# Patrones de Diseño: Patrones de Comportamiento. Tema 5-6. Mediator

### Descripción del patrón

#### Nombre:

Mediador

#### Propiedades:

Tipo: comportamiento

Nivel: objeto, componente

#### Objetivo o Propósito:

 Proporcionar la posibilidad de que un único objeto se encargue de gestionar la comunicación entre distintos objetos, sin que éstos necesiten conocerse entre sí. Promueve un bajo acoplamiento al evitar que los objetos se refieran unos a otros explícitamente.





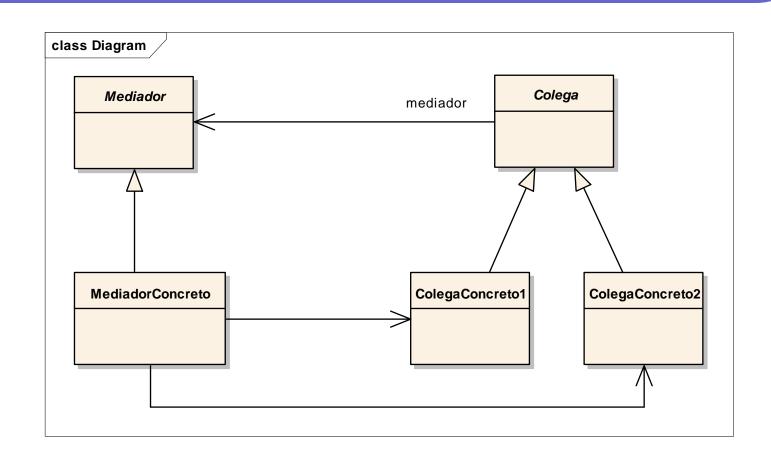
#### Aplicabilidad

- Use el patrón Mediator cuando:
  - Las reglas de comunicación sean complejas entre los objetos de un sistema.
  - Queramos adaptar un comportamiento distribuido entre clases sin necesidad de especializarlas.
  - Se desee que los objetos sigan siendo simples y manejables.
  - Cuando las clases son difíciles de reutilizar porque tienen muchas relaciones de interdependencia con otras.

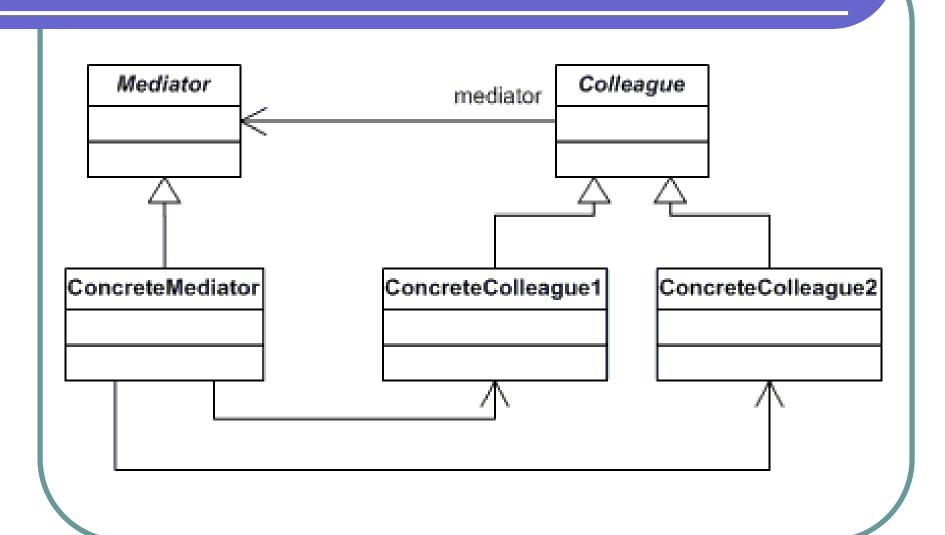




#### Estructura



#### Estructura







#### Estructura. Participantes

- **Mediador:** Interfaz en la que se definen los métodos que pueden ser invocados por los clientes.
- MediadorConcreto: Implementa la interfaz Mediador. Esta clase hace de intermediaria entre los colegas. El mediador concreto puede invocar métodos específicos en el colega o un método genérico para informar a los colegas de algo.
- Colega: Interfaz en la que se definen los métodos para que un mediador pueda informar a los colegas.
- ColegaConcreto: Clase que implementa la interfaz colega. Suele mantener una referencia a una instancia de Mediador para informar a los colegas de un cambio.





#### Consecuencias

- Los componentes individuales de la aplicación se hacen más simples y es más fácil tratar con ellos porque no necesitan pasarse mensajes entre sí.
- Simplifica la estrategia de comunicación ya que es responsabilidad exclusiva del mediador.
- Reduce la herencia. Un mediador localiza el comportamiento que de otra manera estaría distribuido en varios objetos. Para cambiar el comportamiento solo tenemos que crear una subclase de Mediador.
- Desacopla a los colegas.
- Un problema es que el código del mediador puede ser difícil de mantener conforme aumentan el número de clases y la complejidad de estas.





#### Patrones relacionados

- Observer: Las clases Mediator frecuentemente utilizan el patrón Observer para recibir notificaciones de las diferentes clases.
- Facade: Difiere del Mediador en que abstrae un subsistema para proporcionar una interfaz más conveniente. Su protocolo es unidireccional, sin embargo el patrón Mediador proporciona un protocolo multidireccional.
- Adapter: Se puede utilizar el patrón Adapter para independizar la clase Mediator de las clases concretas que gestiona.





# Código de ejemplo

# Chat





# Código de ejemplo

