Segunda Parte de la Clase

Repaso General de C#

Conceptos Básicos

1. C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft.

2. Se utiliza para crear aplicaciones de escritorio, web, móviles y videojuegos.

Variables en C#

Definición: Una variable es un espacio en la memoria que almacena datos.

Ejemplo cotidiano:
Una **variable** es como una caja donde guardamos algo y le ponemos una etiqueta.

```
int edad = 25;
string nombre = "Carlos";
double precio = 9.99;
bool esProgramador = true;
```

Mini reto

Crea variables para almacenar:
Tu nombre
Tu edad

Tipos de Datos en C#

Definición: Especifican qué tipo de información puede almacenar una variable.

Ejemplo cotidiano:
Si tienen un estante con cajas de diferentes tamaños, cada tipo de caja solo puede guardar ciertos objetos (dinero, ropa, documentos).

```
int numero = 10;       // Número entero
double decimal = 5.75;  // Número con decimales
char letra = 'A';      // Un solo carácter
string texto = "Hola";  // Texto
bool esVerdadero = true; // Booleano (true o false)
```

Mini reto

Declara tres variables:
1. Un número entero
2. Un texto
3. Un valor booleano

Operadores en C#

Definición: Los operadores realizan operaciones matemáticas o lógicas.

Ejemplo cotidiano:
Los operadores son como las herramientas que usamos para sumar, comparar o tomar decisiones.

```
int suma = 10 + 5;  // Operador de suma
int resta = 10 - 5; // Operador de resta
bool comparacion = (10 > 5); // Comparación (true)
```

Mini reto

Calcula el área de un rectángulo con dos variables: **base** y **altura**.

Estructuras de Control en C#

Definición: Permiten tomar decisiones y repetir acciones.

Ejemplo cotidiano:
Si llueve, llevamos paraguas; si no llueve, no lo llevamos.

```
int edad = 18;

if (edad >= 18) {
   Console.WriteLine("Eres mayor de edad.");
} else {
   Console.WriteLine("Eres menor de edad.");
}

//Ejemplo de bucle
for (int i = 1; i <= 5; i++) {
   Console.WriteLine("Número: " + i);
}
```

Mini reto

Crea un **if** que verifique si un número es par o impar.

Métodos (Funciones) en C#

Definición: Un método es un bloque de código que realiza una tarea específica.

Ejemplo cotidiano:
Un método es como una receta de cocina: sigues los pasos y obtienes un resultado.

```
void Saludar() {
   Console.WriteLine("Hola, bienvenido!");
}

// Llamar al método
Saludar();
```

Mini reto

Crea un método que reciba tu nombre y lo imprima en pantalla.

Entrada y Salida de Datos en C#

Definición: Permiten interactuar con el usuario.

Ejemplo cotidiano:
Es como cuando alguien te pregunta tu nombre y tú respondes.

```
Console.WriteLine("¿Cómo te llamas?");
string nombre = Console.ReadLine();
Console.WriteLine("Hola, " + nombre);
```

Mini reto

Pide al usuario su edad y dile cuántos años tendrá en 10 años.

Programación Orientada a Objetos (POO) en C#

Clases y Objetos en C#

Definición: Una **clase** es un modelo y un **objeto** es una instancia de esa clase.

Ejemplo cotidiano:
Una **clase** es como un molde para hacer galletas, y
cada **objeto** es una galleta diferente hecha con ese molde.

Ejemplo

```
class Persona {
    public string Nombre;
    public int Edad;

    public void Saludar() {
        Console.WriteLine("Hola, mi nombre es " + Nombre);
    }
}

// Crear un objeto
Persona persona1 = new Persona();
persona1.Nombre = "Carlos";
persona1.Edad = 30;
persona1.Saludar();
```

Mini reto

Crea una clase **Persona** con las propiedades **nombre** y **edad**.
Luego crea un objeto e imprime sus valores.

Encapsulamiento en C#

Definición: Protege los datos dentro de una clase.

Ejemplo cotidiano:
Una **caja fuerte** que solo puede abrirse con la clave correcta.

```
class Banco {
   private double saldo = 1000;

   public void MostrarSaldo() {
      Console.WriteLine("Tu saldo es: " + saldo);
   }
}
```

Mini reto

Crea una clase **Coche** con una propiedad **velocidad** privada y un método público para establecer su valor.

Herencia en C#

Definición: Una clase puede heredar atributos y métodos de otra.

Ejemplo cotidiano:
Un **perro** hereda características de un **animal**.

```
class Animal {
   public void HacerSonido() {
      Console.WriteLine("Hace un sonido...");
   }
}

class Perro : Animal {
   public void Ladrar() {
      Console.WriteLine("¡Guau Guau!");
   }
}
```

Mini reto

Crea una clase **Animal** con un método **HacerSonido()** y una clase **Perro** que herede de **Animal**.

Polimorfismo en C#

Definición: Un mismo método con diferentes comportamientos.

Ejemplo cotidiano:
Un **pájaro** y un **avión** vuelan, pero de manera diferente.

```
class Ave {
   public virtual void Volar() {
      Console.WriteLine("El ave vuela.");
   }
}

class Aguila : Ave {
   public override void Volar() {
      Console.WriteLine("El águila planea.");
   }
}
```

Mini reto

Crea una clase **Figura** con un método **CalcularArea()**, y dos clases **Cuadrado** y **Círculo** que lo implementen.

Abstracción en C#

Definición: Oculta los detalles y muestra solo lo necesario.

Ejemplo cotidiano:
Cuando manejan un carro, solo usas el volante y los pedales sin ver el motor.

```
abstract class Animal {
   public abstract void HacerSonido();
}

class Perro : Animal {
   public override void HacerSonido() {
      Console.WriteLine(”Sonido de ladrido!");
   }
}
```

Mini reto

Crea una clase **abstracta** llamada Figura con un método **abstracto** CalcularArea(), y luego crea dos clases que la hereden:

1. **Cuadrado** (con atributo lado).
2. **Círculo** (con atributo radio).

Cada una debe implementar el método CalcularArea() según su fórmula matemática.

Cual fue el objetivo de esta sesion?

Reforzar los conceptos basicos vistos en la sesion anterior pero reforzar los temas de **dynamic, object, herencia y polimorfismo**

Existen diferencias entre "dynamic" y "object”?

Respuesta corta

SI!

Pero cuales son las diferencias?

dynamic: Puede cambiar de tipo en tiempo de ejecución.

object: Puede almacenar cualquier tipo, pero necesita conversión para usarse.

Y como lo aplicariamos en la vida real?

Ejemplo cotidiano
dynamic: Es como una caja mágica donde metes cualquier cosa y cambia de forma según lo que pongas.

object: Es como una caja cerrada donde puedes guardar cualquier cosa, pero para usar su contenido, primero debes abrirla y verificar qué hay adentro.

Mini reto

Crea una variable dynamic, almacena un número y luego cámbiala a un texto.

Usa object para almacenar un número decimal y conviértelo a double antes de imprimirlo.