



LABORATORIO 4

Escribir el CÓDIGO y la INTERFAZ GRÁFICA (cuando proceda)
de los correspondientes casos prácticos.

NOTA: Según lo acordado en el laboratorio anterior, seguir guardando toda la documentación de estos laboratorios en la carpeta

LAB FUND INFOR QUIMICA 07-08 XXXX XXXX

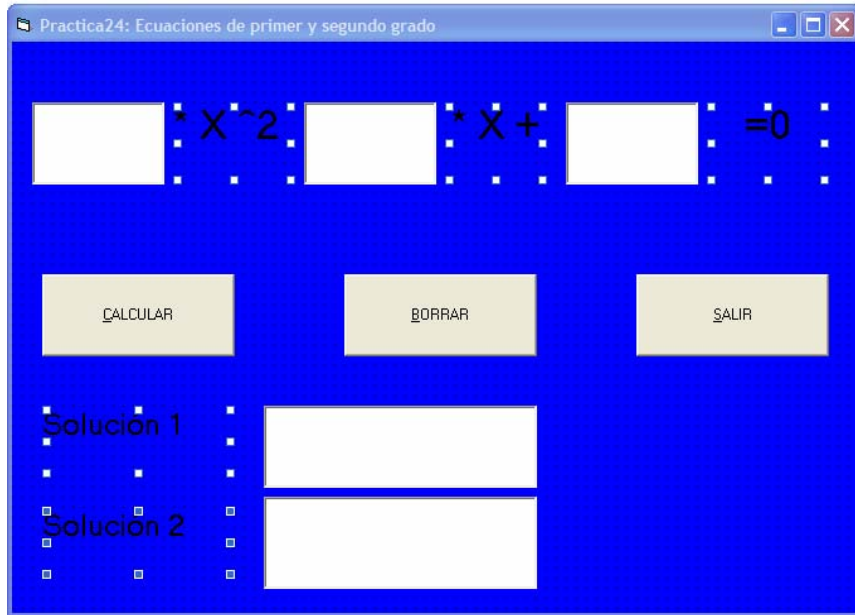
OBJETIVOS LABORATORIO 4

- Cálculo de ecuaciones de primer y segundo grado
- Generación de números aleatorios. Función RANDOM.
- Uso de **IF ... ELSE anidadas**.
- Iniciación a la función PRINT
- Iniciación a la función LEN para calcular la longitud de una cadena
- Introducción a la **sentencia repetitiva FOR**



PRACTICA 24: Ecuaciones de primer y segundo grado

Abrimos un nuevo proyecto (Practica24.vbp) y dentro del mismo creamos un formulario con 3 botones, 5 cajas de texto y 5 etiquetas:



The screenshot shows a Windows form titled "Practica24: Ecuaciones de primer y segundo grado". The form has a blue background. At the top, there is a text label "X^2" followed by a text box, then "X +", a text box, and "=0". Below this, there are three buttons: "CALCULAR", "BORRAR", and "SALIR". At the bottom, there are two labels: "Solución 1" and "Solución 2", each followed by a text box.

INTERFAZ DE USUARIO

Formulario (nombre interno): frmPractica24.frm.

Mensaje de título: Práctica 24 : Ecuaciones de primer y segundo grado

Botones (nombre interno): cmdCalcular, cmdBorrar, cmdSalir

Labels (nombre interno): lblCoefA, lblCoefB, lblIgual, lblSolucion1, lblSolucion2

TextBoxes (nombre interno): txtCoefA, txtCoefB, txtCoefC, txtSolucion1, txtSolucion2

ENUNCIADO DEL EJERCICIO

Diseñar e implementar una aplicación que permite resolver ecuaciones de primer y segundo grado.

- Cuando el usuario pulse el botón **Calcular**, la aplicación, en función de los coeficientes introducidos por el usuario, determina el tipo de ecuación y calcula los resultados. Si la ecuación es de primer grado, para no confundir al usuario



oculta los campos asociados a la segunda solución y modifica la etiqueta asociada a la primera solución

' Ecuación de primer grado

lblSolucion1.Caption = "Solución:"

lblSolucion2.Visible = False

txtSolucion2.Visible = False

Si la ecuación es de segundo grado, se restablecen los campos

' Ecuación de segundo grado

lblSolucion1.Caption = "Solución 1:"

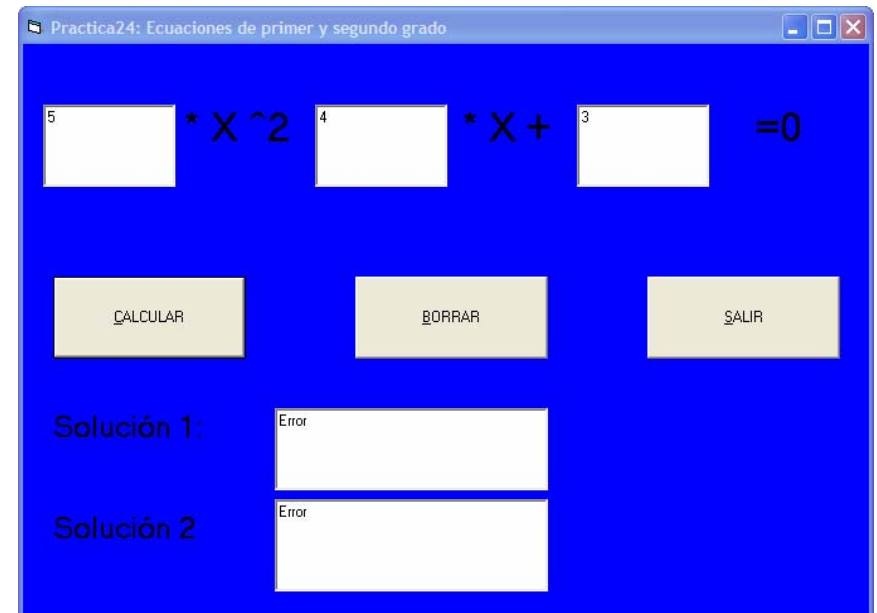
lblSolucion2.Visible = True

txtSolucion2.Visible = True

Cuando haga click en el botón Borrar , se debe borrar el contenido de las Caja de Texto (Siempre cambiando el valor de las propiedades)

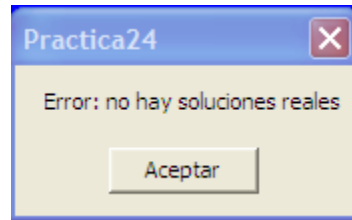
b) Cuando haga **click** en el botón **Salir** , el programa **finalizará**. Para ello usaremos la función **End**.

Así al introducir los siguientes valores por ejemplo

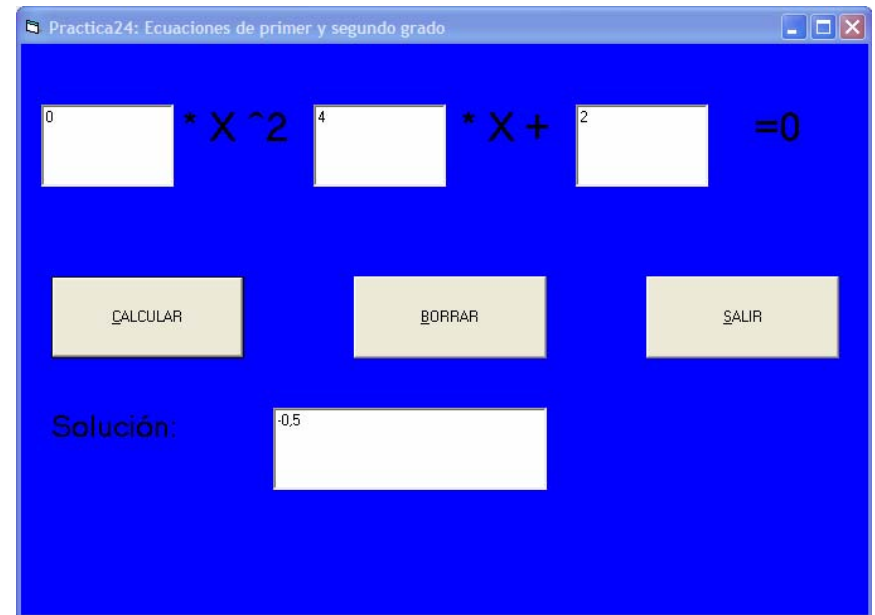




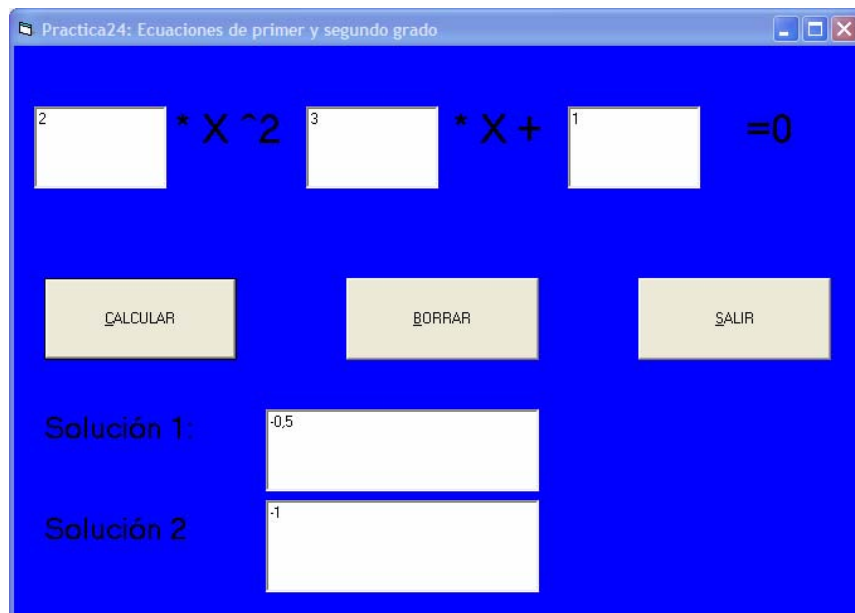
... y dar a Calcular, aparece el mensaje:



Al introducir estos otros valores:



Al introducir estos otros valores:





Al introducir estos otros valores...

Practica24: Ecuaciones de primer y segundo grado

0 * X² 0 * X + 4 = 0

CALCULAR BORRAR SALIR

Solución: -0.5

Y por último, al introducir:

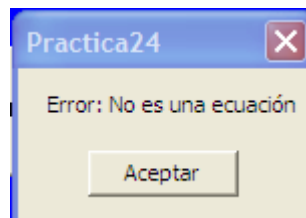
Practica24: Ecuaciones de primer y segundo grado

0 * X² 4 * X + 0 = 0

CALCULAR BORRAR SALIR

Solución: 0

... aparece el mensaje:

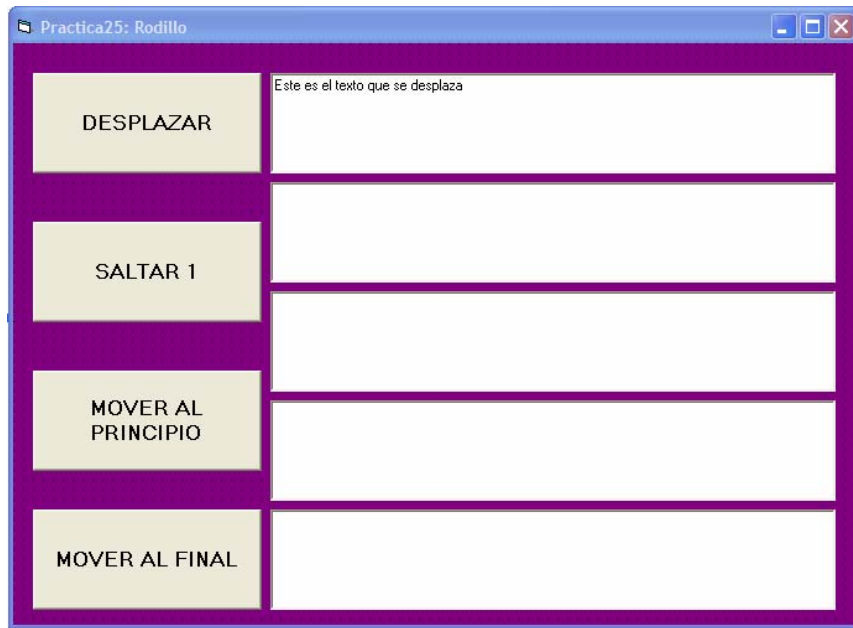


CÓDIGO
 (...)



PRACTICA 25: Rodillo

Abrimos un nuevo proyecto (Practica25.vbp) y dentro del mismo creamos un formulario con 4 botones y 5 cajas de texto :



The screenshot shows a Visual Basic 6.0 form titled "Practica25: Rodillo". The form has a purple border and contains a vertical stack of four buttons on the left and five text boxes on the right. The buttons are labeled "DESPLAZAR", "SALTAR 1", "MOVER AL PRINCIPIO", and "MOVER AL FINAL". The text boxes are arranged in a column, with the top one containing the text "Este es el texto que se desplaza".

INTERFAZ DE USUARIO

Formulario (nombre interno): frmPractica25.frm.

Mensaje de título: Práctica 25 : Rodillo

Botones (nombre interno): cmdDesplazar, cmdSaltar, cmdMoveralprincipio, cmdMoveralFinal

TextBoxes (nombre interno): txt1, txt2, txt3, txt4, txt5

ENUNCIADO DEL EJERCICIO

La aplicación contiene 5 cuadros de texto **no editables** (*Enabled = false*). El primero de ellos contiene el texto “*Este es el texto que se desplaza*”. Los botones de la ventana permiten desplazar el texto de la siguiente manera:

- **Botón Desplazar:** El texto se desplaza a la siguiente posición. Si el texto se encuentra en la última posición, éste pasa a la primera.
- **Botón Saltar 1:** El texto se desplaza dejando un hueco entre la posición que ocupaba y la nueva posición. De



nuevo, el desplazamiento es circular, si llega al final vuelve a desplazar por arriba.

- **Mover al principio:** Coloca el texto en la primera casilla.
- **Mover al final:** Coloca el texto en la última casilla.

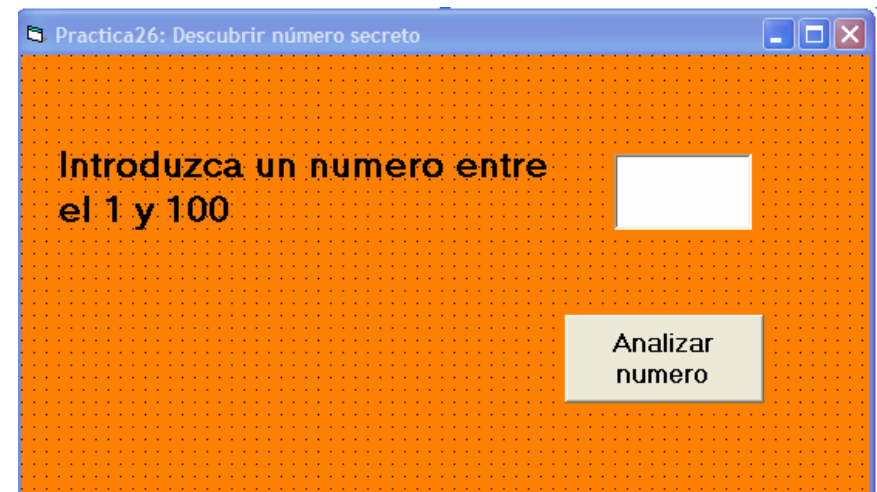
Nota. Utilizar las sentencias condicionales para determinar en que casilla se encuentra el texto y así decidir la nueva ubicación de éste.

CÓDIGO

(...)

PRACTICA 26: Descubrir número secreto. Números aleatorios RANDOM

Abrimos un nuevo proyecto (Practica26.vbp) y dentro del mismo creamos un formulario con 1 label, 1 caja de texto y un botón:



INTERFAZ DE USUARIO

Formulario (nombre interno): frmPractica26.frm.

Mensaje de título: Práctica 26 : Descubrir número secreto.



Botón (nombre interno): cmdAnalizar
Label (nombre interno): lblIntroducir
Textbox (nombre interno): txtNumero

ENUNCIADO DEL EJERCICIO

Este programa genera un “número secreto” aleatorio (que el usuario desconoce) entre 1 y 100 y a continuación le pide al usuario que introduzca números en la caja de texto hasta que acierta el número generado. Cada vez que el usuario introduzca un número la aplicación le indicará si ha acertado o si el número introducido es mayor o menor al número que debe acertar., mediante las correspondientes MsgBox.

Ejemplo:

Introduzca un número entre el 1 y 100: 34

Fallo: El número 34 es mayor que el número a
acertar

Introduzca un número entre el 1 y 100: 26

Correcto!!!!

A continuación se describen las funciones utilizadas para la generación de números aleatorios:

- **Rnd():** Genera un número decimal entre 0 y 1 de manera aleatoria.
- **Randomize:** La función *Rnd* utiliza un valor inicial para generar números aleatorios mediante una serie de operaciones matemáticas.

La función *Randomize* permite generar un nuevo valor cada vez que se ejecute el programa. De lo contrario la secuencia de valores aleatorios generados será siempre la misma.

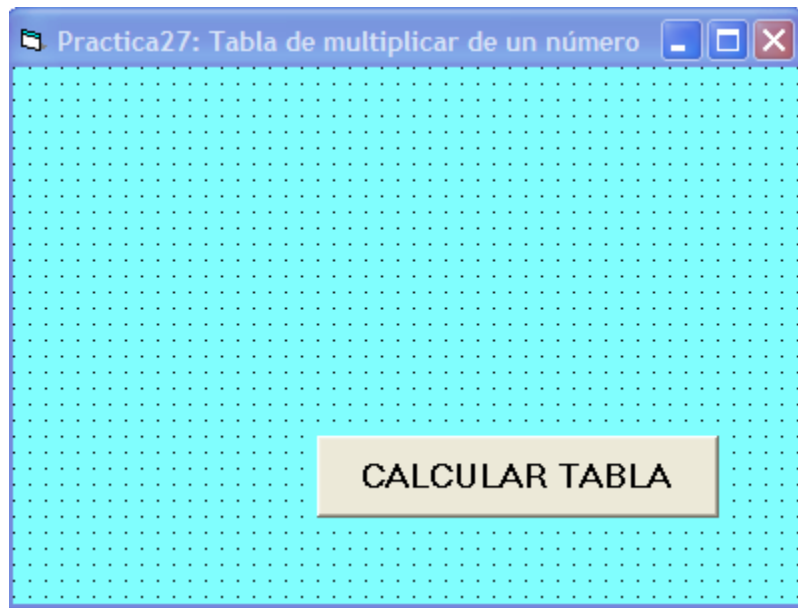
Nota: Consultar la página 26 de la Guía de Fundamentos de Programación en Visual Basic de Química.

CÓDIGO
(...)



PRACTICA 27: Tabla de multiplicar de un número

Abrimos un nuevo proyecto (Practica27.vbp) y dentro del mismo creamos un formulario con 1 botón:



INTERFAZ DE USUARIO

Formulario (nombre interno): frmPractica27.frm.

Mensaje de título: Práctica 27 : Tabla de multiplicar de un número

Botón (nombre interno): cmdCalcular

ENUNCIADO DEL EJERCICIO

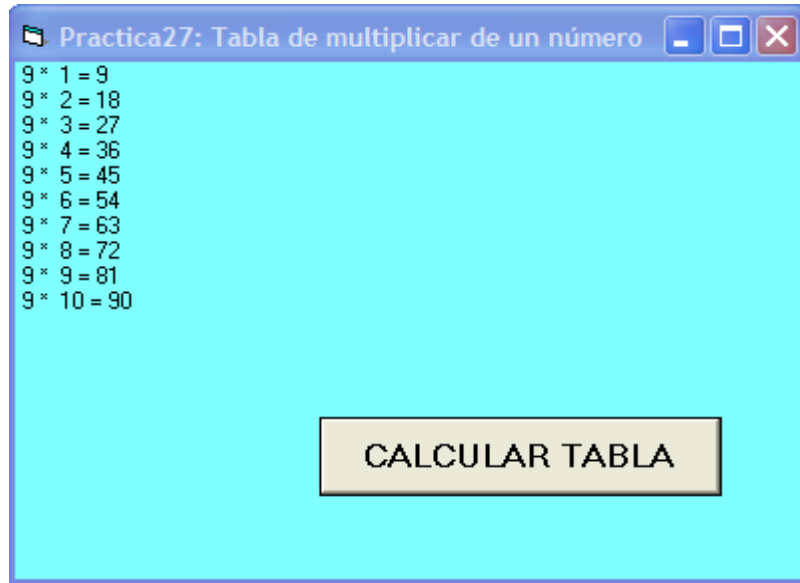
Este programa calcula la tabla de multiplicar de cualquier número, y la genera en el mismo formulario (ojo porque en este caso hay que emplear la función Print).

Así por ejemplo, al hacer Click en el botón aparece una Inputbox que recoge un valor, y si por ejemplo le introducimos el 9 en la Inputbox, aparecerá en pantalla su tabla de multiplicar correspondiente:

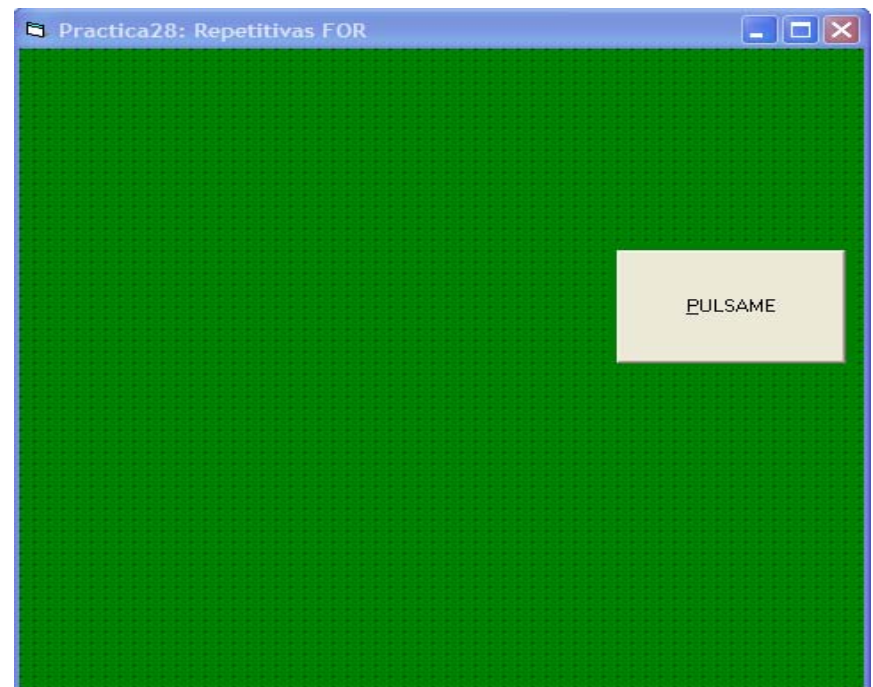


PRACTICA 28: Repetitivas FOR

Abrimos un nuevo proyecto (Practica28.vbp) y dentro del mismo creamos un formulario con 1 botón:



CÓDIGO
(...)





INTERFAZ DE USUARIO

Formulario (nombre interno): frmPractica28.frm.

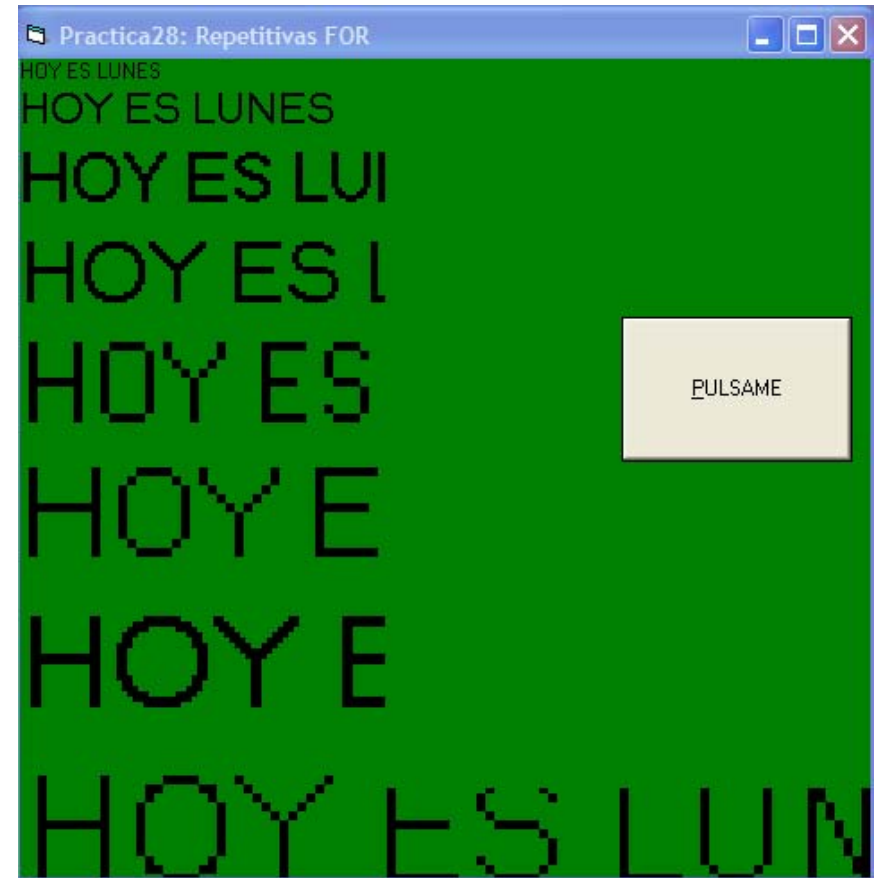
Mensaje de título: Práctica 28 : Repetitivas FOR

Botón (nombre interno): cmdPulsame

ENUNCIADO DEL EJERCICIO

Al hacer Click en el botón aparece la frase “Hoy es lunes” repetida 10 veces en el formulario, cada vez más grande. Utilizaremos para ello la función Print,

Así por ejemplo, al ejecutarse aparece el formulario de la siguiente manera:



CÓDIGO

(...)



PRACTICA 29 : N° de dígitos de un número

Diseñar e implementar un programa que dado un número, calcule cuantos dígitos tiene dicho número.

INTERFAZ DE USUARIO (...)

CÓDIGO

(...)