

Variables

Variables

□ espacio reservado en la memoria que almacena un conjunto de información, haciendo referencia al mismo por un nombre y correspondiéndole un tipo de dato.

▪ Declaración:

Dim nombre_variable As tipo_dato

Tipos de Datos

- **Boolean (2 Bytes)**
 - **Un valor verdadero o falso.**
 - **Valores posibles:**
 - **True**
 - **False**
 - ***Dim vacio As Boolean = True***

Tipos de Datos

□ Byte (1 Byte)

- **Un valor positivo, sin signo, para contener datos binarios.**
- **Valores posibles: de 0 a 255**
- **Puede convertirse a:**
 - **Short, Integer, Long.**
 - **Single, Double o Decimal**
- **Dim *unByte As Byte* = 129**

Tipos de Datos

□ Char (2 Bytes)

- **Un caracter Unicode.**
- **Valores posibles: de 0 a 65535 (sin signo).**
- **No se puede convertir directamente a un tipo numérico.**
- ***Dim caracter As Char = "L"***

Tipos de Datos

□ Date (8 Bytes)

- Una fecha
- Valores posibles:
 - Del **1/1/0001** al **12/31/9999**
 - Se recomienda que las fechas se representen entre # y usando el formato:
 - > **#mm/dd/aaaa#**

Dim d As Date = #10/27/2013#

Tipos de Datos

□ Date (8 Bytes)

- Una fecha y/o una hora.
- Valores posibles:
 - Desde **1/1/0001 0:00:00**
 - Hasta **12/31/9999 23:59:59**
 - Se recomienda que las fechas se representen entre # y usando el formato:
 - **#m-d-aaaa hh:MM:ss#**

Dim d As Date = #10-27-2001 12:01#

Tipos de Datos

□ Decimal (16 Bytes)

- Un número decimal.
- Valores:
- de 0 a +/- 79, 228, 162, 514, 264, 337, 593, 543, 950, 335 sin decimales
- de 0 a +/- 7.9228162514264337593543950335 con 28 lugares a la derecha del decimal
- el número más pequeño es:
+/-0.00000000000000000000000000000001

Tipos de Datos

□ Decimal (16 Bytes)

- El separador decimal es el punto(.)
- Ejemplos:

Dim unDecimal As Decimal

=9223372036854775808

Dim unDecimal2 As Decimal

=987654321.125

Tipos de Datos

□ Double (8 Bytes)

- Un número de punto (coma) flotante de doble precisión.
- Valores:
 - de $-1.79769313486231570E+308$
 - a $-4.94065645841246544E-324$ para valores negativos.
 - de $4.94065645841246544E-324$
 - a $1.79769313486231570E+308$ para valores positivos.

Tipos de Datos

□ Double (8 Bytes)

- Se puede convertir a Decimal.
- Se puede indicar que un número es Double agregándole al final # o @
- **Dim unDoble As Double = 125897.0235#**
- **Dim unDoble2 As Double = 987456.0125@**

Tipos de Datos

□ Integer (4 Bytes)

- Un número entero (sin decimales)
- Valores:
- de -2,147,483,648 a 2,147,483,647.
- Se puede convertir a Long, Single, Double o Decimal
- Se puede utilizar la letra i o el signo % para indicar que es un número entero:

Dim unEntero As Integer = 250009i

Dim unEntero2 As Integer = 652000%

Tipos de Datos

□ Long (8 Bytes)

- Un número entero largo o grande.
- Valores:
- de -9,223,372,036,854,775,808
- a 9,223,372,036,854,775,807
- Se puede convertir a Single, Double o Decimal.
- Se puede utilizar la letra L o el signo & para indicar que un número es Long

Dim unLong As Long = 12345678L

Dim unLong2 As Long = 1234567890&

Tipos de Datos

□ Object (4 Bytes)

- **Cualquier tipo de dato se puede almacenar en una variable de tipo Object.**
- **Todos los datos que se manejan en .NET están basados en el tipo Object.**

Tipos de Datos

□ Short (2 Bytes)

- **Un entero corto o pequeño**
- **Valores:**
 - **de -32,768 a 32,767.**
- **Se puede convertir a: Integer, Long, Single, Double o Decimal.**
- **Se puede utilizar la letra S para indicar que es un número entero es corto:**

Dim unShort As Short = 32000S

Tipos de Datos

□ Single (4 Bytes)

- **Número de punto (coma) flotante de precisión simple.**
- **Valores:**
- **de $-3.4028235E+38$ a $-1.401298E-45$ para valores negativos.**
- **de $1.401298E-45$ a $3.4028235E+38$ para valores positivos.**
- **Se puede convertir a: Double o Decimal**
- **Se pueden usar la letra F o el símbolo ! para indicar que un número es Single**

Tipos de Datos

- **String (2 Bytes por caracter)**
 - **Una cadena de caracteres Unicode.**
 - **Valores:**
 - **de 0 a 2^{31} caracteres Unicode.**

Dim oracion As String = "Hola Mundo".

Constantes

□ Constantes

- **Const MaximoValor = 500**
- **Const Direccion =
“C:\Miss”**

Variables

□ Alcance de las variables

▪ Locales

- Al procedimiento o función en donde se declaren
- Al formulario si se declaran antes de los Procedimientos de Eventos

▪ Globales

- Se deben declarar en un módulo
- **Global** nombre_variable as
Integer

Variables

□ Operadores Matemáticos

Operador Operación

+	Suma - Concatena
-	Resta
*	Multiplicación
/	División Real
\	División Entera
Mod	Resto de la División Entera
&	Concatena
=	Asignación

Variables

❑ Operadores de Comparación

Operador	Operación
<	Menor que
<=	Menor o igual
=	Igual
>=	Mayor o igual
>	Mayor que
<>	Distinto

Variables

□ Operadores Lógicos

Operador	Operación	En Java
AND	Y lógico	&&
OR	O Lógico	
NOT	Negación	!
AndAlso	Y también	
OrElse	O sino	

Conversión entre tipos de datos

- ❑ **Existen dos formas de realizar conversiones entre tipos de datos distintos.**
- ❑ **A través de:**
 - **Métodos**
 - **Funciones**

Conversión entre tipos de datos

□ A través de Métodos

- Cada tipo de dato provee en su clase base dos métodos para realizar conversiones entre “Strings” y el tipo de dato específico.
 - `TipoDeDato.Parse(String)`
 - `TipoDeDato.TryParse(String, var)`
 - Devuelve verdadero o falso dependiendo si logró o no realizar la conversión

Conversión entre tipos de datos

□ A través de Métodos

- **Ejemplo Parse:**

```
Dim i As Integer
```

```
i = Integer.Parse("1234")
```

- **Ejemplo TryParse**

```
Dim j as Long
```

```
If Long.TryParse("?", j) Then
```

```
    'OK
```

```
Else
```

```
    'Error
```

```
End If
```

Conversión entre tipos de datos

□ **CBool(*expresión*)**

- **Devuelve:**
 - **Boolean**
- **Expresión:**
 - **Cualquier expresión numérica o de cadena de caracteres.**

Conversión entre tipos de datos

□ **CBool(*expresión*) - Ejemplo:**

Dim a, b, c As Integer

Dim verificar as Boolean

a=5 : b=5

verificar = CBool (a = b)

– **verificar --> True**

c=0

verificar = CBool(c)

– **verificar --> False**

Conversión entre tipos de datos

□ **CByte(*expresión*)**

- **Devuelve:**
 - **Byte**
- **Expresión:**
 - **De 0 a 255**
 - **Las fracciones se redondean.**

Conversión entre tipos de datos

□ CByte(*expresión*) - Ejemplo

Dim unDouble as Double = 126.5678

Dim unByte as Byte

unByte = CByte(unDouble)

– unByte --> 127

Conversión entre tipos de datos

□ **CChar(*expresión*)**

- **Devuelve:**
 - **Char**
- **Expresión:**
 - **Cualquier cadena.**
 - **Los valores deben ser de 0 a 65535**

Conversión entre tipos de datos

□ CChar(*expresión*) - Ejemplo

Dim unString As String

Dim unCaracter as Char

unString="ABCD"

unCaracter=CChar(unString)

- **UnCaracter --> "A"**
- **Solo la primer letra**

Conversión entre tipos de datos

□ **CDbl(*expresión*)**

- **Devuelve:**
 - **Double**
- **Expresión:**
 - **Cualquier valor Double.**
 - **Ver los valores posibles en este material.**

Conversión entre tipos de datos

□ **Cdbl(*expresión*) - Ejemplo**

Dim unDble as Double

Dim unDec as Decimal

unDec = 0.000000000000123D

unDble = Cdbl(unDec * 0.00001D)

– UnDble --> 1.23E-16

Conversión entre tipos de datos

□ CDec(*expresión*)

- **Devuelve:**
 - **Decimal**
- **Expresión:**
 - **Cualquier valor Decimal.**
 - **Ver los valores posibles en este material.**

Conversión entre tipos de datos

□ **CDec(*expresión*) - Ejemplo**

Dim unDec As Decimal

Dim unDble As Double

unDble = 1.23E-19

unDec = CDec(unDble / 1.0E-19)

– unDec --> 1.23

Conversión entre tipos de datos

□ **CInt(*expresión*)**

- **Devuelve:**
 - **Integer**
- **Expresión:**
 - **Cualquier valor Integer.**
 - **Las fracciones se redondean.**
 - **Ver los valores posibles en este material.**

Conversión entre tipos de datos

□ **CInt(*expresión*) - Ejemplo**

Dim entero As Integer

Dim unDbl As Double

unDbl = 234.5678

entero = CInt(unDbl)

– entero --> 235

Conversión entre tipos de datos

□ CShort(*expresión*)

- **Devuelve:**
 - **Short**
- **Expresión:**
 - **Cualquier valor Short.**
 - **Las fracciones se redondean.**
 - **Ver los valores posibles en este material.**

Conversión entre tipos de datos

□ **CLong(*expresión*)**

- **Devuelve:**
 - **Long**
- **Expresión:**
 - **Cualquier valor Long.**
 - **Las fracciones se redondean.**
 - **Ver los valores posibles en este material.**

Conversión entre tipos de datos

□ CType(*expresión, tipo*)

- **Devuelve:**
 - El *tipo* de datos indicado
- **Expresión:**
 - Cualquier tipo de dato.

Conversión entre tipos de datos

□ CStr(*expresión*)

– **Devuelve:**

- **String**

– **Expresión:**

- **Depende del tipo de datos de la *expresión***

- **Todos los objetos de vb.NET tienen un método para convertir un número a una cadena.**

Objeto.ToString

Conversión entre tipos de datos

□ CSng(*expresión*)

- **Devuelve:**
 - **Single**
- **Expresión:**
 - **Cualquier valor Single.**
 - **Ver los valores posibles en este material.**

Conversión entre tipos de datos

□ CSng(*expresión*) - Ejemplo

Dim unSin as Single

Dim unDec as Decimal

unDec = 0.000000000000123D

unSin = Cdbl(unDec * 0.00001D)

– UnSin --> 1.23E-16

Conversión entre tipos de datos

□ **Val(*expresión*)**

- **Devuelve:**
 - **Double**
- **Expresión:**
 - **Una cadena de caracteres (string).**

Conversión entre tipos de datos

□ **Fix(*expresión*)**

– **Devuelve:**

- **La parte entera. Depende del tipo de datos de la *expresión***

– **Expresión:**

- **Cualquier tipo de datos.**

Ejemplo:

Dim a As Decimal = -9.23

Fix(a) --> devuelve “-9”

Conversión entre tipos de datos

□ **Int(*expresión*)**

– **Devuelve:**

- **La parte entera. Depende del tipo de datos de la *expresión***

– **Expresión:**

- **Cualquier tipo de datos.**

Ejemplo:

Dim a As Decimal = -9.23

Int(a) --> devuelve “-10”

Conversión entre tipos de datos

□ **CObj(*expresión*)**

- **Devuelve:**
 - **Object**
- **Expresión:**
 - **Cualquier expresión válida.**