

**Variable: elemento cuyo valor puede cambiar**

```
string ciudad = "Mérida"
```



**Tipo de dato**  
string



**Nombre**  
ciudad



**Valor**  
"Mérida"

# Tipos de datos

## Texto:

- **string** – cadena de caracteres (palabras)
- **char** – un solo carácter

## Números:

- **int** – enteros
- **decimal** – decimales

## Booleanos:

- **bool** – verdadero o falso

# Tipos de datos

## Cualquier tipo:

- **var** – cualquier tipo de dato

Cuando no se sabe con certeza el tipo,  
resultado de alguna operación

# Operaciones aritméticas

|          |                |
|----------|----------------|
| <b>P</b> | Paréntesis     |
| <b>E</b> | Exponentes     |
| <b>M</b> | Multiplicación |
| <b>D</b> | División       |
| <b>A</b> | Adición        |
| <b>S</b> | Sustracción    |



# Operadores relacionales

$X == Z$

X igual a Z

$X != Z$

X diferente a Z

$X < Z$

X menor a Z

$X > Z$

X mayor a Z

$X <= Z$

X menor o igual a Z

$X >= Z$

X mayor o igual a Z

# Métodos

`decimal CalcularIVA(decimal precio){...}`



**Valor de retorno**  
`decimal`



**Nombre**  
`CalcularIVA`



**Parámetro**  
`precio`



**Bloque/  
cuerpo**

**Arreglo: variable que contiene múltiples  
elementos del mismo tipo**

Una vez que se define su tamaño, no se puede  
modificar



**Lista: variable que contiene una colección de elementos del mismo tipo**

A diferencia del arreglo, su tamaño es dinámico

# Operadores lógicos

AND     $X \&\& Z$     Verdadero cuando ambos son verdaderos

OR     $X || Z$     Verdadero cuando por lo menos uno es verdadero

NOT     $!X$     Modifica al valor opuesto

# Trabajar con archivos

La **estructura de los directorios** varía entre los diferentes sistemas operativos.

Por ejemplo, la ruta al directorio **Documents** es la siguiente:

- **Windows**: C:\Users\'usuario'\Documents
- **Linux**: home\'usuario'\Documents

.NET incluye constantes y métodos para trabajar con rutas de archivos de manera constante **independientemente del SO**.

# Excepciones

Errores que ocurren durante la **ejecución** de la aplicación.

**try**

Bloque de código a “probar”; se espera algún tipo de excepción.

**catch**

Bloque que se ejecuta para tratar con un tipo de excepción.

**finally**

Bloque que se ejecuta siempre, si ocurre una excepción o no (se utiliza en casos muy específicos).

# Referencias y dependencias

**using** Palabra clave (keyword) para referenciar un espacio de nombres (namespace) que contiene clases que queremos usar.

Podemos referenciar clases definidas por .NET (.NET Class Library), por nosotros mismos o por un tercero.

**nuget** es el administrador de paquetes (package manager) de .NET