



**Universidad  
Nacional de  
Cajamarca**  
"Norte de la Universidad Peruana"

# Variables y Tipos de datos en C#

## Programa Aplicada I

**Ing. Roger M. Sánchez Chávez**

Ingeniería de Sistemas

Facultad de Ingeniería

Universidad Nacional de Cajamarca

**2023-II**

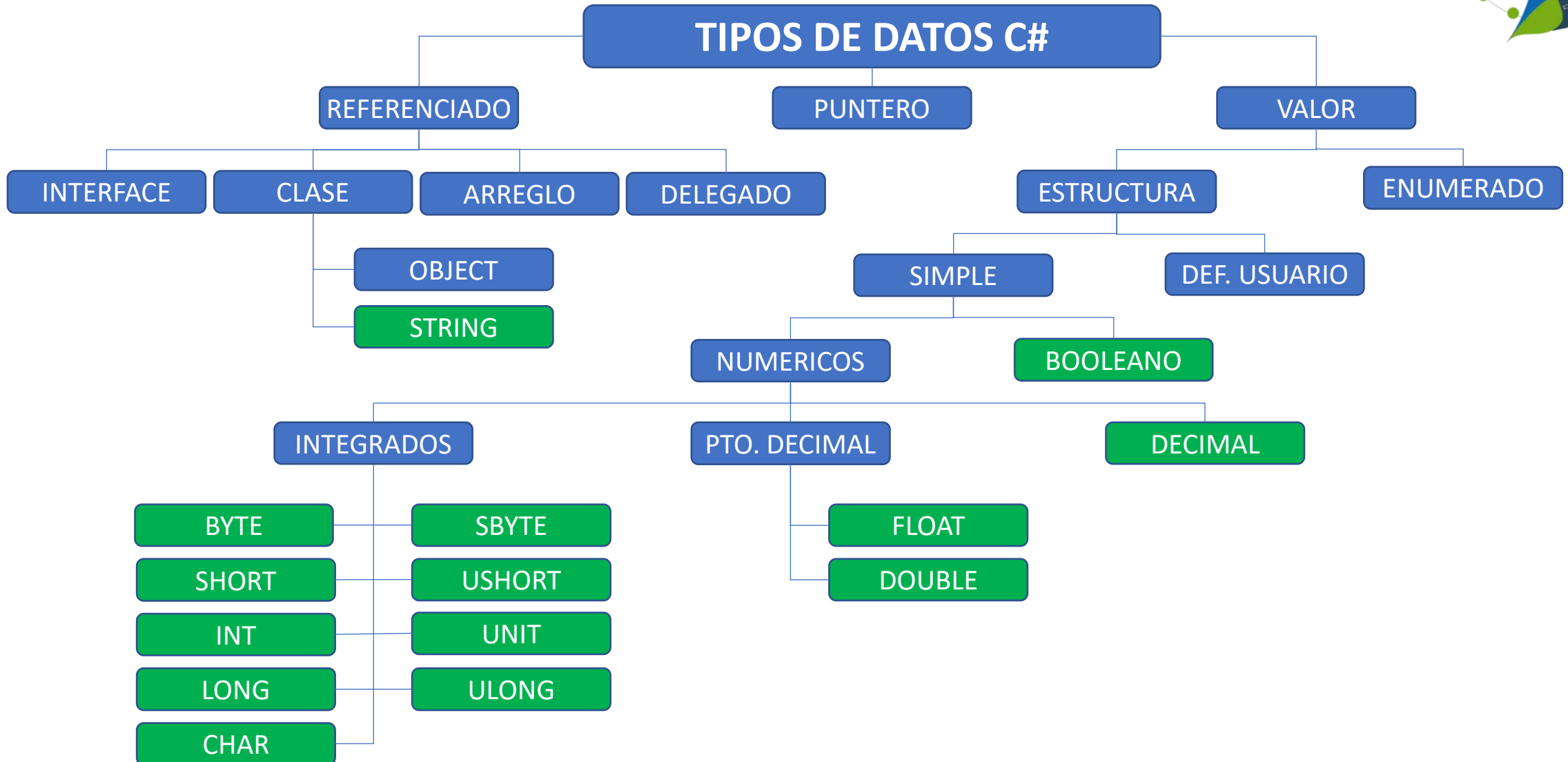


# Contenidos

- Tipos de datos
- Identificadores
- Palabras reservadas
- Literales
- Métodos matemáticos
- Conversiones
- Windows
- Label
- TextBox
- Button
- Ejercicios



# Tipos de datos - Clasificación





# Tipos de datos - Capacidad

- Tipos Numéricos

Tipo	Tamaño	Capacidad	System de .Net
byte	1 byte	0 a 255	Byte
sbyte	1 byte	-128 a 127	Sbyte
short	2 bytes	-32768 a +32767	short
ushort	2bytes	0 a 65535	ushort
int	4 bytes	- 2147483648 a 2147483647	Int32
uint	4 bytes	0 a 4294967295	UInt32
long	8 bytes	-9223372036854775808 a 9223372036854775807	Int64
ulong	8 bytes	0 a 18446744073709551615	UInt64



# Tipos de datos - Capacidad

- Tipos Numéricos Decimales

Tipo	Tamaño	Capacidad	System de .Net
float	4 bytes	$\pm 1.401298E-45$ a $\pm 3.402823E+38$	Single
double	8 bytes	$\pm 4.94065645841246E-324$ a $\pm 1.79769313486232E+308$	Double
decimal	16 bytes	-7.9228162514264337593543950335 a +7.9228162514264337593543950335	Decimal

- Tipos Caracter

Tipo	Tamaño	Capacidad	System de .Net
char	4 bytes	'\u0000' a '\uFFFF'	Char



# Tipos de datos - Capacidad

- Tipos Boolean

Tipo	Tamaño	Capacidad	System de .Net
bool	1 byte	true o false	Boolean

- Tipos Cadena de caracteres

Tipo	Tamaño	Capacidad	System de .Net
String	Variable	El permitido por la memoria	String

- Tipos Fecha

Tipo	Tamaño	Capacidad	System de .Net
DateTime	4 bytes	{00:00:00 AM} equivale a {12:00:00 AM}, medianoche. {00:00:00 PM} equivale a {12:00:00 PM}, mediodía. {00:00:00} a {11:59:59} equivale a {12:00:00 AM} a {11:59:59 AM} {12:00:00} a {23:59:59} equivale a {12:00:00 PM} a {11:59:59 PM}	Char



# Identificadores

Son símbolos léxicos que permiten asignar un nombre a una entidad. Se usa para la identificación de los diversos elementos de una aplicación, como variables, atributos de clase, constantes, métodos, clases, estructuras, etc.

Valores de entrada y salida	Posibles identificadores (válido)	Identificadores no válidos
Nombre del producto	producto, nomProducto	@producto_
Cantidad comprada	cantidad, unidades, cant	_ cantidad,
Precio del producto	precio, precio_producto, costo	precio del producto
Importe de compra	importeCompra, iCompra	iCompr@
Importe de descuento	importeDescuento, iDescuento	10 % Descuento
Importe a pagar	importePagar, iPagar	Importe a Pagar



# Palabras reservadas

Son identificadores previamente definidos y reservados por el lenguaje los cuales tienen un significado especial para el compilador

abstract	continue	extern	in	Object	return	this	using
as	decimal	false	int	operador	sbyte	throw	virtual
base	default	finally	interface	out	sealed	true	void
bool	delegate	fixed	internal	override	short	try	volatile
break	do	float	is	params	sizeof	typeof	while
byte	double	for	lock	private	stackalloc	uint	
case	else	foreach	Long	protected	static	ulong	
catch	enum	goto	namespace	public	string	unchecked	
char	Event	if	new	readonly	struct	unsafe	
checked	explicit	implicit	null	ref	switch	ushort	





# Literales

Representa un determinado valor según el tipo de dato

Tipos de datos	Literal	Literal personalizado
int	10	10
double	23.5D	23.5
decimal	780.50M	780
long	120L	120
char	'A'	'A'
String	"Visual C#"	"Visual C#"
bool	true	true
DateTime	"10/10/2019"	"10/10/2019"

Literal	Descripción
\'	Comilla simples en cadena de caracteres
\"	Comillas dobles en cadena de caracteres
\\	Barra inclinada en cadena de caracteres
\n	Nueva línea en cadena de caracteres
\t	Tabulación en cadena de caracteres



# Métodos matemáticos

La clase Math, que proviene del espacio de nombres System, permite desarrollar expresiones trigonométrica, logarítmicas y otras funciones matemáticas.

Método	Descripción	Ejemplo
<b>Abs</b>	Método que determina el valor absoluto de un valor.	int n=-10; int valorAbsoluto = Math.Abs(n);
<b>Ceiling</b>	Método que determina el valor más pequeño que es mayor o igual que el número decimal especificado.	double n = 2.1; int valorCeiling = (int) Math.Ceiling(n); //Resultado > 3
<b>Floor</b>	Método que determina el valor más grande menor o igual que el número decimal especificado	double n = 2.7; int valorFloor = (int) Math.Floor(n); //Resultado > 2
<b>Max</b>	Determina el mayor valor entre dos valores del mismo tipo.	int n1=10, n2=20; int mayor = Math.Max(n1, n2);
<b>Min</b>	Determina el menor valor entre dos valores del mismo tipo.	int n1=10, n2=20; int menor = Math.Min(n1, n2);
<b>Pow</b>	Determina el valor de la potencia entre un número y su base.	//Elevando un numero al cuadrado int n=2; int potencia = (int)Math.Pow(n,2);  //Raiz al cubo de un numero int n=3; double raizC= Math.Pow(n , 1.0/3); //Respuesta > 1.44224957030741
<b>Round</b>	Determina el redondeo de un valor al integral más próximo.	double n = 2.675; double valorR = Math.Round(n,2); //Respuesta > 2.68
<b>Sqrt</b>	Determina la raíz cuadrada de un valor.	double n = 2; double raizC = Math.Sqrt(n); //Respuesta > 1.4142135623731
<b>E</b>	Representa la base logarítmica natural, especificada para la constante e.	double e = Math.E; //Respuesta > 2.71828182845905
<b>PI</b>	Representa el valor de PI.	double e = Math.PI; //Respuesta > 3.14159265358979



# Conversiones de tipos en C#

Las conversiones de tipos se refieren a una conversión realizada a un valor previamente declarado como un tipo de datos

- Si se declara una variable como entera y el valor entero viene representado como cadena.
- Cuando se necesita concatenar o imprimir un valor numérico en una cadena de texto.
- Cuando dos valores numéricos necesitan concatenarse y no sumarse.
- Cuando el resultado de una expresión es diferente al tipo de datos declarado.
- Cuando la función envía un valor de retorno en un tipo de datos específico.





# Conversiones de tipos en C# - implícitas

Es toda conversión que se realiza con seguridad de tipos y sin pérdida de valor

Tipo original	Tipo destino
Sbyte	short , int, long, float, double o decimal
Byte	short , ushort, int, uint, long, ulong, float, double o decimal
short	int , long, float, double o decimal
ushort	int , uint, long, ulong, float, double o decimal
Int	long , float, double o decimal
UInt	long , ulong, float, double o decimal
Long	float , double o decimal
char	ushort , int, uint, long, ulong, float, double o decimal
float	Double
ulong	float , double o decimal

```
int n = 32767; //Tipo original  
long nLargo = n; //Tipo destino
```

```
//Resultado nLargo = 32767
```



# Conversiones de tipos en C# - explicitas

Es la conversión que se realiza con seguridad de tipos y con perdida de valor. Cuando se necesita el valor perdido, es mas conveniente usar una función de conversión

Tipo original	Tipo destino
sbyte	byte , ushort, uint, ulong o char
byte	Sbyte o char
short	sbyte , byte, ushort, uint, ulong o char
ushort	sbyte , byte, short o char
int	sbyte , byte, short, ushort, uint, ulong,o char
uint	sbyte , byte, short, ushort, int o char
Long	sbyte , byte, short, ushort, int, uint, ulong o char
ulong	sbyte , byte, short, ushort, int, uint, long o char
char	sbyte , byte o short
float	sbyte , byte, short, ushort, int, uint, long, ulong, char o decimal
double	sbyte , byte, short, ushort, int, uint, long, ulong, char, float o decimal
decimal	sbyte , byte, short, ushort, int, uint, long, ulong, char, float o double

```
double valor = 16.78; //Tipo original
int valorEntero = (int) valor; //Tipo destino

//Resultado valorEntero > 16
```

# Conversiones de tipos en C# - parse



Método Parse	Descripción	Ejemplo
Int.Parse()	Convierte el valor de una cadena en un valor numérico de tipo entero.	string n = "20"; int numero = int.Parse(n);
Int32.Parse()	Convierte el valor de una cadena en un valor numérico de tipo entero de 32 bits.	string n = "20"; Int32 numero = Int32.Parse(n);
Int64.Parse()	Convierte el valor de una cadena en un valor numérico de tipo entero de 64 bits.	string n = "20"; Int64 numero = Int64.Parse(n);
Double.Parse()	Convierte el valor de una cadena en un valor numérico de tipo double.	string n = "14.5"; double numero = Double.Parse(n);
DateTime.parse()	Convierte el valor de una cadena en un valor de tipo fecha.	string valor = "10/10/2015"; DateTime fecha = DateTime.Parse(valor);





# Conversiones de tipos en C# - Convert

Tipo	Método	Descripción
decimal	Convert.ToDecimal()	Convierte cualquier valor de tipo numérico a tipo decimal.  int n = 1500; decimal valorDecimal = Convert.ToDecimal(n);
double	Convert.ToDouble()	Convierte cualquier valor de tipo numérico a tipo double.  int n = 1500; double valorDouble = Convert.ToDouble(n);
short	Convert.ToInt16()	Convierte cualquier valor de tipo numérico a tipo short.  int n = 50; short valorCorto = Convert.ToInt16(n);
int	Convert.ToInt32()	Convierte cualquier valor de tipo numérico a tipo int.  double n = 1500; int valorEntero = Convert.ToInt32(n);
long	Convert.ToInt64()	Convierte cualquier valor de tipo numérico a tipo long.  int n = 1500; long valorLargo = Convert.ToInt64(n);



Universidad  
Nacional de  
Cajamarca  
*"Norte de la Universidad Peruana"*



MessageBox.Show("Muchas gracias");

