Capítulo 4
Estructuras de control

Duración: 1 hora 55

Palabras clave

condición, elección, prueba, alternativa, salto condicional, contador, iteración, incrementar, disminuir, salida

Objetivos

Aprender las estructuras de decisión para probar condiciones y realizar acciones diferentes según el resultado obtenido. Aprender las instrucciones de iteración que, asociadas a las instrucciones condicionales, permiten escribir código Visual Basic para la toma de decisiones y la repetición de acciones. Volverá a encontrar estas estructuras en el resto del libro. En este capítulo, nos limitaremos al uso de los cuadros de diálogo que ya hemos visto.

Requisitos

Para validar los requisitos necesarios, antes de abordar la práctica, responda a las siguientes preguntas (algunas preguntas pueden tener varias respuestas):

1. Las siguientes estructuras son estructuras de decisión:

- **a.** `If ... Then ... Else ... End If`
- **b.** `Do ... Loop`
- **c.** `Select Case ... Case ... End Select`

2. `Resultado = IIf(7 / 2 > 3, IIf(2.8 * 3.3 < 11, "X", "Y"), "Z")`
La variable `Resultado` contiene el valor:

- **a.** X
- **b.** Y
- **c.** Z

3. Las siguientes instrucciones correspondientes cada una a una estructura de control distinta son correctas:

a. `Case If N1 > N2`

b. `Case A, B, C`

c. `Case 1 to 10`

d. `Case Número, Is > 50`

4. La palabra clave `ElseIf` :

a. puede aparecer a continuación de una cláusula `Else`.

b. es opcional.

c. se puede utilizar varias veces en un bloque `If`.

5. Repetición de instrucciones mientras la condición tenga el valor `True` :

a. `For Each . . . Next`

b. `For ... Next`

c. `Do ... Loop`

d. `While ... Wend`

e. `With ... End With`

6. Utilización de un contador para ejecutar instrucciones un cierto número de veces:

a. `For Each . . . Next`

b. `For ... Next`

c. `Do ... Loop`

d. `While ... Wend`

e. `With ... End With`

7. Repetir un grupo de instrucciones para cada elemento de una tabla o de una colección:

a. `For Each . . . Next`

b. `For ... Next`

c. `Do ... Loop`

d. `While ... Wend`

e. `With ... End With`

8. Repetir un grupo de instrucciones un número de veces determinado:

a. `For Each . . . Next`

b. `For ... Next`

c. `Do ... Loop`

d. `While ... Wend`

e. `With ... End With`

9. Ejecutar una serie de instrucciones aplicadas a un solo objeto o a un tipo definido por el usuario:

a. `For Each . . . Next`

b. `For ... Next`

c. `Do ... Loop`

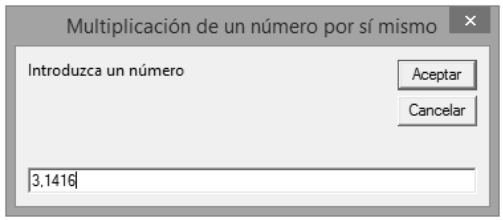
d. `While ... Wend`

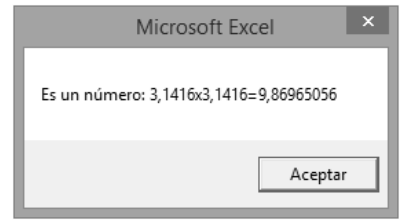
e. `With ... End With`

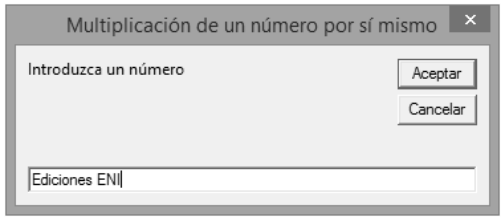
Solución pág. 249

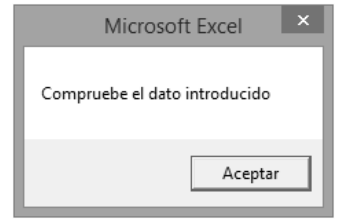
Enunciado 4.1 Comprobar que se introduce un número

Ejercicio 1
Duración estimada: 10 minutos
Cree el procedimiento **ControlEntrada** que muestre una caja de texto y compruebe que se trata de un número. Si lo es, multiplique el número por sí mismo; si no, muestre un mensaje.
Ejemplo:





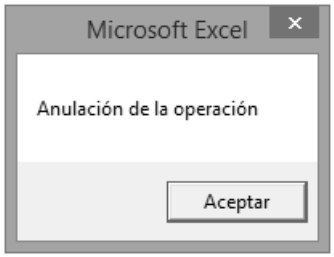




Pista

Utilice la función `IsNumeric` *para comprobar el dato introducido.*

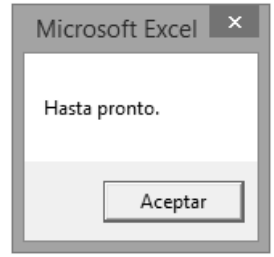
Ejercicio 2
Duración estimada: 5 minutos
Complete el procedimiento anterior que compruebe que el usuario ha hecho clic en el botón **Aceptar**. En caso contrario, muestre un mensaje. Ejemplo:



Solución pág. 249

Enunciado 4.2 Despedida

Duración estimada: 5 minutos
Complete el procedimiento **Despedida** que ofrezca un cuadro de diálogo y pregunte si se desea salir de Excel. Debe mostrar el mensaje "Hasta pronto" si el usuario hace clic en el botón **Sí**, y si no, el mensaje "Continuemos". Ejemplo:



```
Sub Despedida()
  Dim strMensaje As String
  Dim intEstilo As Integer
  Dim intEleccion As Integer
  strMensaje = "¿Desea salir de Excel?"
  intEstilo = vbYesNo + vbDefaultButton2
  ...  = MsgBox(strMensaje, intEstilo)
  If ... = vbYes Then
    MsgBox "Hasta pronto."
  ...
    MsgBox "Continuemos."
  ...
End Sub
```

Pista

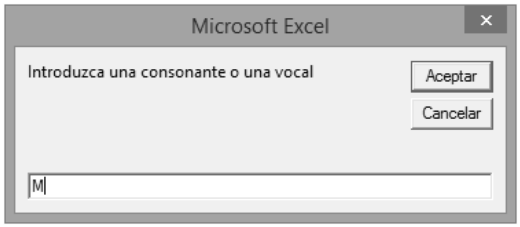
Aquí tiene un extracto de la ayuda en línea de VBA:

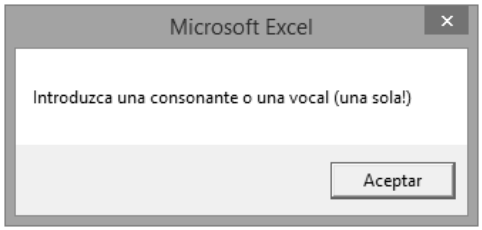
Miembro	Valor	Descripción
OKOnly	0	Muestra sólo el botón Aceptar.
OKCancel	1	Muestra los botones Aceptar y Cancelar.
AbortRetryIgnore	2	Muestra los botones Anular, Reintentar y Omitir.
YesNoCancel	3	Muestra los botones Sí, No y Cancelar.
YesNo	4	Muestra los botones Sí y No.

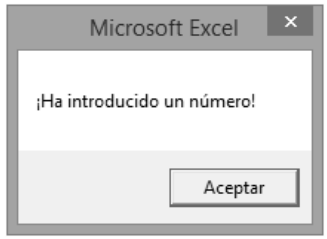
Solución pág. 250

Enunciado 4.3 Controlar la introducción de una consonante o una vocal

Duración estimada: 10 minutos
Cree el procedimiento **ConsonanteVocal** que pida introducir una vocal o una consonante. Muestre un mensaje según la letra introducida: "Vocal", "Consonante" o un mensaje apropiado en caso de error. Ejemplo:







Pista

Para comprobar si la letra es mayúscula se hará la siguiente transformación.

```
varEleccion = VBA.UCase(varEleccion)
```

Solución pág. 250

Enunciado 4.4 Mostrar un mensaje según la edad y el sexo

Duración estimada: 15 minutos
Cree el procedimiento **SuEdad** que pida el sexo y la edad del usuario. Los mensajes serán diferentes según se trate de un hombre o de una mujer. Tome igualmente en consideración la cancelación de la operación y la introducción de letras para la edad.
0 - 17 años: Niño
18 - 30 años: Es joven
31 - 50 años: Todavía es joven
> 50 años: Comienza a envejecer