

# codehouse {ACADEMY}



# Modulo 2 Introducción

# 1. Tipos de valor

## 1.1. Tipos simples

- ✓ Entero con signo: `sbyte`, `short`, `int`, `long`
- ✓ Entero sin signo: `byte`, `ushort`, `uint`, `ulong`
- ✓ Caracteres Unicode: `char`
- ✓ Punto flotante binario IEEE: `float`, `double`
- ✓ Punto flotante decimal de alta precisión: `decimal`
- ✓ Booleano: `bool`

## 1.2. Tipos de enumeración

- ✓ Tipos definidos por el usuario con el formato `enum E {...}`

## 1.3. Tipos de estructura

- ✓ Tipos definidos por el usuario con el formato `struct S {...}`

## 1.4. Tipos de valor que aceptan NULL

- ✓ Extensiones del resto de tipos con un valor `null`

## 2. Tipos de referencia

### 2.1. Tipos de clase

- ✓ Clase base definitiva de todos los tipos: `object`
- ✓ Cadenas Unicode: `string`
- ✓ Tipos de clase definidas por el usuario con el formato  
`class C {...}`

### 2.2. Tipos de interfaz

- ✓ Tipos definidos por el usuario con el formato  
`interface I {...}`

## 2.3. Tipos de matriz

- ✓ Unidimensional y multidimensional; por ejemplo,  
`int[...]`, `int[..., ...]`

## 2.4. Tipos delegados

- ✓ Tipos definidos por el usuario con el formato `delegate`  
`int D(...)`

## 3. Operadores

### 3.1. Operadores de asignación

- ✓ Operador de asignación =

### 3.2. Operadores Aritméticos

- ✓ Operadores unarios: ++(incremento), --(decremento), +(mas), -(menos)
- ✓ Operadores binarios: \*(multiplicación), /(división), %(resto), +(suma), -(resta)

## 3.3. Operadores lógicos booleanos

- ✓ Operador unario: `!(negación lógica)`
- ✓ Operadores binarios: `&(AND lógico)`, `|(OR lógico)`, `^(OR exclusivo lógico)`
- ✓ Operadores condicionales: `&&(AND lógico condicional)`, `||(OR lógico condicional)`

## 3.4. Operadores de comparación

- ✓ Operadores de igualdad: `==(igualdad)`, `!=(desigualdad)`
- ✓ Operadores de comparación: `<(menor que)`, `>(mayor que)`, `<=(menor o igual)`, `>=(mayor o igual)`



## 3.5. Operadores de acceso a miembros

- ✓ `.` Acceso a miembros
- ✓ `[]` elemento de matriz o acceso a indizador
- ✓ `?` `Y` `[]` operadores condicionales NULL
- ✓ `()` invocación
- ✓ `^` índice desde el final
- ✓ `..` intervalo

# 4. Conversiones de tipos

## 4.1. Tipos

- ✓ Conversiones implícitas
- ✓ Conversiones explícitas (casting)
- ✓ Conversiones definidas por el usuario
- ✓ Conversiones con clases del asistente

## 4.2. Boxing

## 4.3. Unboxing

codehouse  
{ ACADEMY }