecla F5.

10) o usar icono “Start” en barra de herramientas.

El programa ya en ejecución debe ser similar (pero mejor diseñado) al siguiente ejemplo:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Nota:

Si no cargan un dato o valor en Text1 y Text2 y se oprime OK, se obtiene un error al ejecutarse, porque se esta intentando multiplicar y dividir la nada.

Para terminar usar la [X]de arriba o poner un segundo Command con letrero Exit y código END como se indico en el tema del control COMMAND.

11.- Ya creado y ejecutado el programa, se deberá grabar al disco con la opción File, Save

12.- Cerrar el Visual Basic con la opción File, Exit.

TAREAS VISUAL BASIC

1.- Convertir a programas de visual basic todos los problemas vistos en el modelo de solución.

Notas:

El procedimiento general es:

1.-Cargar el Visual Basic

2.- File, save project as.. crear el folder de tarea, renombrarlo, abrirlo y grabarle Form1 y Project1.

3.-Crear y Ejecutar el programa

4.-Grabarlo con File, save

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

5.-Cerrar el Visual Basic con File, Exit

6.-Repetir paso 1

También es posible mandar a impresora toda la forma, en caso de que se quiera tener una copia impresa de la misma, solo poner la instrucción:

PrintForm

Ya sea como ultima instrucción del command1 o de otra manera poner un segundo botón de command2, con caption = impresora y evento clic con la orden ya mencionada.

Recordar que la impresora ya debe estar encendida, conectada a la computadora y con papel.

14.- VISUAL BASIC PRESENTACIÓN Y FORMATO DE DATOS

Como se observa en los programas ya hechos, la salida o despliegue de datos deja mucho que desear, por ejemplo muchos decimales, o muchos ceros, etc.

Para darles mejor presentación se usa la siguiente función de formato de datos.

FORMAT([DATO, VAR, EXPR, CONTROL], "KTE DE FORMATO")

En Visual Basic las constantes de formato son:

CONSTANTE SIGNIFICADO

General Number Formato numérico normal

Currency Separa miles, dos dígitos derecha

Fixed Al menos un dig izq., dos dig derecha

Standard Separa miles, al menos un dig izq, dos dig der.

Percent Numero * 100, signo %, dos dig derecha

Scientific Notación científica exponencial

Yes/No si numero es cero despliega No, en otro caso yes

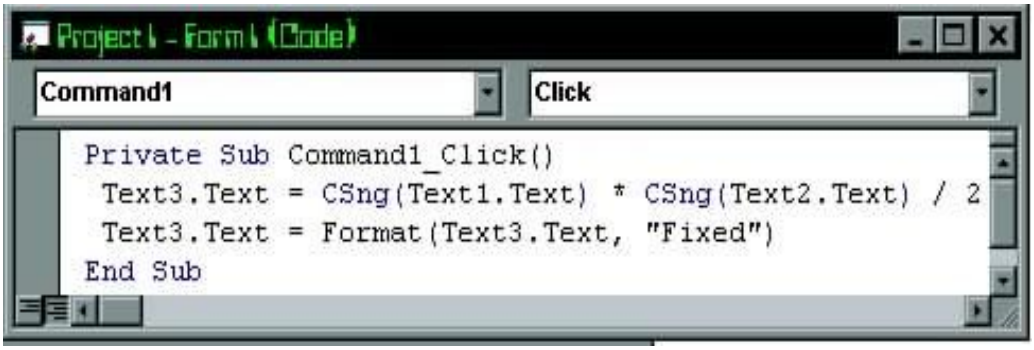
True/False si numero es cero despliega False, en otro caso True

On/Off si numero es cero despliega Off, en otro caso On

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Lo mas recomendable es formatear al final el control de salida : ejemplo área triángulo:



Nota, observar en visual basic que la KTE va entre comillas

15.-CONTROL ComboBox 

Existen muchas ocasiones en donde el usuario del programa tiene que proporcionar datos que provienen de un conjunto finito y muy pequeño de posibles respuestas, esto significa que cada vez que se ejecute el programa, el usuario estará proporcionando las mismas respuestas.

Ejemplo de esta clase de datos, son Municipio en BC, las posibles respuestas solo son (Tecate, Tijuana, Mexicali, Ensenada, Rosarito), otro ejemplo es Sexo (Hombre, Mujer), etc.

Para situaciones como esta, existen componentes que permiten programar por adelantado las posibles respuestas, y el usuario solo debe seleccionar la respuesta apropiada , en lugar de tener que escribirla.

Este componente ComboBox nos permite definir en primera instancia un conjunto de datos o valores respuestas asociados a una caja de edición cualesquiera, así ahora el usuario tendrá la oportUNIDAD VISUAL BASIC de seleccionar un dato del conjunto de datos o respuestas ya predefinido.

Este componente ComboBox tiene dos partes, una parte de encabezado, para poner el nombre del grupo de respuestas(por ejemplo municipios, sexo, etc.), que se carga usando la propiedad Text del componente.

La segunda parte es la lista de opciones o respuestas que se debe cargar al tiempo de diseño de la ventana, en el momento de poner el componente ComboBox1, solo hacer click a un lado de la propiedad list, en la pagina de propiedades y cargar la primera respuesta y enter, luego click otra vez a un lado de propiedad list, cargar segunda respuesta y enter, luego click otra vez a un lado de la propiedad list cargar tercera

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

respuesta y enter, y así sucesivamente, como ya se indico es para cuando son pocas respuestas de antemano conocidas.

Al momento de ejecución del programa, toda la lista de respuestas, estarán a la vista del usuario, para que este último la seleccione.

Recordar que el usuario al momento de ejecución del programa, solo vera el encabezado, para seleccionar su respuesta deberá apretar la flechita que esta a un lado del encabezado.

Para procesar o programar este componente solo usar su propiedad Text de manera normal como si fuese un Control Text, ejemplo:(suponer que se tiene un Combo, cargado con tres números pares)

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un cuestionario de 5 combos con tres respuestas cada uno de ellos sobre hábitos alimenticios de niños, cinco TextBoxs abajo recogen las respuestas.

2.- Evaluar la función $Y = 5X^2 - 3X + 5$ cuando X = 2, 3, 5

16.- CONTROLES DE AGRUPAMIENTO

Como ya se empieza a notar en las aplicaciones construidas, la cantidad de datos e información empiezan a amontonarse en la ventana simple que se ha venido construyendo.

Para resolver este problema, se tienen dos métodos, el primero de ellos consiste de una serie de componentes que permiten agrupar datos o información(resultados) de una manera mas lógica y estética.

El segundo método consiste de construir y trabajar con dos o mas ventanas a la vez.

En este curso de visual basic se empieza por el primero método , es decir componentes de agrupamiento.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

17.- VISUAL BASIC CONTROL FRAME 

También se le conoce como marco o panel, observar que incluye un pestaña, donde se describe en forma lógica todos los controles que agrupa, por ejemplo captura, cuestionario, datos, etc.

Solo se deberá recordar colocar primero todos los paneles en la forma y encima de ellos los componentes que contendrán.

Es decir se puede dividir una sola ventana en varias partes, por ejemplo en un panel se ponen los componentes donde se capturan los datos de un problema junto con el botón de OK, y en otro panel se construye la salida, otro ejemplo se crea un panel para capturar los datos de un empleado incluyendo sueldo diario y días trabajados y un segundo panel contiene su cheque semanal de pago (problema sugerido también).

Para modificar programas construidos sin paneles, el procedimiento para agregarlos es:

1.- Mover todos los controles abajo en la ventana.

2.- Colocar el panel en su posición.

3.- Click en control a relocalizar.

4.- DobleClick, Copy

5.- DobleClick, Cut

6.- Click dentro del panel, donde se quiere el componente

7.- DobleClick, Paste

Para cargar la pestaña, usar la propiedad caption.

Para que no aparezca la pestaña, solo borrar o dejar en blanco la propiedad caption.

TAREAS PROGRAMACION VISUAL BASIC

1.- Usando en visual basic este control Frame otro Cuestionario sobre hábitos de estudios de los alumnos.

18.- VENTANAS (WINFORM) EN VISUAL BASIC

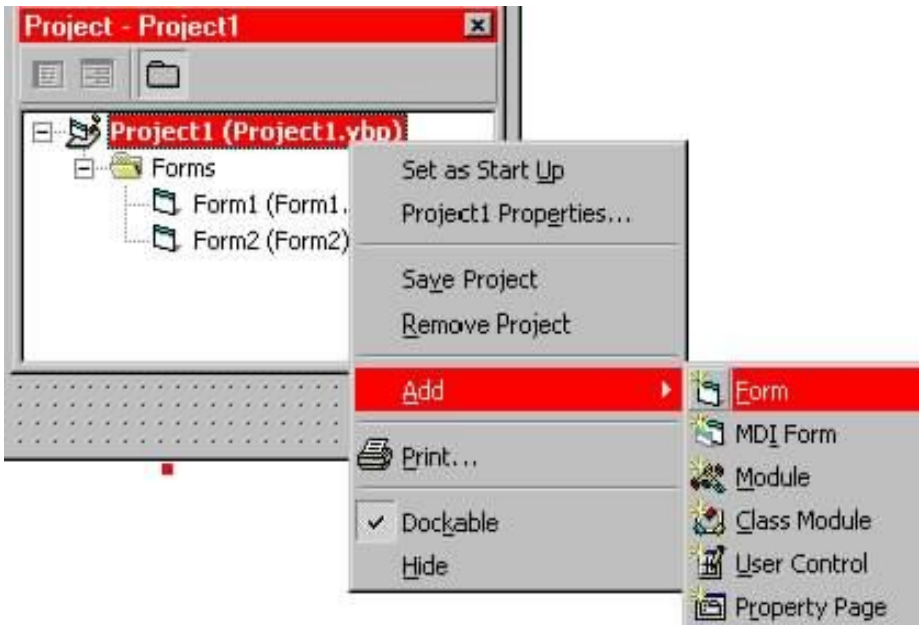
El siguiente problema común, con el manejo de programas en Visual Basic, es el de poder crear, controlar y administrar mas de dos formas o ventanas a la vez.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Lo primero que hay que entender para poder resolver este problema es que en Visual Basic, el default es un proyecto MDI, es decir un proyecto de ventanas dentro de ventanas.

Es decir es muy fácil crear, procesar y administrar varias ventanas a la vez, para esto solo usar el explorador de proyectos:



Para administrar formas:

1.-Creacion de nuevas formas o ventanas, solo click en PROJECT1 y se van abriendo los minimenus arriba indicados, usar ADD, Form, y sale una caja con varios tipos de FORMAS, seleccionar FORM.

Se puede repetir este paso, para crear todas las formas o ventanas necesarias, para el problema.

2.- Eliminación de Formas o ventanas, también en el explorador de proyectos, clik derecho en forma a eliminar, para que salga minimenu y usar opción remove form#

Nota: VB no renumera las formas restantes, si se crean cinco formas y se elimina form3, queda activas form1, form2, form4, form5, si se quieren bien numeradas mejor eliminar las cuatro ultimas y volver a crear las tres que se ocupan.

3.- En la ventana de formas o proyectos, no se muestra, mas que una ventana a la vez, es decir aunque hayan sido creadas cinco ventanas, en la ventana de proyectos sigue apareciendo Form1.

4.-Para poner al frente Form4 y poderla editar, solo hacer un dobleclik en form4 dentro del explorador de proyectos.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Para procesar o programar las ventanas, se ocupa entender dos problemas diferentes:

1.- Movimiento de ventanas.- En este caso, al ejecutarse el programa, estará al frente Form1, como se le hace para llamar las otras formas o ventanas?

Solo Agregar un control extra de Command en la primera forma o ventana y usar las propiedades siguientes de form1,

Form#.Visible = True, False

Form#.Show

Form#.Show 1 (forma modal)

Form#.Hide

Ej:

Recordar que este control extra deberá ir en cada forma o ventana del programa, o no habrá manera de regresarse de la forma5 a la forma1.

2.- Procesar o programar los controles que contiene cada forma#

Recordar que si un programa contiene 5 ventanas y en cada ventana se ponen tres labels, en cada ventana el numero de label empieza en 1(uno) y así pasa con todos los demás controles, incluyendo el command.

Entonces para procesar un control con datos y que Visual Basic los pueda diferenciar, se deberá usar ahora el siguiente formato en el código:

Form#.Control.Propiedad

Ejemplo

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un programa donde la primera ventana capture los datos de un alumno incluyendo las calificaciones de 3 materias diferentes y una segunda ventana despliega un reporte de calificaciones del alumno incluyendo su promedio final.

19.- VISUAL BASIC CONTROLES GRAFICOS

A) CONTROL PICTUREBOX VISUAL BASIC

B) CONTROL IMAGE

Ambos controles permiten desplegar archivos gráficos de tipo, gif, bitmap, icon, jpeg.

Se pueden usar para poner fondos o backgrounds en las ventanas, por ejemplo las nubes de Windows.

Es más conveniente usar el control IMAGE en programas, porque ocupa menos recursos de Windows y lo despliega mas rápido.

Sus propiedades mas importantes son:

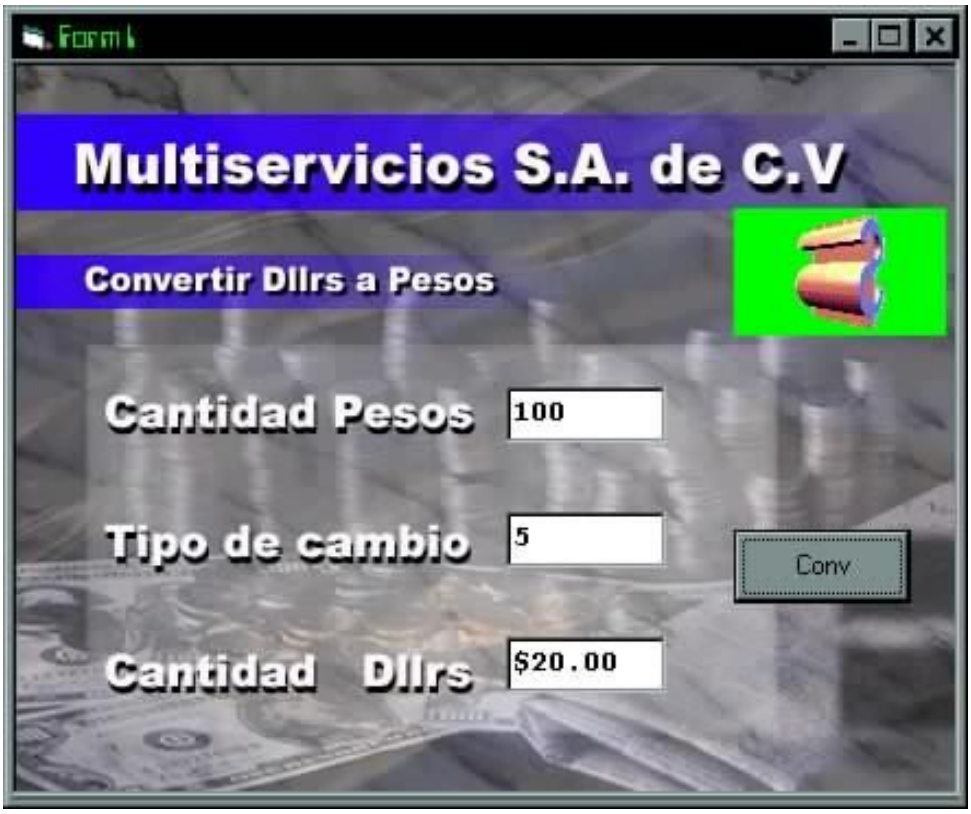
a) Picture: para buscar y cargar el archivo de imagen

b) Autosize = True, para que el control se ajuste a la imagen, ojo, si la imagen es mas chica el control se hace chico, si la imagen es mas grande el control se hace mas grande.

c) Stretch = True, para que la imagen se ajuste al control, si la imagen es mas chica, sola se agranda para cubrir todo el control.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

La cuestión de imágenes y diseño de programas e interfaces, es un aspecto muy importante en programación moderna, como se dice una imagen dice mas que mil palabras, observar el siguiente programa.



Es un control Image de visual basic, con un gráfico BMP editado con el MSPaint y los resultados son:

a) más claro para el usuario

c) menos controles en nuestros programas

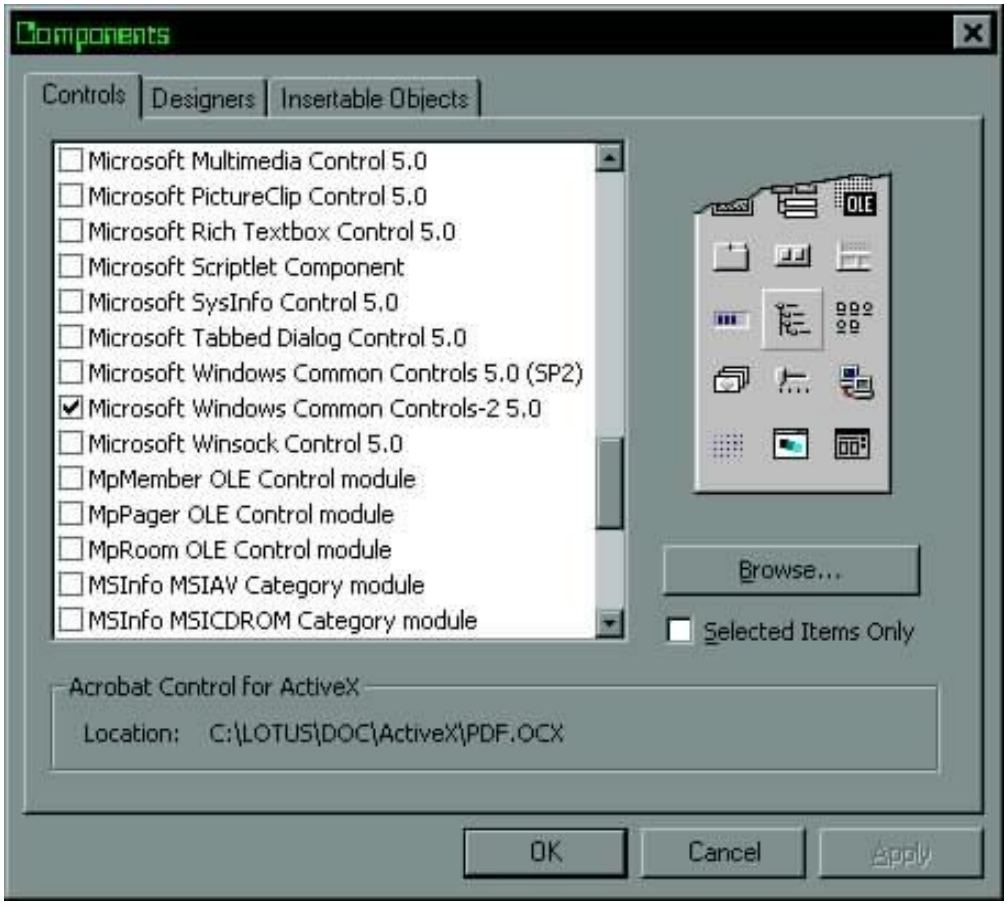
En resumen es mas conveniente, juntar una gran galería de clip art(existen miles de cds en el mercado), un gran conjunto de imágenes escaneadas por ejemplo formas fiscales, formas internas de empresas, etc. y empezar a usar una gran cantidad de imágenes en nuestros programas.

20.- CONTROL ANIMATION VISUAL BASIC

En primer lugar este control no se encuentra entre los 20 controles estándar del Tool Box que proporciona Visual Basic, así que el primer problema a resolver, es como incorporar este control al Tool Box.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

1.- Click Derecho para minimenu, dentro del Tool Box, y usar la opción componentes, aparece la siguiente pantalla:



Como se observa existen 500 controles que pueden incorporarse a nuestros programas.

Para incorporar una galería de ellos, solo marcar con flecha la cajita que tienen al principio y usar el botón apply, observar que aparecen uno o mas controles en Tool Box.

Si no aparecen los controles que se quieren importar a Tool Box, solo desmarcar la flechita que se puso y otra vez usar el botón apply.

El componente animation, se encuentra(marcar con flechita) en Microsoft Windows common controls-2 5.0, recordar usar apply.

Para terminar de importar controles, usar el botón OK que esta abajo en la pantalla.

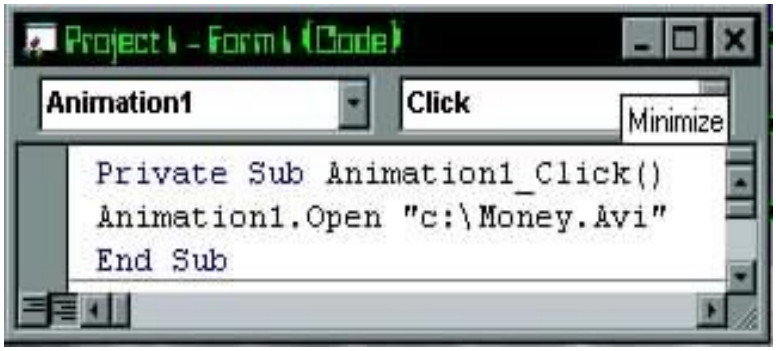
El Control Animation se usa para ejecutar archivos de vídeo con extensión AVI, si observan con cuidado el programa ejemplo del tema anterior, el signo de pesos esta girando.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Atención, solo se puede ejecutar archivos avi, no comprimidos y sin vídeo.

Solo poner su propiedad AutoPlay en True.

Y en el evento Click de Animation (doble click sobre el control animation para cargarlo), escribir el siguiente código:



Archivos Avi de este tipo, también se pueden conseguir en internet, o en cd's comerciales de clip art.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

UNIDAD II

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

II.- INSTRUCCIONES DE CONTROL DE PROGRAMA

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

UNIDAD VISUAL BASIC II INSTRUCCIONES DE CONTROL DE PROGRAMA VISUAL BASIC

1.- INTRODUCCIÓN VISUAL BASIC

En este capitulo se continua siguiendo el esquema de trabajo ya planteado en el capitulo anterior, es decir:

Enfoque Tradicional o Estructurado de programación que hace énfasis en limpieza y eficiencia de los programas, usando las instrucciones normales y apropiadas al problema planteado, estas instrucciones son las comunes en el lenguaje de programación Visual Basic y generalmente están contenidas dentro de los diversos eventos de los diversos componentes aunque de momento solo se han puesto dentro del evento Clic del componente BUTTON.

Enfoque Visual como Visual Basic que hace énfasis en un buen diseño y constante interactividad con el usuario de los programas, esto se hace usando componentes visuales que facilitan dicho proceso o como en este capitulo, donde se analizan componentes que en cierta manera son similares o realizan funciones parecidas a las instrucciones comunes de un lenguaje de programación.

2.- INSTRUCCIONES DE CONTROL DE PROGRAMA EN VISUAL BASIC

Instrucciones de control de programa permiten alterar la secuencia normal de ejecución de un programa.

Estas instrucciones se dividen en tres grandes categorías:

1. Instrucciones Condicionales que en Visual Basic se implementan con las instrucciones if y select case.

b) Instrucciones de ciclos con

* for * do while loop * do loop while

c) Instrucción de salto incondicional

* goto

En Visual Basic varias de ellas tienen sus correspondientes componentes visuales.

3.- VISUAL BASIC INSTRUCCIONES CONDICIONALES

Una de las mas poderosas características de cualquier computador es la capacidad que tiene de tomar decisiones.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Es decir al comparar dos alternativas diferentes el computador puede tomar una decisión, basándose en la evaluación que hace de alguna condición.

Ejemplo de instrucciones condicionales;

si sueldo > 3000

desplegar rico

si no

desplegar pobre

Fin-si

si sexo = 'm'

imprime mujer

si no

imprime hombre

Fin-si

De los ejemplos, observar que los caminos a seguir por el computador dependerán de la evaluación que el computador hace con y de la condición.

Todo lenguaje de programación debe tener instrucciones que permitan formar condiciones e instrucciones que pueden evaluar esas condiciones.

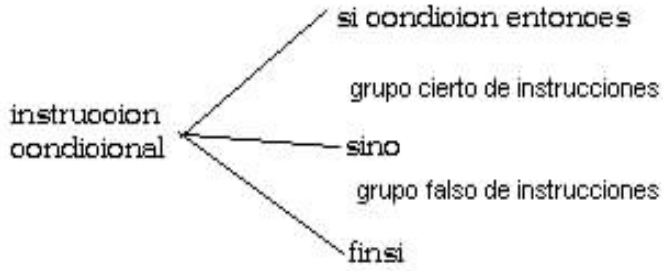
Para propósito de construcción visual de programas, este tipo de instrucciones condicionales se usaran en forma interna es decir son parte del código del programa que se empotra dentro de los eventos de componentes, no son formas o componentes en si.

Pero recordar que lenguajes visuales como visual basic de igual forma tienen componentes que permiten del mismo modo al usuario tomar decisiones, incluso directamente en pantalla, es decir existen los llamados componentes de selección y decisión.

El formato general de una instrucción condicional es:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



Como se observa, son cuatro partes bien diferenciadas entre si;

* La propia instrucción condicional en si * La condición * El grupo cierto de instrucciones * El grupo falso de instrucciones

Cuando el computador evalúa una condición,, el resultado de esa evaluación solo es evaluado de dos maneras o la condición es CIERTA o la condición es FALSA,

Esto dependerá del valor que tenga asignado o que se haya capturado para la variable que esta en la condición, por ejemplo si se capturo 6000 en sueldo en el ejemplo a), entonces el computador indicaría que la condición es CIERTA, pero en otro caso, si a la variable sueldo primero se le asigno un valor de 250 entonces el computador indicaría que la condición es FALSA.

Ya dependiendo del resultado de la evaluación, el computador ejecuta las instrucciones contenidas en el grupo de cierto o falso respectivamente.

En Visual Basic empezaremos el análisis por la CONDICIÓN.

4.- VISUAL BASIC CONDICIONES SIMPLES

En Visual Basic todas las condiciones se forman con;

Variables operadores relaciónales constante o var.

sexo = m

sueldo > 300,000

Una condición simple se define como el conjunto de variables y/o constantes unidas por los llamados operadores relaciónales.

5.- VISUAL BASIC OPERADORES RELACIONALES

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Los operadores relaciónales que reconoce el lenguaje Visual Basic son:

OPERADOR	SIGNIFICADO
”=”	Igual que
”>”	Mayor que
”<”	Menor que
”>=”	Mayor o igual que
”□ ”	Menor o igual que
”<>”	No es igual que o es diferente que.

6.- VISUAL BASIC INSTRUCCIÓN IF

Es la instrucción condicional mas usada en los diversos lenguajes de programación, su formato completo y de trabajo en Visual Basic es :

cargar o asignar la variable de condición

If condición Then

grupo cierto de instrucciones

Else

grupo falso de instrucciones

End If

Si un if no ocupa un grupo falso de instrucciones, entonces no se usa un else, ejemplos:

Se esta usando una caja de mensajes, para evitar usar un segundo TextBox, se deberá analizar el formato completo de MsgBox, porque es muy útil en muchas ocasiones para despliegues de mensajes pequeños.

Salida:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

TAREAS PROGRAMACION VISUAL BASIC

1.- Capturar un número cualesquiera e informar si es o no es mayor de 100

2.- Capturar un numero entero cualesquiera e informar si es o no es múltiplo de 4 (recordar el operador mod, analizado en el tema de operadores aritméticos).

3.- Capturar los cinco datos mas importantes de un Empleado, incluyendo el sueldo diario y los días trabajados esto en un panel, desplegarle su cheque semanal en un segundo panel solo si ganó mas de $500.00 en la semana, en caso contrario desplegarle un bono de despensa semanal de $150.00 en un tercer panel.

4.- Capturar los datos mas importantes de un estudiante incluyendo tres calificaciones, todo esto en una ventana, una segunda ventana que contiene una boleta de calificaciones es llamada si el estudiante es de la carrera de medicina, en caso contrario una tercera ventana despliega un oficio citando a los padres del estudiante a una platica amistosa con los maestros de la escuela.

5.- Capturar los datos mas importantes de un producto cualesquiera, incluyendo cantidad, precio, etc. , desplegar una orden de compra, solo si el producto es de origen nacional, en caso contrario no hacer nada.

7.- VISUAL BASIC CONDICIONES COMPUESTAS

En muchas ocasiones es necesario presentar mas de una condición para su evaluación al computador.

Por ejemplo que el computador muestre la boleta de un alumno, si este estudia la carrera de medicina y su promedio de calificaciones es mayor de 70.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Una condición compuesta se define como dos o mas condiciones simples unidas por los llamados operadores lógicos.

Los operadores lógicos que Visual Basic reconoce son;

OPERADOR	**SIGNIFICADO**
And	y
Or	o
Not	no

Para que el computador evalúe como CIERTA una condición compuesta que contiene el operador lógico “and”, las dos condiciones simples deben ser ciertas.

Para que el computador evalúe como CIERTA una condición compuesta que contiene el operador lógico “or”, basta con que una de las condiciones simples sea cierta.

La cantidad total de casos posibles cuando se unen dos o mas condiciones simples esta dada por la relación 2^n donde n = cantidad de condiciones, la primera mitad de ellos empieza en cierto y la segunda mitad en falso.

Ejemplo, si formamos una condición compuesta con dos condiciones simples y el operador lógico “y”, la cantidad total de casos posibles serian $2^2 = 4$, y se puede construir la siguiente tabla de verdad.

Tabla de verdad con “y”

1ra CS 2da CS Eval

C C C

C F F

F C F

F F F

La evaluación final, se obtiene usando la regla anteriormente descrita para una condición compuesta, que contiene el operador “and”.

Esta tabla significa lo siguiente;

1.- Cualquiera que san la cantidad de datos procesados,, siempre caerá en uno de estos cuatro posibles casos.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

2.- En general cuando se usa una condición compuesta que incluya el "y", el 75% en promedio de la veces que se ejecuté o evalúe la condición, se ejecutara la parte falsa de instrucciones.

La tabla de verdad para una condición compuesta con "O" es la sig.;

1ra Cond2da CondEval

C C C

C F C

F C C

F F F

Como se observa, una condición compuesta con "O", es menos restrictiva, o el 75% de los casos terminarían ejecutando el grupo CIERTO de instrucciones de la instrucción condicional.

Construir una tabla de verdad para una condición compuesta de tres o mas condiciones simples, es también tarea sencilla, solo recordar que;

1.- La cantidad posible de casos es $2^3 = 8$ casos posibles, la mitad empiezan con Cierto y la otra mitad empiezan con Falso.

2.- Para evaluar esta condición triple, primero se evalúan las dos primeras incluyendo su operador, bajo las reglas ya descritas y luego se evalúa, el resultado parcial contra la ultima condición, y ultimo operador, para obtener la evaluación final.

Ejemplo una condición compuesta de tres condiciones simples, donde el primer operador lógico es el "y" y el segundo operador lógico es el "O", daría la siguiente tabla de verdad.

1ra 'y' 2da Eva Parcial 'o' 3ra Eva final

C C c C c

C C c F c

C F f C c

C F f F f

F C f C c

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

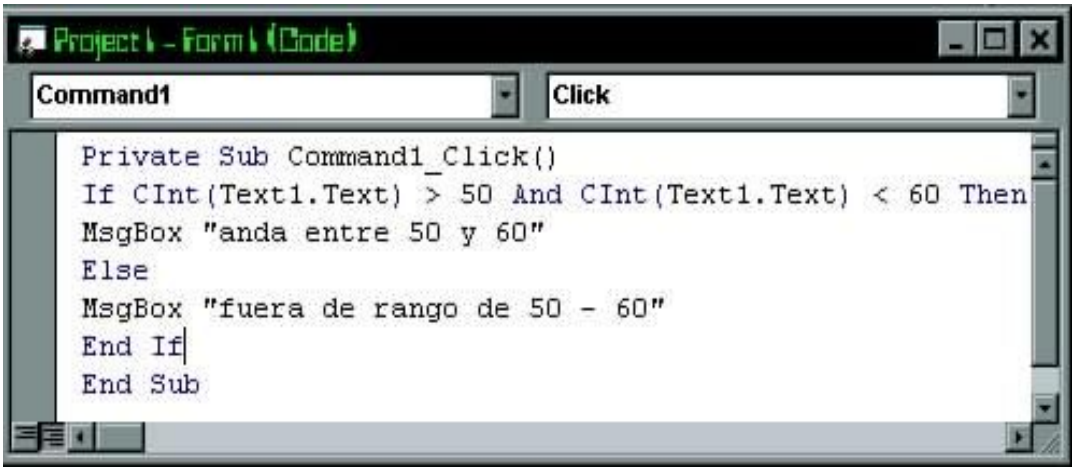
F C f F f

F F f C c

F F f F f

ejemplos de condiciones compuestas:

a)



TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un programa que capture un numero cualesquiera e informe si es o no es mayor de 50 y múltiplo de tres. (solo escribir el mensaje de respuesta de manera muy clara y esto resuelve el problema)

2.- Construir un programa que indique si un numero es un par positivo.

3.- Capturar los datos de un producto incluyendo su cantidad en existencia, construir un panel que despliegue una orden de compra si la cantidad en existencia del producto es menor que el punto de reorden, o si el origen del producto es nacional.

4.- Construir un programa que capture los datos de un empleado, desplegar en una ventana su cheque semanal si gana mas de $500.00 y si esta en el departamento de producción, en caso contrario desplegarle en otra ventana un bono de despensa del 25% de su sueldo semanal.

8.- INSTRUCCIÓN SELECT CASE

También existen ocasiones o programas donde se exige evaluar muchas condiciones a la vez, en estos casos, o se usan una condición compuesta muy grande o se debe

intentar convertir el problema a uno que se pueda resolver usando la instrucción SELECT CASE.

Esta instrucción, es una instrucción de decisión múltiple, donde el compilador prueba o busca el valor contenido en una variable contra una lista de constantes, cuando el computador encuentra el valor de igualdad entre variable y constante, entonces ejecuta el grupo de instrucciones asociados a dicha constante, si no encuentra el valor de igualdad entre variable y constante, entonces ejecuta un grupo de instrucciones asociados a un default, aunque este ultimo es opcional.

El formato de esta instrucción es el siguiente;

capturar o asignar variable de condición;

Select Case variableopcion

case const1 instrucción(es)

case const2 instrucción(es)

case const3 Instrucción(es)

…

Case Else instrucción(es);

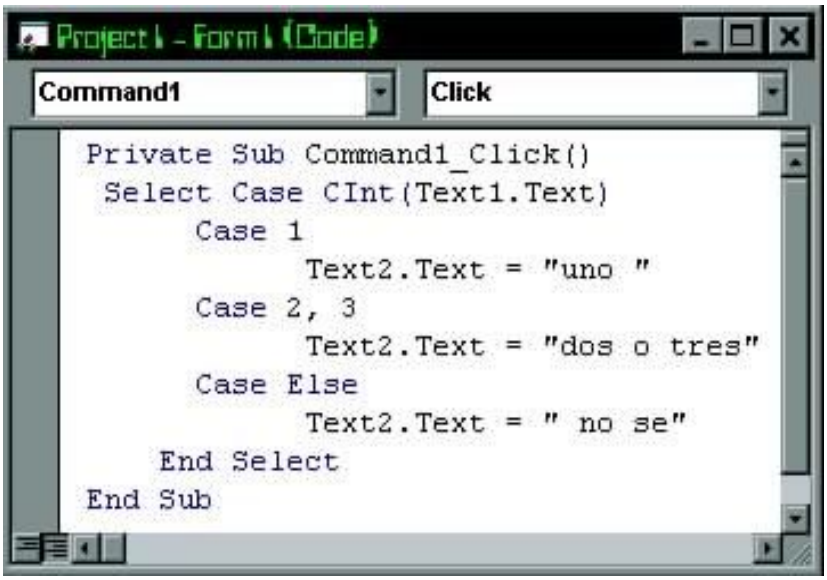
End Select

ejemplo:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Para el caso de constantes numéricas, solo convertir Text1.Text a byte, integer y long, y poner las constantes sin comillas.

ejemplo:



En particular, instrucciones de este tipo se usaban para construir programas de selección de menús, donde al usuario se le planteaban dos o tres problemas distintos y el propio usuario seleccionaba cual de ellos quería ejecutarse.

TAREAS VISUAL BASIC

1.- PROGRAMACION Construir una ventana que contenga el siguiente menú

1. conversión de pesos a dólares 2. conversión de libras a kilogramos 3. conversión de kilómetros a millas 4. fin de menú

Seleccionar opción [ ]

Para resolver este programa, primero diseñar las cuatro formas o ventanas que se ocupan, y en la primera forma que contiene el menú para el usuario, programar el evento clic del botón de ordenes con la instrucción select case, los case solo contienen código para llamar o poner a la vista del usuario la ventana o forma respectiva.

Y además recordar poner en cada ventana de solución de un problema un botón de orden, con código de regreso a la ventana de menú, solo escribir en el Caption de este botón la palabra [menú] y en su evento Clic ocultar la ventana de problema y poner visible la ventana o forma del menú.

2.- Construir un programa que capture un deporte y despliegue dos implementos deportivos apropiados.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

3.- Evaluar cualquier función vista para cuando x =3,-4, 5

9.- VISUAL BASIC COMPONENTES VISUALES DE SELECCIÓN Y DECISIÓN

Las instrucciones if y select, permiten tomar decisiones o realizar seleccionase dentro del código de un programa.

Visual Basic proporciona una serie de componentes visuales que permiten al usuario, no al programador, tomar decisiones en pantalla, el programador solo se encarga de implantar código adecuado a la decisión tomada por el usuario.

10.- COMPONENTE CHECKBOX VISUAL BASIC 

El componente CheckBox de Visual Basic permite seleccionar una opción al usuario del programa o tomar una decisión, directamente en pantalla.

Es la propiedad caption del componente donde se escribe el sentido de la selección ej.;

En los ejemplos, los componentes checkbox, son las cajas donde el usuario toma un decisión (ej. 3) o realiza una selección (ej. 1,2)

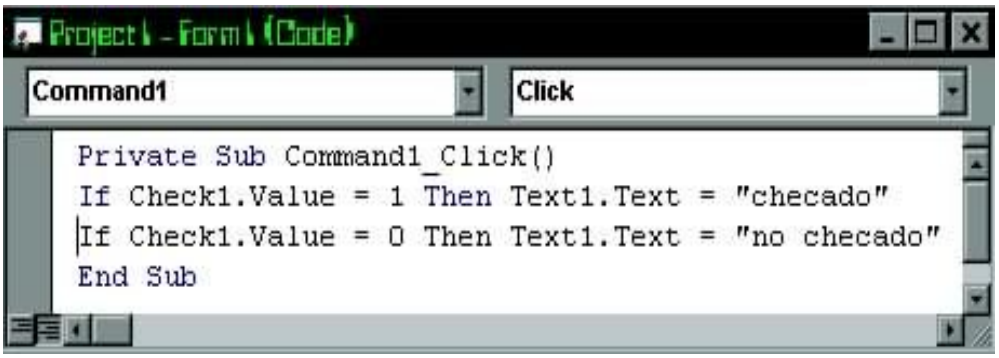
Igual que en controles anteriores, al pasarlo a la forma o al código del programa, ya no se llama checkbox, sino check#

Existen dos maneras de programar este componente:

1. Cuando el usuario selecciona un checkbox la propiedad value queda cargada con 0(cero) si esta deseleccionado o con 1(uno) si fue seleccionado, en estos casos solo

validar con un if por cada CheckBox dentro de nuestro botón de ordenes, el estado de dicha propiedad.

Ej.;



Para el ejemplo c) el botón de ordenes en la forma o ventana respectiva usando el método anterior, contendría 3 ifs, uno para construir boleta otro para construir citatorio y otro para construir un diploma.

2. El segundo método para programar el componente, involucra el evento Click de este componente CheckBox, este evento Click es activado automáticamente en cuanto el usuario realiza o marca o toma su selección, es claro que si no se programa este evento el usuario no observara ningún proceso, sino que tendrá que indicar que ya hizo su decisión, apretando el botón de OK.

Pero si se programa el evento Click de este componente con el código adecuado, ni se tendrá que agregar un botón OK, ni se ocupara usar un if(Checked), porque el usuario ya indico cual es su decisión o selección.

Recordar que para programar este evento Click, solo hacer un dobleclick, dentro del control.

Este método es mejor porque evita código de mas y cada componente solo tiene asociado el código que ocupa.

Aunque es la lógica del programa quien decide como programar el control, en resumen, si es una sola decisión, usar primer método, si son varias decisiones, usar el segundo método.

TAREAS PROGRAMACION VISUAL BASIC

1.- Evaluar la función $y = 3x^2 - 4x + 2$ para x→ 2,-5, 8 (usar un CheckBox por cada valor de x, y programar cada evento Click de cada CheckBox con la operación correspondiente y el despliegue del resultado).

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

2.- Construir un panel con los datos de un automóvil, un segundo panel muestra un plan de financiamiento a dos años y un tercer panel muestra un plan de financiamiento a tres años.(son dos checkbox en el primer panel y no hay botón de ok).

3.- Construir el programa de menú, que se dejo en el tema de instrucción select case.

11.- COMPONENTE OptionButton 

Se utiliza para presentar al usuario un conjunto de opciones mutuamente excluyentes entre si, es decir si el usuario selecciona un componente OptionButton todos los demás componentes OptionButton en la forma, se deseleccionan solos.

Es su propiedad Caption donde se pone el texto que identifica el propósito del botón, es su propiedad value quien refleja el cambio (True,False), también su evento click es activado automáticamente cada vez que es seleccionado el OptionButton por el usuario.

Recordar también que cuando el usuario selecciona un OptionButton, todos los demás OptionButton en el objeto(forma o ventana) son deseleccionados automáticamente, esto es, por que dos OptionButton son mutuamente excluyentes entre si.

Esta ultima situación deberá resolverse por parte del programador, es decir se supone un programa donde el usuario debe seleccionar uno de entre dos sexos y uno de entre cinco municipios, en este caso se ocupan ocho OptionButton, pero como todos son mutuamente excluyentes entre si, cuando el usuario seleccione uno de ellos, todos los demás se des marcan automáticamente.

Para resolver este problema se deberá usar el componente de agrupamiento ampliamente conocido, el control FRAME.

Es decir se deberá encerrar en su propio Frame todos los OptionButton lógicos, es decir en un Panel los de sexo, en otro Panel los de municipios, etc.

De esta manera Visual Basic los evalúa por separado y se puede tener seleccionado un OptionButton en cada frame.

También pueden programarse de las dos maneras ya vistas para el control CheckBox, es decir usando un if por cada optionbutton y revisar si su propiedad value esta cargada con TRUE o FALSE, o cargando el código en el evento click del control.

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un cuestionario de 6 preguntas sobre los hábitos de estudio de un estudiante y pasar sus respuestas a un frame abajo en textboxs.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

2.- Construir un cuestionario de 5 preguntas con las preferencias políticas de una persona, un panel abajo despliega un concentrado con totales y porcentajes acumulados por cada pregunta.

Es decir un usuario responde el cuestionario aprieta el botón de ok, el panel de abajo se actualiza para mostrar totales y porcientos de cada pregunta, y se limpian las respuestas, un segundo usuario responde el cuestionario…. y así sucesivamente.

* Para totales y porcentajes, solo recordar dos de los conceptos más elementales de introduccion a la programación

* Contadores cont=cont+1 ? resultado 1, 2, 3,4,5,……. * acumuladores acum = acum + cont resultado con el ejemplo de arriba

1,3,4,10,15,.....

12.- MENUS VISUALES EN VISUAL BASIC

Ya se han construido programas de selección de menús, usando instrucciones de programación (SELECT CASE) y componentes visuales (CheckBox con su componente Click programado).

Se analiza ahora los componentes visuales especializados en la construcción de programas de selección de menús.

13.- MENÚ VISUALES O EDITOR DE MENÚS

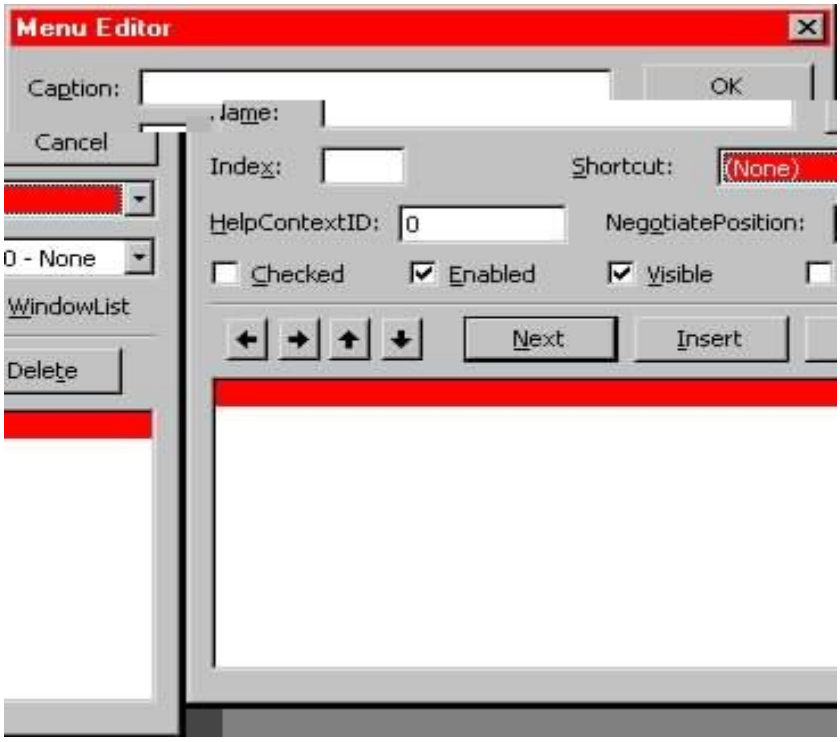
A diferencia de otros lenguajes visuales, no existen controles de default, para la creación de menús visuales, en su lugar se usa un editor de menús.

Procedimiento para la creación de menús:

1.- Cargar el editor de menús, con la orden Tools, Menú Editor, y aparece la siguiente ventana: 1.- Cargar el editor de menús, con la orden Tools, Menú Editor, y aparece la siguiente ventana:<a name

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



Con este editor se construyen, las barras de menú en los programas normales de Windows.

Se construye una aplicación paso a paso, que tiene tres opciones, ellas son, geometría con triángulos y rectángulos, casa cambio con pesos a dólares y dólares a pesos y una opción final de salir.

Del editor de menú, se usan primero las siguientes partes:

CAPTION: Aquí se escriben el nombre de las opciones y subopciones, tales como las observara el usuario.

NAME: Un alias legal para cada opción y subopcion, en este caso se usan como alias, las palabras uno, unouno, unodos, dos, dosuno, dosdos, tres, aunque tambien puede ser el CAPTION pero en forma mas completa y entendible.

Estos alias corresponden a las opciones y subopciones del codigo que tendrá el menú.

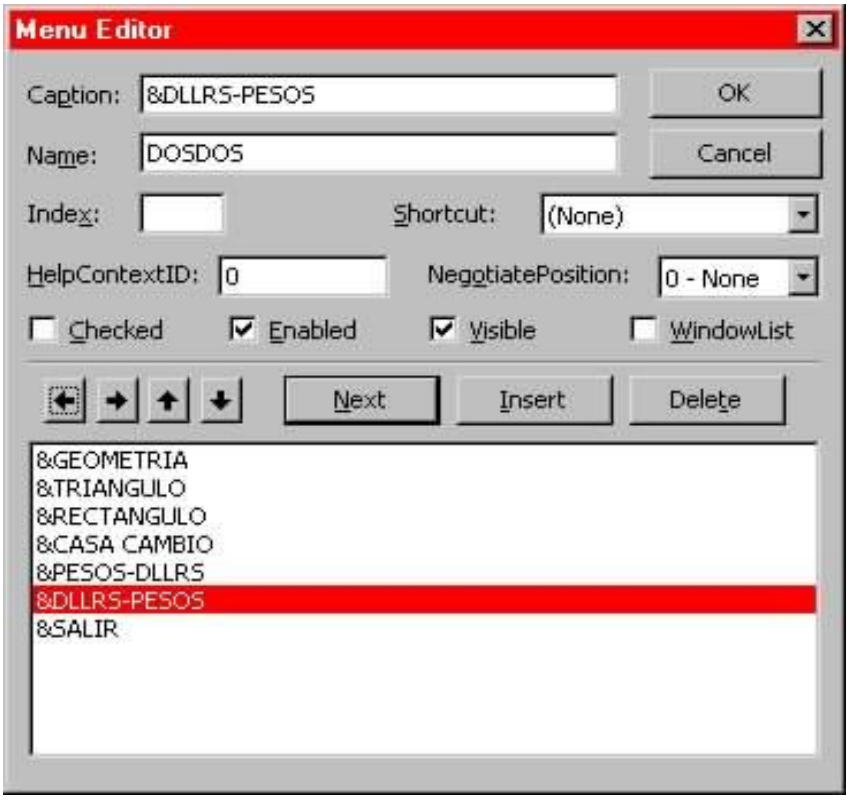
SHORTCUT: Primero es opcional poner un shortcut a cada opción o subopcion del programa, en el editor esta parte es un combobox que ya trae cargado muchos shorcuts, un shortcut es una combinación de la tecla CTRL-LETRA

LISTBOX MENÚ: Es la ventana de abajo completamente en blanco, es en esta ventana donde irán apareciendo las opciones y subopciones(NAME) correspondientes.

Ya escrito el Caption, Name Y Shorcut de una opcion o subopcion correspondiente se usan los botones NEXT, INSERT y DELETE para administrarlos en LISTBOX MENU

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

2.- Se empieza cargando las opciones y subopciones correspondientes, observar la ventana del editor de menús.



El símbolo (&) antes de una opción y subopcion, es para que se generen las llamadas (HOT-KEYS), es decir que el usuario pueda usar la tecla ALT-Primera letra de la opción.

No confundir HOT-KEY con shortcut, por ejemplo para activar triángulo su hot-key es ALT-T y un shorcut apropiado, seria CTRL-T

Las opciones y subopciones se pueden escribir en cualquier orden, porque seleccionando una opción (en el ejemplo esta seleccionada con click la opción Dllrs-Pesos) y usando los controles con las flechas de up y down (no las del teclado sino las del editor de menu) se pueden reacomodar opciones y subopciones.

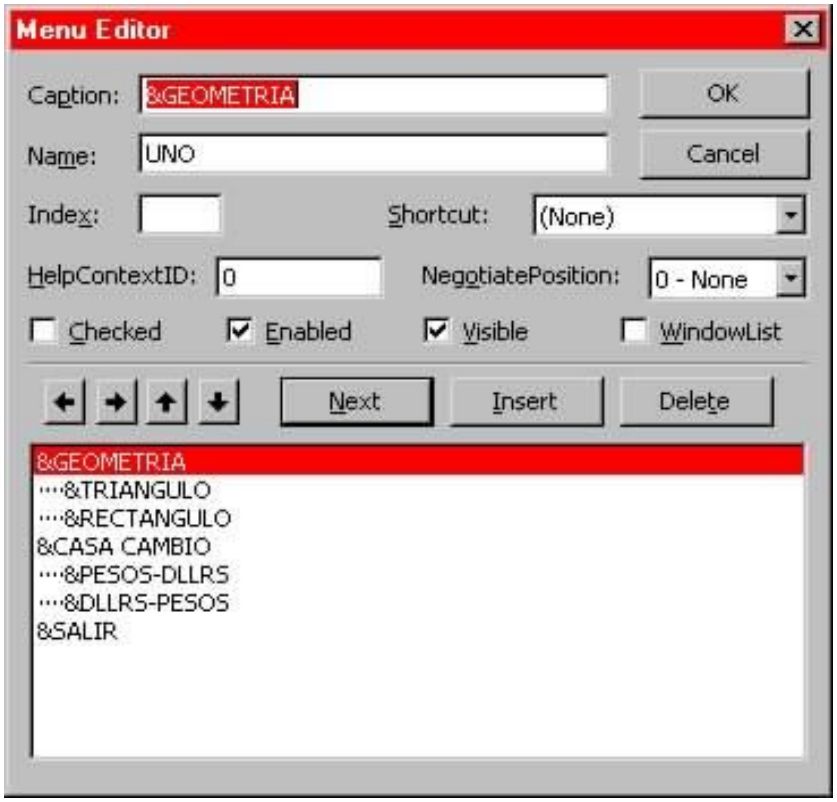
Observar que si se selecciona una opción o subopcion y se usa el control delete, esta se eliminara.

3.- Ahora lo mas importante, en este ejemplo, tanto las opciones y subopciones están al mismo nivel, si se ejecuta el programa, aparecerá un menú con siete opciones, no lo que se quiere, es decir tres opciones con sus subopciones correspondientes.

Lo que se tiene que hacer, es darles un nivel mas bajo a las subopciones, solo seleccionando con click la subopcion y usando el control con flecha (è)del editor de menus, esta flecha baja el nivel de la subopcion, la otra flecha (ß)lo sube un nivel.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Observar ahora la pantalla del editor de menú, con las opciones en nivel y orden correcto.

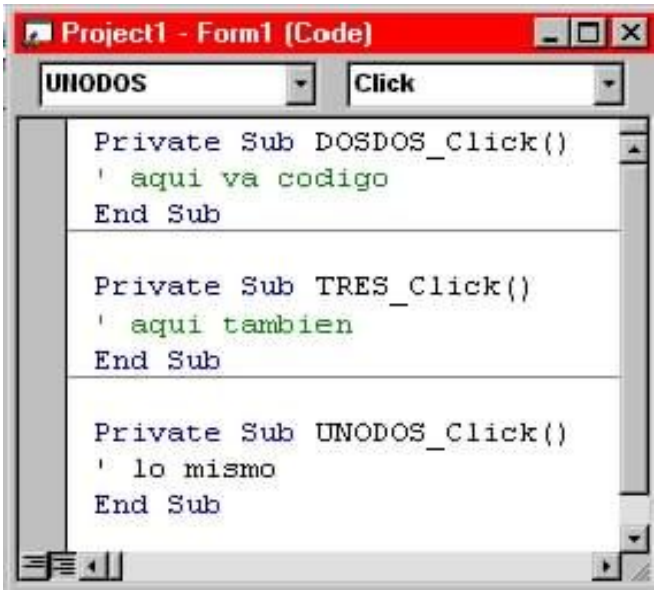


4.- Cerrar el editor de menú con OK, y observar ahora como form1, ya contiene su nuevo menú principal.

5.- Por ultimo, falta lo principal, el cargarle el código a cada opción, en este caso, es poner visibles o invisibles otras ventanas.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

El procedimiento es sencillo, solo hacer un click en cada opción o subopcion que lleve código aqui en Form1, observar que el editor de código, usa el NAME del editor de menús.



6.- Ejecutar el programa con la opción Start

TAREAS PROGRAMACION VISUAL BASIC

1.- El de la muestra con dos conversiones monetarias, dos opciones de temperaturas y dos de distancias.

* Primero diseñar y construir todas las formas que va a ocupar * no olvide poner algunas labels de encabezado en la ventana principal

En el botón ok o en otro botón similar de cada ventana de trabajo no olvidar agregar código para ocultar dicha ventana y regresar a la ventana con el menú principal.

14.- POPUPMENU

PopupMenus, son los pequeños menús que aparecen al hacer un click derecho sobre algún componente o alguna parte de la forma.

Visual BASIC facilita la construcción de estos popupmenus, solo recordar y tomar en cuenta lo siguiente.

Un popupmenu deberá pegarse a cualquier otro control, pero algunos de los controles generales ya traen incorporado por default su propio popupmenu, por ejemplo TextBox, ya trae su propio popupmenu, así que no será posible pegarle un segundo popupmenu, así que será necesario hacer pruebas primero, es decir en una forma poner los 20 controles de default, ejecutar o correr el programa, y hacer un clic

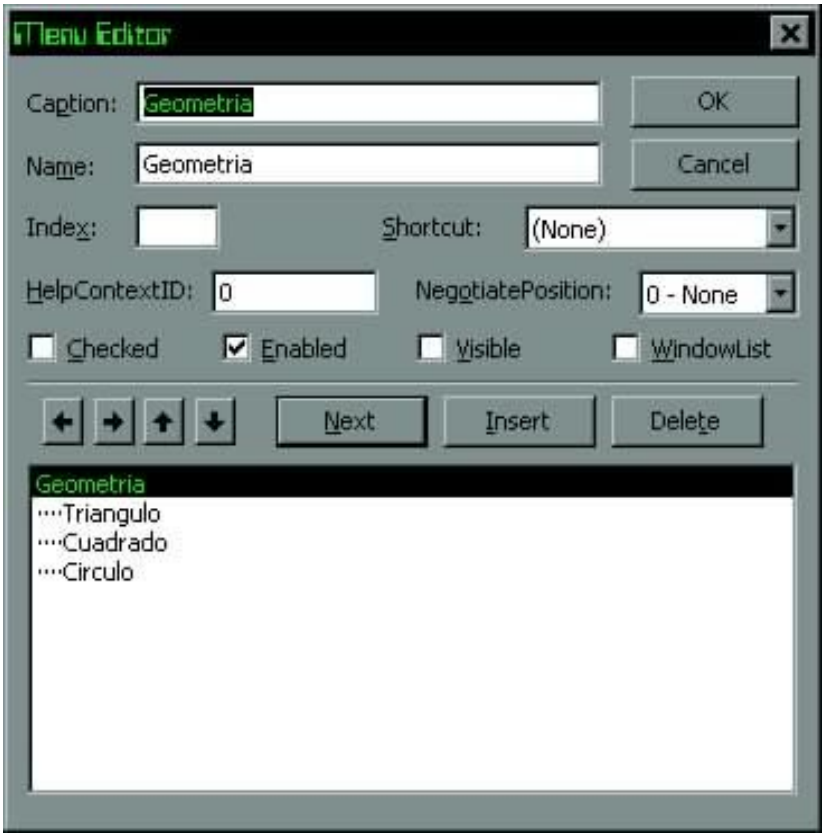
Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

derecho sobre cada control, para investigar cuales son los que ya traen su propio popupmenu, a estos controles no pegarles nuestros popupmenus.

Para crear y pegar popumenus, solo seguir el siguiente procedimiento.

1.- Usar el editor de menús para crear un menú, con solo una opción y todas las subopciones que se ocupen.

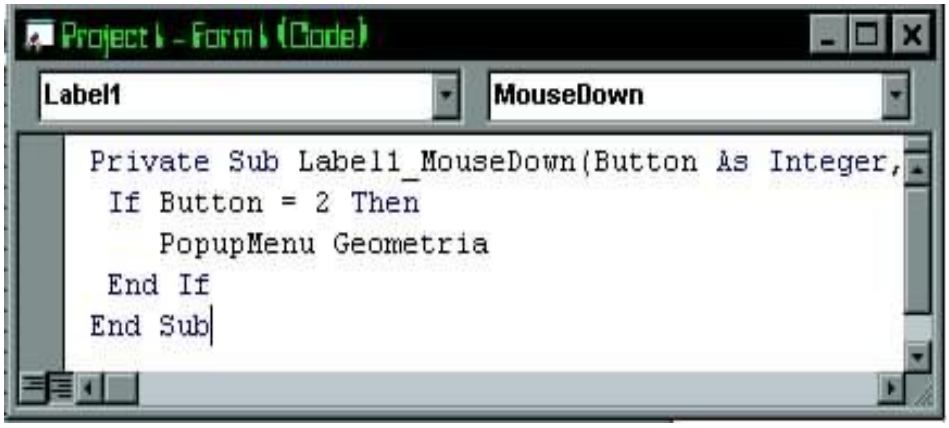
2.- La opción, deberá tener su propiedad visible en false(solo desmarcar el cuadrito visible), ejemplo



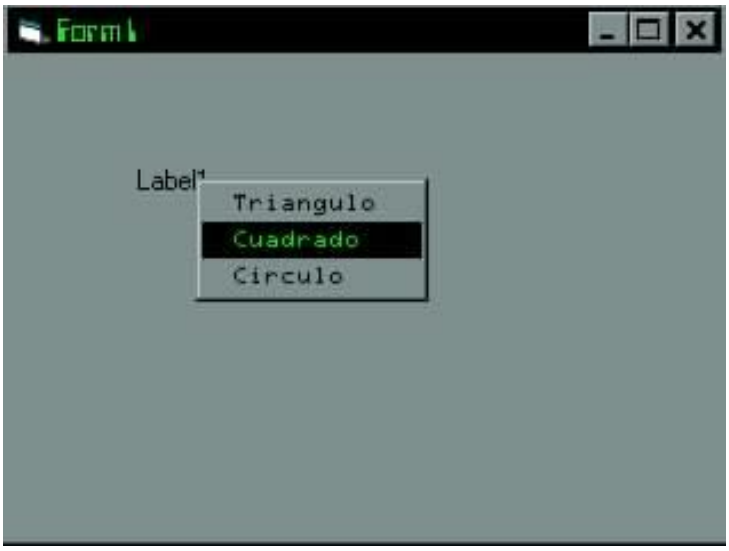
3.- Para pegarle código a las subopciones, solo llamar al editor de código, y en las ventanilla de objetos, se encontraran las subopciones, seleccionar la adecuada y en la ventanilla de eventos, seleccionar el evento clic y colgarle el código adecuado.

4.- Ya construido el popumenu, se lo pegamos a un control label, solo poner un control label en la forma y usar su evento MouseDown con el siguiente código:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



5.-Primero se investiga si el usuario apretó el segundo botón del Mouse (Microsoft cree que los ratones solo tienen dos botones :>)) y luego se usa el método popupmenu y el nombre de la primera opción para activar el popumenu.



TAREA PROGRAMACION VISUAL BASIC

1.- Construir con visual basic el programa ejemplo

15.- CICLO FOR VISUAL BASIC

Este ciclo es uno de los más usados para repetir una secuencia de instrucciones, sobre todo cuando se conoce la cantidad exacta de veces que se quiere que se ejecute una instrucción simple o compuesta.

Su formato general es:

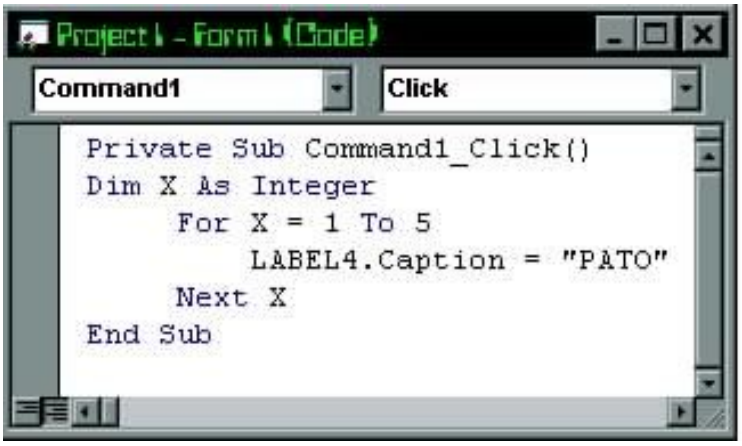
FOR VARCICLO=VALORINICIAL TO VALORFINAL [STEP INCR O DECR]

INSTRUCCION(ES)

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

NETX VARCICLO

EJEMPLO



Como se observa, instrucciones de ciclo, si ocuparan que se declaren variables de control de ciclo, ya sea de tipo byte,integer o long.

Casos Particulares;

1.- El ciclo comienza en uno y se incrementa de uno en uno, este es el caso mas general.

2.- Pero el valor inicial puede se diferente de uno, ejemplo;

DIM X AS INTEGER

FOR X=5 TO 28

LABEL4.CAPTION=X

NEXT X

3.- Incluso el valor inicial puede ser negativo, ejemplo;

DIM X AS INTEGER

FOR X= -5 TO 18

LABEL4.CAPTION=X

NEXT X

4.- Los incrementos también pueden ser diferentes al de uno en uno, ej.;

DIM X AS INTEGER

```
FOR X=1 TO 50 STEP 4

LABEL4.CAPTION=X

NEXT X
```

5.- Incluso pueden ser decrementos, solo que en este caso, recordar;

* el valor inicial de la variable debe ser mayor que el valor final.

```
DIM X AS INTEGER

FOR X=100 TO 20 STEP - 5

LABEL4.CAPTION = X

NEXT X
```

6.- Un ejemplo para usarlo en los problemas sugeridos mas adelante;

ejemplo;

Desplegar los números enteros, comprendidos entre el 1 y el 20.

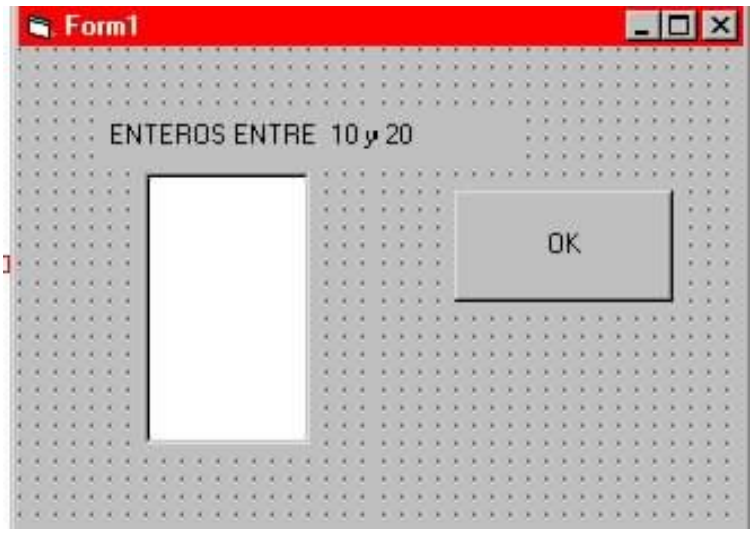
Se ocupa ahora un componente que pueda almacenar y desplegar un conjunto de los 10 resultados, el único componente visto hasta ahora con esta capacidad es el componente ComboBox, sin embargo existe otro componente llamado ListBox muy similar a ComboBox, excepto que no tiene encabezado y todos sus elementos los mantiene a la vista del usuario, no ocultos como el ComboBox, dicho componente ListBox se analiza a fondo en la siguiente UNIDAD VISUAL BASIC , pero de momento permite resolver el problema del for (desplegar un conjunto de resultados a la vez).

Tanto ComboBox como ListBox permiten cargar todos sus elementos o valores, dentro de un programa, usando un método llamado AddItem(valor), como se ve en el siguiente programa ejemplo;

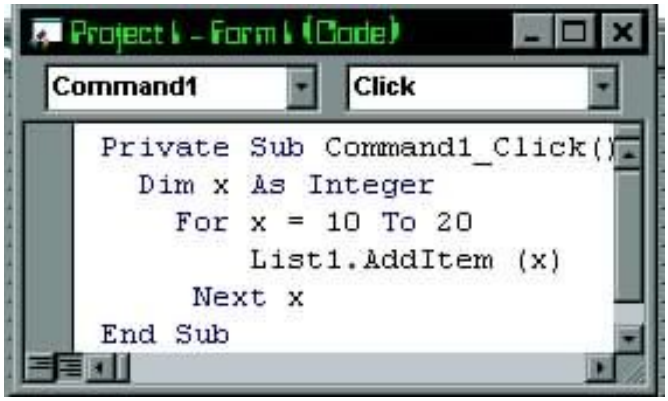
Para este problema se ocupa poner en Form1, un componente Command1 OK que en su evento Click contiene el for y la carga del componente ListBox; Pantalla de diseño

Lic. Jose Luis Dominguez C.

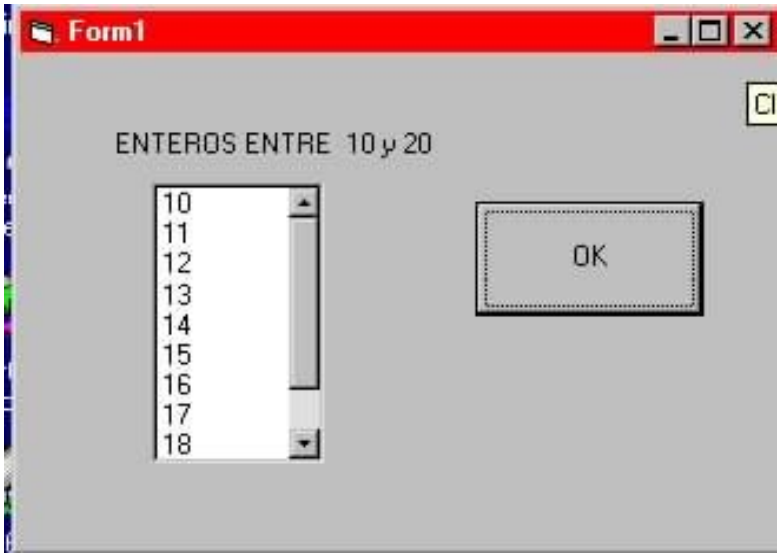
Joseluisdc10@yahoo.es



B) Programa



la pantalla de salida es:



Este procedimiento y método igualmente trabaja con un componente ComboBox.

Practicar hasta conseguir tener esta pantalla de salida o de ejecución, cuando se consiga entonces ya se esta listo para lo siguiente;

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un programa que despliegue los números del 20 al 30.

2.- Desplegar los enteros entre 50 y 30 acompañados de su potencia cuadrada y raíz cubica respectiva (ocupa tres listbox).

3.- Desplegar los múltiplos de 5, entre 10 y 50, acompañados de su factorial y logaritmo respectivo.

4.- Desplegar la tabla de multiplicar que el usuario indique.

5.- Evaluar la funcion <m> y=5x^2 + 3x + 8 </m> cuando x→ -3…10 (rango de -3 hasta 10)

16.- VISUAL BASIC CICLO DO WHILE LOOP

En este ciclo el cuerpo de instrucciones se ejecuta mientras una condición permanezca como verdadera, en el momento en que la condición se convierte en falsa el ciclo termina.

Su formato general es :

cargar o inicializar variable de condición

DO WHILE CONDICION(ES)

INSTRUCCION(ES) CIERTAS

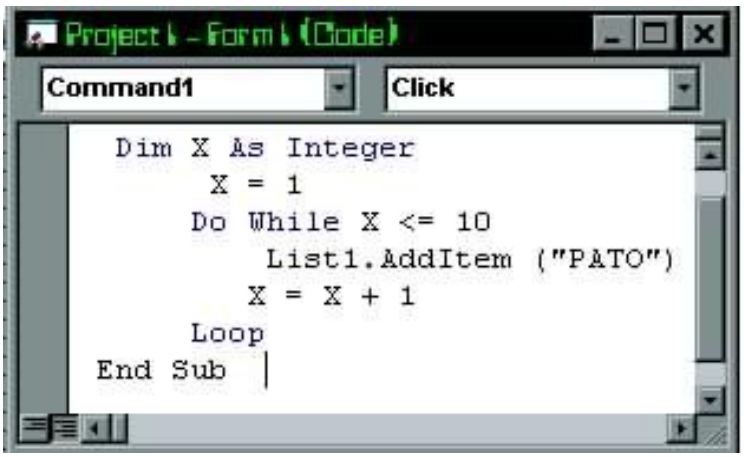
INSTRUCCION(ES) DE SALIDA DE CICLO

LOOP

Ejemplo #1 :

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



DO While puede llevar dos condiciones, en este caso inicial izar 2 variables de condición y cuidar que existan 2 de rompimiento de ciclo.

El grupo cierto de instrucciones puede ser una sola instrucción o todo un grupo de instrucciones.

La condición puede ser simple o compuesta.

A este ciclo también se le conoce también como ciclo de condición de entrada, o prueba por arriba, porque este ciclo evalúa primero la condición y posteriormente ejecuta las instrucciones.

TAREAS PROGRAMACION VISUAL BASIC

1.- Usando visual basic para desplegar enteros entre 50 y 80

2.- Usar visual basic para desplegar múltiplos de 4 entre 60 y 20 acompañados de su logaritmos de base 10 y base e respectivos (a revisar funciones visual basic)

3.- Construir la tabla de dividir que el usuario indique

4.-Evaluar la funcion <m>y = -5x^2 + 4x - 20</m> cuando x→ -5… 5

17.- VISUAL BASIC CICLO DO LOOP WHILE

Su diferencia básica con el ciclo while es que la prueba de condición es hecha al finalizar el ciclo, es decir las instrucciones se ejecutan cuando menos una vez, porque primero ejecuta las instrucciones y al final evalúa la condición.

También se le conoce por esta razón como ciclo de condición de salida.

Su formato general es :

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

cargar o inicializar variable de condición

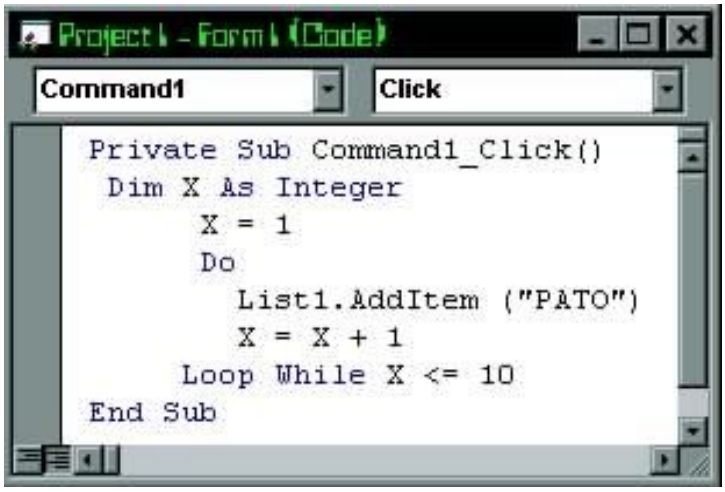
DO

INSTRUCCION(ES) CIERTAS

INSTRUCCION(ES) DE SALIDA DE CICLO

LOOP WHILE CONDICION(ES)

Ejemplo



Otra diferencia básica con el ciclo while es que, aunque la condición sea falsa desde un principio, el cuerpo de instrucciones se ejecutara por lo menos una vez.

TAREAS PROGRAMACION VISUAL BASIC

1.- dos del for

2.- dos de do while loop

18.- VISUAL BASIC CONCLUSIONES ACERCA DE CICLOS

En Visual Basic el problema de dado un problema cualesquiera, cual ciclo se debe usar se resuelve con:

Si se conoce la cantidad exacta de veces que se quiere que se ejecute el ciclo o si el programa de alguna manera puede calcularla usar for.

Si se desconoce la cantidad de veces a repetir el ciclo o se quiere mayor control sobre la salida o terminacion del mismo entonces usar do while loop.

Si se quiere que al menos una vez se ejecute el ciclo entonces usar do loop while.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

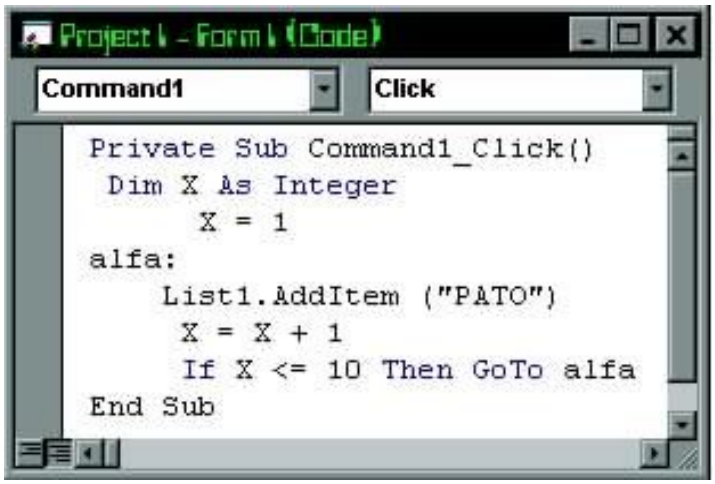
19.- INSTRUCCIÓN DE SALTO INCONDICIONAL, ETIQUETAS Y GOTO

Esta instrucción fue una de las favoritas de la comUNIDAD VISUAL BASIC de programadores hace 20 años, en la actualidad la aparición de nuevas estructuras o instrucciones de programación hacen innecesario su uso.

El uso principal que se le dio, acompañada de una instrucción if fue la de simular ciclos condicionales.

Esta instrucción requiere una llamada etiqueta que es un identificador valido del lenguaje VB seguida de dos puntos.

Ejemplo



TAREAS VISUAL BASIC

1.- Con programacion visual basic construir las tablas de sumar, y multiplicar con if-goto

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

UNIDAD III

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

III.- ARREGLOS

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

VISUAL BASIC III UNIDAD VISUAL BASIC ARREGLOS

1.- INTRODUCCIÓN

Uno de los problemas mas comunes en los diversos sistemas de información, es el tratamiento o procesamiento de una gran volumen de datos o de información.

Variables o componentes visuales manejados hasta ahora, no pueden ayudar a resolver este problema.

Las variables usadas hasta ahora reciben propiamente el nombre de variables escalares, porque solo permiten almacenar o procesar un dato a la vez.

No confundir esto, con el tipo de dato o rango de datos que una variable tiene la capacidad de almacenar.

Por ejemplo si se quiere almacenar nombre y edad de 15 personas, con el método tradicional se ocuparan 30 variables o 30 componentes visuales, y solo es nombre y edad de 15 personas, agregar mas datos y mas personas y ya es tiempo de empezar a analizar otro tipo de variables y de componentes.

Es decir, en problemas que exigen manejar mucha información o datos a la vez, variables escalares o componentes visuales de manipulación de datos normales (text, label, etc.), no son suficientes, ya que su principal problema es que solo permiten almacenar un dato a la vez.

Se ocupa entonces variables o sus correspondientes componentes visuales que sean capaces de almacenar y manipular conjuntos de datos a la vez.

Variables de tipo arreglo y sus correspondientes componentes visuales, si permiten almacenar y procesar conjuntos de datos del mismo tipo a la vez.

Cada dato dentro del arreglo, se llama elemento del arreglo y se simboliza y procesa (captura ,operación ,despliegue), usando el nombre del arreglo respectivo y un subíndice indicando la posición relativa del elemento con respecto a los demás elementos del arreglo, ej:

Juan

Pedro——> Nombres(2)

José

Ana———> Nombres(4)

Carmen

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

18—–> Edad(1)

20

25

30—→ Edad(4)

Sin embargo sus problemas son similares a los de variables normales, es decir hay que declararlos, capturarlos, hacer operaciones con ellos, desplegarlos, compararlos, etc.

Para propósitos del aprendizaje se analiza o clasifican en tres grupos diferentes los arreglos que ofrece Visual Basic, ellos son;

1.- Arreglos tradicionales (internos dentro del programa) 2.- Arreglos dinámicos (internos) 3.- Componentes Visuales de tipo Arreglo

2.- ARREGLOS TRADICIONALES VISUAL BASIC

En programación tradicional siempre se manejan dos tipos de arreglos, los arreglos tipo listas, vectores o unidimensionales y los arreglos tipo tablas, cuadros, concentrados, matrices o bidimensionales, en ambos casos son variables que permiten almacenar un conjunto de datos del mismo tipo a la vez, su diferencia es en la cantidad de columnas que cada uno de estos tipos contiene, como en los siguientes ejemplos;

1. ARREGLOS TIPO LISTA <li style

Juan
Pedro---->Nombres(2)
José
Ana-----> Nombres(4)
Carmen

18----> Edad(1)
20
25
30----> Edad(4)

2. ARREGLOS TIPO TABLAS

CIA ACME

INGRESOS POR VENTAS

(MILES DE $)

FEB MAR ABR MAY

SUC A 10 12 15 10 9

SUC B 8 7 5 9 6

SUC C 11 18 20 14 17

INST TECN DE TIJUANA

CONCENTRADO DE CALIF

MAT FIS QUIM HIST

JUAN 5 5 5 5

JOSE 4 4 4 4

PEDRO 3 3 3 3

ANA 9 9 9 9

Como se observa, la diferencia principal entre un arreglo tipo lista, y un arreglo tipo tabla, son las cantidades de columnas que contienen.

NOTA IMPORTANTE.- Los conceptos manejados aquí estan enfocados a los sistemas de información contables financieros administrativos.

En álgebra matricial, si son importantes los conceptos de vectores y matrices, pero las operaciones y métodos son precisamente los del álgebra matricial y no es propósito o lugar indicado para tratar problemas y operaciones con matrices.

3.- ARREGLOS TIPO LISTA

Un arreglo tipo lista se define como una variable que permite almacenar un conjunto de datos del mismo tipo organizados en una sola columna y uno o mas renglones.

También reciben el nombre de vectores en álgebra, o arreglos unidimensionales en programación.

Los procesos normales con una lista o con sus elementos, incluyen declarar toda la lista, capturar sus elementos, desplegarlos, realizar operaciones con ellos, desplegarlos, etc.

Para declarar una lista se usa el siguiente formato;

DIM nomlista(1(uno) TO Cant elementos o reng) AS tipo dato

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

ejemplos;

DIM EDAD(1 TO 12) AS INTEGER

DIM SUELDOS(1 TO 10) AS SINGLE

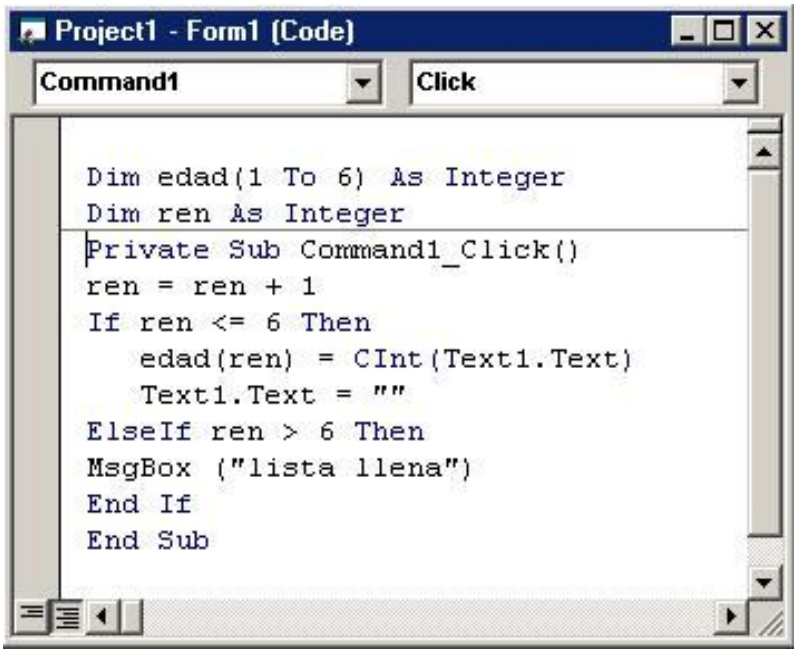
DIM MUNICIPIOS(1 TO 5) AS STRING * 20

Notas:

Declaración.- Es necesario recordar, que la declaración de un arreglo tipo lista se puede hacer de dos maneras diferentes, dependiendo de si solo se usa un botón de órdenes en la pantalla, o si dos o mas botones de ordenes estarán procesando el arreglo, el segundo caso, es el mas común.

Si un solo botón, en toda la ventana va a realizar, todos los procesos (declaración, captura, operaciones, comparaciones, despliegue), con la lista, solo hacer la declaración de la lista, al principio del evento click, como lo muestra el programa ejemplo.

Para capturar se debera usar un textbox y un boton de comando con el siguiente codigo que estara alimentando la lista en memoria: <ol type Programa



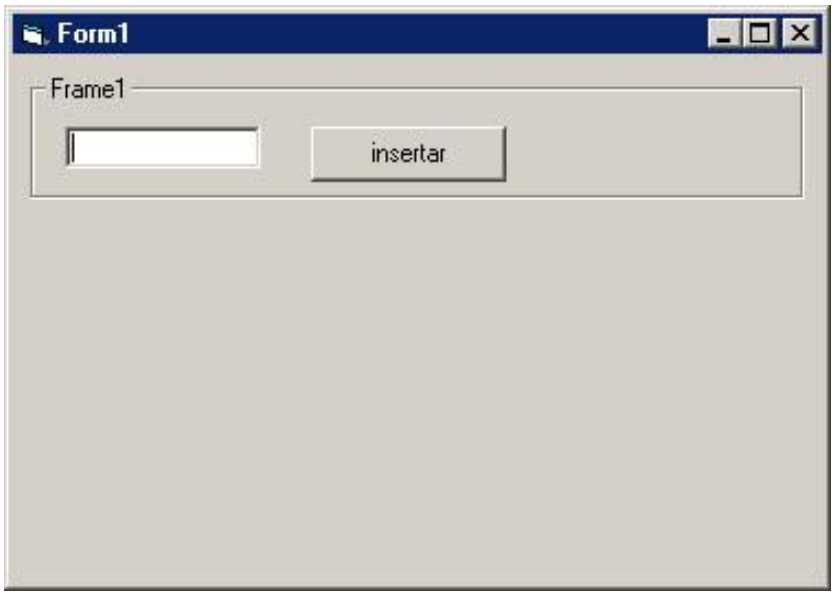
Observar que la declaracion de la lista y la variable de control va fuera del click del boton, para que todos los botones de comando que se pongan en el programa las puedan usar.

Se usa un if para evitar que se capturen datos de mas, y un segundo if con un mensagebox para avisar que ya se lleno la lista.

Lic. Jose Luis Dominguez C.

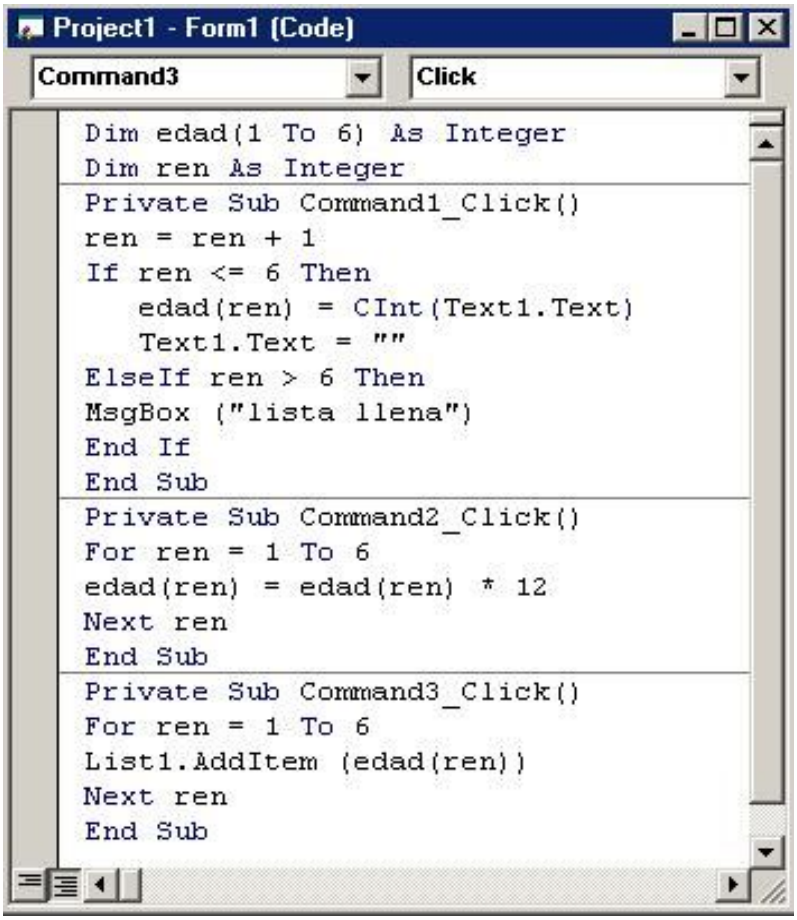
Joseluisdc10@yahoo.es

Pantalla de corrida



Para el caso de operaciones y comparaciones con todos los elementos de la lista a la vez, se deberá usar un ciclo for, con una variable entera llamada renglón, misma que también se usara como índice de la lista, el despliegue de la lista usara un control Listbox y el método AddItem(), para este ejemplo se pretende convertir las edades a meses:

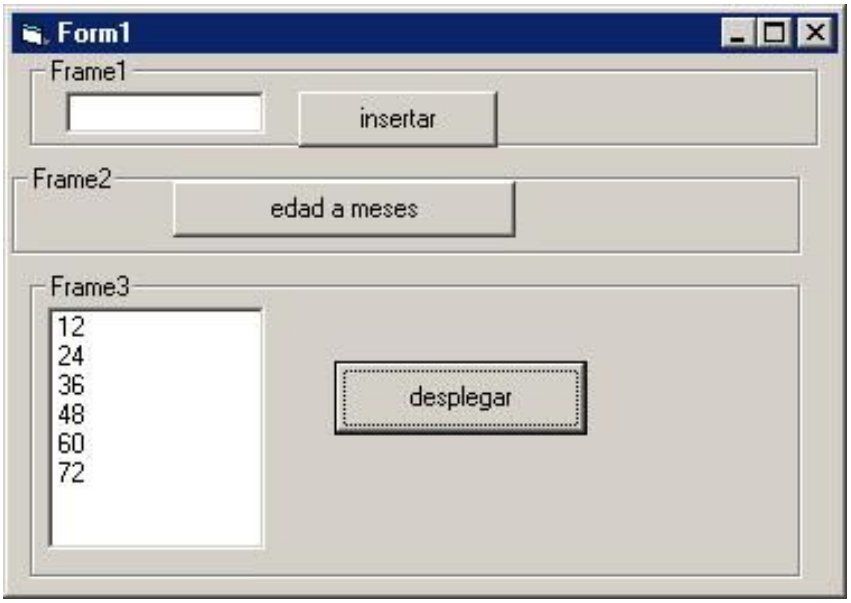
Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Recordar que todos los datos internos de la lista estarán almacenados en la memoria ram del computador, para desplegados se usara un componente visual que permite manipular un conjunto de datos a la vez, el ListBox, pero se tiene que usar un ciclo for para ir añadiendo o agregando elemento por elemento como se observa en el problema ejemplo que se ha venido desarrollando, en este caso se quiere desplegar las cuatro edades convertidas a meses;

B) Pantalla de salida:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



TAREAS PROGRAMACION VISUAL BASIC

1.- Capturar y desplegar 5 precios de productos cualesquiera, usando dos frames, uno para capturar y uno para desplegar.

2.- Capturar 4 sueldos en un panel, desplegarlos aumentados en un 25% en otro panel.

3.- Capturar los datos de 5 productos comprados en una tienda, incluyendo nombre, precio y cantidad en sus 3 listas respectivas, después calcular una cuarta lista con el gasto total por cada producto desplegarlo todo en un segundo panel e incluir también el gran total.

4.- Capturar en una lista solamente 6 números múltiplos de 5, se debe de estar capture y capture números hasta que se completen los 6 múltiplos de 5.

4.- VISUAL BASIC SORTEOS U ORDENAMIENTOS

Un proceso muy común con listas, es el llamado sorteo u ordenamiento.

Este proceso consiste en reacomodar los elementos de la lista en un nuevo orden, de acuerdo a algún criterio.

Sorteo creciente y decreciente

Existen muchos métodos u algoritmos de sorteos, el mas común de ellos, es el denominado algoritmo de burbuja, que se basa en el siguiente algoritmo:

N=CANTIDAD DE ELEMENTOS DE LA LISTA

FOR K = 1 TO N-1

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

```
RENGLÓN = 1

DO WHILE RENGLÓN  N - K

IF LISTA(RENGLON) > LISTA(RENGLON + 1) THEN

TEMP = LISTA(RENGLON)

LISTA(RENGLON)=LISTA(RENGLON + 1)

LISTA(RENGLON + 1) = TEMP

END IF

RENGLÓN = RENGLÓN + 1

LOOP

NEXT K
```

Las notas a considerar con respecto al algoritmo son:

- Las variables n, k, renglón, son variables de control y deberán ser declaradas de tipo integer.

- La variable temp, deberá ser declarada de acuerdo al tipo de dato de los elementos de la lista.

- Todas las referencias a LISTA, deberán ser cambiadas por el nombre verdadero de la lista real.

- Es el símbolo del if, quien determina el tipo de sorteo, es decir, (>)ascendente, (<) descendente.

ejemplo, ordenar 6 números cualesquiera:

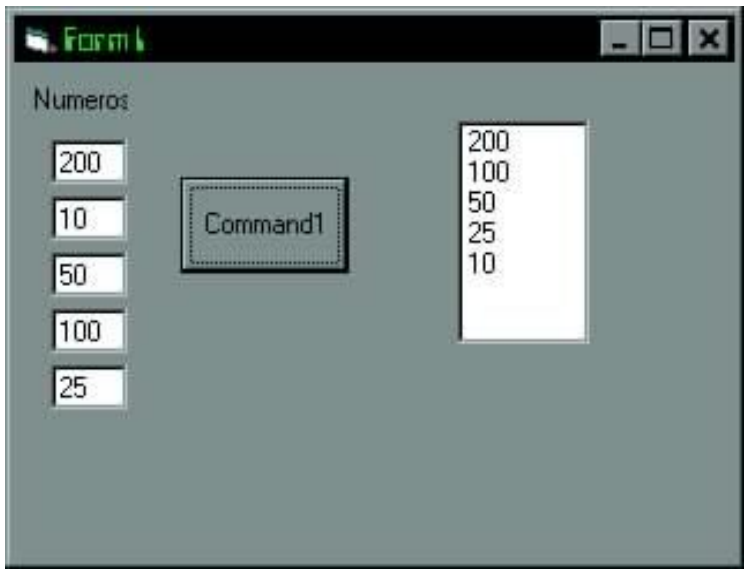
A) Código:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

```
Command1                                   Click
Private Sub Command1_Click()
Dim Num(1 To 5), n, k, reng, temp As Integ
Num(1) = CInt(Text1.Text)
Num(2) = CInt(Text2.Text)
Num(3) = CInt(Text3.Text)
Num(4) = CInt(Text4.Text)
Num(5) = CInt(Text5.Text)
Rem METODO DE BURBUJA
 n = 5
 For k = 1 To n - 1
     reng = 1
     Do While reng <= n - k
        If Num(reng) < Num(reng + 1) Then
           temp = Num(reng)
           Num(reng) = Num(reng + 1)
           Num(reng + 1) = temp
        End If
        reng = reng + 1
     Loop
  Next k
  Rem cargando listbox
  For x = 1 To 5
      List1.AddItem (Num(x))
  Next x
  End Sub
```

B) corrida:



TAREAS VISUAL BASIC

1.- Programacion visual basic ordenar ascendentemente 5 matriculas

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

2.- Programacion visual basic ordenar descendentemente 6 ciudades

3.- Programacion visual basic ordenar a criterio del usuario 7 animalitos

5.- ARREGLOS TIPO TABLA

Un arreglo tipo tabla se define como un conjunto de datos del mismo tipo organizados en dos o mas columnas y uno o mas renglones.

Para declarar un arreglo tipo tabla se usa el siguiente formato:

DIM NOMTABLA(1 TO CANTRENG, 1 TO CANTCOL) AS TIPODATO

EJ:

DIM VTAS(1 TO 3, 1 TO 5) AS SINGLE

DIM CALIF(1 TO 30, 1 TO 6) AS INTEGER

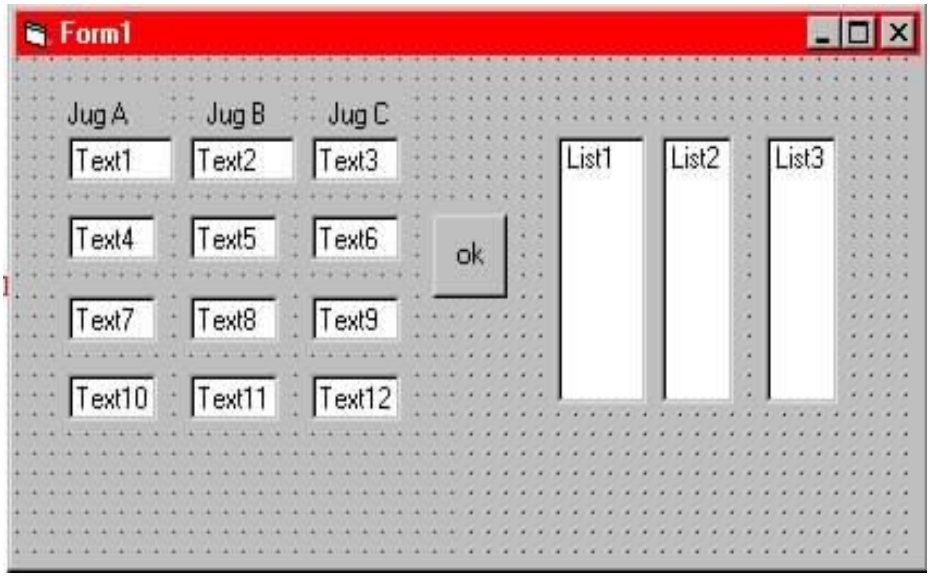
Solo recordar que en capturas, se deberán usar tantos componentes Text como celdas tenga la tabla y en despliegue usar tantos controles ListBox como columnas tenga la tabla, estos métodos son provisionales mientras se analizan los componentes visuales apropiados y respectivos.

Para procesar (recordar solo operaciones y comparaciones) internamente todos los elementos de la tabla se ocupan dos ciclos for, uno externo para controlar renglón y uno interno para controlar columna.

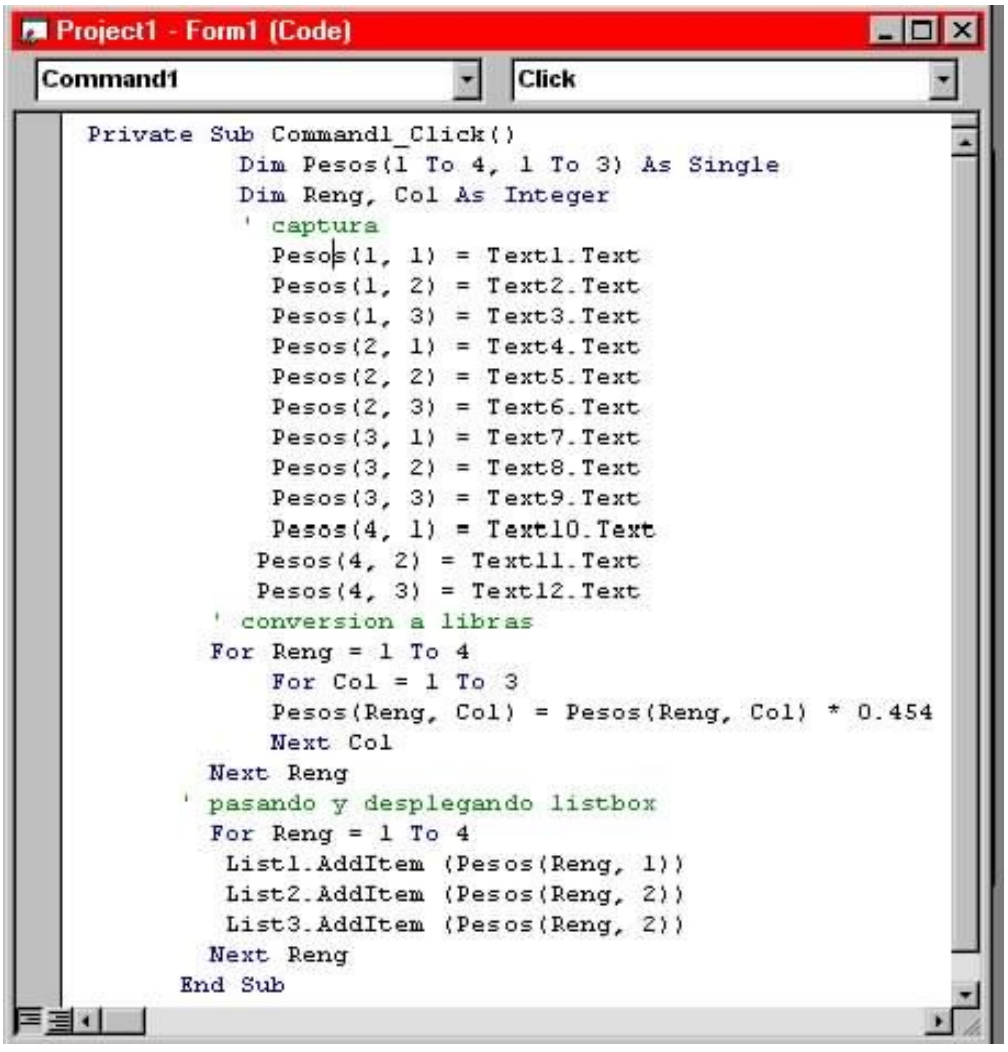
Problema ejemplo, capturar una tabla que nos muestre el peso en lbs de los tres jugadores claves de 4 equipos de fútbol, desplegarlos en otra tabla pero convertidos a kg. (una libra = .454 kg.), el programa y la pantalla de salida son; <ol type Diseño: <li Style

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

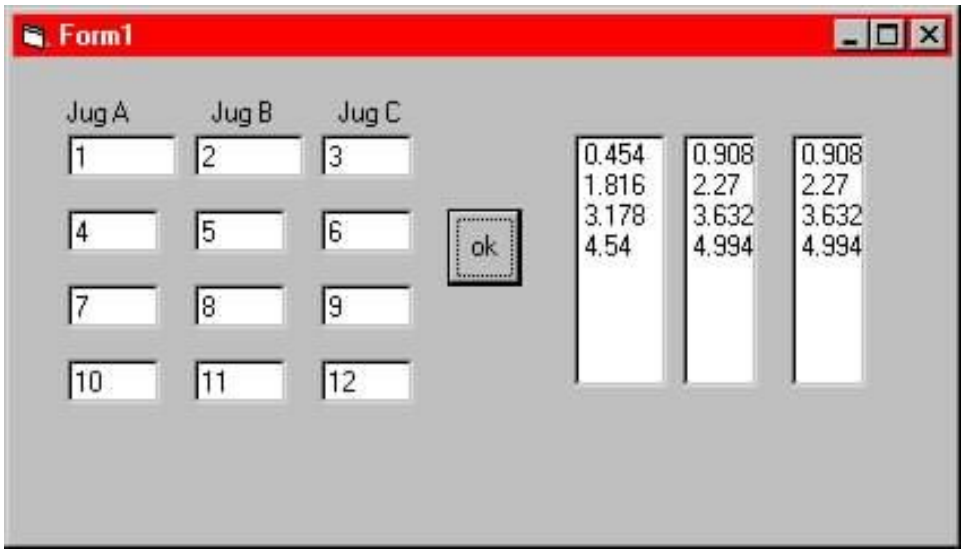


Codigo:



Observar que en procesos, son dos for's, y en despliegue, solo un for.

C)Salida:



Recordar que en este nivel de instruccion, solo se pretende, entender los conceptos asociadosa arreglos, mejores maneras de procesarlos existen, como se vera mas adelante, pero ya se puede:

TAREAS VISUAL BASIC

1.- Tareas programacion visual basic construir un cuadro que contenga los costos fijos de cuatro productos cualesquiera, que se producen en tres plantas diferentes de una empresa maquiladora.

2.- Tareas programacion visual basic construir un cuadro que contenga los ingresos mensuales por ventas durante los tres primeros meses del año de cuatro sucursales de una cadena de auto refacciones, agregar al final una lista que muestre los ingresos mensuales totales por meses y una segunda lista que muestre los ingresos mensuales totales por sucursal.

3.-Tareas programacion visual basic construir un cuadro que contenga las comisiones ganadas por tres vendedores, de los 5 tipos de linea blanca de conocida muebleria, ademas listas de comisiones totales y promedios ganadas por los vendedores, asi como listas de comisiones totales y promedios por tipo de linea blanca.

ANALIZAR ESTE CODIGO:

```
' PARA TOTALES Y PROMEDIOS POR RENGLON

FOR R = 1 TO 4

FOR C = 1 TO 3

TOTRENG(R) = TOTRENG(R) + TABLA(R,C)
```

```
NEXT C

PROMRENG(R) = TOTRENG(R)/3

NEXT R

'PARA TOTALES Y PROMEDIOS POR COLUMNA

FOR C = 1 TO 3

FOR R = 1 TO 4

TOTCOL(C)=TOTCOL(C) + TABLA(R,C)

NEXT R

PROMCOL(C) = TOTCOL(C) / 4

NEXT C
```

SUGERENCIA: CONSTRUIR PRIMERO LOS CUADROS EN PAPEL.

6.- VISUAL BASIC ARREGLOS DINÁMICOS

Hasta ahora los arreglos vistos son de tipo estáticos, es decir, son de tamaño fijo, ya definido o declarado.

Visual Basic, contiene los mecanismos apropiados para crear arreglos dnamicos, es decir, arreglos que tienen la capacidad de ir creciendo, para ajustarse a las necesidades del problema, incluso si perder los elementos de datos que ya contenga.

Para esto:

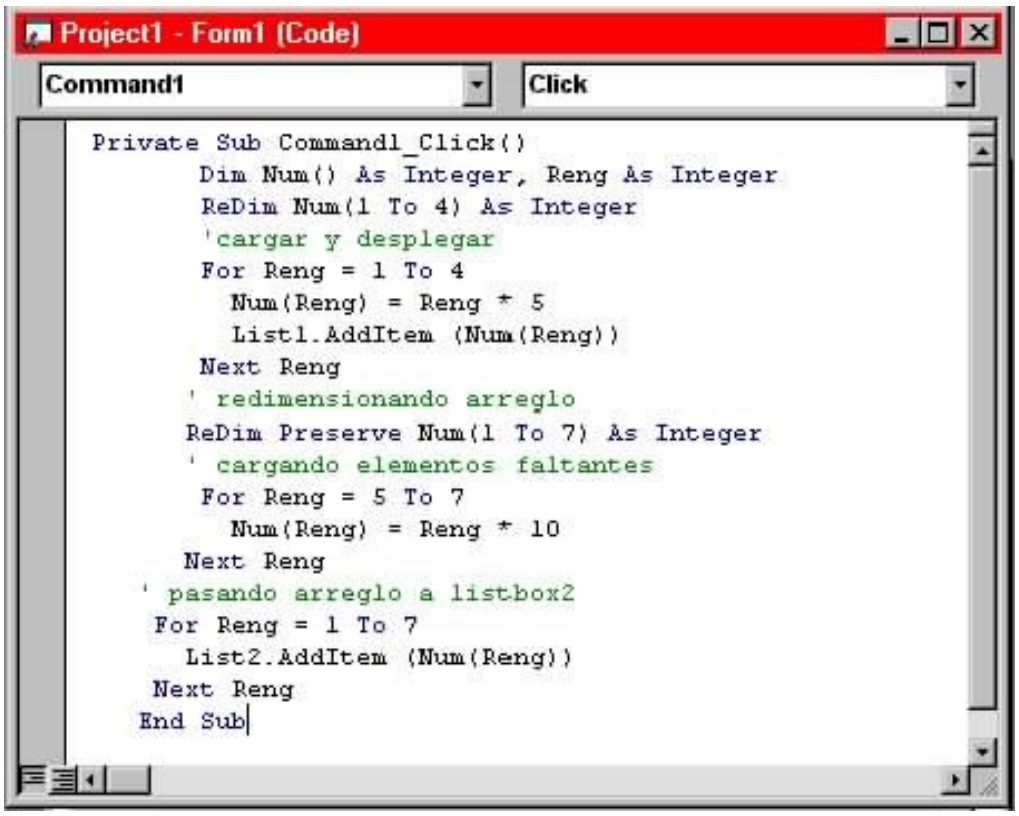
A) Declarar primero el arreglo sin tamaño fijo:

B) Usar la instrucción siguiente cada vez que se desee redimensionar.

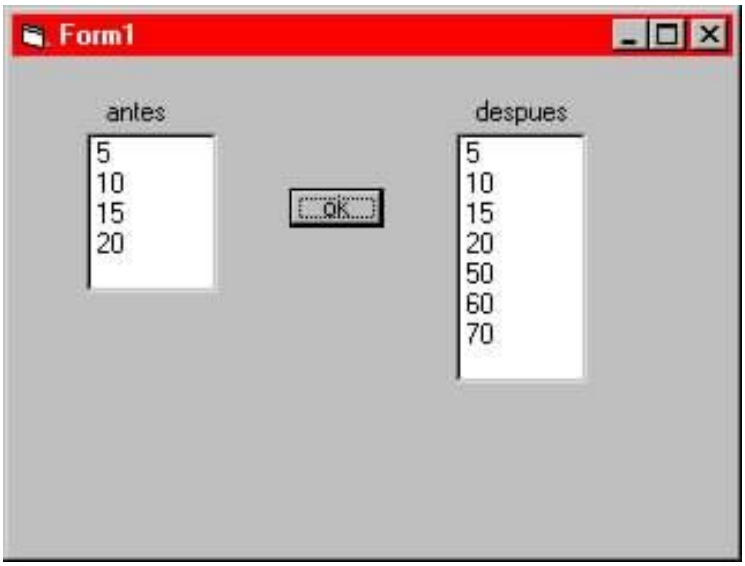
```
REDIM [PRESERVE] NOMARREGLO(1 TO REN,[1 TO COL]) AS TIPODATO
```


Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

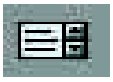


Pantalla de salida:



7.- CONTROLES VISUALES DE TIPO ARREGLO

8.- CONTROLES VISUALES TIPO LISTA

9.- CONTROL LISTBOX VISUAL BASIC 

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Este componente permite procesar visualmente un conjunto de elementos de tipo string.

Su primer y mas importante aspecto a recordar, cuando se procese o programe, es que el primer indice de la lista, es el indice numero 0(cero).

Este componente, contiene muchas propiedades y métodos que facilitan el trabajo con datos, entre ellas se encuentran:

PROPIEDAD ACCIÓN O SIGNIFICADO

AddItem(item, index) Inserta un elemento en posición indicada

Columns Para desplegar en una o mas columnas

Clear Elimina todos los elementos de la lista

List(index) Para accesar un elemento en posición

ListCount Regresa la cantidad de elementos en lista

RemoveItem(index) Elimina ítem en posición indicada

Sorted=true Ordena los elementos de la lista usada solo al tiempo de diseño

Notas: <ol type Capturas: Solo se ocupara un Text, el evento click del TextBox, y el método AddItem del ListBox. Procesos: Se ocupara un ciclo for , y los métodos list y listcount de ListBox Despliegues: No se ocupa, porque todos los cambios son visibles.

d) Despliegues: Pero si se quiere pasar de un ListBox a otro ListBox, entonces ciclo for, list y listcount

Ejemplo:

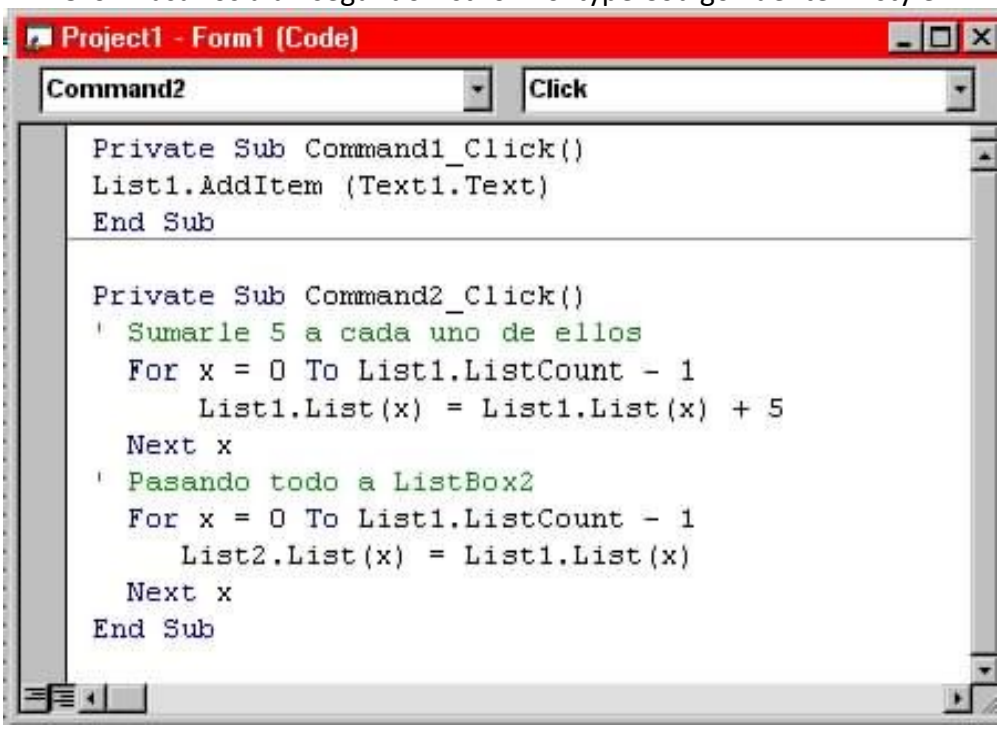
1ro.- Capturar números enteros en un ListBox

2do.- Sumarles 5 a cada uno de ellos

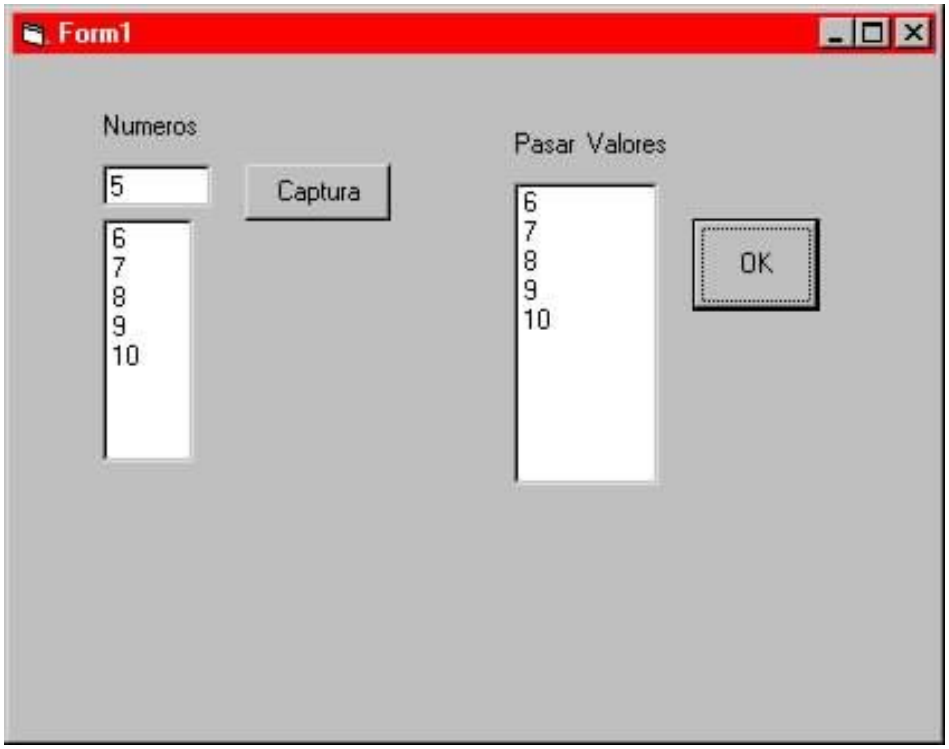
Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

3ro.- Pasarlos a un segundo ListBox <ol type Código Fuente <li style



Recordar que el primer índice en un ListBox es el cero, por eso el ciclo va desde el cero, hasta la cantidad de elementos menos uno. Pantalla de salida:



TAREAS PROGRAMACION VISUAL BASIC

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

1.- Capturar en una lista los sueldos de 6 empleados de un casino y desplegarlos en una segunda lista aumentados en un 30%

2.- Capturar en una lista los pesos en kilogramos de 6 personas desplegarlos en una segunda lista convertidos a libras y ademas solo los mayores de 100 libras.

3.- Capturar en sus 4 listas respectivas matricula, nombre y dos calificaciones de 5 alumnos, después calcular una lista de promedios de calificaciones.

4.- Capturar en sus listas respectivas numempleado, nomempleado, días trabajados y sueldo diario de 5 empleados, desplegar en otra pantalla o panel la nomina pero solo de aquellos empleados que ganan mas de $300.00 a la semana.

10 .- CONTROL MSFLEXGRID VISUAL BASIC 

Este control, no aparece entre los veinte controles de default que trae Visual Basic, importarlo al Tool Box, siguiendo el procedimiento que se dio en el ultimo tema de la primera UNIDAD VISUAL BASIC (Componente Animatión), la librería que lo contiene se llama Microsoft FlexGrid Control 5.0

Este componente es de los mas importantes, para el procesamiento de muchos datos, permite concentrar, procesar y mostrar gran cantidad de información para la vista del usuario.

Este componente presenta, manipula y procesa conjuntos de datos de tipo strings en forma tabular, es decir en forma de tablas, matrices, cuadros concentrados, ejemplo;

CIA ACME

INGRESOS POR VENTAS MENSUALES

MILLONES DE PESOS

ENE FEB MAR ABR

SUC A 1 2 3 4

SUC B 5 6 4 5

SUC C 6 7 8 9

Recordar que son los datos numéricos internos quienes se procesan (es decir, se capturan, se realizan operaciones con ellos, se despliegan, etc.), es la información externa quien le da sentido.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Algunas de sus propiedades y métodos mas interesantes son:

Cols.- Determina la cantidad de columnas que contendrá la tabla.

Recordar que para efectos de programación, la primera de ellas es la columna 0.

Rows.- Determina la cantidad de renglones que contendrá la tabla.

Recordar que para efectos de programación, el primero de ellos es el renglón 0.

Fixedcols , Fixedrows.- Determinan la cantidad de columnas y renglones fijos o de encabezado, estas propiedades ponerlas en 0.

Col, Row.- Al tiempo de ejecución del programa, regresan la posición de la celda actual, no confundir con Cols, Rows.

TextMatrix(Row,Col) = String, Es la propiedad mas importante, porque permite el acceso a cualquier celda de la tabla, ej.

ej.:

MsFlexGrid1.TextMatrix(2,4) = “PATO”

Observar que para accesar y manipular una celda, se debe primero indicar, el renglón y la columna adecuadas.

Otro aspecto importante a recordar, es que MSFlexGrid no permite edición directa por parte del usuario de sus celdas, por ese motivo se usara un componente externo TextBox para capturas, así como el evento click de MSFlexGrid.

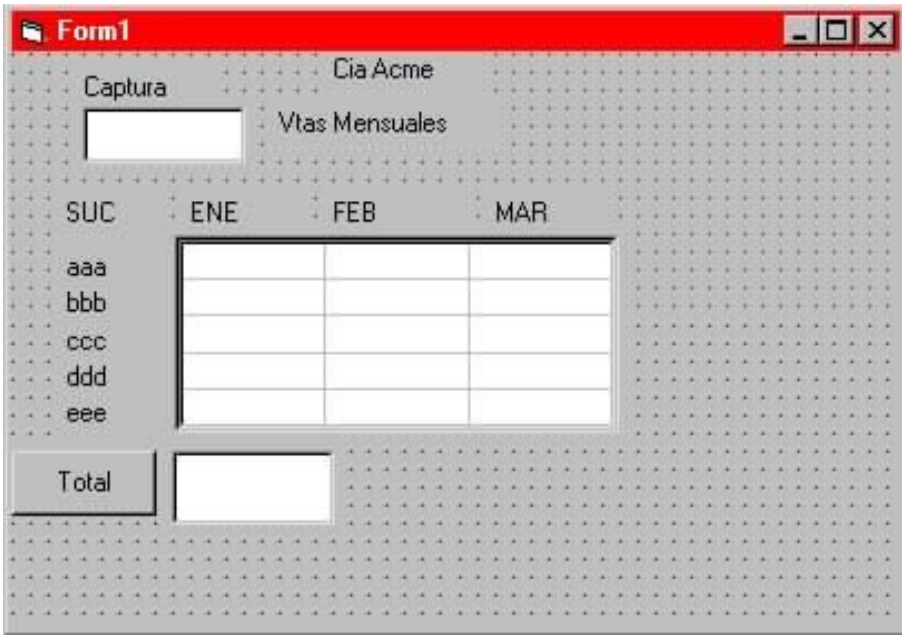
Para procesar todos los elementos de la tabla, solo recordar que se deben usar dos ciclos for, uno externo para controlar renglones, y uno interno para controlar columna.

Si solo se quiere procesar un solo renglón o columna, entonces solo se ocupara el ciclo contrario, y el renglón o columna original se darán como constantes, ver programa ejemplo.

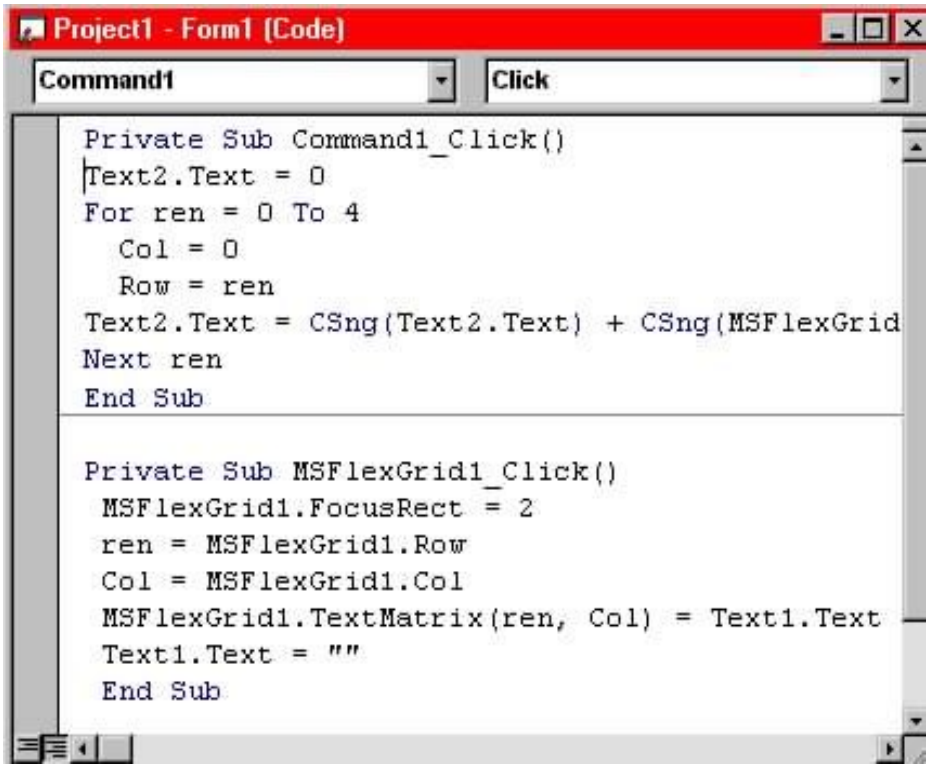
Ejemplo, Capturar una tabla de ingresos por ventas de la CIA Acme y obtener el total de las ventas del primer mes: <ol type Pantalla de Diseño: <li style

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



Código: <li style

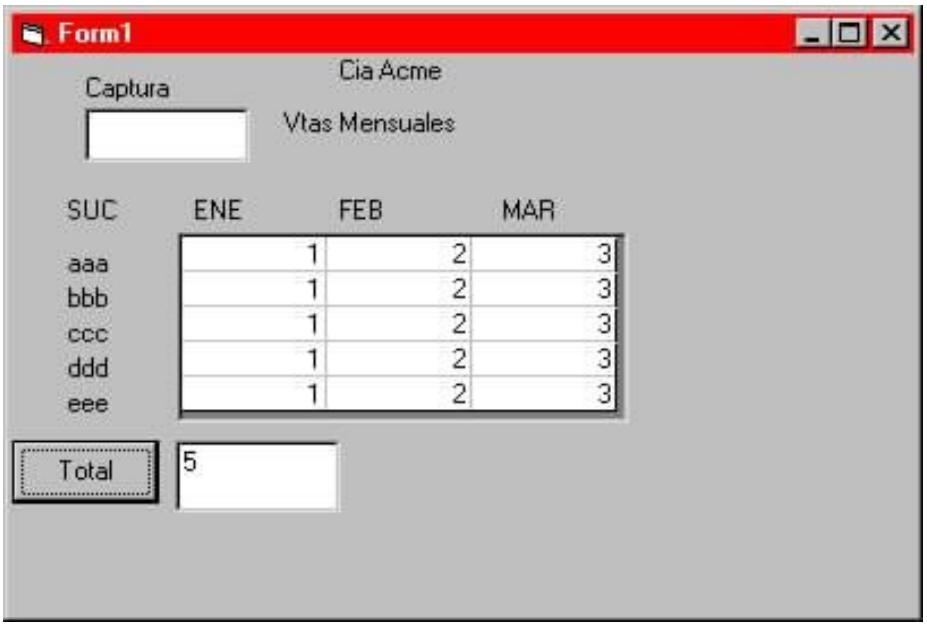


El Click del MSFlexGrid, usa la propiedad FocusRect, para graficar un rectángulo alrededor de la celda.

Se usa la propiedad MatrixText, para cargar la celda con el dato que se encuentra en el TextBox, observar que la posición, renglón, columna de MatrixText se obtienen usando las propiedades Row Y Col, al final se deja en blanco la caja TextBox, para que el usuario capture otro dato.

El Click del Command, primero se asegura de que este en 0(cero) la caja Text2 y luego se usa un ciclo renglón, porque como ya se indico , se quiere procesar una sola columna, misma que se dejo como constante, dentro de la operación.

Se esta usando el concepto de acumulador(Acum=Acum+NvoDato), para acumular el resultado. Pantalla de Ejecución:



Un proceso muy común con tablas, cuadros y concentrados es agregarles listas de totales y promedios ya sea por columna o por renglón, o ambas , por ejemplo;

CIA ACME

INGRESOS MENSUALES

(MILES DE PESOS)

ENE FEB MARZO TOTALSUC PROMSUC

SUC A 1 2 3 6 2

SUC B 4 5 6 15 5

SUC C 7 8 9 24 8

SUC D 10 11 12 33 11

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

TOTMES 22 26 30

PROMMES 5.5 6.5 7.8

En este ejemplo aparte de la tabla se ocupan 4 listas, dos para totales y dos para promedios.

El Codigo, para este tipo de problemas ya se dio en el tema de arreglos normales tipo tabla.

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un concentrado que despliegue los costos fijos de tres diversos productos que se fabrican en cuatro sucursales de una empresa MAQUILADORA.

2.- Construir un concentrado que contenga los ingresos por ventas mensuales de los 4 primeros meses del año de tres sucursales de una cadena refaccionaría, agregar listas de ingresos totales por mes e ingresos promedios por sucursal.

3.- Construir un cuadro que contenga las calificaciones de 5 materias de cuatro alumnos cualesquiera, incluir promedios de calificaciones por materia y por alumno.

CUESTIONARIO

1.- Cuando se usan arreglos

2.- Que es variable escalar

3.- Que es variable arreglo

4.- Que es elemento de un arreglo

5.- Como se simboliza o manipula un elemento de un arreglo

6.- Que es un arreglo tipo lista

7.- Que es un arreglo tipo tabla

8.- Formato para declarar una lista

9.- Cuantos for's se usan para manipular una lista

10.- Cuando se usa PUBLIC en la declaración de una lista

11.- Que es sorteo

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

12.- Algoritmo de sorteo usado en este curso

13.- Formato de declaración de un arreglo tipo tabla

14.- Cuantos ciclos se usan para manipular una tabla

15.- Que son arreglos dinámicos

16.- Como se declaran arreglos dinámicos en V.B.

17.- Que es control ListBox

18.- Numero del primer indice en un control ListBox

19.- Cinco propiedades importantes del control ListBox

20.- Que es control MSFlexGrid

21.- Propiedades importantes del control MSFlexGrid

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

UNIDAD IV

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

IV.- INT A LAS BASES DE DATOS

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

IV UNIDAD VISUAL BASIC BASES DE DATOS VISUAL BASIC

1.- INTRODUCCIÓN

En este capitulo se analizan en general dos problemas:

a) Variables que permitan almacenar conjuntos de datos como los arreglos pero de distintos tipos de datos, este primer problema se resolvía en la antigüedad usando las llamadas variables registro.

b) Permanencia de los datos, hasta ahora todos los datos capturados, calculados, creados, etc. al terminar o cerrarse el programa se pierden y es necesario volver a capturarlos, etc., en la siguiente ejecución o corrida del programa.

Tradicionalmente en programación antigua este segundo problema se resolvía usando el concepto de archivos, que son medios permanentes de almacenamiento de datos en los dispositivos o periféricos apropiados, generalmente disco, cinta magnética, etc.

2.- MODELOS DE ALMACENAMIENTO DE DATOS

En general existen dos modelos de almacenamiento de datos en los sistemas de información.

a) El modelo tradicional de archivos que se construye con los siguientes elementos:

1.- **Variables Registros,** como ya se indico son variables que permiten almacenar conjuntos de datos de diverso tipo.

También se pueden definir como representaciones simbólicas y programáticas de entidades lógicas de información, ejemplos de variables registros son alumnos, empleados, clientes, proveedores, productos, autos, etc.

Estas variables registros, también ocupan programas o rutinas de programas para procesarlas, por ejemplo un procedimiento, modulo o subrutina, se encargara de capturar los datos que contendrá la variable registro, otro procedimiento para corregir los datos que ya contiene, otro procedimiento para desplegarlos en pantalla ya cuando ha sido capturada y así sucesivamente.

2.- **Archivos**, que en principio pueden entenderse como una especie de almacenes o bodegas para almacenamiento de datos en forma permanente en disco, es decir un archivo de empleados en disco contiene todos los datos de todos los empleados de una empresa.

Igualmente los archivos ocupan su propios programas o subrutinas o procedimientos especializados por ejemplo, procedimientos para crear los archivos, para almacenar o

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

dar de altas los registros en el archivo, procedimientos para buscar un registro determinado, procedimiento para dar de baja un registro, etc.

3.- **Una aplicación**, que es un programa que se encarga de coordinar todos los programas descritos y presentar a usuarios de manera clara, fácil, accesible y entendible.

Salta a la vista que construir un sistema de información por ejemplo para una tienda de vídeo o para un refaccionaría, etc. involucra un gran cantidad de trabajo de programación, puesto que hay que programar muchas variables registros, muchos archivos en disco, y una o varias aplicaciones.

Este modelo se usa todavía en la actualidad, pero es obvio que mejores maneras, mas rápidas, seguras y eficientes existen en la actualidad para resolver estos problemas, y esto nos lleva al segundo modelo de datos.

b) Modelo de Bases de Datos Relaciónales, este modelo intenta simplificar la construcción de sistemas de información como los antes descritos, este modelo solo incluye en forma simple los siguientes elementos:

b.1) Tablas, es una combinación de las variables registro y de los archivos del modelo anterior.

Es decir cuando un programador moderno define o declara una tabla en un programa, realmente esta haciendo dos cosas por el precio de una, es decir crea una variable registro en memoria que almacenara los datos y al mismo tiempo ya esta creando un archivo en disco que se llamara igual que la tabla o variable registro y que automáticamente se convertirá en un espejo de la tabla en memoria, es decir cuando se cargan los datos en la tabla en memoria, también se estarán cargando en disco.

Otra vez cuando el programador escribe código para capturar los datos y mandarlos a la tabla en pantalla-memoria, realmente también lo esta haciendo para darlos de alta en disco.

b.2) Aplicación, que tiene la misma función que en el modelo anterior.

No confundir este concepto de tablas en base de datos con el concepto de tablas, vistos en el capitulo de arreglos, aunque en la practica se parecen mucho, la diferencia es que los datos no se van a disco.

Como se observa en este modelo, es mas sencillo construir sistemas de información con el, puesto que la parte programática se reduce ampliamente.

6.- APLICACIONES O PROGRAMAS CON TABLAS VISUAL BASIC

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Básicamente una aplicación consiste de un programa o forma que permite accesar, manipular, editar, procesar, etcétera, los datos, registros, o renglones de una tabla, es decir es la aplicación la que constituye, la vista sobre la que trabaja el usuario del programa.

Como ya se explico en el modelo tradicional de almacenamiento de datos, existían muchos procesos, tanto para la manipulación de registros en memoria, capturas, modificaciones, etc., como para el almacenamiento de todos ellos en archivos en disco, el famoso ABC de un sistema de archivos (altas, bajas, consultas, modificaciones, etc.) y todos ellos había que construirlos programarlos a mano, era terrible la situación de los programadores de antaño.

En el modelo relacional-visual de datos, mucho de este trabajo no existe, solo se construye una forma, y se pegan unos cuantos componentes para que aparezcan las propias tablas y componentes para que se procesen los datos que contendrán las tablas.

Recordar también que la aplicación ya no deberá preocuparse, donde se almacenan los datos, de validar las condiciones y restricciones impuestas en ellos, etc., todo este trabajo ahora es responsabilidad del miniDBMS que usado, el DataBase Desktop.

Existen dos maneras sencillas de construir aplicaciones, la primera de ellas es presentarle al usuario un solo renglón de información para su proceso y manipulación y la segunda es presentarle toda la tabla a la vez, también para su manipulación y proceso.

4.- TABLAS (CONTINUACIÓN)

El trabajo correcto con bases de datos relaciónales, se divide en dos grandes pasos o etapas bien diferenciadas entre si:

En la primera etapa se diseña la tabla , con sus campos, llaves y condiciones especiales, luego se usa un paquete o programa de software especializado en la construcción, mantenimiento y administración de la base de datos, este software se usa para convertir la tabla o tablas ya bien diseñadas en un archivo en disco.

Estos paquetes, o software reciben el nombre de DBMS(DATA BASE MANAGEMENT SYSTEM) o sistema administrador de bases de datos.

Este software se especializa en la creación, mantenimiento, seguridad, privacidad, etc. de un conjunto de tablas o mejor dicho una base de datos, los DBMS mas comunes son Oracle, SqlServer, Informix, Sysbase, etc.

Visual BASIC, lleva incorporado un pequeño administrador de bases de datos, denominado Visual Data Manager.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

La segunda etapa consiste en construir la aplicación o aplicaciones que ya tendrán acceso o podrán manipular los datos contenidos en la tabla, estas aplicaciones se escriben usando ya sea lenguajes clásicos de programación como BASIC, PASCAL, COBOL, CBUILDER, DEL PHI, etc., o también se pueden accesar paquetes comunes de manipulación de archivos o bases de datos como DBASE, CLIPPER, VISUALFOX, ACCESS, etc.

DISEÑO Y CREACIÓN DE UNA TABLA

El primer paso antes de usar el paquete correspondiente a esta tarea, es diseñar la tabla completamente, esto exige:

a) Nombre apropiado y determinación de atributos y campos correspondientes.

b) Seleccionar y determinar el atributo principal o campo clave o llave primaria que se utiliza como el identificador único que permite diferenciar cada instancia o renglón diferente dentro de la tabla.

c) También se puede seleccionar otros campos que puedan servir mas adelante para ordenar de manera diferente la tabla, es decir una tabla en principio ya está ordenada por campo clave, por ejemplo, la matricula de un alumno, el numero de empleado, etc., pero existirán muchas ocasiones, mas adelante donde se puede pedir un orden diferente, por ejemplo, por ciudad, por carrera, por nombre, por edad, etc., la buena ingeniería de una base de datos exige tomar en cuenta estos y otros muchos problemas y detalles.

d) A estos atributos o campos especiales se les conoce como claves o llaves secundarias, que internamente generan otra tabla especial llamada tabla o archivo de índices, (tabla o archivo que contiene dos campos, el primero es la clave secundaria ordenada y el segundo la posición o renglón donde se encuentra en la tabla original).

e) Escribir restricciones y condiciones apropiadas para ciertos atributos, por ejemplo el número de empleado deben comenzar en 500, la edad no debe ser mayor de 150 años, etc.

Ya listo el diseño de la tabla, se pasa a el programa correspondiente para su creación y almacenamiento, dicho ADD-IN se encuentra dentro de Visual BASIC, bajo Add-Ins, Visual Data Manager:

5.- VISUAL DATA MANAGER

No es un componente de la barra de herramientas, es un programa que se encuentra dentro de Visual BASIC.

Su Pantalla de arranque es:

El procedimiento completo para construir la tabla, es:

1) Primero crear un folder o subdirectorio especial en el disco duro C: para guardar o almacenar las tablas y la base de datos (conjunto de tablas), esto se hace con un clic en My Computer, luego clic en disco C;, luego File, New Folder, y renombrarlo por ejemplo a Datos1, o Tablas o Base de Datos (Win95 permite directorios con espacios en blanco intermedios).

2) Ahora cargar o ejecutar el VISUAL DATA MANAGER(VDM), aparece el VDM de la gráfica anterior.

3) File, New, Microsoft Access, Versión 7.0 MDB

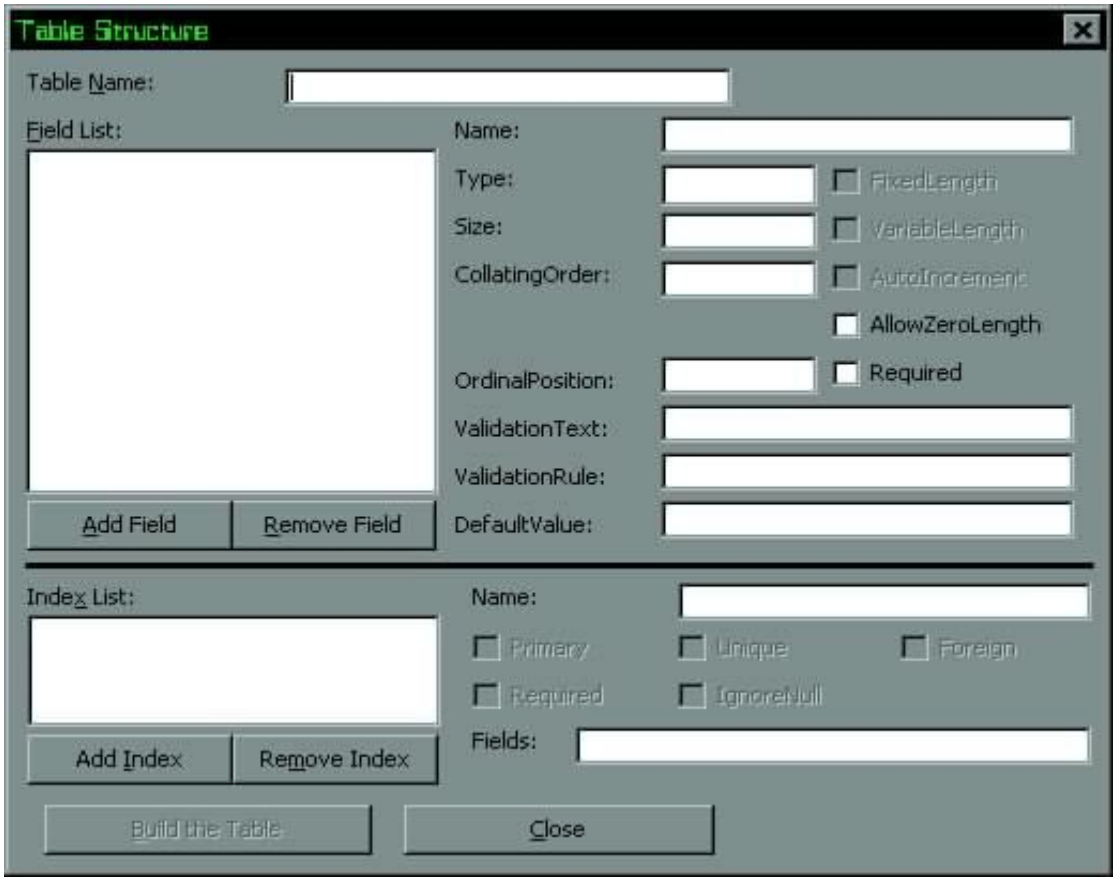
4) Aparece la pantalla normal de grabación de archivos, buscar y abrir el folder donde quedara grabada la base de datos(aunque también recordar que aquí se puede crear un new folder y renombrarlo).

5) En File Name: Escribir el nombre de la base de datos (no el de la tabla), por ejemplo(BaseDatos, Cía Acme, Nomina, Inventarios, etc.), recordar que un sistema de información completo, contiene muchas tablas, dado el nombre, para el ejemplo el folder donde queda guardado se llamara datos, la base de datos se llamara BaseDatos, usar la opción Save, aparece el siguiente editor de bases de datos:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

6) Crear ahora una tabla nueva, en este ejemplo la de alumnos, con un clic derecho de minimenu en la parte donde dice +PROPIERTY y usar la opción new table, y aparece la siguiente pantalla (TABLE STRUCTURE DIALOG):

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Sus elementos más importantes son:

* Table Name Nombre de la Tabla

* Field List Lista todos los campos de la tabla

* Name Permite Renombrar un campo ya hecho

* Type Tipo de dato del campo

* Size Determina un tamaño fijo de campo(string)

* ADD FIELD Ver grafico(ADD FIELD DIALOG)

* Remove Field Elimina el campo seleccionado

* Index List Lista todos los indices disponibles

* Primary Indica que el indice es llave primaria

* Unique Indica que el indice es unico(no duplicados)

* Foreing Indica que el indice es una llave foránea

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

* Required Indica que el indice es requerido

* ADD INDEX Ver gráfico (ADD INDEX DIALOG)

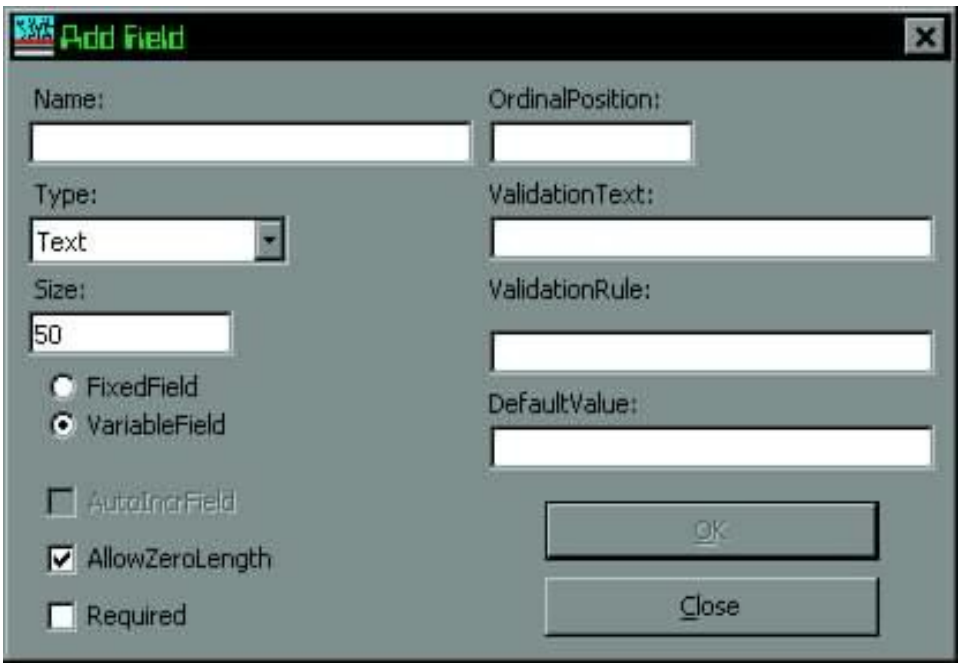
* Remove Index Elimina índex seleccionado

* Build Table Construye y añade la tabla a la base de datos

* Print Structure Imprime la estructura de la tabla, ya debe estar creada

ADD FIELD DIALOG

Así como TABLE STRUCTURE se usa para construir y validar toda la tabla, este dialogo(ADD FIELD) se usa para construir y validar, todos y cada uno de los campos de la tabla.



Sus elementos son:

* Name Nombre del campo

* OrdinalPosition Posición del campo (empiezan en 0)

* Type Tipo de dato del campo

* ValidationText Mensaje a mandar cuando usuario se equivoca de tipo

* DefaultValue Carga un valor de default

* Size Tamaño de campo(strings)

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

* OK Añade el campo a la tabla

* Close Cierra el editor de campos

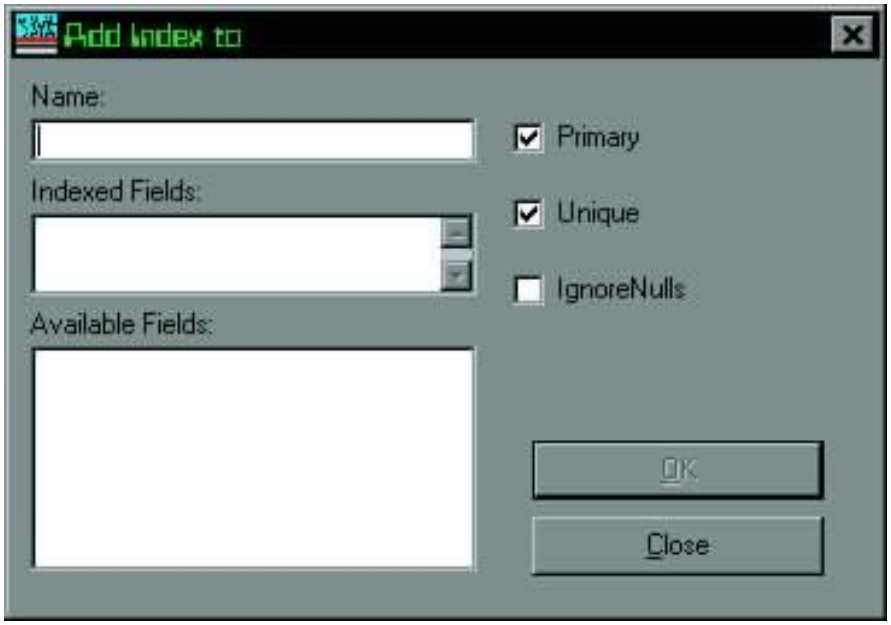
ADD INDEX DIALOG

De Nuevo:

Table Structure.- Construye y valida toda la tabla

Add Dialog.- Construye y valida cada campo

ADD ÍNDEX.- Construye y valida cada indice de la tabla



NOTA: ESTE DIALOGO SE DEBE USAR YA QUE ESTÉN CONSTRUIDOS TODOS LOS CAMPOS.

Sus elementos son:

* Name.- Nombre del indice

* IndexedFields.- Lista los campos indices(solo clic enel campo o campos a indexar en la ventana de abajo llamada Available Fields)

*AvailableFields.- Muestra todos los campos que ya se debieron haber hecho para la tabla

* Primary.- Indica que el campo indice es la llave primaria de la tabla.

* Unique.- Indica que este indice debe ser único (no permite duplicados)

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

* OK.- Añade el indice a la tabla

* Close.- Cierra el ADD ÍNDEX DIALOG

7) Crear la tabla alumnos, con los siguientes datos, usando los diálogos respectivos.

Tabla: alumnos(campos, usar add field dialog)

Matricula Long requerid

Nombre Text * 30

Edad Integer

Domicilio Text * 20

Ciudad Text * 20

alumnos(indices, usar índex field dialog)

nombre del indice campo

matriculaindice matricula Primary Unique

ciudadindice ciudad atn: desmarcar

primary, unique

NOTA ESTE FORMATO TAMBIÉN PUEDE PEDIRSE DE LA SIGUIENTE MANERA

CAMPO TIPO SIZE LLAVE PRIMARIA INDICE

_

Matricula Long * I

Nombre Text 30

Edad Integer

Domicilio Text 20

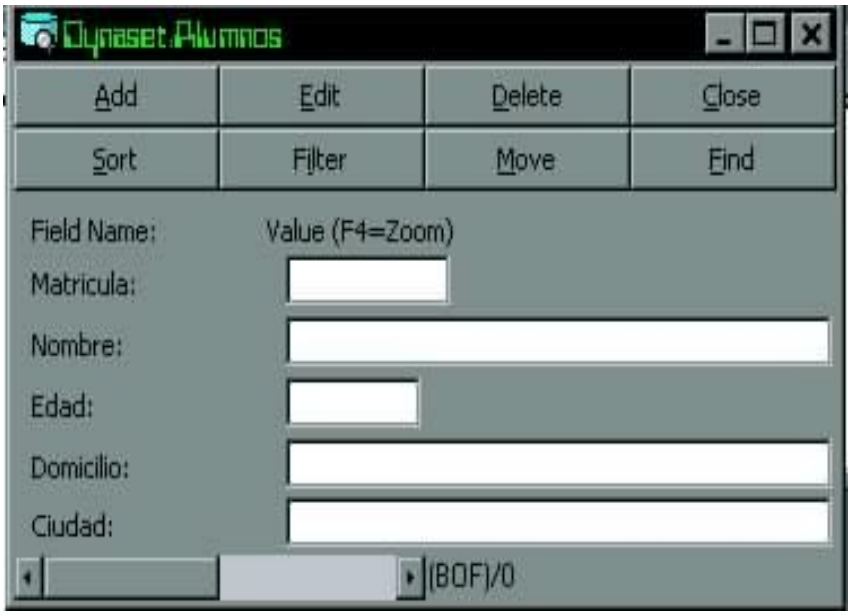
Ciudad Text 20 I

_

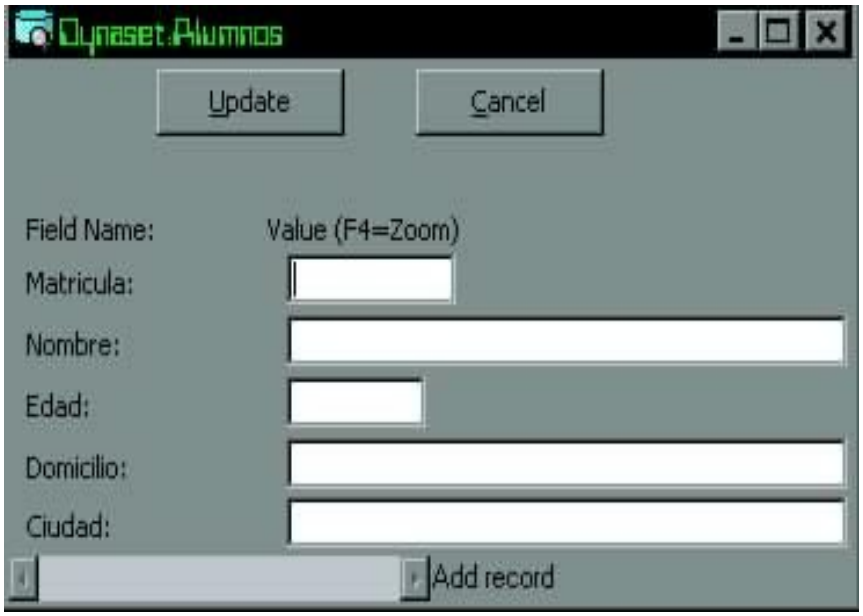
Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

8)Ya creada la tabla, y de regreso al Visual Data Manager, cargarle algunos datos o renglones de prueba, para hacer esto, solo clic derecho en tabla alumnos para minimenu y usar la opción open, aparece el siguiente editor de datos:



Para cargar un renglón, usar primero la opción ADD y aparece la siguiente gráfica



Cargar varios(10) Renglones de datos para pruebas, no olvidar usar el botón de update.

La barra de abajo es llamada de navegación, para recorrer todos los renglones de la tabla, ya cargados unos 10 renglones se puede usar, para navegar entre renglones.

Para corregir un renglón, primero usar la barra de navegación para seleccionar y luego usar el botón EDIT.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

Si existen muchos renglones el botón SEEK, los busca usando la llave primaria (MatriculaIndice) o la llave secundaria (CiudadIndice), revisar que en el editor de datos, la opción índex, indique si es MatriculaIndice o CiudadIndice, sino la opción seek va a fallar.

La opción Delete, funciona directamente en el editor de datos, solo oprimir y pregunta si se elimina el renglón actual.

La opcion Filter, se usa para dar instrucciones en SQL un lenguaje de consulta de bases de datos muy especializado.

9) Ya construida y cargada la tabla, cerrar el Visual Data Manager, para regresar al Visual BASIC.

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir con el Visual data Manager, las diez tablas diseñadas.

6.- APLICACIONES O PROGRAMAS CON TABLAS VISUAL BASIC

Básicamente una aplicación consiste de un programa o forma que permite accesar, manipular, editar, procesar, etcétera, los datos, registros, o renglones de una tabla, es decir es la aplicación la que constituye, la vista sobre la que trabaja el usuario del programa.

Como ya se explico en el modelo tradicional de almacenamiento de datos, existían muchos procesos, tanto para la manipulación de registros en memoria, capturas, modificaciones, etc., como para el almacenamiento de todos ellos en archivos en disco, el famoso ABC de un sistema de archivos (altas, bajas, consultas, modificaciones, etc.) y todos ellos había que construirlos programarlos a mano, era terrible la situación de los programadores de antaño.

En el modelo relacional-visual de datos, mucho de este trabajo no existe, solo se construye una forma, y se pegan unos cuantos componentes para que aparezcan las propias tablas y componentes para que se procesen los datos que contendrán la tablas.

Recordar también que la aplicación ya no deberá preocuparse, donde se almacenan los datos, de validar las condiciones y restricciones impuestas en ellos, etc., todo este trabajo ahora es responsabilidad del miniDBMS que usado, el DataBase Desktop.

Existen dos maneras sencillas de construir aplicaciones, la primera de ellas es presentarle al usuario un solo renglón de información para su proceso y manipulación y la segunda es presentarle toda la tabla a la vez, también para su manipulación y proceso.

7.- APLICACIONES POR RENGLON

Una de las ventajas de Visual BASIC es que muchos de sus controles standards son "data aware" o "data bound", es decir permiten conectarse o tomar sus datos directamente de una tabla de la base de datos, estos controles estandars son:

CheckBox, ComboBox, Image, Label, ListBox, PictureBox, TextBox

De los controles proporcionados en forma extra por Visual Basic(Customs Controls), son "data aware":

DataList, DataCombo, DataGrid, MSFlexGrid, RichTextBox, Microsoft Chart, ImageCombo, DateTimerPicker, MonthView

Para construir una aplicación, presentándole un solo renglón al usuario para su procesamiento en Visual BASIC, solo se ocupa una forma (Form1) y dos componentes diferentes para el acceso y y manipulación de tablas, estos componentes son:

PROCEDIMIENTO Y COMPONENTES

1.- Colocar un componente Data Control

Abajo en la forma y abrirlo.

Este componente Data Control, sirve de enlace físico entre la tabla real que se creo con el Visual Data Manager y el resto de los controles que se usaran para la construcción de la aplicación.

Sus propiedades más importantes son:

Propiedad DataBaseName = Clic en elipsis (…) a la derecha, abrir, buscar y seleccionar la base de datos indicada, en el ejemplo (basedatos).

Propiedad RecordSetType = Poner en 0.-Table (las otras dos opciones son vistas en Dynaset o Snapshot).

Propiedad RecordSource = Clic a un lado y seleccionar la tabla apropiada, en este ejemplo alumnos.

Propiedad ReadOnly = Ponerla en False(es default), se usa para dar permisos de edición de registros.

Caption = mensaje a desplegar en este control

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

EofAction = AddNew, esto es, cuando se llegue al final de la tabla, se dará permiso al usuario de agregar un nuevo renglón.

2.- Control(es) TextBoxs: 

Ya ampliamente conocido, este componente ahora permite presentar y manipular un dato de la tabla a la vez.

De este control se deben colocar tantos de ellos en la forma, como columnas existan en la tabla a procesar.

Sus dos propiedades mas importantes y en el orden abajo descrito son:

Propiedad DataSource = Clic a un lado y seleccionar el DataControl al que se quiere enlazar, por ejemplo Data1.

Propiedad DataField = Clic a un lado y seleccionar el campo o atributo o columna que se quiere representar o enlazar por ejemplo Matricula, Nombre, Edad, etc.

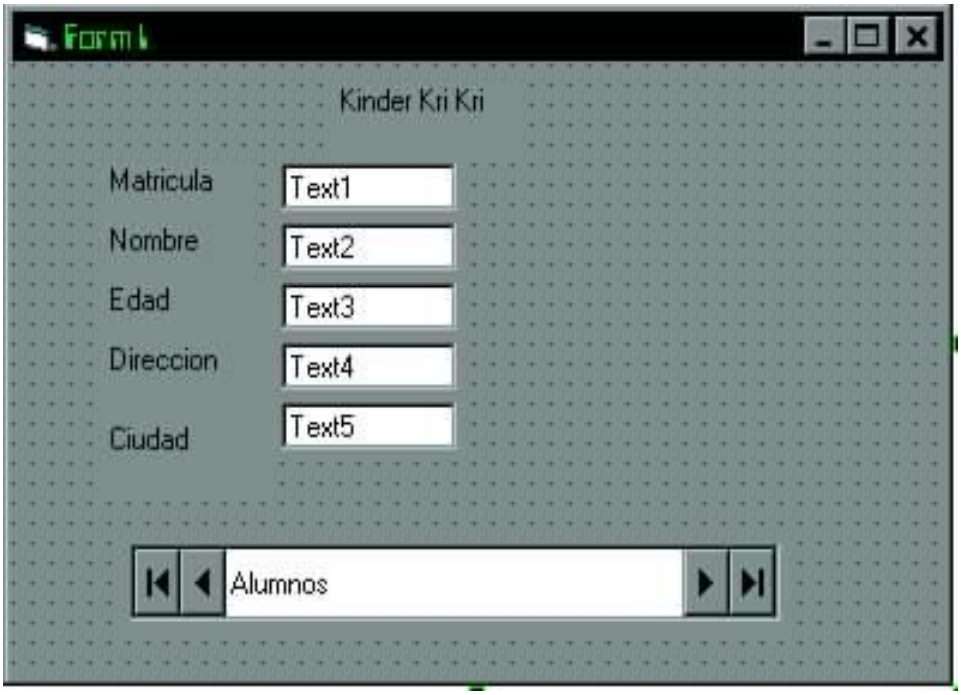
Recordar que se deben colocar tantos TextBox en la forma, como columnas existan en la tabla, o como columnas se quieran presentar al usuario para su manipulación.

Recordar poner unos cuantos componentes Label, arriba del componente TextBox y cargarlos en su propiedad caption con el nombre del campo o columna respectivo.

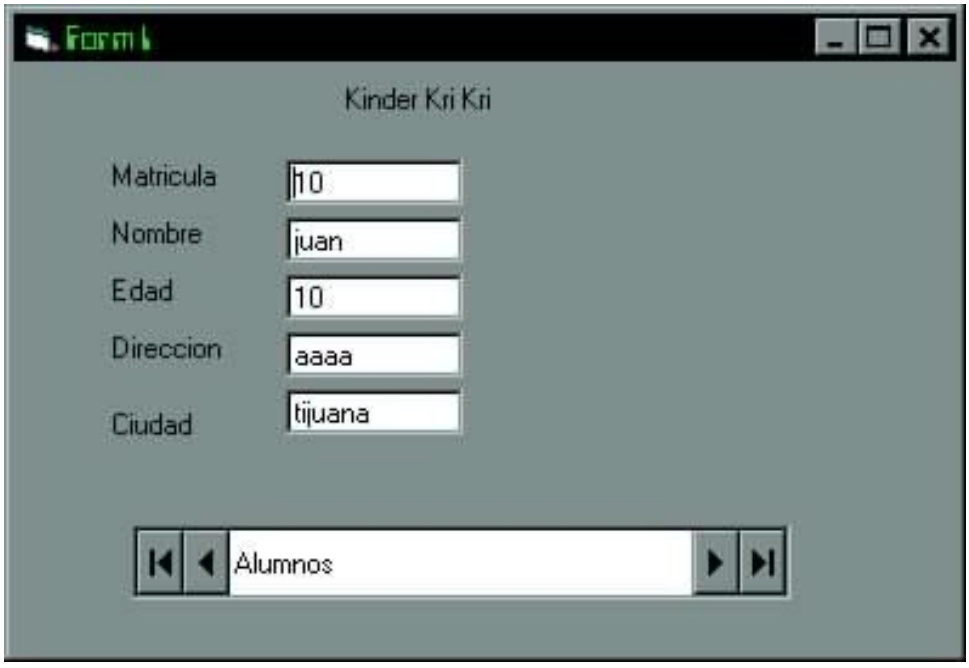
Con estos dos componentes ya se construye una aplicación, ejemplo;

Pantalla de Diseño

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Pantalla de Ejecución



Notas Importantes:

Para navegar la tabla, usar el datacontrol, su elementos son:

* Renglón Anterior

* Renglón Siguiente

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

* Primer Renglón

* Ultimo Renglón

Si el usuario modifica o edita un dato, y avanza a renglón siguiente, la tabla en disco se actualiza, si no se quiere dar permiso de edición o modificación, usar la propiedad ReadOnly en DataControl, o usar Labels para desplegar los datos.

Recordar que también se pueden usar otros controles, para desplegar datos, por ejemplo DBListBox, despliega toda una columna completa.

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir aplicaciones para la mitad de las tablas hechas con el Visual Data Manager, y usar controles de despliegue de datos diferentes.(DBListBox importarlo)

8.- APLICACIÓN POR TABLA

En este tipo de aplicación, al usuario se le presenta toda la tabla completa a la vez para su procesamiento y manipulación.

Para construir este tipo de aplicación, solo se ocupan también dos controles:

a) Control DataControl

* propiedad DataBaseName = basedatos * propiedad RecorsetType = 0.-Table * propiedad RecorSource = Alumnos * propiedad readOnly = false * propiedad EofAction =AddNew

b) Control DBGrid 

Primero se deberá importar a TOOLBOX, se llama Microsoft DataBound Grid Control.

Este componente le presenta todos los renglones a la vez al usuario para su procesamiento.

Propiedades

* DataSource = Clic y seleccionar Data1

* HeadLines = 1 , numero de renglones de encabezado

* AllowAddNew = True, permite agregar nuevos renglones

* AllowDelete = True, permite eliminar un renglón (para esto, clic en cuadrito blanco al principio del renglón, el renglón debe seleccionarse completamente y usar tecla del)

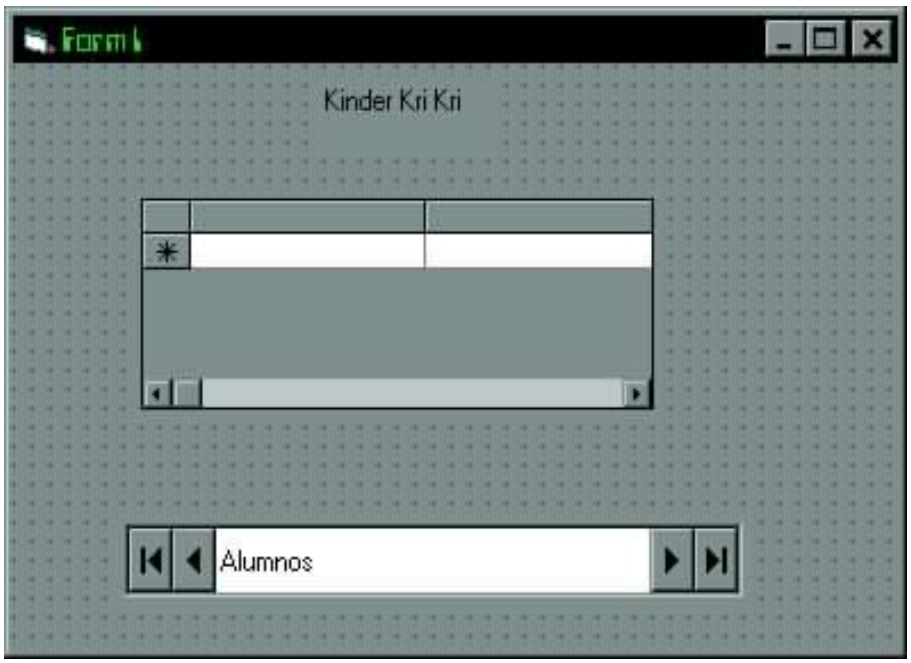
Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

* AllowUpdate = True, permite actualizar la tabla en disco

* AllowArrows = true, permite usar las teclas de flechas

Ejemplo <ol type Pantalla de Diseño



<ol type Pantalla de Corrida:

TAREAS PROGRAMACION VISUAL BASIC

Hacer las 5 tablas DBGRID faltantes

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

9.- PROCESOS BASICOS

Analizaremos ahora algunos procesos también básicos que pueden realizarse con los dos tipos de aplicaciones y que ademas no están contemplados en el navegador.

10.- VISUAL BASIC OPERACIONES CON CAMPOS

Para el caso de aplicaciones construidas con componentes TextBox de visual basic , solo usarla en forma normal, por ejemplo en una aplicación donde la tabla productos tiene un campo costounitario(Text5) y un campo o columna utilidad(Text7), poner un Text10 para el precio de venta y realizar la siguiente operación en el clic de un botón apropiado.

Notas:

1.- Los TextBox de visual basic deben estar enlazados a DataControl.

2.- Si el de total o precio de venta esta enlazado, se actualizara, también en la tabla en disco.

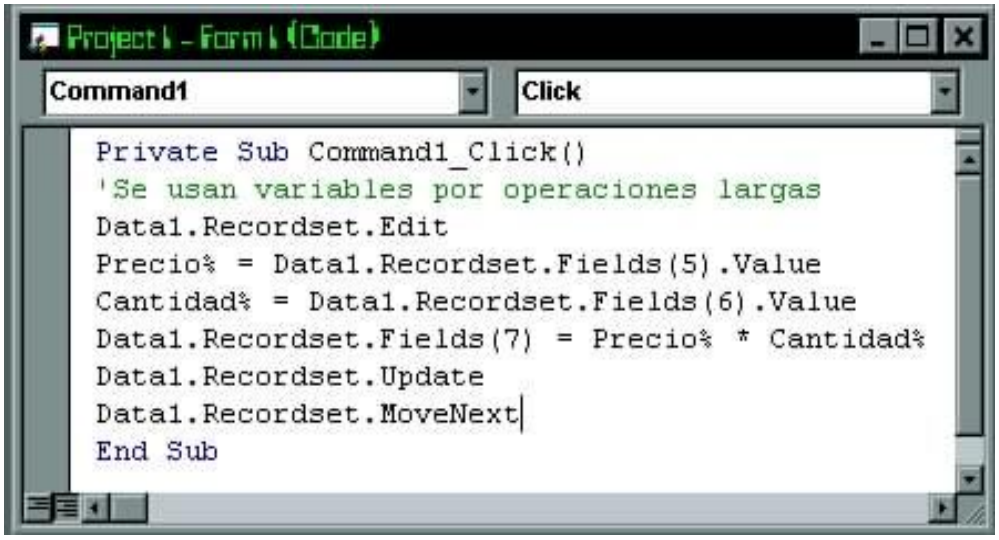
3.- Recordar que la actualización solo se realiza, hasta que se avanza el renglón.

4.- Observar que es de rigor, forzar el TextBox a su tipo de dato adecuado, usando las funciones de conversión normales.

TAREA VISUAL BASIC

1.- Una tabla de productos con campos costo unitario, margen de utilidad y tasa de IVA, calcular el precio de venta.

Para el caso de aplicaciones con DBGrid, se pueden accesar sus celdas para su lectura y proceso usando código como en el siguiente ejemplo, en algún botón o evento apropiado:



Observar que todo el proceso, es con DataControl, no con DBGrid.

Se esta usando la propiedad RecordSet y muchos métodos asociados a ella, como edit, fields, value, etc., es conveniente sobre todo en cursos mas adelantados, que se conozcan a fondo las propiedades de RecordSet.

En cuanto al programa, es una tabla de productos que contiene, los campos precio, cantidad y total, el código lo que realiza es multiplicar los campos precio y cantidad (en este caso se usa valúe para leer el dato) y se calcula el total, para este ultimo caso se usa para cargar el campo respectivo de la tabla.

Las propiedades de RecordSet usadas son:

Edit → Para que renglón entre a modo edición

Fields(Numcampo)→ Para determinar campo a accesar, tomar en cuenta que primer campo es el 0(cero)

Valúe → Para leer o cargar un dato

Update → Para actualizar el dato en tabla disco

MoveNext → Para avanzar renglón

TAREA PROGRAMACION VISUAL BASIC

1.- La misma tarea anterior, pero ahora con DBGrid.

11.- BUSQUEDAS VISUAL BASIC

Un problema muy común en los sistemas de información basados en bases de datos es el de búsquedas, en estos casos el usuario proporciona un dato, generalmente la clave del registro, para que se localice toda la información pertinente.

Para resolver este problema, solo se ocupa:

1.- Una aplicación construida por renglón, es decir con un DataControl y TextBoxs enlazados a Data1, para desplegar los resultados.(no incluir el campo clave)

2.-Un TextBox no enlazado a Data1, para almacenar el dato o valor a buscar por parte del usuario.

3.-Haber estudiado en la ayuda del Visual BASIC las propiedades de RECORDSET.

4.-Un botón de ordenes (OK) con el siguiente código:

Notas:

Primero se pone el indice a buscar, por supuesto que una tabla solo tiene una llave primaria o indice primario, pero puede tener varios indices secundarios, para el ejemplo de la tabla alumnos, recordar que también se hizo un indice secundario, por ciudad (ciudadindice).

Segundo se usa el método Seek, que es quien realiza propiamente la búsqueda, sus parámetros son:

.Seek “op relacional”, datoabuscar

TAREAS PROGRAMACION VISUAL BASIC

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

1.- CONSTRUIR UNA APLICACIÓN VISUAL BASIC DE BÚSQUEDAS PARA UNA TABLA DE AUTOS, CON OPCIONES DE BÚSQUEDA POR CLAVE, MODELO, COLOR, PRECIO.(recordar crear los indices secundarios también)

12.- VISUAL BASIC FILTROS

Otro problema similar al anterior es el de filtros, es decir en muchas ocasiones es necesario obtener información acerca de un conjunto de renglones de la tabla. Por ejemplo todos los estudiantes que sean mayores de 17 años, todos los clientes que sean de Tijuana, etc., a esto le llamamos filtros o condiciones.

Visual BASIC, para resolver este problema, usa un lenguaje especial de consulta de datos, llamado SQL(Structured Query Language), un lenguaje especial que traen todos los DBMS, no es propósito de este libro, ni de este curso enseñar otro lenguaje, solo nos concretamos a analizar una de sus instrucciones mas importantes y especial para resolver el problema planteado en este tema:

SELECT [campos , all, *]

FROM tabla

WHERE condición ;

Ejemplos

1.- Select Nombre, dirección

From Estudiantes

Where carrera = “Informática”;

2.- Select All From Clientes Where Estado = “BC”;

3.- Select * From Estudiantes Where Ciudad = “Tijuana” And Edad > 20 ;

Para construir una aplicación de este tipo solo se ocupan, los dos controles ya analizados:

A) DataControl:

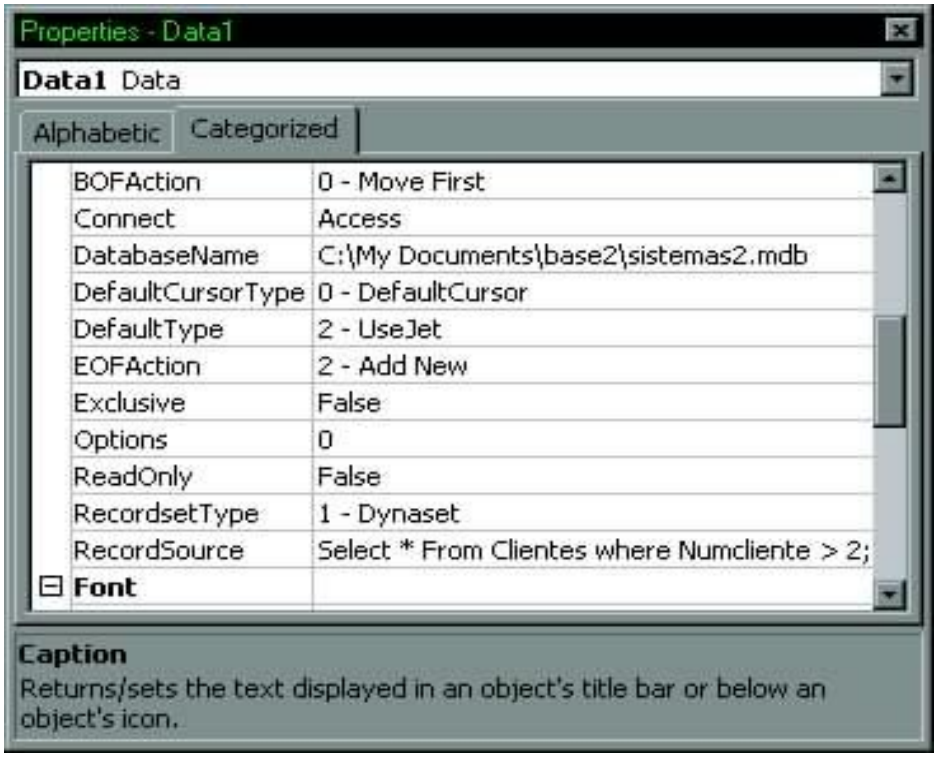
- DatabaseName = buscar y seleccionar la base de datos a usar

-RecordSetType = poner en 1-Dynaset (este tipo de tablas, solo permite consulta no edición, y los filtros en general son para consultas).

-RecordSource = escribir la instrucción select apropiada, ejemplo gráfico

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



B) DBGrid:

* DataSource = Seleccionar Data1

Pantalla de corrida:

Por supuesto que tambien se puede capturar el select en un textbox y en un boton de commando, cargar el recordsource = textbox.

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir tres aplicaciones de búsqueda con tres tablas y selects diferentes

13.- GRÁFICOS O IMÁGENES PICTUREBOX

Campos de gráficos o de imágenes, se han convertido en una de las grandes atracciones y características de las bases de datos modernas.

En Visual BASIC, el manejo de dichos campos es muy fácil solo:

1.- Cuando se crea la tabla con el Visual Data Manager,incluir un campo de tipo de dato Binary, por ejemplo en la tabla de Alumnos, agregar campo foto de tipo Binary.

2.- En aplicaciones por renglones, usar un componente Image o PictureBox.

Si el renglón o registro ya esta cargado con su imagen respectiva, al hacer la navegación entre renglones, este componente Image las va desplegando.

3.- En aplicaciones por tablas, poner a un lado del DBGrid un componente Image, esto hace la misma función, es decir al momento que el usuario navega entre los renglones de la tabla, este componente va desplegando la imagen del renglón donde se encuentra el cursor.

4.- El problema, es como cargar las imágenes, la respuesta es sencilla, las imágenes solo pueden cargarse, capturarse o provenir de solo dos fuentes o lugares distintos:

4.1) Un archivo de tipo *.BMP(u otros formatos similares)

4.2) Del Clipboard de Windows

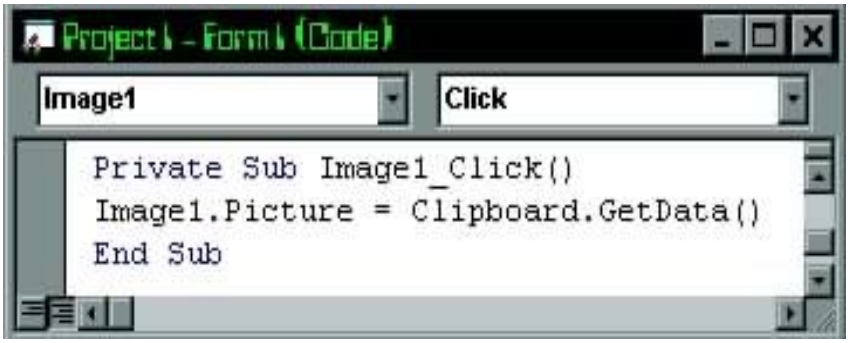
5.- Cualquier dispositivo o periférico de este tipo (scaners, cámaras digitales, etc.) o programas de imágenes (paintbrush, paint, corel, etc.) tienen estas dos capacidades, es decir pueden mandar sus imágenes ya sea al Clipboard o ya sea a un archivo en disco.

6.- Para capturar una imagen en una aplicación ya sea por renglón o por tabla, recordar usar un campo Binary y un control Image, para usar el Clipboard para capturar solo:

6.1- Poner el siguiente código en el evento clic del componente Image:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



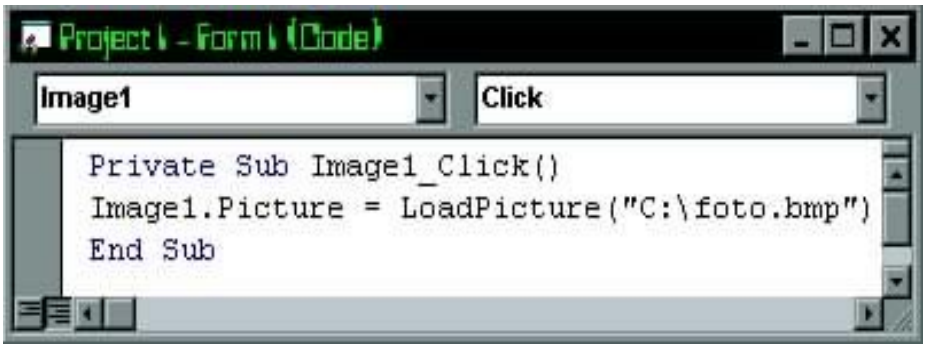
Recordar:

1ro.- El gráfico ya debe estar en el clipboard

2do.- Hacer un clic en la caja Image, al diseñar, ponerle un buen borde para que se note.

3ro.- Cambiarse de renglón, para que se actualice la tabla en disco.

7.- Para cargar o capturar una imagen directamente desde un archivo en disco de extensión .BMP (u otros similares) solo usar un control Image ya sea en aplicaciones por renglón o por tabla, un componente Command de ordenes y el siguiente código:

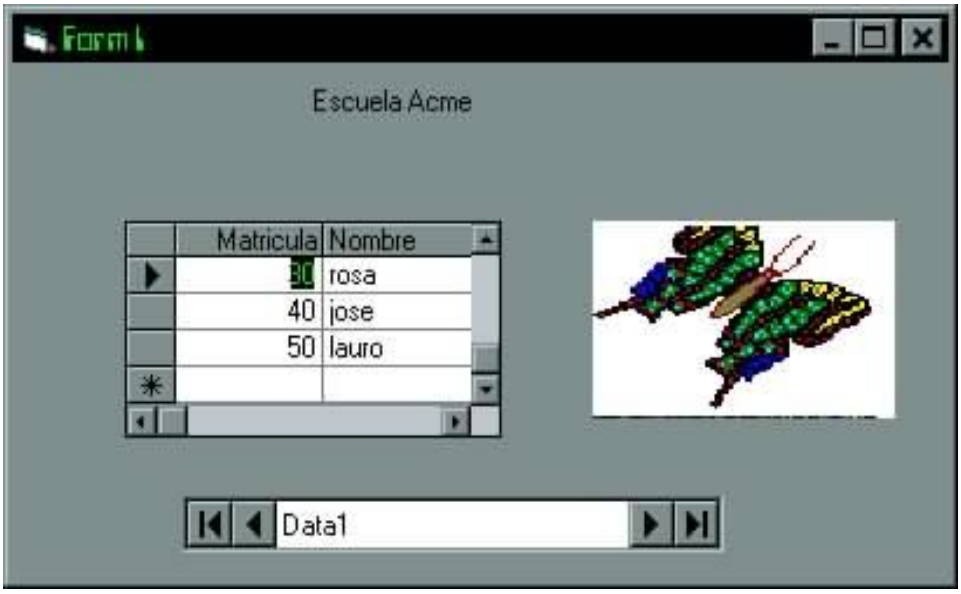


Ya debe estar en el directorio o folder datos dicho archivo .BMP y además otra vez recordar, que aunque la imagen ya esta en la pantalla, todavía no se manda a la tabla a disco si no se ejecuta un avance de renglón o un Move.Next, etc.

8.- La propiedad Stretch en Image, la amplia, otro método es ponerlas en Clipboard o archivo ya con el tamaño apropiado.

Pantalla de Corrida:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



TAREAS PROGRAMACION VISUAL BASIC

1.- Una tabla de mamíferos completa incluyendo imágenes

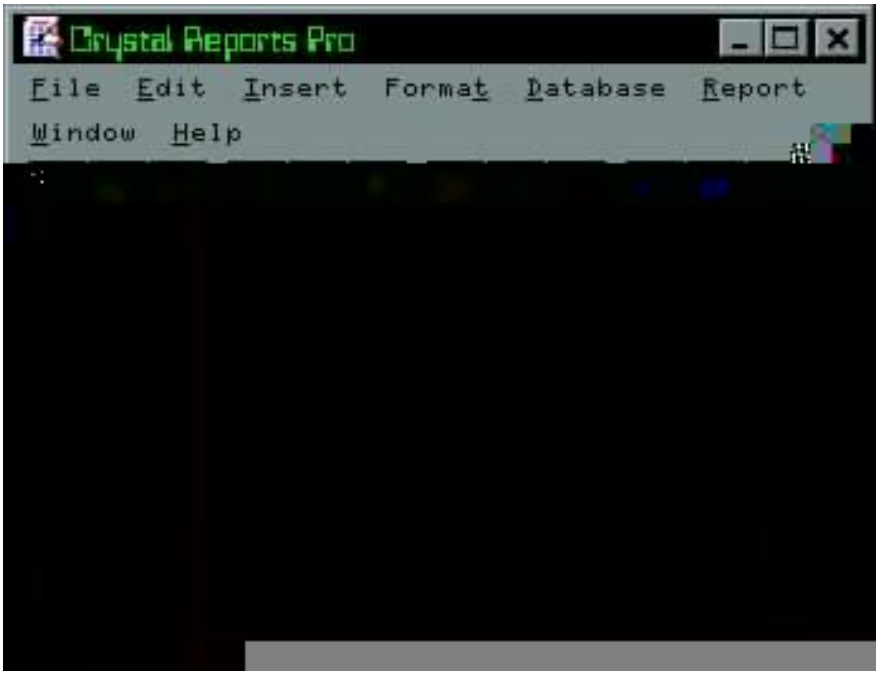
2.- Una tabla de mercancías incluyendo su foto

14.- IMPRESIÓN

Otro problema común, con tablas, es la impresión de las mismas, Visual BASIC proporciona un software especializado en la impresión de reportes, llamado Crystal Report, el procedimiento para usarlo es:

1.- Cargar el reporteador, con clic en Start, Programs, Visual BASIC, Crystal Reports y aparece la siguiente pantalla:

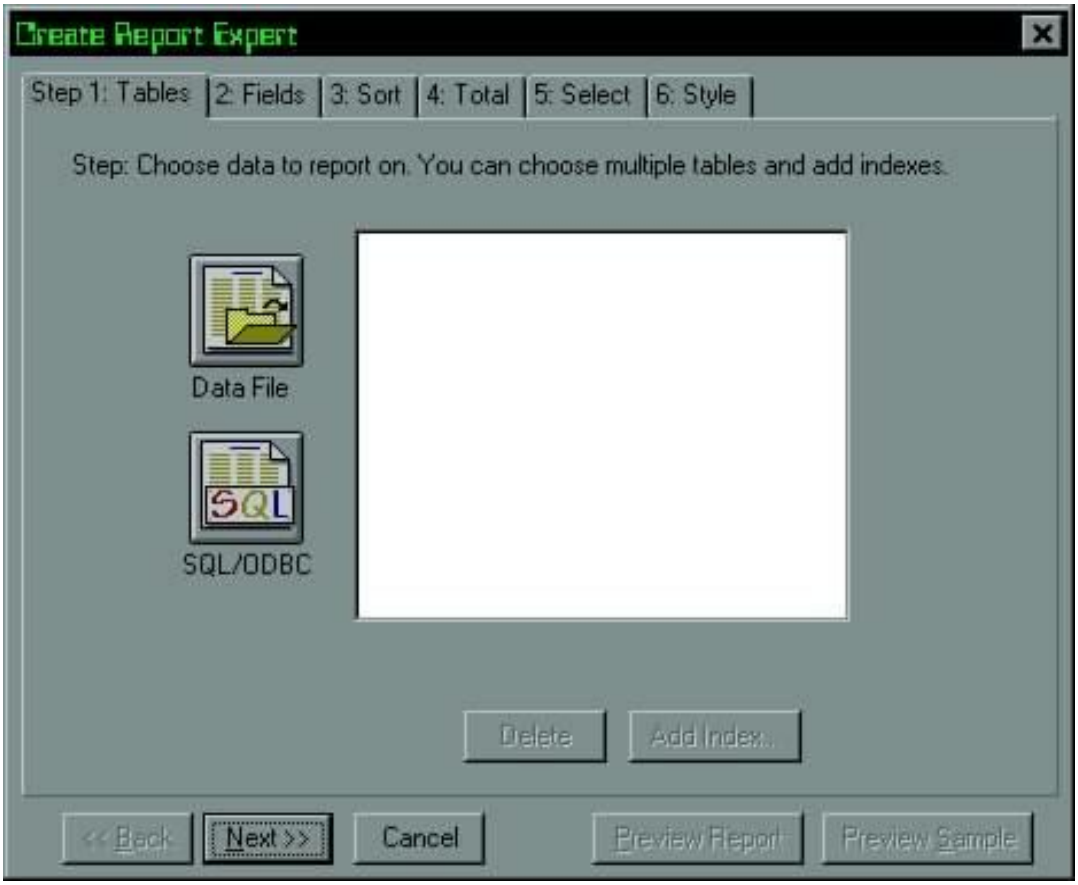
Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



2.- File, New y aparece la siguiente pantalla:

Como se observa, Crystal Report, trae una serie de reportes distintos, apropiados para diversas tareas de impresión, usaremos el formato STANDARD de reporte, para este ejercicio, clic en botón STANDARD y aparece la siguiente pantalla:

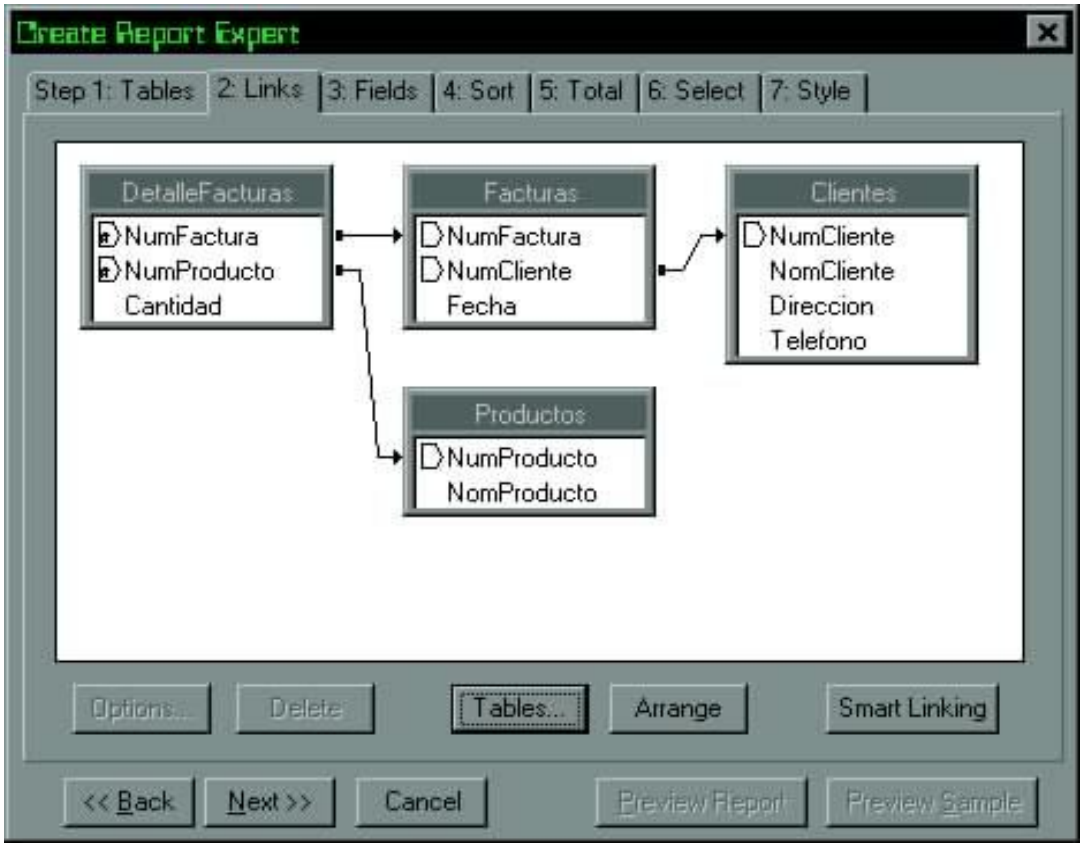
Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



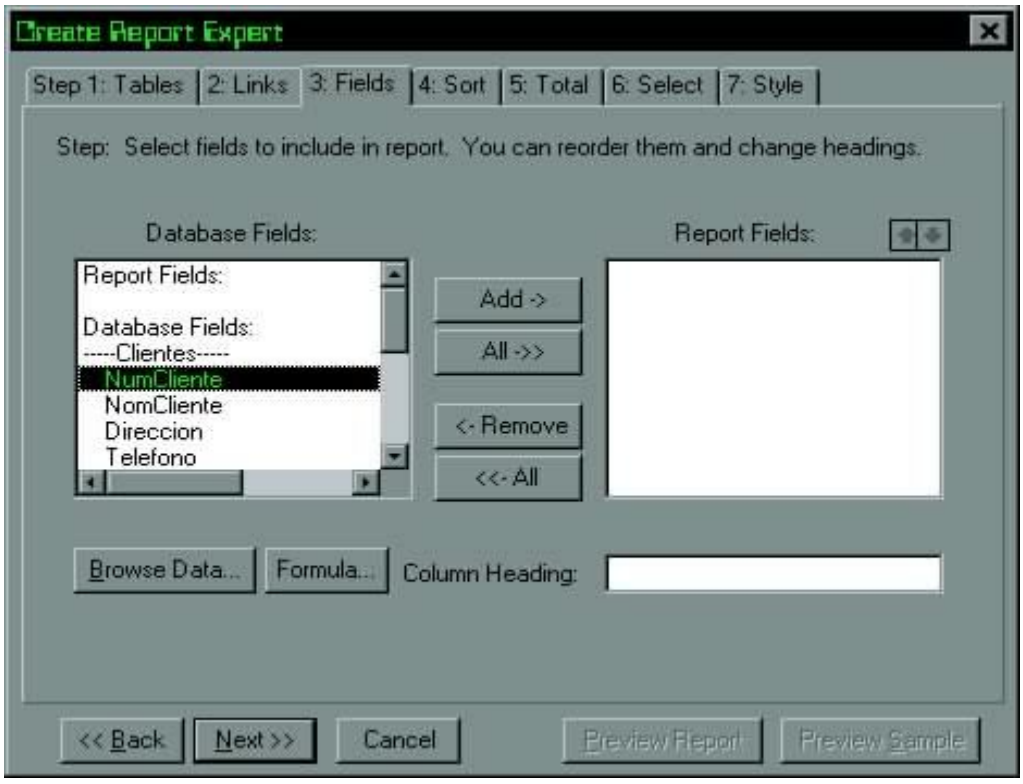
Clic en botón Data File, para seleccionar la base de datos a usar y aparece la pantalla normal de selección de archivos, recordar buscar y seleccionar, base de datos a imprimir:

Después de seleccionar, la base de datos apropiada, usar primero el botón ADD y luego el botón Done y aparece:

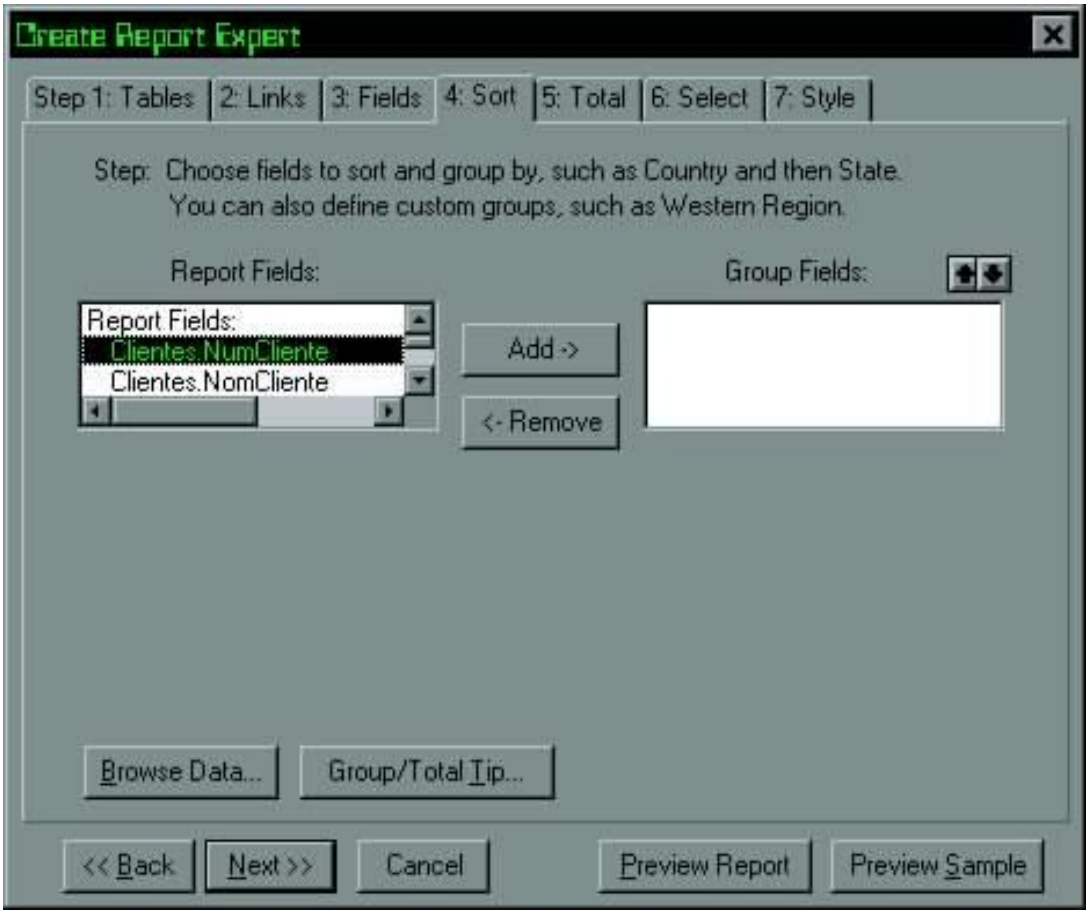
Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Solo usar botón Next, y aparece:

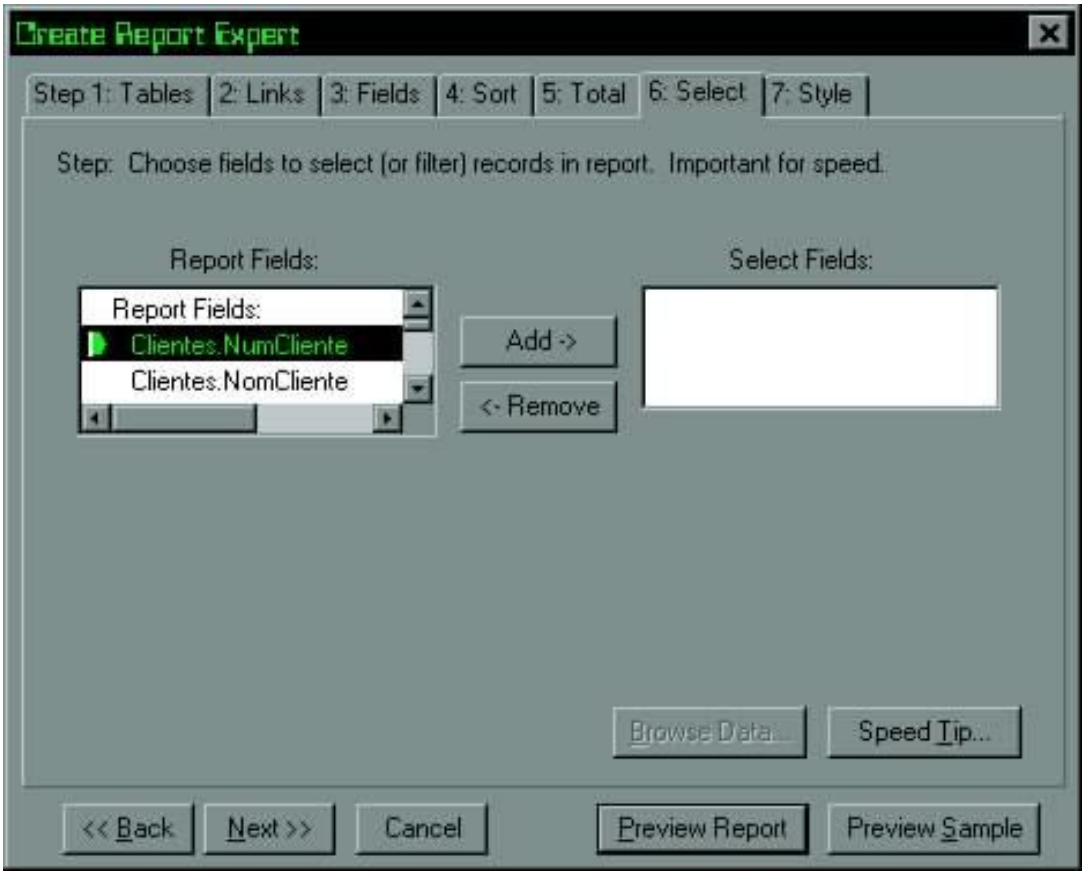


Seleccionar de la ventanilla de la izquierda, los campos o columnas a imprimir, solo clic en la columna y luego en el botón Add, al finalizar usar el botón Next, aparece:



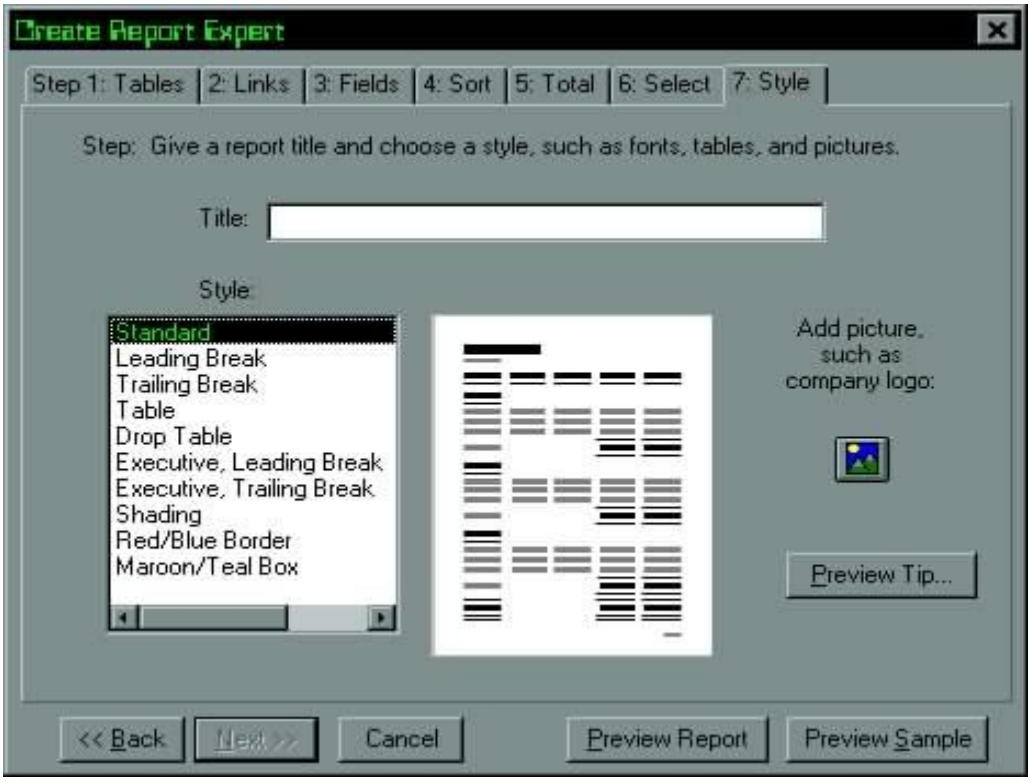
Esta ventana se usa, para que sortear u ordenar, los registros o renglones de la tabla, seleccionar en la ventanilla izquierda el campo a ordenar (numcliente), luego botón Add y al final abajo usar el botón Next, la siguiente pantalla (similar a la gráfica arriba y no mostrada) se usa para poner subtotales o conteos por columnas, seleccionar columnas, Add, Next y la siguiente pantalla es:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Esta pantalla se usa para filtrar la tabla, solo seleccionar el campo a filtrar e ir seleccionando o construyendo el filtro apropiado, luego usar el botón Next, aparece la pantalla final:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es



Escribir el titulo, seleccionar la estilo apropiado, y usar el botón preview, en la pantalla de preview reporte, usted podrá imprimir el reporte usando file, print o el icono de impresión, al final usar el botón close, y en la pantalla normal de grabación se debera guardar el reporte junto con la base de datos y ponerle un nombre apropiado, en este ejemplo ACME.RPT.

2.- El reporte ya esta listo y en la base de datos, el problema a resolver ahora, es como activarlo dentro de una aplicación en Visual BASIC, para esto:

2.1.- Solo se ocupan tres controles, ellos son:

A)DataControl

-DataBaseName = clic, buscar y seleccionar la base de datos apropiada.

B)CrystalReport Control, importarlo al ToolBox, con component, se llama, Crystal Report Control 4.6, ponerlo en una aplicación, donde no estorben aparece en la aplicación al tiempo de la corrida), y sus propiedades son:

-DataSource = Data1

-ReportSource= clic en elipsis(…) y buscar y seleccionar el reporte apropiado (ACME.RPT)

C)Command, con el siguiente código:

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

TAREAS PROGRAMACION VISUAL BASIC

1.-Construir 2 aplicaciones que incluyan uno o dos reportes apropiados.

CUESTIONARIO

1.- Problemas resueltos en esta UNIDAD VISUAL BASIC

2.- Que es variable registro

3.- Que es archivo

4.- Cuales son los dos modelos de datos

5.- Que es aplicación

6.- Que es tabla en el modelo de datos

7.- Cuales son las tres principales reglas al diseñar una tabla

8.- Que es DBMS

9.- Que es Visual Data Manager

10.- Que es llave primaria

11.- Que es llave secundaria

12.- Que es base de datos

13.- Cuales son los dos tipos de aplicaciones

14.- Que es control DataControl

15.- Cuales son las propiedades mas importantes del control DataControl

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

16.- Cuales son las dos propiedades de TextBox usadas en problemas de bases de datos

17.- Cuales son las opciones de navegación que trae el control DataControl

18.- Que es control DBGrid

19.- Cuales son las propiedades importantes del control DBGrid

20.- Que es propiedad RecordSet

21.- Propiedades usadas de RecordSet para procesos con tablas

22.- Que es búsqueda

23.- Formato del método SEEK

24.- Que es filtro

25.- Que es SQL

26.- Formato de la instrucción SELECT

27.- Que es un campo bynary de una tabla

28.- Tipos de archivos gráficos usables en una tabla

29.- Cuales son los dos métodos usados para capturar gráficos en una tabla

30.- Que es Crystal Report

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

UNIDAD V

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

V.- MODELO RELACIONAL DE DATOS

1.- INTRODUCCION

2.- TIPOS DE RELACIONES

3.- MODELO RELACIONAL Y VDM

4.- APLICACIONES CON TABLA DE RELACION

CUESTIONARIO

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

CAPITULO V MODELO RELACIONAL DE DATOS

1.- INTRODUCCIÓN

Entre dos tablas básicas o tablas simples cualesquiera, se debe y puede buscar, identificar y establecer una o varias relaciones entre ellas, ejemplo;

tabla Clientes tabla Productos

R1= El Cliente compra Productos

R2= El Cliente devuelve Productos Dañados

R3= El Cliente aparta Productos

tabla Autos tabla tallermecanico

R1= El auto ingresa al taller

R2= El auto es diagnosticado en el taller

R3= El auto es reparado en el taller

R4= El auto sale del taller

Una relación simple es la unión o combinación de dos tablas básicas mediante una y solo una acción, hecho o conducta especifica.

Entiéndase de otra manera, como una frase que relaciona las dos tablas y un y solo un verbo que las une.

Si se observan y analizan detenidamente las relaciones de los ejemplos, es también posible deducir que un conjunto de relaciones forman o constituyen un proceso administrativo, contable, fiscal, o de otro tipo cualesquiera, en el primer ejemplo el proceso es el de ventas, en el segundo es el proceso de reparación de un auto.

Debe ser obvio que un proceso cualesquiera no se podrá describir completamente, con tres o cuatro relaciones simples nadamas.

Aun mas, en un sistema de información cualesquiera cada una de las relaciones genera una tabla especial llamada “de relación”, pero también genera en muchos casos un documento especifico, por ejemplo el cliente compra al contado productos genera la tabla de relación y el documento llamado “Factura”, en la relación el auto ingresa al taller se genera la tabla de relación y/o documento llamado “ORDEN DE ENTRADA”, en la relación el cliente aparta productos se genera la tabla de relación y/o documento llamado “NOTA O RECIBO DE APARTADO”, etc.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Existirán casos o relaciones donde será casi imposible identificar o nombrar el documento o relación existente, para resolver este problema, existen dos soluciones básicas, la primera de ellas es crear por nuestra cuenta el documento, es decir si en un modelo practico no existe un documento para esta parte del proceso lo mas sencillo es crearlo en la empresa, documentarlo y pedir a la empresa que lo ponga en practica, en algunos casos es también posible o necesario no crear documento alguno, solo llamar a esta relación con el nombre de las dos tablas, por ejemplo rel perros/gatos, rel clientes/productos, etc. (aunque no es recomendable o muy explicativo).

Igualmente es muy recomendable,, al describir un proceso cualquiera y su conjunto de relaciones, no usar o buscar relaciones muy abstractas, porque será casi imposible pasarlas a un modelo de información implementado en computadora, por ejemplo la relación al cliente le gustan los autos, por ejemplo los perros corretean gatos, etcétera.

En resumen las relaciones y en general el proceso deben de ser simples, y documentales.

Para terminar de complicar las cosas un modelo completo de información, no se construye con dos tablas básicas, un par de procesos y una cuantas relaciones o documentos, el modelo completo incluye un montón de tablas básicas, otro montón de procesos diferentes entre ellas, y cada proceso contiene un conjunto amplio de relaciones.

Por ejemplo en una Empresa de “AUTO REFACCIONES”, fácilmente se encuentran las tablas básicas, de clientes, mecánicos, proveedores, partes, proceso de ventas al publico, proceso de compras a proveedores, etcétera y cada proceso con su propio conjunto de relaciones y o documentos.

TAREAS PROGRAMACION VISUAL BASIC

1.- Para una empresa de vídeo, identificar sus tres tablas mas básicas sus dos procesos mas importantes y cuando menos cinco relaciones en cada proceso.

2.- Construir un modelo de tres tablas básicas, dos procesos y cuatro relaciones para una casa de empeño.

3.- Un empresario de éxito, tiene varios lotes para la venta de autos, varios talleres mecánicos para su reparación, vende los autos a crédito y al contado, construir un modelo simple de base de datos relaciónales para su posterior implementación.

2.- TIPOS DE RELACIONES

Analizando a fondo, la mayoría de las relaciones y sus documentos son del tipo llamado 1:M (uno a muchos), es decir cada tabla de relación o documento contiene un renglón de la primera tabla y muchos renglones de la segunda tabla, por ejemplo en una factura solo hay o se encuentra, un cliente y muchos productos, en una receta por

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

ejemplo solo hay un paciente pero hay muchas medicinas, en una orden de compra o de pedido hay un solo proveedor pero hay muchos productos y así sucesivamente.

En general existen los siguientes tipos de relaciones:

a) 1:1 (uno a uno), por ejemplo un recibo de renta , una boleta individual de calificaciones, etc.

b) 1:CTE (uno a constante), por ejemplo una nota de renta de películas, donde la política de la empresa es que solo se rentan hasta cuatro películas.

c) 1:M (uno a muchos), el caso mas común, facturas, notas de ventas, ordenes de salida de almacén, solicitudes o requisiciones de material o de equipo o de personal, etc.

d) M:M (muchos a muchos), cuadros o concentrados, muchos de ellos de tipo fiscal, por ejemplo informes fiscales al gobierno acerca de los muchos clientes que hacen muchas transacciones mayores de $ 10,000.00, informes al gobierno de los muchos distribuidores e importadores de equipo de computo y las muchas transacciones que hacen mensualmente de equipo, etc.

Repitiendo el caso mas común de relaciones, son las de 1:M, en este caso, solamente la tabla de relación no permite resolver el problema de información, en casos de relaciones 1:M se ocupara una segunda tabla llamada detalle-relación.

El modelo completo de tablas para una y solamente una relación de tipo 1:M es;

tabla de 1(uno) tabla de M(muchos)

- Clave de uno - Clave de muchos

* otros campos - otros campos

tabla de R(relación) tabla detallerelacion

- Clave de relación - Clave de relación

- Clave de uno - Clave de muchos

- otros campos - otros campos

Este es el modelo principal de una base de datos relacional y se deberá siempre aplicar cuando se tenga que construir una base de datos que tenga una relación 1:M.

Ejemplo;

Clientes(1) Productos (M)

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

Clave Cliente - Clave Producto

Nombre Cliente - Nombre Producto

Dirección Cliente - Descripción Producto

Teléfono Cliente - Precio Producto

etcétera Cliente

Facturas(relación) Detalle Factura (detalle relación)

Clave Factura - Clave Factura

Clave Cliente - Clave Producto

Fecha Factura - Cantidad Producto

etcétera Factura - Etcétera

ejemplo;

Pacientes (1) Medicinas (muchos)

Clave Paciente - Clave Medicina

Nombre Paciente - Nombre Medicina

etcétera - Presentación Medicina

Recetas (relación) Detalle Receta

Clave Receta - Clave Receta

Clave Paciente - Clave Medicina

Fecha Receta - Dosis

Etcétera - Cantidad

Como se observa en los dos ejemplos, las tres claves (la de uno, la de muchos y la de relación) solo se repiten una sola vez en otra tabla.

De nuevo un problema o sistema de información, que descansa en una base de datos de tipo relacional, y solo contiene una relación de tipo 1:M ocupa cuatro tablas (la de uno, la de muchos, la de relación y la de detalle-relacion).

Si el problema incluye una segunda relación 1:M por ejemplo NotaDeVenta que se deriva de las mismas dos tablas básicas, entonces el modelo completo se resuelve con seis tablas (las dos básicas, las dos de relación y las dos de detalle) y así sucesivamente.

Como se dijo anteriormente un problema sencillo de información por ejemplo el de una tienda de vídeo, ocupa unas cuatro o cinco tablas básicas, unos dos o tres procesos y cada proceso dos o tres relaciones mínimo, entonces el sistema se resuelve con 20 o mas tablas, las quejas sobre este modelo a Codd (creador del modelo relacional de bases de datos).

El caso de relaciones 1:1 se resuelve con solo tres tablas, solo agregar también en la tabla de relación la clave de muchos.

El caso de relaciones 1:Cte también se resuelve con tres tablas, solo agregar en la tabla de relaciones tantas claves de la tabla de constantes como constantes existan en la relación, por ejemplo si la relación es boleta de calificaciones y solo contiene tres materias, entonces se ocupan la tabla de alumnos, la de materias y la tabla de boleta, esta ultima tendría clave alumno, clave material, clave materia2, clave materia3.

El caso de relaciones M:M solo se resuelve localizando o identificando o de plano construyendo, la dos relaciones 1:M originales de donde salió este documento, por ejemplo en el informe al gobierno de todos los distribuidores de equipo de computo y todos los equipos que compraron en el mes, es probable que este informe se construya con las tablas de relaciones de facturas y notas de ventas.

Para propósitos de este curso y libro, solo se profundiza en relaciones 1:M. TAREAS

PROGRAMACION VISUAL BASIC

1.- Construir el modelo de las cuatro tablas para la relación el cliente compra a crédito partes o refacciones en una “refaccionaría”.

2.- Construir el modelo del cliente aparta joyas en una “joyería”.

3.- Construir el modelo completo de clientes y películas con las relaciones, renta películas, devuelve películas, daña películas, compra películas.

4.- Considerando el problema de lotes de autos, autos, clientes, talleres construir modelo para los procesos de compra de autos, y restauración de autos(es cuando el lote compra los autos y les da una arreglada para su venta).

3.- MODELO RELACIONAL Y VISUAL DATA MANAGER

El modelo relacional completo, antes de traspasarlo al VISUAL DATA MANAGER, debe incluir todo lo siguiente:

tabla de uno

Clave de Uno P*

otros campos

tabla de muchos

Clave de Muchos P*

otros campos

tabla de relación

Clave Relación P*

Clave de Uno

otros campos

tabla de detalle

Clave Relación C*

Clave Muchos C*

notas:

P* , significa que dicho campo es una clave o llave primaria, en Visual Data Manager (VDM)recordar que solo se ocupa usar el EDITOR DE INDICES, tal como se observa en el procedimiento mas adelante.

C* , se conocen como claves o llaves compuestas, para crear una clave o llave compuesta solo agregar los dos campos involucrados en el EDITOR DE INDICES.

Párrafos mas adelante, se muestra el procedimiento para crearlos dentro de VDM.

El procedimiento para crear el modelo relacional de bases de datos en VDM, es:

1.- Cargar el Visual BASIC, remove project, y cargar el Visual Data Manager.

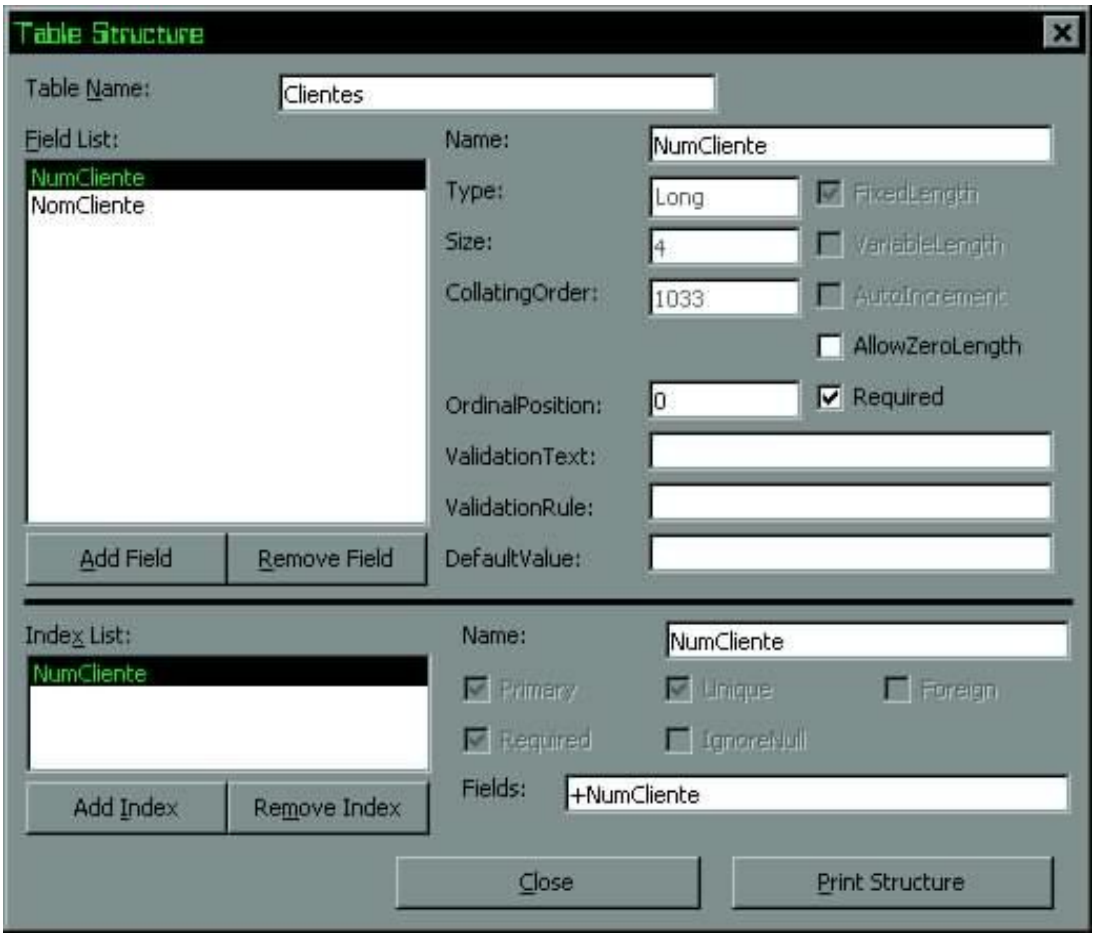
2.- Crear una sistema2 de bases de datos, y crear en propiedades la 4(cuatro nuevas tablas), y cargarlas nada mas con sus campos apropiados.

3.- Crear las llaves primarias y llaves compuestas, para cada tabla, la creación de estos elementos, deberá ser siempre en orden, a continuación se muestra el editor de indices y editor de campos para cada tabla respectiva:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

A) Tabla de uno (Clientes)



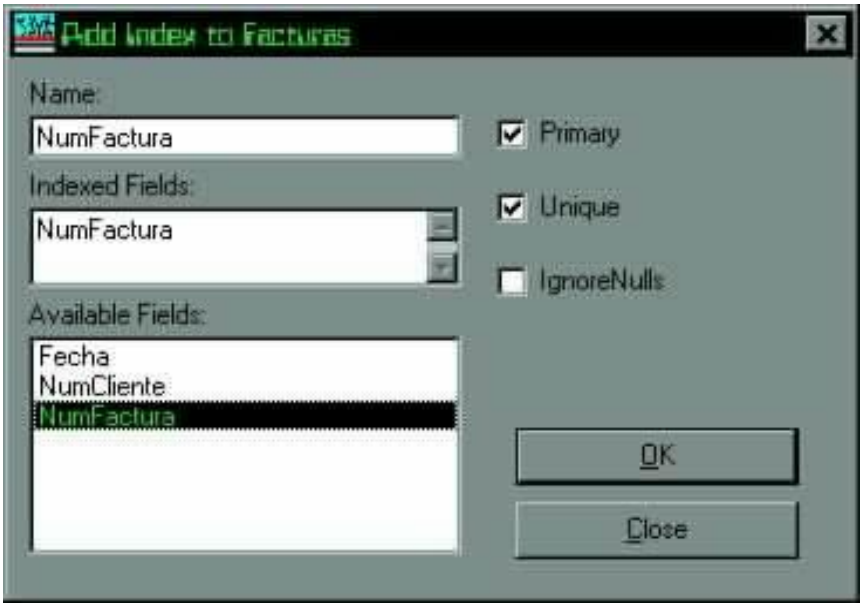
A.1)Editor de Indices(Add Index en grafico anterior)

B) Tabla de muchos (Productos)

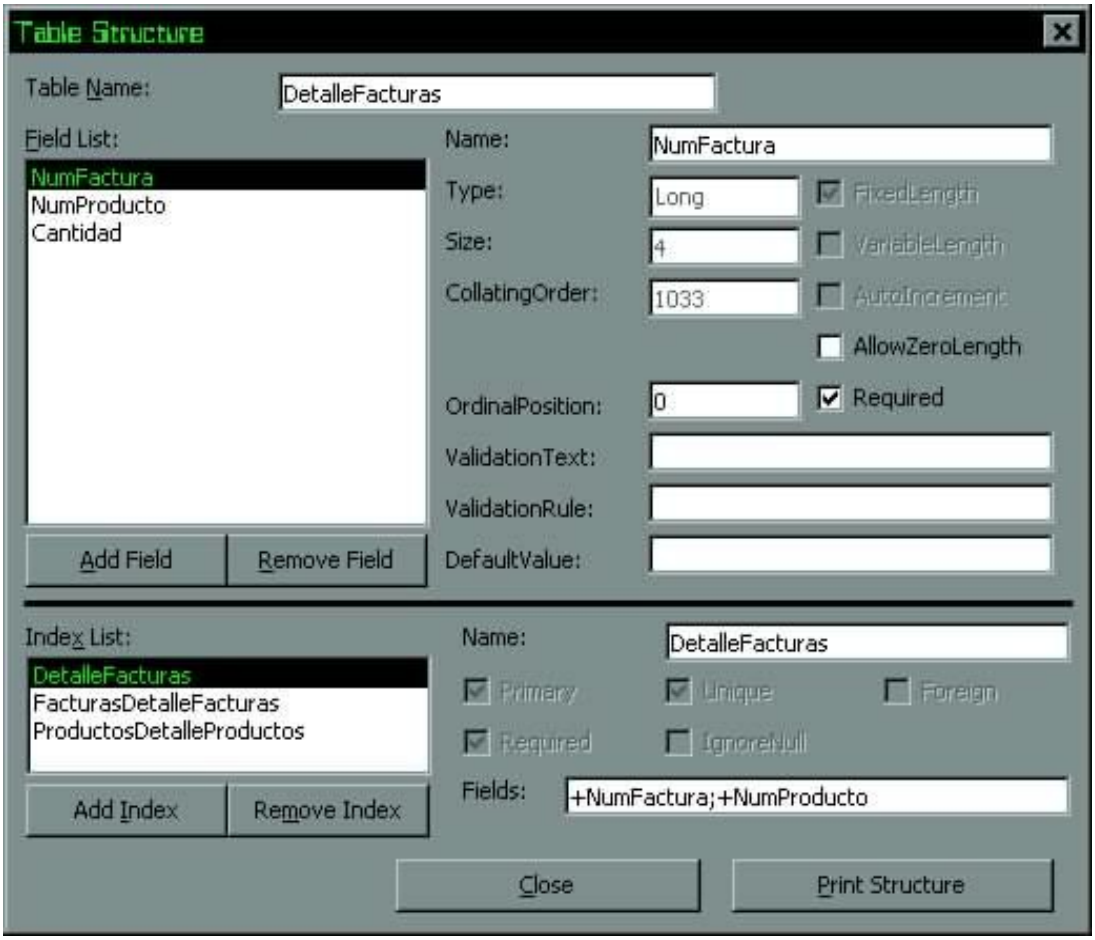
Su editor de tablas es similar al anterior, su editor de indices queda como:

C) Tabla de Relación (Factura)

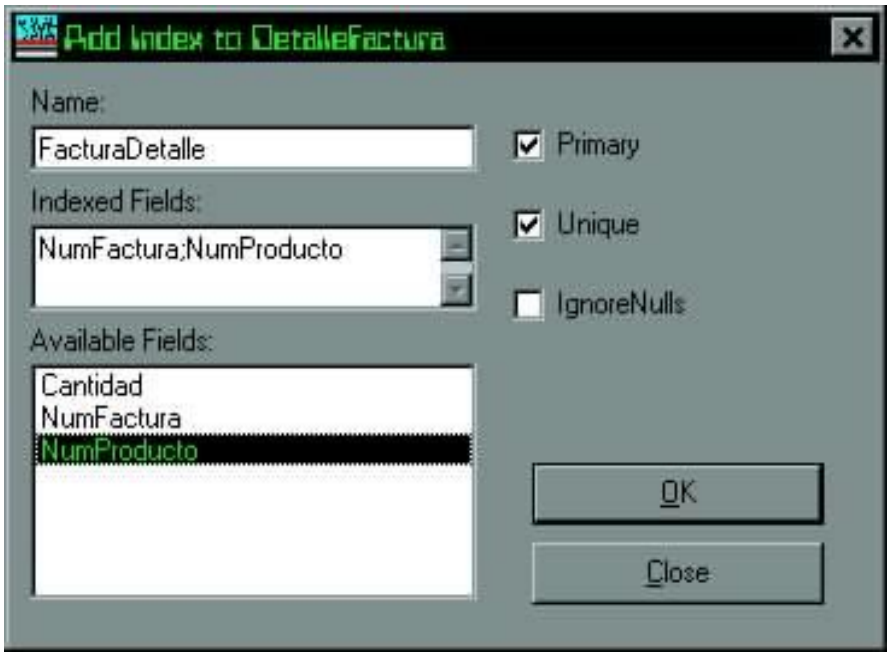
Su editor de tablas es similar a los dos anteriores, su editor de indices es:



D) Tabla de DetalleRelacion(detallefactura), su editor de tabla es:



D.1) Su editor de indices es:



Su llave compuesta es (FACTURA DETALLE), observar que para marcar los dos campos primarios, solo clic primero en NumFactura y luego en NumProducto, al final OK.

Con esto, se termina de crear el modelo de datos en el Visual Data Manager, no olvidar cargar cada tabla con unos cuantos renglones de prueba, para usarlos con la aplicación.

El siguiente paso es construir la aplicación:

TAREAS PROGRAMACION VISUAL BASIC

1.- Construir y cargar en vdm la relación receta(son cuatro tablas).

2.- Construir y cargar en vdm la relación orden de compra.

3.- Construir en vdm la relación de la nota de renta de película.

4.- Construir en vdm la relación recibo de renta(aquí solo son tres tablas como ya se explico.

4.- APLICACIONES CON TABLA DE RELACIÓN

Es tiempo ahora de construir la aplicación final con las tablas construidas con el modelo relacional de bases de datos.

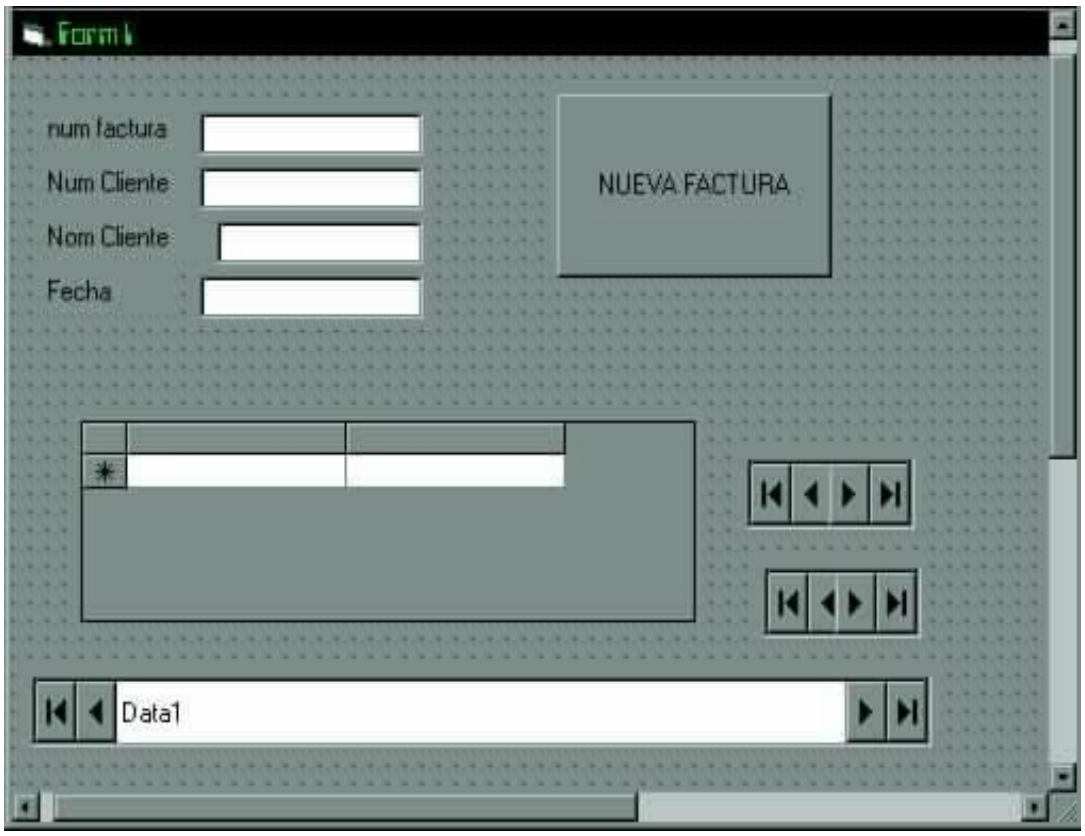
Recordar primero que en general, el usuario solo tendrá acceso y podrá manipular tres de ellas, la tabla de uno, la tabla de muchos y la tabla de relación, las dos primeras de ellas son construidas usando aplicaciones normales por renglón y por tabla, como se realizo en los primeros temas del capitulo, es recomendación y sugerencia que la tabla de uno se construya usando el formato de renglón y la tabla de muchos se construya usando el formato de tabla(DBGrid).

Para construir una aplicación para la tabla de relación:

1.- Observar la siguiente pantalla de diseño:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



Como se observa se ocupan los siguientes controles, deberán ponerse en el orden siguiente.

1.-Data1.- Es el control de abajo, abierto y se usa para navegar las facturas, es el principal y esta enlazando a la tabla de Facturas y a los TextBox apropiados, sus propiedades son:

-DataBaseName= clic en elipsis y seleccionar la base de datos apropiada.

-RecordSource= Facturas

2.-Data2.- Enlaza a la tabla de DetalleFacturas y al DBGrid que los muestra, sus propiedades son:

-DataBaseName.- Clic en elipsis y seleccionar la base de datos apropiada.

-Observar que no usa propiedad RecordSource

3.Data3.- Enlaza la tabla de clientes y el Textbox del nombre del cliente, u otros campos de clientes, atención el Textbox de NumCliente, no deberá ser enlazado a Data3, sino a Data1, sus propiedades son:

-DataBaseName= clic en elipsis y seleccionar la base de datos apropiada.

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es

-RecordSource= Clientes

4.-DbGrid1.- Contendrá el detalle de la factura solo enlazarlo a DATA2, su propiedad es:

-DataSource= Data2

5.-TextBox1.- Muestra el número de factura, propiedades:

-DataSource=Data1

-DataField=NumFactura

6.-TextBox2.-Muestra el numero del cliente, propiedades:

-DataSource=Data1

-DataField=NumCliente

7.-TextBox3.-Muestra el nombre del cliente, propiedades:

-DataSource=Data3

-DataField=NomCliente

8.-TextBox4.-Muestra la Fecha, propiedades:

-DataSource=Data1

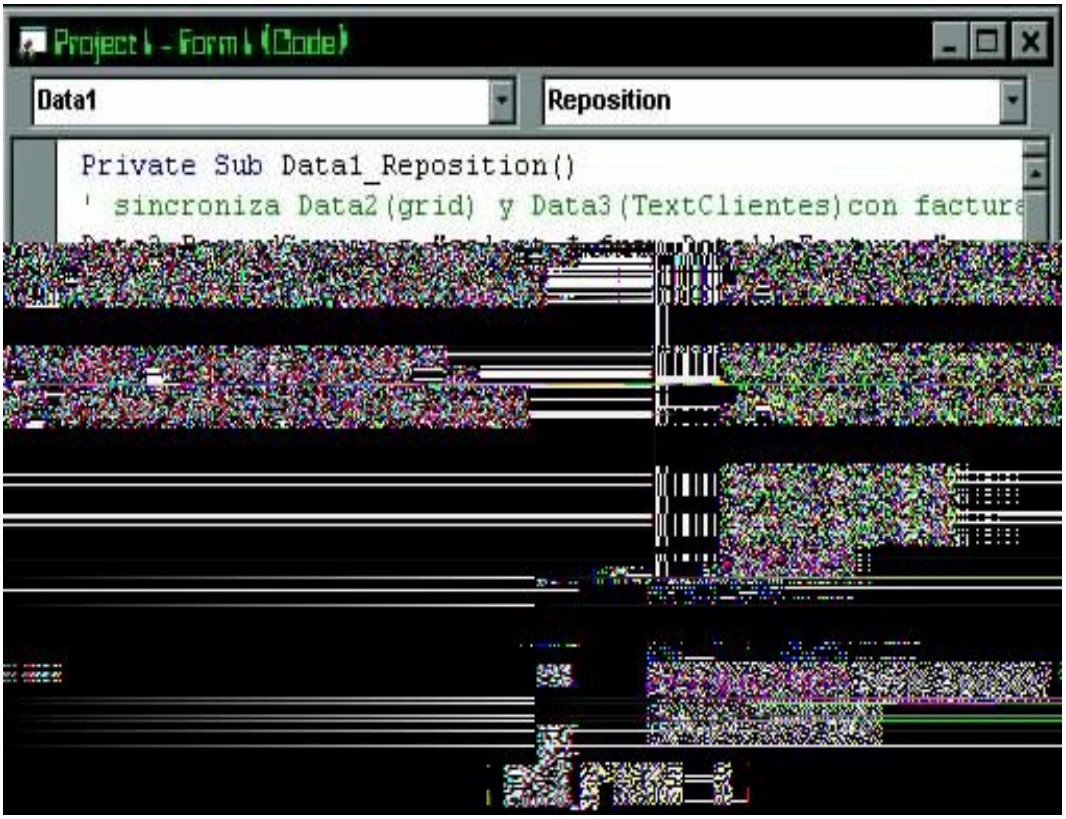
-DataField=fecha

9.-Command1.- este botón, deberá llenarse con código para ocultar la ventana y llamar a las ventanas que capturan todos los datos de una nueva factura, usando lo visto en la UNIDAD VISUAL BASIC anterior.

El Código Completo que usa esta aplicación, es el siguiente:

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es



A) Primero el evento Load() de Form1, se carga con la orden

Data1.Refresh, esto es para que al momento de ejecutarse el programa, aparezcan los datos de la primera factura.

B) Se esta usando también el evento Reposition(), en Data1, porque este evento se activa inmediatamente al cambiarse de renglón o registro el apuntador de registros, esto dentro de la tabla de Facturas, hacia quien esta apuntando Data1.

Observar que se están dando instrucciones selects, dentro del RecordSource de Data2 y Data3, para sincronizar los renglones de las tablas de detalles y de clientes con la tabla maestra o principal de Facturas.

Las ordenes Selects en forma completa son:

1) D2.RS= “select * from DetalleFactura where NumFactura=”

& Data1.Recordset!NumFactura

* Atención todo debe escribirse en el mismo renglón, en el editor de código, los símbolos & se usan para concatenar strings y el símbolo ! Para convertir a single el numero de factura.

* También es posible descomponer un renglón en varios dentro de la ventana de código, usando un espacio en blanco y el símbolo de subrayado(_) al final del renglón,

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

en el renglón siguiente se usa el & para encadenar strings, como en el ejemplo gráfico que se esta mostrando en gráfica anterior.

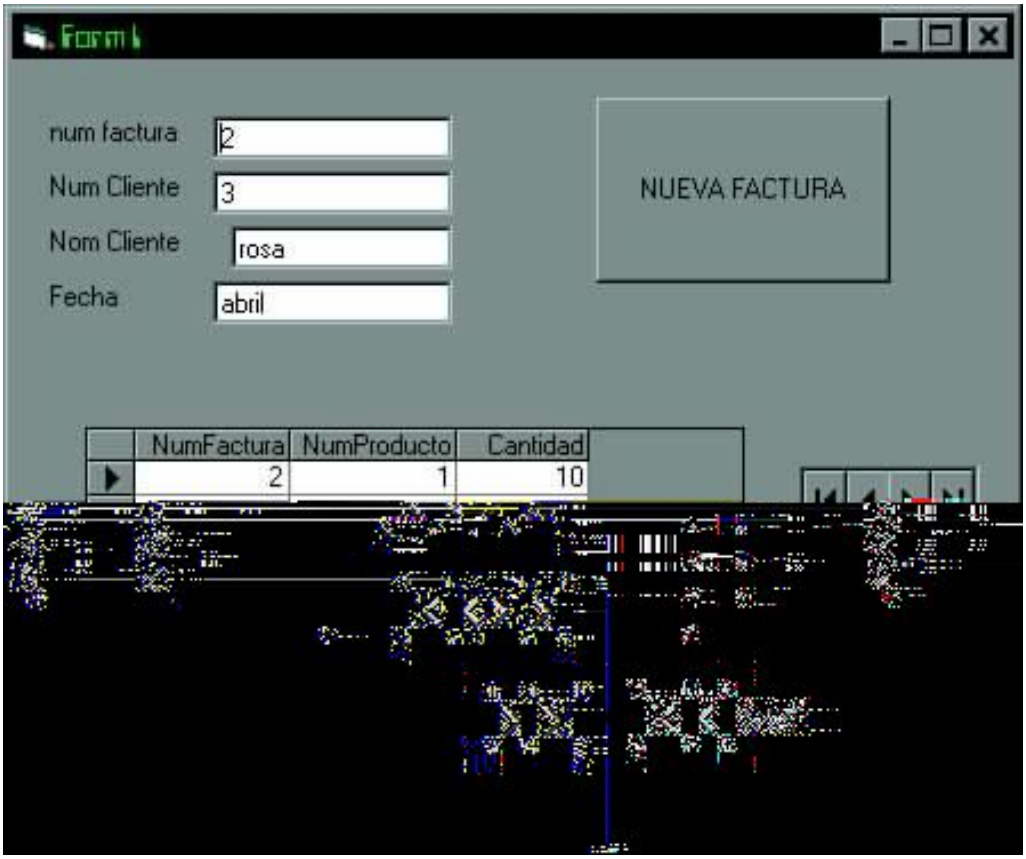
2) D3.RS= “select * from Clientes where NumCliente=”

& Data1.Recordset!NumCliente

* Atención escribir todo en el mismo renglón en el editor de código.

C) Al final se esta usando la orden Refresh, tanto para Data2 como para Data3, para actualizar los Dbgrids y TextBoxs correspondientes.

D) La pantalla de corrida es:



TAREAS PROGRAMACION VISUAL BASIC

1.- Construir un menú que contenga y llame las tablas clientes, películas y recibo de renta de una videotienda

2.- Construir un menú que contenga y llame las tablas pacientes, medicinas y recetas de una farmacia

3.- Construir un menú que contenga y llame las tablas proveedor, productos y orden de compra de una refaccionaría

Lic. Jose Luis Dominguez C.
Joseluisdc10@yahoo.es

CUESTIONARIO

1.- Que es relación

2.- Que es proceso

3.- Que es sistema de información

4.- Cuales son los tipos de relaciones que existen

5.- Grafique el modelo completo de una relación 1:M

6.- Cuantas Tablas se ocupan para resolver una relación 1:1M

7.- Como se resuelve una relación 1:1

8.- Como se resuelve una relación 1:C

9.- Como se resuelve una relación M:M

10.- Que es P*

11.- Que es C*

12.- Mínimo de DataControl's en una aplicación 1:M

13.- Que es evento Reposition

BIBLIOGRAFIA

http://www.programacionfacil.com/visual_basic/start

Lic. Jose Luis Dominguez C.

Joseluisdc10@yahoo.es