**Padrão de Projeto: Template Method**

**O que é?**

O padrão de projeto Template Method é um padrão comportamental que define a estrutura de um algoritmo em uma classe base, permitindo que as subclasses alterem partes desse algoritmo sem modificar sua estrutura geral.

O objetivo principal desse padrão é permitir a reutilização do código ao definir a estrutura comum do algoritmo na classe base, enquanto as subclasses podem alterar ou estender partes específicas do algoritmo conforme necessário.

Em resumo, o padrão Template Method define um esqueleto de algoritmo em uma classe base, com partes flexíveis que podem ser personalizadas pelas subclasses, promovendo a reutilização e a padronização da estrutura do algoritmo.

**Como funciona?**

Ele funciona da seguinte forma:

**Classe Abstrata**: Define um método (o "método template") que declara a estrutura do algoritmo, dividido em etapas ou passos. Além disso, essa classe pode conter métodos concretos (com implementações padrão) e/ou métodos abstratos (que as subclasses devem implementar).

**Subclasses**: Estendem a classe abstrata e implementam os métodos abstratos, fornecendo suas próprias implementações para certas etapas do algoritmo.

**Método Template**: Este método na classe abstrata coordena a execução do algoritmo, chamando os métodos concretos ou abstratos de acordo com a estrutura predefinida.

**Customização**: As subclasses podem substituir ou estender partes específicas do algoritmo, mantendo a estrutura geral definida pelo método template.

**Qual problema resolve?**

O padrão Template Method resolve o problema da reutilização de código ao definir a estrutura comum de um algoritmo em uma classe base e permitir que subclasses personalizem partes específicas desse algoritmo sem alterar sua estrutura geral.

Isso é útil quando há um algoritmo comum que possui uma estrutura fixa, mas algumas etapas desse algoritmo podem variar entre diferentes implementações. Em vez de replicar o código comum em todas as subclasses, o Template Method permite que essa estrutura comum seja definida uma vez na classe base, enquanto as variações específicas são implementadas nas subclasses.

Assim, ele evita a duplicação de código e promove a consistência na estrutura do algoritmo, facilitando a manutenção, a extensão e a compreensão do código, ao mesmo tempo em que fornece flexibilidade para personalização em partes específicas do algoritmo.