**Práctica Seis de visual Basic**
**metas de comprensión**

- Comprende la necesidad del trabajo multitarea en el uso de de software para optimizar el diseño de su trabajo final de síntesis (proyecto de grado)
- Desarrolla comprensión al elaborar programación para su página web empresarial.

## SENTENCIAS

1. **Randomize:** la instrucción **randomize** utiliza number para inicializar el generador de números aleatorios de la función **rnd** asignándole un nuevo valor de inicialización. si se omite number, el valor devuelto por el temporizador del sistema se utilizará como nuevo valor de inicialización.

si no se emplea **randomize**, la función **rnd** (sin argumentos) utilizará el mismo número como valor de inicialización la primera vez que se le llame y más adelante utilizará como valor de inicialización el último número generado.

**ejemplo**

---

en este ejemplo se utiliza la instrucción **randomize** para inicializar el generador de números aleatorios. al haberse omitido el argumento de número, **randomize** utiliza el valor devuelto por la función **timer** como nuevo valor de inicialización.

copiar:

```
randomize()  ('initialize the random-number generator.)
dim value as integer = cint(int((6 * rnd()) + 1))'generate random value between 1 and 6.
```

2. **El control timer**: es un temporizador que nos permite ejecutar instrucciones de código, rutinas, funciones etc..., cada cierto intervalo de tiempo.

este control es **invisible** en tiempo de ejecución, esto quiere decir que no tiene interfaz gráfica, solo es visible cuando lo agregamos a un formulario y estamos en **modo de diseño** la propiedad mas importante de este control es la **propiedad interval**

definición del propiedad interval: devuelve o establece el número de milisegundos entre dos llamadas al evento timer de un control timer. en castellano, esto quiere decir que la propiedad es la que determina el **intervalo** en el que ejecutará las instrucciones que estén en el evento llamado timer, que es el único evento que posee el control.

***ejemplo 1 - utilizando la propiedad interval para mostrar la hora***

coloca un timer llamado **timer1**, seleccionalo, y desde la **ventana de propiedades** en la propiedad **interval** coloca el **valor 1000**. o sea que el timer1 se va a ejecutar cada 1 segundo.

ahora coloca un control **label1** en el formulario y en la propiedad **autosize** del control label1 coloca **true**, esto hará que el control label1 se ajuste al contenido del mismo. lo que hará el ejemplo será mostrar la hora del sistema en el control de etiqueta entonces dale doble click en el timer1 para crear el procedimiento que se describió arriba que es el que ejecuta el timer. y pega está instrucción label1.caption = time, el código del procedimiento quedaría así:

```
privatesubtimer1_timer()
```

```
label1.caption=time

end sub
```

ahora ejecuta el programa y podés ver como se actualiza el label1 de acuerdo a la hora del sistema, usando claro la **función time de visual basic** que devuelve la hora del sistema operativo.

otra propiedad importante y que se utiliza mucho en este control es la propiedad **enabled**. la propiedad enabled lo que hace es **habilitar o deshabilitar un control**. si está en true funciona, si está en false no funciona. esta propiedad es común en la mayoría de los controles de visual basic.

3. **MSGBOX:** la funcion **msgbox** de vbscript, nos permite crear alertas altamente personalizadas en las que podemos configurar no solo el texto, sinó que también el título, el icono, los botones, el icono de la barra de titulo y el botón por defecto.

**valores**

en la siguiente tabla se incluyen los valores de enumeración de **msgboxstyle**:

| miembro | valor | descripción |
|---|---|---|
| **okonly** | 0 | muestra sólo el botón aceptar. |
| **okcancel** | 1 | muestra los botones aceptar y cancelar. |
| **abortretryignore** | 2 | muestra los botones anular, reintentar y omitir. |
| **yesnocancel** | 3 | muestra los botones sí, no y cancelar. |
| **yesno** | 4 | muestra los botones sí y no. |
| **retrycancel** | 5 | muestra los botones reintentar y cancelar. |
| **critical** | 16 | muestra el icono mensaje crítico. |
| **question** | 32 | muestra el icono consulta de advertencia. |
| **exclamation** | 48 | muestra el icono mensaje de advertencia. |
| **information** | 64 | muestra el icono mensaje de información. |
| **defaultbutton1** | 0 | el primer botón es el predeterminado. |
| **defaultbutton2** | 256 | el segundo botón es el predeterminado. |
| **defaultbutton3** | 512 | el tercer botón es el predeterminado. |
| **applicationmodal** | 0 | aplicación modal: el usuario debe responder al cuadro de mensaje antes de continuar trabajando en la aplicación actual. |
| **systemmodal** | 4096 | sistema modal: se suspenden todas las aplicaciones hasta que el usuario responda al cuadro de mensaje. |
| **msgboxsetforeground** | 65536 | especifica la ventana del cuadro de mensaje como ventana de primer plano. |
| **msgboxright** | 524288 | texto alineado a la derecha. |
| **msgboxrtlreading** | 1048576 | especifica que el texto debe aparecer para ser leído de derecha a izquierda en los sistemas árabe y hebreo. |

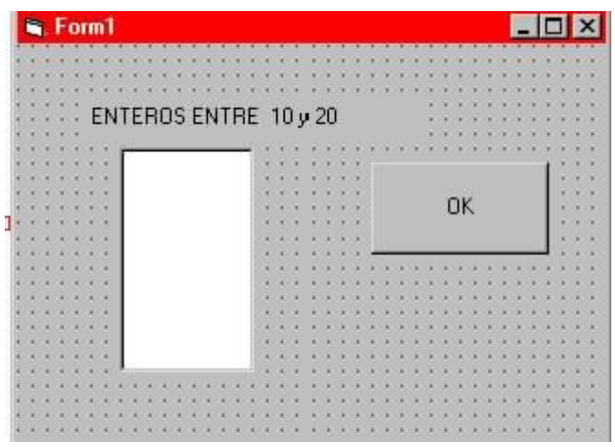
4. **CICLO FOR**: visual basic: este ciclo es uno de los más usados para repetir una secuencia de instrucciones, sobre todo cuando se conoce la cantidad exacta de veces que se quiere que se ejecute una instrucción simple o compuesta. su formato general es: for varciclo=valorinicial to valorfinal [step incr o decr]

**ejemplo para usarlo en los problemas sugeridos mas adelante;**

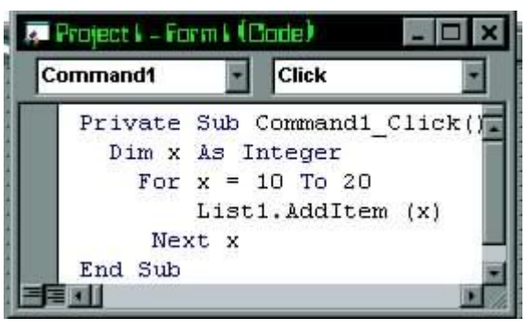
desplegar los números enteros, comprendidos entre el 1 y el 20.

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;

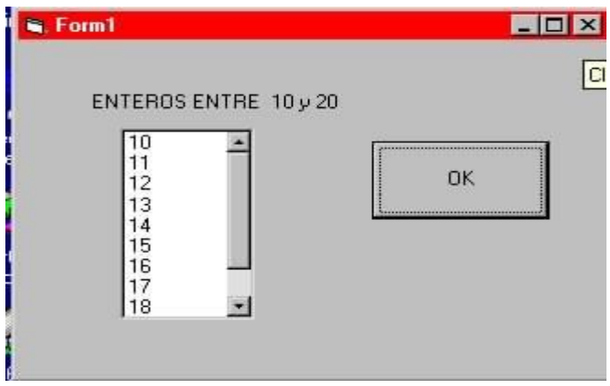
Para este problema se ocupa poner en form1, un componente command1 ok que en su evento click contiene el for y la carga del componente listbox; pantalla de diseño




b) programa




la pantalla de salida es:




este procedimiento y método igualmente trabaja con un componente combobox.

5. **IF:** es utilizada para ejecutar, o no, un bloque de instrucciones de acuerdo con el valor lógico con que fue evaluada la condición.

Sintaxis:
**If** condición **then**
Intruccion 1
Instuccion 2

Instrucción n
**Else**
Otras instrucciones 1
Otras instrucciones 2
Otras instrucciones n
**End if**

Su funcionamiento comienza al evaluar la condición (es), si son verdaderas se ejecutan las instrucciones iníciales y si estas se cumplen el programa salta la línea **end if,** la cual indica el final de la sentencia **if.** En caso contrario se evalúan las otras condiciones que se establecen en el bloque **Else** en donde se ejecutan las otras instrucciones.

6. **Select case:** estructura de control para ejecutar un bloque de instrucciones solo cuando el resultado de la comparación de dos expresiones coincida. Es importante mencionar que las instrucciones asociadas al bloque **CASE** de la expresión coincidente se ejecutaran una vez, y que el control del programa pasara a la siguiente líneas. **END SELECT.**

Sintaxis:
Select case Expresión X (variable numérica, cadena de caracteres, condiciones)
**Case** expresión 1
Instruccionloque1
**Case** expresión 2
Instruccionloque1
**Case** expresión 2
Instruccionloque1 (lista de posibilidades que puede tomar X)
**Case Else**
Nstruccionloque Else (serie de instrucciones que permite ejecutar las instrucciones a X en caso de no cumplir con las primeras)

**END SELECT**

7. **Sentencia DO:** estructura de control que, al igual que la sentencia FOR, genera un ciclo repetitivo; la diferencia radica en que para ejecutar las instrucciones desde evaluarse un condición.

Sintaxis:

**Do While** condición (variable numérica, cadena de caracteres, condiciones)
Instruccionloque1

**Do Until** condición (variable numérica, cadena de caracteres, condiciones)
Instruccionloque1

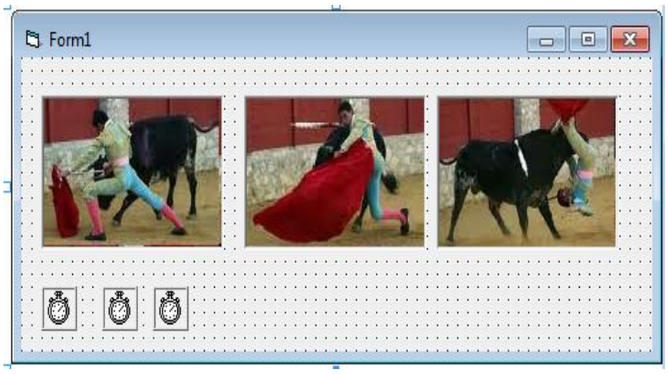
**Exit Do (poprociona una slida para el ciclo sin necesidad de cumplir la condición.)**
Instruccionloque1
**Loop**

8. **La función Val** deja de leer la cadena de caracteres inscritos en un text y lo convierte en un valor numérico en caso de que el primer carácter que no puede reconocer como parte de un número se debe tener en cuenta los símbolos y caracteres que se suelen considerar parte de valores numéricos, como signos de moneda y comas, no se reconocen. Sin embargo, la función reconoce los prefijos de base &O (para octal) y &H (para hexadecimal). Los espacios en blanco, los tabuladores y los avances de línea se eliminan del argumento.
   La función **Val** sólo reconoce el punto (**.**) como separador decimal válido.
   Sintxis:
   Val (tetx) (se puede combinar con los operadores aritméticos que se desee)

**Elaboraremos algunas aplicaciones con algunas de la sentencias explicadas anteriormente, manos a la obra.**

***Creando una animación sencilla:***

***1.*** Baja tres figuras que representen una secuencia.
***2.*** Ingrese al programa visual Basic 6.0
***3.*** En el formulario ingrese tres timer y tres picturebox.
***4.*** En cada picturebox inserte cada una de las imágenes que descargo para la secuencia. (debe quedar si).




***5.*** Los picturebox deben quedar igual de grandes (ajustar el height, width, top y el left).
***6.*** Vamos a programar cada uno de los timer.
***7.*** Haga doble clic en Timer1 y escriba

```
Picture1.Visible = True
Picture3.Visible = False
Timer2.Enabled = True
Timer1.Enabled = False
```

***8.*** Haga doble clic en Timer2 y escriba

```
Picture2.Visible = True
Picture1.Visible = False
Timer3.Enabled = True
Timer2.Enabled = False
```

***9.*** Haga doble clic en Timer3 y escriba

```
Picture2.Visible = False
Picture3.Visible = True
Timer1.Enabled = True
Timer3.Enabled = False
```

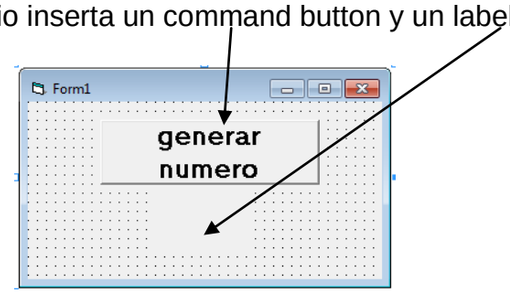
*10.* Haga doble clic en el formulario y en el evento **LOAD** escriba

```
Timer2.Enabled = False
Timer3.Enabled = False
```

*11.* Ubica un picturebox encima del otro.
*12.* Presione F5 y vera una pequeña animación

**Generando números aleatorios:**

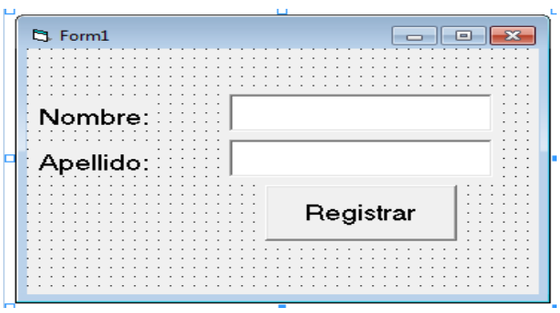
1. En el mismo formulario inserta un command button y un label.




2. Haga doble clic en el formulario y en el evento **LOAD** escriba.
   *Randomize*
3. Haga doble clic en el command button y en el evento CLIC escriba
   Label1.Caption = Int(-40 * Rnd + 40)
4. Presione F5 y vera que se van generando números desde el 1 al 40 en forma aleatoria

**Ingresando el nombre:**

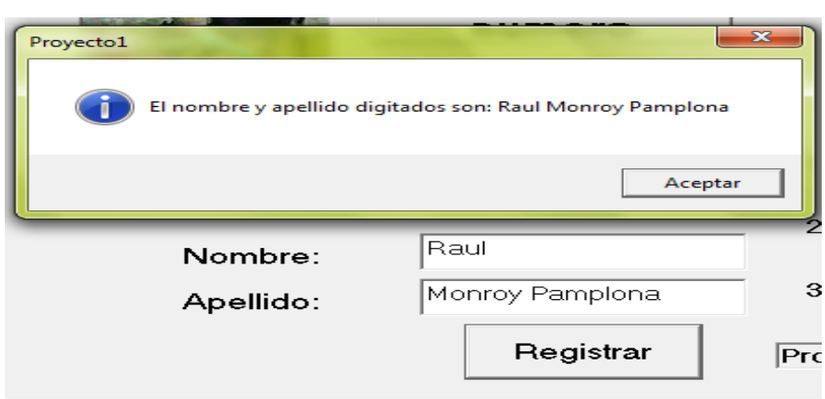
**1.** En el mismo formulario inserta 2 label, dos text box y un command button asi:




2. Deje los text box limpios
3. Vamos a trabajar la propiedad vbInformation del msgbox que nos hace aparecer un mensaje con los datos digitados, y el botón aceptar.
4. Haga doble clic en el botón Registrar y digite las siguientes líneas.

MsgBox "El nombre y apellido digitados son: " + Trim(Text1) + " " + Trim(Text2), vbInformation

5. La expresión Trim permite recoger la información digitada en el text y mostrarla en el cuadro de dialogo.
6. Presione F5 y saldrá un mensaje como el siguiente y con los datos digitados.



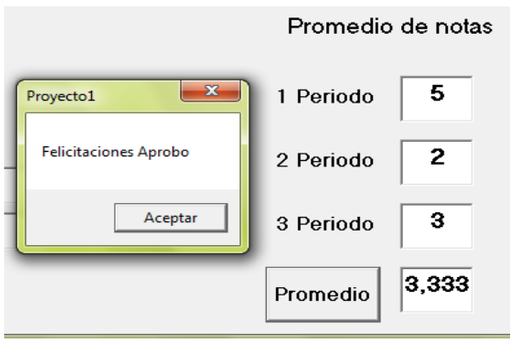
**Promedio de Notas:**

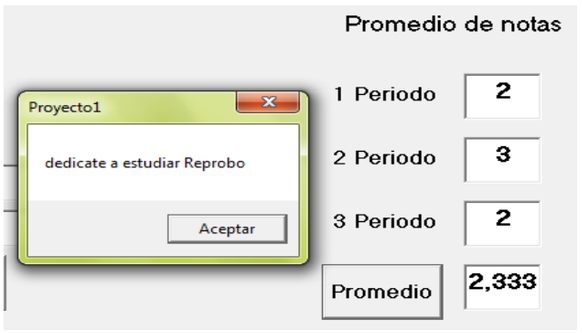
1. En el mismo formulario inserte cuatro label, cuatro text box y un command button como aparece a continuación.


2. Limpie los text box
3. Haga doble clic en el botón promedio y digite las siguientes líneas.

```
Text6 = (Val(Text3) + Val(Text4) + Val(Text5)) / 3
If Text6 >= 3 Then
MsgBox "Felicitaciones Aprobo"
Else
If Text6 < 3 Then
MsgBox "dedica a estudiar Reprobo"
End If
End If
```

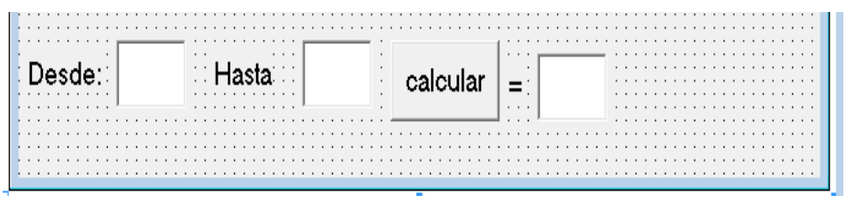
4. Verifique el número del text box para evitar errores en los procedimientos.
5. Presione F5 y corra el programa y vera el cálculo de acuerdo a las notas que usted digite así como el mensaje correspondiente.




**Suma de números (serie consecutiva):**

1. En el mismo formulario inserte tres label, tres text box y un command button asi:




2. Limpie los text box

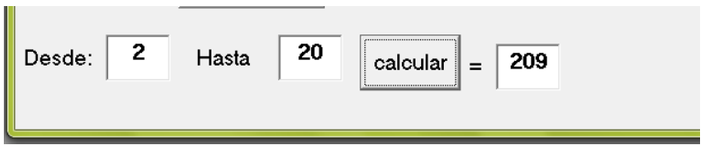
3. Haga doble clic en el formulario y en el objeto **GENERAL** evento declaraciones digite las siguiente líneas que como ya se explico se están declarando las variables necesarias para el programa.

```
Dim n1, n2 As Integer
Dim suma As Double
```

4. Cierre la ventana de código
5. Haga doble clic en el botón calcular y escriba las siguientes líneas.

```
n1 = Text7
n2 = Text8
For n1 = n1 To n2     (desde hasta)
suma = suma + n1
next  n1              (siguiente numero)
Text9 = suma          (asignele a tex el valor de la suma)
```

6. Presione F5 y digite el numerito inicial y luego el número final y haga clic en calcular y obtendrá el resultado de la suma.




Al presionar F5 y ejecutar cada aplicación el formulario debe verse así

***Como están tan pilo y dedicado al trabajo con Visual Basic le propongo las siguientes situaciones problemicas para darle solución mediante aplicaciones sencillas.***

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

2.- construir un programa que despliegue la tabla de multiplicar que el usuario indique.

3.- Crear un programa que permita calcular el valor del **IVA, TOTAL A PAGAR, LAS VUELTAS** y a la vez pueda escoger la forma de pago (efectivo y/o tarjeta de crédito) para un cliente que realiza una compra.

4.- En una oficina de bienes raíces ofrecen casas de interés social bajo las siguientes condiciones:

- Si los ingresos del comprador son menores de $1500000, la cuota inicial será del 15% del costo de la casa y el resto se distribuye en cuotas mensuales a pagar durante 10 años.
- Si los ingresos del comprador son mayores de $1500000, la cuota inicial será del 30% del costo de la casa y el resto se distribuye en cuotas mensuales a pagar durante 7 años.

**La compañía desea conocer el valor de la cuota inicial de la casa así como el valor de la cuota mensual en cada caso.**