

Tema 3: Sentencias de Control

Oscar Perpiñán Lamigueiro - David Álvarez

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

Introducción

- Sin sentencias de control los programas se ejecutan de manera secuencial

Sentencias de Control

Sentencias Condicionales ejecutan unas secuencias u otras según el cumplimiento de unas condiciones.

- `if, if - else`
- `switch - case`

Sentencias Repetitivas repiten un conjunto de sentencias en unas determinadas condiciones.

- `for`
- `while, do - while`

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

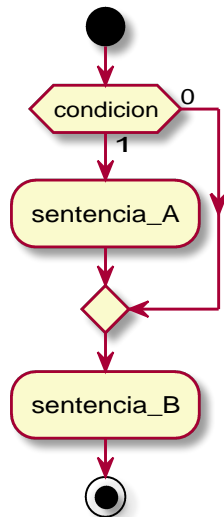
if

- Si se cumple condicion (resultado **diferente de 0**) se ejecuta la sentencia_A.
- **Siempre** se ejecuta la sentencia_B.

```
if (condicion)
    sentencia_A;
sentencia_B;
```

- Usar **llaves** para más de una sentencia.

```
if (condicion)
{
    sentencia_A1;
    sentencia_A2;
    ...
}
sentencia_B;
```



Ejemplo if, uso de llaves, condiciones y colocación

```
# include <stdio.h>
int main ()
{
    int n;
    printf("Escribe un número entero\n");
    scanf("%i", &n);
    // Completar y avisar al ususario si el número es par

    return 0;
}
```

Ejemplo if, uso de llaves, condiciones y colocación

```
# include <stdio.h>
int main ()
{
    int n;
    printf("Escribe un número entero\n");
    scanf("%i", &n);
    if (n % 2 == 0) // Condición
    { // Uso de llaves
        printf("Se cumple la condición: ");
        printf("El número %i es par.\n", n);
    } // Fin de if
    printf("Gracias por participar.\n");
    return 0;
}
```

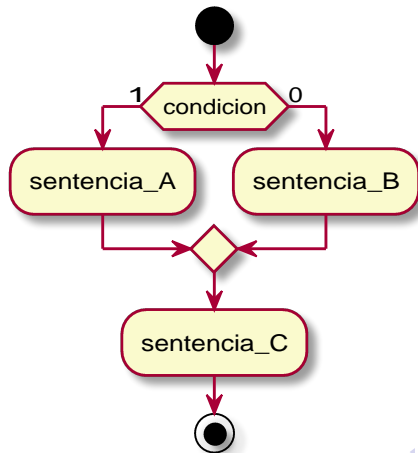

if - else

if Si se cumple condicion se ejecutan las sentencia_A1 y 2.

else En caso contrario se ejecutan las sentencia_B1 y 2.

Siempre se ejecuta la sentencia_C.

```
if (condicion)
{
    sentencia_A1;
    sentencia_A2;
    ...
}
else
{
    sentencia_B1;
    sentencia_B2;
    ...
}
sentencia_C;
```



Ejemplo if-else

```
# include <stdio.h>
int main ()
{
    int n;
    printf("Escribe un número entero\n");
    scanf("%i", &n);

    // Rellene para distinguir entre numeros pares e impares
}
```

Ejemplo if-else

```
# include <stdio.h>
int main (){
    int n;
    printf("Escribe un número entero\n");
    scanf("%i", &n);
    if (n % 2 == 0) // condicion
    { // Inicio de if
        printf("Se cumple la condición: ");
        printf("El número %i es par.\n", n);
    }
    else
    { // Inicio de else
        printf("No se cumple la condición: ");
        printf("El número %i es impar.\n", n);
    } // Fin de if - else
    printf("Gracias por participar.\n");
    return 0;
}
```

if - else - if

if Si se cumple condicion_1 se ejecuta la sentencia_A.

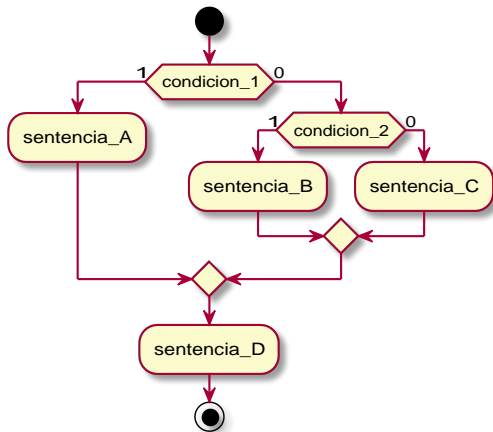
else En caso contrario ...

if si se cumple condicion_2 se ejecuta la sentencia_B.

else En caso contrario se ejecuta sentencia_C.

Siempre se ejecuta la sentencia_D.

```
if (condicion_1) {  
    sentencia_A;  
}  
else {  
    if (condicion_2) {  
        sentencia_B;  
    } else {  
        sentencia_C;  
    }  
}  
sentencia_D;
```



Ejemplo if - else - if

```
#include <stdio.h>
int main() {
    int x;
    printf("Escribe un número: ");
    scanf("%i", &x);

    // Complete para distinguir entre números positivos, negativos y cero

    return 0;
}
```

Ejemplo if - else - if

```
#include <stdio.h>
int main() {
    int x;
    printf("Escribe un número: ");
    scanf("%i", &x);

    if (x < 0) { // se cumple la primera condición
        printf("El número es negativo.\n");
    }
    else
    {
        if (x == 0) // se cumple la segunda
            printf("El número es 0.\n");
        else // no se cumple ninguna
            printf("El número es positivo.\n");
    }
    return 0;
}
```

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

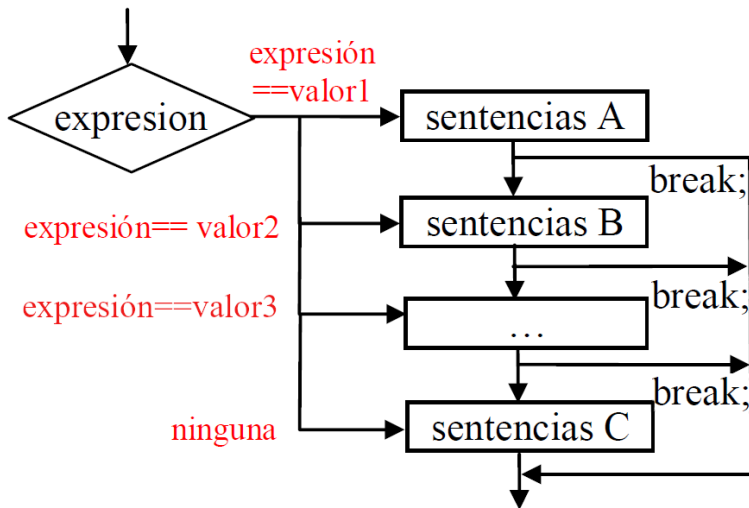
- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

switch-case

- Tomar una decisión múltiple dependiendo del valor **entero** de una expresión.

```
switch (expr)
{
  case valor1:
    sentencia_1;
    break;
  case valor2:
    sentencia_2;
    break;
  case valor3:
    sentencia_3;
    break;
  ...
  default:
    sentencia_n;
    break;
}
```



Ejemplo con switch

```
#include <stdio.h>
int main ()
{
    float v1, v2;
    char op;
    printf("Escriba 2 números y la operación que quiera realizar: \n");
    scanf("%f %c %f", &v1, &op, &v2);
    // Usar switch para realizar la operación desada por el usuario

    return 0;
}
```

Ejemplo con switch

```
#include <stdio.h>
int main (){
    float v1, v2;
    char op;
    printf("Escriba 2 números y la operación que quiera realizar: \n");
    scanf("%f %c %f", &v1, &op, &v2);
    switch(op)
    {
        case '+':
            printf("Operación Suma:\n");
            printf("%.2f\n", v1 + v2);
            break;
        case '-':
            printf("Operación Resta:\n");
            printf("%.2f\n", v1 - v2);
            break;
        default:
            printf("No se hacer esa operación.\n");
    }
    return 0;
}
```

Atención al uso de break

```
#include <stdio.h>
int main ()
{
    float v1, v2;
    char op;
    scanf("%f %c %f", &v1, &op, &v2);
    switch(op)
    {
        case '+':
            printf("%.2f\n", v1 + v2);
        case '-':
            printf("%.2f\n", v1 - v2);
        default:
            printf("No se hacer esa operación.\n");
            break;
    }
    return 0;
}
```

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

for : Código y flujo

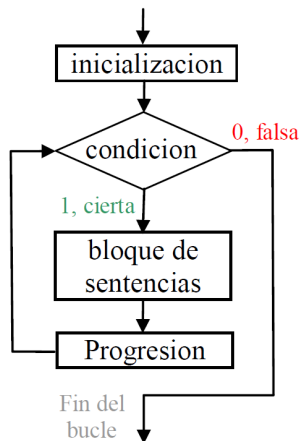
Ejecuta las sentencias un número determinado de veces hasta que el resultado de la expresión final sea falso.

```
for (expr_inicio; expr_final; expr_avance)
{
    sentencia_1;
    sentencia_2;
    ...
}
```

expr_inicio Expresión de inicialización (se ejecuta una sola vez). Iniciar las variables de control del bucle.

expr_final Expresión de finalización. Si es falsa se acaba el bucle.

expr_avance Expresión de progresión de las variables de control bucle.



Ejemplo: suma de enteros

```
#include <stdio.h>

int main()
{
    // Use un bucle FOR para calcular la suma de los 9 primeros números naturales

    return 0;
}
```

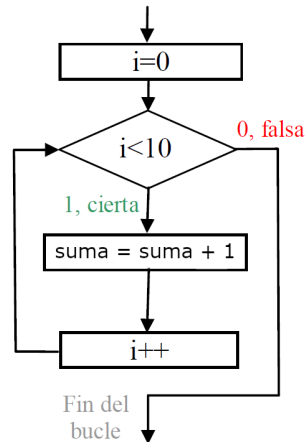
Ejemplo: suma de enteros

```
#include <stdio.h>

int main()
{
    // MUY IMPORTANTE asignar valor inicial 0 a suma
    int i, suma = 0;

    for (i = 1; i < 10; i++)
    {
        suma = suma + i;
    }
    printf("La suma de los 9 primeros enteros es %i", suma);

    return 0;
}
```



Ejemplo: alfabeto

```
#include <stdio.h>

int main()
{
    // Use un bucle FOR para escribir el alfabeto en minúsculas

    return 0;
}
```

Ejemplo: alfabeto

```
#include <stdio.h>

int main()
{
    char i;
    // Se pueden usar char en las expresiones
    for (i = 'a'; i <= 'z'; i++)
    {
        printf("%c", i);
    }
    return 0;
}
```

Ejemplo: bucles anidados. Imprimir la tabla de multiplicar del 1 al 9.

```
#include <stdio.h>

int main()
{
    int i, j;
    // Use 2 bucles FOR anidados para imprimir las tablas de multiplicar del 1 al 9.

    return 0;
}
```

Ejemplo: bucles anidados. Imprimir la tabla de multiplicar del 1 al 9.

```
#include <stdio.h>

int main()
{
    int i, j;
    for (i = 1; i < 10; i++)
    {
        printf("Tabla del %i\n", i);
        for (j = 1; j < 10; j++)
        {
            printf("%i x %i = %i\n", i, j, i * j);
        }
    }
    return 0;
}
```

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- **`while`**
- `do-while`
- Ejemplo equivalencia

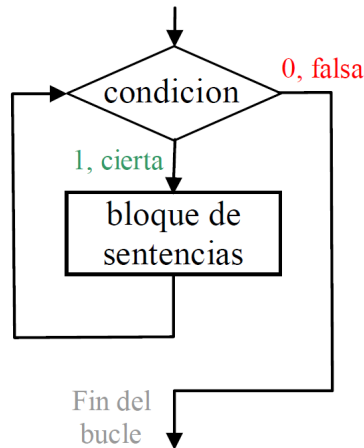
4 Rupturas

while: Código y flujo

Ejecuta una sentencia (simple o compuesta) **cero o más veces** dependiendo del resultado booleano de una expresión.

```
while (expresion)
{
    sentencia_1;
    sentencia_2;
    ...
    cambio_valor_expresión;
}
```

- Si la primera vez que se evalúa la condición es falsa, las sentencias **no se ejecutan nunca**.
- Si la condición es siempre verdadera **el bucle es infinito**.



Ejemplo, cuenta atrás.

```
#include <stdio.h>

int main()
{
    int i = 10;
    // Use un bucle WHILE para imprimir una cuenta atrás
    // que empiece en 10 y acabe en 1

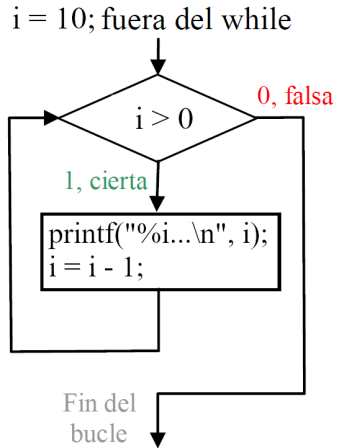
    return 0;
}
```

Ejemplo, cuenta atrás.

```
#include <stdio.h>

int main()
{
    int i = 10;
    while (i > 0)
    {
        printf("%i...\n", i);
        i = i - 1;
    }
    printf("Despegue!");

    return 0;
}
```



1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

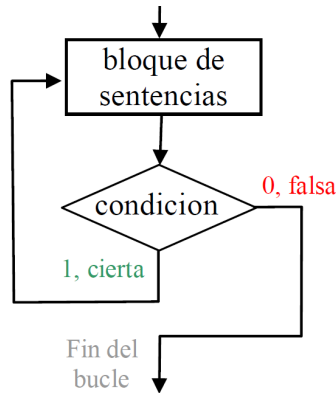
4 Rupturas

do-while: Código

Ejecuta las sentencias **una o más veces** dependiendo del resultado de la expresión.

```
do
{
    sentencia_1;
    sentencia_2;
    ...
}
while (expresion);
```

- Si la primera vez que se evalúa la expresión es falsa, la sentencia se habrá ejecutado **una vez**.
- Si la expresión es siempre verdadera **el bucle es infinito**.



Ejemplo: número entero al revés

```
#include <stdio.h>

int main()
{
    int num = 123456, cifra;
    // Escriba un programa que escriba las cifras de un número entero
    // de menos a más significativa
    // usando un bucle DO WHILE

    return 0;
}
```

Ejemplo: número entero al revés

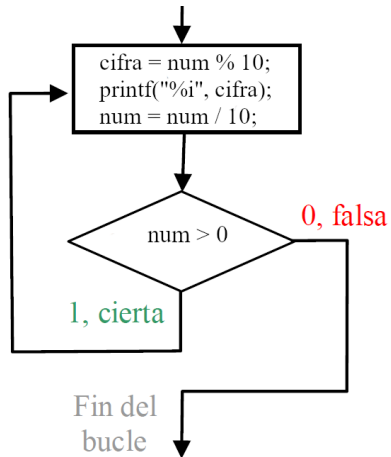
```
#include <stdio.h>

int main()
{
    int num = 123456, cifra;

    do
    {
        cifra = num % 10;
        printf("%i", cifra);
        num = num / 10;
    } while (num > 0);

    printf("\n");

    return 0;
}
```



¿Qué bucle elegir?

- Si se conoce el número de veces que debe ejecutarse el bucle: usar `for`.
- Si el número de veces es desconocido a priori:
 - ▶ Si debe realizarse al menos una vez se debe usar `do-while`.
 - ▶ Si no es imprescindible que se ejecute alguna vez, se puede usar `while`.

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

Equivalencia, imprimir los número del 0 al 10

```
#include <stdio.h>
int main() {
    int i,n = 10;
    // Bucle for
    for (i = 0; i <= n; i++) {
        printf("%i ",i);
    }
    // Bucle while
    i=0;
    while( i<= n) {
        printf("%i ",i);
        i++;
    }
    // Bucle do-while
    i=0;
    do {
        printf("%i ",i);
        i++;
    } while(i <= n);
    return 0;
}
```

1 Introducción

2 Sentencias Condicionales

- `if - else`
- `switch - case`

3 Sentencias Repetitivas (Bucles)

- `for`
- `while`
- `do-while`
- Ejemplo equivalencia

4 Rupturas

break y continue

break

- Finaliza la ejecución de un bucle (si el bucle está anidado sólo finaliza él, pero no los bucles más externos).

continue

- Ejecuta la siguiente iteración del bucle.
- En un bucle while o do-while vuelve a `expresion`.
- En un bucle for ejecuta `expr_avance` y a continuación comprueba `expr_final`