

Construcciones condicionales

Una de las construcciones importantes que pueden especificarse en un programa es el hecho de realizar diferentes tareas en función de ciertas condiciones. Esto es, ejecutar una parte del código u otra, condicionalmente. Para ello será necesario especificar dichas condiciones y disponer de un mecanismo para indicar qué acciones tomar dependiendo de cómo se evalúe una determinada condición en un momento dado de la ejecución del programa.

Antes de empezar, un recordatorio. El lenguaje C no dispone de valores booleanos o lógicos, que podrían usarse en la evaluación de condiciones. En su defecto, C "simula" los valores falso y cierto, como el valor numérico cero, y cualquier valor no cero (incluyendo negativos), respectivamente.

Así pues, en este capítulo veremos las distintas maneras que C ofrece para controlar el flujo de ejecución de un programa de forma condicional, que son:

- La sentencia `if`.
- La sentencia `if-else`.
- La sentencia `while`.
- La sentencia `do-while`.
- La sentencia `for`.
- * La sentencia *`switch`*.

Construcción *if*

La construcción *if* es similar a la existente en otros lenguajes de programación, aunque en C posee ciertas peculiaridades. El formato general de esta construcción para decidir si una determinada sentencia debe ejecutarse o no (alternativa simple) es el siguiente:

```
if (condición)
sentencia;
```

```
if (condición)
{
sentencia 1;
.....
sentencia N;
}
```

La construcción *if* puede escribirse también de forma más general para controlar la ejecución de un grupo de sentencias, de la siguiente manera:

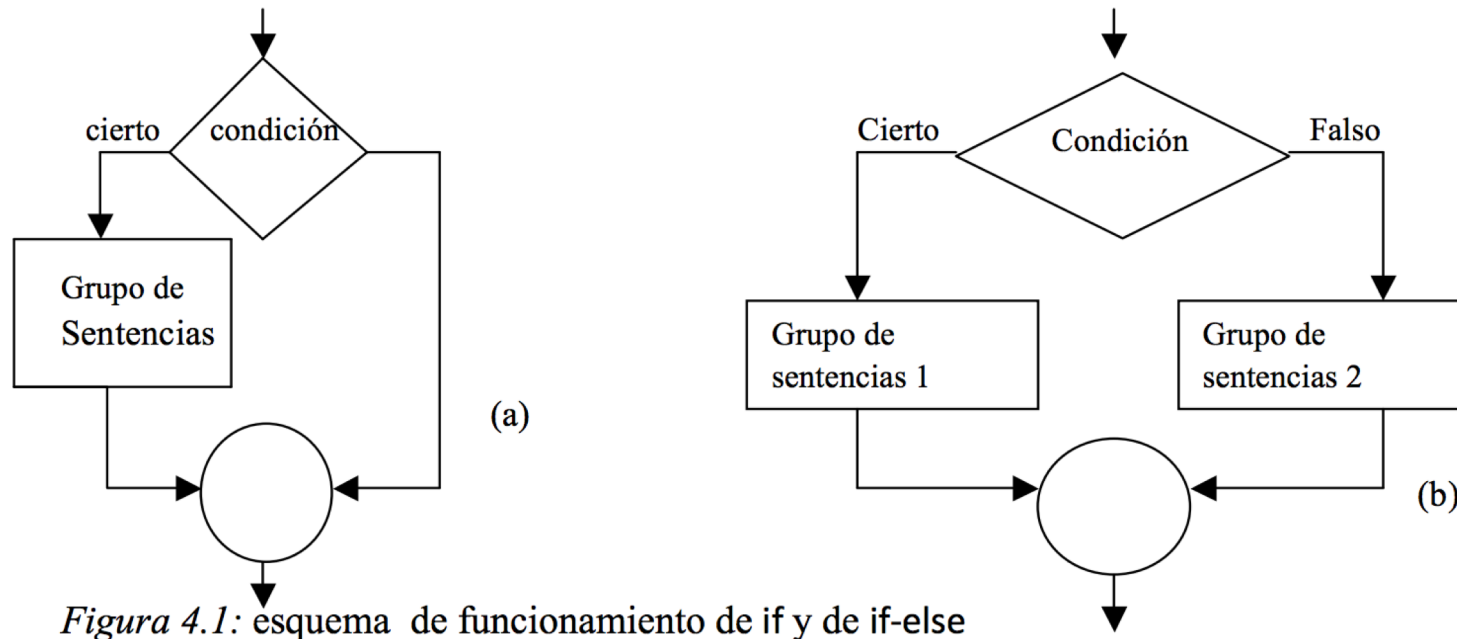


Figura 4.1: esquema de funcionamiento de if y de if-else

```
If (condición)
{
  Sentencia_1;
  Sentencia_2;
  . . .
  Sentencia_N;
}
```

El funcionamiento de la *construcción if* es muy simple. En primer lugar se evalúa la condición, que no es otra cosa que una expresión de tipo entero. A continuación, si la expresión se ha evaluado como cierta, se ejecuta la sentencia o grupo de sentencias. En caso contrario la ejecución del programa continua por la siguiente sentencia en orden secuencial (ver Fig. 4.1 (a)).