## **S06.s2 - Polimorfismo. Diseño del diagrama de clases del UML usando polimorfismo**

## **¿Qué es el polimorfismo en programación orientada a objetos?**

* El **polimorfismo** significa **“muchas formas”**.  
   En POO se refiere a la capacidad de un objeto de una clase derivada de **responder de manera diferente a un mismo mensaje o método**.  
    
  **2. ¿Cuál es la ventaja principal del polimorfismo?**

La **gran ventaja** es la **flexibilidad y reutilización del código**:

* Permite trabajar con **referencias genéricas** (de una superclase o interfaz) y que, en tiempo de ejecución, se invoque la implementación correcta de la subclase.
* Facilita la **extensibilidad**: podemos añadir nuevas clases sin modificar el código existente.
* Hace posible el **polimorfismo de comportamiento**, clave en patrones de diseño y frameworks.

## **Pasos para diseñar un diagrama UML que ilustre polimorfismo:**

### **1. Define la superclase**

* Coloca el nombre de la clase abstracta o interfaz arriba.
* En la segunda sección van atributos (si los tiene).
* En la tercera sección van métodos, indicando cuáles son abstractos.

**2. Crea las subclases concretas**

* Cada una **hereda** de la superclase y **sobrescribe** sus métodos.

### **3. Representa la relación de herencia**

En UML se dibuja con una **flecha de línea continua con triángulo vacío** desde la subclase hacia la superclase.

Resumen:

* **Ventaja clave:** manejar múltiples objetos distintos con un mismo enfoque.
* **Flexibilidad/extensibilidad:** añadir nuevas clases sin modificar lo existente.
* **Ejemplos cotidianos:** transporte, instrumentos, métodos de pago, electrodomésticos.