

Presentación *resumen* del libro:

# "EMPEZAR DE CERO A PROGRAMAR EN **lenguaje C**"

Autor: Carlos Javier Pes Rivas (correo@carlospes.com)

## **Capítulo 7**

### **OPERADORES Y EXPRESIONES**



# OBJETIVOS

- Conocer los tipos de **operadores** y **expresiones** que existen en programación.
- Saber **evaluar las expresiones** que aparecen en un programa.
  - En programación, los operadores realizan operaciones sobre los datos que utiliza un programa. Por su parte, las operaciones que se pueden realizar sobre cada dato, dependen del tipo de dato en cuestión, pero, ¿qué es una operación? y ¿qué tipos de operaciones existen?

# CONTENIDO

**7.1 INTRODUCCIÓN**

**7.2 EXPRESIONES ARITMÉTICAS**

**7.3 EXPRESIONES LÓGICAS**

**7.4 EXPRESIONES DE CARÁCTER**

**7.5 EXPRESIONES DE CADENA**

**7.6 PRIORIDAD DE LOS OPERADORES ARITMÉTICOS,  
RELACIONALES, LÓGICOS Y DE CADENA**

## 7.1 INTRODUCCIÓN (1/4)

- El tipo de un dato determina las operaciones que se pueden realizar con él.
- Por ejemplo, con los datos de tipo entero se pueden realizar operaciones aritméticas (suma, resta, multiplicación,...).

111 + 6 (operación suma) (dos operandos y un operador)

19 - 72 (operación resta) (dos operandos y un operador)

24 \* 3 (operación multiplicación) (dos operandos y un operador)

111 + 6 - 8 (tres operandos y dos operadores)

-( ( +19 ) + 72 ) (dos operandos y tres operadores)

-( -72 ) (un operando y dos operadores)

( 3 \* 7 ) + ( 2 - 1 ) (cuatro operandos y tres operadores)

## 7.1 INTRODUCCIÓN (2/4)

- **Un operador:** indica el tipo de operación a realizar sobre los operandos (datos) que actúa.
- Los **operandos** pueden ser:
  - **Constantes** (expresadas por su valor o con su nombre).
  - **Variables.**
  - **Llamadas a funciones.**
  - **Elementos de formaciones (arrays).**
- Cuando se combinan uno o más operadores con uno o más operandos se obtiene una **expresión**.
- **Una expresión:** es una secuencia de operandos y operadores escrita bajo unas reglas de sintaxis.

## 7.1 INTRODUCCIÓN (3/4)

- **EJEMPLO:** Dadas las siguientes declaraciones de constantes y variables en pseudocódigo.

```
PI = 3.141592
entero numero = 2
real radio_circulo = 3.2
```

Algunos ejemplos de expresiones son:

```
2 * PI * radio_circulo
( PI * PI )
numero * 5
```

- Un operador siempre forma parte de una expresión, en la cual, el operador siempre actúa sobre al menos un operando.
- Por el contrario, un operando sí puede aparecer solo en una expresión.

## 7.1 INTRODUCCIÓN (4/4)

- De la evaluación de una expresión siempre se obtiene un valor.
- Dicho valor puede ser de tipo: entero, real, lógico, carácter o cadena.
- Por consiguiente, una expresión puede ser:
  - Aritmética (devuelve un número entero o real).
  - Lógica (devuelve un valor lógico: **verdadero** o **falso**).
  - De carácter (devuelve un carácter representable por el ordenador).
  - De cadena (devuelve una cadena).
- Dependiendo del tipo de expresión, pueden participar unos operadores u otros.

## 7.2 EXPRESIONES ARITMÉTICAS (1/3)

- En las **expresiones aritméticas** se pueden utilizar los siguientes operadores aritméticos:

<i>Operadores aritméticos en pseudocódigo:</i>	
+	Suma
-	Resta
*	Multiplicación
**	Potencia (también se utiliza el carácter <i>flecha arriba</i> ↑ o el carácter <i>acento circunflejo</i> ^)
/	División real
div	División entera (también se utiliza el carácter <i>barra invertida</i> \)
mod	Módulo (resto de la división entera)
<hr/>	
+	Signo más
-	Signo menos

## 7.2 EXPRESIONES ARITMÉTICAS (2/3)

- **Prioridad** de los operadores aritméticos:

<i>Prioridad de los operadores aritméticos (de mayor a menor) en pseudocódigo:</i>	
+ -	Signos más y menos
**	Potencia
* / div mod	Multiplicación, división real, división entera y módulo
+ -	Suma y resta

## 7.2 EXPRESIONES ARITMÉTICAS (3/3)

- **EJEMPLOS:** De las siguientes expresiones:

`-3 * 6 mod 4`

`3 ** 3 - 1`

`-3 * ( 6 mod 4 )`

`3 ** ( 3 - 1 )`

Se obtienen los valores:

-2 (actúan en orden los operadores: signo menos (-), (\*) y (mod))

26 (actúan en orden los operadores: (\*\*)) y resta (-))

-6 (actúan en orden los operadores: (mod), signo menos (-) y (\*\*))

9 (actúan en orden los operadores: resta (-) y (\*\*))

## 7.3 EXPRESIONES LÓGICAS (1/9)

- De la evaluación de una **expresión lógica** siempre se obtiene un valor de tipo lógico (**verdadero** o **falso**).
- En las expresiones lógicas se pueden utilizar dos tipos de operadores:
  - Relacionales
  - Lógicos

## 7.3 EXPRESIONES LÓGICAS (2/9)

- **Operadores relacionales:**

<i>Operadores relacionales en pseudocódigo:</i>	
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
<hr/>	
=	Igual que
<>	Distinto que

- Para escribir una expresión relacional (lógica) se utiliza la sintaxis:

<expresión\_1> <operador\_de\_relación> <expresión\_2>

## 7.3 EXPRESIONES LÓGICAS (3/9)

- **EJEMPLOS:** De las siguientes expresiones:

`22 > 13` (comparación de dos expresiones aritméticas)

`22.5 < 3.44` (comparación de dos expresiones aritméticas)

`verdadero = falso` (comparación de dos expresiones lógicas)

`'c' > 'f'` (comparación de dos expresiones de carácter)

`"coche" = "Coche"` (comparación de dos expresiones de cadena)

Se obtienen los valores:

**verdadero** (22 es mayor que 13)

**falso** (22.5 no es menor que 3.44)

**falso** (`verdadero` no es igual que `falso`)

**falso** (`'c'` no es mayor que `'f'`)

**falso** (`"coche"` no es igual que `"Coche"`)

## 7.3 EXPRESIONES LÓGICAS (4/9)

- **Operadores lógicos:**

<i>Operadores lógicos en pseudocódigo:</i>	
$\text{y}$	Conjunción
$\text{o}$	Disyunción
$\text{no}$	Negación

- El operador **conjunción** ( $\text{y}$ ) y el operador **disyunción** ( $\text{o}$ ) siempre actúan sobre dos operandos, mientras que, el operador **negación** ( $\text{no}$ ) sólo actúa sobre un operando, o dicho de otra forma, es un operador monario.

## 7.3 EXPRESIONES LÓGICAS (5/9)

- **Tabla de verdad del operador (y):**

<expresión_1>	<expresión_2>	<expresión_1> y <expresión_2>
verdadero	verdadero	verdadero
verdadero	falso	falso
falso	verdadero	falso
falso	falso	falso

- **EJEMPLO:** De evaluar:

$9 > 3$  y  $8 > 9$

se obtiene:

**falso** ( $9 > 3$  es verdadero y  $8 > 9$  es falso)

## 7.3 EXPRESIONES LÓGICAS (6/9)

- **Tabla de verdad del operador (o):**

<code>&lt;expresión_1&gt;</code>	<code>&lt;expresión_2&gt;</code>	<code>&lt;expresión_1&gt; o &lt;expresión_2&gt;</code>
verdadero	verdadero	verdadero
verdadero	falso	verdadero
falso	verdadero	verdadero
falso	falso	falso

- **EJEMPLO:** De evaluar:

`9 > 3 y 8 > 9`

se obtiene:

**verdadero** (`9 > 3` es verdadero y `8 > 9` es falso)

## 7.3 EXPRESIONES LÓGICAS (7/9)

- **Tabla de verdad del operador negación (no):**

<code>&lt;expresión&gt;</code>	<code>no &lt;expresión&gt;</code>
<b>verdadero</b>	<b>falso</b>
<b>falso</b>	<b>verdadero</b>

- **EJEMPLO:** De evaluar:

`no ( 9 > 3 )`

se obtiene:

**falso** (9 > 3 es verdadero)

## 7.3 EXPRESIONES LÓGICAS (8/9)

- **Prioridad** de los operadores relacionales y lógicos:

<i>Prioridad de los operadores relacionales y lógicos (de mayor a menor) en pseudocódigo:</i>	
no	Negación
< <= > >=	Menor que, menor o igual que, mayor que, mayor o igual que
= <>	Igual que, distinto que
y	Conjunción
o	Disyunción

## 7.3 EXPRESIONES LÓGICAS (9/9)

- **EJEMPLOS:** De las siguientes expresiones:

`3 > 1 o 4 < 1 y 4 <= 2`

`( 3 > 1 o 4 < 1 ) y 4 <= 2`

`falso = verdadero <= verdadero`

`( falso = verdadero ) <= verdadero`

Se obtienen los valores:

**verdadero** (actúan en orden los operadores: (>), (<), (<=), (y) y (o))

**falso** (actúan en orden los operadores: (>), (<), (o), (<=) e (y))

**falso** (actúan en orden los operadores: (<=) y (=))

**verdadero** (actúan en orden los operadores: (=) y (<=))

## 7.4 EXPRESIONES DE CARÁCTER

- Aunque no existe ningún operador de caracteres, sí que existen expresiones de carácter.
- De la evaluación de una **expresión de carácter** siempre se obtiene un valor de tipo carácter.
- **EJEMPLO:** Dadas las siguientes declaraciones de constantes y variables en pseudocódigo.

```
CONSONANTE = 'S'  
caracter letra = 'X'  
caracter opcion = '3'
```

Algunas expresiones de carácter son:

```
opcion  
letra  
CONSONANTE  
'a'
```

## 7.5 EXPRESIONES DE CADENA

- Sólo existe un operador de cadena:

<p><i>Operador de cadena en pseudocódigo:</i></p> <hr/> <p>+ Concatenación</p>
--

- **EJEMPLO:** Dadas las siguientes declaraciones de constantes y variables en pseudocódigo.

```
OLIMPIADA = "Atenas 2004"
```

```
cadena nombre = "Pedro", apellido = "Cosín"
```

Algunas expresiones de cadena son:

```
OLIMPIADA
```

```
nombre + " " + apellido
```

```
nombre + " fue a las Olimpiadas de " + OLIMPIADA
```

## 7.6 PRIORIDAD DE LOS OPERADORES ARITMÉTICOS, RELACIONALES, LÓGICOS Y DE CADENA (1/2)

*Prioridad de los operadores aritméticos, relacionales, lógicos y de cadena (de mayor a menor) en pseudocódigo:*

+ - no	Signo más, signo menos y negación
**	Potencia
* / div mod	Multiplicación, división real, división entera y módulo
+ -	Suma y resta
+	Concatenación
< <= > >=	Menor que, menor o igual que, mayor que, mayor o igual que
= <>	Igual que y distinto que
Y	Conjunción
o	Disyunción

## 7.6 PRIORIDAD DE LOS OPERADORES ARITMÉTICOS, RELACIONALES, LÓGICOS Y DE CADENA (2/2)

- **EJEMPLOS:** De las siguientes expresiones:

`5 * 4 > 5 + 4 o falso y "ab" < "aa"`

`( 5 * 4 > 5 + 4 o falso ) y 'f' < 'b'`

`no verdadero < falso`

`no ( verdadero < falso )`

Se obtienen los valores:

**verdadero** (actúan en orden los operadores: (\*), suma (+), (>), (<), (y) y (o))

**falso** (actúan en orden los operadores: (\*), suma (+), (>), (o), (<) e (y))

**falso** (actúan en orden los operadores: (no) y (<))

**verdadero** (actúan en orden los operadores: (<) y (no))

# EJERCICIOS RECOMENDADOS

- **Resueltos:** 1 y 2.
- **Propuestos:** 1 y 2.

# GRACIAS POR SU ATENCIÓN

Para más información, puede visitar la web del autor:

**<http://www.carlospes.com>**

