

Unidad temática 3: Estructuras de repetición

1. Introducción

Las estructuras de repetición permiten realizar un conjunto de acciones de manera repetida (o cíclica), siempre que el resultado de evaluar una determinada condición sea verdadero. A este tipo de estructuras también se las conoce como *bucles*.

Existen tres tipos de estructuras de repetición: **while**, **do-while** y **for**.

2. Estructura de repetición *while* (mientras)

En un bucle *while* primero se evalúa la condición, y en el caso de que ésta sea verdadera, se realizarán las acciones. Una vez completado el anterior ciclo, se repetirá el mismo procedimiento. Las acciones se realizarán repetidamente *mientras* el resultado de evaluar la condición sea verdadero.

A continuación, se muestra el diagrama de flujo y su sintaxis en C de una estructura de repetición *while*:

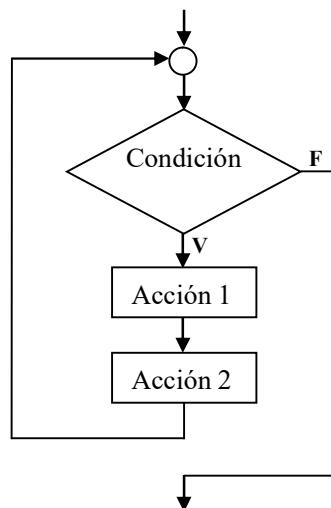


Diagrama de flujo de una estructura de repetición *while*

```
while(condicion)
{
    accion1;
    accion2;
}
```

Sintaxis en C en una estructura de repetición *while*

Para comprender como se aplica esta estructura a la hora de resolver problemas, a continuación se muestra un ejemplo:

Ejemplo: Se desea realizar un algoritmo que imprima en pantalla los primeros diez números naturales, utilizando una estructura de repetición *while*. Realizar el diagrama de flujo y programa en C del algoritmo.

Solución: El algoritmo deberá, en primera instancia, declarar la variable que se va a utilizar (*conta*), y asignarle el primer número natural. Luego, se utilizará la estructura de repetición para imprimir el número en pantalla, además de incrementar en una unidad a la variable *conta* (de esa forma, en el siguiente ciclo se imprimirá el siguiente número natural). El diagrama de flujo del algoritmo es el siguiente:

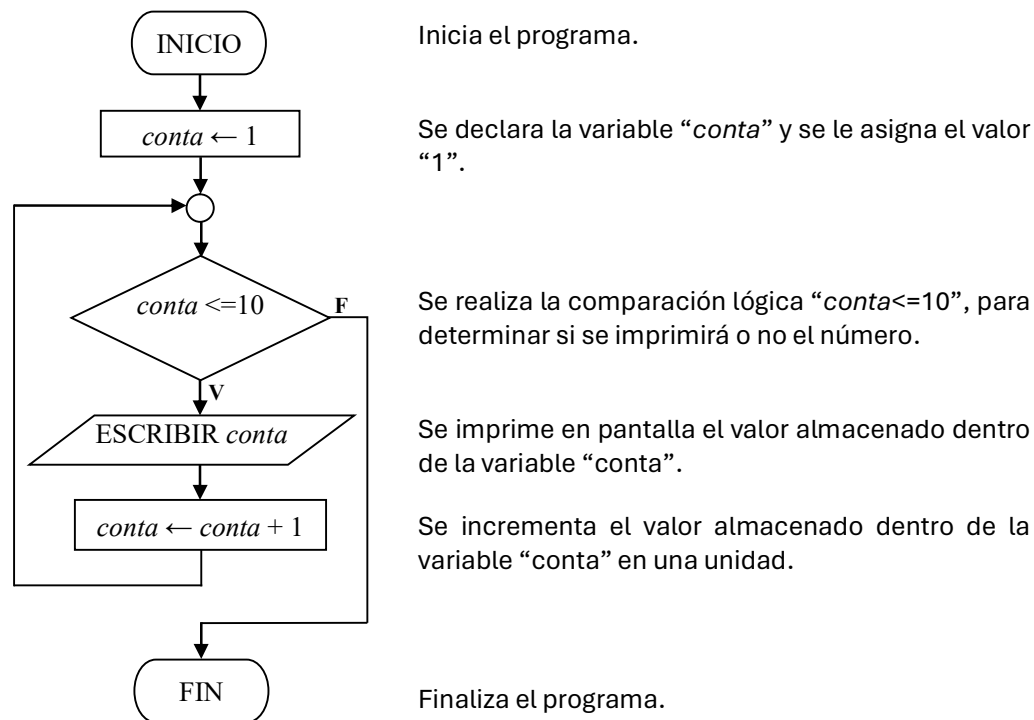


Diagrama de flujo del algoritmo

Una vez realizado el diagrama de flujo del algoritmo, se deberá expresar en un lenguaje de programación. El lenguaje pedido por la consigna es el C:

```
#include <stdio.h>

int main(void)
{
    int conta = 1; // Declara la variable entera "conta" asignándole 1.

    while(conta <= 10) // Se evalúa la condición de la estructura
    {
        printf("%d ", conta); // Se imprime en pantalla el número.
        conta = conta + 1; // Incrementa el valor de conta en uno.
    }
    return 0;
}
```

Programa en C del algoritmo

3. Estructura de repetición *do-while* (hacer mientras)

En un bucle *do-while* primero se realizan las acciones y después se evalúa la condición. En el caso de que ésta sea verdadera, se vuelven a realizar las acciones. Las acciones se realizarán en forma repetida hasta que la condición sea falsa.

A continuación, se muestra el diagrama de flujo y su sintaxis en C de una estructura de repetición *do-while*:

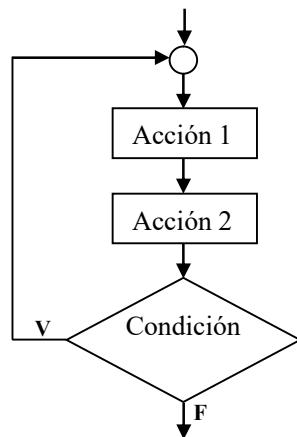


Diagrama de flujo de una estructura de repetición do-while

```
do
{
    accion1;
    accion2;
} while(condicion);
```

Sintaxis en C en una estructura de repetición do-while

Para comprender como se aplica esta estructura a la hora de resolver problemas, a continuación se muestra un ejemplo:

Ejemplo: Se desea realizar un algoritmo que imprima en pantalla los primeros diez números naturales, utilizando una estructura de repetición *do-while*. Realizar el diagrama de flujo y programa en C del algoritmo.

Solución: El algoritmo deberá, al igual que en el ejemplo anterior, declarar la variable que se va a utilizar (*conta*), y asignarle el primer número natural. Luego, se utilizará la estructura de repetición para imprimir el número en pantalla, además de incrementar en una unidad a la variable *conta* (de esa forma, en el siguiente ciclo se imprimirá el siguiente número natural). El diagrama de flujo del algoritmo es el siguiente:

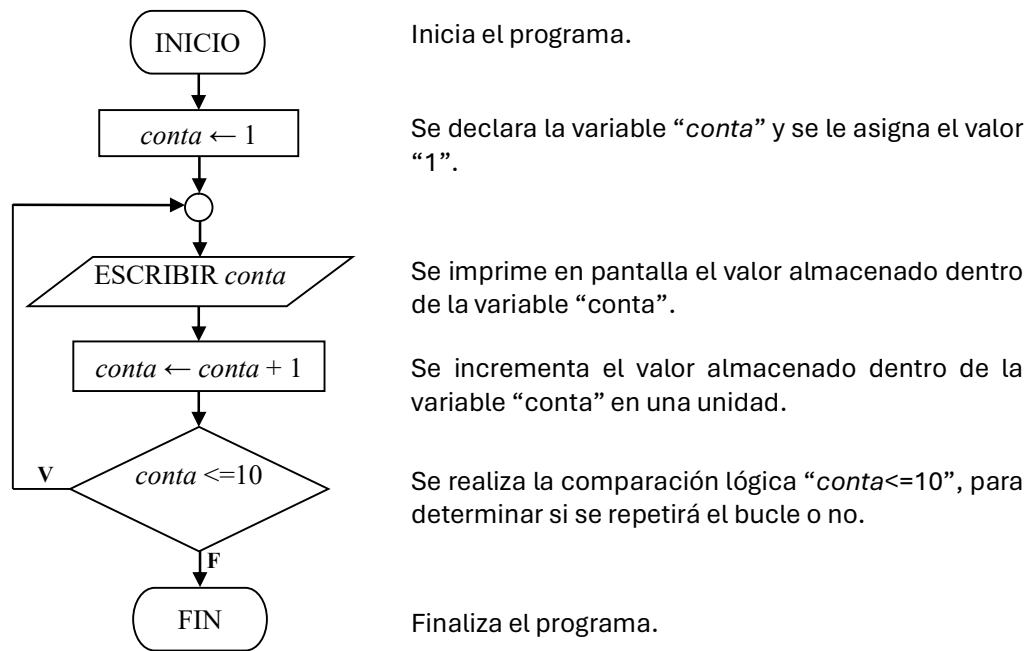


Diagrama de flujo del algoritmo

Una vez realizado el diagrama de flujo del algoritmo, se deberá expresar en un lenguaje de programación. El lenguaje pedido por la consigna es el C:

```

#include <stdio.h>

int main(void)
{
    int conta = 1; // Declara la variable entera "conta" asignándole 1.

    do
    {
        printf("%d ", conta); // Se imprime en pantalla el número.
        conta = conta + 1; // Incrementa el valor de conta en uno.
    } while (conta <= 10); // Se evalúa la condición de la estructura.

    return 0;
}
  
```

Programa en C del algoritmo

4. Estructura de repetición *for* (desde y hasta)

En un bucle *for* las acciones se realizan un número exacto de veces. Se deben definir una variable contadora con su valor de inicio, la condición de permanencia del bucle, y el incremento de la variable contadora. El *for* realizará en forma iterativa las acciones hasta que no se cumpla la condición de permanencia, y en cada iteración incrementará el valor de la variable contadora.

A continuación, se muestra el diagrama de flujo y su sintaxis en C de una estructura de repetición *for*:

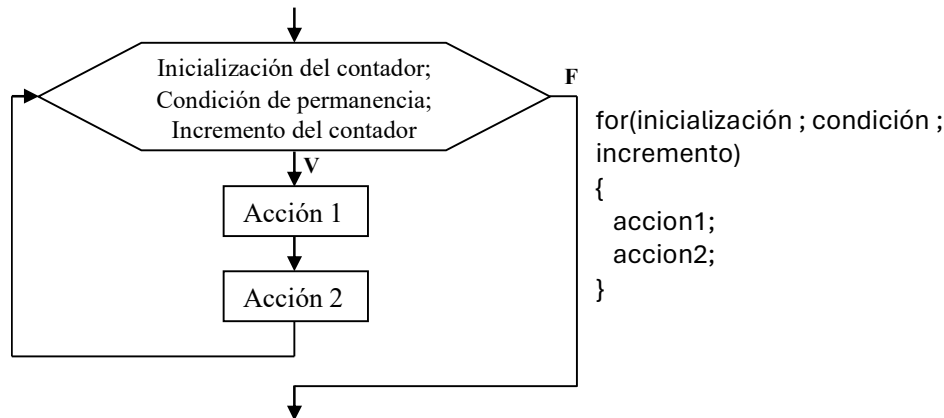


Diagrama de flujo de una estructura de repetición for

Sintaxis en C en una estructura de repetición for

A continuación, se muestran algunas consideraciones para tener en cuenta al utilizar esta estructura:

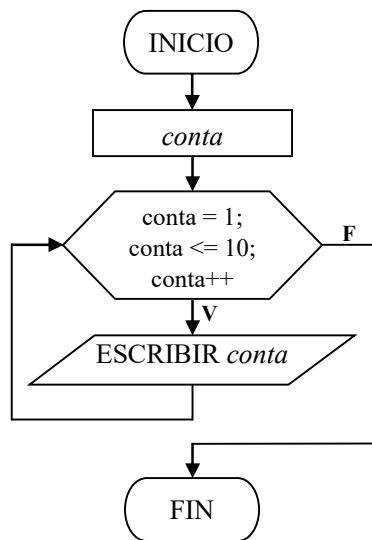
- Cuando el bloque de las acciones del bucle *for* sólo tiene una única acción, los tokens “{” y “}” son opcionales.
- En la inicialización del contador se especifica el valor inicial de la variable contador. Si llamamos “i” a la variable contador, ejemplos típicos de inicialización pueden ser “i=0”, “i=1”, “i=-1”, etc.
- La condición de permanencia es la condición que se tiene que cumplir para que el *for* permanezca iterando dentro del bucle. Cuando esta condición no se cumpla más, el *for* finalizará. Ejemplos típicos de condiciones de permanencia son “i<10”, “i<=100”, etc.
- El incremento de la variable contadora definirá cuánto se incrementará el contador en cada iteración. Ejemplos típicos de incrementos son “i=i+1”, “i++”, “i=i+2”, “i+=2”, “i=i-1”, “i--”, “i=i-2”, “i-=2”, etc.

Para comprender como se aplica esta estructura a la hora de resolver problemas, a continuación se muestra un ejemplo:

Ejemplo: Se desea realizar un algoritmo que imprima en pantalla los primeros diez números naturales, utilizando una estructura de repetición *for*. Realizar el diagrama de flujo y programa en C del algoritmo.

Solución: Al igual que en las soluciones anteriores, el algoritmo deberá declarar la variable que se va a utilizar (*conta*). Luego, se utilizará la estructura de repetición *for* para imprimir los números en pantalla, definiendo la *inicialización del contador*, la *condición de*

permanencia y el incremento del contador. El diagrama de flujo del algoritmo es el siguiente:



Inicia el programa.

Se declara la variable “*conta*” como variable contadora.

Se define la estructura de repetición *for*, indicando la inicializando el contador (*conta* = 1), la condición de permanencia (*conta* <= 10) y el incremento del contador (*conta* incrementa una unidad en cada ciclo).

Se imprime en pantalla el valor almacenado dentro de la variable “*conta*”.

Finaliza el programa.

Diagrama de flujo del algoritmo

Una vez realizado el diagrama de flujo del algoritmo, se deberá expresar en un lenguaje de programación. El lenguaje pedido por la consigna es el C:

```
#include <stdio.h>

int main(void)
{
    int conta;                // Declara la variable entera "conta".

    for(conta=1 ; conta<=10 ; conta++)    // Estructura repetitiva for.
    {
        printf("%d ", conta);           // Se imprime en pantalla el número.
    }
    return 0;
}
```

Programa en C del algoritmo