

Patrones Estructurales

Unidad 3

Patrones de Diseño de Software

Introducción

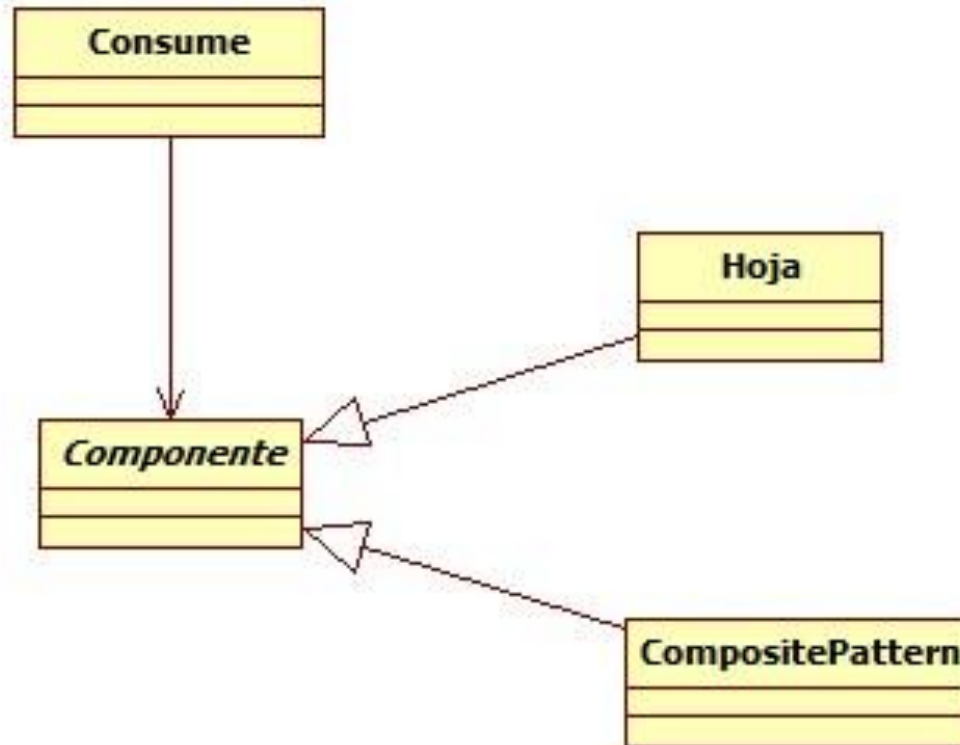
Patrones Estructurales

- Su función es determinar como las clases y objetos se combinan para formar estructuras
- Estas estructuras permitirán que se agreguen nuevas funcionalidades
- Los patrones *GoF* que se clasifican como Estructurales son:
 - *Composite*
 - *Facade*
 - *Adapter*
 - *Proxy*
 - *Decorator*
 - *Bridge*

Patrón *Composite* (Compuesto)

- Se utiliza cuando un componente del sistema puede ser un objeto individual o una colección de objetos
- Modela una estructura de árbol en donde un elemento tiene elementos del mismo tipo
- Una composición puede ser una colección de objetos en donde cada uno de estos objetos puede ser una composición o un objeto sencillo
- Una interfaz o clase abstracta define lo que todos los objetos deben hacer ya sea que se trate de una hoja o una composición
- La hoja implementa lo que debe hacer, mientras que la composición adicionalmente debe manejar la colección

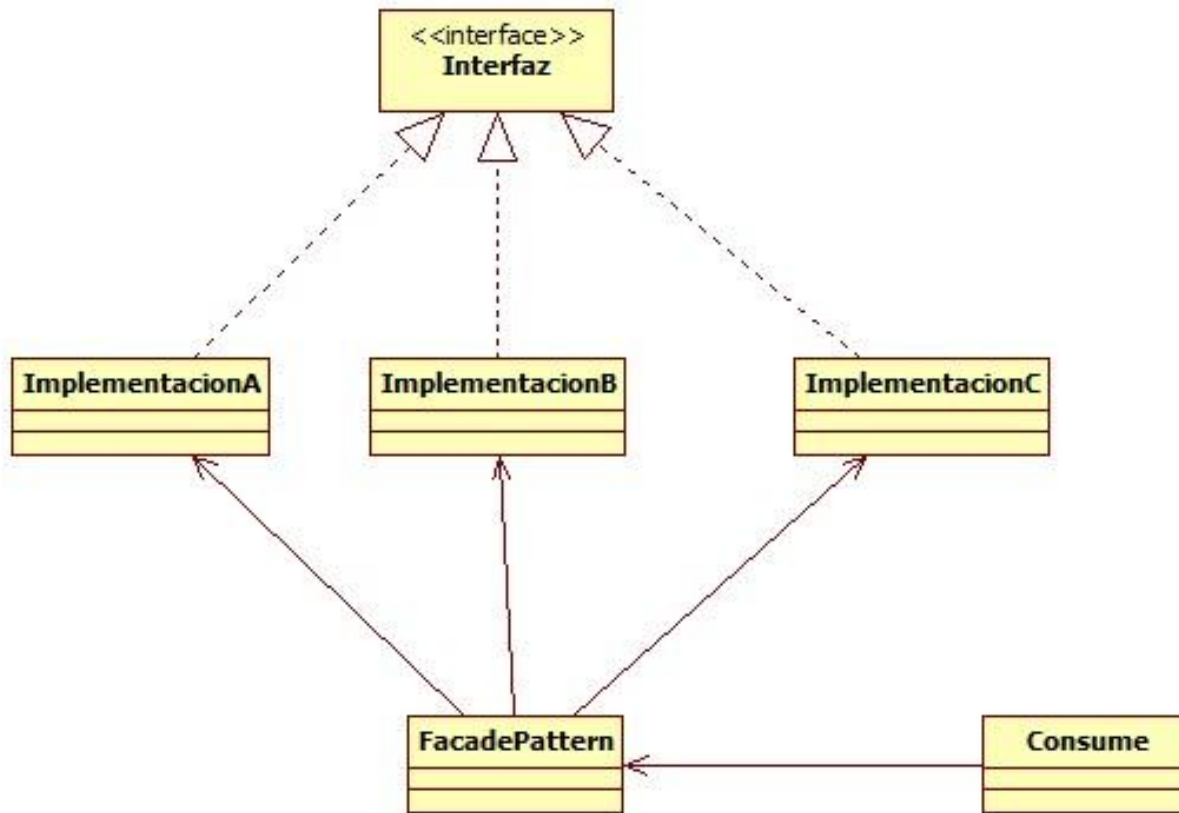
Esquema del Patrón *Composite*



Patrón *Facade* (Fachada)

- Su objetivo es ocultar la complejidad de un sistema y proporcionar una interfaz al cliente que accede al sistema
- Se compone de una clase que ofrece métodos sencillos para el cliente y la implementación se da en otras clases

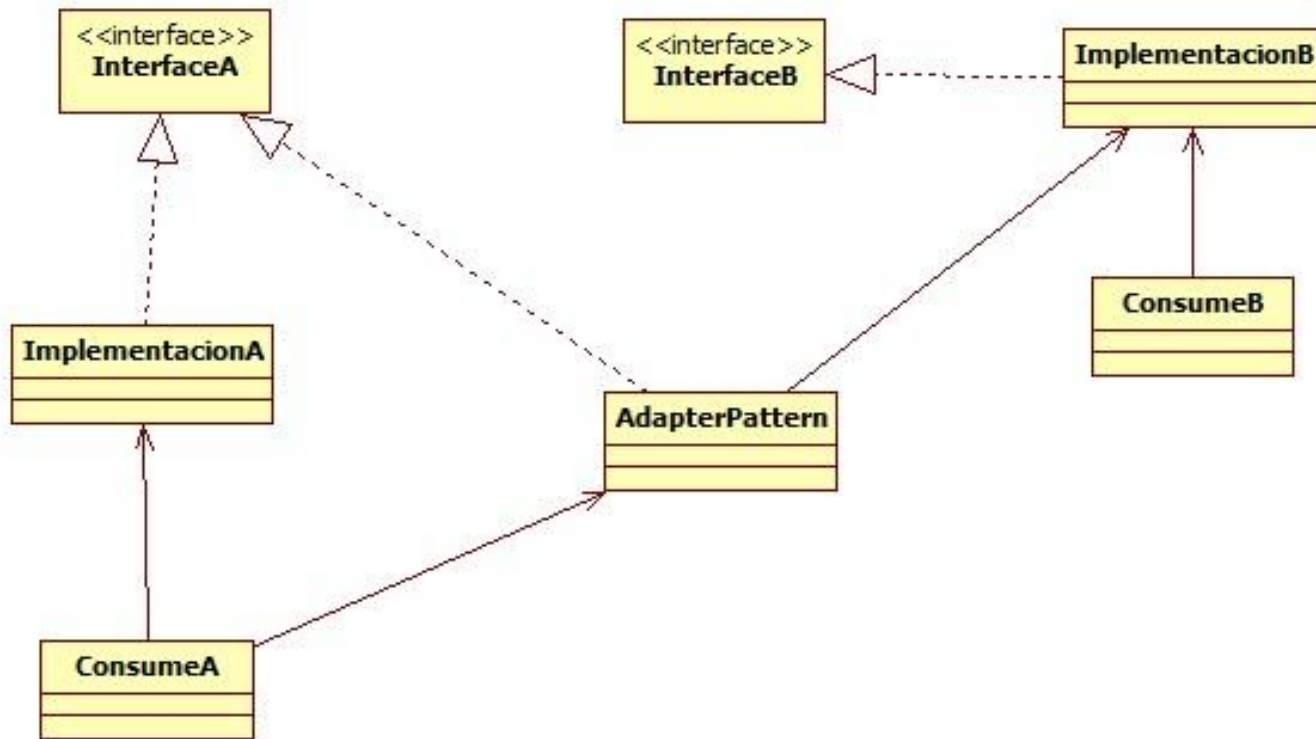
Esquema del Patrón *Facade*



Patrón *Adapter* (Adaptador)

- Permite convertir la Interfaz (Interface) de un objeto en otra (Interfaz) que los clientes pueden utilizar
- Esto permite que varias clases trabajen juntas a pesar de no tener interfaces compatibles
- Es una nueva clase que es responsable de combinar funcionalidades de interfaces independientes o que no son compatibles

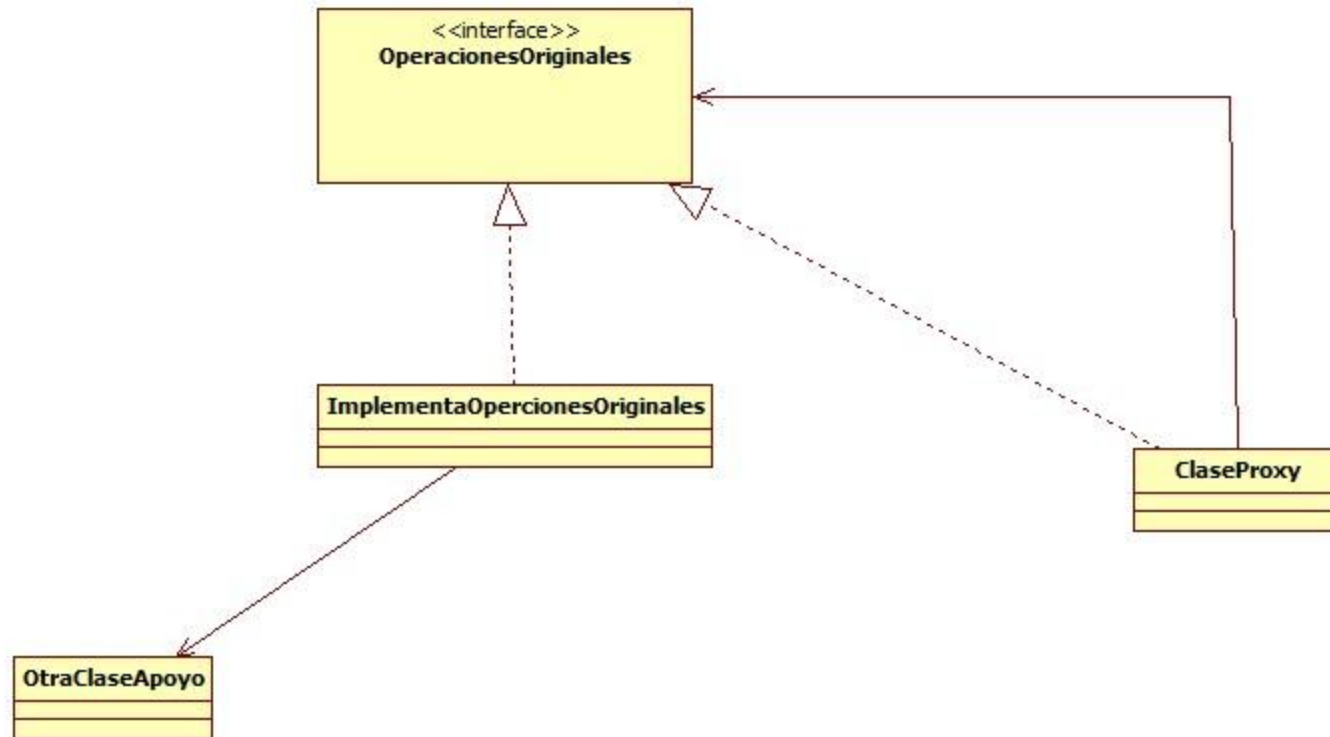
Esquema del Patrón *Adapter*



Patrón *Proxy*

- Se trata de una clase que sirve como intermediario para dar funcionalidad adicional a un método de una clase sin modificar su estructura

Esquema del Patrón *Proxy*



Patrón *Decorator*

- Permite agregar nueva funcionalidad a un objeto sin modificar su estructura
- Para esto se crea una clase conocida como Decoradora

Esquema del Patrón *Decorator*

