

## Fachada

### 1. Introducción (1 minuto)

Buenas tardes. Hoy les voy a hablar sobre el patrón de diseño Façade o Fachada.

Este patrón pertenece a los patrones de diseño estructurales y es muy usado cuando queremos hacer que un sistema complejo sea más fácil de manejar para los usuarios.

### 2. Nombre y Definición (1 minuto)

El patrón se llama Façade. En español significa 'fachada', como la parte visible de un edificio.

Su intención, según el libro de los GoF (Gang of Four), es proporcionar una interfaz unificada a un conjunto de interfaces dentro de un subsistema.

Es decir, crea una 'cara' sencilla que oculta toda la complejidad que hay detrás.

### 3. Motivación (2 minutos)

Este patrón surge porque muchas veces los sistemas son extremadamente complejos y tienen muchas clases que interactúan entre sí.

Como usuarios o programadores, no siempre queremos ni necesitamos saber todos esos detalles.

Un ejemplo podría ser:

Imaginen que organizamos una fiesta con 100 invitados.

En lugar de encargarnos de la comida, las mesas, la decoración, el sonido, etc., vamos con un organizador de eventos.

Solo le decimos el tipo de fiesta, la fecha, la hora y el número de personas, y él se encarga del resto.

Ese organizador sería nuestra Façade: una forma simple de interactuar con un sistema complejo.”

### 4. Solución (1 minuto)

La solución que propone Façade es crear una interfaz que actúe como intermediario entre el cliente y el sistema.

Esto reduce el acoplamiento, oculta los detalles internos y hace que el sistema sea mucho más fácil de usar.

### 5. Descripción y Cuándo Usarlo (1 minuto)

Se usa Façade cuando:

- Queremos simplificar un subsistema complejo.
- Hay muchas dependencias entre el cliente y las clases internas.
- Queremos agregar una capa de abstracción para facilitar el acceso.

Esto permite que los cambios en el sistema interno no afecten al cliente.

## 6. Estructura (1 minuto)

En cuanto a la estructura estática, tenemos dos tipos de clases:

- Fachada: que conoce qué clases internas son necesarias y las coordina.
- Clases del subsistema: que hacen el trabajo real, pero no saben que existe la fachada.

El cliente nunca interactúa directamente con las clases internas, sólo con la fachada.

## 7. Ejemplo de Participantes (1 minuto)

Un ejemplo clásico sería un compilador:

- La fachada sería el Compilador.
- Las clases del subsistema serían cosas como el Scanner, el Parser, o el ProgramNode.

El usuario le dice 'Compila esto', y el compilador se encarga de escanear, analizar y ejecutar, sin que el usuario tenga que preocuparse por esos detalles.

## 8. Consecuencias y Ventajas (1 minuto)

Las principales ventajas de usar Façade son:

- Simplifica el uso de sistemas complejos.
- Reduce el acoplamiento, lo que hace que sea más fácil cambiar el sistema interno sin romper el exterior.
- Facilita la portabilidad a otras plataformas.
- Organiza mejor el sistema en capas.

Además, aunque existe una Façade, los programadores avanzados todavía pueden acceder a las clases internas si lo necesitan.

## 9. Consideraciones Adicionales (30 segundos)

Para reducir aún más el acoplamiento, podemos hacer que la fachada sea abstracta, permitiendo distintas implementaciones.

También ayuda a manejar el ocultamiento de datos: algunas clases pueden ser públicas (como botones), mientras otras son privadas (como circuitos internos).

## 10. Explicación de ejemplo clásico (1 minuto)

Imaginemos que tenemos un teatro en casa, para poder ver una película debemos llevar a cabo ciertos pasos como: encender el proyector, bajar la pantalla, encender el reproductor blu-ray, encender el sistema de sonido y apagar las luces, estamos interactuando con cinco clases, al usar una fachada que se encargue de hacer todo eso solo interactuamos con ella.

## 11. Explicación de diseño (1 minuto)

Si tenemos un subsistema con varias clases la fachada interactúa con ellos, para que así el cliente interactúe con la fachada.

También tenemos un antes y un después, sin la fachada el mismo diagrama se ve caótico y poco elegante, es difícil de leer, sin embargo con la façade aumenta demasiado la legibilidad del diagrama y también será más fácil construir lo representado.

#### 12. Explicación de construcción (3 minutos)

Se tiene un sistema de sonido para guitarra, este tiene un amplificador, cables, pedales y una guitarra, nuestra fachada se encargará de encender cada uno de estos, conectarlos y configurarlos. Gracias a la façade el usuario solo debe “Encender el sistema” para tener todo listo, de la misma manera puede crear perfiles de configuración y cargarlos para que, con la façade, se configure todo automáticamente.

#### 13. Cierre (30 segundos)

En resumen, el patrón Façade nos ayuda a trabajar con sistemas grandes de manera simple, ordenada y segura.

Es una herramienta poderosa para construir sistemas que sean fáciles de usar y mantener.

¡Gracias por su atención!