

#### Objetivos:

- ❖ Diseñar **diagramas de flujo** de programas con **sentencias condicionales**
- ❖ Codificar programas VB con **sentencias condicionales** ([If](#), [If-Else](#), [If-ElseIf-Else](#)).

## Programa de resolución de ecuaciones de primero y segundo grado

### Interfaz

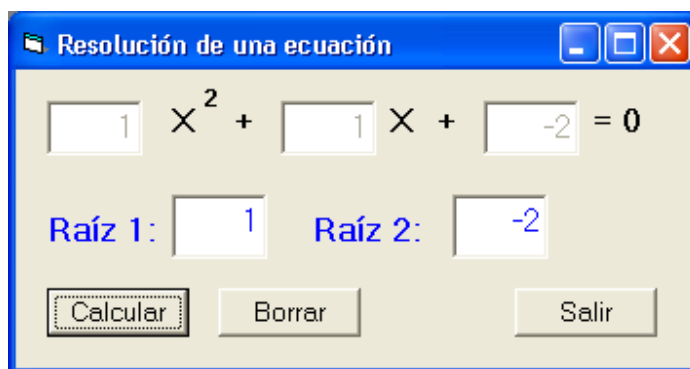


Figura 4.1 Controles de la calculadora de ecuaciones.

### Funcionamiento

1. Al lanzar el programa los controles del **resultado** (en azul) estarán **invisibles**
2. Al hacer **click** sobre el botón **Calcular**, se debe:
  - **Recoger** el valor de los coeficientes **a**, **b** y **c** de la ecuación
  - Calcular las **raíces**
  - Bloquear los campos para evitar que se modifiquen los valores y se pierda la coherencia
3. Al hacer **click** sobre el botón **Borrar** se debe **borrar** el contenido de las cajas de texto volviendo todos los objetos al **estado inicial**
4. Al hacer **click** sobre el botón **Salir**, el programa **finalizará**.

### Estudio de casos

La resolución de una ecuación de segundo grado dados sus tres coeficientes podemos distinguir cuatro casos: que sea una ecuación imposible o trivial (si  $a=0$  y  $b=0$ ), de primer grado (si  $a=0$  y  $b \neq 0$ ), de segundo grado (si  $a \neq 0$  y  $b \neq 0$ ) con soluciones reales (si el discriminante  $d = b^2 - 4ac$  es positivo o nulo) o con soluciones imaginarias (si el discriminante es negativo). Estos casos se resumen en la Tabla 4.1.

Fórmula general	1 <sup>er</sup> grado	2 <sup>o</sup> grado reales	2 <sup>o</sup> grado imaginarias
$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$	$r = \frac{-c}{b}$	$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ $r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$	$r_1 = \frac{-b}{2a} + \frac{\sqrt{4ac - b^2}}{2a}i$ $r_2 = \frac{-b}{2a} - \frac{\sqrt{4ac - b^2}}{2a}i$
<b>Caso particular:</b>	$a = 0$	$d = b^2 - 4ac \geq 0$	$d = b^2 - 4ac < 0$

Tabla 4.1 Estudio de casos.

## Diagrama de flujo

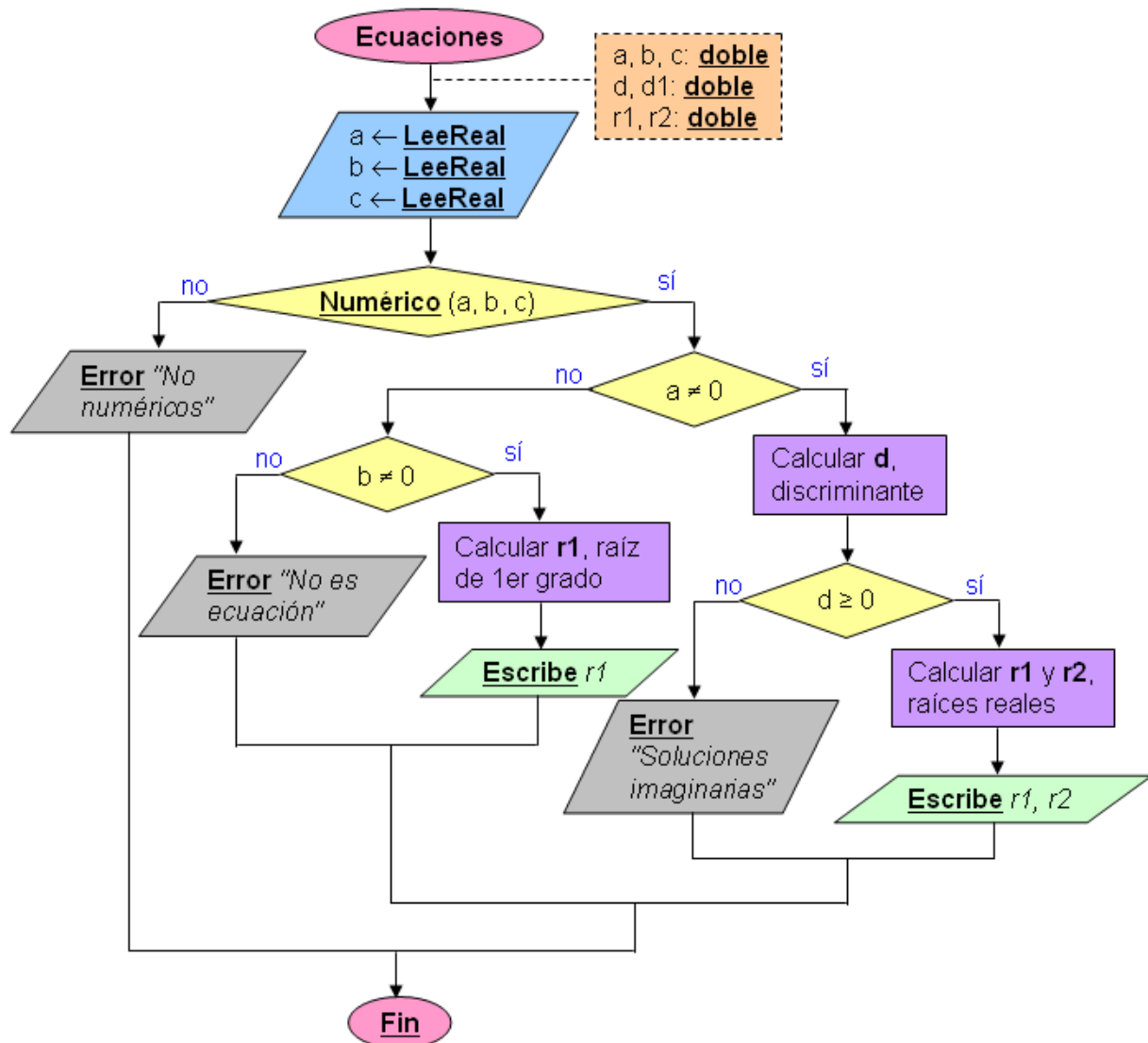


Figura 4.2 Diagrama de flujo.

## Pasos a seguir

- 1) Crearemos los controles del tipo y forma mostrados en la Figura 4.1. Sólo daremos un nombre particular a aquéllos que nos interesa leer o modificar sus propiedades en algún momento del programa. En la Figura 4.3 se muestran en rojo los nombres de estos objetos.

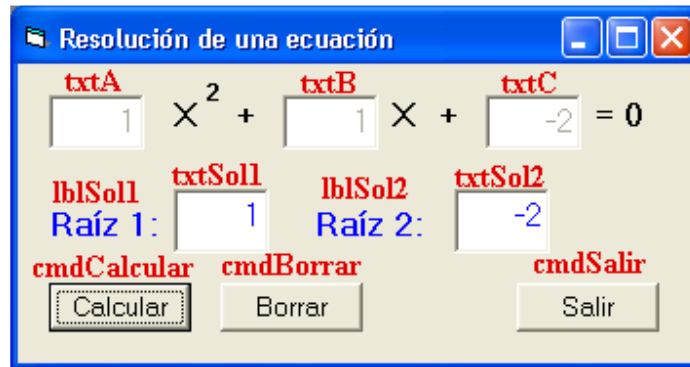


Figura 4.3 Nombres de los controles de la calculadora de ecuaciones.

- 2) Pondremos a **False** la propiedad **visible** de los objetos de la solución, a saber, `lblSol1`, `txtSol1`, `lblSol2` y `txtSol2`. De esta manera, al ejecutar el programa se visualizará una ventana como la que se muestra en la Figura 4.4.

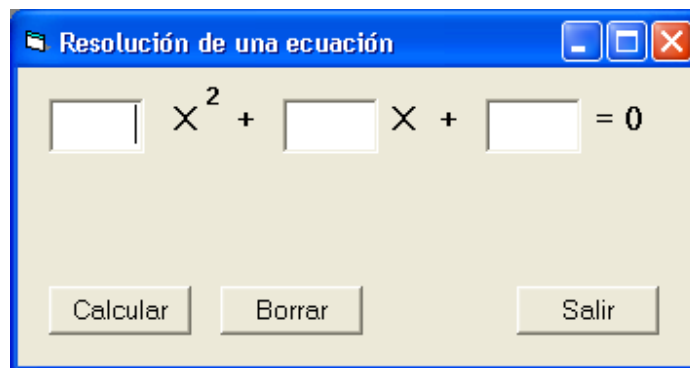


Figura 4.4 Ventana nada más ejecutar el programa.

- 3) **Código del botón Calcular:** se controlará la validez de los coeficientes (han de ser numéricos) y en función de los valores introducidos por el usuario, se determina el tipo de ecuación y se calcularán los resultados. Veamos el código paso a paso.
  - i) **Declararemos** las variables para los coeficientes **a**, **b** y **c** como **Double** (reales), así como el resto de las variables especificadas en el algoritmo. Luego **controlaremos** que sean **numéricas** mediante la función **IsNumeric** y si es así **leeremos** su valor a sus variables correspondientes mediante la instrucción **Cdbl**. El código correspondiente puede verse en la Figura 4.5.

```

Sub cmdCalcular_Click()
    Dim a As Double, b As Double, c As Double
    Dim d As Double, d1 As Double ' Discriminante y su raíz cuadrada
    Dim r1 As Double, r2 As Double
    ' Obtener los valores de los coeficientes
    If IsNumeric(txtA.Text) And _
        IsNumeric(txtB.Text) And _
        IsNumeric(txtC.Text) Then
        ' Los valores son numéricos
        a = Cdbl(txtA.Text)
        b = Cdbl(txtB.Text)
        c = Cdbl(txtC.Text)
        ' Bloquear coeficientes
        txtA.Enabled = False
        txtB.Enabled = False
        txtC.Enabled = False
    ... (1)
    Else
        MsgBox ("Error: Coeficientes no numéricos")
    End If
End Sub ' Fin de cmdCalcular_Click

```

Figura 4.5 Código del botón Calcula (simplificado).

- ii) Tras leer el valor de las variables, siguiendo el algoritmo, determinaremos si se trata de una ecuación de **segundo grado**, de **primer grado** o si **no se trata de una ecuación**. El código de las condicionales correspondientes se muestra en la Figura 4.6. Habrá que tener cuidado con la **indentación** (sangrado) ya que afecta de manera importante a la legibilidad.

```

txtC.Enabled = False (1)
If a <> 0 Then ' Ecuación de segundo grado
... (2)
Else
    If b <> 0 Then ' Ecuación de primer grado
... (3)
    Else ' No es ecuación
        MsgBox ("Error: No es una ecuación")
    End If
End If
Else

```

Figura 4.6 Código de condicionales para diferentes casos.

- iii) Para las ecuaciones de segundo grado (coeficiente **a**  $\neq 0$ ) calcularemos el **discriminante** (variable **d**). Si éste es **nulo** tendremos **dos raíces reales idénticas**, si es **positivo** tendremos **dos raíces reales** y si es **negativo** tendremos **dos raíces imaginarias**. Este último caso lo indicaremos mediante un mensaje. Véase el código de la Figura 4.7.

```

If a <> 0 Then ' Ecuación de segundo grado (2)
    d = b ^ 2 - 4 * a * c ' Cálculo del discriminante
    If d >= 0 Then ' Soluciones reales
... (4)
    Else ' Soluciones imaginarias
        MsgBox ("Sólo hay soluciones imaginarias")
    End If
Else

```

Figura 4.7 Código de cálculo del discriminante y sus casos.

- iv) Para el cálculo de las raíces reales necesitamos obtener la raíz cuadrada mediante la función `Sqr` de Visual Basic. Así, obtendremos las raíces y las mostraremos de manera adecuada, poniendo los textos de las etiquetas y haciendo visibles los controles correspondientes, como se muestra en la Figura 4.8.

```

If d >= 0 Then ' Soluciones reales (4)
    d1 = Sqr(d) ' Raíz cuadrada
    r1 = (-b + d1) / (2 * a)
    r2 = (-b - d1) / (2 * a)
    txtSol1.Text = CStr(r1)
    txtSol2.Text = CStr(r2)
    lblSol1.Caption = "Raíz 1:"
    lblSol1.Visible = True
    txtSol1.Visible = True
    lblSol2.Caption = "Raíz 2:"
    lblSol2.Visible = True
    txtSol2.Visible = True
Else ' Soluciones imaginarias

```

**Figura 4.8** Código para visualizar las soluciones dobles reales.

- v) Finalmente, cuando el coeficiente **a** es nulo nos encontramos ante una ecuación lineal, que sólo tendrá una raíz. Con esto finalizaremos el código asociado al botón **Calcular** y añadiremos un subprograma para visualizar una única raíz. Este código, fácilmente deducible, se muestra en la Figura 4.9.

```

If b <> 0 Then ' Ecuación de primer grado (3)
    r1 = -c / b
    txtSol1.Text = CStr(r1)
    lblSol1.Caption = "Raíz:"
    lblSol1.Visible = True
    txtSol1.Visible = True
Else

```

**Figura 4.9** Código para visualizar las ecuaciones lineales.

- 4) **Código del botón Borrar:** tendremos que volver a hacer invisibles los objetos de la solución, `lblSol1`, `txtSol1`, `lblSol2` y `txtSol2`, tal y como habíamos hecho originalmente en el entorno de desarrollo, poniendo a `False` la propiedad **Visible**. Asimismo desbloquearemos los cuadros de texto de los coeficientes **a**, **b** y **c**, vaciando su contenido. El código de estas acciones se ofrece en la Figura 4.10.

```

Sub cmdBorrar_Click()
    ' Hacer invisibles los objetos de la solución
    lblSol1.Visible = False
    lblSol2.Visible = False
    txtSol1.Visible = False
    txtSol2.Visible = False
    ' Desbloquear A-B-C
    txtA.Enabled = True
    txtB.Enabled = True
    txtC.Enabled = True
    txtA.Text = ""
    txtB.Text = ""
    txtC.Text = ""
End Sub

```

**Figura 4.10** Código del botón Borrar.

- 5) **Código del botón Salir.** Igual que en ejemplos anteriores

## Ejercicio 4.1: Programa rodillo

Diseña la interfaz y codifica el rodillo de la Figura 4.11 **sin hacer referencia al texto en el programa**. Si se cambia el texto en la interfaz el programa debe seguir funcionando de la misma manera con el texto nuevo.

### Interfaz

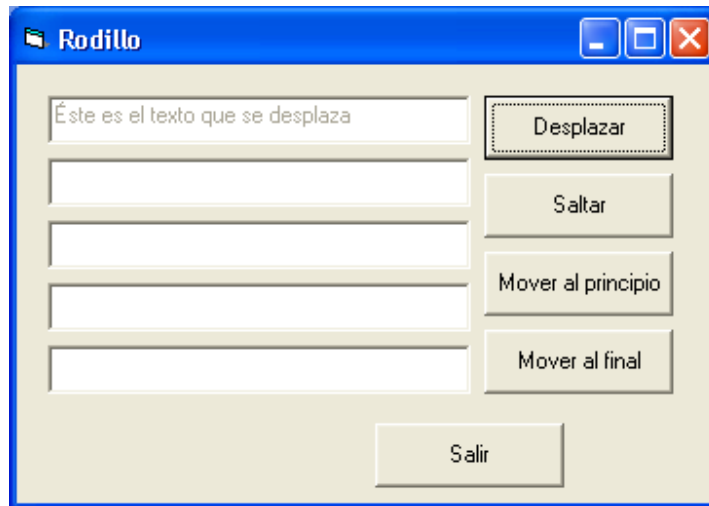


Figura 4.11 Objetos presentes en la interfaz del rodillo.

### Funcionamiento

1. El programa contiene 5 cuadros de texto **no editables** (propiedad **Enabled** a **False**). El primero de ellos contiene un texto, por ejemplo “*Éste es el texto que se desplaza*”. Los botones de la ventana permiten desplazar el texto.
2. **Botón Desplazar**: El texto se desplaza a la **siguiente posición** de modo circular. Si el texto se encuentra en la última posición, éste pasa a la primera.
3. **Botón Saltar**: El texto se desplaza de manera circular **dejando un hueco** entre la posición que ocupaba y la nueva posición. Cuando llega al final vuelve a desplazar por arriba.
4. **Botón Mover al principio**: Coloca el texto en la **primera casilla**.
5. **Botón Mover al final**: Coloca el texto en la **última casilla**.

### Recomendación

Para saber dónde se encuentra el texto compáralo con la cadena vacía.

Utiliza las sentencias condicionales para determinar en qué casilla se encuentra el texto y así decidir la nueva ubicación de éste, según el botón. Copia el texto de la casilla origen a la casilla destino y luego pon la casilla origen a cadena vacía.

Para el programa en papel es suficiente con uno de los botones y una explicación de cómo se generaliza para el resto de los botones.

## Ejercicio 4.2: Calculo de billetes de euro (2)

**Diseña la interfaz y codifica** la calculadora de billetes de euro de la Figura 4.12. La base es la misma que el ejercicio 2.4 pero sólo muestra información relevante (no muestra valores nulos) y distingue el caso de un billete del de varios. Los billetes válidos serán de 500, 200, 100, 50, 20, 10, 5 y euros sueltos.

### Interfaz



**Figura 4.12.** Interfaz del programa de cálculo billetes.

## Ejercicio 4.3: Programa de cálculo de la fecha siguiente

Diseña el **diagrama de flujo**, la **interfaz** y **codifica** la calculadora de fechas de la Figura 4.13.

### Interfaz

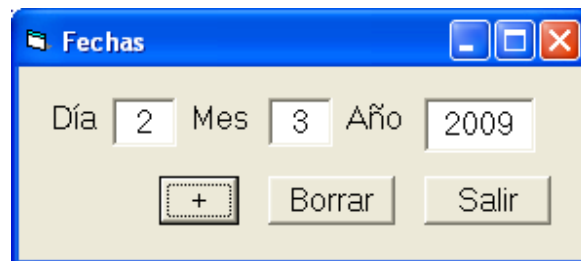


Figura 4.13 Interfaz del programa de cálculo de fechas.

### Funcionamiento

El programa trabaja con fechas (probar ejecutable).

1. **Botón +**: Controla que la fecha introducida es correcta (años positivos de cuatro dígitos) y calcula la fecha del día siguiente, escribiéndola en los cuadros de texto correspondientes.
2. **Botón Borrar**: Elimina el contenido de las cajas de texto.
3. **Botón Salir**: Sale del programa.

### Comprobaciones

Para el diagrama de flujo utiliza los siguientes “predicados”.

<b>c1</b>	Contenido de los campos <b>día</b> , <b>mes</b> y <b>año</b> es <b>numérico</b>
<b>c2</b>	<b>Día OK</b> : mayor o igual que 1 y menor o igual que 31
<b>c3</b>	<b>Mes OK</b> : mayor o igual que 1 y menor o igual que 12
<b>c4</b>	<b>Año OK</b> : mayor o igual que 1000 y menor o igual que 9999
<b>c5</b>	<b>Fecha OK</b> : no hay <b>30 ni 31 de febrero</b>
	y no hay <b>día 31</b> en febrero (ya verificado), abril, junio, septiembre o noviembre
	y no hay <b>29 de febrero</b> los años que no son <b>bisiestos</b> <sup>1</sup>

### Recomendación

Nótese que al incrementar el día podemos llegar a una fecha incorrecta, en cuyo caso habrá que pasar al primer día del mes siguiente y posiblemente al primer día del año siguiente. Esta verificación será similar a la del control de corrección de la fecha.

<sup>1</sup> Son bisiestos los años **divisibles por 4 que no lo son por 100 salvo los que son divisibles por 400**. Por ejemplo, 1996, 2000 y 2004 fueron bisiestos, pero no lo serán el 2100, 2200 y 2300, y sí 2400. A ver quiénes lo contamos.

## Tablas de referencia rápida

Sintaxis de las estructuras condicionales		
Sintaxis	Ejemplo	Diagrama de flujo
<u>If</u> <i>c</i> <u>Then</u> ... <u>End If</u>	<u>If</u> <i>a</i> < 0 <u>Then</u> <i>a</i> = 0 <u>End If</u>	
<u>If</u> <i>c</i> <u>Then</u> ... <u>Else</u> ... <u>End If</u>	<u>If</u> <i>a</i> < 0 <u>Then</u> <i>b</i> = - <i>a</i> <u>Else</u> <i>b</i> = <i>a</i> <u>End If</u>	
<u>If</u> <i>c1</i> <u>Then</u> <i>a1</i> <u>Else</u> <u>If</u> <i>c2</i> <u>Then</u> <i>a2</i> <u>Else</u> <i>a3</i> <u>End If</u> <u>End If</u>	<u>If</u> <i>a</i> > <i>b</i> <u>Then</u> <i>m</i> = "a" <u>Else</u> <u>If</u> <i>b</i> > <i>a</i> <u>Then</u> <i>m</i> = "b" <u>Else</u> <i>m</i> = "=" <u>End If</u> <u>End If</u>	
<u>If</u> <i>c1</i> <u>Then</u> <i>a1</i> <u>ElseIf</u> <i>c2</i> <u>Then</u> <i>a2</i> <u>Else</u> <i>a3</i> <u>End If</u> <u>End If</u>	<u>If</u> <i>a</i> > <i>b</i> <u>Then</u> <i>m</i> = "a" <u>ElseIf</u> <i>b</i> > <i>a</i> <u>Then</u> <i>m</i> = "b" <u>Else</u> <i>m</i> = "=" <u>End If</u> <u>End If</u>	

Tabla 4.2 Sintaxis de las estructuras condicionales.

Funciones Visual Basic	
<b>IsNumeric</b>	Indica si una cadena representa un valor numérico
<b>Sqr</b>	Calcula la raíz cuadrada de un número

Tabla 4.3 Funciones Visual Basic.