## Patrón de diseño de software: Facade

Facade o Fachada es un tipo de patron estructural que tiene como fin ocultar y/o simplificar la complejidad que pueda tener la aplicación de cara al cliente, para que tenga un punto de acceso cómodo y sin tener la necesidad de saber el completo funcionamiento interno del sistema.

Cuando usar el Facade:

Cuando se quiera estructurar un subsistema en capas, con el fin de definir puntos de entrada a cada nivel de un subsistema. Con esto se puede reducir el acoplamiento entre varios subsistemas exigiéndoles que se comuniquen únicamente mediante fachadas.

Resolver una problemática

Los clientes que acceden a un subsistema complejo se refieren directamente a un gran número de objetos con interfaces completamente diferentes o dependen de estos objetos. Esto hace que la implementación, adaptación, prueba y reutilización de los clientes sea particularmente difícil para los desarrolladores.

Tener una fachada resulta útil cuando tienes que integrar tu aplicación con una biblioteca sofisticada con decenas de funciones, de la cual sólo necesitas una pequeña parte. Este patron funciona como intermediario entre el cliente y el sistema en bruto. También pueden existir fachadas adicionales que actúen con el cliente o con otras fachadas.

Este patrón define un objeto de fachada central que:

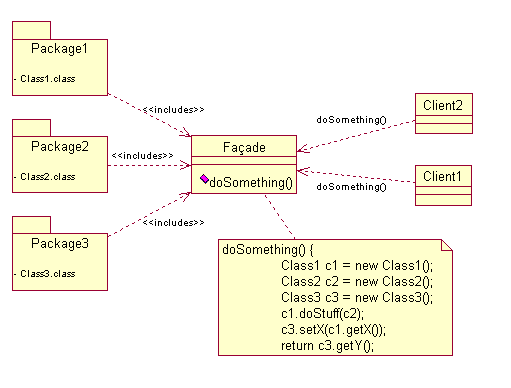
* implementa una **interfaz universal** para las distintas interfaces del subsistema o subsistemas.
* y (si es necesario) puede realizar **funciones adicionales** antes o después de reenviar una solicitud al cliente.

Como intermediario, el objeto de fachada garantiza que el **acceso y la comunicación** con los componentes individuales de un subsistema se simplifiquen y se **reduzca al mínimo la dependencia directa** de estos componentes. Este delega las llamadas de los clientes para que éstos no necesiten conocer las clases o sus relaciones y dependencias.

Se aplica cuando:

* Nuestro sistema cliente tiene que acceder a parte de la funcionalidad de un sistema complejo.
* Hay tareas o configuraciones muy frecuentes y es conveniente simplificar el código de uso.
* Necesitamos hacer que una librería sea más legible.
* Nuestros sistemas clientes tienen que acceder a varias APIs y queremos simplificar dicho acceso.

Estructura:



Los clientes solo tienen acceso a la fachada de manera “limitar” su visión del programa y no interactuar con todos los subsistemas de manera bruta. El facade como interfaz recopila las distintas funciones disponibles del subsistema y las acopla de manera sencilla para su presentación y uso. De esta manera se convierte en un pilar del que dependen directamente los clientes como su única instancia de comunicación con el sistema.

Ventajas y desventajas de Facade:

| Ventajas | Desventajas |
| --- | --- |
| Minimiza la complejidad de los subsistemas | Aplicación compleja (especialmente en un código existente) |
| Promueve el principio de acoplamiento suelto | La aproximación está acompañada por un nivel adicional de indirección. |
| El software se vuelve más flexible y fácilmente expandible | Alto grado de dependencia en la interfaz de la fachada |