

PROGRAMACIÓN DESDE CERO

MATERIAL DE TRABAJO

Estructuras de control con PseInt – Estructuras repetitivas

ENCUENTRO 7



Universidad de
LA PUNTA



GOBIERNO DE
SAN LUIS



EGG



Objetivos de la Guía

En esta guía aprenderemos a:

- Armar estructuras repetitivas.
- Usar estructuras repetitivas.

GUÍA ESTRUCTURAS REPETITIVAS

Vamos a continuar con las estructuras de control, pero ahora vamos a presentar un nuevo tipo de estructura. En los encuentros anteriores, nos centramos en las estructuras selectivas, en estos próximos encuentros veremos **las estructuras repetitivas**.

¿QUÉ SON LAS ESTRUCTURAS REPETITIVAS?

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.

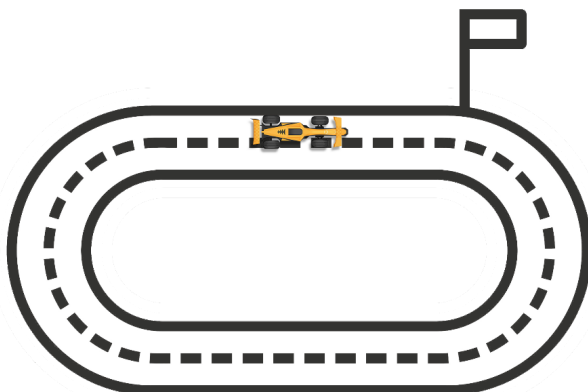
Todo bucle tiene que llevar asociada una condición, que es la que va a determinar cuándo se repite el bucle y cuando deja de repetirse.

Hay distintos tipos de bucles:

- Mientras
- Hacer Mientras
- Para



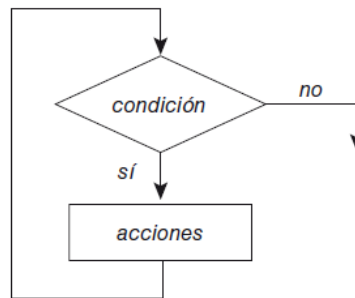
¿NECESITAS UN EJEMPLO?



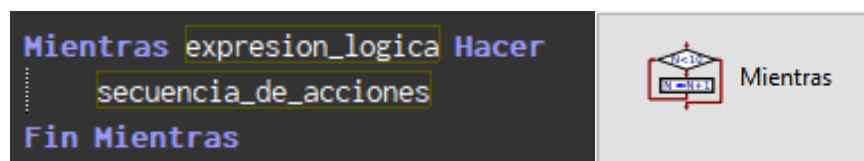
Este ejemplo nos muestra el concepto de bucle. Este auto de F1 debe dar vueltas a la pista hasta que la cantidad de vueltas realizadas sea igual a la cantidad total de vueltas que exige la carrera. Por lo tanto, debe repetir la vuelta tantas veces hasta que termine la carrera.

ESTRUCTURA MIENTRAS

Esta estructura repetitiva **Mientras**, es en la que el cuerpo del bucle se repite siempre que se cumpla una determinada *condición*. Cuando se ejecuta la instrucción *mientras*, la primera cosa que sucede es que se evalúa la condición (una expresión lógica). Si se evalúa *falsa*, no se toma ninguna acción y el programa prosigue con la siguiente instrucción. Si la expresión lógica es *verdadera*, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión lógica. Este proceso se repite una y otra vez mientras la expresión lógica (condición) sea verdadera, para salir del bucle la condición debe ser falsa.



Estructura Mientras en PSeInt



Regla práctica

En las expresiones lógicas es conveniente usar comparaciones mayor o menor en lugar de comparaciones de igualdad o desigualdad. En el caso de la codificación en un lenguaje de programación, esta regla debe seguirse rígidamente en el caso de comparación de números reales, ya que como esos valores se almacenan en cantidades aproximadas las comparaciones de igualdad de valores reales normalmente plantean problemas. Siempre que realice comparaciones de números reales use las relaciones $<$, $<=$, $>$ o $>=$.



```

1  Algoritmo EjemploMientras
2
3      Definir nota Como Entero
4
5      Escribir "Ingrese una nota válida"
6      Leer nota
7
8      Mientras nota < 0 o nota > 10
9
10         Escribir "Ingrese la nota nuevamente"
11         Leer nota
12
13     FinMientras
14
15     Escribir "La nota es correcta"
16
17 FinAlgoritmo
18
  
```

En este bucle analizamos si la nota ingresada **no** es válida, es decir si **no** está en el rango entre 0 y 10. Si la nota se sale del rango la condición será verdadera y se ejecutará el código dentro del bucle. Si la condición es falsa, no se ejecuta el código dentro del bucle y va directo a la línea 15.



Pueden encontrar un ejemplo para descargar del Bucle Mientras en Aula Virtual.



MANOS A LA OBRA!

EJERCICIO VOCAL SECRETA

Diseña un programa que guarde una vocal secreta en una variable, debemos pedirle al usuario que intente adivinar la vocal secreta, e intentará tantas veces como sea necesario hasta que la adivine.

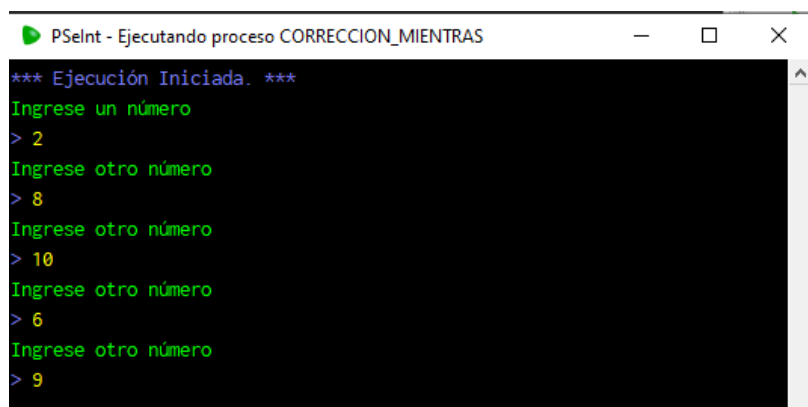
DETECCIÓN DE ERRORES

Copia y pega este código en tu programa. Deberás corregir los errores hasta lograr el siguiente resultado esperado:

```
Algoritmo Correccion_Mientras
Definir num Como Entero
//El programa ingresará números mientras sean PARES
Escribir "Ingrese un número"
Leer num
Mientras num % 2 == 0 Hacer
    Escribir "Ingrese otro número"

FinAlgoritmo
```

¿Cuál es el resultado a lograr?



```
PSeInt - Ejecutando proceso CORRECCION_MIENTRAS

*** Ejecución Iniciada. ***
Ingrese un número
> 2
Ingrese otro número
> 8
Ingrese otro número
> 10
Ingrese otro número
> 6
Ingrese otro número
> 9
```



Revisemos lo aprendido hasta aquí

Definir, implementar y diferenciar la estructura **MIENTRAS**. Sabiendo que es la estructura que **PRIMERO** valida la condición y luego ejecuta el código repetidamente **MIENTRAS** la condición sea verdadera.