

ESTRUCTURAS REPETITIVAS

Catedrático: Ing. David Rajo

INTRODUCCIÓN

- Una de las características de las computadoras que aumentan considerablemente su potencia es su capacidad para ejecutar una tarea muchas veces con gran velocidad, precisión y fiabilidad.
- Las tareas repetitivas es algo que las personas encontramos tediosas de realizar.
- Son operaciones que se deben ejecutar un número repetido de veces. El conjunto de instrucciones que se ejecuta repetidamente cierto número de veces, se llama Ciclo, Bucle o Lazo
- A continuación vamos a estudiar los 3 tipos de estructuras repetitivas que soporta C++
 - While
 - For
 - Do-while

OBJETIVOS

- Conocer las estructuras repetitivas en C++
- Elaborar programas utilizando estructuras repetitivas en c++

¿Qué es un Bucle?

- Es cualquier construcción de programa que repite una sentencia o secuencia de sentencias un número de veces. La sentencia o grupo de sentencias que se repiten en un bloque se denomina **cuerpo**.

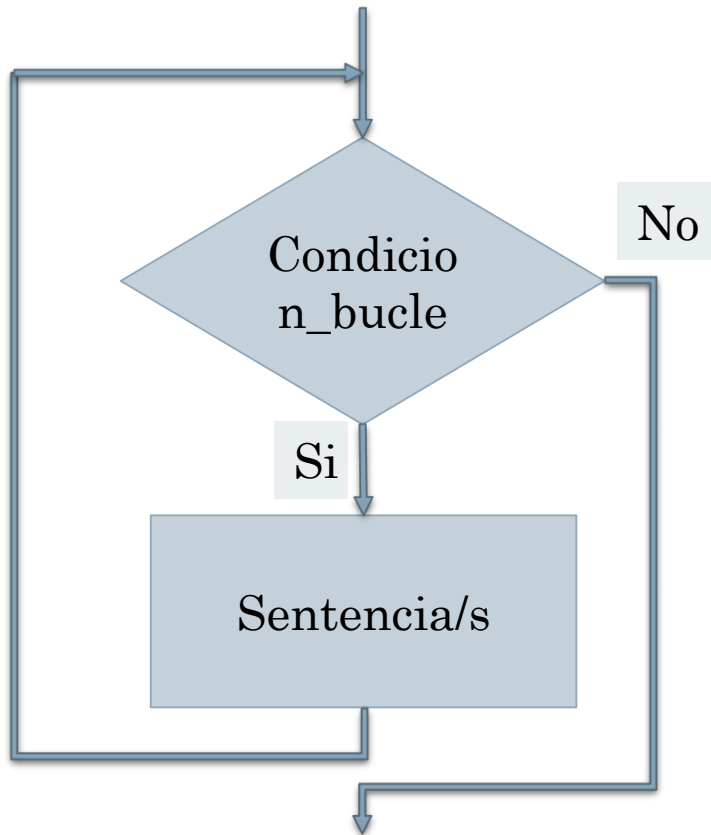
Estructuras repetitivas básicas en C++

- Son 3:
 - For (Desde/Para)
 - While (Mientras)
 - Do-While (Repetir)
- Las dos principales cuestiones del diseño en la construcción de un bucle son:
 - ¿Cuál es el cuerpo del bucle?
 - ¿Cuántas veces se iterará el cuerpo del bucle?

La Sentencia While (Mientras)

- Se llama Mientras a la estructura algorítmica que se ejecuta mientras la condición evaluada resulte verdadera. Se evalúa la expresión y, si es cierta, se ejecuta la instrucción especificada, llamada el cuerpo del bucle. Entonces se vuelve a evaluar la expresión, y si todavía es cierta se ejecuta de nuevo el cuerpo. Este proceso de evaluación de la expresión y ejecución del cuerpo se repite mientras la expresión sea cierta. Cuando se hace falsa, finaliza la repetición
- Un bucle *while* tienen una condición del bucle (una expresión lógica) que controla la secuencia de repetición.

Diagrama de flujo de un bucle while



1. Se evalúa la *condicion_bucle*
2. Si la *condicion_bucle* es verdadera:
 - a) La sentencia/s se ejecutan
 - b) Vuelve el control al paso 1
3. En caso contrario: el control se transfiere a la sentencia siguiente.

Sintaxis while

```
1      while (condicion_bucle)
           sentencia;

2      while (condicion_bucle){
           sentencia1;
           sentencia2;
           ....
           sentencia N;
       }
```


Ejemplo

- Elaborar un programa que cuente del 1 al 10

Bucles controlados por centinela o bandera

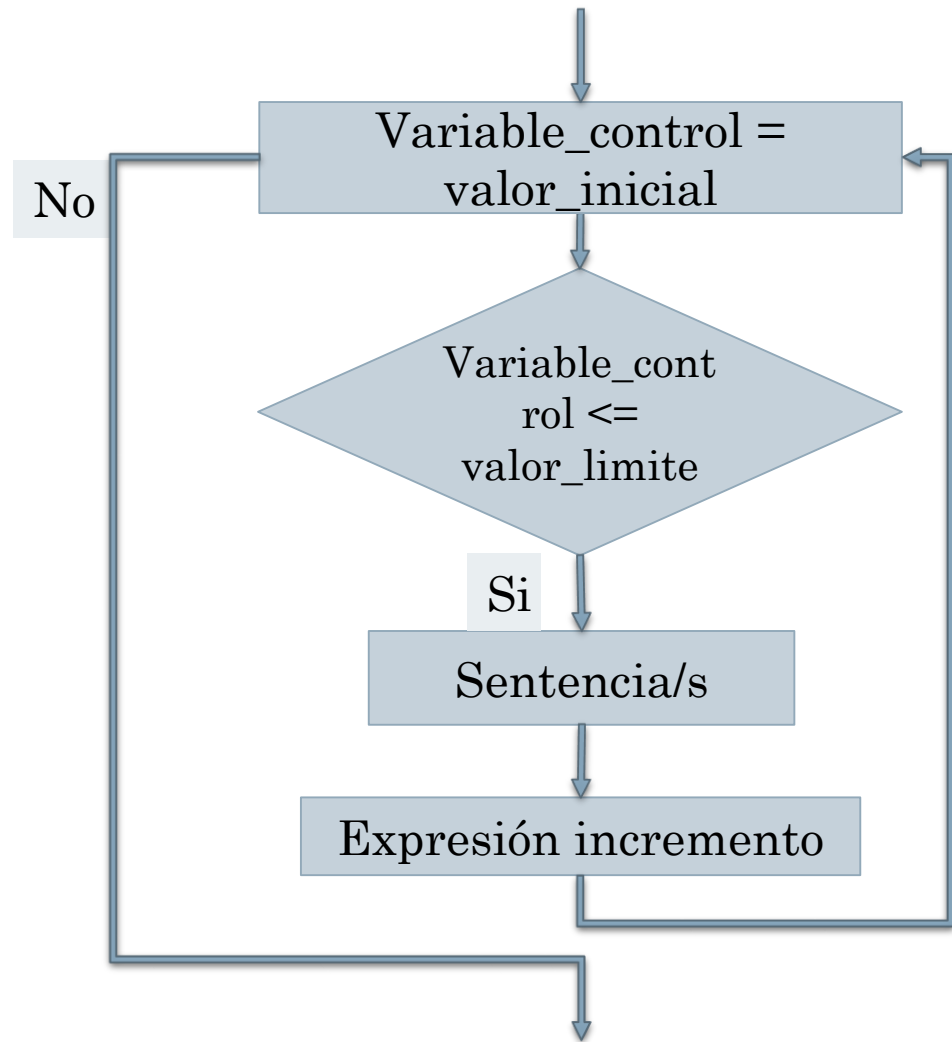
- Centinela
 - Cuando no se conoce con exactitud cuántos elementos de datos se presentarán, el medio para manejar esta situación es el centinela donde el usuario debe introducir un único dato definido denominado *valor centinela* como último dato.
- Bandera
 - Las variables tipo *bool* se utilizan con frecuencia como banderas para controlar la ejecución de un bucle. Un bucle controlado por bandera se ejecuta hasta que se produce un suceso anticipado y se cambia el valor del indicador.

La sentencia For (Desde/Para)

- Son bucles controlados por un contador
- La sentencia for es un método para ejecutar un bloque de sentencias un número fijo de veces.
- El bucle for se diferencia del bucle while en que las operaciones de control se sitúan en un solo sitio:
 - La cabecera de la secuencia

```
for (inicialización; condición; incremento){  
    sentencias  
}
```

Diagrama de flujo de un bucle for



```
1      for (inicialización; condición; incremento)
           sentencia;

2      for (inicialización; condición; incremento){
           sentencia1;
           sentencia2;
           ...
           sentencian;
      }
```

Donde:

Inicialización: inicia las variables de control del bucle.

Condición: contiene la expresión lógica que hace que el bucle realice las iteraciones de las sentencias, mientras que la expresión sea verdadera.

Incremento: incrementa o decrementa la variable de control o variables de control de bucle.

Ejemplo

- Elaborar un programa que cuente del 1 al 10

Contadores y Acumuladores

Contadores

- La construcción de un contador es una de las técnicas más comunes en la realización de diagramas de flujo.
- Es una variable en la memoria que **se incrementará en una unidad constante** cada vez que se ejecute el proceso.
- El contador se utiliza para llevar la cuenta de determinadas acciones que se pueden solicitar durante la resolución de un problema.

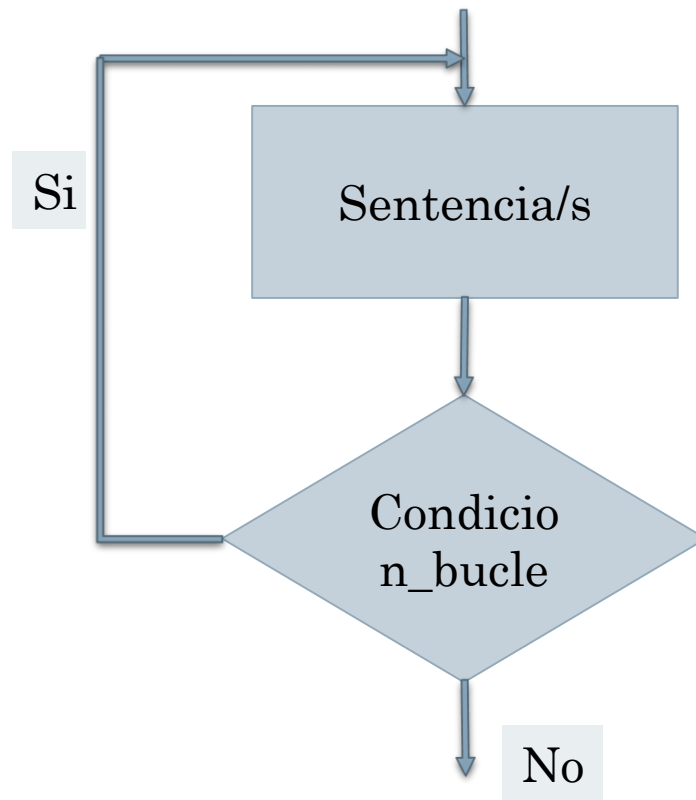
Acumuladores

- Un ***acumulador*** es una variable en la memoria cuya misión es almacenar cantidades variables.
- Se utiliza para efectuar ***sumas sucesivas***. La **principal diferencia** con el contador es que el incremento o decremento de cada suma es ***variable*** en lugar de constante como en el caso del contador.

La sentencia Do-While

- La sentencia do-while se utiliza para especificar un bucle condicional que se ejecuta al menos una vez

Diagrama de flujo de la sentencia **do-while**



Sintaxis do-while

1 do sentencia while (Expresion)

2 do{

 sentencia1

 sentencia2

 sentencian

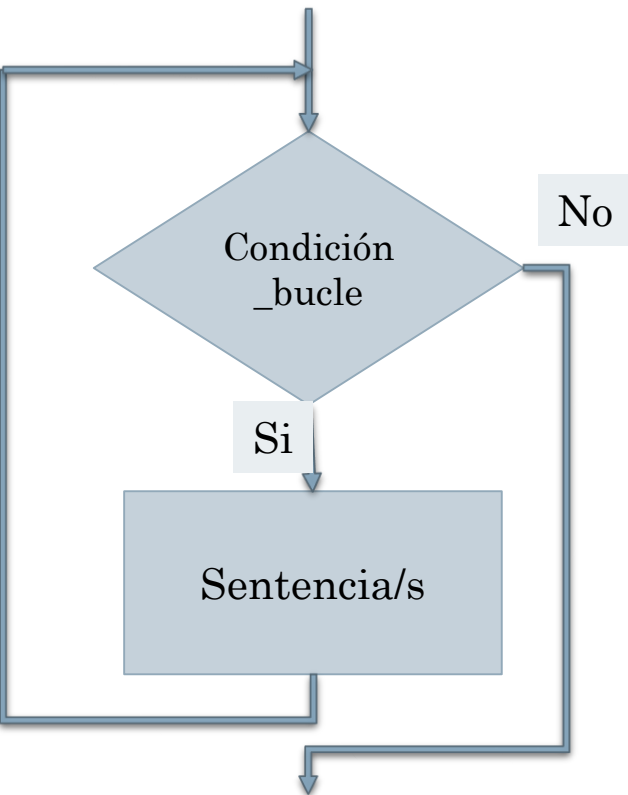
}while (Expresion);

Ejemplo

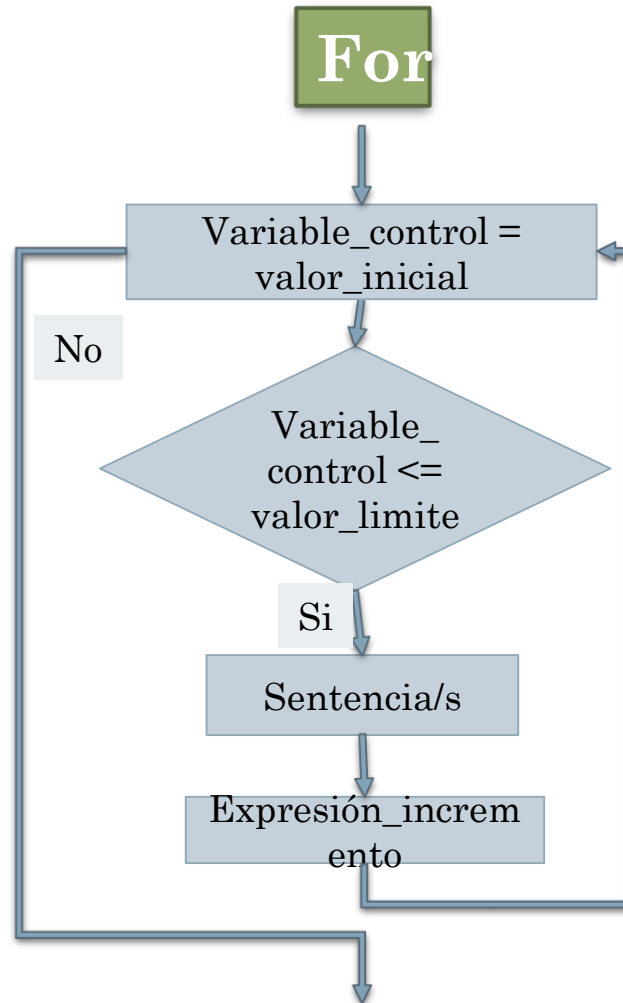
- Elaborar un programa que cuente del 1 al 10

Comparación de bucles

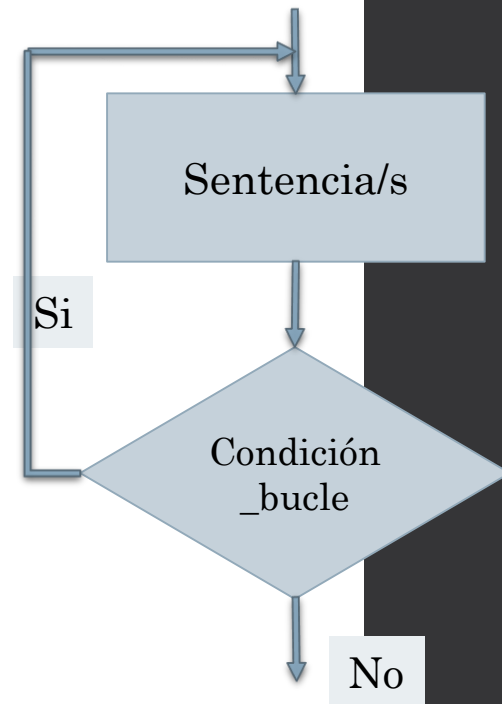
While



For



Do-While



Ejercicios

1. Elaborar un programa que encuentre el mayor de N números.
2. Escribir un programa que imprima una tabla de m columnas por n filas y un carácter prefijado.
3. Elabore un programa que defina si un número es primo o no.
4. Modifique el programa de la calculadora para que pueda continuar realizando operaciones hasta que el usuario decida terminar.
5. Elaborar un programa que calcule el factorial de un número