

Patrones de Diseño:

*Patrones de
Comportamiento.*

Tema 5-2:

Chain of Responsibility

Descripción del patrón

- **Nombre:**
 - Cadena de Responsabilidad
 - También conocido como expert
- **Propiedades:**
 - Tipo: comportamiento
 - Nivel: objeto, componente
- **Objetivo o Propósito:**
 - Desacoplar el emisor de un mensaje del objeto que presta el servicio. Para conseguir esto se establece una cadena en el sistema de forma que el emisor envía un mensaje al primer elemento de la cadena, éste lo procesa si puede, y sino, lo redirige a otro objeto de la cadena hasta que uno pueda procesarlo.

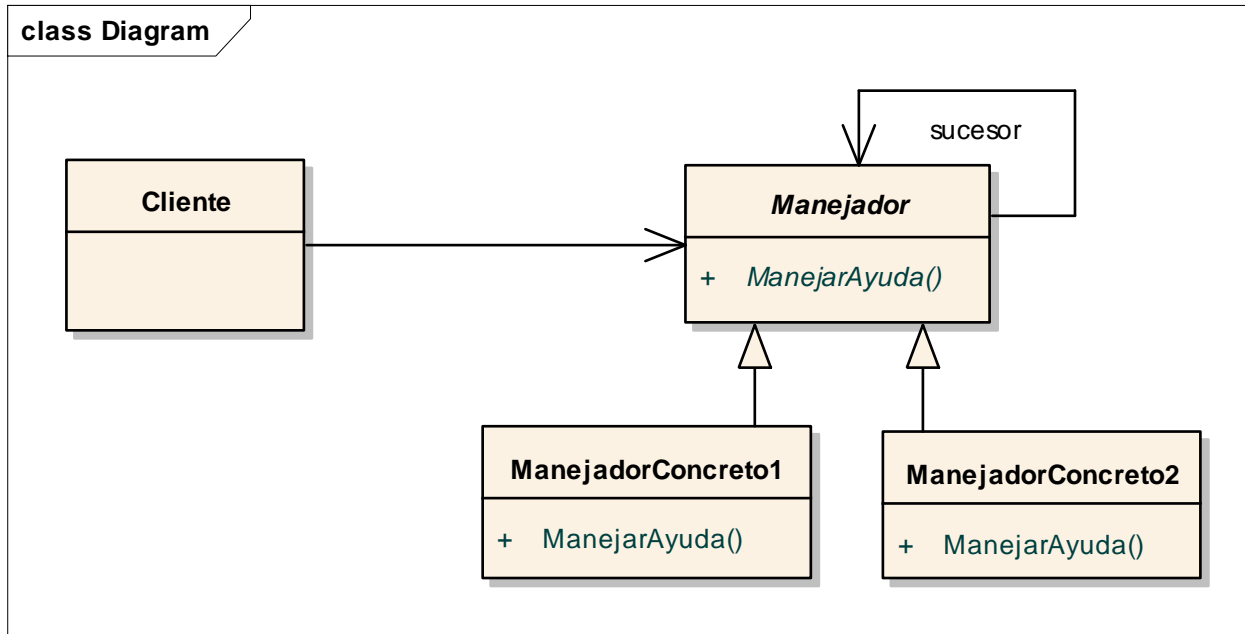


Aplicabilidad

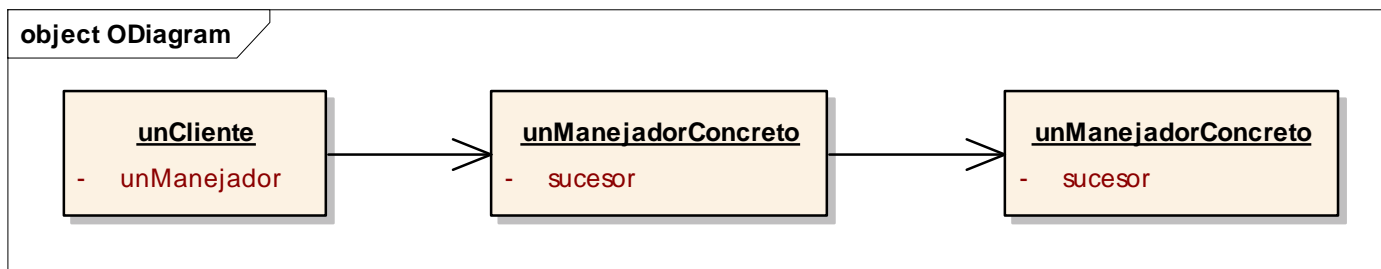
- Use el patrón Chain of Responsibility cuando:
 - Hay un grupo de objetos en un sistema que pueden responder al mismo tipo de mensajes y a priori no se sabe quien es.
 - Los mensajes deben ser manejados por uno entre varios objetos. No se especifica el receptor del mensaje.
 - Algunos mensajes deben ser manejados en el nivel donde se reciben mientras que otros deben ser redirigidos a algún otro objeto (la solicitud se especifica dinámicamente).



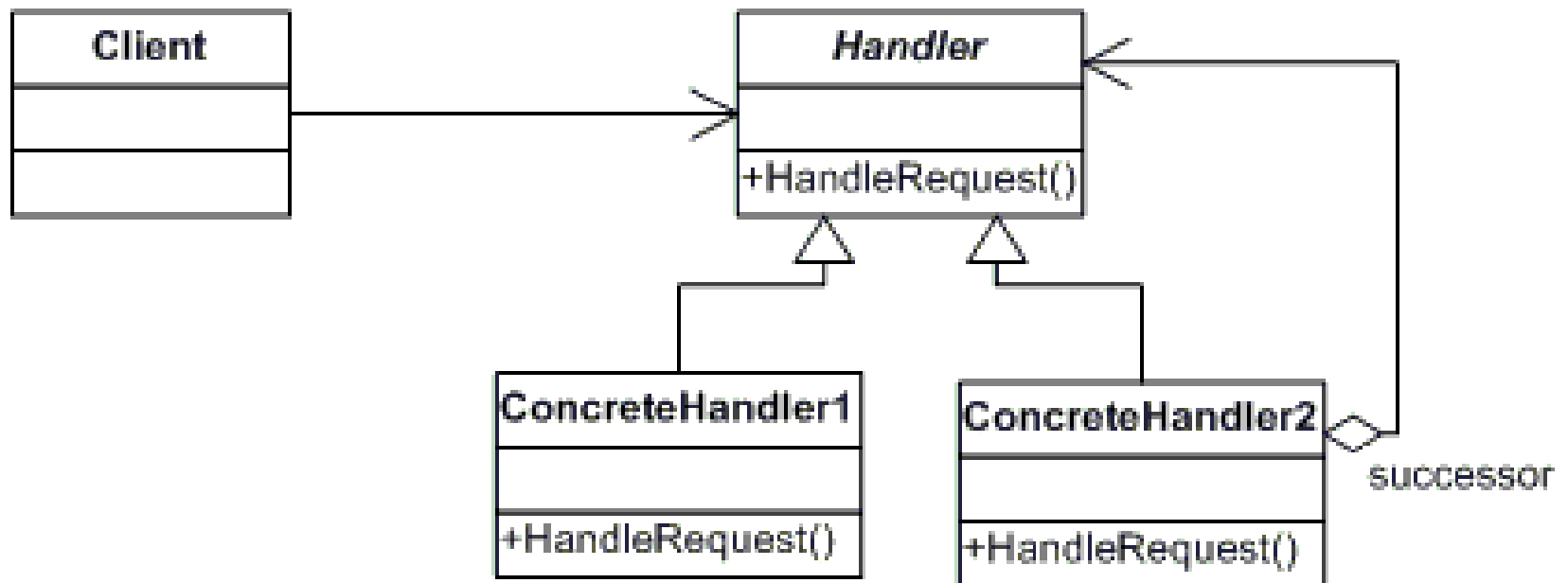
Estructura



Una estructura de objetos típica podría parecerse a esta:



Estructura



Estructura. Participantes

- **Manejador:** Define una interfaz que contiene un método para tratar las peticiones, evalúa si puede responder al mensaje, en caso de que pueda lo hace y sino se pasa el mensaje al siguiente elemento de la cadena.
- **ManejadorConcreto:** Representa a un elemento que forma parte de la cadena de responsabilidad. Mantiene una referencia al elemento sucesor en la cadena. Si puede procesar el mensaje lo hace, en caso contrario, lo reenvía al siguiente elemento (manejador concreto) de la cadena de responsabilidad.
- **Cliente:** Inicializa la petición a un objeto ManejadorConcreto de la cadena.



Estructura. Variaciones

- **Variaciones del patrón:**
- **Estrategias de gestión.** Se centran en cómo se implementa el comportamiento del manejador.
 - Manejador por defecto. Utilizado cuando no hay una clase explícita a la que reenviar el mensaje (mensajes ignorados).
 - Manejar y ampliar. La propagación de los mensajes implica un comportamiento base y uno ampliado.
 - Manejadores dinámicos. Permiten que se cambie la estructura de reenvío de mensajes en tiempo de ejecución mediante un método que establece el siguiente elemento de la cadena.



Estructura. Variaciones

- **Variaciones del patrón:**
- **Estrategias de reenvío.** Permiten gestionar o redirigir los mensajes producidos por un componente.
 - Manejar por defecto. Maneja cualquier mensaje que no sea reenviado de forma específica.
 - Propagar por defecto. Redirige cualquier mensaje que no sea explícitamente reenviado.
 - Reenviar a un manejador por defecto.
 - Ignorado por defecto. Se descarta cualquier mensaje que no sea explícitamente manejado o redirigido. Se reduce el número de mensajes en circulación.



Consecuencias

- El objetivo de este patrón, como el de muchos otros, es el de desacoplar objetos, reduciendo el acoplamiento. Esto se consigue con Cadena de Responsabilidad ya que un objeto solo necesita saber como debe procesar un evento o mensaje.
- Este patrón ofrece una gran flexibilidad ante el procesamiento de eventos ya que delega las responsabilidades en elementos más simples. Todo esto permite que un conjunto de clases se comporte como una unidad.



Consecuencias

- Debido a que no hay un receptor explícito no se garantiza la recepción del mensaje. Se puede alcanzar el final de la cadena sin que se haya procesado la petición.
- Una desventaja es que, a medida que el redireccionamiento de la cadena se hace más complejo, se hace más difícil de comprobar y depurar, ya que hay que controlar si los mensajes se están enviando correctamente.



Patrones relacionados

- **Composite:** Cuando se combina con Chain of Responsibility se tiene soporte para una estructura basada en árbol con propagación de mensajes.

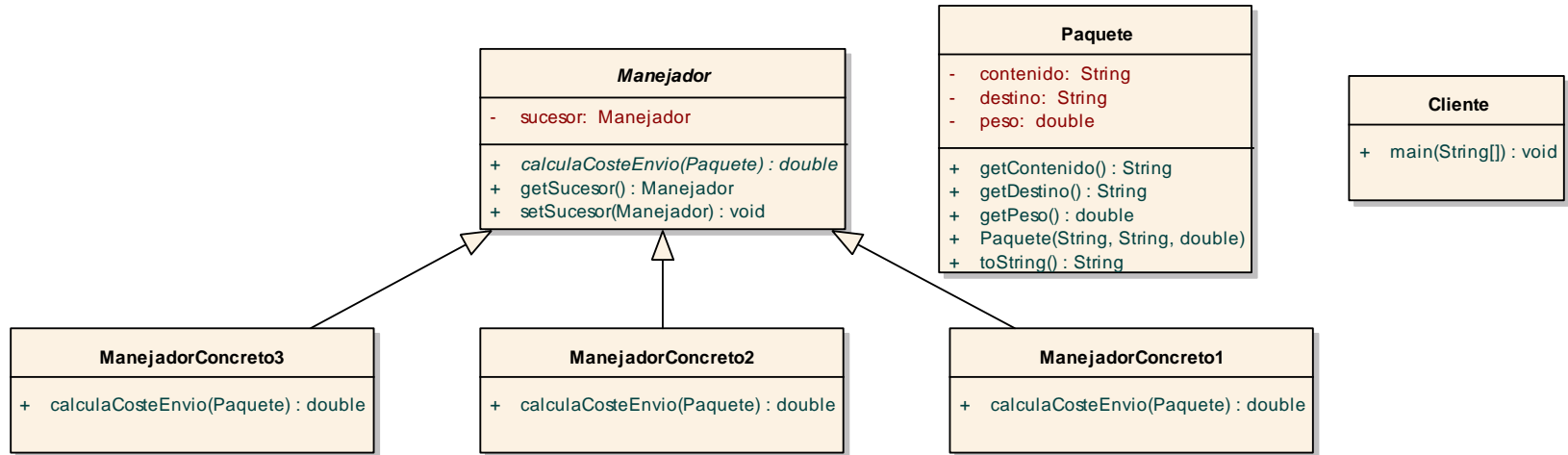


Envío Paquetes



Código de ejemplo

class Class Model



Código de ejemplo

- Identificamos a continuación los elementos del patrón:
 - Manejador: Manejador.
 - ManejadorConcreto: ManejadorConcreto1, ManejadorConcreto2, ManejadorConcreto3.
 - Cliente: Cliente.

