

Fundamentos de la Programación

¿QUÉ ES UNA ESTRUCTURA REPETITIVA ?

Durante el proceso de creación de programas, es muy común encontrarse con que una operación o conjunto de operaciones deben repetirse muchas veces. Para ello es importante conocer las estructuras de algoritmos que permiten repetir una o varias acciones, un número determinado de veces.

Las estructuras que repiten una secuencia de instrucciones un número determinado de veces se denominan bucles, y se denomina iteración al hecho de repetir la ejecución de una secuencia de acciones.

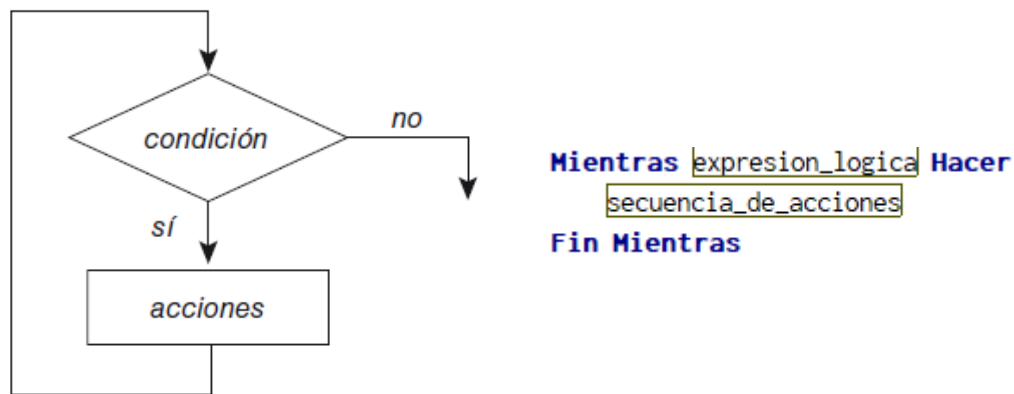
Todo bucle tiene que llevar asociada una condición, que es la que va a determinar cuándo se repite el bucle y cuando deja de repetirse.

Hay distintos tipos de bucles:

- Mientras
- Hacer Mientras // Repetir Mientras
- Para

Estructura Mientras

En la **estructura repetitiva "Mientras"**, **el cuerpo del bucle se repite siempre que una condición específica se cumpla**. Al ejecutar la instrucción "Mientras", primero se evalúa una condición (una expresión lógica). Si esta condición se evalúa como falsa, no se realiza ninguna acción y el programa continúa con la siguiente instrucción. Sin embargo, si la expresión lógica es verdadera, se ejecuta el cuerpo del bucle. Luego, se vuelve a evaluar la expresión lógica. Este proceso se repite continuamente mientras la expresión lógica (la condición) sea verdadera. Para salir del bucle, la condición debe evaluarse como falsa.



💡 **Regla Práctica:** Es recomendable utilizar pruebas de mayor o menor que en lugar de igualdad o desigualdad al realizar pruebas o tests en expresiones lógicas. Esta práctica es especialmente crucial al codificar en un lenguaje de programación, sobre todo al comparar números reales, ya que estos valores se almacenan como cantidades aproximadas y las comparaciones de igualdad de valores reales suelen plantear problemas. Por lo tanto, **siempre que realices comparaciones de números reales, es preferible utilizar las relaciones <, <=, > o >=.**

Ejemplo:

```
Algoritmo Ejemplo_Mientras
  Definir numero Como Entero

  Escribir "Ingrese un número positivo: "
  Leer numero

  Mientras numero ≤ 0 Hacer
    Escribir "El número ingresado no es positivo. Inténtelo de nuevo."
    Escribir "Ingrese un número positivo: "
    Leer numero
  Fin Mientras

  Escribir "¡Número válido! Ha ingresado un número positivo."
FinAlgoritmo
```

En este ejemplo, la estructura Mientras se utiliza para asegurarse de que el usuario ingrese un número positivo. La condición `numero ≤ 0` verifica si el número ingresado es menor o igual a cero. Mientras esta condición sea verdadera, es decir, mientras el número no sea positivo, el programa solicitará al usuario que ingrese un número positivo. Una vez que el usuario ingresa un número válido (mayor que cero), el programa muestra un mensaje indicando que se ha ingresado un número positivo y finaliza.

Estructura Hacer - Mientras // Repetir - Mientras

Esta estructura es muy similar a la anterior; sin embargo, a diferencia del bucle "Mientras", **en el bucle "Hacer-Mientras" el contenido del bucle se ejecuta al menos una vez. Esto se debe a que la evaluación de la condición lógica se realiza al final del bucle.** De esta manera, garantizamos que las acciones dentro de este bucle se lleven a cabo al menos una vez, incluso si la expresión lógica es falsa.

Repetir

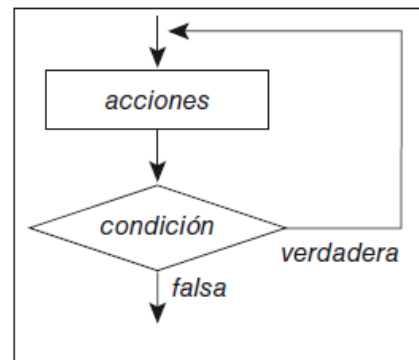
secuencia_de_acciones


Mientras Que expresion_logica

Hacer

secuencia_de_acciones

Mientras Que expresion_logica



 **Regla Práctica:** El bucle Hacer-Mientras finaliza su ejecución cuando la condición se evalúa como falsa. **La decisión entre utilizar un bucle Mientras o un bucle Hacer-Mientras depende del problema específico a resolver.** En la mayoría de los casos, el bucle Mientras es la opción más adecuada. Por ejemplo, si el bucle se utiliza para recorrer una lista de números (o cualquier otro tipo de objetos), y esta lista está vacía, las instrucciones del bucle nunca se ejecutarán. En contraste, la aplicación de un bucle Hacer-Mientras en esta situación podría generar un código propenso a errores.

Ejemplo:

```
Algoritmo Ejemplo_RepetirMientras
  Definir numeroIngresado Como Entero

  Repetir
    Escribir "Ingrese un número positivo:"
    Leer numeroIngresado
  Mientras Que numeroIngresado ≤ 0
FinAlgoritmo
```

```
Algoritmo Ejemplo_HacerrMientras
  Definir numeroIngresado Como Entero

  Hacer
    Escribir "Ingrese un número positivo:"
    Leer numeroIngresado
  Mientras Que numeroIngresado ≤ 0
FinAlgoritmo
```

Tanto 'Hacer' como 'Repetir' son instrucciones válidas para esta estructura. La priorización entre ambas depende de la configuración del perfil del usuario.

En este ejemplo, el bucle Hacer-Mientras se utiliza para solicitar al usuario que ingrese un número positivo. Independientemente del valor inicial de numero, el bucle se ejecutará al menos una vez porque la condición ($\text{numero} \leq 0$) se evalúa después de la ejecución del bloque de código dentro del bucle.