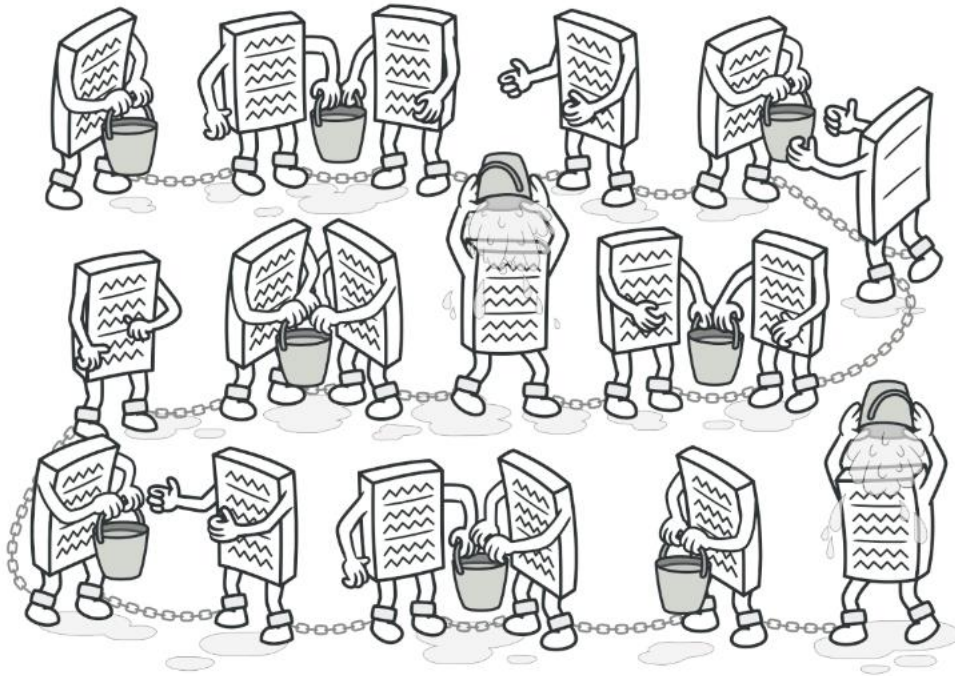
Chain of responsibility

Agenda

- Introducción
- Componentes
- Ventajas
- Aplicabilidad
- Pros y contras
- Relaciones con otros patrones
- Usos conocidos

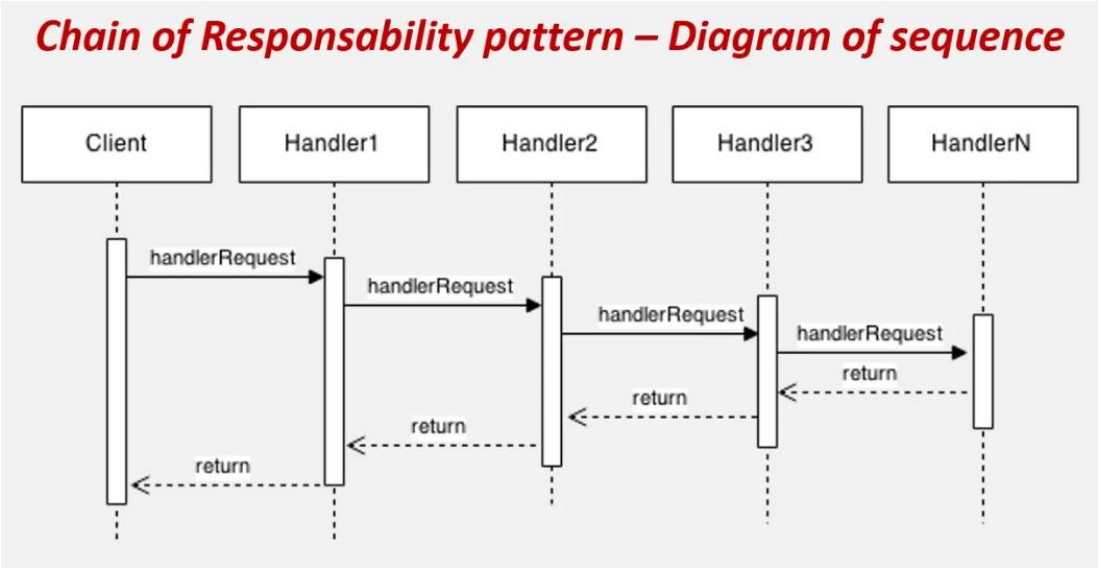
Introducción

- Permite pasar solicitudes a lo largo de una manejadora



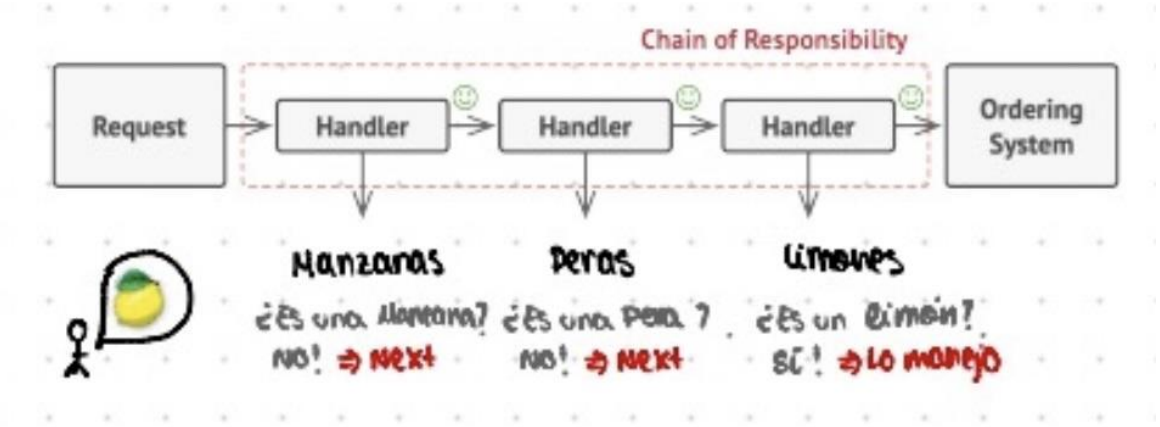
Componentes

- Cliente
- AbstractHandler
- ConcreteHandler



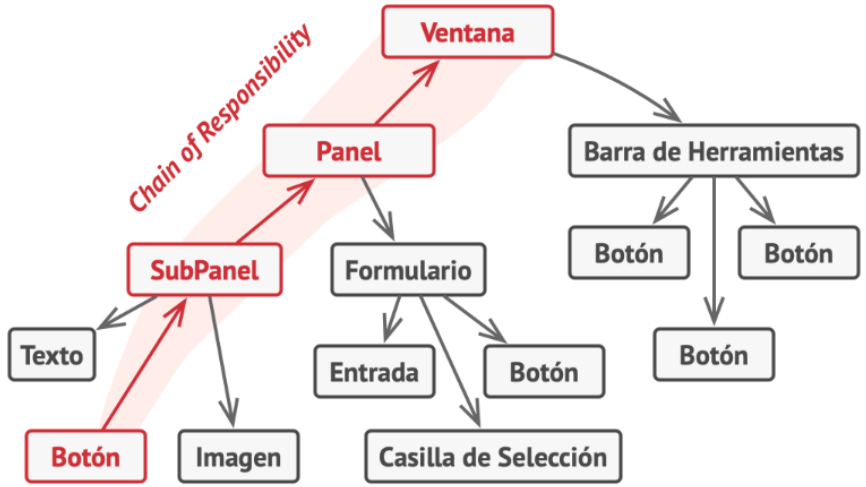
Ventajas

- Principio responsabilidad única
- Principio abierto/cerrado
- Puedes controlar orden de control de solicitudes



Aplicabilidad

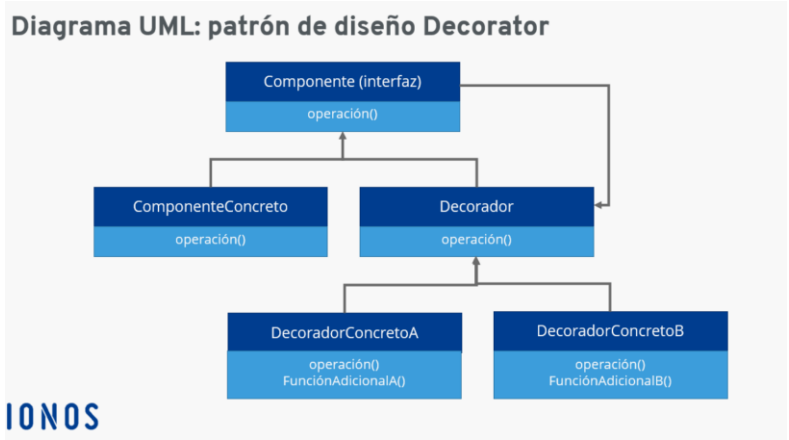
- Procesa diferentes solicitudes sin conocer su orden de antemano.
- Encadena manejadores para evaluar y procesar cada solicitud.
- Permite modificar el orden de los manejadores en tiempo de ejecución.

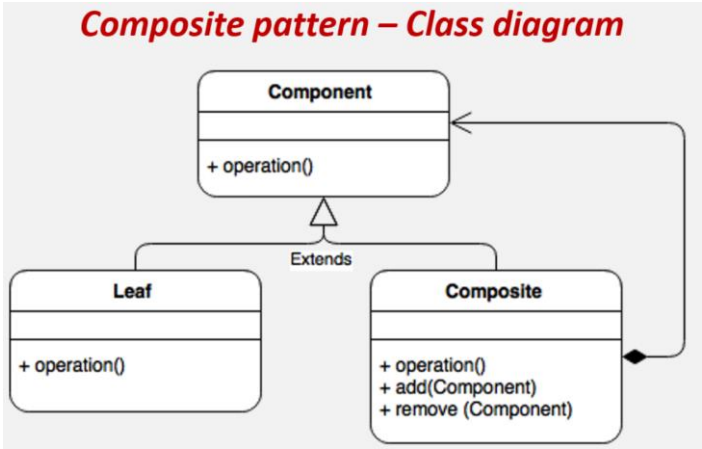


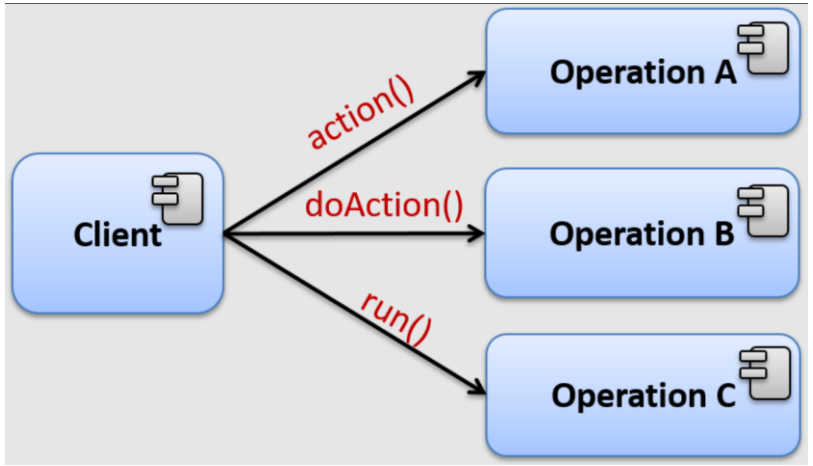
Una cadena puede formarse a partir de una rama de un árbol de objetos.

Relaciones con otros patrones

- Composite
- Command
- Decorator

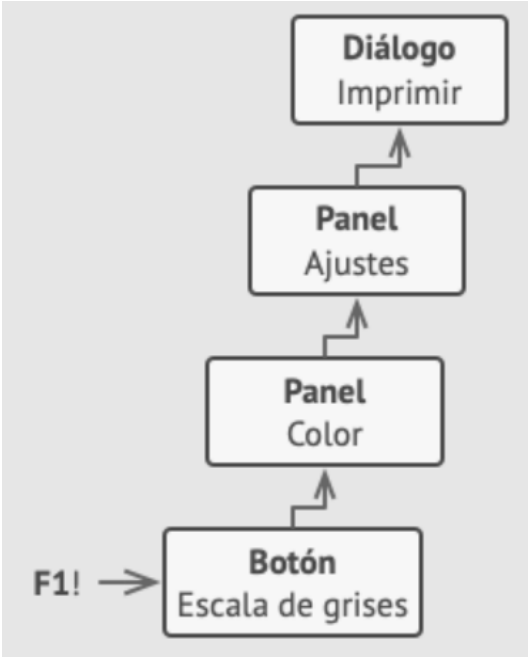






Usos conocidos

- Manejadores de eventos en bibliotecas de usuario
- Editores gráficos
- Sistemas de ayuda



Fuentes

- Chain of Responsibility (refactoring.guru)
- Patrones de diseño: Chain of Responsability | by Adrián Alonso | Medium
- Chain of Responsability (reactiveprogramming.io)
- Patrón Chain of Responsibility | PPT | Descarga Gratuita (slideshare.net)