

Chain of Responsibility

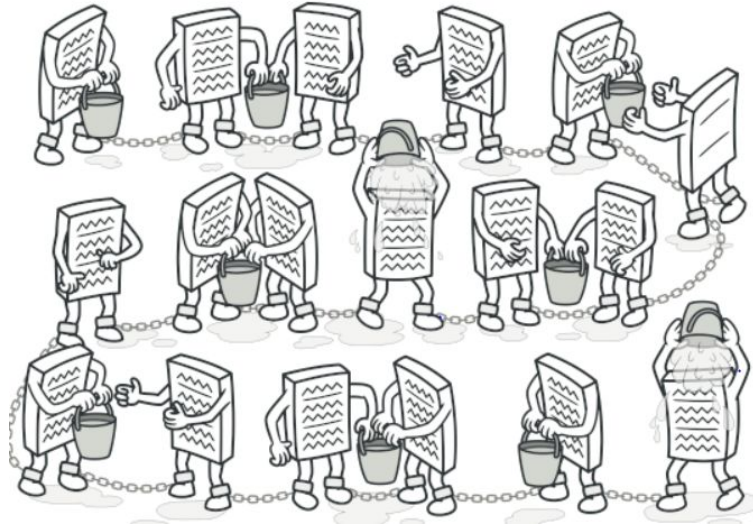
cadena de responsabilidad

Hilario Tun

Que es?

Es un patrón de diseño de comportamiento que te permite pasar solicitudes a lo largo de una cadena de manejadores.

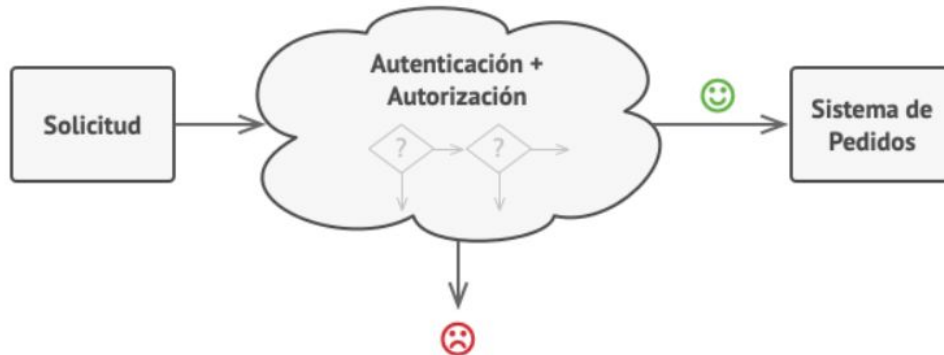
Al recibir una solicitud, cada manejador decide si la procesa o si la pasa al siguiente manejador de la cadena.



Problemática

En un sistema de pedidos online para restringir el acceso al sistema de forma que solo los usuarios autenticados puedan generar pedidos y los usuarios con permisos administrativos cuenten con acceso a todos los pedidos, identificamos de que estas comprobaciones deben realizarse secuencialmente.

La aplicación puede intentar autenticar a un usuario en el sistema cuando reciba una solicitud que contenga las credenciales del usuario. Sin embargo, si esas credenciales no son correctas y la autenticación falla, no hay razón para proceder con otras comprobaciones.



Solución

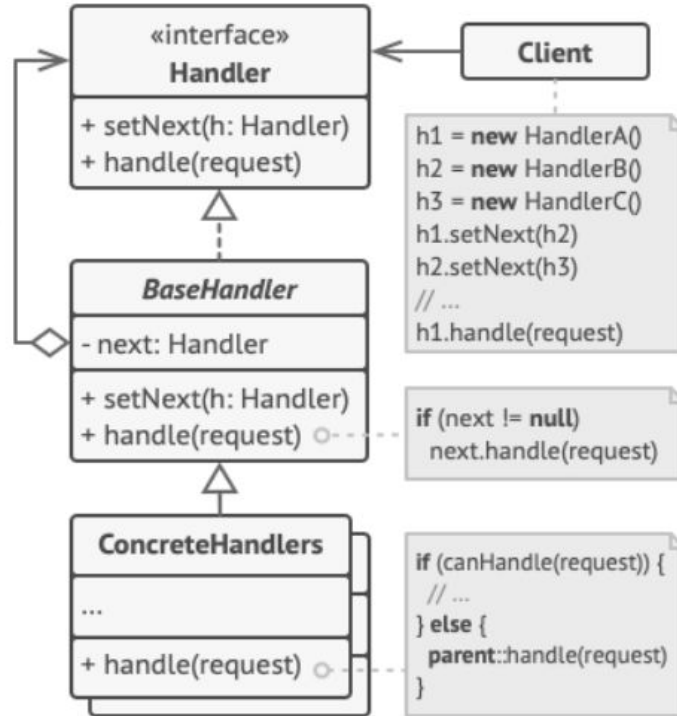
- Transformar comportamientos particulares en objetos autónomos llamados *manejadores* donde cada comprobación debe ponerse dentro de su propia clase con un único método que realice la comprobación donde la solicitud, junto con su información, transforme el método como argumento.
- Vincular manejadores en una cadena donde el manejador vinculado tiene un campo para almacenar una referencia al siguiente manejador de la cadena. para procesar una solicitud, los manejadores la pasan a lo largo de la cadena. La solicitud viaja por la cadena hasta que todos los manejadores han tenido la oportunidad de procesarla.



Estructura

La **clase Manejadora** declara la interfaz común a todos los manejadores concretos. Normalmente contiene un único método para manejar solicitudes, pero en ocasiones también puede contar con otro método para establecer el siguiente manejador de la cadena.

La **clase Manejadora Base** es opcional y es donde puedes colocar el código boilerplate (segmentos de código que suelen no alterarse) común para todas las clases manejadoras. esta clase define un campo para almacenar una referencia al siguiente manejador. La clase también puede implementar el comportamiento de gestión por defecto: puede pasar la ejecución al siguiente manejador después de comprobar su existencia.



El **Cliente** puede componer cadenas una sola vez o componerlas dinámicamente, dependiendo de la lógica de la aplicación.

Los **Manejadores Concretos** contienen el código para procesar las solicitudes. Al recibir una solicitud, cada manejador debe decidir si procesarla y, además, si la pasa a lo largo de la cadena

Aplicabilidad

- Cuando se espera que su programa procese diferentes tipos de solicitudes de diversas maneras, pero los tipos exactos de solicitudes y sus secuencias se desconocen de antemano.
- Vincular varios controladores en una cadena y, al recibir una solicitud, "preguntar" a cada controlador si puede procesarla. De esta manera, todos los manejadores tienen la oportunidad de procesar la solicitud.
- Cuando sea esencial ejecutar varios controladores en un orden particular.
- Si proporciona configuradores para un campo de referencia dentro de las clases de controlador, podrá insertar, eliminar o reordenar controladores dinámicamente.

