



Clase práctica:

### **Creación de una clase sencilla**

- Crear una clase "Persona" con las propiedades "nombre", "edad" y "email".
- Agregar un método para imprimir los datos de la persona en pantalla.

### **Herencia**

- Crear una clase "Empleado" que herede de "Persona".
- Agregar la propiedad "puesto" a la clase "Empleado".
- Agregar un método para imprimir los datos del empleado en pantalla.

### **Polimorfismo e interfaces**

- Crear una interfaz "iVehiculo" con los métodos "acelerar" y "frenar".
- Crear una clase "Automovil" que implemente la interfaz "iVehiculo".
- Crear una clase "Bicicleta" que también implemente la interfaz "iVehiculo".
- Agregar métodos para imprimir los datos del automóvil y la bicicleta en pantalla.

### **Abstracción y encapsulamiento**

- Crear una clase "CuentaBancaria" con las propiedades "saldo" y "numeroCuenta".
- Agregar métodos para depositar y retirar dinero de la cuenta bancaria.
- Utilizar setters y getters para acceder a las propiedades "saldo" y "numeroCuenta" de forma segura.

### **Creación de una clase abstracta**

- Crear una clase abstracta "FiguraGeometrica" con el método abstracto "calcularArea".
- Crear dos clases "Cuadrado" y "Triángulo" que hereden de "FiguraGeometrica".
- Implementar el método "calcularArea" en cada una de las clases hijas.

### **Creación de una clase Singleton**

- Crear una clase "ConexionDB" que utilice el patrón Singleton para mantener una única conexión a una base de datos.
- Implementar un método "getConexion" que permita acceder a la conexión en cualquier parte del código.

### **Creación de una clase Factory**

- Crear una clase "Mascota" con las propiedades "nombre" y "edad".
- Crear una clase "Perro" que herede de "Mascota" y tenga una propiedad adicional "raza".
- Crear una clase "Gato" que herede de "Mascota" y tenga una propiedad adicional "pelaje".





- Crear una clase "MascotaFactory" con el método "crearMascota" que reciba como parámetro una cadena que indique la especie de la mascota ("perro" o "gato") y devuelva un objeto del tipo correspondiente.

### **Creación de una clase Trait**

- Crear una clase "Color" con la propiedad "color".
- Crear dos clases "Automovil" y "Bicicleta" que utilicen el Trait "Color".
- Agregar métodos para imprimir los datos del automóvil y la bicicleta en pantalla, incluyendo el color.

