

El lenguaje de programación C

- Control de flujo y bucles -

5

Isidro González Caballero
(gonzalezisidro@uniovi.es)

Introducción a la Física
Computacional

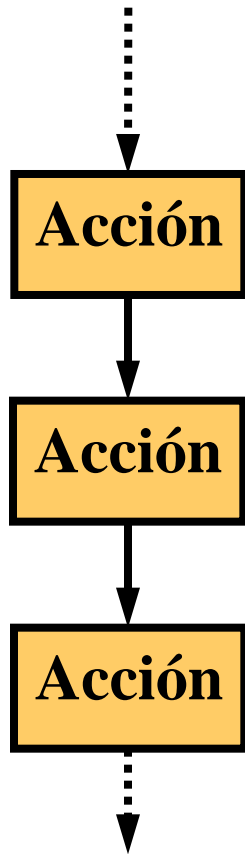
Curso 2010 - 2011



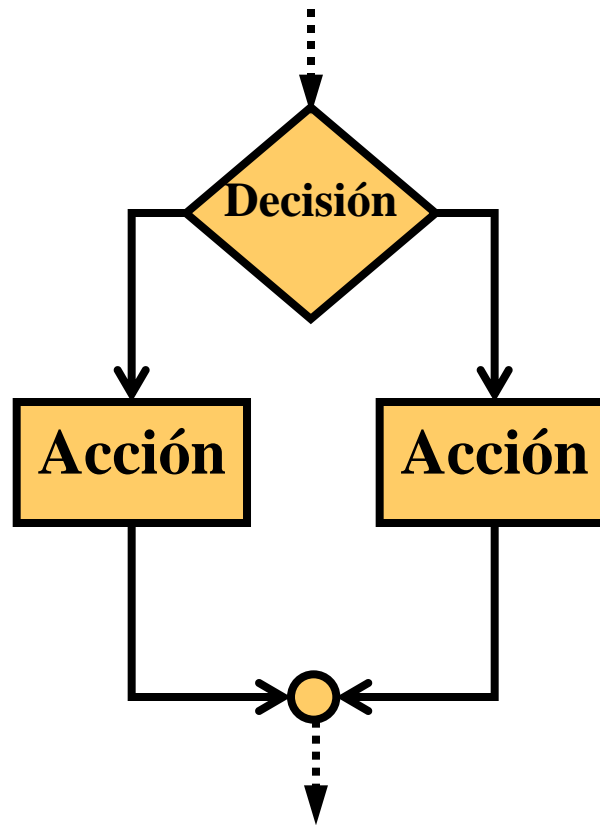


Tres clases de bloques de programación

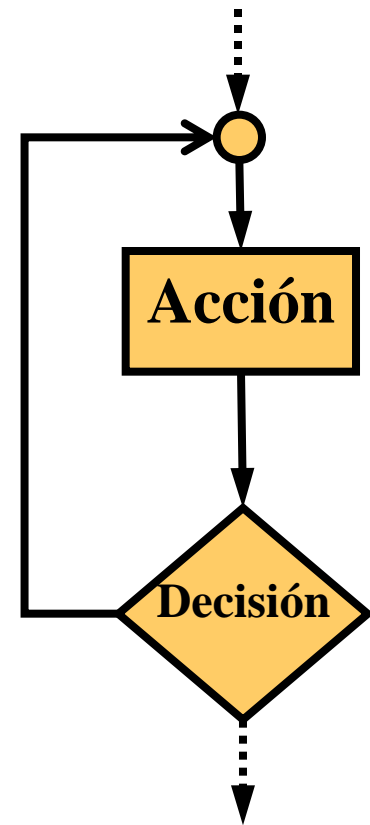
Secuencial



**Rama
(o selección)**

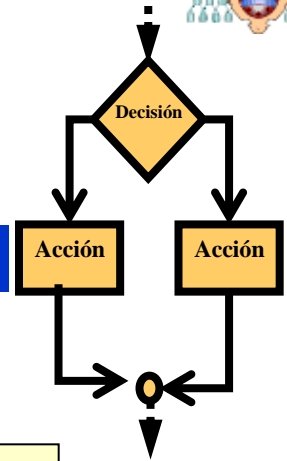


**Bucle
(o iteración)**





Control de flujo: if-else



- Ejecución **condicional**
- Sintaxis simple:

```
if (expresión lógica)
    sentencia1;
```

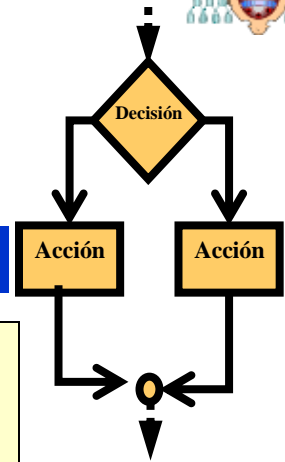
- Si la expresión lógica es cierta se ejecuta sentencia1

- Sintaxis completa:

```
if (expresión lógica)
    sentencia1;
else
    sentencia2;
```

- Si la expresión lógica es falsa se ejecuta la sentencia2
- Si hay más de una sentencia a ejecutar por caso se meten entre llaves: { }
- Se pueden encadenar en caso de querer comprobar múltiples opciones: if - else if - else if - ... - else

if-else ejemplo

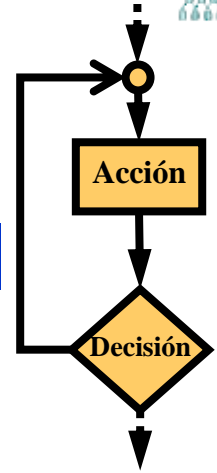


```
int i; /*Un numero entero*/
printf("Introduce un número entero:\n");
scanf("%d", i);

printf("El número introducido es ");
if (i < 0)
    printf("negativo\n");
else if (i > 0)
    printf("positivo\n");
else {
    printf("ni positivo\n");
    printf("ni negativo\n");
}
```



Blucles: for



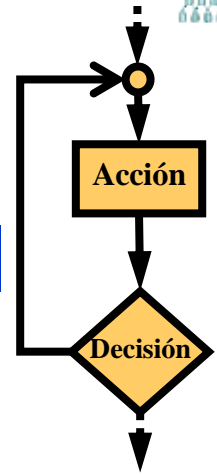
- Iteraciones
- Sintaxis:

```
for (inicialización; parada; acción)  
    sentencia;
```

- La *inicialización* (de un contador) solo se ejecuta la primera vez que entra en el bloque
- La *parada* es una expresión lógica
 - Mientras sea cierta se sigue iterando
 - Cuando sea falsa se sale del bucle
- La *acción* (normalmente el incremento de un contador) se ejecuta al comienzo de cada iteración



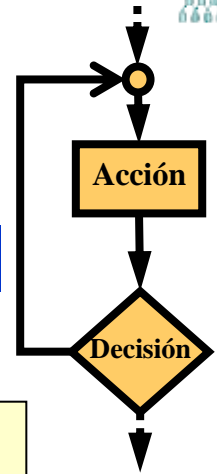
Blucles: for - Ejemplo



```
printf("Tabla de multiplicar del 3...\n");  
  
int i;  
for (i = 1; i <= 10; i++) {  
    printf("3 * %d = %d\n", i, 3 * i);  
}
```



Bucles: while y do-while



- Sintaxis:

```
while (expresion)  
    sentencia;
```

- Sólo si la *expresión* es cierta se ejecuta una nueva iteración
- La comprobación se hace al inicio

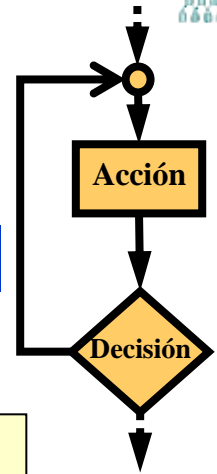
- Sintaxis

```
do  
    sentencia;  
while (expresion)
```

- Sólo si la *expresion* es cierta se ejecuta una nueva iteración
- La comprobación se hace al final: La sentencia se ejecuta al menos una vez!



Bucles: while y do-while - Ejemplo



while

```
printf("Tabla de multiplicar del 3...\n");  
int i = 1;  
while (i <= 10) {  
    printf("3 * %d = %d\n", i, 3 * i);  
    i++;  
}
```

do - while

```
printf("Tabla de multiplicar del 3...\n");  
int i = 1  
do {  
    printf("3 * %d = %d\n", i, 3 * i);  
    i++;  
} while (i <= 10)
```