

PATRONES DE DISEÑO

Mediator



**Universidad
Popular del Cesar**



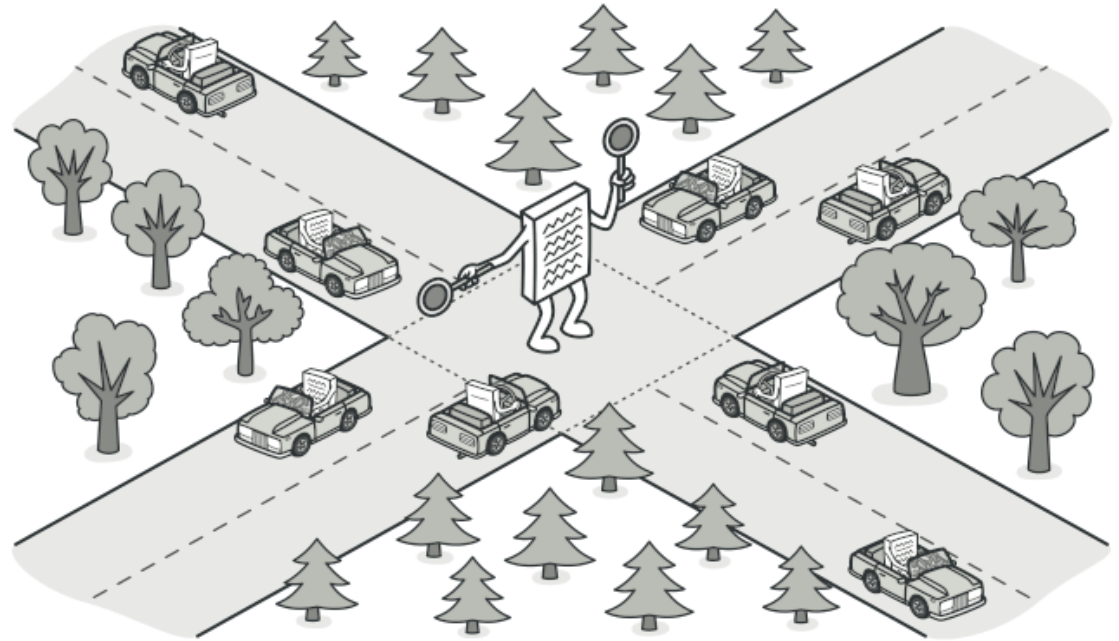
Alberto Rodriguez
Carlos Cruz

Universidad Popular Del Cesar- UPC
Especialización Ingeniería de Software
Valledupar
2024

CONCEPTO

El patrón de diseño Mediator se encarga de gestionar la forma en que un conjunto de clases se comunican entre sí.

Es especialmente útil cuando tenemos una gran cantidad de clases que se comunican de forma directa, ya que podemos crear una capa de comunicación bidireccional entre clases a través de un mediador.

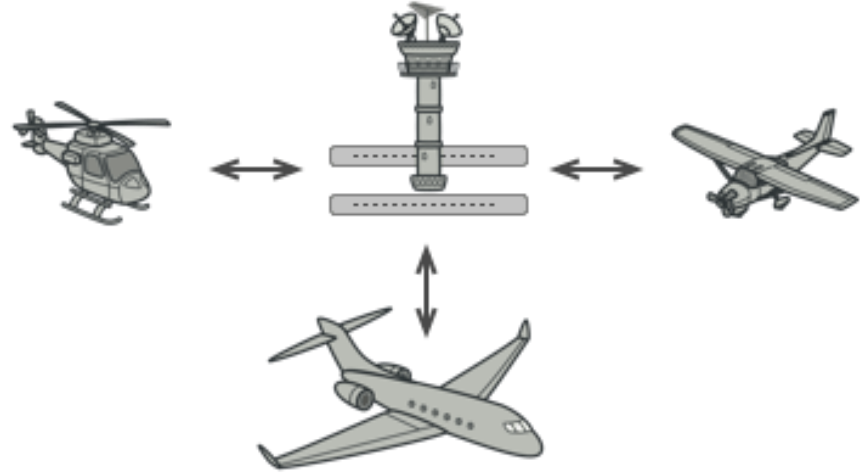


VENTAJAS Y DESVENTAJAS

Ventajas:

Desacoplamiento: Los objetos no necesitan conocer los detalles de otros objetos, solo interactúan con el mediador.

Centralización de la lógica: La lógica de interacción entre objetos se centraliza en el mediador, facilitando el mantenimiento y la evolución del sistema.



Los pilotos de aviones no hablan directamente entre sí para decidir quién es el siguiente en aterrizar su avión. Todas las comunicaciones pasan por la torre de control.

Desventajas:

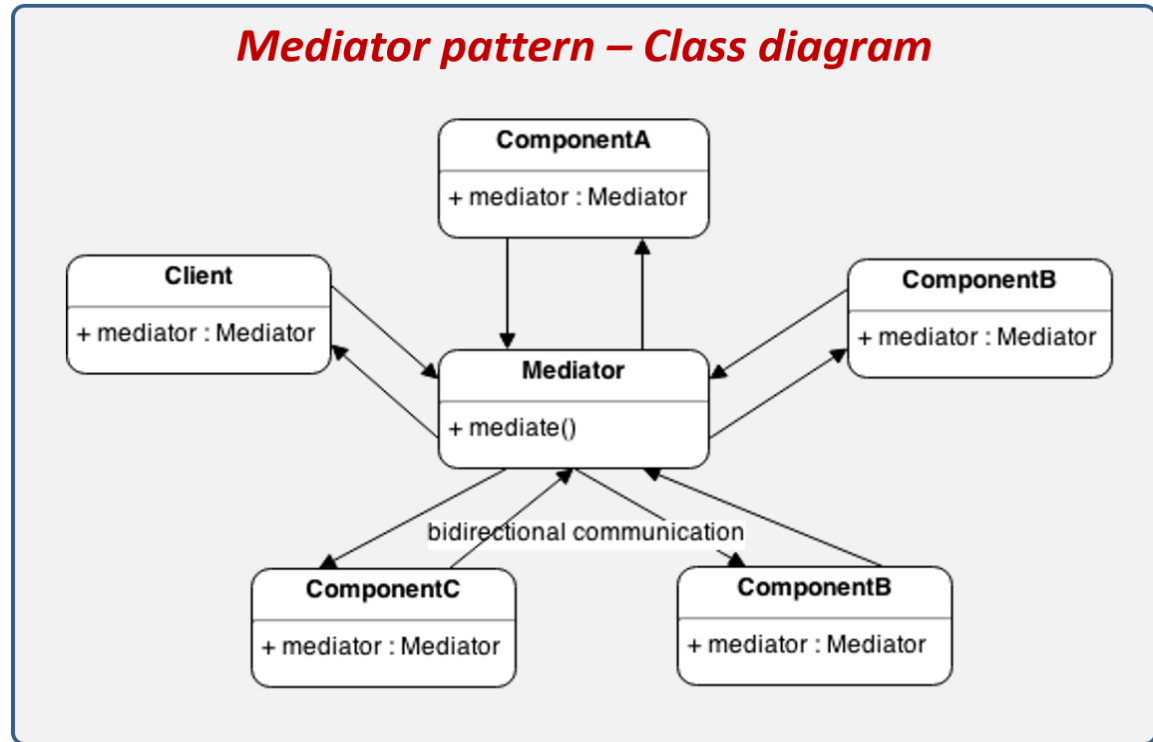
Posible sobrecarga del mediador: El mediador puede convertirse en un punto crítico de complejidad y sobrecarga si no se diseña adecuadamente.

COMPONENTES

Client: Componente que inicia la comunicación con el resto de los componentes por medio del mediador.

Components: Componentes que son parte de la red de comunicación por medio del mediador, éstos pueden ser diversos objetos que comparten el mismo mediador para comunicarse.

Mediador: Componente que sirve de mediador entre el resto de componentes, tiene como principal rol canalizar los mensajes entrantes al destinatario correspondiente.



¿CUANDO IMPLEMENTARLO?

- ☐ Cuando múltiples objetos se comunican de forma compleja y se desea simplificar esta comunicación.
- ☐ Cuando se busca evitar dependencias directas entre objetos, promoviendo un bajo acoplamiento.
- ☐ Cuando se necesita centralizar y controlar la lógica de interacción entre objetos.

Casos de uso:

- ☐ Sistemas de chat donde varios usuarios (colegas) se comunican a través de un servidor central (mediador).
- ☐ Interfaces gráficas de usuario donde varios componentes interactúan mediante un controlador central.

EJERCICIO

Mediator: Define la interfaz para la comunicación entre los colegas.

ConcreteMediator: Implementa la interfaz Mediator y coordina la comunicación entre los objetos Colleague..

Colleague: Define una interfaz para los componentes que se comunican a través del Mediator.

ConcreteColleague: Implementa la interfaz Colleague y utiliza al Mediator para interactuar con otros colegas.

