

# Relaciones en POO

## JAVA

### Agregación

La agregación es un tipo de asociación más específica. Representa una relación "todo-parte", donde una clase contiene a otra como parte de sí, pero ambas pueden existir por separado.

#### Características clave:

- Es una relación débil de "todo-parte".
- Si el objeto principal se destruye, los objetos agregados siguen existiendo.
- La clase contenedora no controla el ciclo de vida de los objetos contenidos.

### Ejemplo en Java

Una Universidad contiene una lista de Estudiantes.

- Los estudiantes forman parte de la universidad.
- Pero si la universidad se elimina, los estudiantes no desaparecen, pueden existir en otro contexto.
- Ventaja: Permite modelar escenarios en los que los objetos tienen independencia, pero aún mantienen relación.

### Asociación

#### Asociación simple

Una clase conoce o usa a otra. Ejemplo: la clase Estudiante puede tener referencia a Universidad.

#### Asociación unidireccional

Una clase tiene referencia a otra, pero no al revés. Ejemplo: Orden conoce a su Cliente, pero el cliente no necesita saber de las órdenes.

#### Asociación bidireccional

Ambas clases se conocen mutuamente. Ejemplo: un Profesor puede tener una lista de Estudiantes, y cada estudiante conoce a su profesor.

#### Dependencia

Una clase depende temporalmente de otra para realizar una tarea. Ejemplo: Pedido consulta al Inventario para verificar productos.

### Asociación - importancia

La asociación permite comunicación entre objetos sin necesidad de dependencias rígidas. Ayuda a modelar interacciones del mundo real de manera clara y flexible.

# Relaciones en POO

## JAVA

### Composición

- Es una relación fuerte de "todo-parte".
- Si el objeto principal se destruye, los objetos contenidos también se destruyen.
- El contenedor es responsable de la creación y destrucción de los objetos internos.

### Características clave

- Es una relación fuerte de "todo-parte".
- Si el objeto principal se destruye, los objetos contenidos también se destruyen.
- El contenedor es responsable de la creación y destrucción de los objetos internos.

### Ejemplo en Java

Un Coche contiene un Motor.

- El motor es parte esencial del coche.
- Si el coche se destruye, el motor deja de existir.

Ventaja: Útil para modelar objetos que no tienen sentido fuera del todo. Ayuda a gestionar memoria y ciclo de vida de forma más controlada.

### Diferencias entre Agregación y Composición

Característica	Agregación	Composición
Tipo de relación	Todo-parte débil	Todo-parte fuerte
Independencia	Los objetos pueden existir solos	Los objetos dependen del contenedor
Ciclo de vida	No controlado por el contenedor	Controlado totalmente por el contenedor
Ejemplo	Universidad – Estudiante	Coche – Motor

### conclusion

- Agregación ➡ los objetos son independientes.
- Composición ➡ los objetos están totalmente vinculados al ciclo de vida del contenedor.

