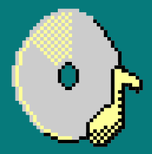
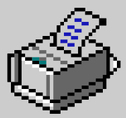
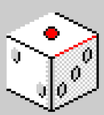



Decorator

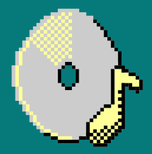
GoF Patterns
Sergio Montoya Badilla
Diseño de software

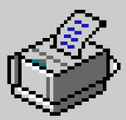


3:00 p. m.

¿Qué es?



Es un patrón estructural, también conocido como Wrapper, cuyo fin es el de asignar responsabilidades adicionales a un objeto dinámicamente



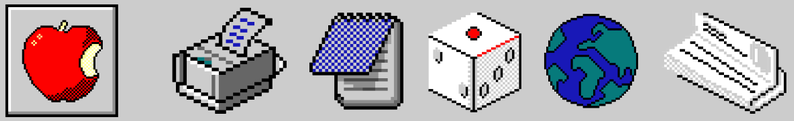
3:00 pm



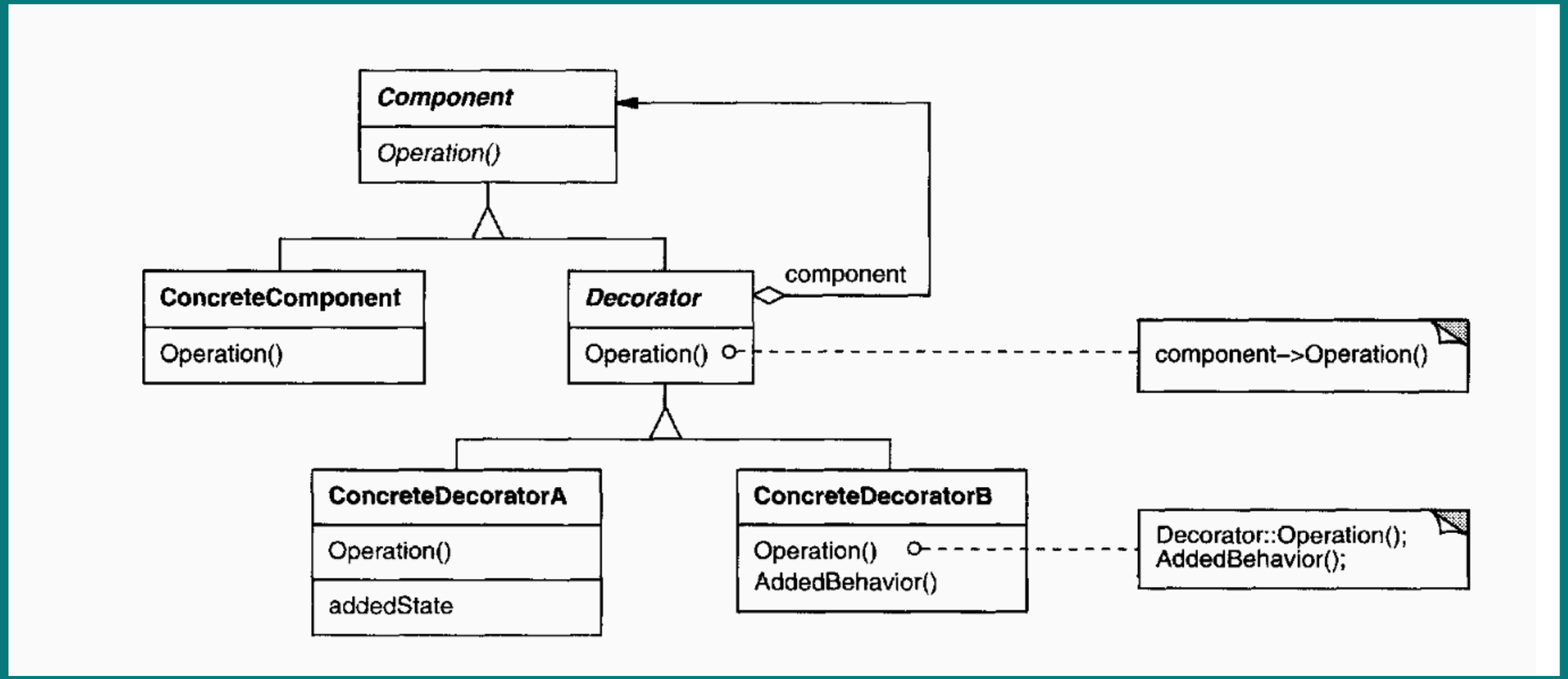

Es útil cuando

- La herencia no es viable, por lo que brinda una alternativa.
- Se desea añadir responsabilidades a objetos individuales sin afectar a los otros objetos.





3:00 pm


Component
Operation()
ConcreteComponent
Operation()
Decorator
Operation()
component
component->Operation()
ConcreteDecoratorA
Operation()
addedState
ConcreteDecoratorB
Operation()
AddedBehavior()
Decorator::Operation();
AddedBehavior();

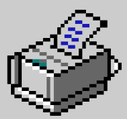



Ayuda a cumplir



- Principio de responsabilidad única: Cada decorador concreto tiene una sola razón de ser

- Principio abierto-cerrado: No es necesario modificar código para añadir nuevas funcionalidades

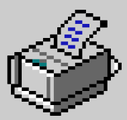





3:00 pm

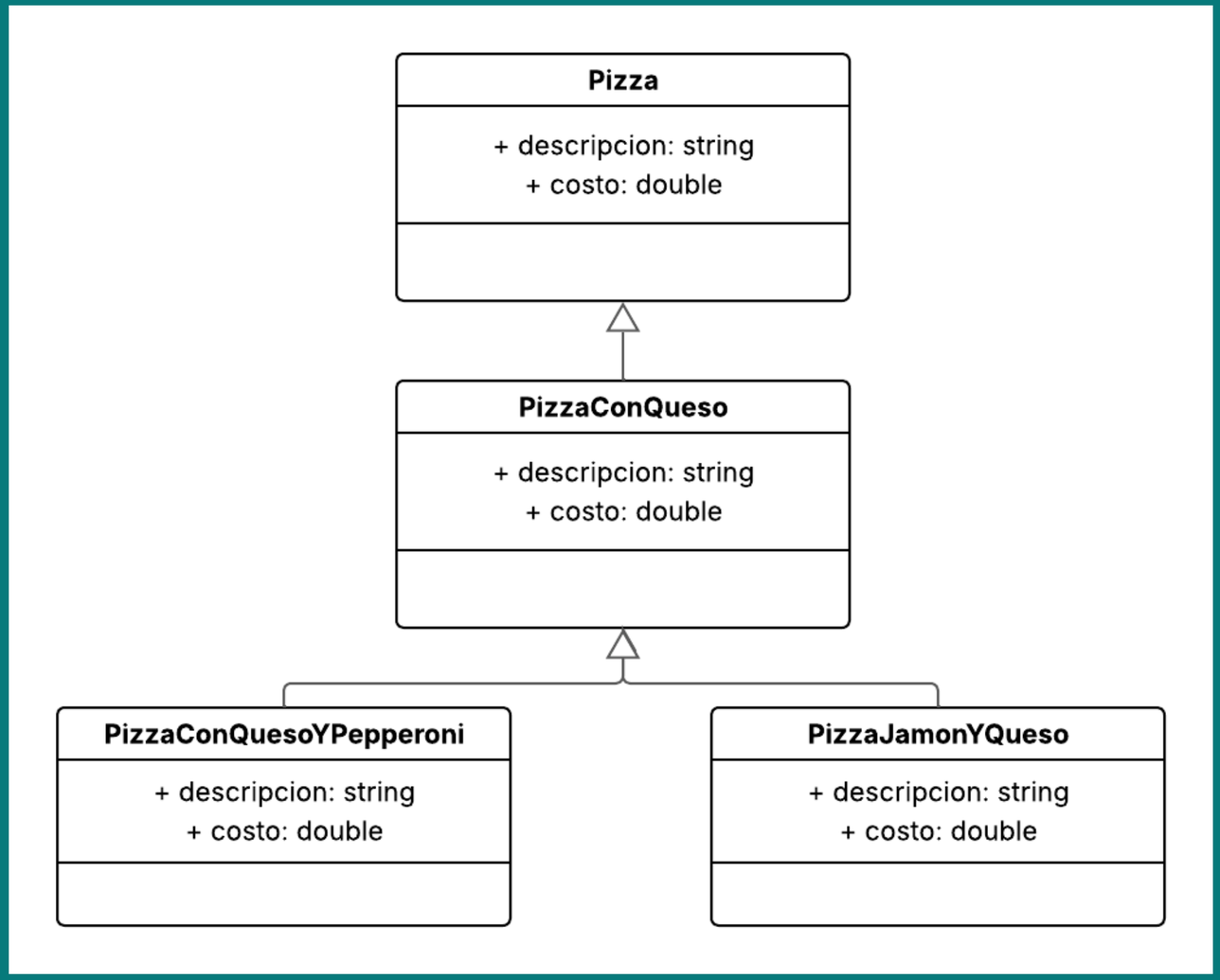
Ejemplos

Tenemos una Pizza básica y queremos agregarle varios ingredientes



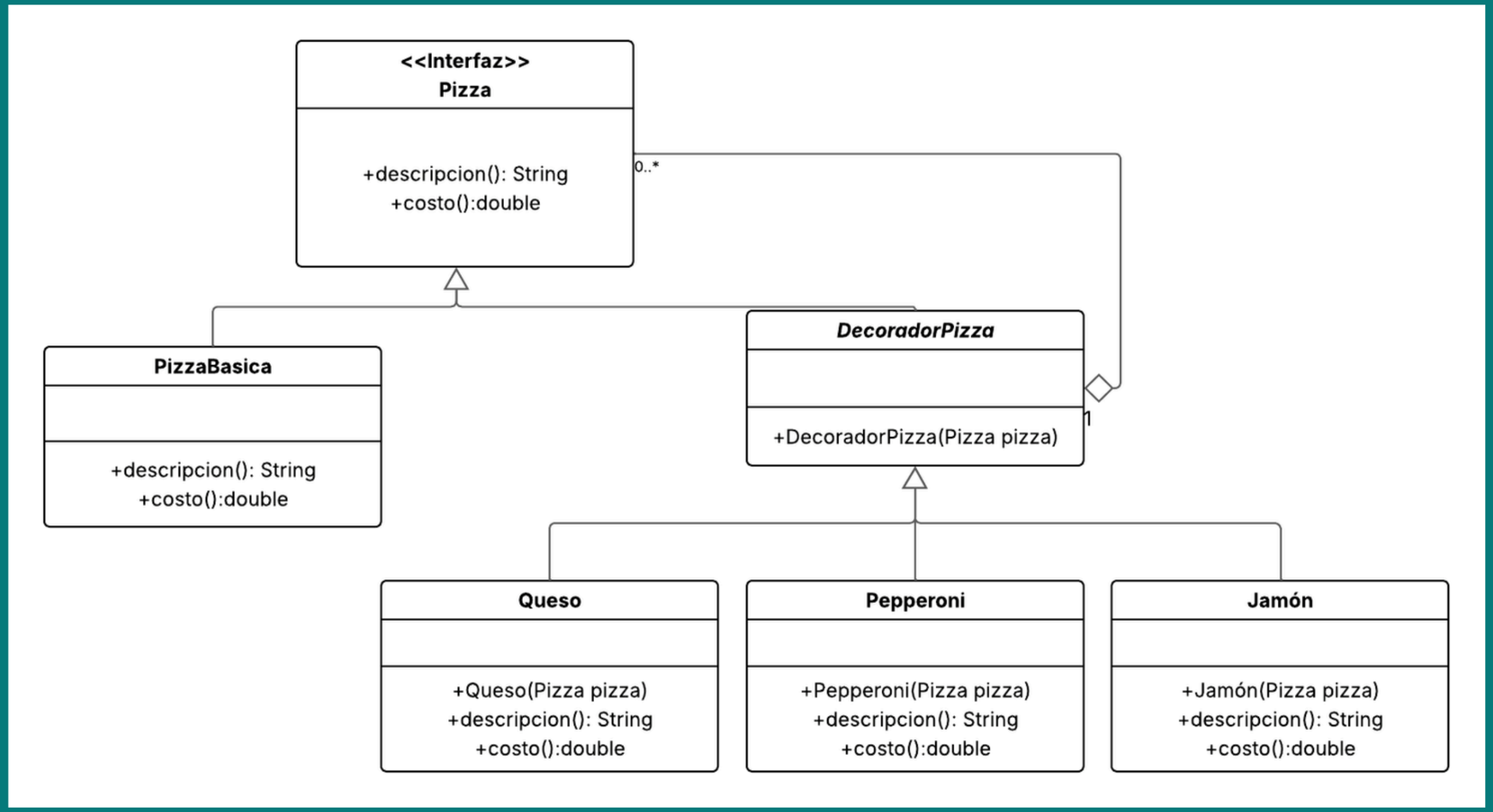
3:00 pm

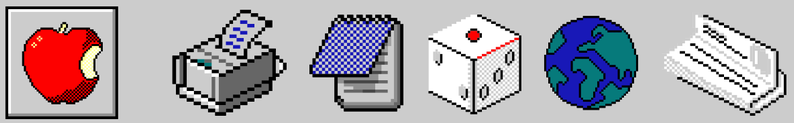
Sin Decorator



Con Decorator

Ver código ->



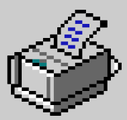


3:00 pm

UI

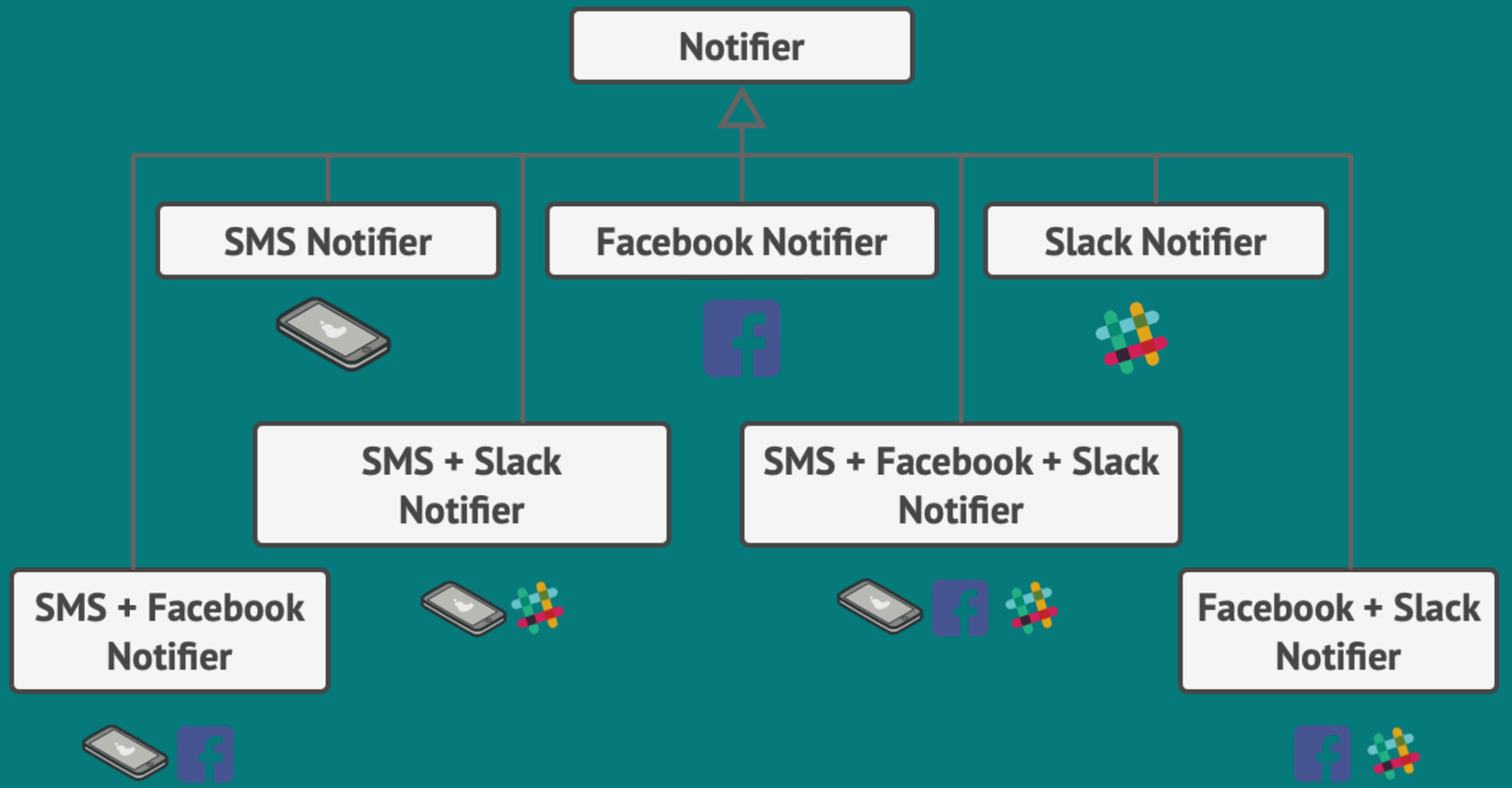
Ejemplos

Tenemos un sistema de notificaciones y queremos implementar varios tipos



3:00 pm

Sin Decorator

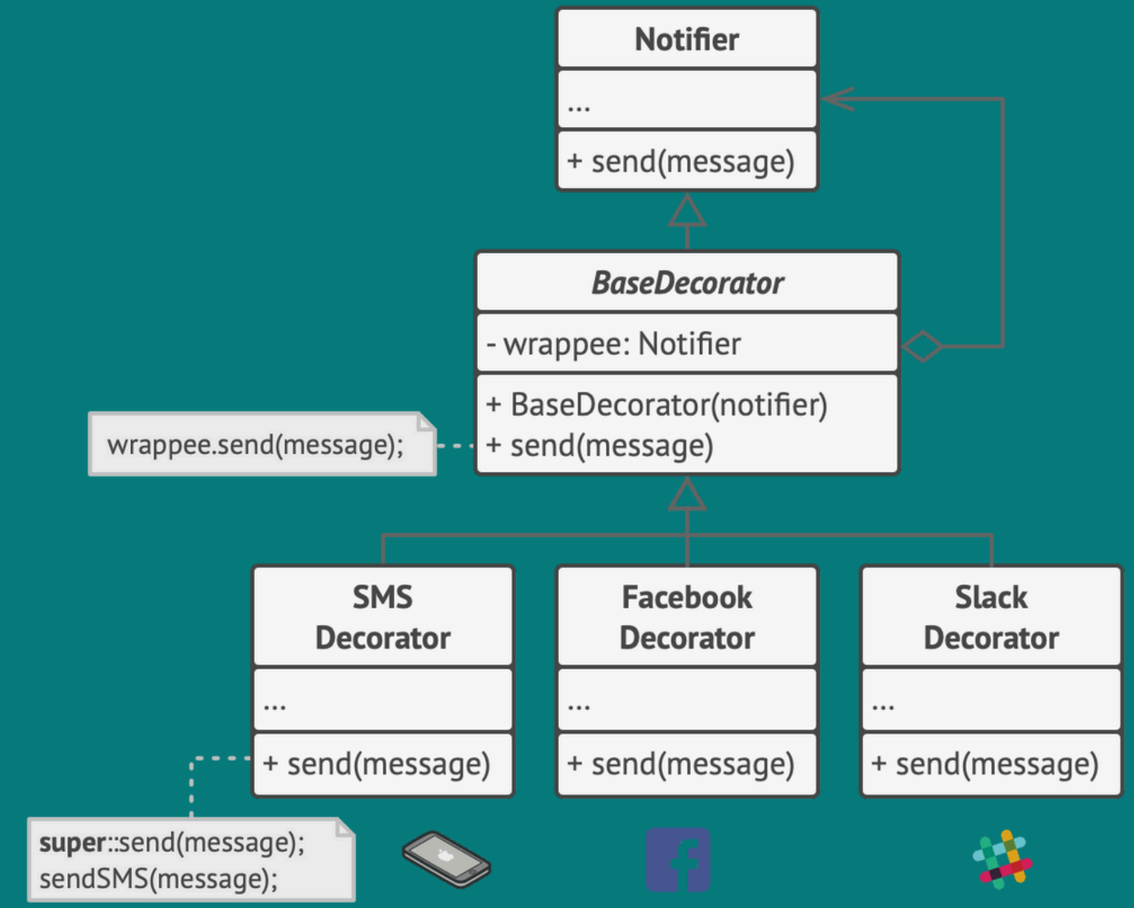


Tomado de: https://refactoring.guru/es/design-patterns/decorator

3:00 pm

Con Decorator

Ver código ->



Tomado de: https://refactoring.guru/es/design-patterns/decorator

3:00 pm

¡MUCHAS GRACIAS!

3:00 pm

Referencias

Román, B. (2023, May 25). Patrón decorator. Medium. https://medium.com/%40bromanv/patr%C3%B3n-decorator-f34a9a213053

Parra, D. (2024, December 28). Patrones de diseño - Decorator. The Power Ups. https://thepowerups-learning.com/patrones-de-diseno-decorator/?utm_source=chatgpt.com

Decorator. (2021). https://reactiveprogramming.io/blog/es/patrones-de-diseno/decorator

Decorator. (n.d.). Refactoring Guru. https://refactoring.guru/es/design-patterns/decorator

Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Pearson Education, 31 oct 1994 - 395. https://www.javier8a.com/itc/bd1/articulo.pdf

3:00 pm