

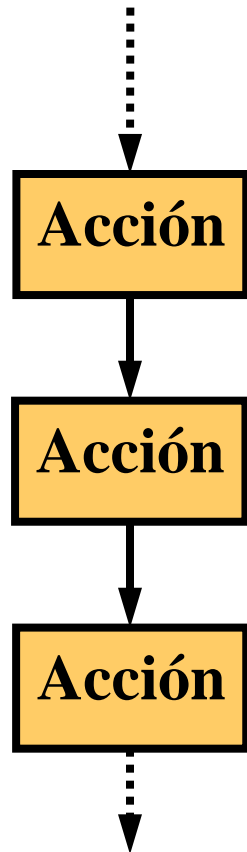
El lenguaje de programación C

- Control de flujo y bucles -

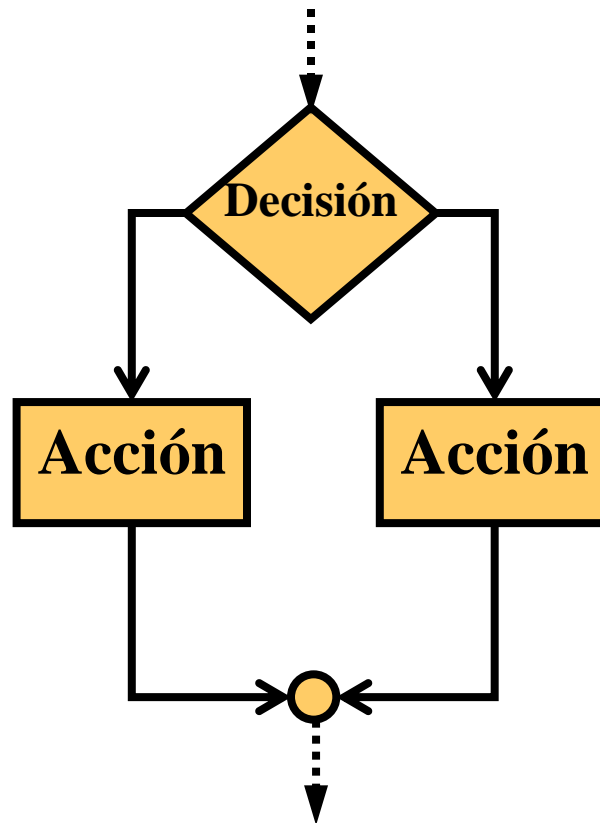


Tres clases de bloques de programación

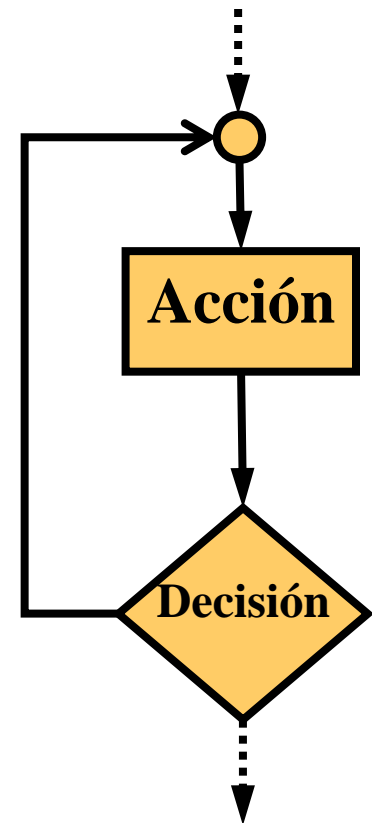
Secuencial



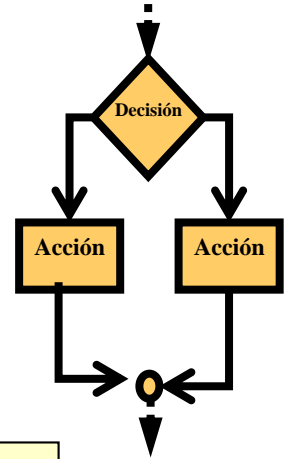
**Rama
(o selección)**



**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:

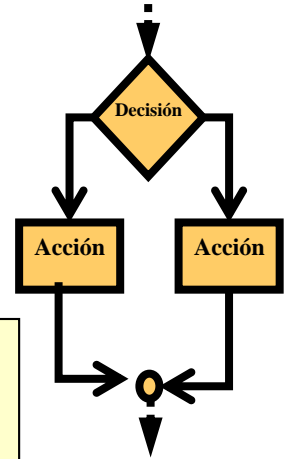
```
if (expresión lógica)  
    sentencia1;  
else  
    sentencia2;
```

- 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");
}
```

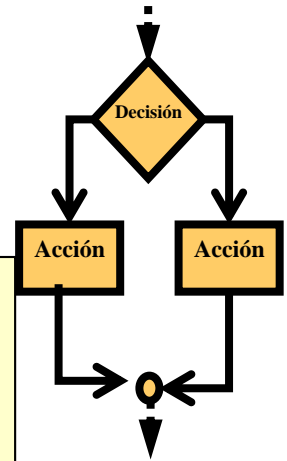


if-else anidados ejemplo

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

printf("El número introducido es ");
if (i < 0)
    if (i < -5)
        printf("menor que -5\n");
    else
        printf("negativo mayor que -5\n");

else if (i > 0)
    printf("positivo\n");
else {
    printf("ni positivo\n");
    printf("ni negativo\n");
}
```



if-else una línea o más

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

if (i < 0)
    printf("negativo ");

else {
    printf("ni positivo\n");
    printf("ni negativo\n");
}
```

if-else

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

printf("El número introducido es ");
if (i < 0)
    if (i < -5)
        printf("menor que -5\n");
    else if (i > 0)
        printf("positivo\n");
else {
    printf("ni positivo\n");
    printf("ni negativo\n");
}
```

if-else anidados ejemplo

- Determinar si un año es bisiesto.
- Un año es bisiesto si cumple alguna de las siguientes:
 - es múltiplo de 4 pero no es múltiplo de 100.
 - múltiplo de 400.
- No son bisiestos: 1900, 1999, 2003
- Sí son bisiestos: 1976, 2000, 2004

if-else anidados ejemplo

```
#include <conio.h>
#include <stdio.h>

int main()
{
    int anio;

    printf( "\n  Introduzca un año: ", 164 );
    scanf( "%d", &anio );

    if ( anio % 4 == 0 )
        if (anio % 100 != 0 )
            printf( "\n    ES BISIESTO" );
        else
            printf( "\n    NO ES BISIESTO" );

    else if (anio % 400 == 0 )
        printf( "\n    ES BISIESTO" );
    else
        printf( "\n    NO ES BISIESTO" );

    getch(); /* Pausa */

    return 0;
}
```

if-else anidados ejemplo

```
#include <conio.h>
#include <stdio.h>

int main()
{
    int anio;

    printf( "\n    Introduzca un año: ", 164 );
    scanf( "%d", &anio );

    if ( anio % 4 == 0 && anio % 100 != 0 || anio % 400 == 0 )
        printf( "\n    ES BISIESTO" );
    else
        printf( "\n    NO ES BISIESTO" );

    getch(); /* Pausa */

    return 0;
}
```

Sentencia de selección SWITCH

```
switch(variable)
{
    case valor_1:
        //acciones
        break;
    case valor_2:
        //acciones
        break;
        .
        .
        .
    case valor_n:
        //acciones
        break;
    defalut:
        //acciones
        break;
}
```

```
void main()
{
    int opcion;
    printf("Ingrese un numero entero: ");
    scanf("%d",&opcion);
    switch(opcion)
    {
        case 1:
            printf("Ha ingresado el numero uno\n");
            break;

        case 2:
            printf("Ha ingresado el numero dos\n");
            break;

        case 3:
            printf("Ha ingresado el numero tres\n");
            break;

        case 4:
            printf("Ha ingresado el numero cuatro\n");
            break;

        case 5:
            printf("Ha ingresado el numero cinco\n");
            break;

        .
        .

        case 9:
            printf("Ha ingresado el numero nueve\n");
            break;

        defalut:
            printf("Ha ingresado un numero no valido\n");
            break;

    }
}
```

switch - if

```
int a;
switch(a)
{
case 0:
    //operaciones a realizar en caso de que a valga 0;
    break;
case 1:
    //mismo proceso
    break;
...
case n:
    //n tiene que ser una constante, o numérica {0,1,2 ....} o una definida #define n 2
    break;
default:
    //en caso de que no se de ninguna de las anteriores hacer...
}
}
```

Switch - if

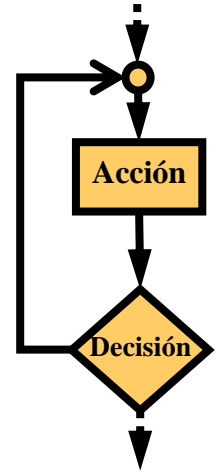
```
int a;  
if(a==0)  
    //operaciones para 0  
else if(a==1)  
    //operaciones para 1  
...  
else if(a==n)  
    //operaciones para n  
else  
    //cualquier otro caso
```

Blucles: for

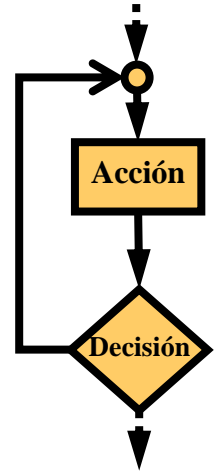
- Iteraciones
- Sintaxis:

```
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

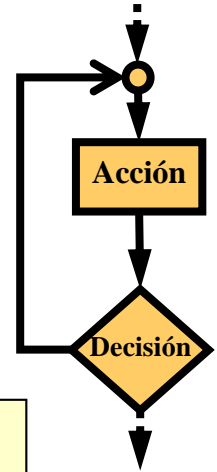


Blucles: for - Ejemplo



```
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;
```

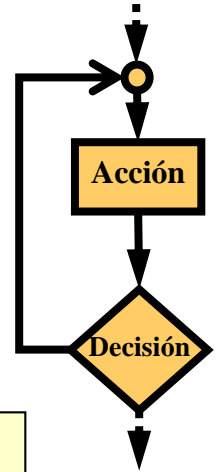
- Sólo si la *expresión* es cierta se ejecuta una nueva iteración
- La comprobación se hace al inicio

- 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)
```