

Taller de Programación Básica en C#

Fundamentos para Ingeniería en Sistemas Computacionales

Instructor: Ing. Alejandro Guzmán Rodríguez



Agenda del Taller (4 Horas)

Bloque I: Fundamentos y Repaso



Entorno (IDE)

Preparación del entorno de desarrollo para las prácticas interactivas.



Variables y Tipos

Repaso de int, string, bool, float y cómo almacenar datos de forma eficiente.



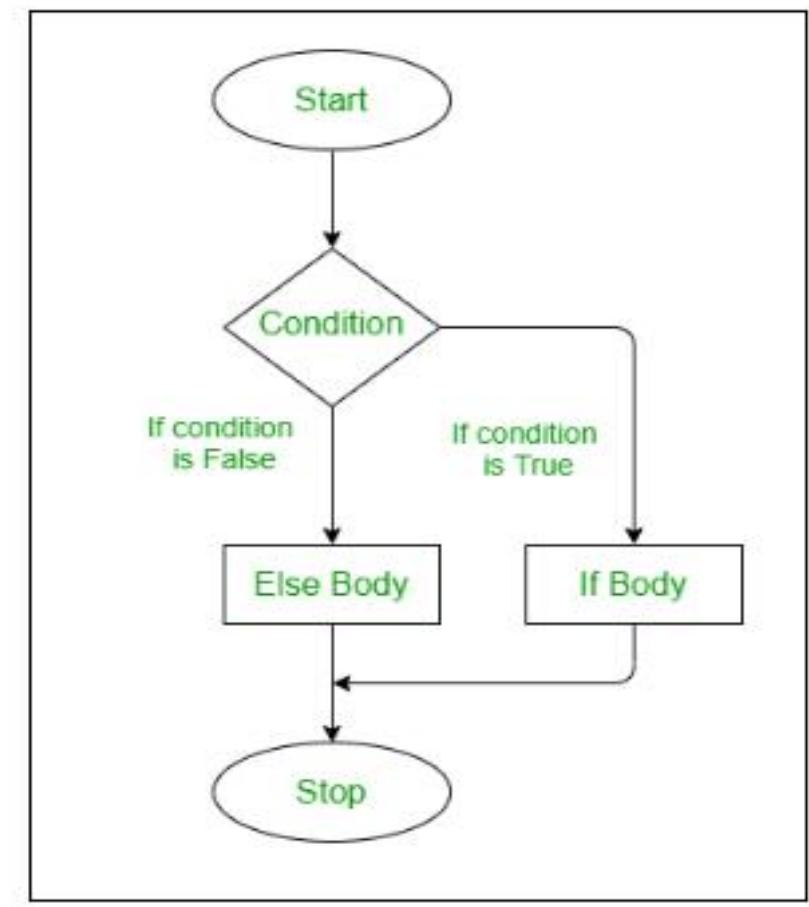
Operadores

Uso de operadores aritméticos (+, -, *, /) y lógicos (&&, ||, !).

Bloque II: Estructuras de Control

Condicionales: Tomando Decisiones

- Permiten que tu código reaccione a diferentes situaciones.
- Usamos **if**, **else if**, y **else** para definir caminos lógicos.
- Son la base fundamental de la lógica de negocio y algoritmos complejos.
- Evaluamos condiciones que resultan en **true** o **false**.



Bloque II: Estructuras de Control

- ⌚ Ciclo `for`: Ideal para repetir una tarea un número **conocido** de veces. (Ej: "procesar los 10 primeros clientes").
- ♾️ Ciclo `while`: Perfecto para repetir **mientras** una condición sea verdadera. (Ej: "mientras el usuario no presione 'salir'").
- 🛠️ **Automatización:** Los ciclos nos permiten procesar colecciones de datos (como arreglos) y automatizar tareas repetitivas.

Ejercicio Práctico 1

Concepto a Aplicar

Uso de variables para capturar entradas, operadores aritméticos para calcular, y condicionales (if/else) para aplicar la lógica de negocio.

Problema: Calculadora de Descuentos

Solicita un precio y un código.

Aplica 10% de descuento si el código es "INGE10".

Aplica 20% si es "PROG20".

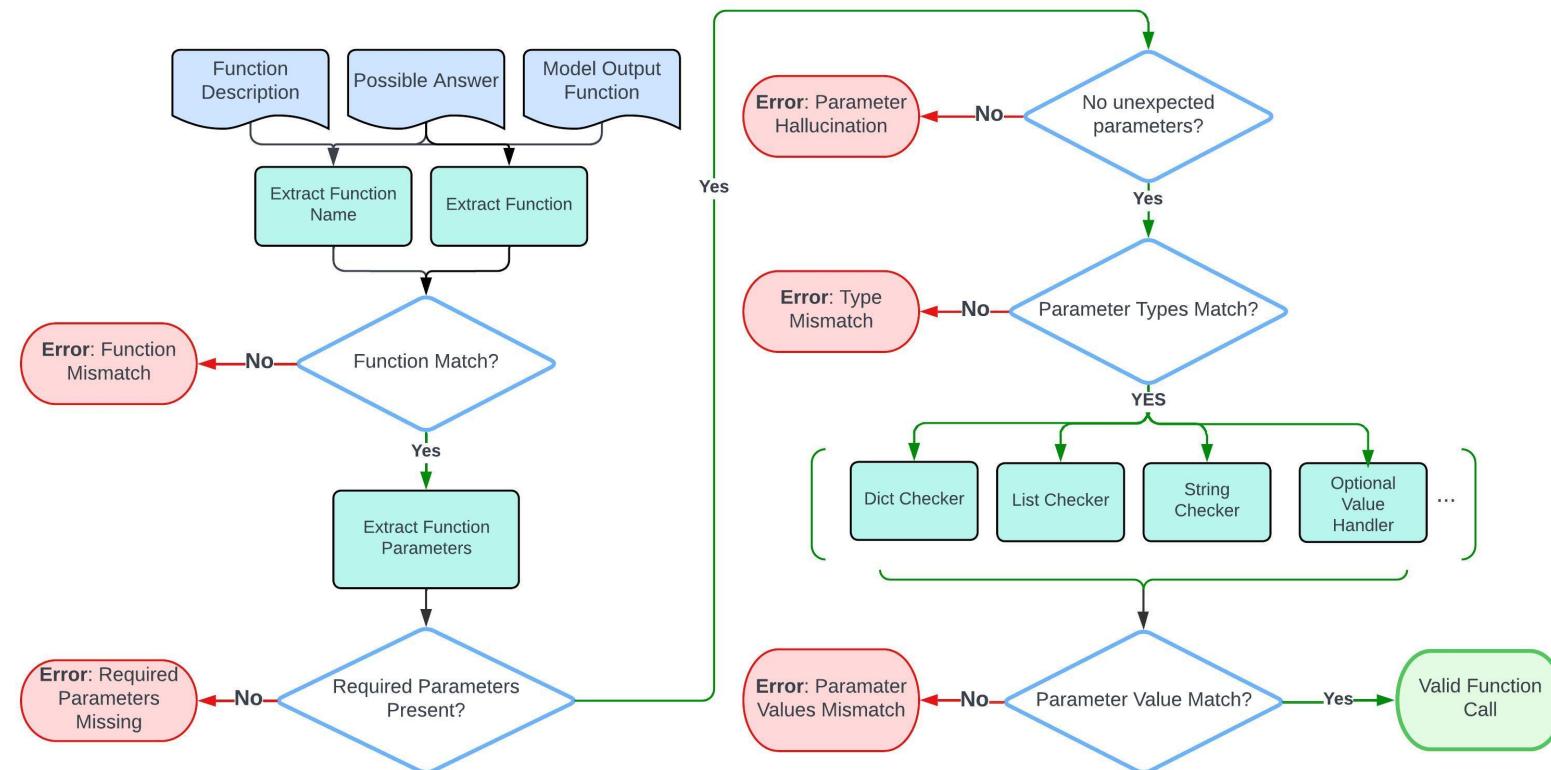
Muestra el precio final.

Bloque IV: Modularidad y Funciones

Divide y Vencerás

Las funciones agrupan código en bloques lógicos y reutilizables. Esto hace que nuestro programa sea más limpio y fácil de mantener.

Reciben **parámetros** (datos de entrada) y pueden **retornar** un valor (el resultado).



Ejercicio Práctico 2

Concepto a Aplicar

Creación de una función que recibe un parámetro y retorna un valor (booleano). Uso de un ciclo `for` para iterar y llamar a la función.

Problema: Generador de Secuencias

Crear una función `esPar(numero)` que retorne `true` o `false`. El programa principal debe pedir un número N. Mostrar los números de 1 a N, indicando "Par" o "Impar" usando la función.

Bloque VI: Estructuras de Datos Simples



Arreglos (Arrays)

Colecciones de tamaño fijo. Permiten almacenar múltiples valores del mismo tipo. Acceso rápido por índice (ej: `notas[0]`).



Listas (Lists)

Colecciones dinámicas (su tamaño puede cambiar). Permiten agregar o quitar elementos fácilmente durante la ejecución.



Recorrido

Usamos ciclos (como `for` o `foreach`) para iterar y procesar cada elemento guardado dentro de un arreglo o lista.

Ejercicio Práctico 3 (Integrador)

-  Almacenar 5 calificaciones (ej: 80, 95, 72, 100, 68) en un **arreglo**.
-  Crear una función para calcular y retornar el **promedio** del arreglo.
-  Crear una función para encontrar y retornar la **nota más alta**.
-  Imprimir el promedio, la nota alta, y el estado de cada alumno ('Aprobado' \geq 70 / 'Reprobado').

Cierre y Siguientes Pasos

Siguientes Pasos

Con estos fundamentos, el siguiente paso lógico es explorar:

- Programación Orientada a Objetos (POO)
- Estructuras de Datos Avanzadas

¿Preguntas?

Gracias por participar en el taller.

¡Sigan practicando!



alexguroding : <https://github.com/alexguroding/taller-Programacion-basica>



(276) 106 3341



alexguroding@gmail.com