UTN F r a
Avellaneda

Programación II

Clase 01

EL LENGUAJE C#

Temario

Introducción a C#

- Lenguaje de Programación
- Plataforma .NET
- Caracterísitcas C#
- Compilación

Tipos de Datos

- Expresiones y valores
- Literales
- CTS
- Numéricos y Booleanos

Instrucciones

- Loops
- Condicionales

01.

Introducción a C#

Lenguaje de Máquina

Lenguaje de bajo nivel

- Unidad Lógico Aritmética (ALU)

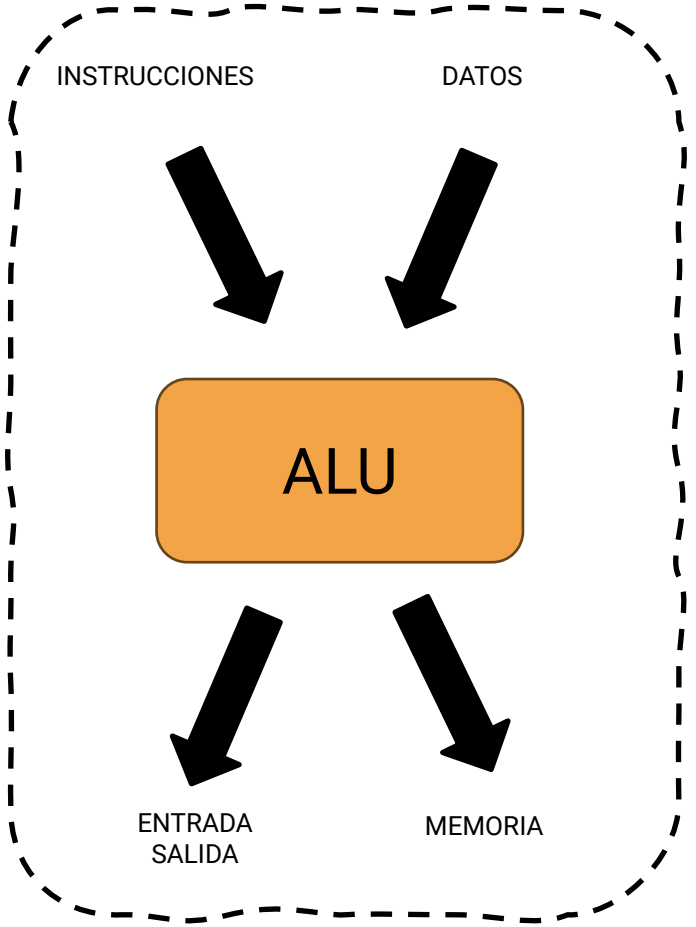
  Procesamiento de información.

  Loop constante.

- Instrucciones

  Alfabeto.

  Lenguaje de máquina.



CPU

CPU: Central Processing Unit

Intel Instruction Set: https://www.intel.com/content/www/us/en/developer/tools/isa-extensions/overview.html

Lenguaje de Programación

Lenguajes de alto nivel

- Lenguaje familiar para humanos

  Reconocible.

  Código fuente (source code).

- Paradigma.

  Orientado a objetos.

  Funcional.

- Compilado / Interpretado

AMBOS TRANSFORMAN CÓDIGO FUENTE EN CÓDIGO MÁQUINA

COMPILADO	**INTERPRETADO**
SE GENERA UN ARCHIVO ANTES DE LA EJECUCIÓN (BUILD)	SE GENERA EL CÓDIGO A MEDIDA QUE SE EJECUTA EL PROGRAMA (RUNTIME)

Lenguaje de Programación

Composición del lenguaje

- **Alfabeto**

  Símbolos que componen una “palabra”.

- **Diccionario** (léxico)

  Conjunto de palabras (todas) que tienen algún significado.

- **Sintaxis**

  Reglas para que una oración sea válida.

- **Semántica**

  Reglas para crear frases.

Lenguaje C#

Características

- **Forma parte del ecosistema .NET CORE**

  Plataforma para el desarrollo y la ejecución de programas escritos en C#.

  COMPILADOR | LIBRERÍAS BÁSICAS | MÁQUINA VIRTUAL

  Open source.
  FOMENTA CONTRIBUCIONES

  Multiplataforma.
  LINUX, MAC, WINDOWS

  Modular.
  COMPUESTO DE PAQUETES DISTRIBUIDOS A TRAVÉS DE NUGET.

Otros lenguajes en .NET: IronPython, VisualBasic, F#

Lenguaje C#

Características

- Orientado a Objeto
  - SOPORTE DE CONTENEDORES PARA DATOS Y MÉTODOS.
  - ADMINISTRACIÓN DE CLASES, OBJETOS E INSTANCIAS.
- Tipado estático (fuerte)
  - CONTROLA EL TIPADO DEL CÓDIGO.
  - AGREGA UNA CAPA DE CONFIABILIDAD.
- Compilado
  - MEJOR RENDIMIENTO.

UTN F r a Avellaneda

Compilación

- **Transformación de código fuente a código máquina.**

EL CÓDIGO ESCRITO POR EL DESARROLLADOR EN FORMATO TEXTO

EL CÓDIGO QUE ENTIENDE LA MÁQUINA (BINARIO)

Common Intermediate Language

- **Transformación de código fuente a código intermedio compatible con la suite .NET**

  Primera compilación.

  No tan legible como el código fuente.

  Diseñado como punto común entre distintos lenguajes permitiendo la compatibilidad de desarrollos.

UTN F r a Avellaneda

Common Language Runtime

- **Entorno de ejecución de aplicaciones**

Maneja la comunicación con el sistema operativo.

COMPILACIÓN DEPENDIENTE DE LA PLATAFORMA

Realiza la compilación Just-in-Time (JIT).

COMPILACIÓN EFICIENTE

ANTES DE USARSE POR PRIMERA VEZ

NO TODO EL CÓDIGO ES COMPILADO (BULK)

Administración de memoria.

Administración de threads (hilos).

02.

Tipos de Datos

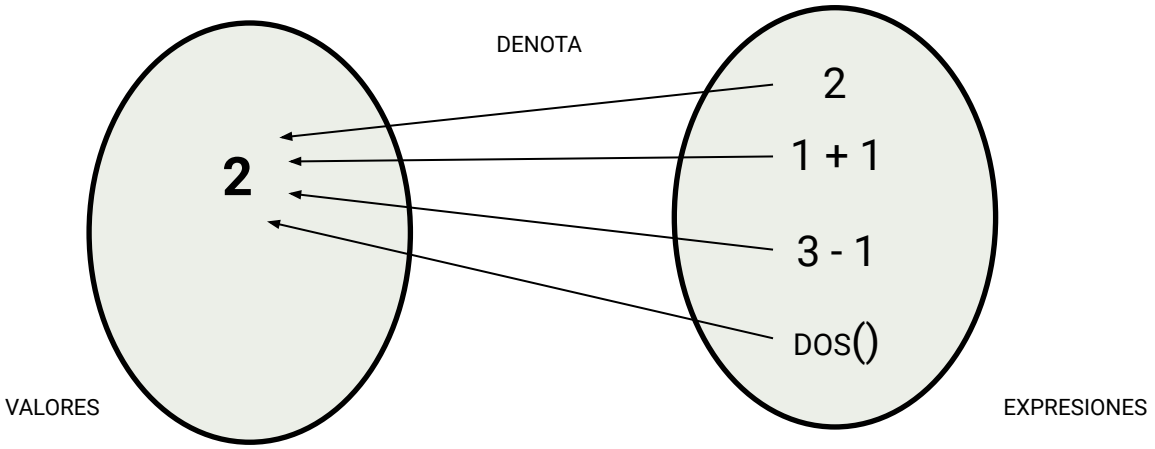
Tipos de Datos

Valores

- **Abstracto**
- **Conceptual**

Expresiones

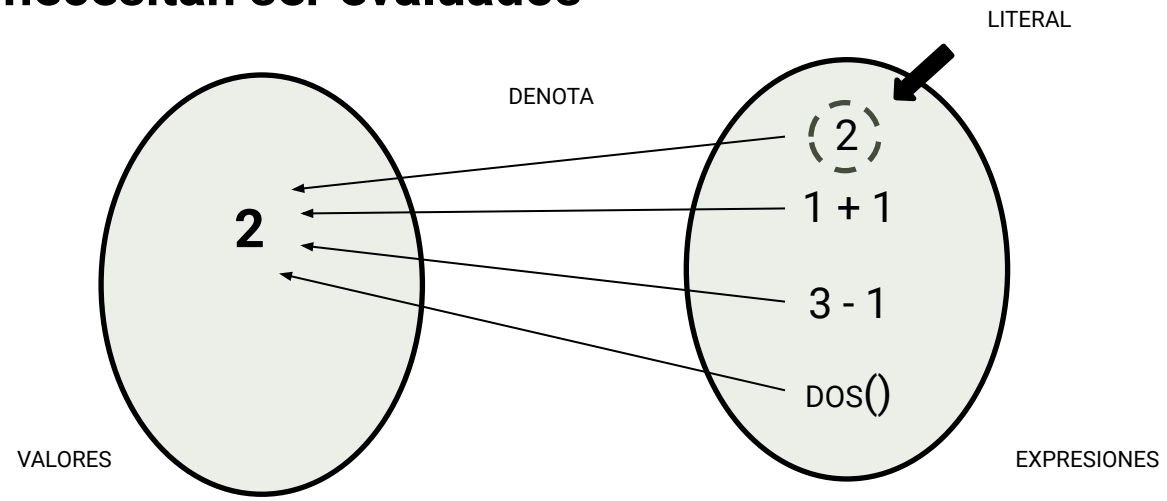
- **Concreto**
- **Evaluable**



Tipos de Datos

Literales

- **Expresan un valor**
- **No necesitan ser evaluados**



EJEMPLO LITERALES

BOOLEAN: true

CHAR: “a”

STRING: “hola”

INT: 2

Tipos de Datos

.NET CTS

- **Common Type System**

Provee los tipos utilizados en el ecosistema .NET.

TIPOS BÁSICOS: INT, CHAR, BOOLEAN, ...

Define las reglas para la creación de tipos.

DECLARACIÓN
UTILIZACIÓN
ADMINISTRACIÓN

} IL

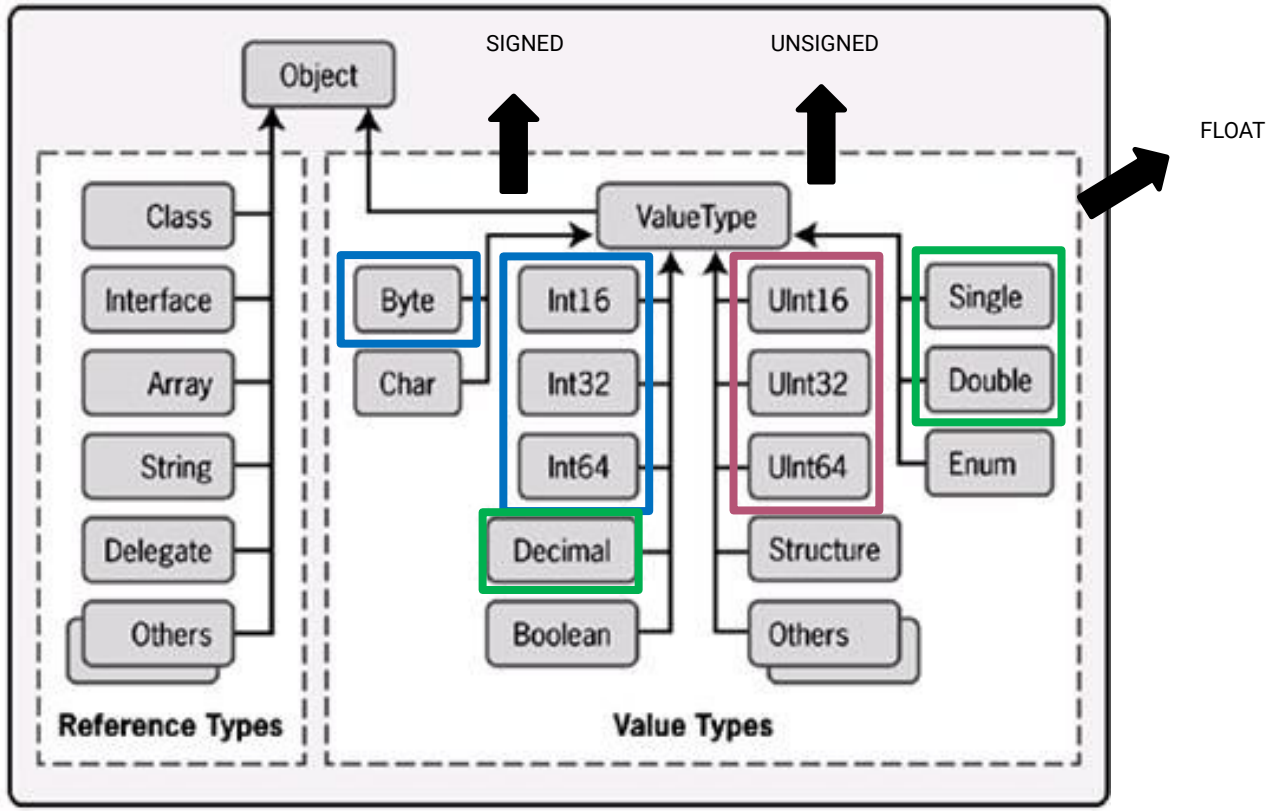
Tipos de Datos

.NET CTS

TIPO REFERENCIA

APUNTA A UN SECTOR DE MEMORIA DONDE SE ENCUENTRA EL VALOR REAL

SE ALMACENA EN MEMORIA **HEAP**



TIPO VALOR

REPRESENTA EL VALOR REAL

SE ALMACENA EN MEMORIA **STACK**

Tipos de Datos

Tipo Numéricos

RACIONALES

ENTEROS

NATURALES

BYTE	0 A 255
UINT**16**	0 A 65.535
UINT**32**	0 A 4.294.967.295
UINT**64**	0 A 18.446.744.073.709.551.615

SBYTE	-128 A 127
INT**16**	-32.768 A 32.767
INT**32**	-2.147.483.648 A 2.147.483.647
INT**64**	-9.223.372.036.854.775.808 A 9.223.372.036.854.775.807

SINGLE	6-9 DECIMALES
DOUBLE	15-17 DECIMALES
DECIMAL	28-29 DECIMALES

COMPLEJOS

COMPLEX

Tipos de Datos

Tipo Numéricos

- **Expresiones en Sistemas**

Decimal. BASE10 **17**

Binario. BASE2 **0B1001**

Hexadecimal. BASE16 **0X11**

SOLO CAMBIA LA **REPRESENTACIÓN**

Tipos de Datos

Tipo Numéricos

- **Operaciones aritméticas**

Operaciones binarias (dos componentes).

+ SUMA

2 + 2 = **4**

- RESTA

2 - 1 = **1**

***** MULTIPLICACIÓN

2 * 4 = **8**

/ DIVISIÓN

5 / 3 = **1**

% MÓDULO O RESTO

5 / 3 = **2**

INT OP. **INT** = **INT**

INT OP. **FLOAT** = **FLOAT**

FLOAT OP. **FLOAT** = **FLOAT**

Tipos de Datos

Tipo Numéricos

- **Operaciones aritméticas**

Operaciones unarias (un componente).

++ INCREMENTO

INT A = 4

++A = **5**

- - DECREMENTO

INT A = 4

--A = **3**

Tipos de Datos

Tipo Numéricos

- **Operaciones aritméticas**

  Prioridades

++ --
* / %
+ -

PRIMERO

ÚLTIMO

Tipos de Datos

Tipo Numéricos

- **Operaciones equidad y relaciones**

TRUE
FALSE

Operaciones binarias (dos componentes). Denotan un booleano

==	IGUAL 2 == 2 DENOTA **TRUE**	>	MAYOR
!=	DISTINTO 2 == 2 DENOTA **FALSE**	>=	MAYOR O IGUAL
		<	MENOR
		<=	MENOR O IGUAL

Tipos de Datos

Tipo Numéricos

- **Operaciones aritméticas y equidad**

Prioridades

++ --
* / %
+ -
> >= < <= == !=

PRIMERO

ÚLTIMO

Tipos de Datos

Tipo Numéricos

- **Operaciones asignación**

Operaciones binarias (dos componentes).

= ASIGNA

INT **A** = 1

A 1

+= SUMA Y ASIGNA

A += 3

A 4

-= RESTA Y ASIGNA

***=** MULTIPLICA Y ASIGNA

/= DIVIDE Y ASIGNA

%= RESTO Y ASIGNA

Tipos de Datos

Tipo Numéricos

- **Operaciones aritméticas, equidad, relaciones y asignación.**

Prioridades

++ --
* / %
+ -
> >= < <= == !=
= += -= *= /= %=

PRIMERO

ÚLTIMO

Tipos de Datos

Tipo Booleano

- **Denotan verdadero o falso**

Representa valores de la lógica binaria.
Tipo de dato lógico que se utilizan en álgebra booleana.

TRUE = 1
FALSE = 0

Tipos de Datos

Tipo Booleano

- **Operaciones de lógica booleana y “bitwise”.**

Operaciones binarias (dos componentes).

	OR LÓGICO DISYUNCIÓN
&	**AND** LÓGICO CONJUNCIÓN
^	**OR EXCLUSIVO**

ARGUMET A	ARGUMENT B	A & B	A | B	A ^ B
FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE

&& CORTOCIRCUITO

||

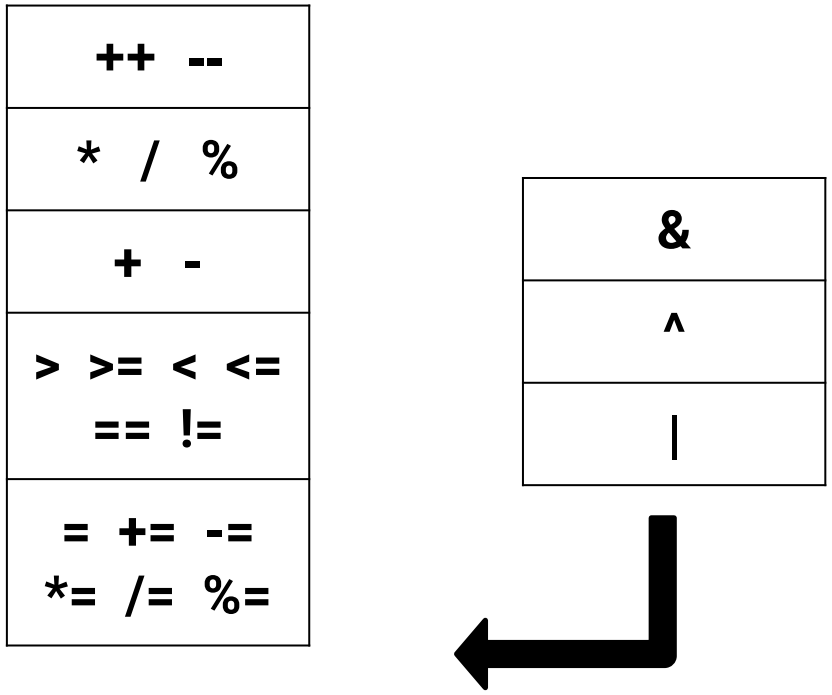
!FALSE = TRUE NEGACIÓN

Tipos de Datos

Tipo Booleano

- **Operaciones aritméticas, equidad, relaciones, asignación y lógica/bitwise.**

Prioridades



Tipos de Datos

Colecciones

- **Arreglos (ARRAY).**

  Mismo tipo.
  Longitud fija.
  Ordenadas.
  Mutables.
  Representación de matrices.
  N DIMENSIONALES

- **Listas.**

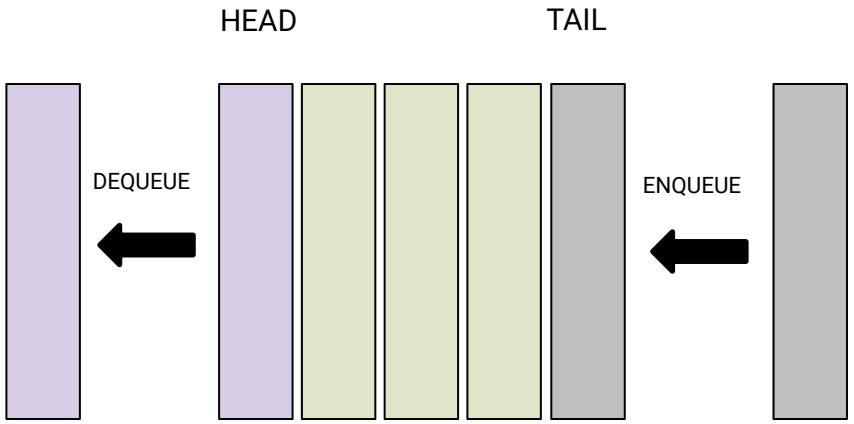
  Mismo tipo.
  Longitud variable.
  Ordenadas.
  Mutables.
  Representación de matrices.

  QUEUE **STACK** **LIST**

Tipos de Datos

Listas

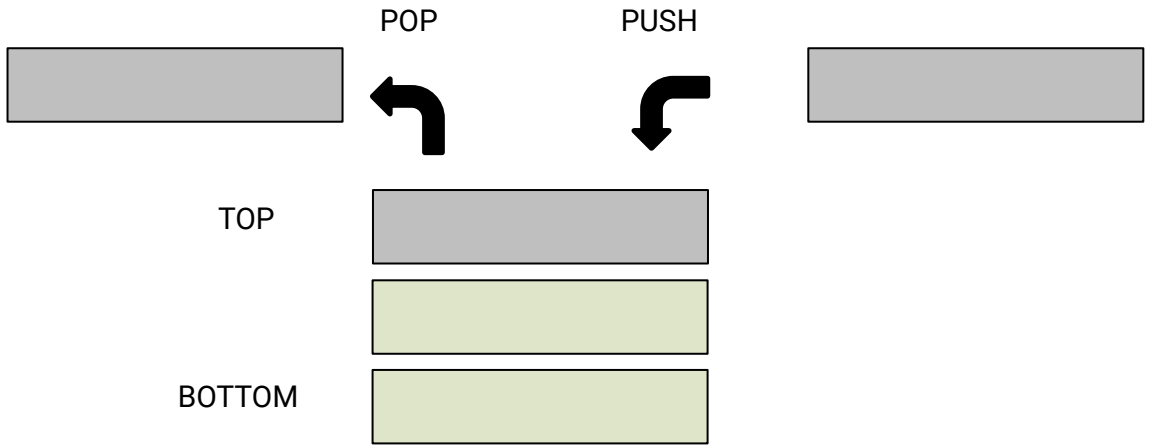
- **QUEUE**



FIFO

FIRST IN FIRST OUT

- **STACK**



LIFO

LAST IN FIRST OUT

02.

Instrucciones

Instrucciones

Iteraciones

- **También conocido como LOOP (bucle)**

BLOQUE DE CÓDIGO

Ejecuciones repetitivas hasta que se alcanza cierto **estado** o se cumple una **condición**.
Se pueden saltear iteraciones o finalizar loop explícitamente

CONTINUE
BREAK

WHILE

TOMA UNA EXPRESIÓN

SE EJECUTA HASTA QUE LA EXPRESIÓN SE EVALÚA **FALSE**

DO -WHILE

ES IGUAL A WHILE SÓLO QUE EN LA PRIMERA ITERACIÓN NO VERIFICA CONDICIÓN

FOR

TOMA TRES PARÁMETROS

1. SE EJECUTA ANTES
2. ESTABLECE LA CONDICIÓN
3. SE EJECUTA AL FINAL

FOREACH

TOMA UNA COLECCIÓN

SE EJECUTA TANTAS VECES COMO CANTIDAD DE ELEMENTOS TIENE

Instrucciones

Selección

- **También conocido como condicionales**

Establece la condición en la que se debe ejecutar un bloque de código.

	CONDICIÓN ES UNA EXPRESIÓN QUE SE EVALÚA **TRUE**	CONDICIÓN ES UNA EXPRESIÓN QUE SE EVALÚA EN UN VALOR DETERMINADO
	IF... ELSE IF.. ELSE	**SWITCH**
IF	SE EJECUTA SI CUMPLE UNA CONDICIÓN	TOMA UNA EXPRESIÓN
ELSE IF	IDEM IF	EVALÚA QUE CASO CUMPLE CON LA CONDICIÓN
ELSE	SE EJECUTA SI NO SE CUMPLIÓ NINGUNA CONDICIÓN	