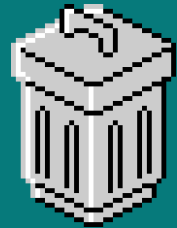




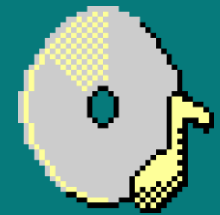
Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas
Computacionales



Ingeniería en Sistemas de Información
Ingeniería de software aplicada

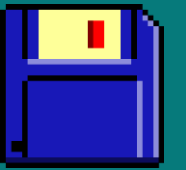


Chain of Responsibility



Estudiante:
Chin, Willie (8-973-698)

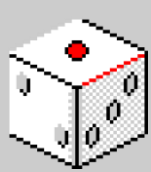
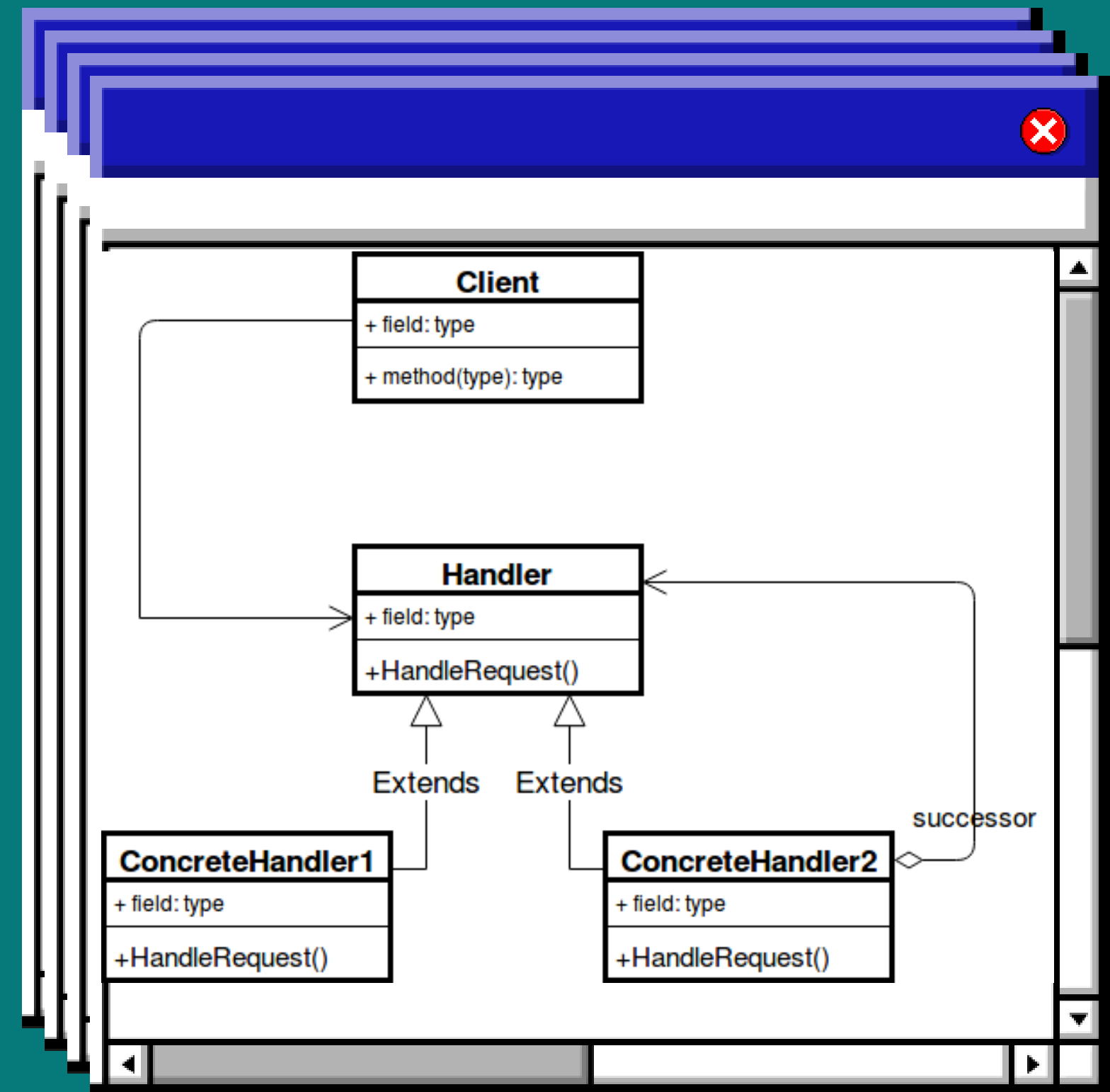
Profesora:
Prof. Jeanette Riley



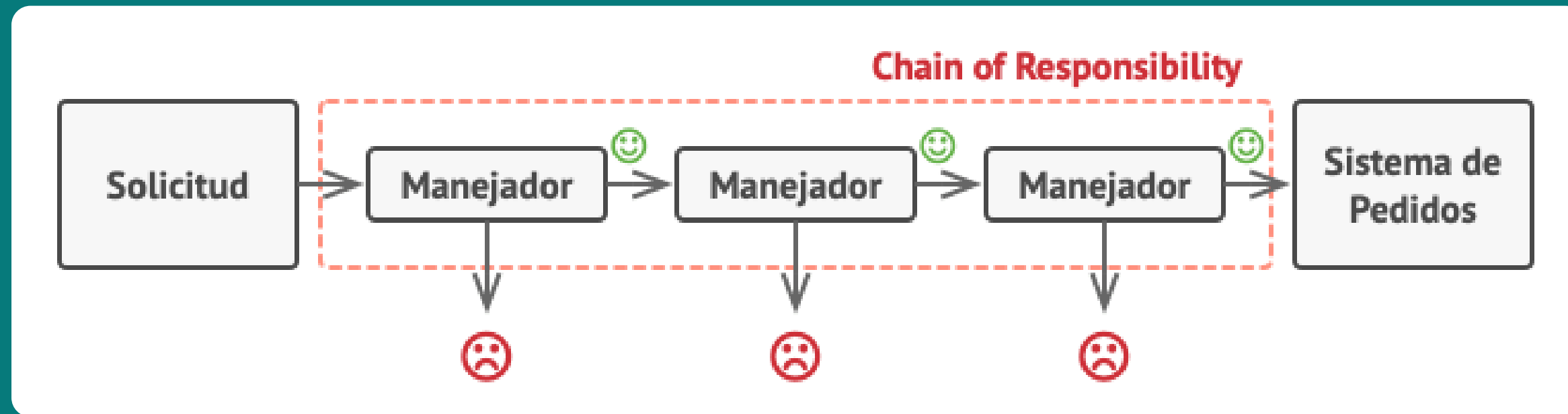
Mié. 7 de junio 2023

Patrones de diseño de software

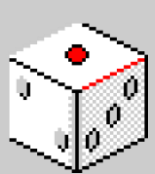
Cadena de responsabilidad ("Chain of Responsibility")



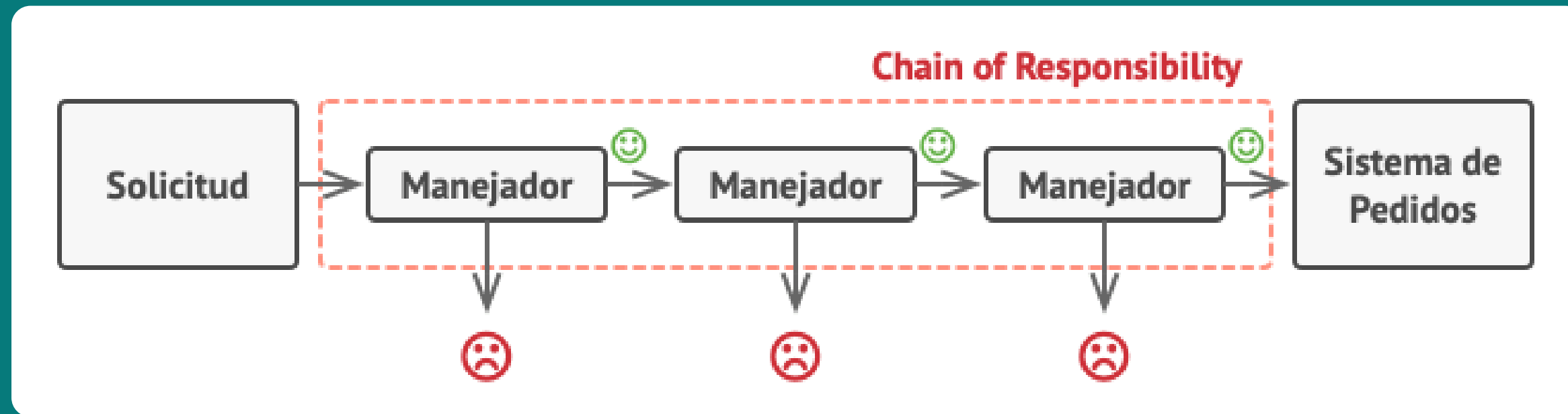
Chain of responsibility



Chain of Responsibility es un patrón de diseño de comportamiento que permite pasar solicitudes a lo largo de una cadena de manejadores. Al recibir una solicitud, cada manejador decide procesar la solicitud o pasarla al siguiente controlador de la cadena.



Chain of responsibility



La cadena de responsabilidad permite desacoplar el remitente de una solicitud de sus receptores, dando a varios objetos la oportunidad de manejar la solicitud de forma independiente.



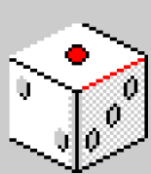
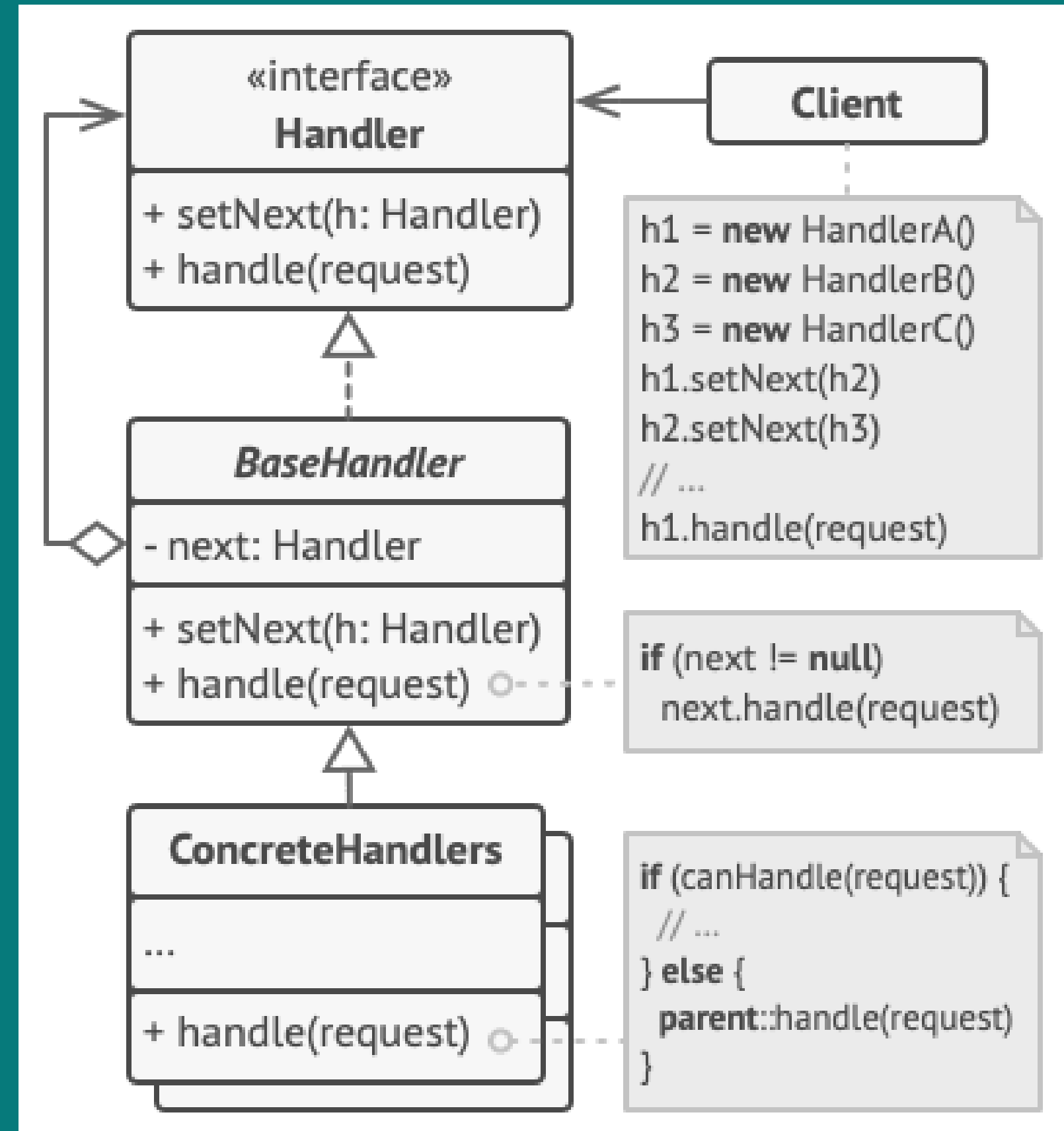
Estructura

1. 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.

2. 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.



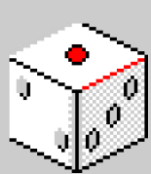
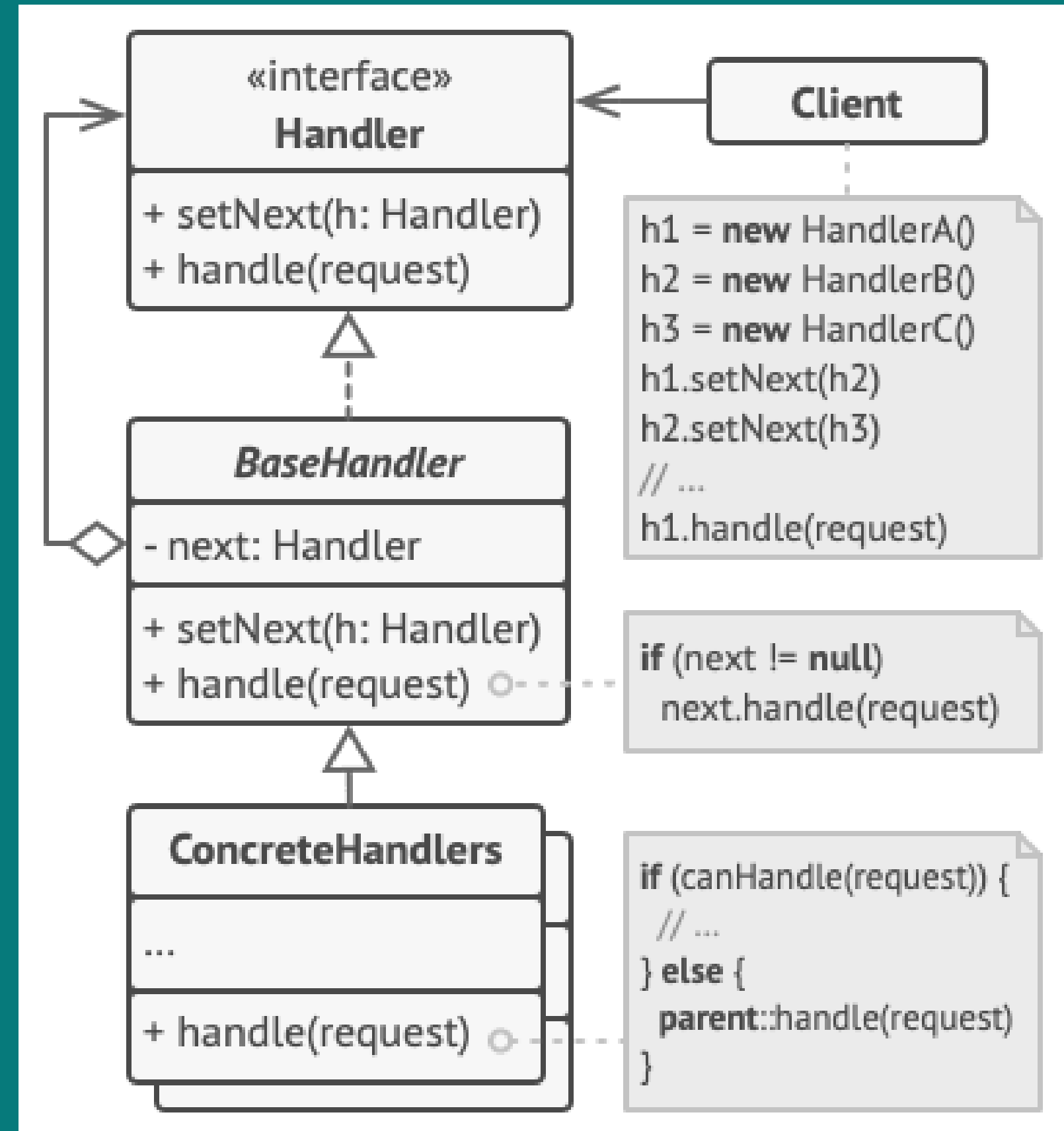
Estructura

3. El Cliente:

Puede componer cadenas una sola vez o componerlas dinámicamente, dependiendo de la lógica de la aplicación. Observa que se puede enviar una solicitud a cualquier manejador de la cadena; no tiene por qué ser al primero.

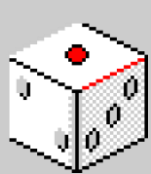
1. Los Manejadores Concretos:

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.



Ejemplo de uso

Cuando trabajamos con sistemas ERP o CRM, es muy común gestionar casi cualquier cosa mediante ordenes, ya sea órdenes de venta, compra, servicio, instalación, etc. Dichas ordenes son objetos muy complejos que contiene dentro un gran número de otros objetos, como clientes, proveedores, productos, etc. y los sistemas están forzados a validar que una orden este completa y que todos los datos capturados sean válidos para el tipo de orden.



Ejemplo de uso

Mediante el patrón Chain Of Responsibility se puede crear un mecanismo óptimo para validar cualquier tipo de orden, reutilizando los validadores entre los distintos tipos de órdenes.

