Patrones GOF

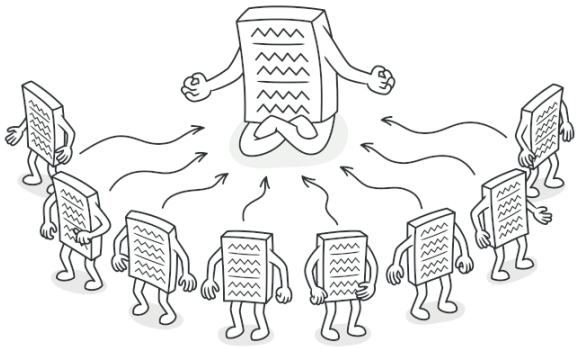
Los patrones GoF **(Gang of Four)** son un conjunto de patrones de diseño de software que se utilizan para resolver problemas comunes de diseño de software. Incluyen patrones creacionales, estructurales y de comportamiento.

Los patrones creacionales se utilizan para crear objetos de manera eficiente y flexible.

Ejemplos de patrones GOF:

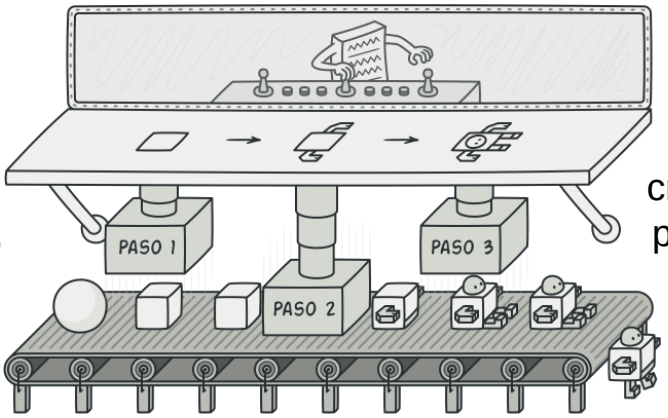
Patrones de creación:

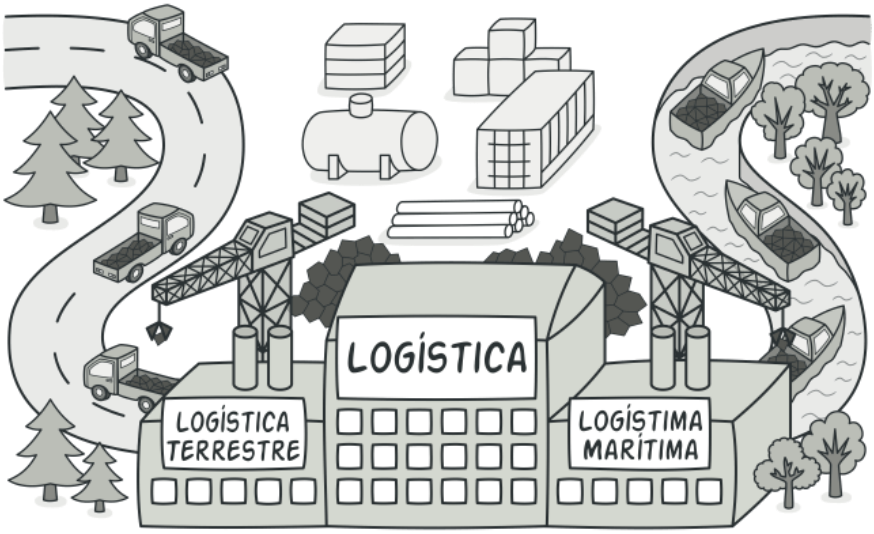
- Singleton: Garantiza que solo haya una instancia de una clase y proporciona un punto de acceso global a esa instancia.



- Builder: implica dividir la construcción de un objeto complejo en una serie de pasos más simples, de modo que cada paso pueda ser realizado por un objeto distinto.

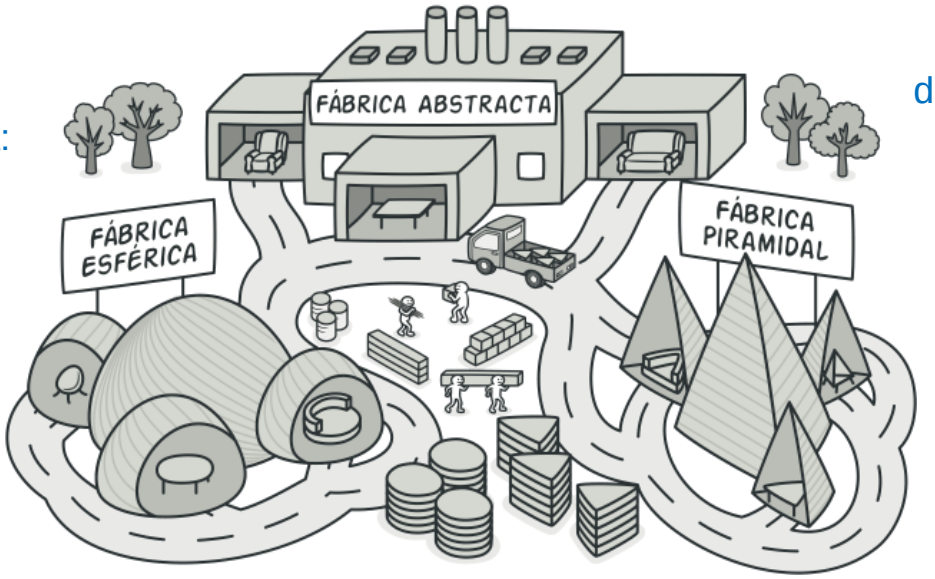
- Factory Method: Define una interfaz para crear objetos en una clase, pero permite a las subclases decidir qué clase instanciar.



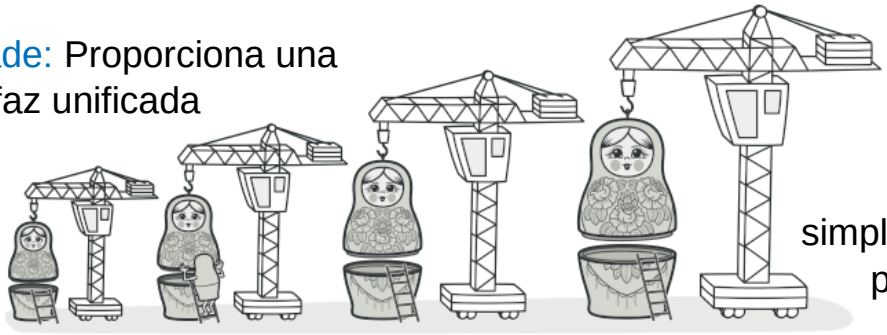


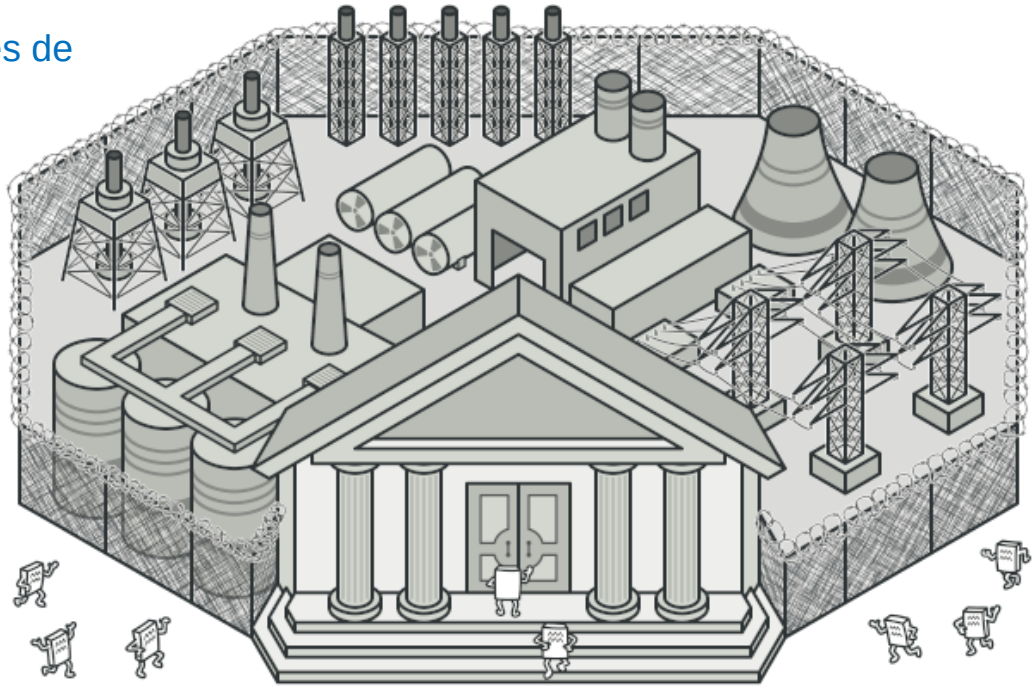
- Abstract Factory: Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas.

Patrones de estructura:



- Decorator: Agrega comportamiento a un objeto existente dinámicamente sin cambiar su interfaz.

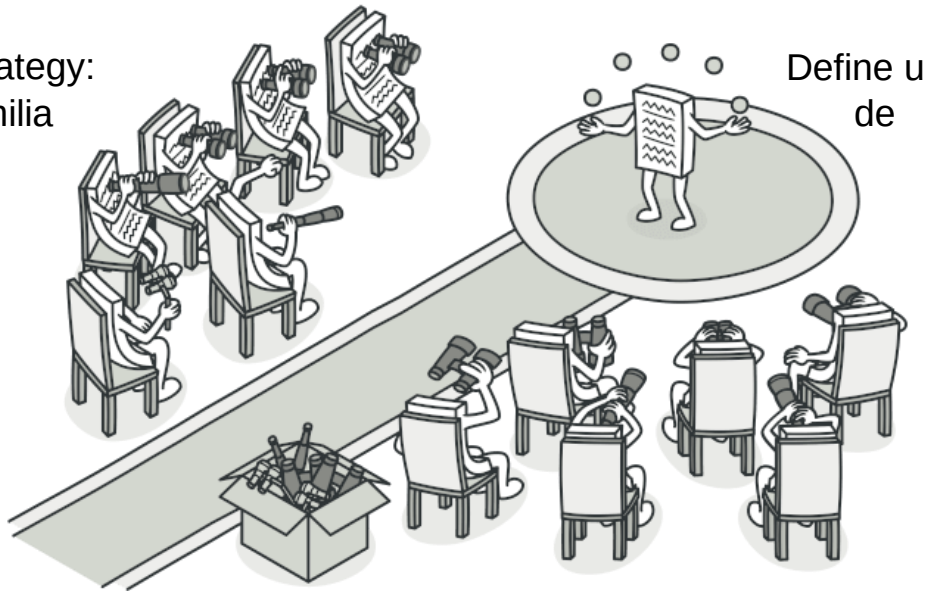
- Facade: Proporciona una interfaz unificada  y simplificada para un conjunto de complejas  interfaces más en un subsistema.

Patrones de 

comportamiento:

- Observer: Define una dependencia uno a muchos entre objetos, de modo que cuando uno cambia su estado, todos los dependientes son notificados y actualizados automáticamente.

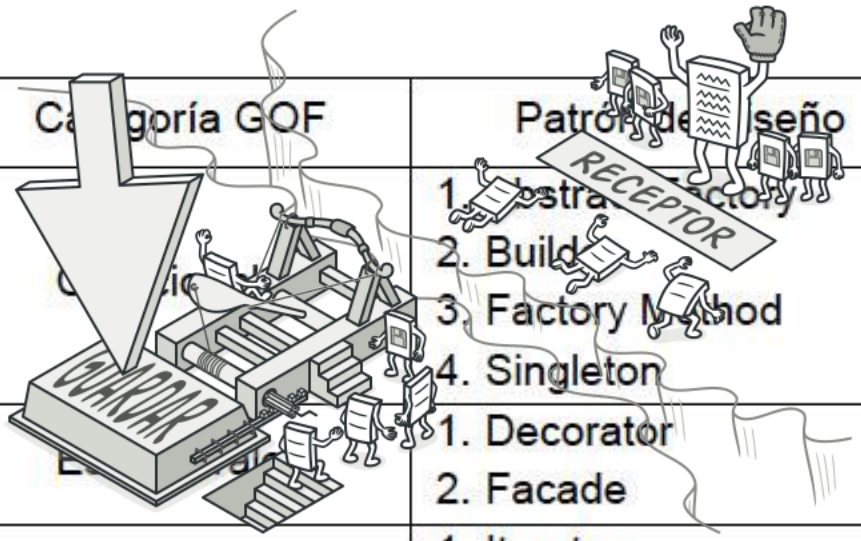
- Strategy: Define una familia de



algoritmos, encapsula cada uno y los hace intercambiables. Permite que el algoritmo varíe independientemente de los clientes que lo usan.

- Command: Encapsula una solicitud como un objeto, lo que le permite parametrizar a los clientes con diferentes solicitudes, hacer cola o registrar solicitudes y admitir operaciones reversibles.

Los tipos de



Categoría GOF	Patrón de diseño
	1. Abstract Factory 2. Builder 3. Factory Method 4. Singleton
	1. Decorator 2. Facade
De Comportamiento	1. Iterator 2. Observer 3. Strategy 4. Template Method

prueba de software, hay varios enfoques que se pueden utilizar para probar software. Algunos de los tipos más comunes de prueba de software incluyen:

1. Pruebas unitarias: Estas pruebas se centran en probar componentes individuales del software para garantizar que funcionen correctamente.

2. Pruebas de integración: Estas pruebas se centran en probar cómo funcionan los diferentes componentes del software cuando se combinan.

3. Pruebas de sistema: Estas pruebas se centran en probar el software en su conjunto para garantizar que se cumplan los requisitos del cliente.

4. Pruebas de aceptación: Estas pruebas se realizan para garantizar que el software cumpla con los requisitos del cliente y que se ajuste a su propósito.

5. Pruebas de regresión: Estas pruebas se realizan para asegurarse de que las modificaciones o mejoras al software no hayan afectado la funcionalidad previamente probada.