

Tema 5: Programación Orientada a Objetos (POO) con C#.

Introducción a la Programación Orientada a Objetos.

La Programación Orientada a Objetos (POO) es un paradigma de programación que utiliza objetos para estructurar el código. Un objeto es una instancia de una clase y puede contener atributos (propiedades) y métodos (funciones).

Clases y Objetos.

Una clase es una plantilla para crear objetos, define los atributos y métodos comunes a todos los objetos de ese tipo. Un objeto es una instancia específica de una clase.

Encapsulamiento.

El encapsulamiento es el concepto de agrupar los datos (atributos) y los métodos que operan sobre esos datos en una sola unidad llamada clase. Se utiliza para ocultar la implementación interna de un objeto y exponer solo lo necesario.

Herencia.

La herencia permite que una clase herede propiedades y métodos de otra clase. La clase que hereda se llama clase derivada, y la clase de la que hereda se llama clase base o padre.

Polimorfismo.

El polimorfismo permite que un objeto se comporte de múltiples formas. Puede referirse a diferentes tipos de objetos a través de una interfaz común.

Ejemplos:

Ejemplo 1: Definición de una Clase

```
```csharp
public class Persona
{
 // Atributos
 public string Nombre { get; set; }
 public int Edad { get; set; }

 // Método
 public void Presentarse()
 {
 Console.WriteLine($"Hola, mi nombre es {Nombre} y tengo {Edad} años.");
 }
}
```
```

Ejemplo 2: Uso de Herencia.

```
```csharp
public class Estudiante : Persona
{
 // Nuevo atributo
 public string Curso { get; set; }

 // Nuevo método
}
```

```
public void Estudiar()
{
 Console.WriteLine($"Estudiando en el curso de {Curso}.");
}
}
...
```

## Actividad Práctica

### Desarrollo de Aplicación de Gestión de Empleados.

1. Define una clase `Empleado` con atributos como nombre, salario, y método para calcular el salario anual.
2. Crea una clase `Gerente` que herede de `Empleado` y tenga un nuevo atributo para el departamento que supervisa.
3. Implementa el polimorfismo creando una interfaz `IMostrarInformacion` que tenga un método para mostrar información.

## Resolución de la Actividad Práctica

### Pasos para Desarrollar la Aplicación

1. Define la clase **Empleado** con sus atributos y método.
2. Crea la clase **Gerente** que herede de **Empleado** y agregue su propio atributo.
3. Implementa la interfaz **IMostrarInformacion** en ambas clases.
4. Crea instancias de **Empleado** y **Gerente** y muestra su información utilizando el polimorfismo.

### ¿Qué Aprendimos?

En este tema, aprendimos los conceptos fundamentales de la Programación Orientada a Objetos (POO) en C#, incluyendo la definición de clases, el uso de herencia,

encapsulamiento y polimorfismo. Desarrollamos una aplicación práctica de gestión de empleados aplicando estos conceptos, sentando las bases para la creación de sistemas más complejos y modularizados.

