



PROXY - PATRONES DE DISEÑO DE SOFTWARE

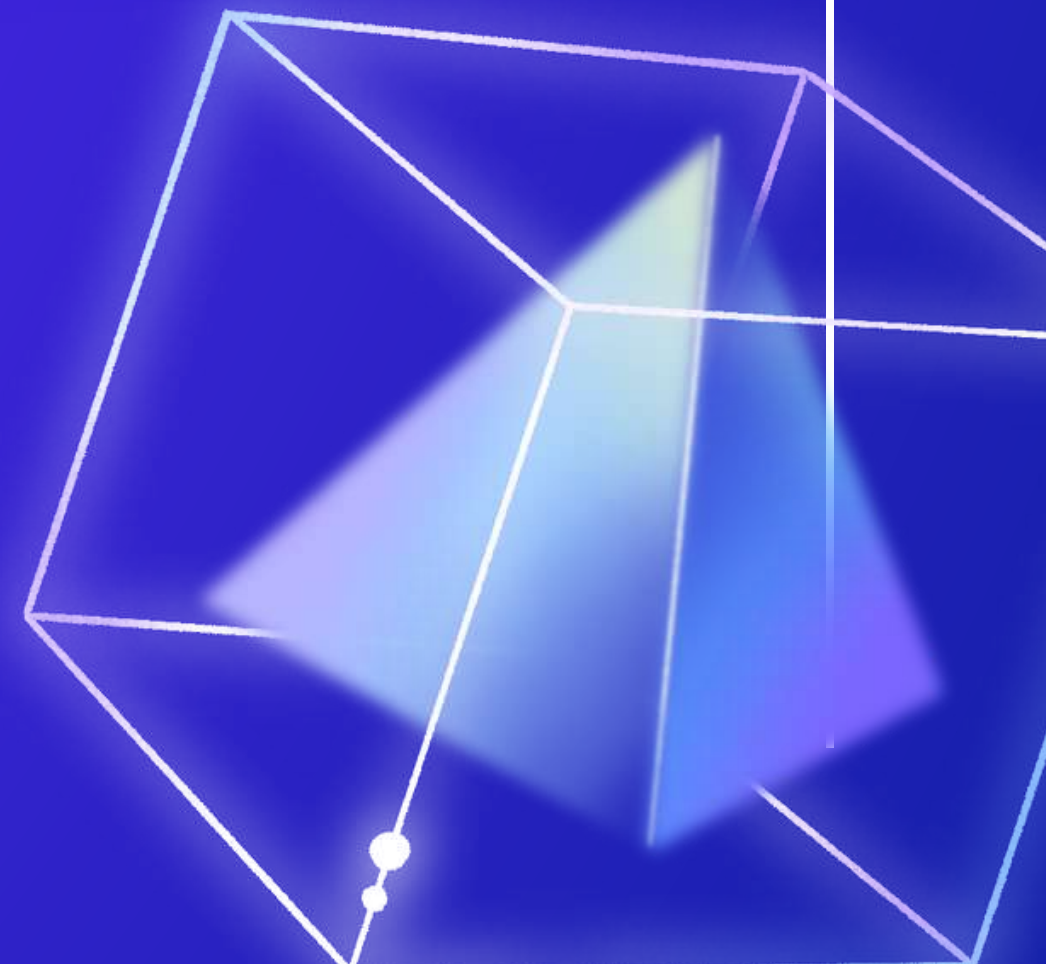
Ordoñez, Génesis



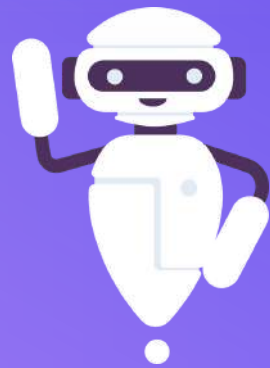


¿QUÉ ES EL PATRÓN DE DISEÑO PROXY?

es un patrón estructural que se utiliza para controlar el acceso a un objeto o para proporcionar una interfaz alternativa a dicho objeto. El proxy actúa como un intermediario entre el cliente y el objeto real, permitiendo controlar el acceso a este último y realizar tareas adicionales si es necesario.

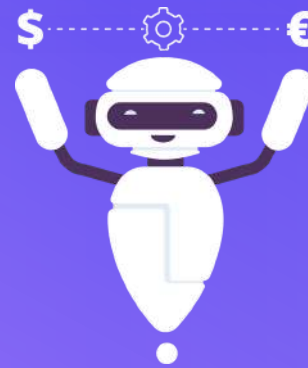


PARTICIPANTES DEL PATRÓN



SUJETO

es la interfaz común que tanto el Proxy como el objeto real implementan.



PROXY

es un objeto que actúa como intermediario entre el cliente y el objeto real.



CLIENTE

es el código que utiliza el Proxy para acceder al objeto real.

TIPOS DE PROXIES



Proxy remoto

Proporciona una representación local de un objeto remoto. El proxy maneja la comunicación y transmite las solicitudes al objeto remoto.

Proxy virtual

Crea objetos pesados o costosos de inicializar solo cuando sea necesario. El proxy crea el objeto real solo cuando se llama a un método específico o se accede a una propiedad.

TIPOS DE PROXIES



Proxy de protección

Controla el acceso al objeto real y puede aplicar restricciones de seguridad adicionales según los permisos del cliente.

Proxy inteligente

Realiza tareas adicionales de optimización, como el almacenamiento en caché de resultados, o gestiona los recursos del objeto real.



VENTAJAS

- Puedes controlar el objeto de servicio sin que los clientes lo sepan
- Puedes administrar el ciclo de vida del objeto de servicio cuando a los clientes no les importa
- El proxy funciona incluso si el objeto de servicio no está listo o disponible
- Principio abierto/cerrado. Puedes introducir nuevos proxies sin cambiar el servicio o los clientes

EJEMPLO DE USO

Imagina que tienes una clase que carga imágenes desde un servidor remoto. Puedes implementar un proxy para esta clase que realice el almacenamiento en caché de las imágenes ya cargadas, evitando la carga repetida si se solicita la misma imagen.