



**Ingeniería
Industrial**

PONTIFICIA UNIVERSIDAD
CATOLICA DE VALPARAISO

Introducción a la programación en VBA para Excel

VBA

02

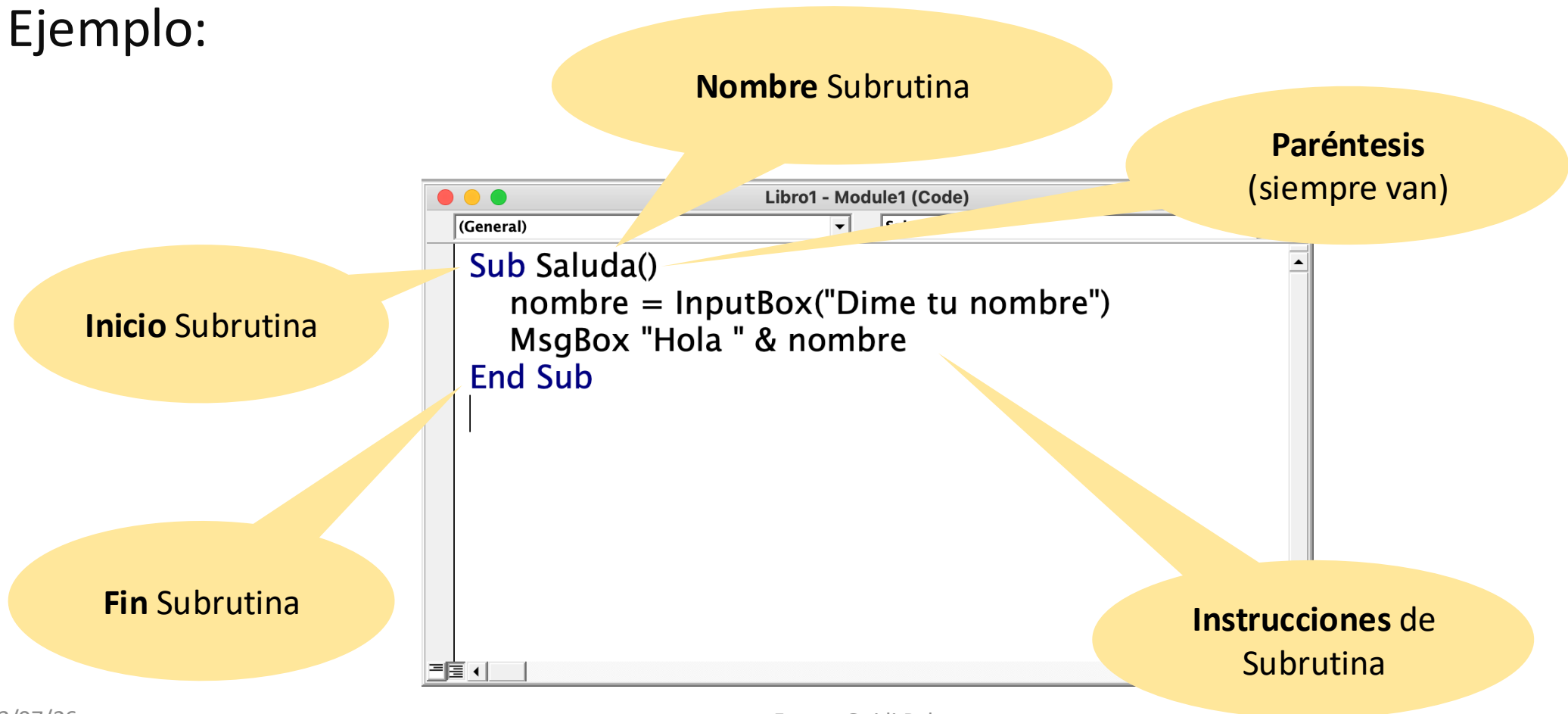
Programación de subrutinas

Prof. Franco Guidi

franco.guidi@pucv.cl

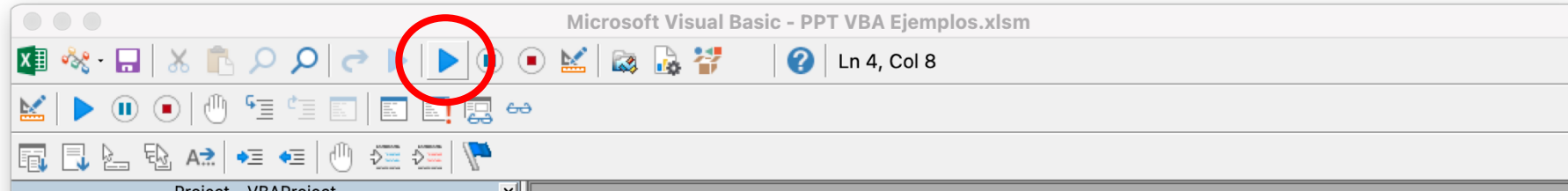
Identificación de subrutinas

- Comienzan con la palabra **Sub** y terminan con **End Sub** .
- Se identifican por un nombre (seguido de paréntesis)
- Ejemplo:

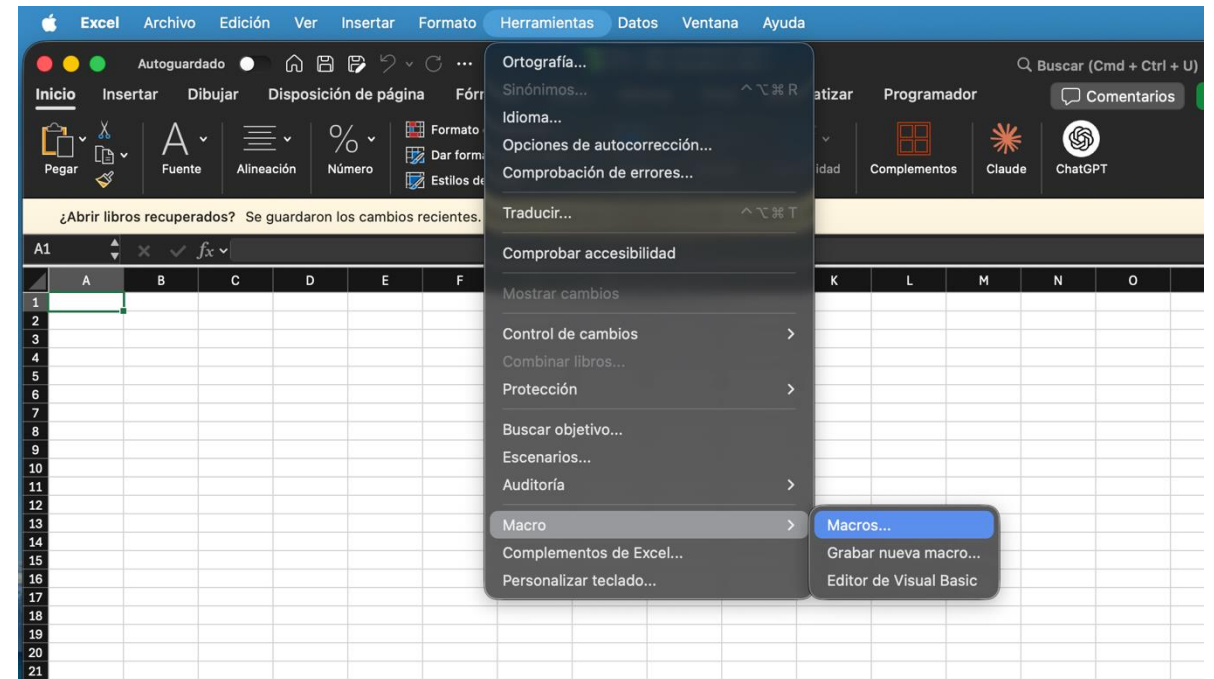


Ejecución de Subrutinas

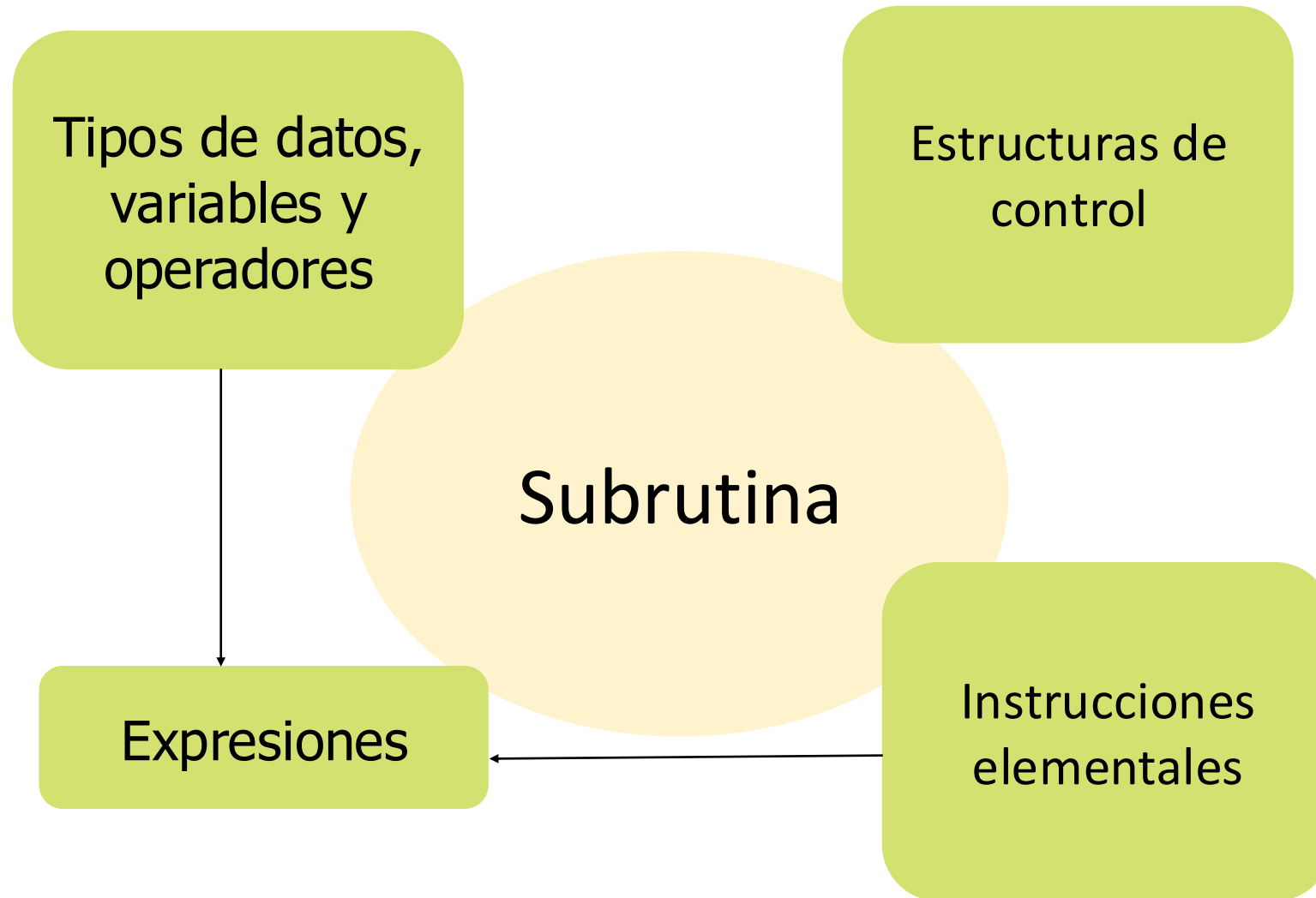
- Botón ejecutar en editor de VBA



- En hoja de Excel: Herramientas – Macro - Ejecutar



Elementos básicos de cualquier programa (incluyendo *Sub procedures*)



Valores y tipos de datos

- Los tipos de datos determinan los conjuntos de valores posibles que un lenguaje de programación maneja.
- Los tipos de datos definidos en un lenguaje de programación determinan:
 - Variables
 - Argumentos y resultados de expresiones.
 - Argumentos y resultados de funciones.
- Cada lenguaje especifica sus propios tipos de datos.

Principales tipos de datos de VBA para Excel

Tipo	Significado	Rango de valores
Boolean	Lógico	True o False
Integer	Entero	-32.768 a 32.767
Long	Entero largo	-2.147.483.648 a 2.147.483.647
Single	Punto flotante de precisión simple	-3,402823e38 a -1,401298e-45 (negativos)
		1,401298e-45 a 3,402823e38 (positivos)
Double	Punto flotante de precisión doble	-1,79769313486231e308 a -4,94065645841247e-324 (negativos)
		4,94065645841247e-324 a 1,79769313486232e308 (positivos)
String	Cadena de caracteres	Longitud variable

Variables en VBA para Excel

- Las **variables** permiten el almacenamiento de valores (uno a la vez).
- Cada variable se identifica por su nombre (recomendable que sea significativo).
- El nombre de una variable puede contener casi cualquier letra o símbolo, pero debe comenzar siempre por una letra.
- No se deben usar espacios ni comas como parte de un nombre de variable, como tampoco los símbolos: #, \$, %, & o !, ni operadores aritméticos.
- Por ejemplo:

edad, cantidad_asistentes, NivelLiquido, GASTOS

- No hay distinción entre mayúsculas y minúsculas, por lo tanto los identificadores: **balanceFinal**, **BalanceFinal** y **BALANCEFINAL** hacen referencia a la misma variable.

Declaración de una variable

- Las variables se declaran mediante la instrucción:

```
Dim variable As tipo_dato
```

- Ejemplos:

- **Dim** nombre **As** String
- **Dim** bandera **As** Boolean

- Es posible declarar más de una variable en una instrucción Dim:

- **Dim** peso **As** Integer, edad **As** Integer, i **As** Integer

Variables: ¡Advertencia!

- VBA, por omisión, **crea automáticamente** variables cuando encuentra en el código un identificador válido, al cual se le asigna un valor.
- Esta práctica, sin embargo, **puede generar errores** y está **completamente desaconsejada**.
- Por lo tanto, para efectos de esta asignatura:

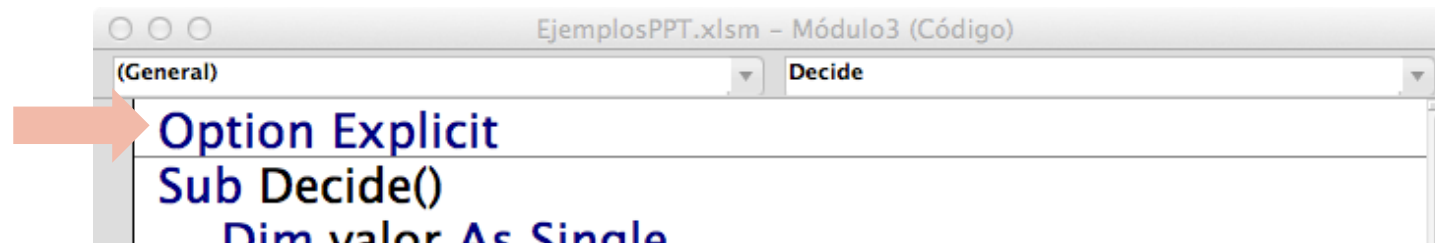
Toda variable debe ser siempre DECLARADA.

Variables

- Para verificar que todas las variables estén declaradas, se escribirá la instrucción

Option Explicit

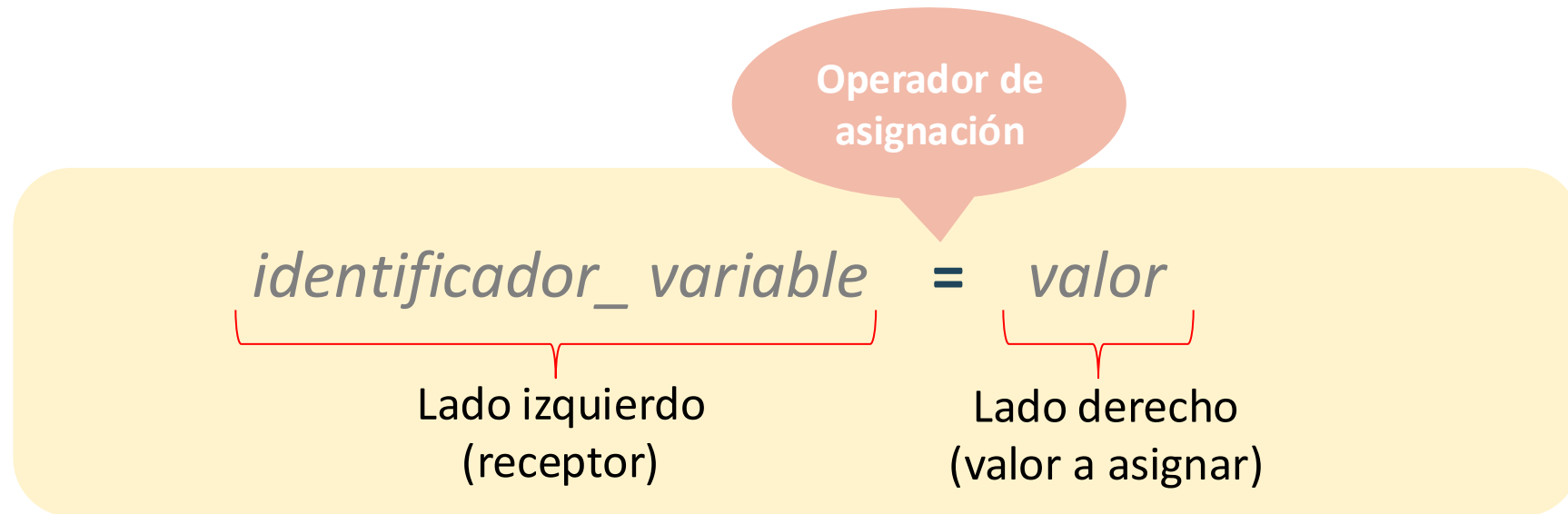
al comienzo de cada módulo (antes de la declaración de la Subrutina).



- **Option Explicit** obliga a declarar todas las variables. Si el código utiliza una variable no declarada, VBA detectará el error antes de ejecutar correctamente el procedimiento.

Variables y el operador de asignación

- Para asignar un valor a una variable se debe utilizar el operador de asignación (=).



Operadores aritméticos

- Los argumentos de los operadores aritméticos son valores numéricos (números o variables numéricas) y generan como resultado un número.
- Operadores aritméticos:

■ Suma:	+
■ Resta:	-
■ Multiplicación:	*
■ División:	/
■ División entera:	\
■ Elevación a potencia:	^

Operadores relacionales

- Los operadores relacionales permiten comparar valores en expresiones. Generan como resultado un valor lógico (verdadero o falso).
- Operadores relacionales:

■ Mayor:	>
■ Menor:	<
■ Mayor o igual:	>=
■ Menor o igual:	<=
■ Igual:	=
■ Distinto:	<>

Operadores lógicos

- Los argumentos de operadores lógicos son valores lógicos y generan como resultado otro valor lógico.
- Operadores lógicos (existen otros):

- Conjunción: **And**
- Disyunción : **Or**
- Negación : **Not**

Operador de concatenación

- Se utiliza para concatenar dos cadenas de caracteres:

- **&** (ampersand)

- Ejemplos:

“Hola” **&** “Mundo”  “HolaMundo”

nombre = “Hola”
nombre **&** “Mundo”  “HolaMundo”

Expresiones

- **Aritméticas:** generan como resultado un número. Ejemplo:

$$(3 * 4) / 5$$

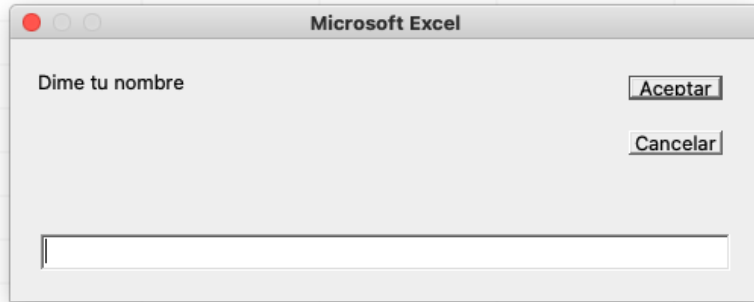
- **Lógicas:** generan como resultado un valor lógico (se utilizan generalmente como condiciones). Ejemplo:

$$\text{Edad} > 18$$

$$(\text{Edad} > 18) \text{ And } (\text{Edad} < 25)$$

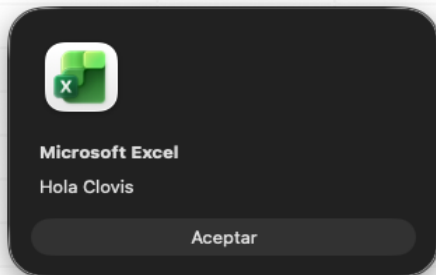
$$(\text{Valor} \neq 8) \text{ Or } (\text{Numero} = 3) \text{ Or } (\text{Numero} < 0)$$

Instrucciones elementales



- Para ingreso de datos mediante un cuadro de diálogo:

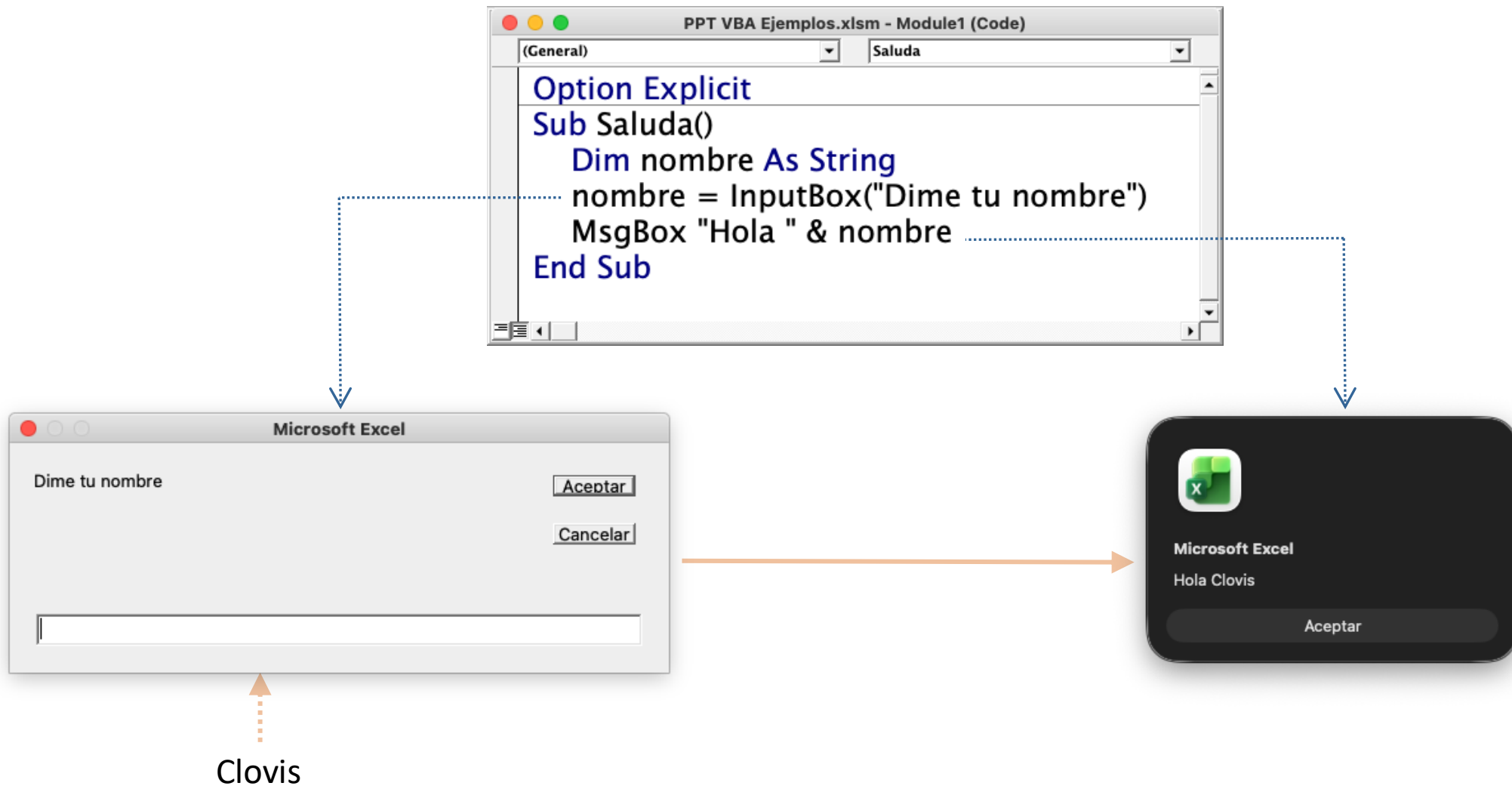
InputBox(*texto a desplegar*)



- Para despliegue de resultados mediante cuadro de diálogo:

MsgBox *texto a desplegar*

Instrucciones elementales



Buena práctica para el uso de InputBox (1/2)

- **InputBox** solo lee **Strings** (cadenas de caracteres). Al proporcionarse un valor numérico este es en realidad leído como una cadena de caracteres. Al asignarlo a una variable numérica (por ejemplo, Integer o Double) el valor es convertido de String a número. Ejemplo:

```
Dim edad As Integer  
edad = InputBox( "Ingresa tu edad" )
```

- La práctica recomendada es que el programador explicita la conversión de String a número haciendo uso de funciones de conversión. Algunas son:

- **CInt**(*cadena que representa un entero*)
- **CLng**(*cadena que representa un entero*)
- **CDbl**(*cadena que representa un decimal*)

Buena práctica para el uso de InputBox (2/2)

- Ejemplo:

```
Dim edad As Integer
```

```
Dim distancia As Long
```

```
Dim factor As Double
```

```
edad = CInt( InputBox( “Ingresa tu edad” ) )
```

```
distancia = CLong( InputBox( “Ingresa la distancia” ) )
```

```
factor = CDbl( InputBox( “Ingresa el factor” ) )
```

- Observar que en estos casos cada función de conversión recibe como argumento el valor retornado por InputBox.

Instrucciones elementales

- Para redondear valores:

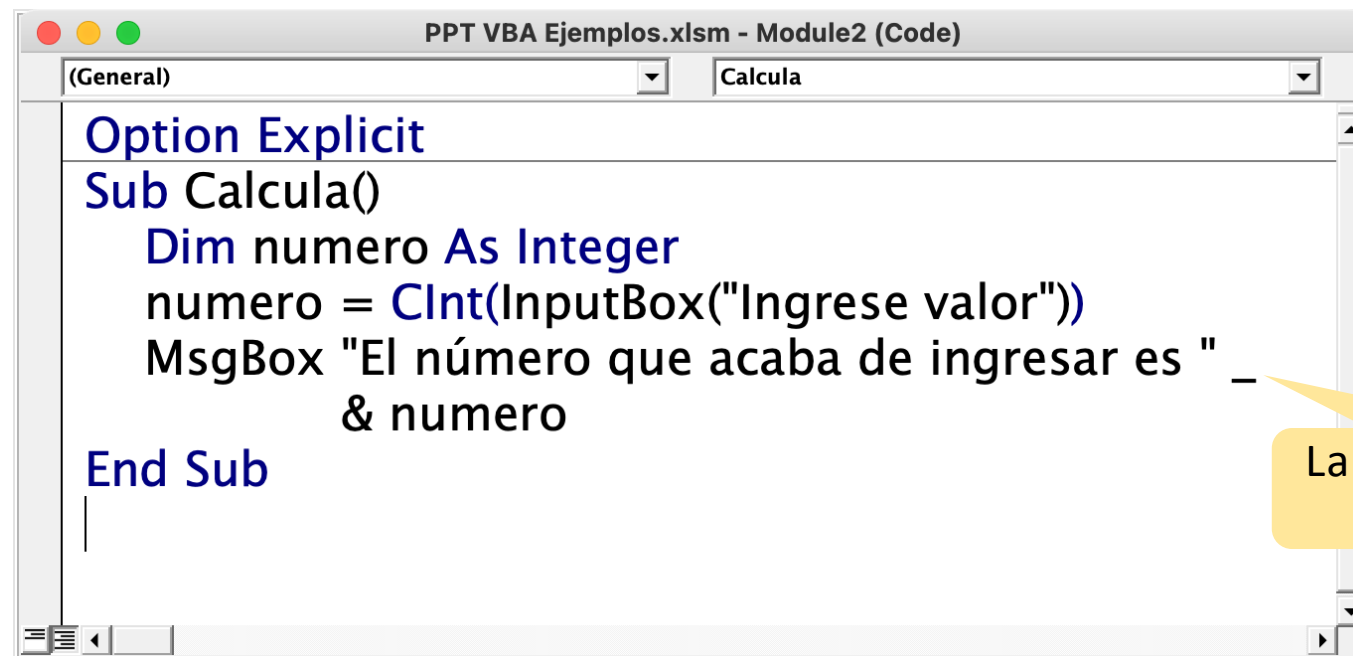
Round(*valor*, *decimales*)

- Ejemplo:

- Round(1234.567 , 1) → 1234.6

Estructuras de control: Secuencia

- Las instrucciones se escriben línea por línea.
- Si se desea escribir una instrucción en más de una línea, se debe utilizar el guión bajo (_) que indica la continuación en la línea siguiente.



The screenshot shows the VBA Code Editor window for 'PPT VBA Ejemplos.xlsm - Module2 (Code)'. The 'General' tab is selected, and the 'Calcula' procedure is visible. The code is as follows:

```
Option Explicit
Sub Calcula()
    Dim numero As Integer
    numero = CInt(InputBox("Ingrese valor"))
    MsgBox "El número que acaba de ingresar es " _
        & numero
End Sub
```

A yellow callout bubble points to the underscore character at the end of the line 'MsgBox "El número que acaba de ingresar es " _', indicating that the instruction continues on the next line.

La instrucción continúa en la línea siguiente

Estructuras de control: Decisión

- Determina el curso de acción a seguir dependiendo del valor de verdad de una condición.

If *condición* **Then**

acciones si condición es verdadera

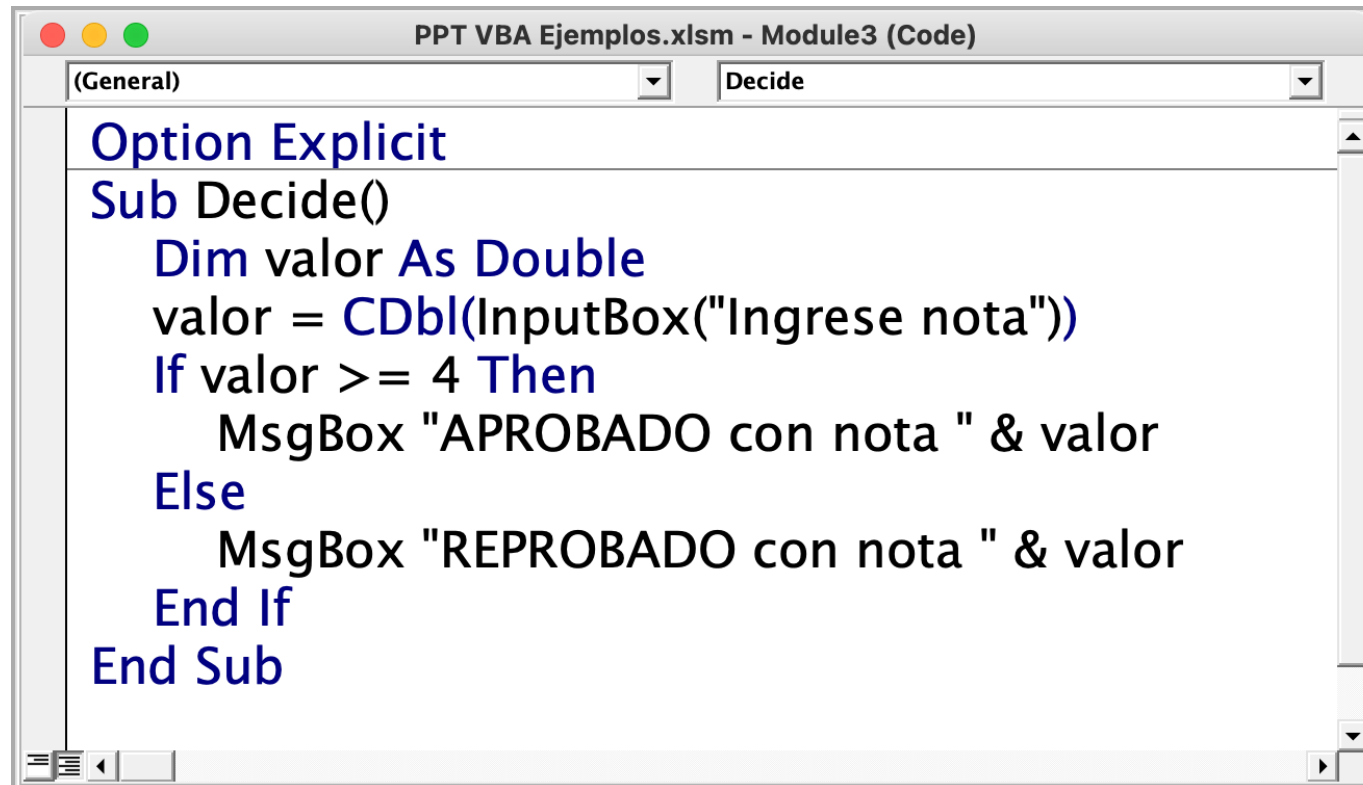
Else

acciones si condición es falsa

End If

Estructuras de control: Decisión

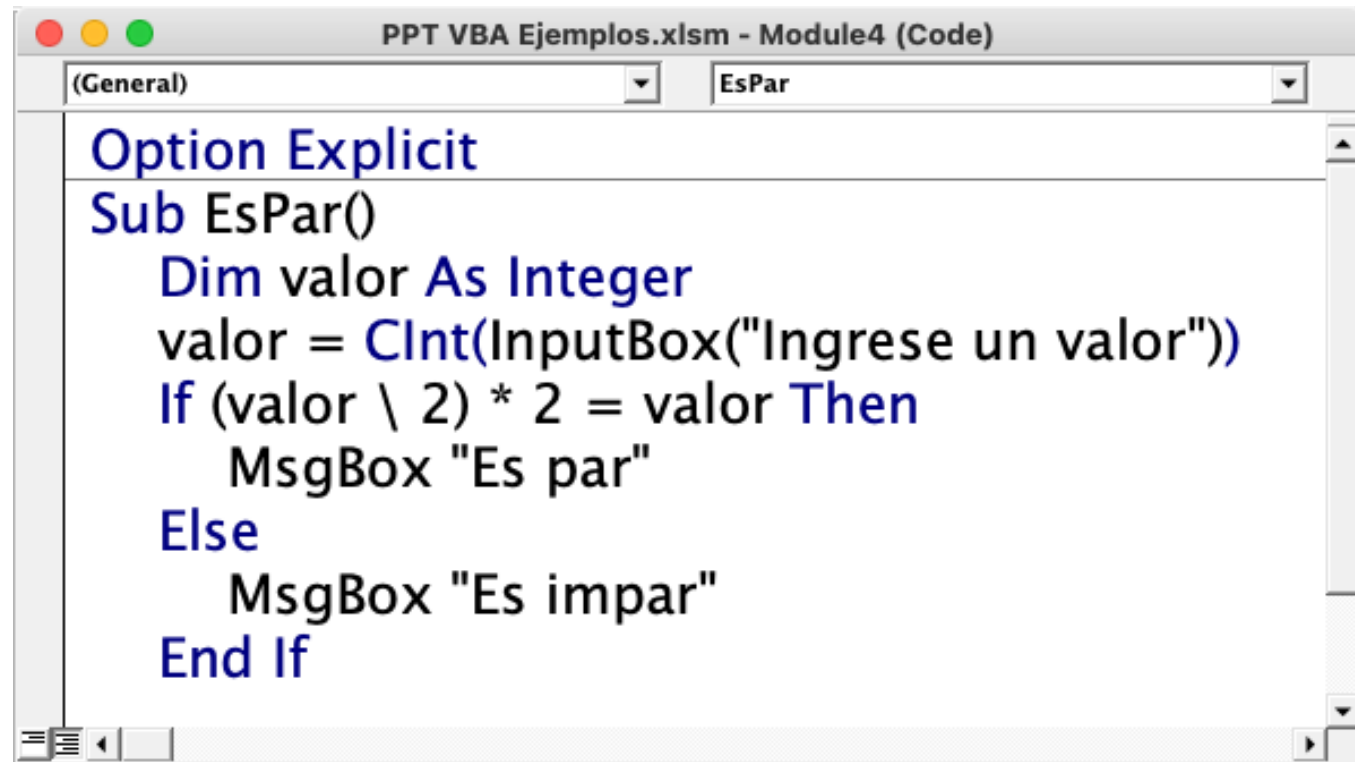
- Ejemplo: programa que pide al usuario una nota. Si la nota es igual o superior a cuatro, escribe APROBADO, en caso contrario escribe REPROBADO.



```
Option Explicit
Sub Decide()
    Dim valor As Double
    valor = CDbI(InputBox("Ingrese nota"))
    If valor >= 4 Then
        MsgBox "APROBADO con nota " & valor
    Else
        MsgBox "REPROBADO con nota " & valor
    End If
End Sub
```

Ejemplo

- Determinar si un número ingresado es o no par:



```
Option Explicit
Sub EsPar()
    Dim valor As Integer
    valor = CInt(InputBox("Ingrese un valor"))
    If (valor \ 2) * 2 = valor Then
        MsgBox "Es par"
    Else
        MsgBox "Es impar"
    End If
End Sub
```

Estructuras de control: Iteración

- Realiza iteraciones mientras una condición es verdadera.

Do While *condición*

instrucción 1

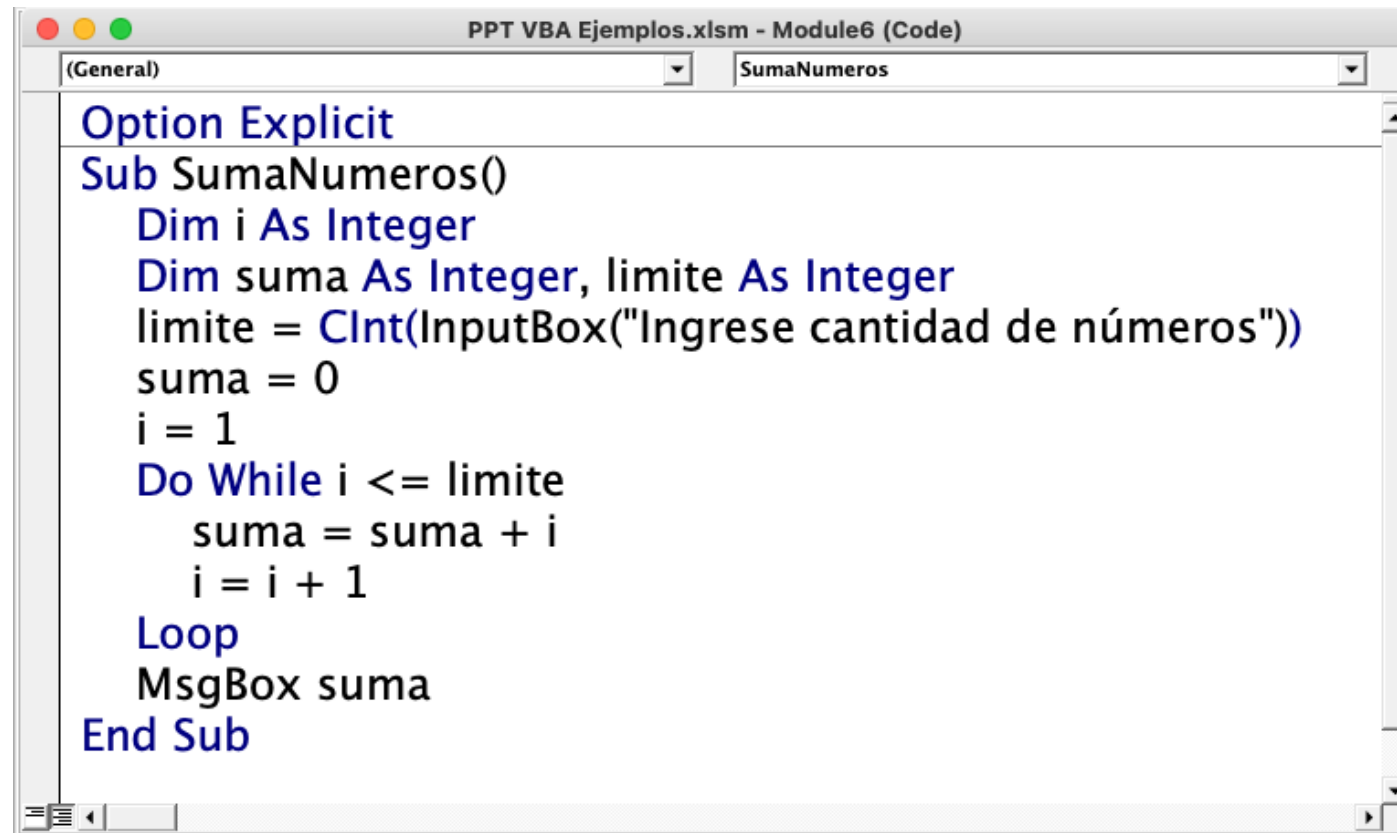
instrucción 2

...

Loop

Estructuras de control: Iteración

- Construir un procedimiento que calcule la suma de los N primeros enteros, con N proporcionado por el usuario.



```
PPT VBA Ejemplos.xlsm - Module6 (Code)
(General) SumaNumeros

Option Explicit
Sub SumaNumeros()
    Dim i As Integer
    Dim suma As Integer, limite As Integer
    limite = CInt(InputBox("Ingrese cantidad de números"))
    suma = 0
    i = 1
    Do While i <= limite
        suma = suma + i
        i = i + 1
    Loop
    MsgBox suma
End Sub
```