Instrucciones de control de flujo: sentencias para la realización de iteraciones bucles while, for y do-while

| **Summary** |  |
| --- | --- |
| **URL** | cp4 |
| **Category** | Web |
| **Environment** | web |
| **Status** | Published |
| **Feedback Link** |  |
| **Author** |  |

[Introducción](#_tte8vcs9wano).

While

For

Do While

# Introducción

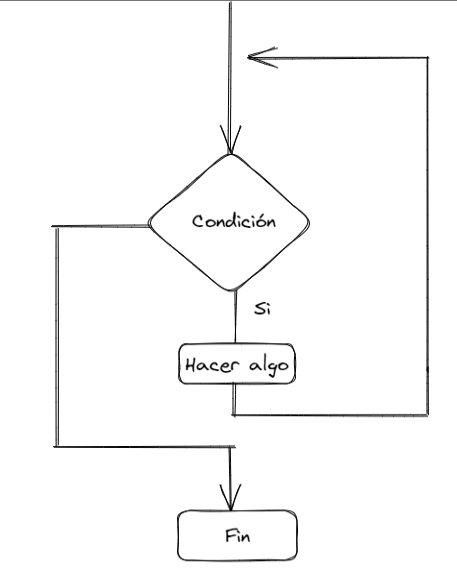
Los bucles son una herramienta fundamental en programación que te permiten repetir una serie de instrucciones varias veces sin tener que escribir el mismo código una y otra vez. En lugar de escribir el mismo código repetidamente, puedes utilizar un bucle para que el programa lo ejecute automáticamente. Esto ahorra tiempo y hace que tu código sea más eficiente y fácil de leer y mantener.

En la mayoría de los lenguajes de programación, hay varios tipos de bucles, cada uno con su propia sintaxis y uso. Algunos de los más comunes son el bucle while, el bucle for y el bucle do-while. Cada tipo de bucle tiene sus propias características y ventajas, y es importante conocerlos para poder utilizarlos de manera efectiva en diferentes situaciones.

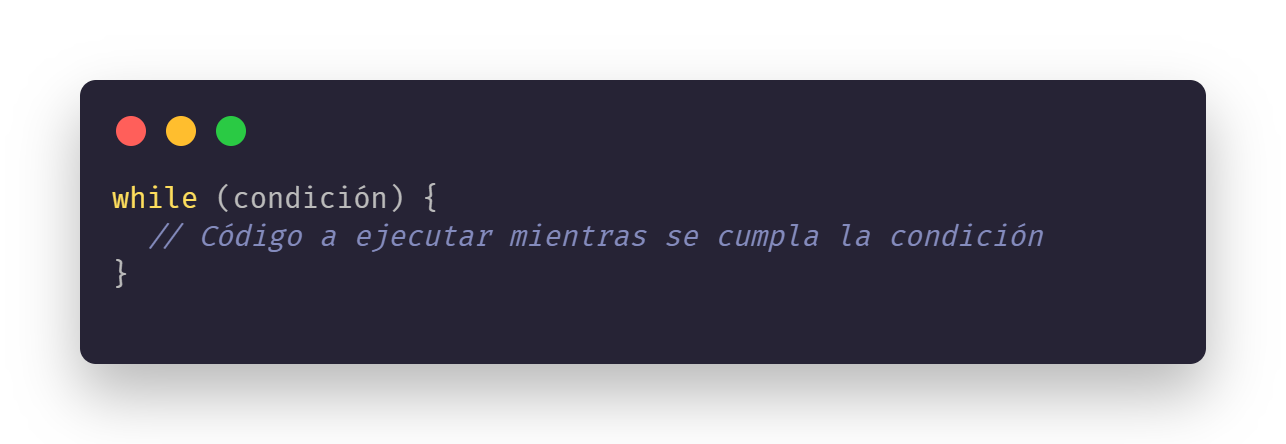
Los bucles son una herramienta poderosa en programación, y su uso adecuado puede hacer una gran diferencia en la eficiencia y calidad de tu código.

# While

El bucle while es una estructura de control de flujo en programación que se utiliza para repetir una acción o un conjunto de acciones mientras se cumple una determinada condición. El bucle se ejecuta siempre que la condición especificada sea verdadera y se detiene cuando la condición se vuelve falsa.



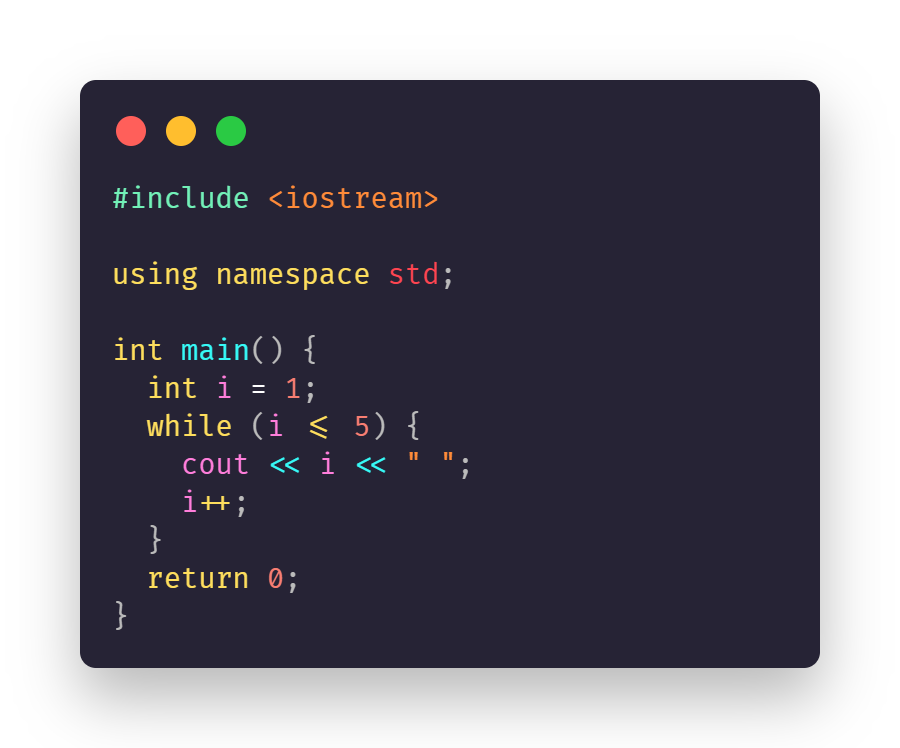
La sintaxis básica en C++ es la siguiente:



La condición es una expresión booleana que se evalúa en cada iteración del bucle. Si la condición es verdadera, el código dentro del bucle se ejecuta. Si la condición es falsa, el programa sale del bucle y continúa con la siguiente instrucción.

Es importante tener en cuenta que la condición debe cambiar de valor en algún momento dentro del bucle, de lo contrario, el bucle se convertirá en un ciclo infinito y el programa se quedará atascado.

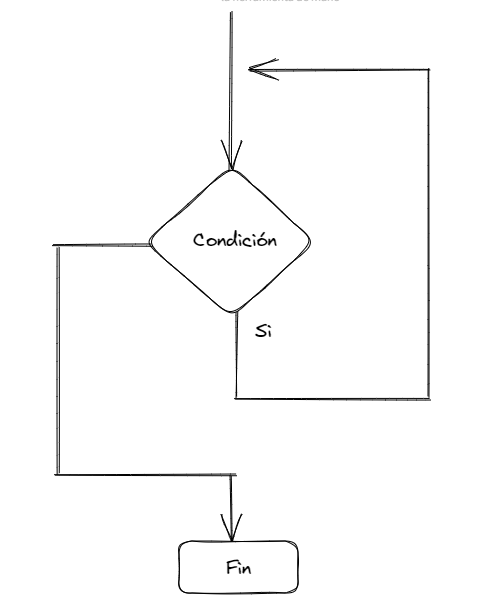
Un ejemplo sencillo de uso del While en C++ es el siguiente:

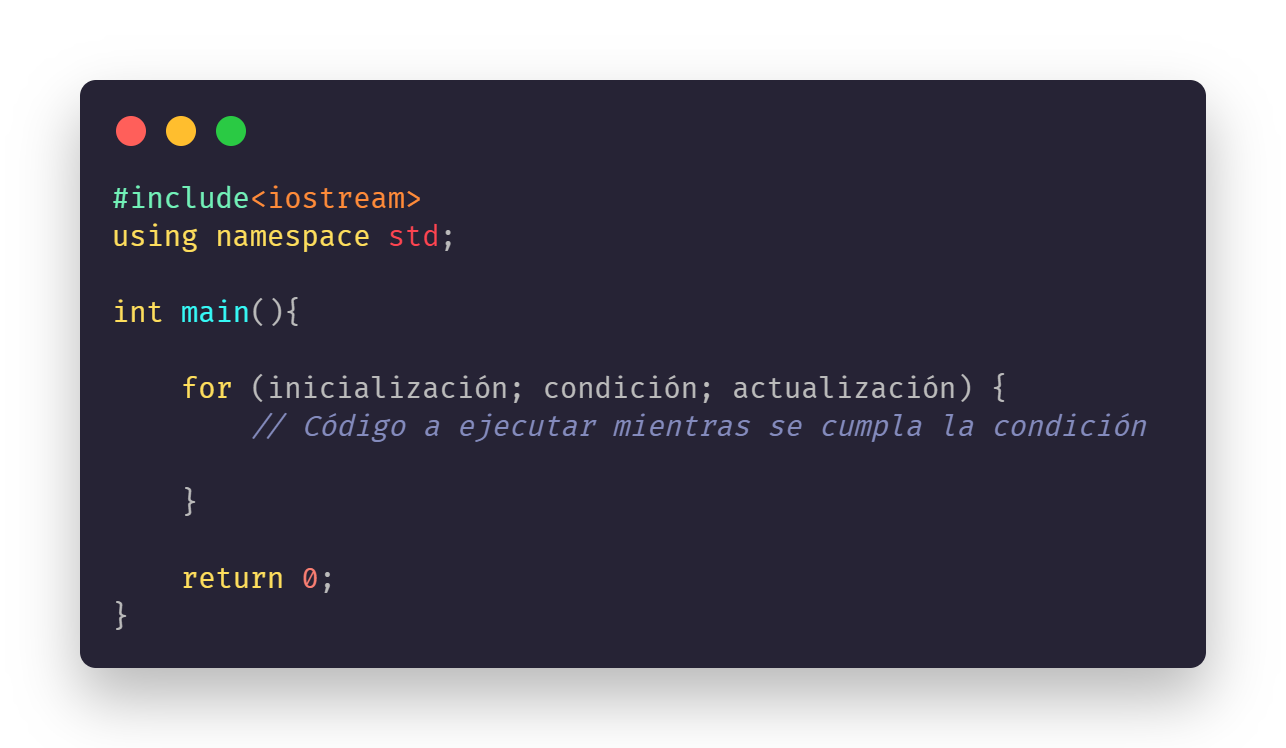


En este ejemplo, se inicializa la variable i con el valor de 1. Luego, se especifica la condición i <= 5, lo que significa que el bucle se ejecutará mientras el valor de i sea menor o igual a 5. Dentro del bucle, se muestra el valor de i en la consola y se incrementa su valor en 1. El bucle se ejecutará cinco veces, mostrando los números del 1 al 5 en la consola. Al final, el programa retorna 0 para indicar que finalizó correctamente.

# For

El ciclo for es una estructura de control de flujo en programación que se utiliza para repetir una acción o un conjunto de acciones un número determinado de veces. Su funcionamiento es similar al ciclo While pero el for es muy útil para iterar sobre una secuencia de elementos o para realizar tareas repetitivas con una cantidad conocida de iteraciones.



La sintaxis básica del for es la siguiente:

La inicialización se ejecuta una sola vez al principio del ciclo y se utiliza para establecer el valor inicial de la variable de control del bucle. La condición se evalúa en cada iteración del bucle y determina si el bucle debe seguir ejecutándose. La actualización se ejecuta al final de cada iteración del bucle y se utiliza para cambiar el valor de la variable de control.

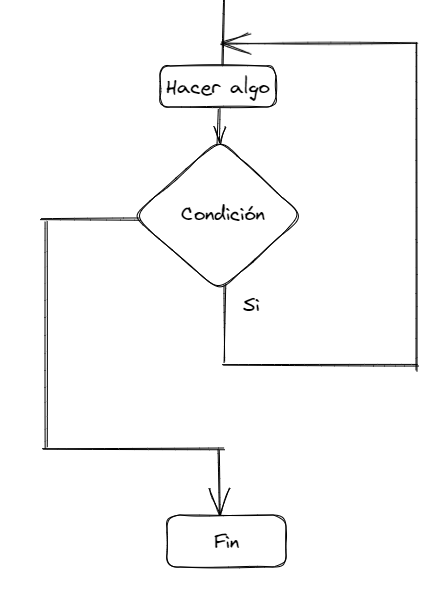
Dentro del bloque de código, es necesario asegurarse de que en algún momento se cambie el valor de la variable que se utiliza en la condición del bucle, para evitar que el bucle se convierta en un ciclo infinito.

Un ejemplo sencillo de uso del For en C++ es el siguiente:

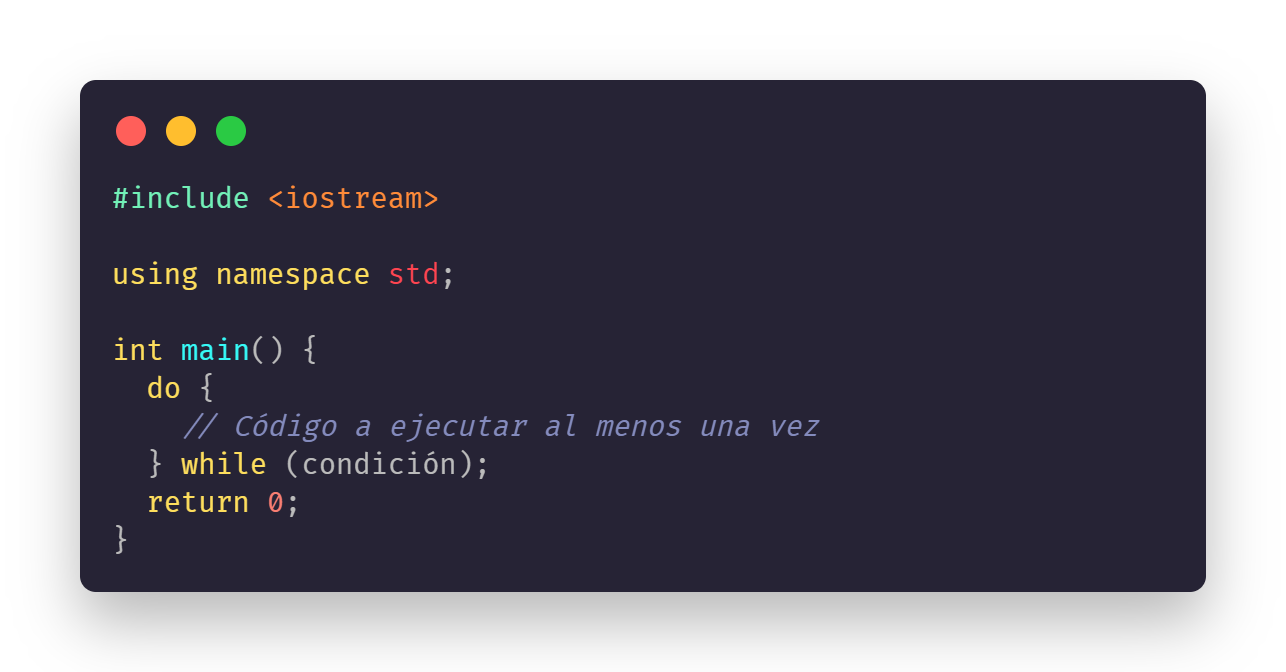


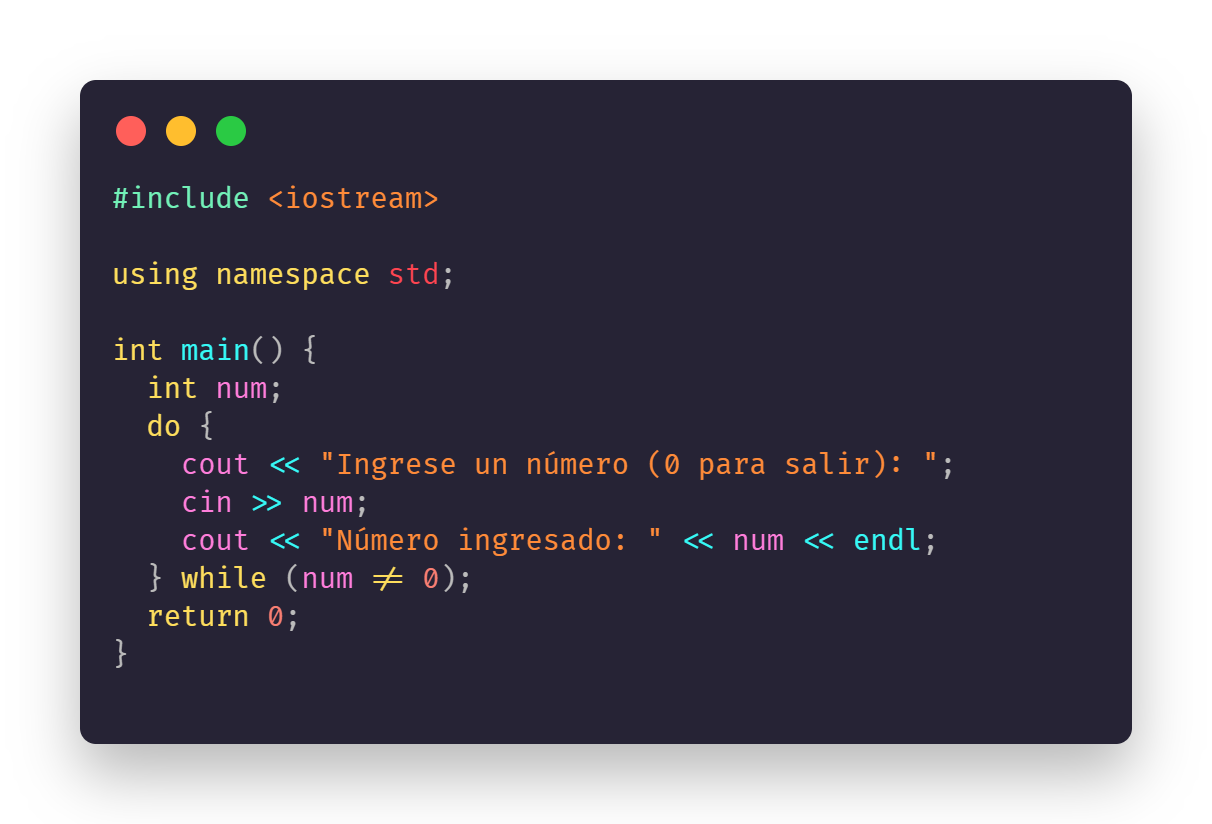
# Do While

El ciclo do-while es una estructura de control de flujo en programación que se utiliza para repetir una acción o un conjunto de acciones al menos una vez, y luego repetirla mientras se cumpla una condición específica. Esta estructura es muy útil cuando se quiere asegurarse de que una acción se ejecute al menos una vez, independientemente de la condición.



La sintaxis básica del ciclo do-while es la siguiente:



En este caso, el bloque de código dentro del do se ejecuta al menos una vez, sin importar si se cumple o no la condición especificada en el while. Después de la primera ejecución, la condición se evalúa y si es verdadera, el bloque de código se ejecuta nuevamente. Si la condición es falsa, el ciclo do-while termina y la ejecución continúa con la siguiente instrucción después del bloque de código.

En este ejemplo, se utiliza el ciclo do-while para pedir al usuario que ingrese un número y mostrarlo en la consola, y se repite hasta que el usuario ingrese el número 0. El bloque de código dentro del do se ejecuta al menos una vez, sin importar el valor inicial de num. Después de la primera ejecución, la condición num != 0 se evalúa y si es verdadera, el bloque de código se ejecuta nuevamente. Si la condición es falsa (es decir, si el usuario ingresa el número 0), el ciclo do-while termina y la ejecución continúa con la siguiente instrucción después del bloque de código.

# Ejercicio evaluado

1. Accede a la tarea que está en Moodle llamada cp4
2. Busca tu nombre y acepta la tarea
3. Clona el repositorio
4. Realiza el ejercicio que asigne el instructor/a
5. Puedes verificar en tu repositorio de GitHub los cambios que realizaste al código.
6. Borrar las credenciales de windows
7. Borrar de la computadora el o los programas creados durante la práctica