

Template Method

Classificação: Comportamental

Contexto

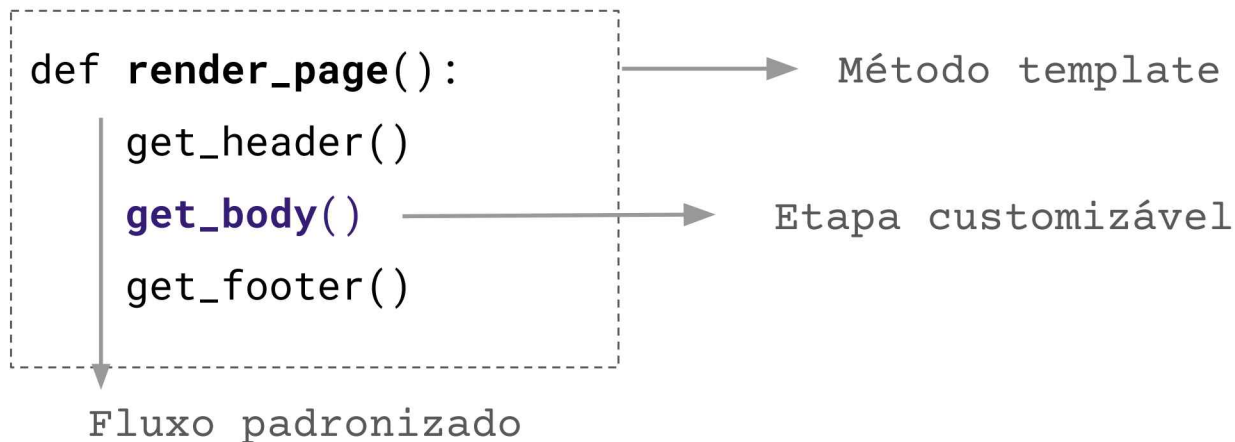
Como garantir a execução de uma sequência de passos e ao mesmo tempo permitir que algum desses passos sejam customizáveis?

Template Method

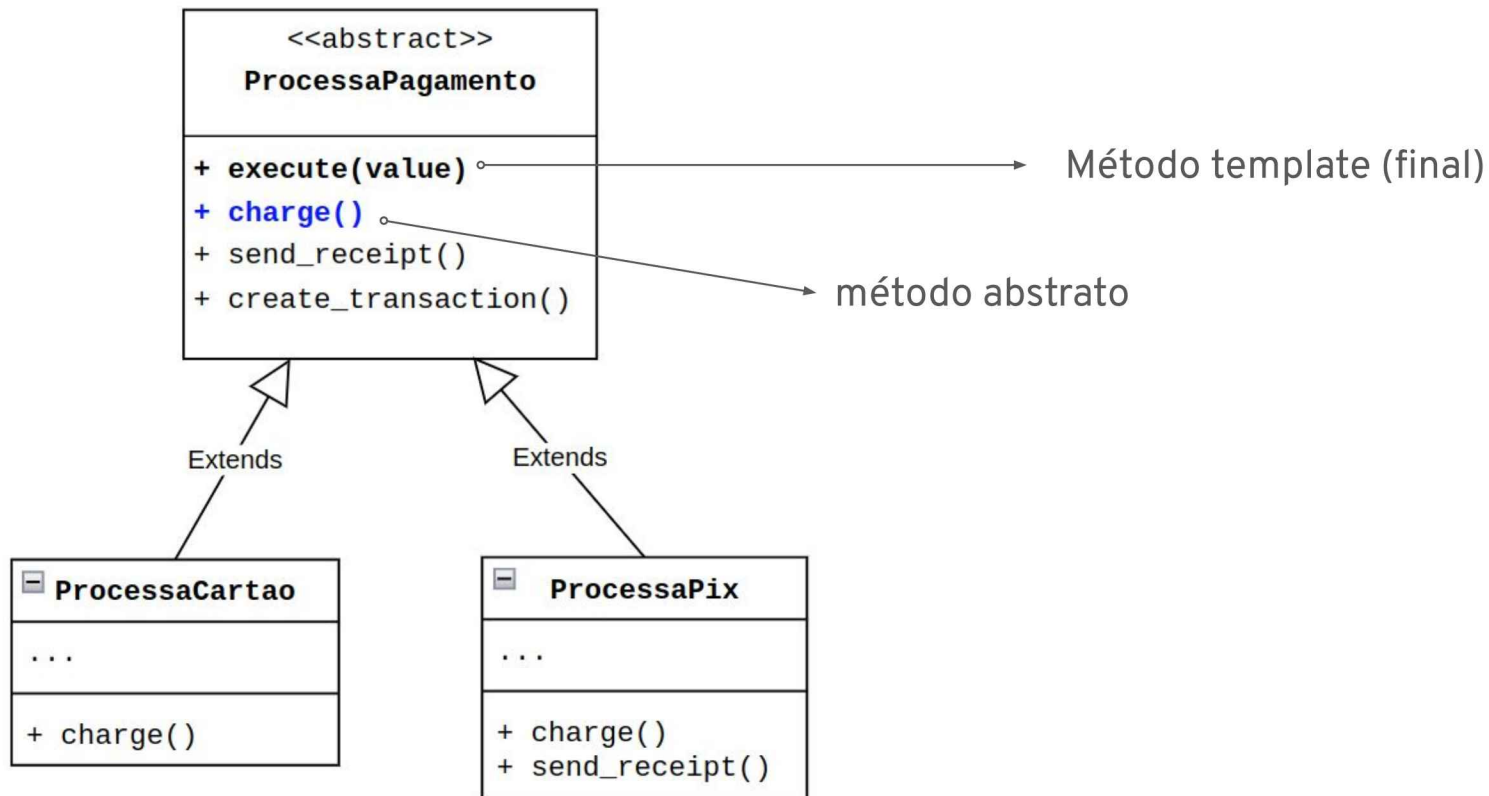
- Define os passos de um algoritmo, permitindo que a implementação desses seja fornecida por subclasses.
- Garantindo a estrutura (esqueleto) podemos manter a consistência de execução do algoritmo.
- Encapsula: a sequência de passos

Objetivos

- Obter um método modelo que **padroniza** os passos de execução do algoritmo, porém com flexibilidade para modificar a lógica interna da etapa.
- A **ordem** das etapas é **fixa**, mas a **implementação** desta pode **variar**.



Diagrama



Cenários

Exemplos de processos que requerem padronização da execução porém com pontos de customização:

Autenticação do usuário:

1. gerar log da ação;
2. validar credenciais;
3. autorização,
4. criar sessão do usuário,
5. redirecionar.

Processamento de pagamento:

1. criar transação;
2. gerar log;
3. gerar cobrança;
4. verificar a transação;
5. enviar recibo;
6. logar transação.

Atividade

Cenário: Um sistema deve permitir o login de dois tipos de usuários: (1) funcionários e (2) clientes. O processo de autenticação deve ser padronizado, isto é, deve ser garantido a ordem de execução das seguintes etapas:

1. **Validação** das credenciais (Google, Github, Email)
2. **Logar** resultado da validação (salva email do usuário);
3. Carrega **preferências** do usuário;
4. **Redirecionar**
 - a. página Home (usuário cliente).
 - b. página Dashboard (usuário funcionário).

Faça o código python usando o *Template Method* para definir esse processo.