



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	Miércoles, 07/08/2024
Hora de inicio:	15:40

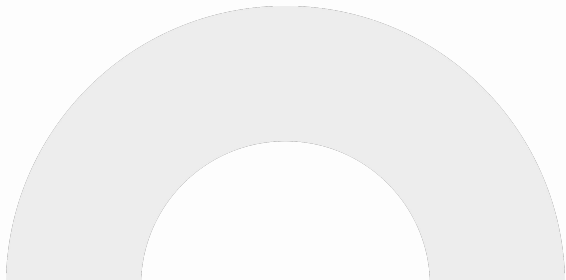
Análisis y Diseño de Sistemas 2 [B]

Luis Angel Barrera Velásquez



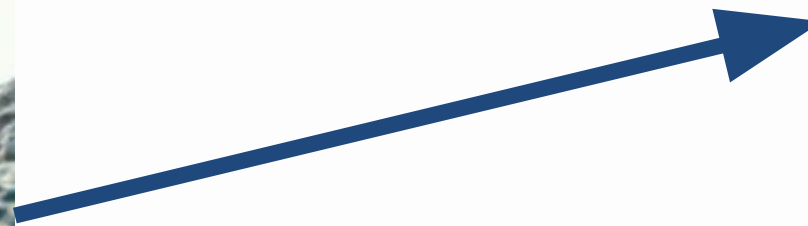
AGENDA

1	Avisos
2	Horarios de Calificación
3	Teoría Patrones de diseño
4	Ejemplo Práctico



Patrones de diseño de software

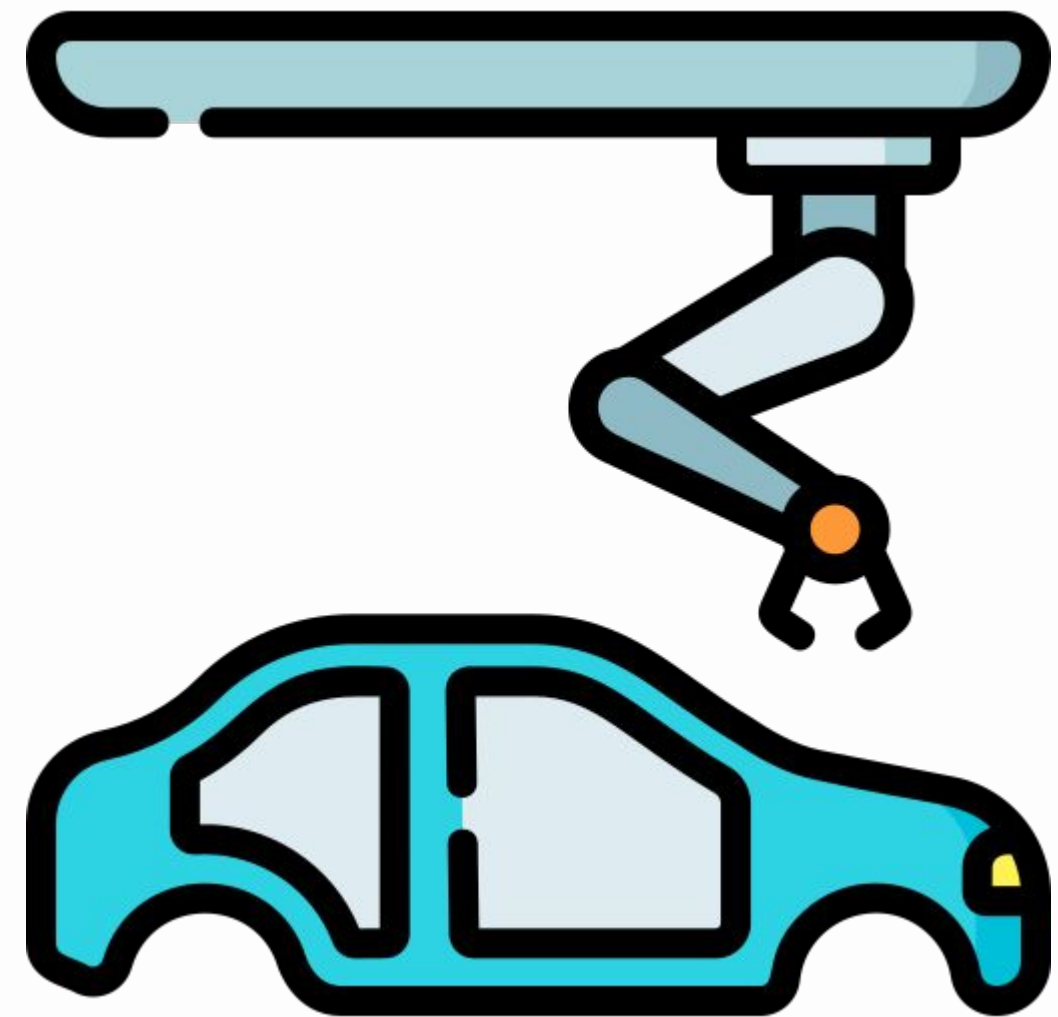
Los patrones de diseño son soluciones reutilizables a problemas comunes que enfrentan los diseñadores y desarrolladores de software. Estos patrones proporcionan una estructura para elaborar diseños, lo que permite la creación de soluciones eficientes y mantenibles.



Tipos de Patronos de diseño

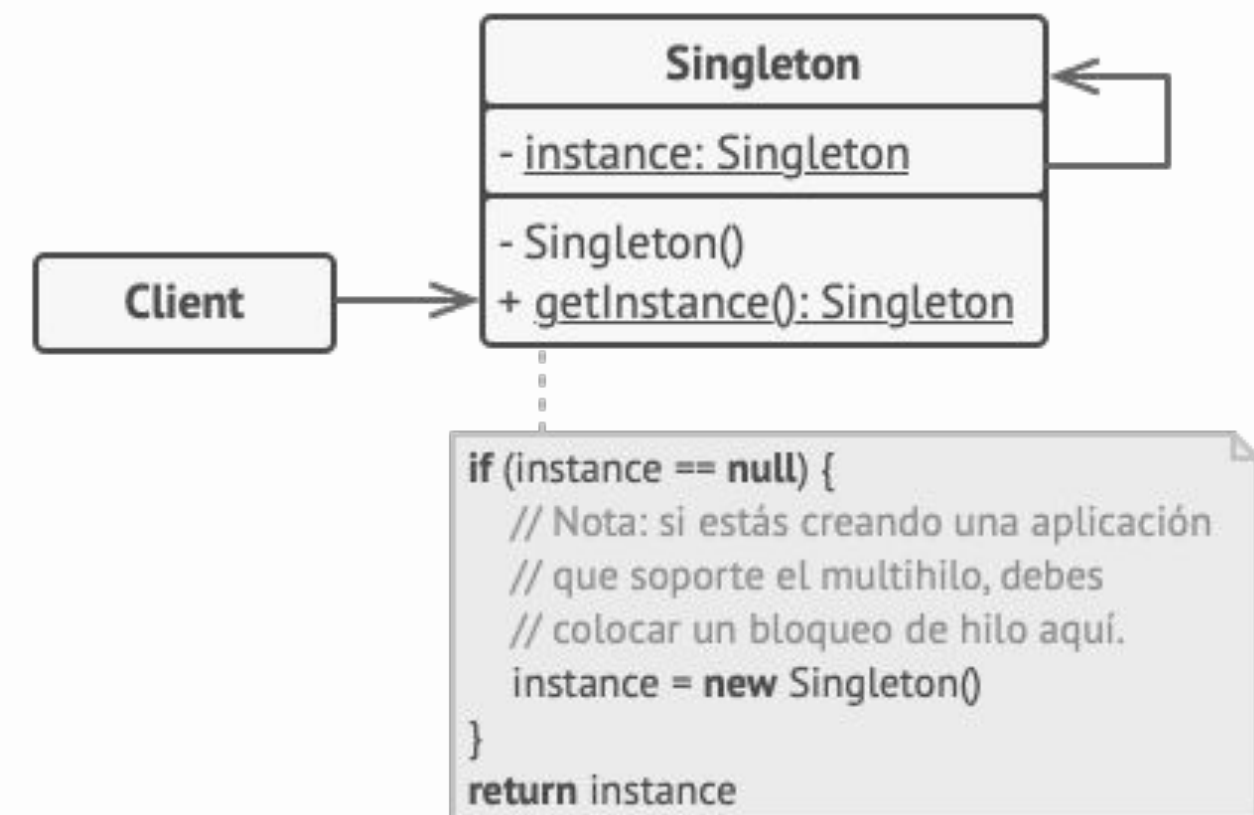
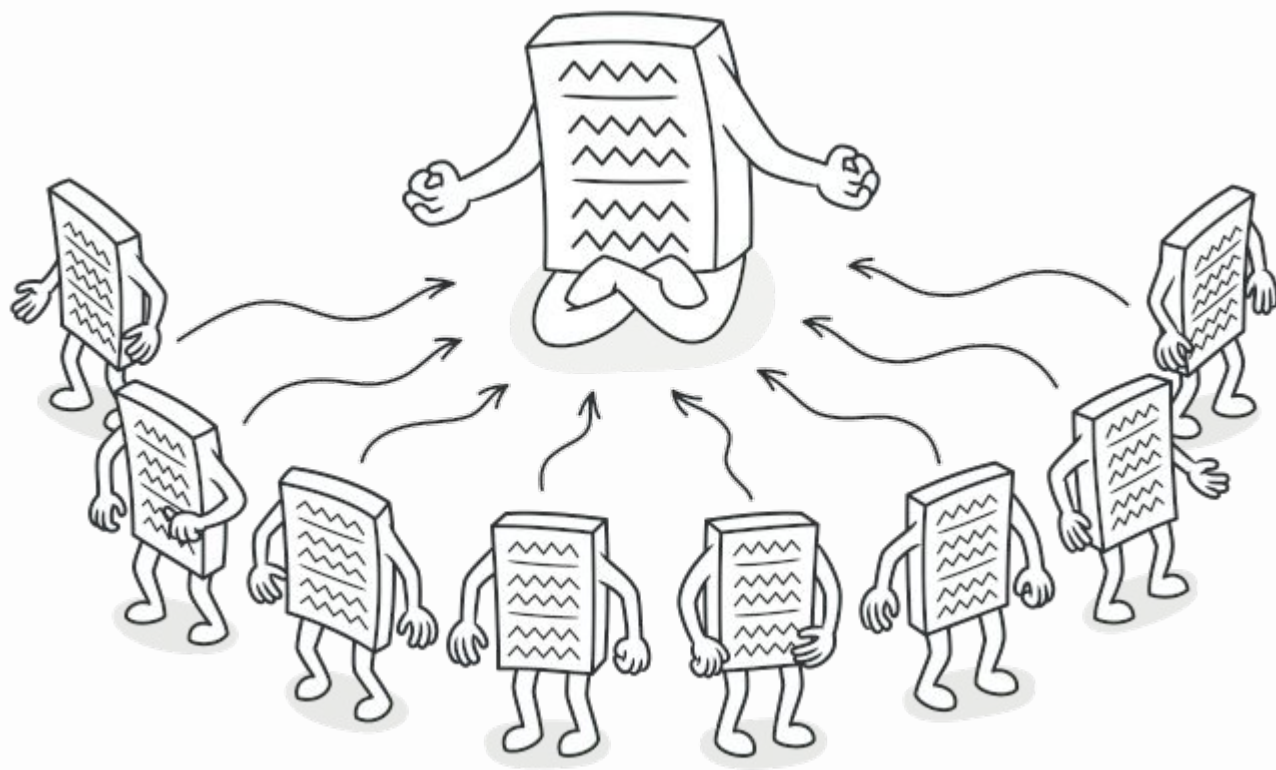
Patrones Creacionales

Son patrones de diseño relacionados con la creación o construcción de objetos. Estos patrones intentan controlar la forma en que los objetos son creados implementando mecanismos que eviten la creación directa de objetos.



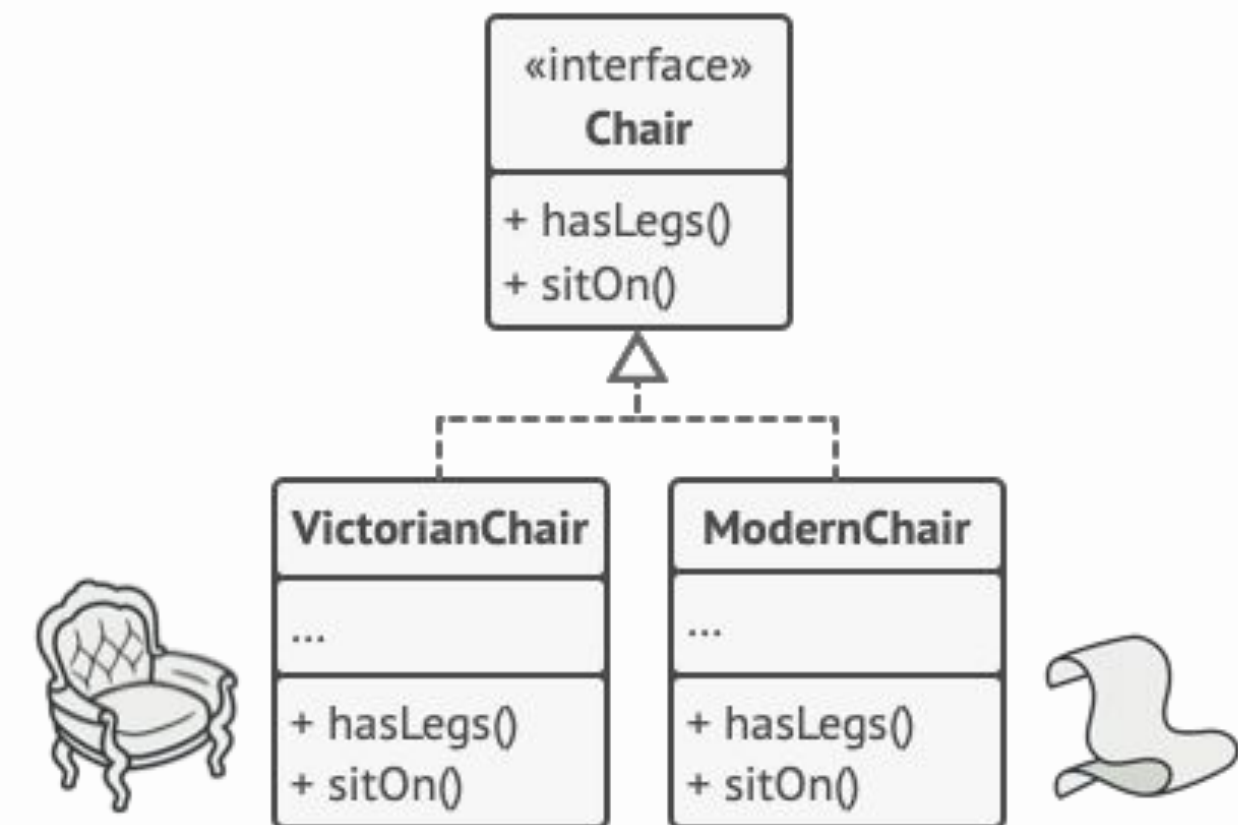
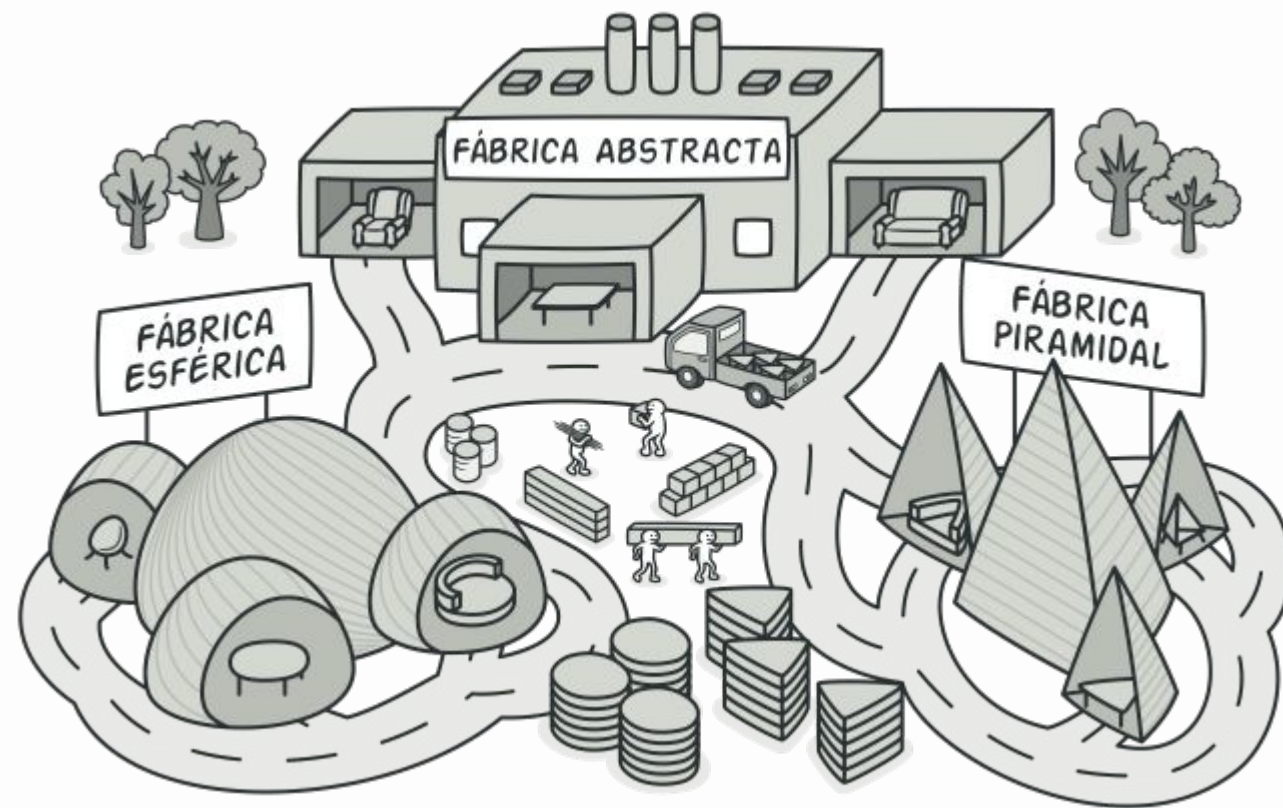
Patrón Singleton

Singleton es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.



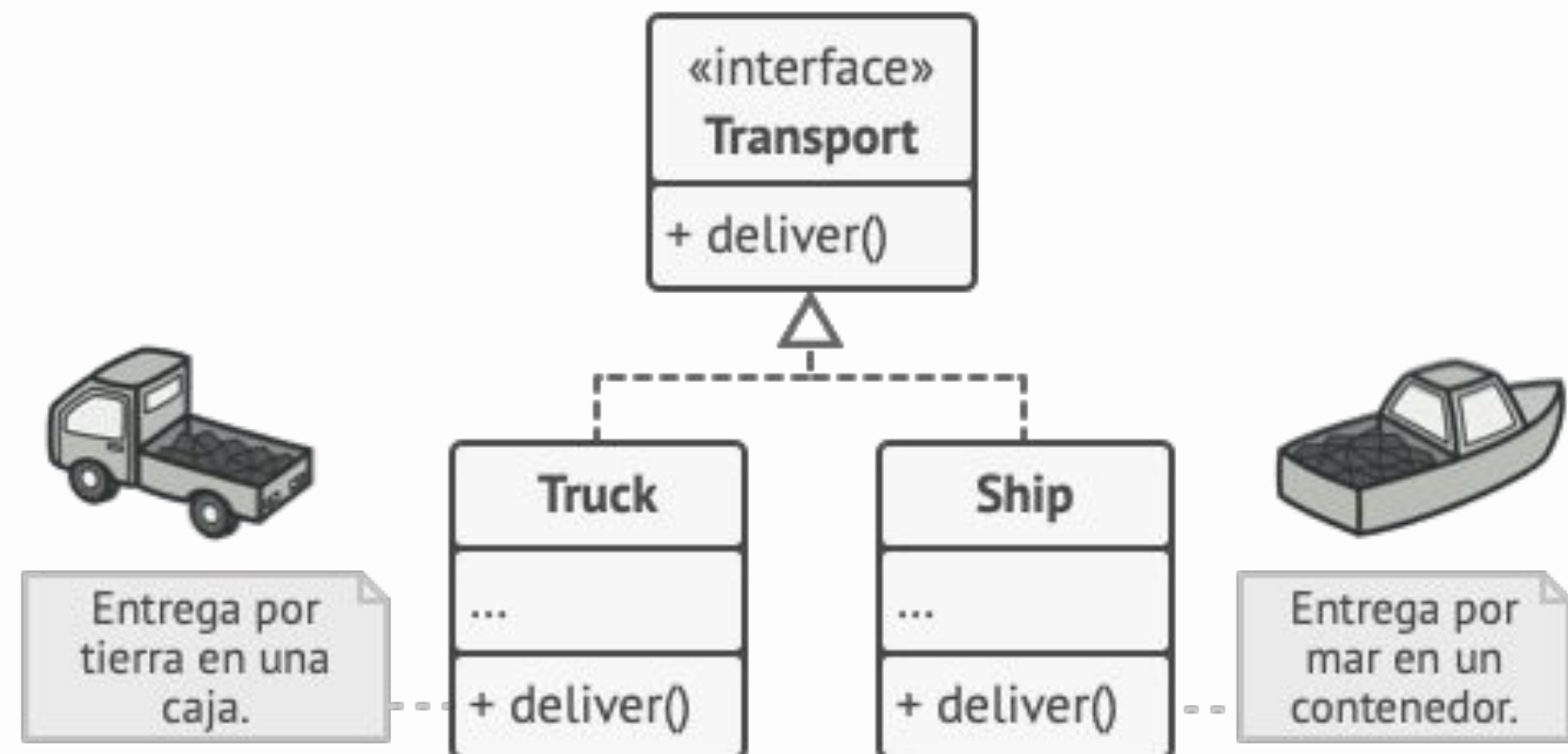
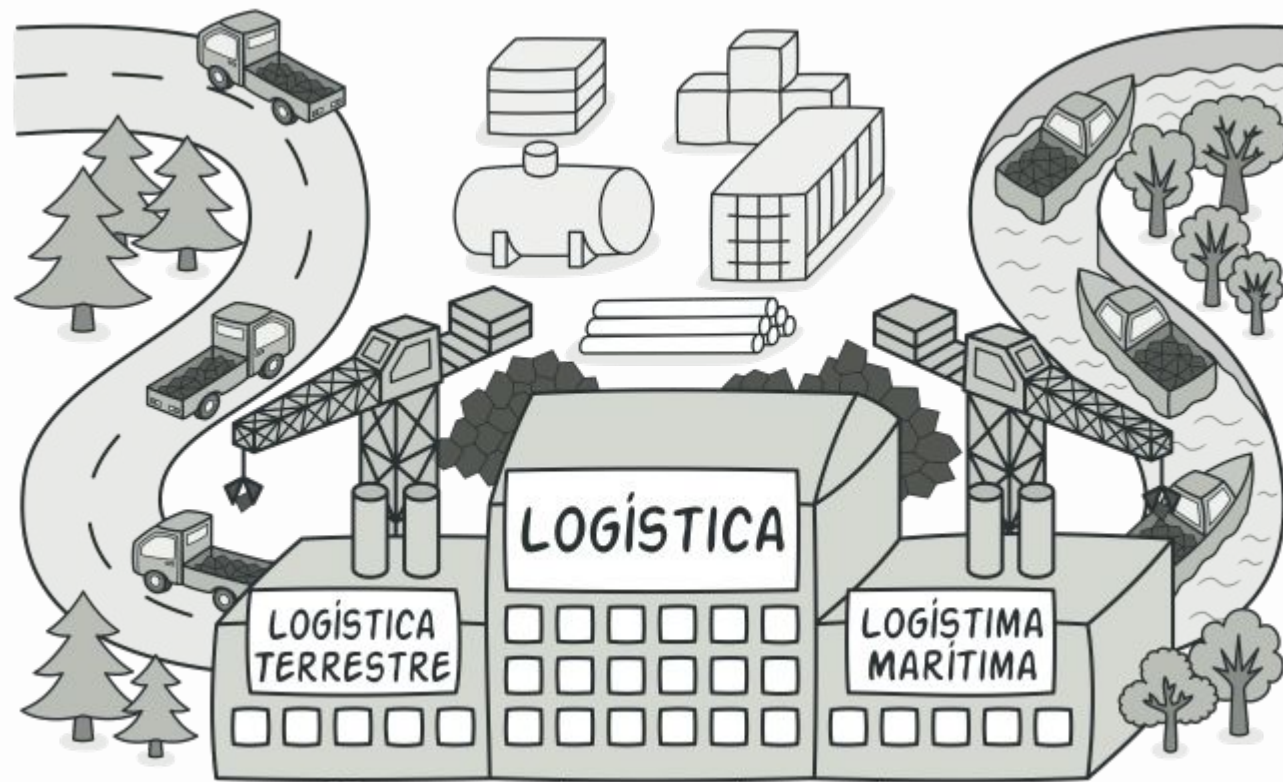
Patrón Abstract Factory

Abstract Factory es un patrón de diseño creacional que nos permite producir familias de objetos relacionados sin especificar sus clases concretas.



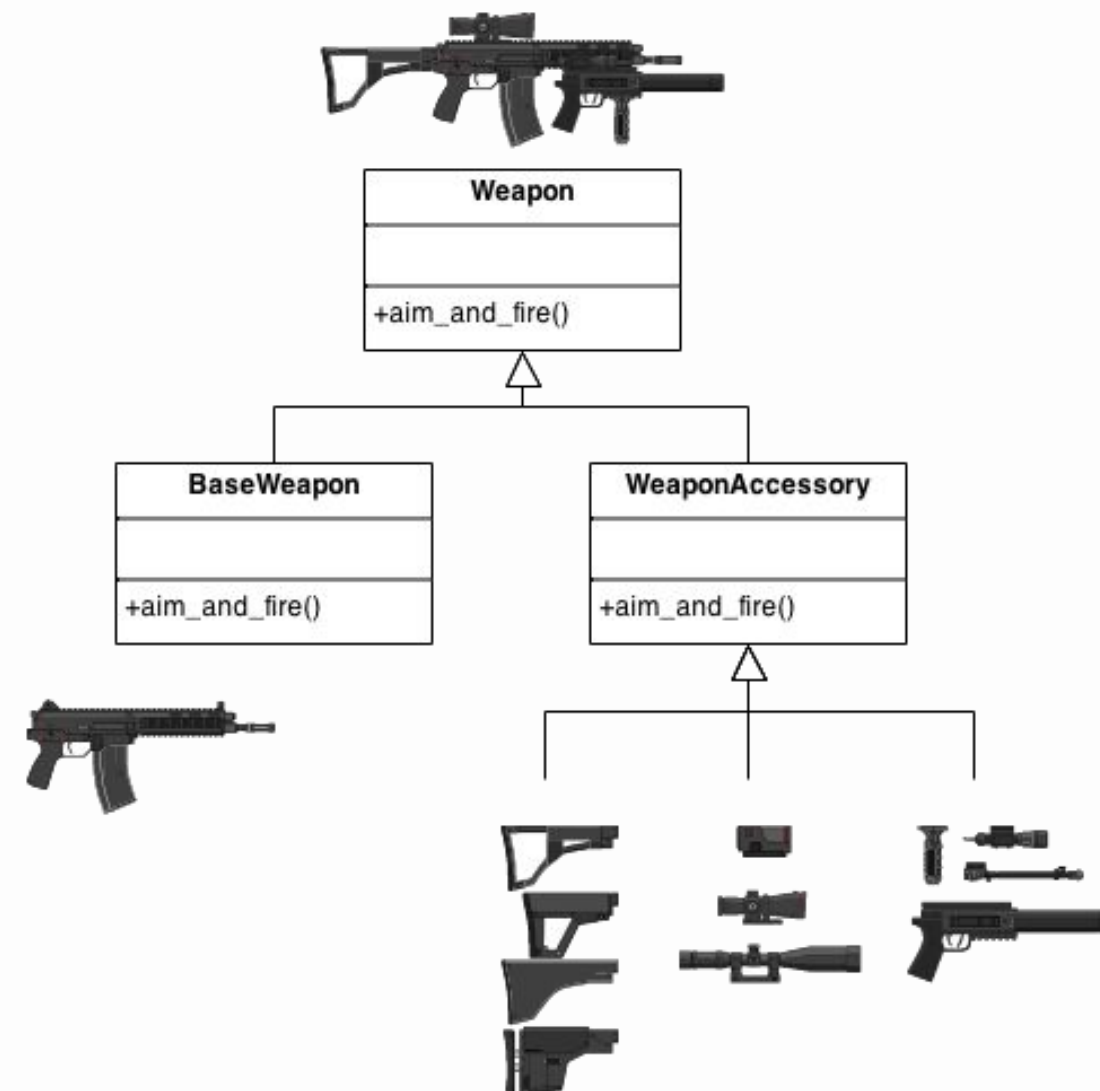
Patrón Factory Method

Factory Method es un patrón de diseño creacional que proporciona una interfaz para crear objetos en una superclase, mientras permite a las subclases alterar el tipo de objetos que se crearán.



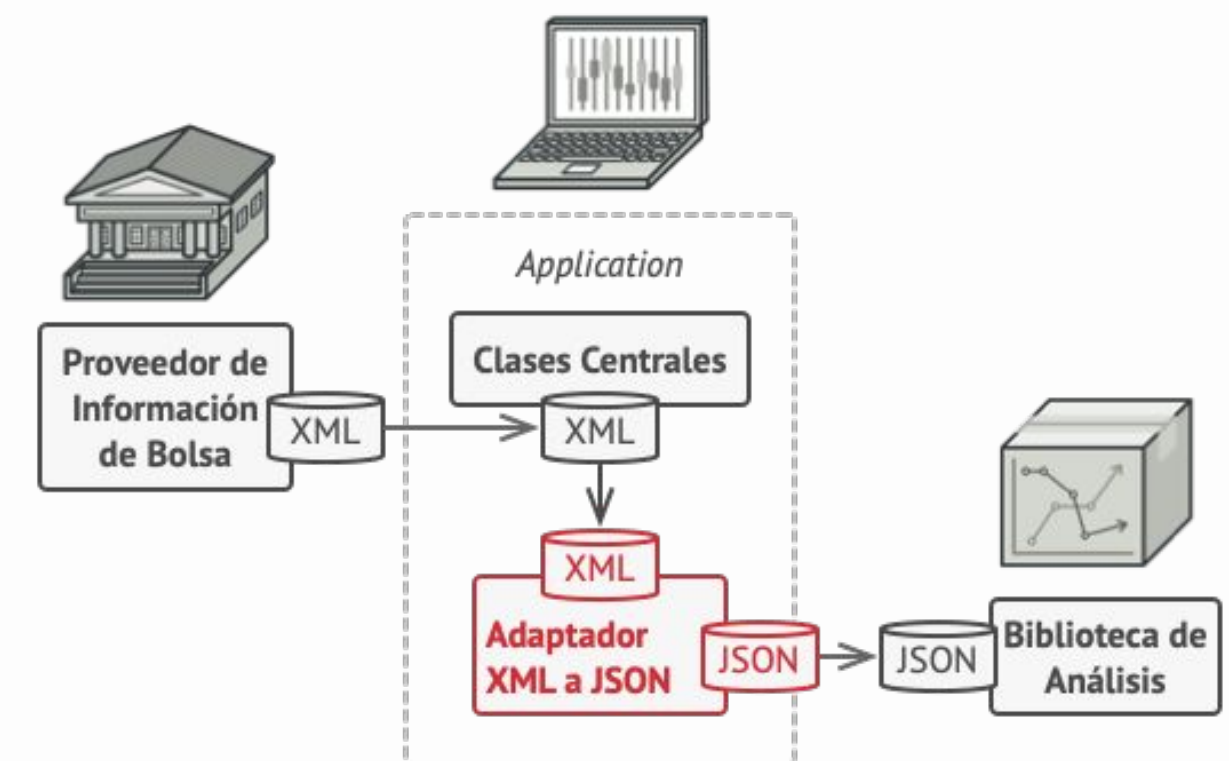
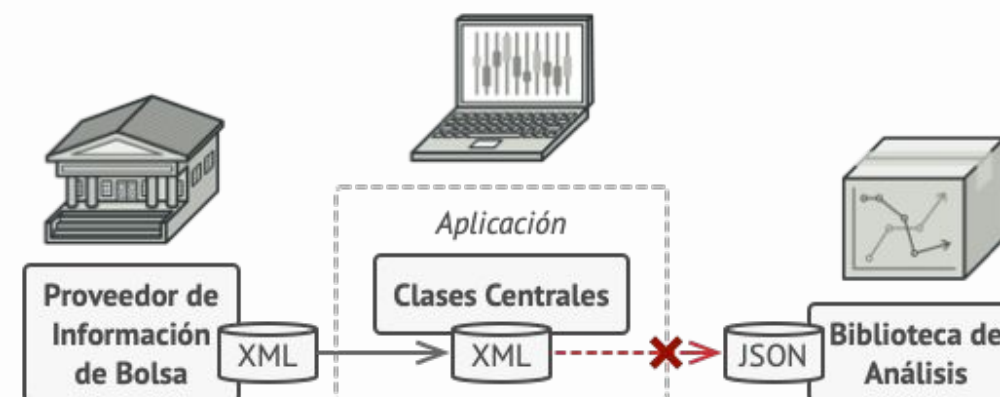
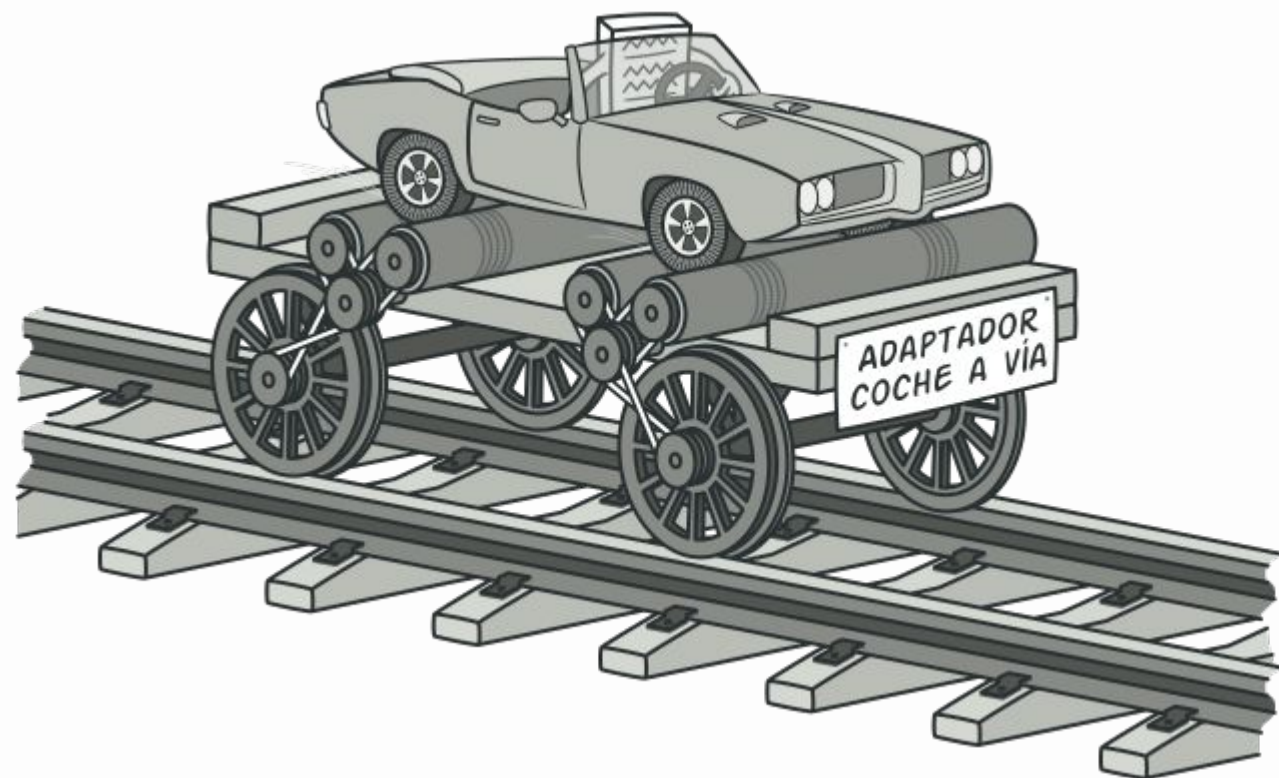
Patrones Estructurales

Son patrones que tiene que ver con la forma en que las clases se relacionan con otras clases. Estos patrones ayudan a dar un mayor orden a nuestras clases ayudando a crear componentes más flexibles y extensibles.



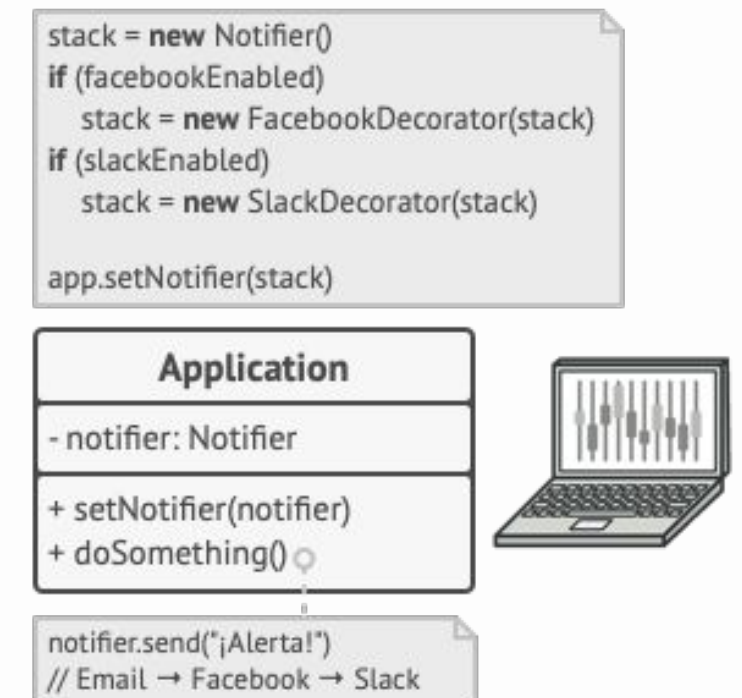
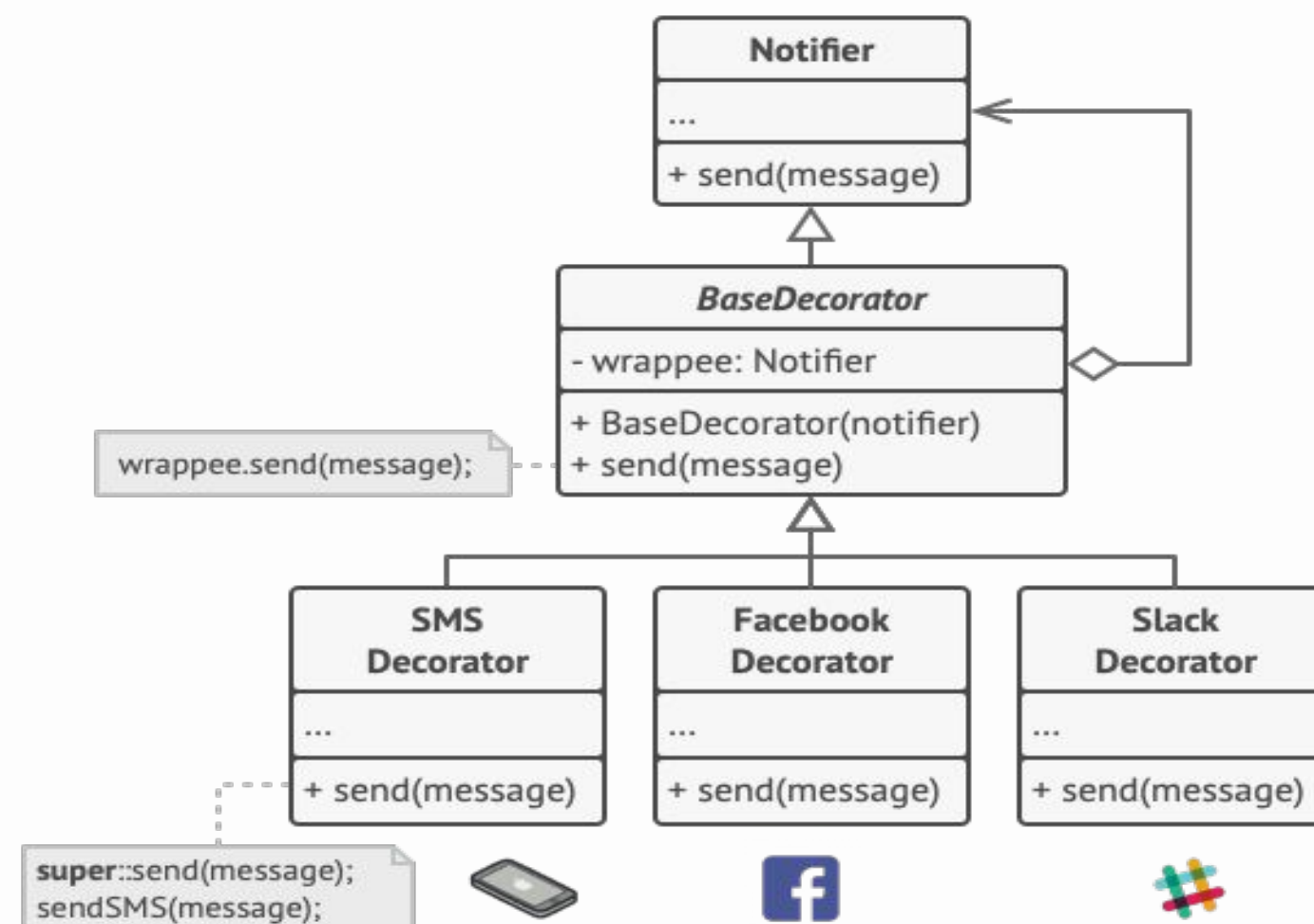
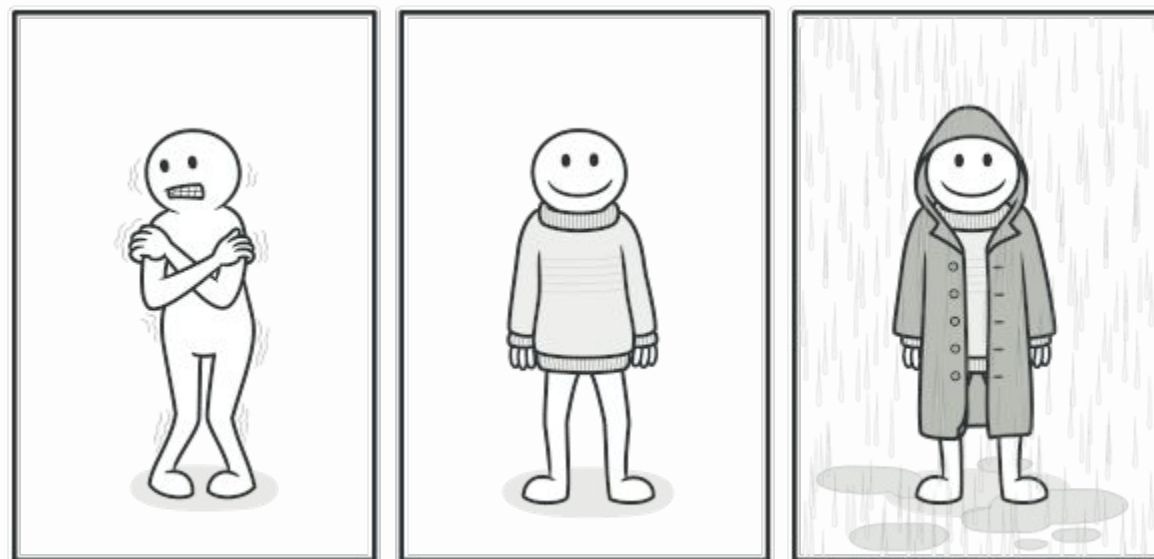
Patrón Adapter

Adapter es un patrón de diseño estructural que permite la colaboración entre objetos con interfaces incompatibles.



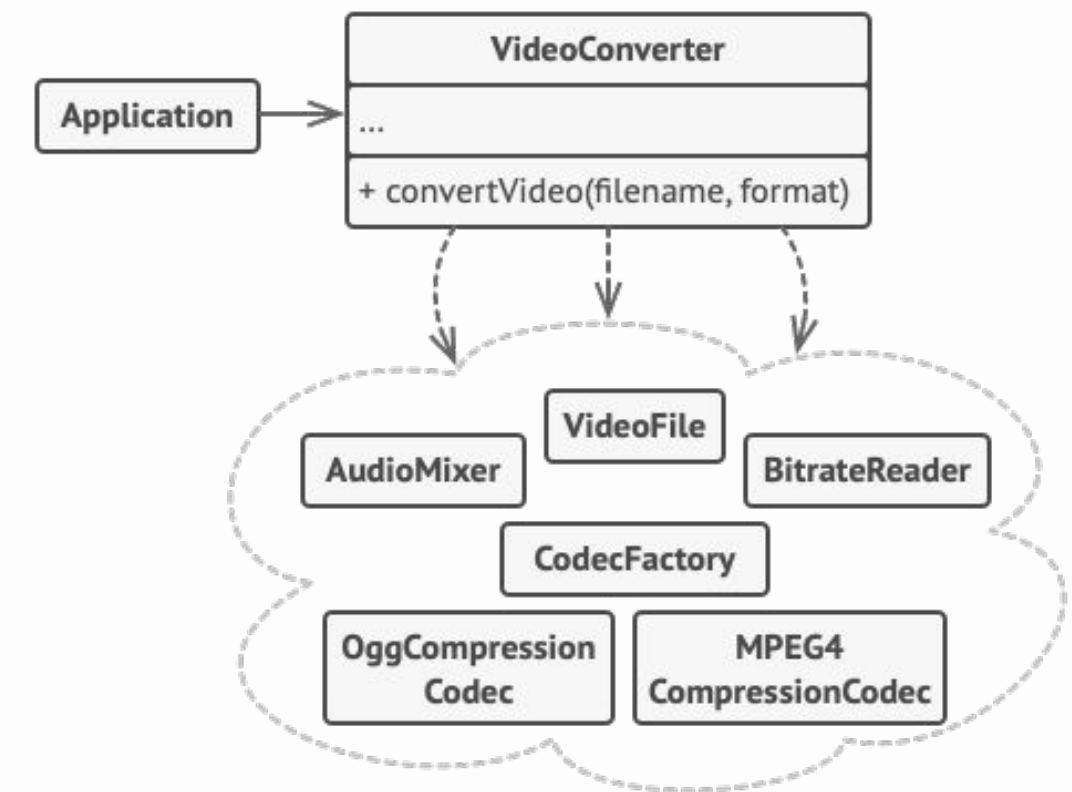
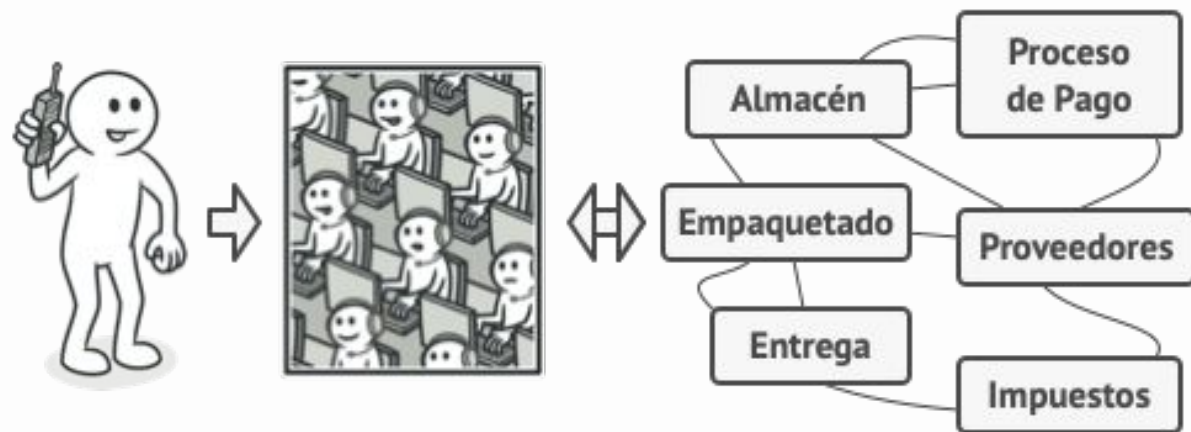
Patrón Decorator

Decorator es un patrón de diseño estructural que te permite añadir funcionalidades a objetos colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades.



Patrón Facade

Facade es un patrón de diseño estructural que proporciona una interfaz simplificada a una biblioteca, un framework o cualquier otro grupo complejo de clases.



Patrón Proxy

Proxy es un patrón de diseño estructural que te permite proporcionar un sustituto o marcador de posición para otro objeto. Un proxy controla el acceso al objeto original, permitiéndote hacer algo antes o después de que la solicitud llegue al objeto original.

