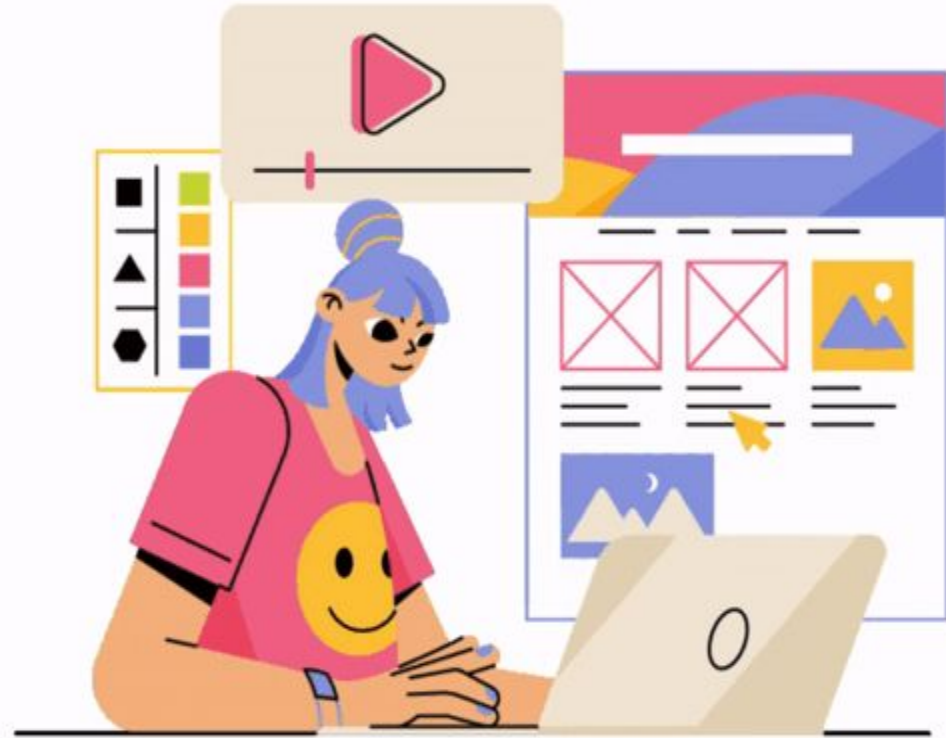




Patrones de diseño Factory

Jordano Escalante
S1 2025

Patrones de diseño creacionales





Las patrones creacionales abstraen el proceso de instanciación de clases, encapsulan el proceso de instanciación de los objetos.

- Ocultan detalles de creación.
- Permiten mayor desacoplamiento.
- Proveen un mayor control del proceso de instanciación de objetos en un programa.

Factory Method





Este patrón aprovecha el polimorfismo para definir interfaces y crear objetos de una superclase dejando a las subclases la responsabilidad de decidir el tipo de los objetos.

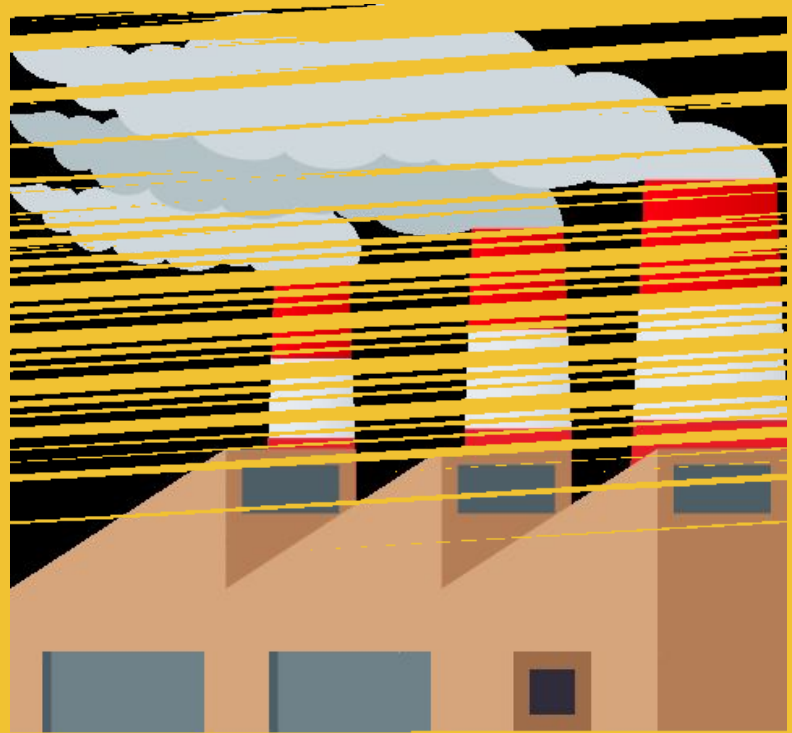
Problema

Cuando se tiene una clase que tiene métodos para definir varios objetos distintos, cada vez que se requiera implementar un nuevo tipo de objetos para su creación será necesario modificar la clase generadora, esto incumple el principio **open and close**.



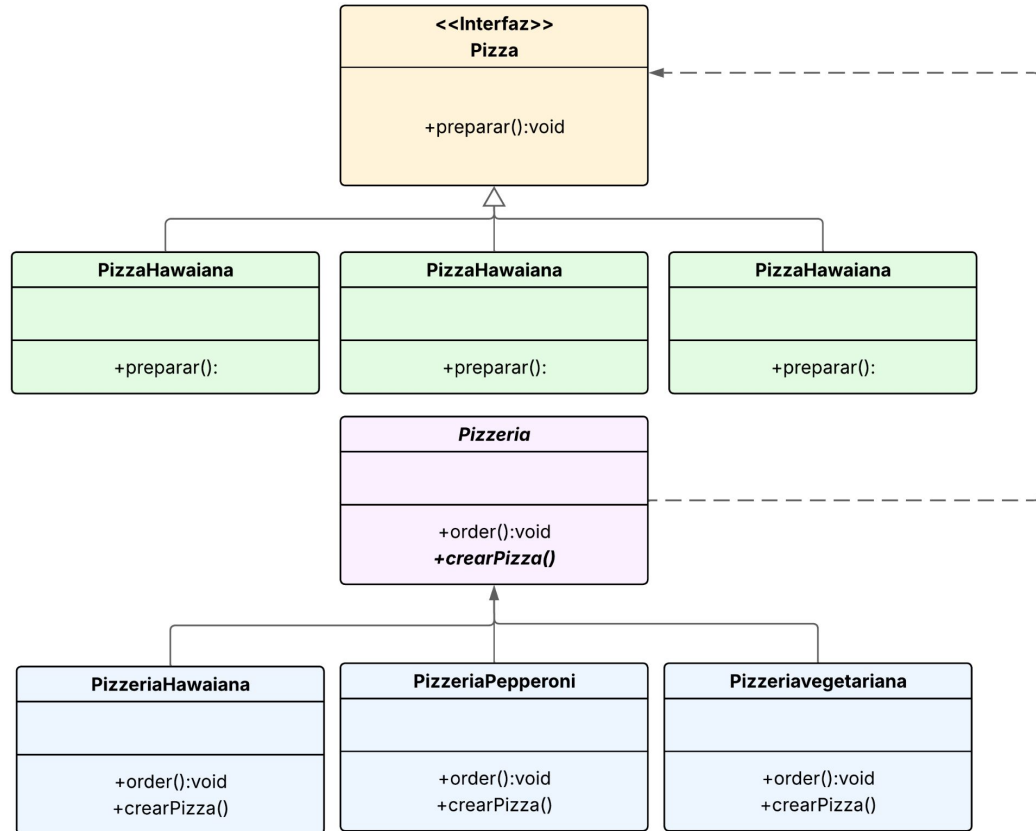
Factory Method

Al implementar Factory Method se define una interfaz o superclase fábrica abstracta en la que se define un método constructor especial a través del cuál se crearán los objetos que hereden de la clase que implementa la interfaz o clase abstracta y dejar a estos objetos la responsabilidad de definir el tipo de objeto que son de manera que el programa maneja todos los objetos como la superclase (polimorfismo) sin depender de clases concretas.



Diagrama

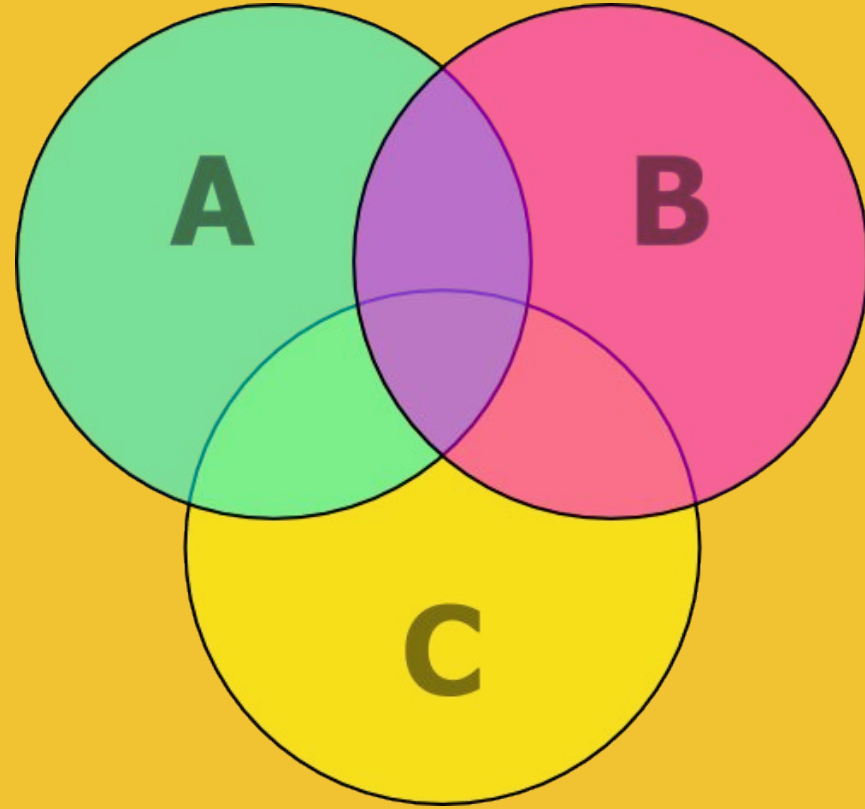
FactoryMethod



Abstract Factory



Patrón de diseño
creacional que proporciona
una interfaz para crear
familias de objetos
relacionados o
dependientes sin
especificar sus clases
concretas.



FRUIT



TOYS



Problema

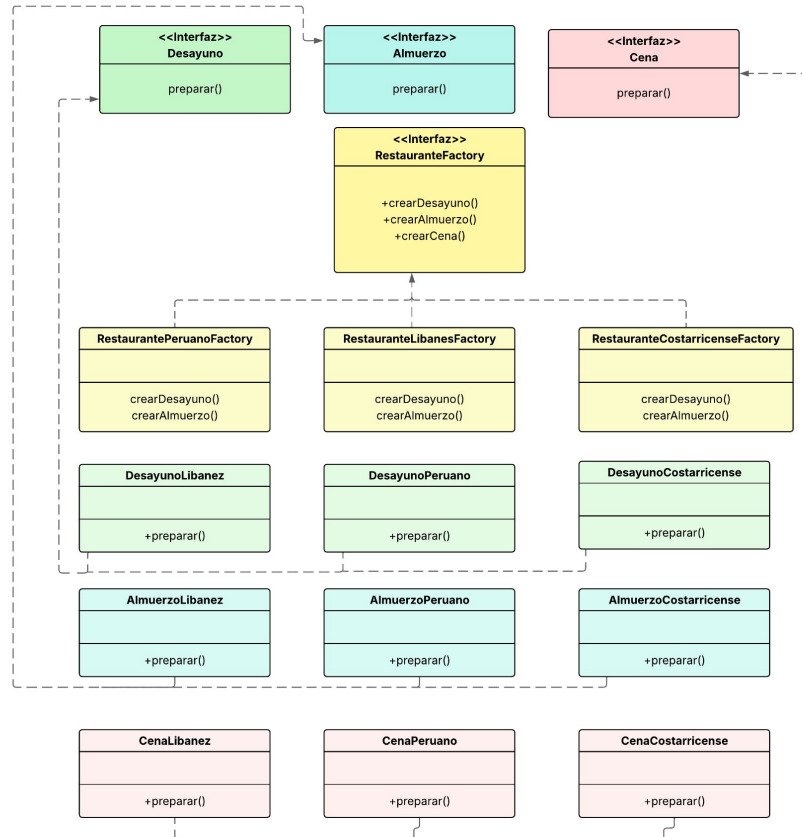
Este patrón se utiliza para casos en los que se deben crear familias o conjuntos de objetos con alguna similitud y contar con la capacidad de generar un conjunto o otro sin necesidad de realizar grandes cambios al código.

Abstract factory
soluciona este
problema utilizando una
interfaz general para
cada objeto del
conjunto que se desea
implementar y una
interfaz concreta para
cada tipo de los
objetos.



Diagrama

AbstractFactory



Bibliografía

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.

Refactoring.Guru. (n.d.). *Abstract Factory*. En *Design Patterns*. Recuperado [30 de mayo del 2025], de <https://refactoring.guru/design-patterns/abstract-factory>

Refactoring.Guru. (s. f.). *Factory Method (patrón de diseño)*. Refactoring.Guru. Recuperado el [30 de Mayo del 2025], de <https://refactoring.guru/es/design-patterns/factory-method>



That's all Folks!