



## INTRODUCCION A C#

### ESTRUCTURA DE DATOS

- 1- Muestra los números del 1 al 100 (ambos incluidos) divisibles entre 2 y 5. Utiliza las iteraciones `for`, `while`, `do/while` para resolver el mismo problema.
- 2- Codifique un programa de consola en C# que permita realizar las siguientes acciones:

Generar un número aleatorio entre 0 y 100, para ello use la función **random** de C#

Una vez generado el número codifique la lógica necesaria para encontrar el número aleatorio ayudando al usuario informando al mismo si el número ingresado es mayor o menor al número aleatorio buscado. Una vez encontrado el número muestre la cantidad de intentos necesarios para lograrlo.

#### Ejemplo:

##### Número aleatorio generado: 63

Ingrese un número entre 0 y 100.

Numero Ingresado: 50

Respuesta: Es muy bajo

Ingrese un número entre 0 y 100.

Numero Ingresado: 75

Respuesta: Es muy alto

Ingrese un número entre 0 y 100.

Numero Ingresado: 60

Respuesta: Es muy bajo

Ingrese un número entre 0 y 100.

Numero Ingresado: 65

Respuesta: Es muy alto

Ingrese un número entre 0 y 100.

Numero Ingresado: 63

Respuesta: Correcto, numero encontrado, cantidad de intentos 5

## METODOS / FUNCIONES

- 3- Codifique un método denominado `esPar` que reciba como parámetro un valor numérico entero y retorne `true` en caso de que el número sea par, caso contrario retorne `false`. Compruebe el funcionamiento del método solicitando al usuario del programa que ingrese números aleatorios.

- 4- Codifique 5 funciones C# que resuelvan las operaciones matemáticas:

Suma +  
Resta –  
Multiplicación \*  
División /  
Modulo %

Aplicadas sobre 2 números decimales.

Posteriormente codifique un programa que solicite el ingreso por parte del usuario de 2 números decimales y una vez ingresados le pregunte qué operación quiere realizar:

a- Suma +  
b- Resta –  
c- Multiplicación \*  
d- División /  
e- Modulo %

En caso de que no ingrese ninguna de las opciones permitidas emita el mensaje “La opción ingresada no es válida” y vuelva nuevamente a preguntarle qué operación matemática quiere ejecutar.

Si la opción es válida envíe los números decimales ingresados por el usuario a la función que corresponda y retorne el resultado de la operación.

Finalmente muestre un mensaje indicando el resultado de la operación realizada.

- 5- Cree las funciones necesarias para calcular el sueldo de los trabajadores dependiendo las horas trabajadas. El valor de la hora en horario diurnos (8:00 a 20:00) es de \$10/hora, si el horario es nocturno (20:00 a 8:00) se incrementa en un 50%. El programa debe pedir al usuario en qué horario trabajo el empleado (hora ingreso, minuto ingreso, hora salida, minuto salida), validar que la hora/minuto de ingreso no sea mayor a la hora/minuto de salida, validar que la cantidad de horas trabajadas no supere las 8 horas, calcular cuántas horas trabajo en total y cuánto corresponde pagarle. Nota: Se recomienda calcular el tiempo laboral en minutos.

## RECURSION

- 6- Programar un algoritmo recursivo que en cada recursión multiplique a si mismo un número ingresado por el usuario hasta que el valor resultante sea mayor

100.000.000. Valide que el número ingresado por el usuario sea mayor a 1, ejemplo:

Se ingresa el valor 6

1 iteración  $\Rightarrow 6 * 6 = 36$

2 iteración  $\Rightarrow 36 * 36 = 1296$

3 iteración  $\Rightarrow 1296 * 1296 = 1679616$

4 iteración  $\Rightarrow 1679616 * 1679616 = 2821109907456$

**Fin Recursión**

## CADENAS

- 7- Explique y ejemplifique los métodos para el manejo de cadenas en C#, Length, Contains, IndexOf, Remove, Replace, Substring, Append, ToUpper, ToLower.