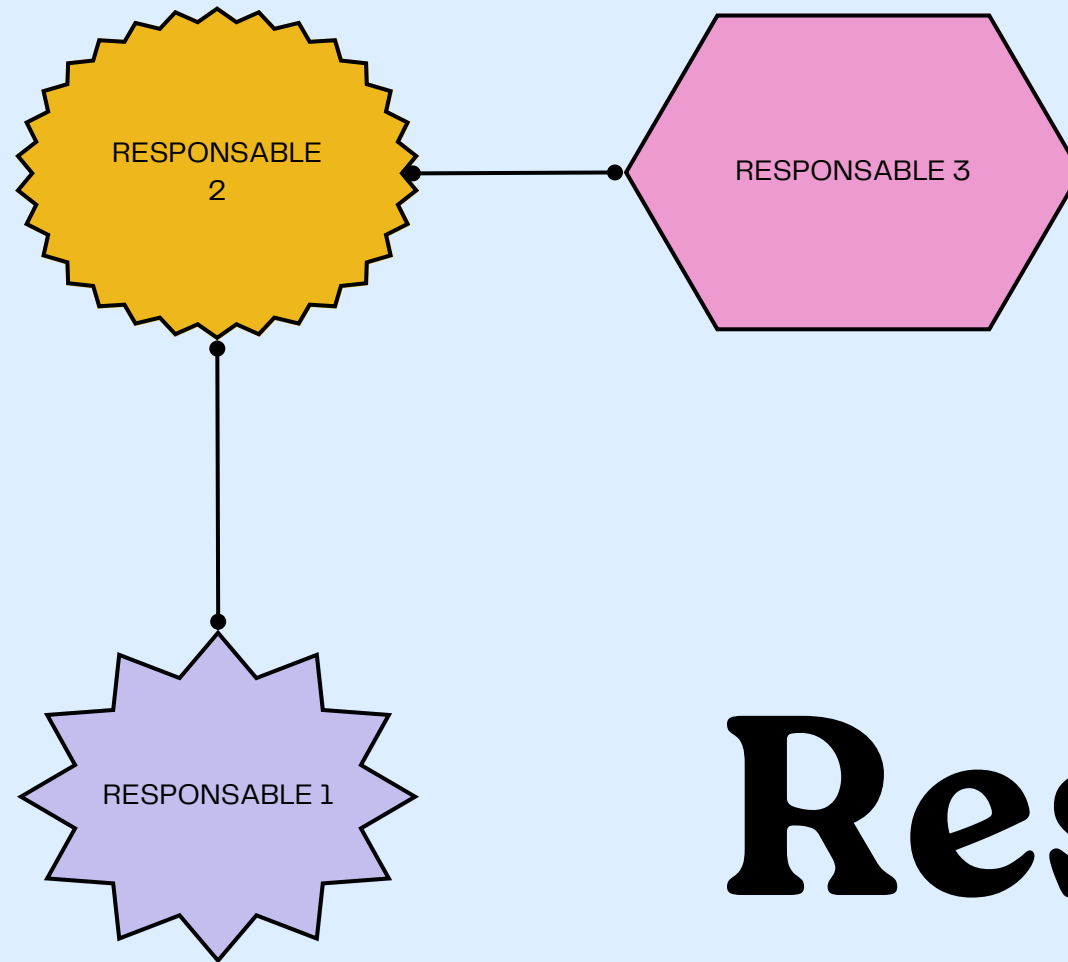
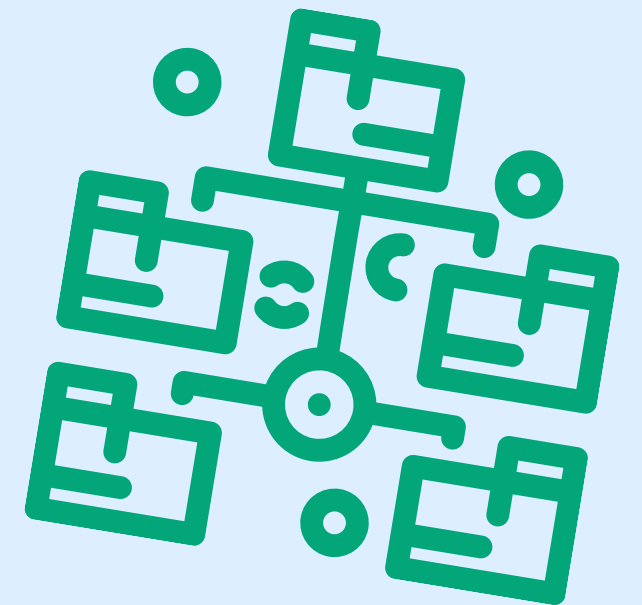


JUNIO 2025

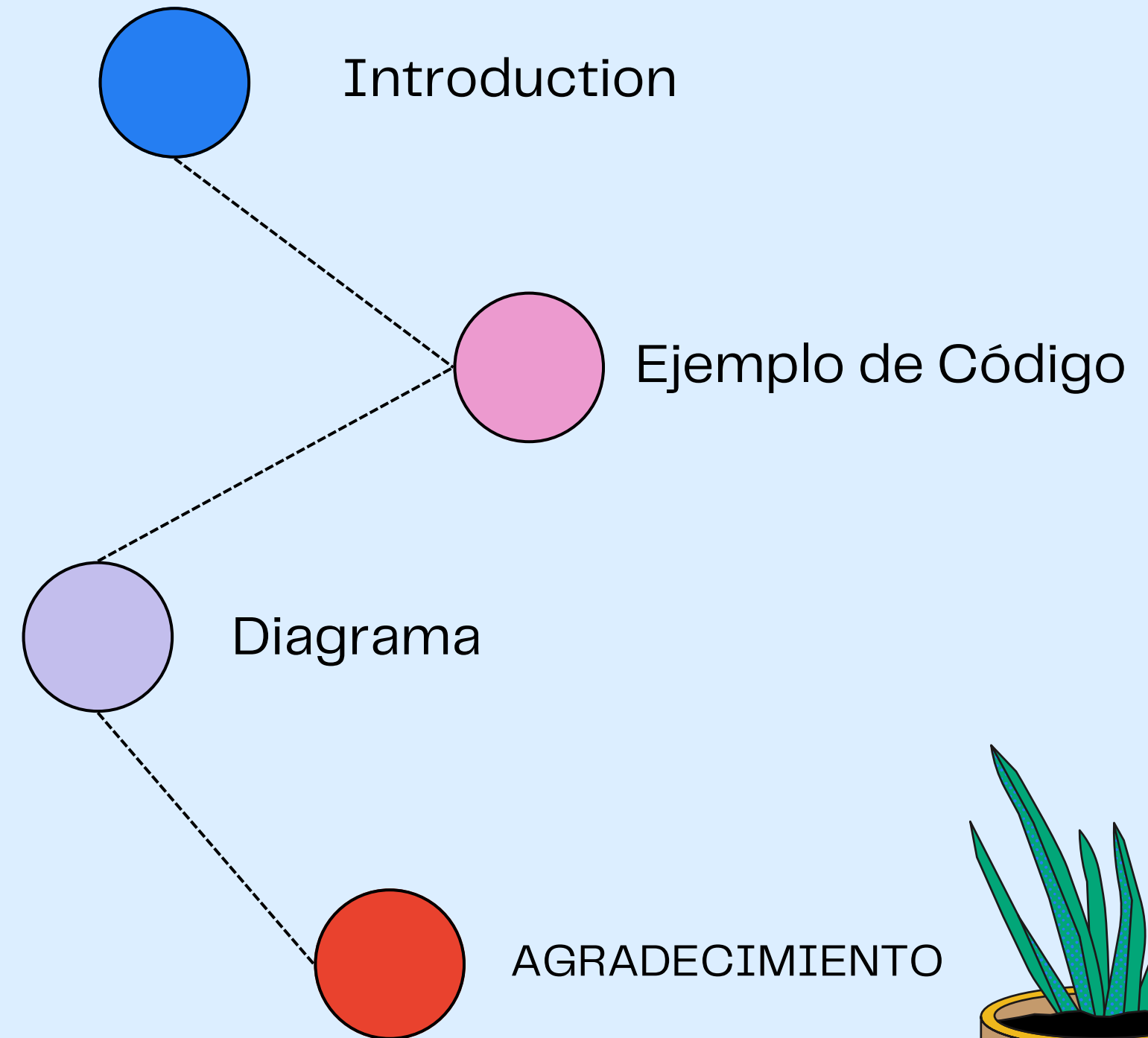


# Cadena de Responsabilidad

ESTEFANI VALVERDE

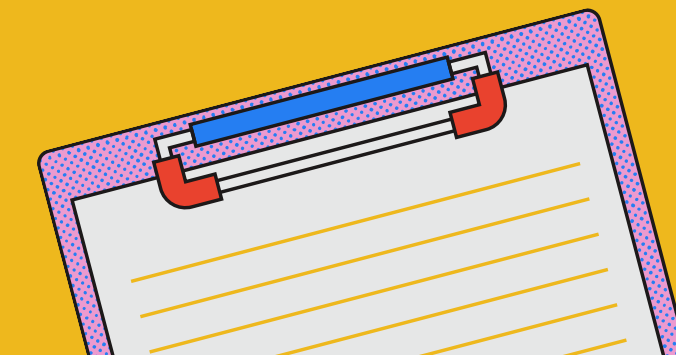
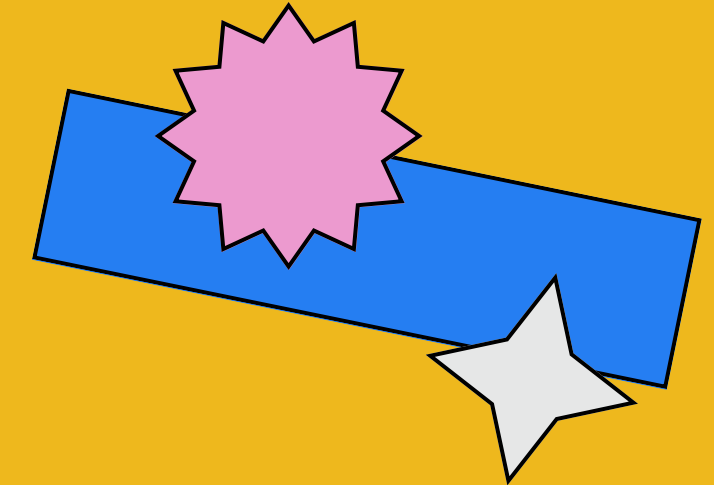
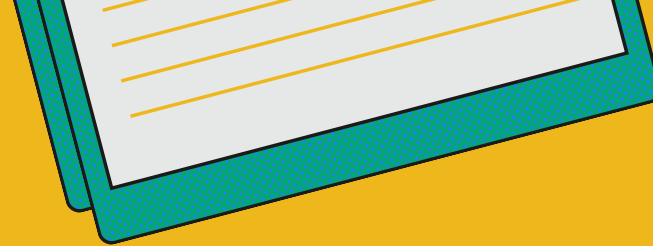
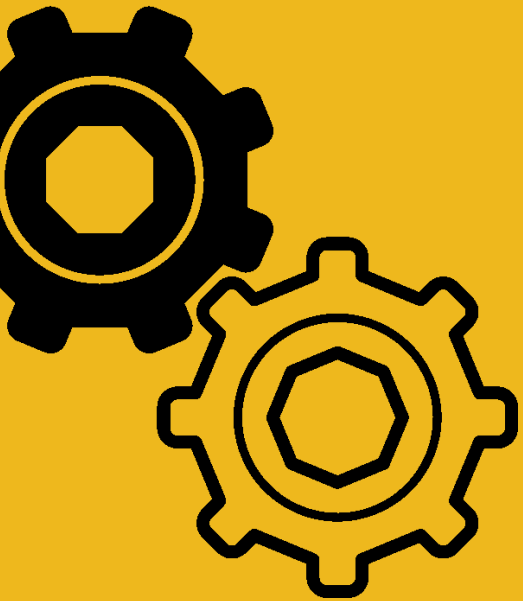


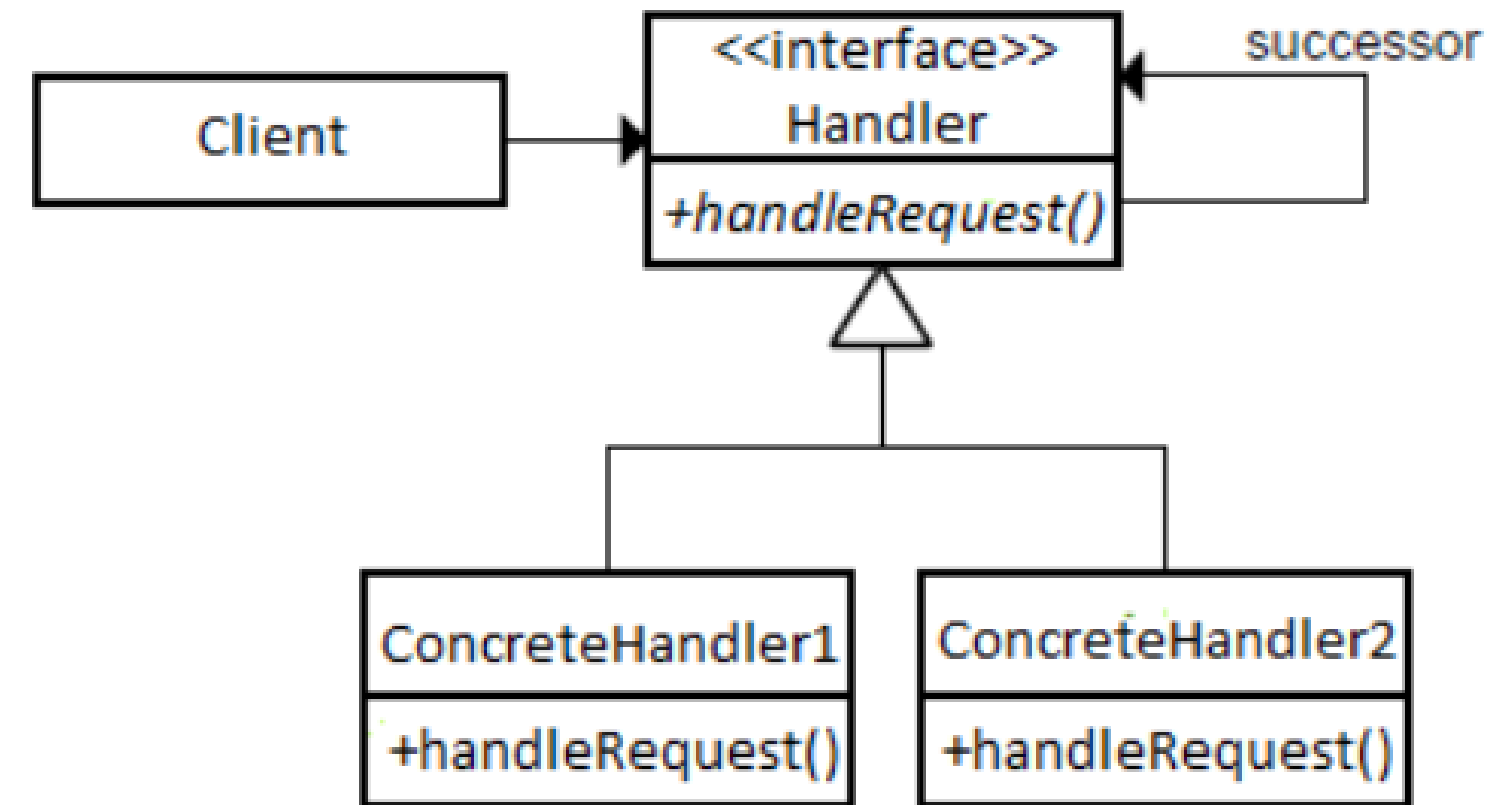
# Agenda

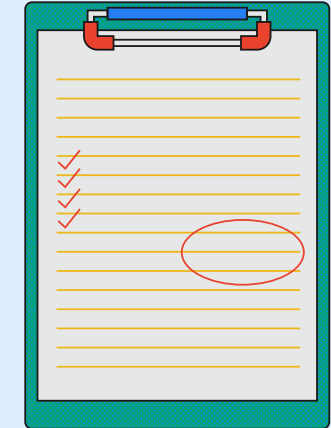
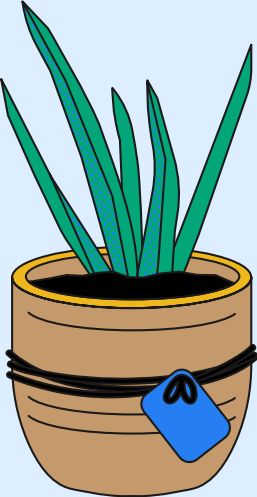


# CHAIN RESPONSABILITY

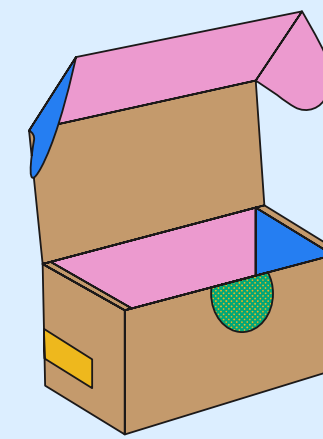
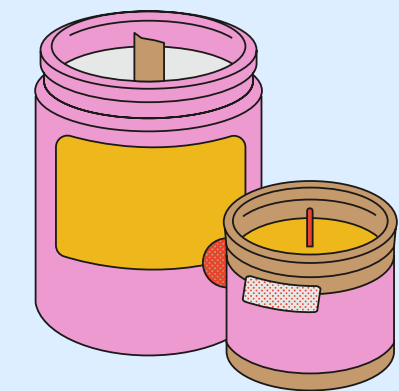
Permitir que varios objetos tengan la oportunidad de manejar una solicitud sin que el emisor conozca cuál objeto la procesará. Los objetos se encadenan formando una cadena y la solicitud se pasa por la cadena hasta que un objeto la maneja.



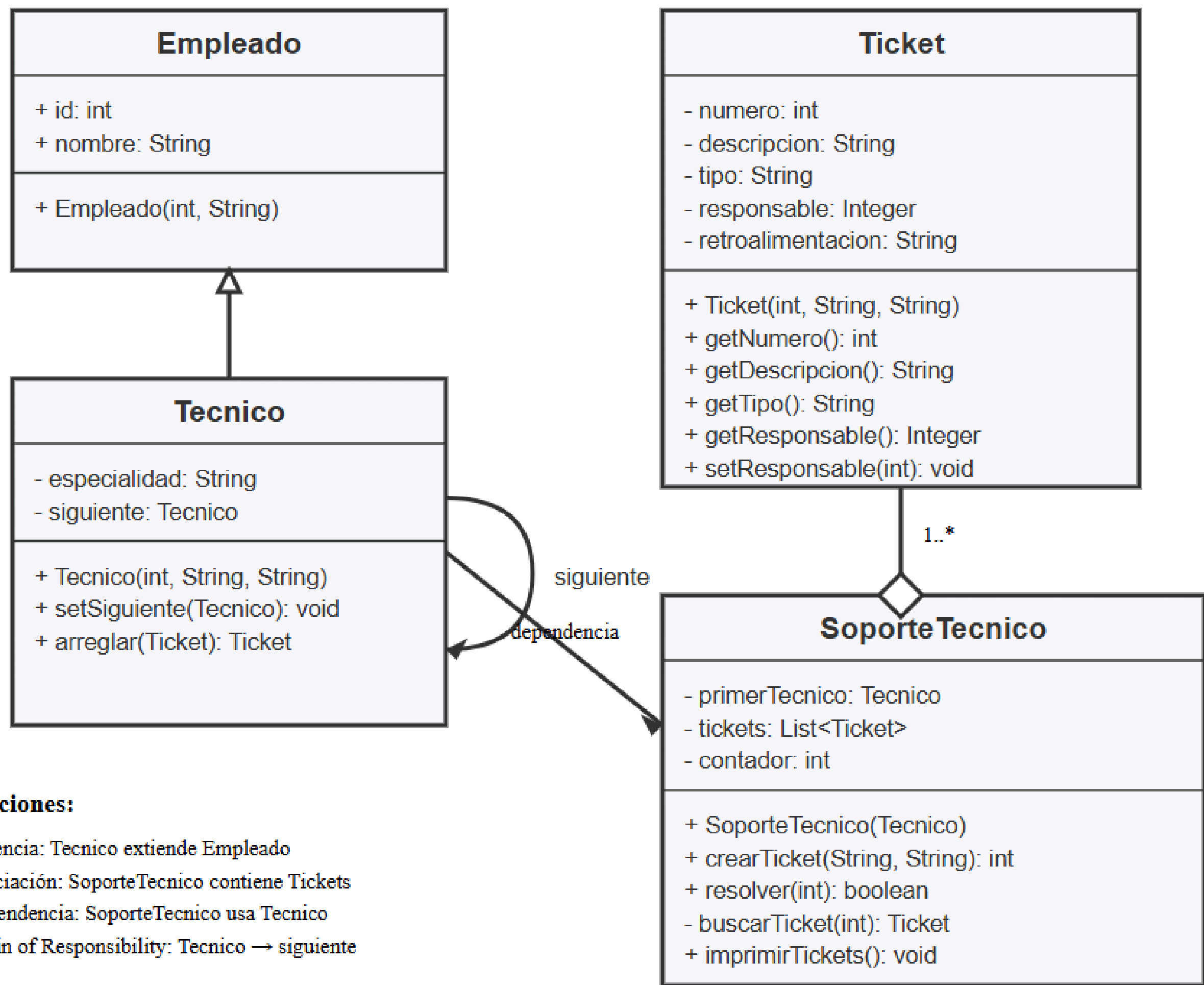
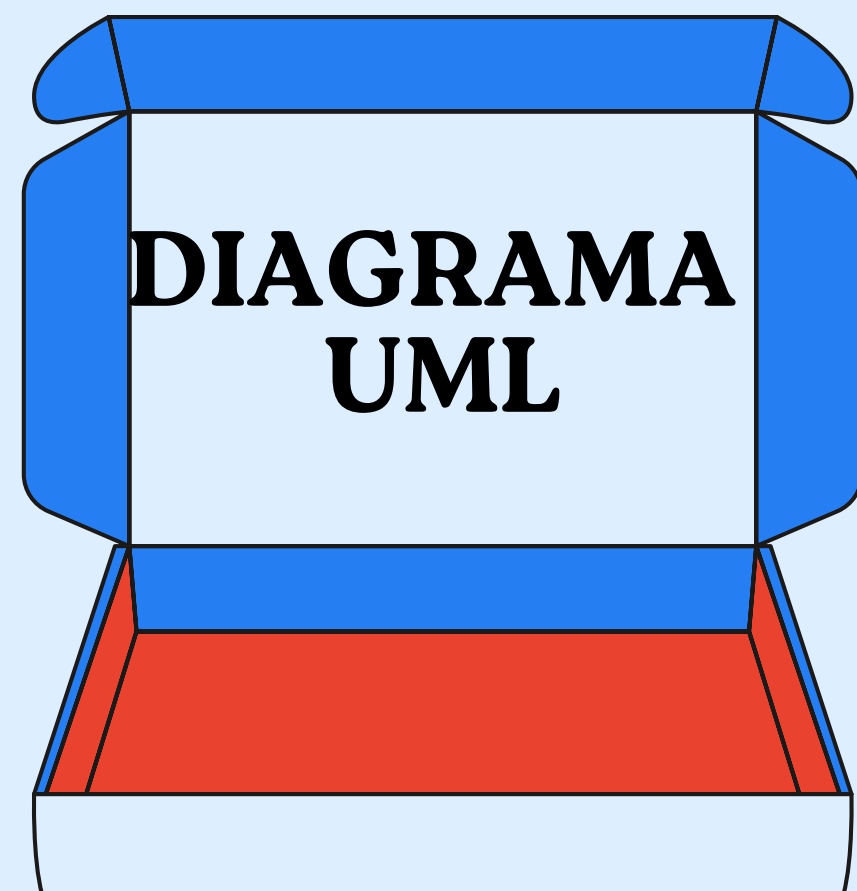




# EJEMPLO



# Sistema de Soporte Técnico - Diagrama UML



## Relaciones:

- Herencia: Tecnico extiende Empleado
- Asociación: SoporteTecnico contiene Tickets
- Dependencia: SoporteTecnico usa Tecnico
- Chain of Responsibility: Tecnico → siguiente



### 🎯 Ventajas

Reduce el acoplamiento entre objetos.

Permite añadir nuevos manejadores sin modificar el código existente.

Favorece el principio de responsabilidad única (SRP).

### ⚠️ Desventajas

No garantiza que la solicitud sea manejada (si ningún objeto la procesa).

Puede ser difícil depurar si la cadena es larga o compleja.

# REFERENCIAS

- Design Patterns: Elements of Reusable Object-Oriented Software

<http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>

- Chain of Responsibility

<https://refactoring.guru/es/design-patterns/chain-of-responsibility>





# MUCHAS GRACIAS

