

# PROGRAMACIÓN

# Guía de estudio 5:

# 80 veces la vuelta al mundo

## Estructuras cíclicas / iterativas.

## While, Do-While y for

**Nivel:** Secundario - Modalidad Educación Técnico-Profesional.

**Ciclo:** Segundo ciclo.

**Especialidades:** Informática, Programación.

### Introducción

En esta guía veremos las estructuras cíclicas/iterativas. Como les debe dar a entender su nombre, son estructuras que se repiten. Pero... ¿cómo las creamos? ¿cuántas veces se repiten? Todo esto lo veremos en las siguientes páginas

**¿Qué estamos aprendiendo?** Qué son las estructuras cíclicas/iterativas, tipos de estructuras y diferencias entre ellas.

### Recursos

Video nro 12 del curso:

<https://youtu.be/3VKdy0Xuc9M>



|  |   |
|--|---|
| Video nro 13 del curso:<br><a href="https://youtu.be/B8nbCUTMJ0w">https://youtu.be/B8nbCUTMJ0w</a> |  |
| Video nro 14 del curso:<br><a href="https://youtu.be/FE-vf4hws8">https://youtu.be/FE-vf4hws8</a>   |  |

## Secuencia didáctica

Con los videos de esta guía podremos sumergirnos un poco más en las estructuras más usadas dentro de los lenguajes de programación: los ciclos de repetición. Cada uno de las formas de hacer ciclos es un poquito diferente. Suelen ser reemplazables (en el sentido que se puede lograr el mismo comportamiento usando dos estructuras diferentes), pero te recomendamos que entiendas bien todas para que, cuando veas el código, sepas interpretar cuál es la que más conviene usar según el caso.

|   |   |
|---|---|
| <a href="https://youtu.be/3VKdy0Xuc9M">https://youtu.be/3VKdy0Xuc9M</a> |  |
| <a href="https://youtu.be/B8nbCUTMJ0w">https://youtu.be/B8nbCUTMJ0w</a> |  |
| <a href="https://youtu.be/FE-vf4hws8">https://youtu.be/FE-vf4hws8</a>   |  |

## Actividades

Después de mirar el video del link o del QR, te invitamos a que reflexiones con las siguientes preguntas.

No olvides escribir tus respuestas, te facilitarán realizar trabajos prácticos y repasar los temas que hayas estudiado con esta guía.

- 1) Hacé un programa que realice verificación de identidad. El programa deberá pedir una clave al usuario y este tendrá tres oportunidades para poner la clave correcta.
  - a) Si el usuario pone la clave incorrecta 3 veces, se deberá imprimir "Acceso denegado".
  - b) Si puso la clave mal, pero todavía le quedan oportunidades, imprimir "Vuelva a ingresar la clave".
  - c) Si se ingresó la clave correcta, imprimir "Acceso otorgado".

Consideremos que la clave es "MeGustaProgramar".

- 2) Hacer un programa que haga rectángulos de ancho X y alto Y. Los valores de X e Y serán ingresados por el usuario, y se deberá imprimir el rectángulo (formado por símbolos "\*" en pantalla. Por ejemplo, si se ingresan X=7 e Y=3, en la pantalla debería escribirse:

```
*****  
*****  
*****
```

¡Advertencia! Tené en cuenta que los números enteros también pueden ser negativos. En ese caso, debería imprimir un mensaje de error: "No se pueden tener unidades negativas".

**Pista:** Tené en cuenta que dentro de un ciclo ¡puede haber otro ciclo!

- 3) Desarrollá un programa en el cual se pida ingresar un texto letra por letra, hasta que se ingrese el símbolo "-". En ese momento se deberá imprimir la palabra completa. En este caso, ¿conviene un while, do-while o un for?
- 4) Creá un programa que haga lo siguiente:
  - a) Deberá pedir números al usuario hasta que se ingrese "0".
  - b) Mostrar el resultado de la suma de todos los números ingresados
  - c) Preguntar si se quiere volver a ejecutar el código. Si la respuesta es "sí", se deberá volver al ítem a), si la respuesta es "no", terminar el programa.
- 5) El usuario deberá ingresar 5 números. Luego de eso se deberá indicar:
  - a) Cuántos números son positivos.
  - b) Cuántos números son negativos.

- c) Cuántos números son pares.
  - d) Cuantos "0" se ingresaron.
- 6) Hacer un programa el cual se le deban ingresar frases hasta que se ingrese un "\*". Luego, mostrar la frase más larga de las que se hayan ingresado.

*Nota:* se puede saber el largo de un string agregando un "length" después de la variable. Por ejemplo:

```
string hola = "hola";  
int largo = hola.Length;  
Console.WriteLine(largo);  
Console.ReadKey();
```

- 7) Hace un código en el cual se pida ingresar un número y luego se identifique si ese número es primo o no (¡buscar en Internet cuándo se considera primo a un número!).

## Para seguir aprendiendo...

¡Hagamos un juego! Tal vez algunos lo conozcan, se llama "mayor o menor" y sus reglas son bastante fáciles.

Se juega con una baraja española (no nos importan los palos, pero sí nos importa que son 12 números). Hay un "repartidor" (en este caso el programa) que va tirando de a una carta. Por cada carta que tira, se le pregunta al jugador si cree que la próxima carta va a ser un número mayor o menor al que se acaba de mostrar. Si acierta, se revela una nueva carta, si no, termina el juego.

*Extra:* Para realizar este juego de azar, ¡necesitamos azar! por eso es que te damos este código para que lo pruebes y lo uses... (Dale varias veces a iniciar para entender mejor).

```
Random rnd = new Random();  
  
int number = 0;  
number = rnd.Next(5);  
Console.WriteLine(number);  
Console.ReadKey();
```

¿Qué hace rnd.Next(N)? ¿Qué valor devuelve? ¿Entre qué rango?