

# Fundamentos de Programación

## Variables y Operadores

# Escribir Código es Escribir para Humanos

```
/*
```

```
Un nombre largo y descriptivo es mejor que un nombre corto y  
enigmático.
```

```
Un nombre largo y descriptivo es mejor que un comentario largo  
y descriptivo.
```

```
-- Robert C. Martin
```

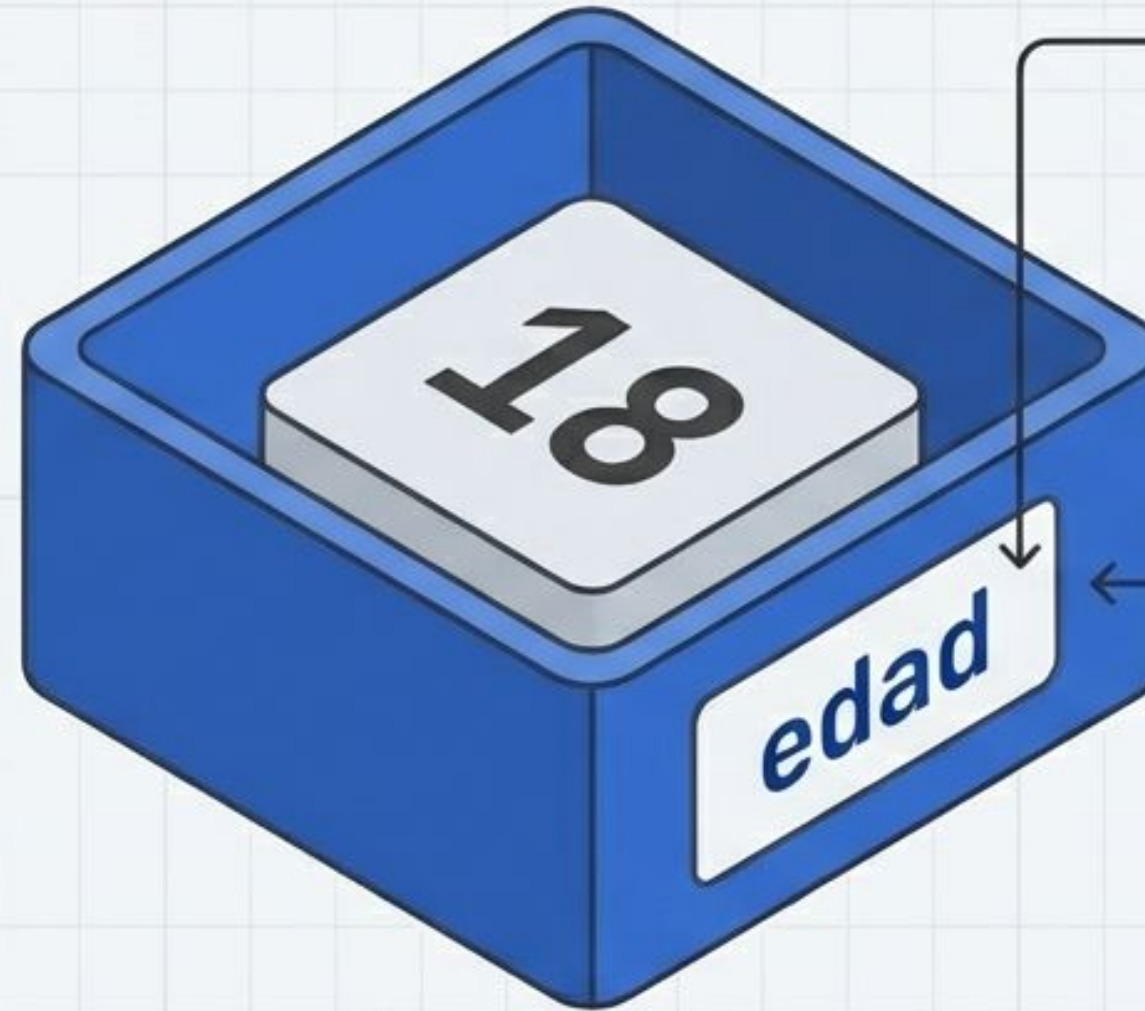
```
*/
```

El objetivo principal del código limpio es la legibilidad.

# ¿Qué es una Variable?

## 1. El Contenedor (Memoria RAM)

Una posición física en la memoria del sistema.



## 2. El Identificador (Nombre)

La etiqueta que le damos para acceder y modificar su contenido (ej. edad).

## 3. El Contenido (Valor)

El dato almacenado dentro, que puede cambiar durante la ejecución (ej. 18).

Nota: Toda variable necesita ser definida con un nombre y un tipo de dato antes de ser utilizada.



# Anatomía de un Identificador y la Convención camelCase

## Reglas de Oro

- ✓ Letra inicial o guión bajo (\_).
- ✓ Solo letras, números o \_ después del primer carácter.
- ✗ Cero espacios en blanco o caracteres especiales.

## Code Review

### camelCase (Correcto)

cuentaBancaria

Entero

nombreAlumno

Cadena

esVocal

Booleano

### Incorrectos (Error)

el\_salario

Error: Espacio

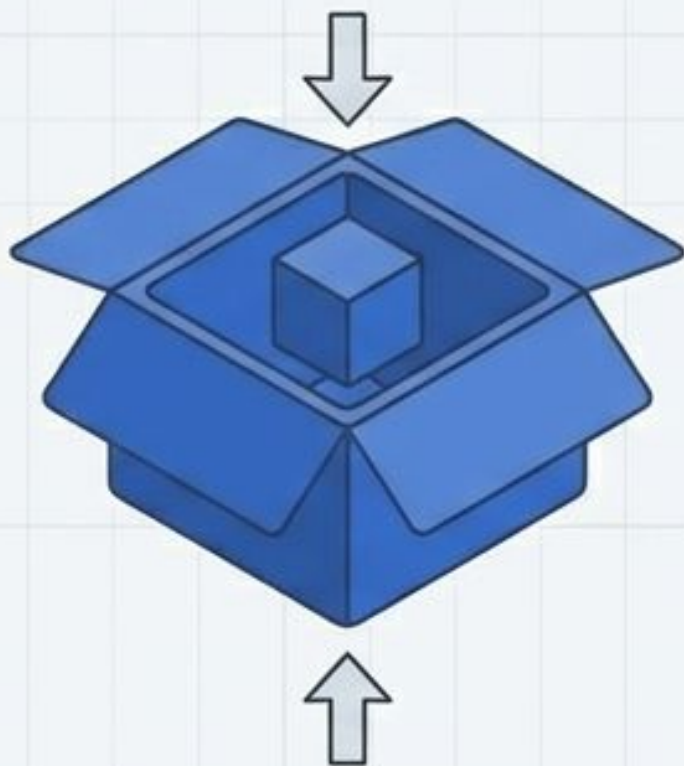
4\_letra

Error: Empieza con número

esVocal?

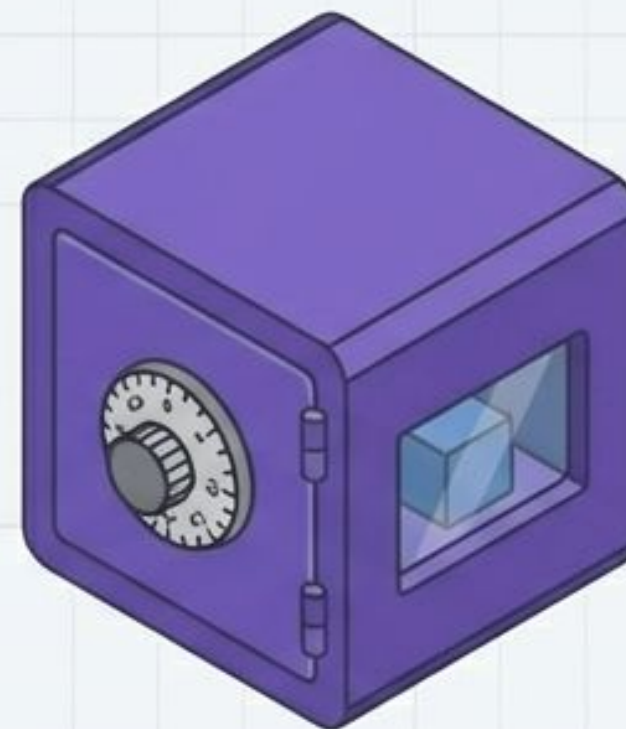
Error: Carácter especial

# Variables vs. Constantes



**Variable**

El valor puede leerse y modificarse múltiples veces durante la ejecución del programa.



**Constante**

El valor se asigna una sola vez y nunca puede cambiar. Es de solo lectura.

```
PI = 3.14159; // Una vez definido, queda bloqueado.
```



# Taxonomía de los Tipos de Datos Primitivos

Tipo de Dato	Definición	Rango/Valores	Ejemplo en Código
Entero	Números sin decimales	-10, 0, 100	Definir num como Entero;
Real	Números con fracción decimal	-10.5, 3.14, 9.9	Definir precio como Real;
Caracter	Un solo símbolo o letra	'a', '\$', '7'	Definir letra como Caracter;
Cadena	Conjunto de caracteres (texto)	"Pepe", "Hola Mundo"	Definir nombre como Cadena;
Lógico	Estado binario (Booleano)	Verdadero, Falso	Definir esNumero como Lógico;

❶ El tipo de dato define el rango de valores posibles y las operaciones que el programa permite realizar con ellos.

# La Asignación: Moviendo Datos a la Memoria



## Regla de Compatibilidad

El valor de la derecha debe ser del mismo tipo de dato (o compatible) que la variable de la izquierda.

```
1 Definir nombre como Cadena;  
2 nombre = "Pepe" * 2;
```

! Operación no definida para tipo Cadena



# Herramientas Numéricas: Operadores Aritméticos



(Suma)

$$4 + 5 = 9$$



(Resta)

$$3 - 2 = 1$$



(Multiplicación)

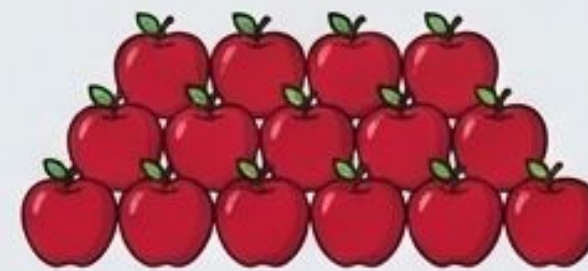
$$4 * 5 = 20$$



(División)

$$19 / 5 = 3.8$$

## El Módulo (%)



`manzanas_sobrantes = 19 % 5; // El resultado es 4, que es el resto de dividir 19 entre 5.`

Exclusivo para variables y constantes de tipo Entero o Real.



# Operadores Relacionales: Máquinas de Comparación

- >** (Mayor que)
- <** (Menor que)
- >=** (Mayor o igual)
- <=** (Menor o igual)
- ==** (Igual a)
- !=** (Distinto de)

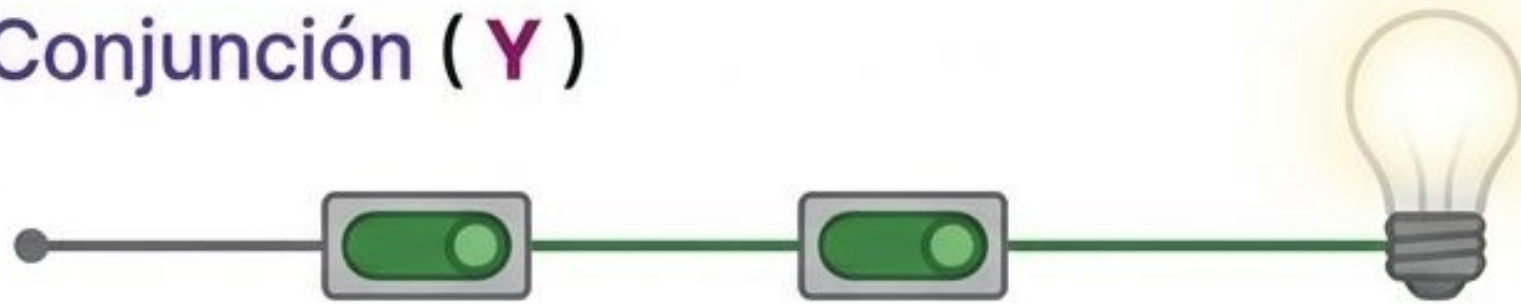
resultado = 4 > 5;



Las comparaciones **nunca** devuelven un número; siempre devuelven un valor Lógico (Booleano).

# Operadores Lógicos: Controlando el Flujo

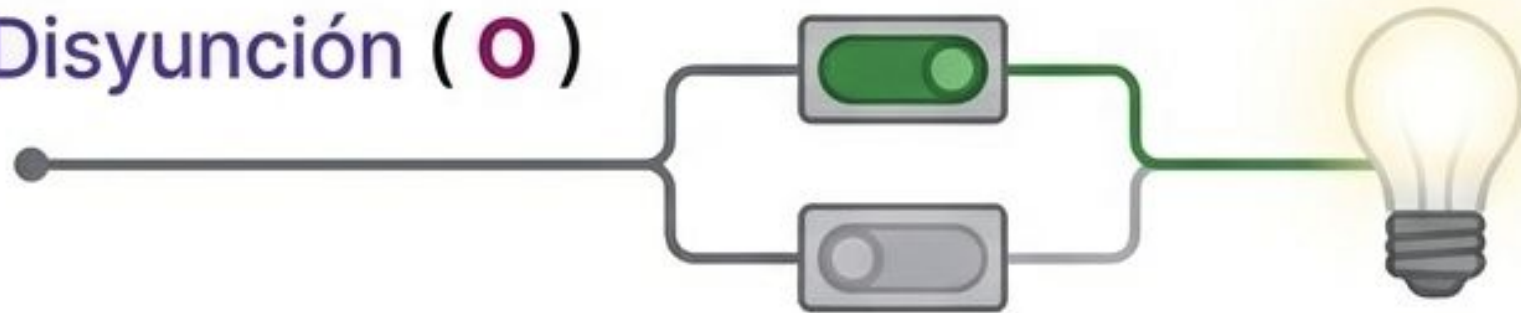
## Conjunción ( Y )



Devuelve **V** solo si ambas condiciones son **V**.

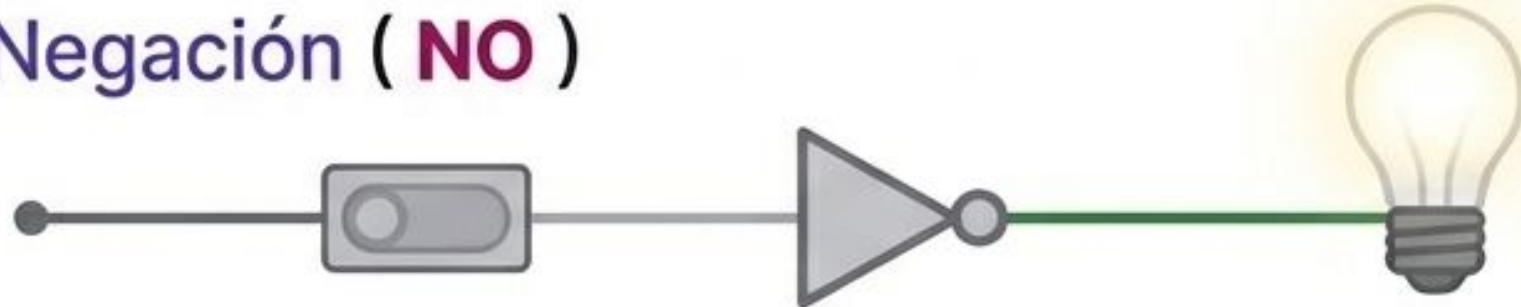
(4 > 3 Y 2 >= 2 -> V)

## Disyunción ( O )



Devuelve **V** si al menos una condición es **V**.

## Negación ( NO )



Invierte el valor lógico.  
(**V** se vuelve **F**).



# La Jerarquía de Operaciones



**¿Empate en el mismo nivel?** El programa evalúa de izquierda a derecha. La asignación (=) es siempre el paso final.

# Evaluando Expresiones Combinadas

$$j = 1 + 3 * 4;$$

Paso 1: Prioridad de Nivel 1

$$j = 1 + 3 * 4;$$

12

Paso 2: Prioridad de Nivel 2

$$j = 1 + 12;$$

13

Paso 3: Asignación Final

$$j = 13;$$



El operador = espera pacientemente hasta que toda la matemática a su derecha se haya resuelto.



# El Poder de los Paréntesis: Forzando la Precedencia

`k = ((12 - 2) * (21 - 11)) / ((1 + 1) * (15 - 10)) + 1;`

`k = ( 10 * 10 ) / ( 2 * 5 ) + 1;`

`k = 100 / 10 + 1;`

`k = 10 + 1;`

`k = 11;`

Nota: Los paréntesis anidados siempre se evalúan de adentro hacia afuera y no afectan negativamente el rendimiento del programa.

# Recordatorio

## Declarar (El Contenedor)

- Concepto: Toda variable/constante necesita memoria y tipo.
- Sintaxis:  
**Definir** [nombre]  
**como** [Tipo];
- Tipos Primitivos: Entero, Real, Caracter, Cadena, Lógico.

## Nombrar (La Etiqueta)

- Regla: Sin espacios, no empezar con números.
- Estilo: Usa camelCase para legibilidad.
- Ejemplo:  
"salarioMinimo",  
"esValido"

## Operar (La Acción)

- Aritmética: **+**, **-**, **\***, **/**, **%**
- Lógica: **Y**, **O**, **NO** (Genera Booleanos)
- Precedencia:  
Matemáticas >  
Comparación > Lógica >  
Asignación.



# La Lógica al Servicio del Problema

```
codigoLimpio = Verdadero;
```

Dominar las **variables**, los **tipos de datos** y los operadores no se trata solo de hacer que la máquina compile. Se trata de construir estructuras lógicas claras, precisas y predecibles para resolver problemas del mundo real.