

**Pontifícia Universidade Católica de Minas Gerais**

Anna Laura Reis, Vinicius Levi, João Pedro Mayrink

**Laboratório de Programação Modular: Trabalho Final**  
Padrões de projeto

Belo Horizonte  
12 de dezembro de 2021

## **1. Introdução**

Levando em consideração o projeto apresentado era de se esperar que algumas melhorias fossem feitas para que alguns problemas pontuais fossem consertados. Estes foram a criação de dois new Classe() ao utilizar o menu de um restaurante em que possui mais de um tipo de comida, a criação de uma operação em que se é possível criar um molho extra, também era preciso criar uma verificação para ver se a quantidade de comidas estava dentro do permitido..

## **2. Padrões de projeto implementados**

Os padrões de projetos escolhidos foram: Factory Method, Flyweight e Observer. A Factory Method foi escolhida pois mudaria a necessidade de se instanciar dentro do main qual seria a subclasse instanciada. O Observer foi utilizado pois resolveria de um modo mais efetivo e limpo a verificação de quantidade de itens do pedido. O FlyWeight foi usado nesses objetos que eram parecidos entre si, mas ao mesmo tempo tinham certas particularidades entre eles.

## **3. Factory Method**

Quando for escolhido o tipo de lanche da lanchonete no menu que temos no método main não seria feito desnecessariamente a criação de duas classes, com o new.

Se foi criada a classe ComidaSimpleFactory, que dependendo se for o caso 1, vai se adicionar ao pedido a Pizza(nova = new Pizza(aux)) e se farão algumas perguntas necessárias como se é necessário adicionar borda, já se o caso 2 for escolhido vai ser adicionado ao pedido o sanduíche(nova = new Sanduiche(aux)) e outras perguntas serão feitas, como por exemplo, se vai se utilizar o dobro de hambúrguer.

## **4. Flyweight**

Foi feito um objeto simples que referencia um objeto maior e todos os objetos que eles compartilham. Foi criada uma interface IFlyweight com o método de operação que vai ser reutilizado tanto na classe Sanduíche quanto na classe Pizza, que são duas classes que herdam de comida. Nesse operação será feito como os adicionais são inseridos e como devem ser mostrados, como preço e etc. Sendo que na interface se tem o método genérico.

## **5. Observer**

Quando um pedido é feito é necessário verificar a quantidade de comidas que se tem disponível. Para que essa funcionalidade fosse implementada foi feita a utilização do Observer.

Foi criada as classes PedidoObersavel e a VerificaComida. A PedidoObersavel é feito o conectar e o desconectar, o atualizar que nos avisa no atualizar caso o método tenha algum problema e que atualiza a lista de pedidos. A classe VerificaComida pega o pedido, a listagem de comidas do pedido e verifica se a quantidade está dentro do limitado, caso esteja o pedido é atualizado, caso não um erro aparece no terminal.

Algumas atualizações tiveram que ser feitas, como chamar a classe Comida como uma coleção e não como vetor, consequentemente outros métodos tiveram que ser atualizados para seguir o padrão da lista.

## **6. Link do github**

- <https://github.com/annii202/padroesProjeto>