

# Programación I (Codificación)

# Del código al programa

El proceso de transformar un código fuente en un programa que contiene las instrucciones que sean comprensibles por la computadora se llama compilación.



# Codificación

Cada elemento que utilicemos en un lenguaje de programación debe estar sujeto a una estricta sintaxis. Los elementos que el lenguaje admite son:

- Variables y constantes
- Operadores
- Expresiones
- Palabras reservadas del lenguaje

# Tipos de datos

Son los tipos de valores que se pueden almacenar.  
Algunos tipos de datos en C++

- int
- float
- bool
- char
- string

# Tipos de datos: enteros

Se utilizan para almacenar números enteros, positivos o negativos, sin decimales.

Ejemplo:

```
int edad = 25;
```

# Tipos de datos: flotantes

Para representar números decimales o en punto flotante.

Ejemplo:

```
float precio = 19.99;
```

# Tipos de datos: booleanos

Para representar valores de verdad: true o false.

Ejemplo:

```
bool esMayor = true;  
bool aprobado = false;
```

# Tipos de datos: caracteres

En C++, el tipo de dato `char` se utiliza para almacenar un carácter individual, como una letra, un número o un símbolo especial. Representa caracteres utilizando la codificación ASCII.

Ejemplo:

```
char letra = 'A';           // Almacena la letra 'A'  
char simbolo = '#';        // Almacena un símbolo  
char numero = '5';         // Almacena el carácter '5',  
                           // no el número 5
```



# Tipos de datos: cadenas de texto

En C++, el tipo de dato string se utiliza para representar y manipular cadenas de texto (es decir, una secuencia de caracteres).

A diferencia de char, que almacena un único carácter, string puede almacenar una colección de caracteres formando una palabra, oración o cualquier texto.

Ejemplo:

```
string saludo = "Hola, mundo!";
```

# Variables

Representación simbólica de la memoria.

Ejemplo

```
int edad;  
edad = 20;
```

Al espacio de memoria identificado  
por edad se le asignó el valor 20.

XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX
XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX
20 edad	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX
XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX
XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX

# Variables

## Definición de variables

```
int edad = 25;  
char caracter = 'J';  
float precio = 99.99;  
bool aprobado = true;  
string nombre = "Juan";
```

# Constantes

Representación simbólica de la memoria. Es donde se almacenan los datos en procesamiento.

Una constante se identifica con la palabra reservada **const**, un tipo, un nombre (lo elige el programador), y un valor que no puede ser modificado durante el transcurso del programa en ejecución.

# Constantes

## Definición de constantes

```
const int EDAD_MIN = 18;  
const float IMPUESTO = 10.5;  
const char PAIS = 'A';
```

# Expresiones

Conjunto de variables, constantes, números y operadores ordenados de acuerdo a las reglas sintácticas establecidas en el lenguaje de programación.

Tienen como objetivo la construcción de instrucciones para la resolución del problema (o de parte del problema) planteado.

Ejemplos

10

50 + 100

aux - 20

'B'

# Operadores

Conjunto de símbolos y palabras reservadas que nos permiten hacer operaciones con expresiones.

Existen diferentes categorías de operadores:

- Asignación
- Matemáticos
- Relacionales
- Lógicos
- De dirección e indirección
- Para memoria dinámica
- Etcétera

# Operadores matemáticos

Necesarios para realizar cálculos matemáticos. Los paréntesis tienen el mismo efecto que en la matemática en las expresiones algebraicas. Sin embargo, en programación no se utilizan corchetes ni llaves para la separación de términos.

OPERADOR	OPERACIÓN
+	Suma
-	Resta
*	Producto
/	División real Cociente de la división entera
%	Resto de la división entera



# Operadores matemáticos

## Ejemplos

$((2+3)*5)+10$

$2+3*5+10$

$5 / 2$

$5.0 / 2$

$5 \% 2$

# Operadores matemáticos

## Ejemplos

$((2+3)*5)+10$

$2+3*5+10$

$5 / 2$

$5.0 / 2$

$5 \% 2$

$((2+3)*5)+10 \rightarrow 35$

$2+3*5+10 \rightarrow 27$

$5 / 2 \rightarrow 2$

$5.0 / 2 \rightarrow 2.5$

$5 \% 2 \rightarrow 1$

# ¿División real vs División entera?



Tenemos seis kilos de helado para repartir entre ocho personas. ¿Cuántos kilos les corresponde a cada uno?



Tenemos diecisiete chupetines para repartir entre cinco personas. ¿Cuántas chupetines les corresponde a cada uno?

# ¿División real vs División entera?



Tenemos seis kilos de helado para repartir entre ocho personas. ¿Cuántos kilos le corresponde a cada uno?

$$\begin{array}{r} 6 \overline{) 8} \\ 60 \\ 40 \\ 0 \end{array}$$

Corresponde 0.75 kilos por persona



Tenemos diecisiete chupetines para repartir entre cinco personas. ¿Cuántas chupetines le corresponde a cada uno?

$$\begin{array}{r} 17 \overline{) 5} \\ 2 \\ 3 \end{array}$$

Resto      Cociente

Corresponden 3 chupetines por persona y sobran 2 chupetines.

# Operadores relacionales

Son necesarios para decisiones y ciclos. Nos permiten establecer proposiciones lógicas. El resultado de una proposición lógica puede ser **verdadero** o **falso**.

OPERADOR	SIGNIFICADO
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
==	Igual que
!=	Distinto que

# Operadores lógicos

Nos permiten combinar dos o más proposiciones lógicas

OPERADOR	SIGNIFICADO
&& (AND)	Y lógico
(OR)	O lógico
! (NOT)	Negación

A	B	A && B
verdadero	verdadero	verdadero
verdadero	falso	falso
falso	verdadero	falso
falso	falso	falso

A	B	A    B
verdadero	verdadero	verdadero
verdadero	falso	verdadero
falso	verdadero	verdadero
falso	falso	falso

# Operador de asignación

Nos permite asignar la expresión que se encuentra a la derecha del operador en la variable que se encuentra a la izquierda.

Correcto

```
edad = 50;  
a = b;  
nombre = "Angel";  
caracter = 'X';  
precio = cant * pu;  
cont = cont + 1;
```



Error de sintaxis

```
50 = edad;  
50 = 50;
```



Sintácticamente correcto

```
pero sin sentido  
edad = edad;
```