

Programación Estructurada en ANSI C

Sesión 2A



Rafael Menéndez de Llano Rozas

DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA

Este material se publica bajo licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



Índice

1. Introducción.

1. Elementos lexicográficos y estructura.

1. Datos escalares, expresiones y entrada/salida básica.

2. Selección.

2. Iteración.

3. Funciones, punteros y estructuración.

4. Datos estructurados.

5. Otros aspectos.

2. Expresiones booleanas

- Para seleccionar que sentencias se realizan en un programa antes se debe establecer una condición (pregunta) que se evaluará a verdadero o falso (valores booleanos).

- El pseudocódigo sería:

SI (se cumple está condición) ENTONCES

Haz estas sentencias;

[SINO

Haz estas otras;]

IF condicion THEN

sentencias(s);

ELSE

otras sentencia(s);

- En C estas condiciones son expresiones (booleanas) de **tipo entero** (en C99 hay tipo bool). **Cero para falso** (FALSE) y uno (o **distinto de cero**) para verdad (TRUE).
- Todas la expresiones en C son de tipo entero, lo cual las hace susceptibles de ser condiciones, incluidas las variables (las más sencillas). Además se pueden combinar con **operadores** de dos tipos:
 - **Relacionales:** == != >= <= > <
 - **Lógicos:** AND con && OR con || NOT con !
 - AND dará Falso si algún operando es Falso, OR dará Verdad si algún operando es Verdad.

2. Expresiones booleanas

- Los operadores relacionales son binarios por definición.
- Los operadores lógicos, salvo el NOT, también.
- La **precedencia** aparece en la tabla de mayor a menor (los binarios son menores que los aritméticos):

NOT	!
Relacionales	< <= >= >
Relacionales	== !=
AND	&&
OR	
asignación	=

Precedencia
de mayor a
menor



- Para evaluar una condición se siguen las normas de las expresiones en general, “salvo” que el orden está garantizado que es de **izquierda a derecha** y en **cortocircuito**:

`(ch = getchar()) != EOF && ch != '\n'`

`numero != 0 && 12 / numero == 2`

2. Selección

- La selección se implementa con la sentencia if - else

- ◆ if (expresión)
ejecuta sentencia 1;
- ◆ if (expresión)
ejecuta sentencia 1;
else
ejecuta sentencia 2;

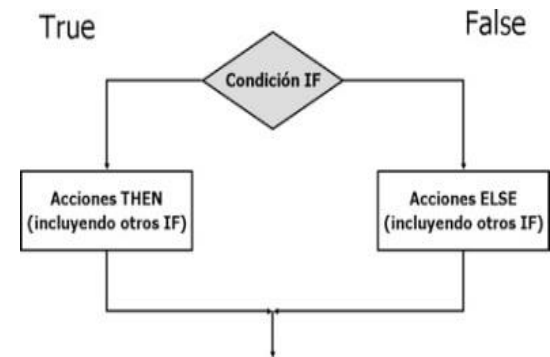
- También se pueden ejecutar bloques { }.

- Anidamiento:

```
if ( numero > 6 )  
    if ( numero < 12 )  
        printf("Caliente !\n");  
else // a cual pertenece ?  
    printf("Lo siento has perdido!\n");
```

- En C también existe el operador condicional “?” (**ternario**):

```
x = (y < 0) ? -y : y;    <=>    if (y < 0) x = -y; else x = y;
```



2. Comparación en Selección

Java	C
<pre>if (exp_booleana) { instrucciones; }</pre>	<pre>if (exp_entera) { instrucciones; }</pre>
<pre>if (exp_booleana) { instrucciones; } else { instrucciones; }</pre>	<pre>if (exp_entera) { instrucciones; } else { instrucciones; }</pre>

Ejercicio 6: Expresiones booleanas

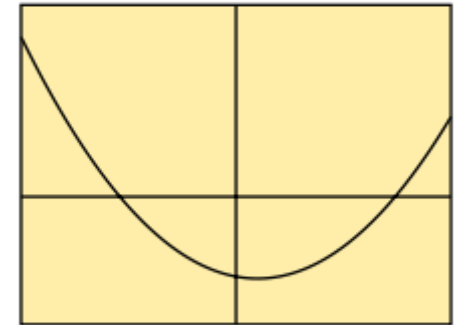
1. Bájate el programa 7.c y guárdalo en tu directorio.
2. Ábrelo con un editor y examínalo y comprende las operaciones que hay.
3. Piensa como aparecerán en pantalla lo indicado en los printf.
4. Observa como las condiciones (expresiones booleanas) se evalúan siempre de izquierda a derecha.
5. Observa como las condiciones (expresiones booleanas) se evalúan siempre en cortocircuito.

Ejercicio 7: Sentencias de selección

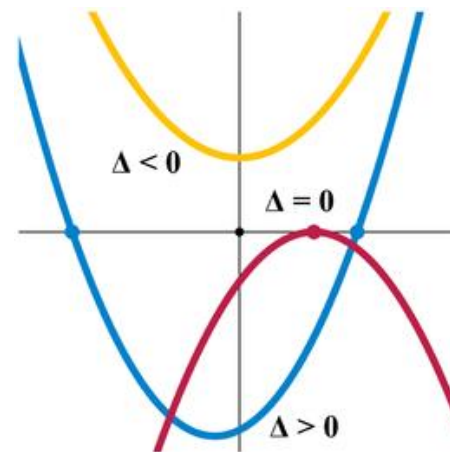
1. Bájate el programa 8.c y guárdalo en tu directorio.
2. Ábrelo con un editor y examínalo y comprende las operaciones que hay.
3. Se trata de resolver una ecuación de segundo grado.
4. Se usa la librería m.

Ecuación cuadrática

$$ax^2 + bx + c = 0$$



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



Ejercicio 8: Sentencias de selección

1. Bájate el programa 9.c y guárdalo en tu directorio.
2. Ábrelo con un editor y examínalo y comprende las operaciones que hay.
3. Practica con las sentencias de selección
if - else if - else.
4. Explica que hace el operador condicional.

2. Selección múltiple

- Cuando hay muchas alternativas queda más claro utilizar la sentencia switch-case:

```
switch ( expresion escalar discreta )
{
    case primero :      sentencias;
                      break;          /* opcional */
    case segundo:      sentencias;
                      break;          /* opcional */
    case ultimo  :      sentencias;
                      break;          /* opcional */
    default :         sentencias;
                      break;          /* opcional */
}
```

- continue: para salir de una iteración (bucles).
- break: en bucles se sale del todo.
- exit(): para terminar el programa.
- goto: mejor no usar.

2. Comparación de selección

Java	C
<pre>switch (exp_discreta) { case valor1 : instrucciones; break; case valor2 : case valor3 : instrucciones; break; default : instrucciones; }</pre>	<pre>switch (exp_entera) { case valor1 : instrucciones; break; case valor2 : case valor3 : instrucciones; break; default : instrucciones; }</pre>

Ejercicio 9: Sentencias de selección

1. Bájate el programa 10.c y guárdalo en tu directorio.
2. Ábrelo con un editor y examínalo y comprende las operaciones que hay.
3. Practica con las sentencias de selección múltiple borrando algún break que no sea necesario.