Patrón proxy

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

“

Veamos cómo podemos controlar el acceso a un objeto, permitiendo hacer algo antes o después de que la solicitud llegue al mismo.

”

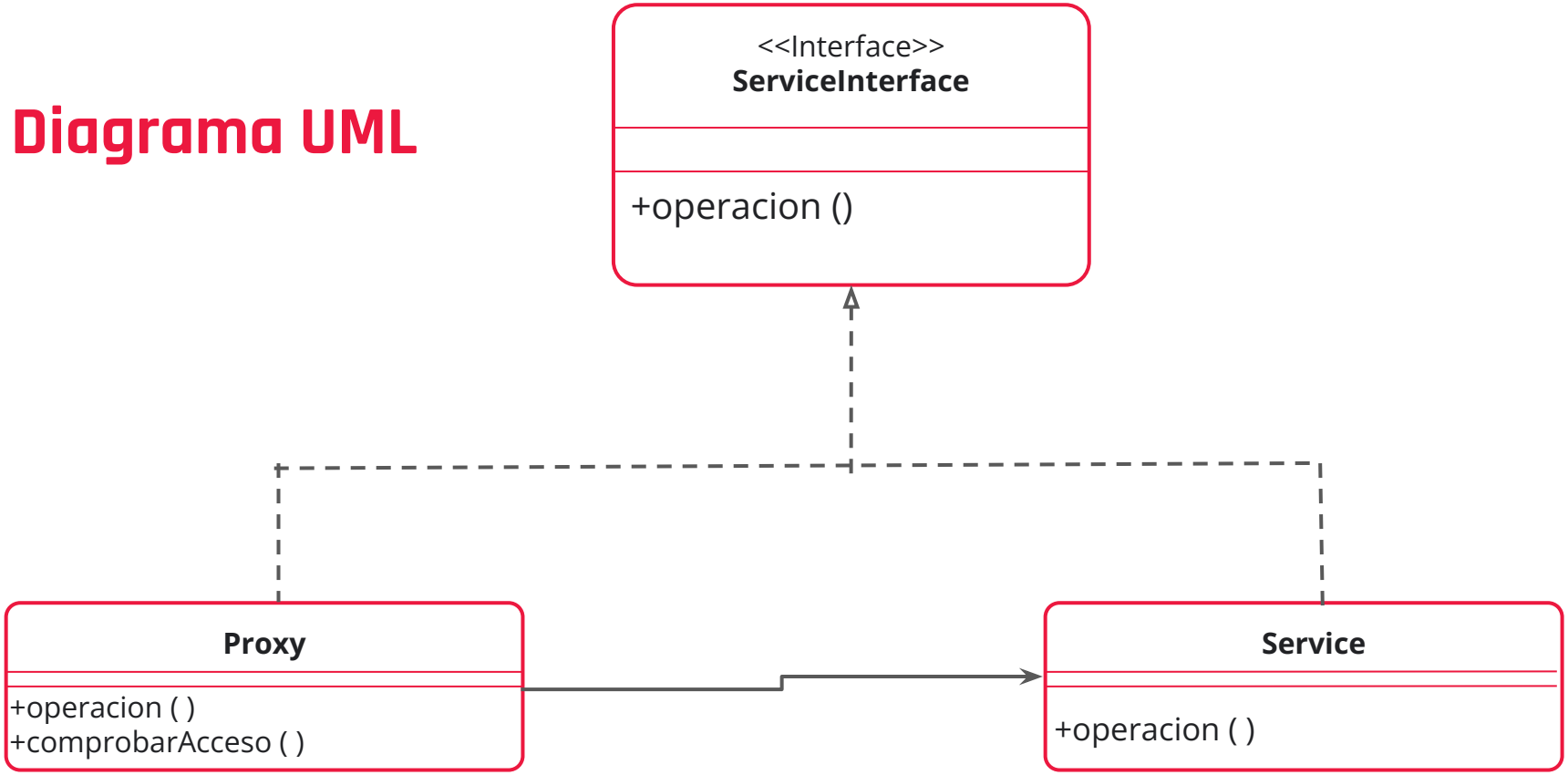
Propósito

Tiene por objetivo desarrollar la función de ser un intermediario que agrega funcionalidad a una clase, sin tocar la misma.

Solución

Definir una clase Proxy con la misma interfaz que el objeto de servicio original. Después, se debe actualizar nuestra aplicación para que los clientes se comuniquen con el proxy y no con el servicio destino. Al recibir la solicitud de un cliente, el proxy le enviará al servicio, pero como intermediario podremos realizar operaciones antes o después de enviarle la solicitud.

Diagrama UML



Ventajas

El proxy funciona incluso si el objeto de servicio no está listo o no está disponible.

Principio de abierto/cerrado: Podemos introducir nuevos proxies sin cambiar el servicio o los clientes.

Desventajas

Al agregar una capa más entre el cliente y el servicio real, la respuesta puede retrasarse.


DigitalHouse>
Coding School


Certified Tech
Developer
The Ultimate Degree