

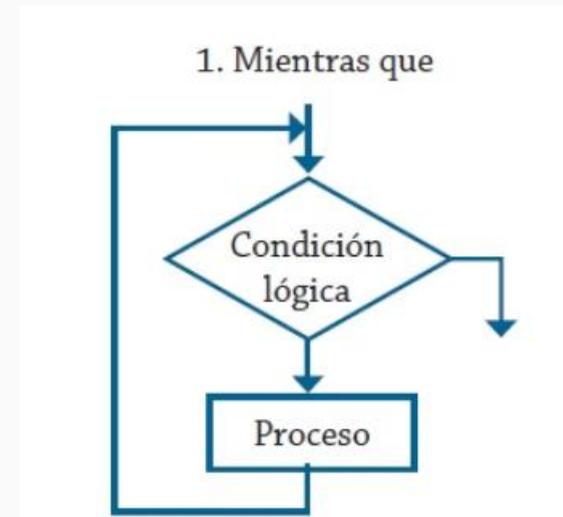
Estructuras de Control Repetitivas Condicionales

En el mundo de la programación, las estructuras de control repetitivas condicionales, también conocidas como bucles, juegan un papel fundamental. Estas herramientas nos permiten repetir una tarea o conjunto de instrucciones de manera eficiente y controlada, basándose en determinadas condiciones.

Ing. Alfredo Colcha O., Mgs.



Bucle "Mientras"

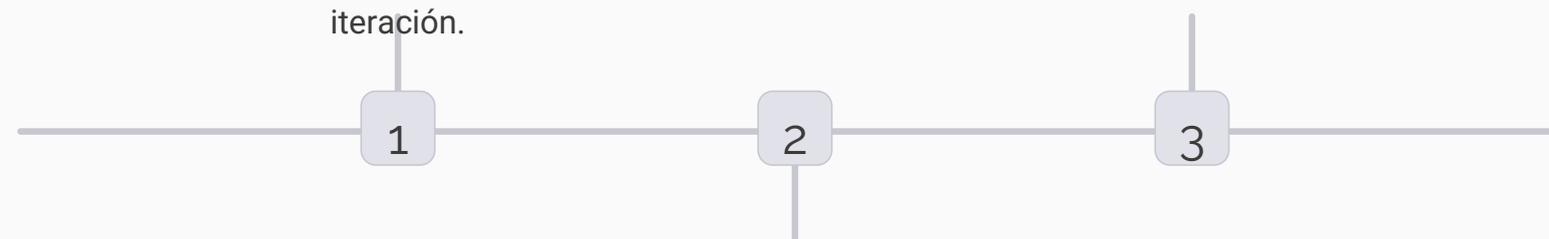


Condición

El bucle "Mientras" se ejecuta mientras la condición especificada sea verdadera. La condición se evalúa al inicio de cada iteración.

Finalización

Una vez que la condición se vuelve falsa, el bucle termina y el programa continúa con las siguientes instrucciones.



Cuerpo

El cuerpo del bucle contiene las instrucciones que se repetirán hasta que la condición ya no se cumpla.

Ejemplo Práctico del Bucle "Mientras"

1 Contar números hasta un límite

Un ejemplo común es un bucle "Mientras" que cuenta números desde 1 hasta un límite específico.

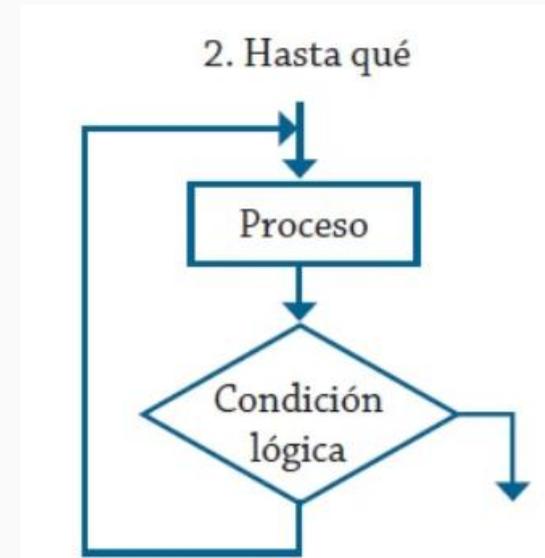
2 Validar entrada del usuario
Otro uso es verificar que la entrada del usuario cumple con ciertos requisitos antes de proceder.

3 Procesar datos iterativamente

Los bucles "Mientras" también se utilizan para trabajar con conjuntos de datos, procesándolos uno por uno.



Bucle "Hacer...Mientras"



Definición

El bucle "Hacer...Mientras" es similar al bucle "Mientras", pero con una diferencia clave: el cuerpo del bucle se ejecuta al menos una vez, independientemente de la condición.

Estructura

Primero se ejecuta el cuerpo del bucle, y luego se evalúa la condición. Si la condición es verdadera, el bucle se repite.

Aplicaciones

Este tipo de bucle es útil cuando se necesita garantizar que el cuerpo se ejecute al menos una vez, como en el caso de leer y validar la entrada del usuario.

Ejemplo Práctico del Bucle "Hacer...Mientras"

Lectura de entrada
del usuario

Un caso de uso común es
utilizar un bucle

"Hacer...Mientras" para leer
y validar la entrada del
usuario hasta que sea
correcta.

Ejecución garantizada

Al menos una vez, el
código dentro del bucle se
ejecutará, lo que lo hace
ideal para escenarios
donde se necesita una
iteración inicial.

Flexibilidad

La estructura
"Hacer...Mientras" permite
una mayor flexibilidad y
control sobre el flujo de
ejecución en comparación
con el bucle "Mientras".

Mejores Prácticas y Consejos



Evitar Bucles Infinitos

Asegúrate de que la condición del bucle eventualmente se vuelva falsa para evitar que el programa se quede atascado.



Elegir el Bucle Adecuado

Analiza cuidadosamente el problema y elige el tipo de bucle que mejor se adapte a tus necesidades.



Documentación y Claridad

Comenta tu código y mantén una estructura clara para facilitar la comprensión y el mantenimiento.

Bucles Condicionales: Resumen

1

Bucle "Mientras"

Repite el cuerpo del bucle mientras la condición sea verdadera.

2

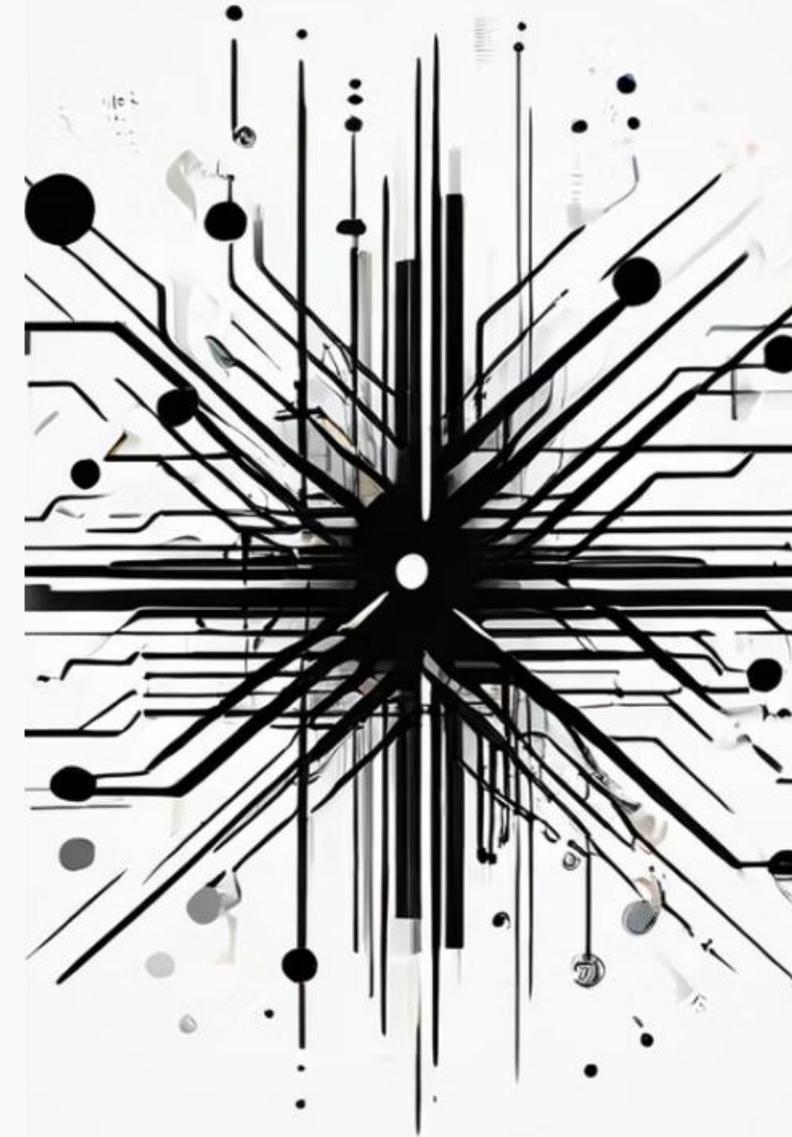
Bucle "Hacer...Mientras"

Ejecuta el cuerpo del bucle al menos una vez, y luego repite si la condición es verdadera.

3

Mejores Prácticas

Evitar bucles infinitos y elegir el tipo de bucle adecuado para cada situación.





Conclusión

Los bucles condicionales son herramientas poderosas en la programación, permitiéndonos automatizar tareas repetitivas y procesar datos de manera eficiente. Al comprender los conceptos clave de los bucles "Mientras" y "Hacer...Mientras", y aplicar las mejores prácticas, podemos crear programas más robustos y flexibles.