

## **Análise Técnica e Propostas de Melhoria: Sistema de Lanchonete**

O sistema desenvolvido para a gestão de pedidos da lanchonete demonstra uma aplicação sólida dos pilares da Orientação a Objetos, especialmente no que tange à Herança e ao Polimorfismo. A utilização de uma classe base `Item` para derivar especializações como `Pizza`, `Lanche` e `Salgadinho` permite que o sistema trate diferentes produtos de forma genérica dentro de uma mesma lista de pedidos. Contudo, a lógica de cálculo de preços, que diferencia salgadinhos (por unidade) de outros itens, está atualmente dispersa entre a classe `Pedido` e a classe principal. Uma melhoria arquitetural importante seria a aplicação do método de Sobrescrita (`Override`): a classe `Item` poderia definir um método abstrato `getPrecoTotal()`, obrigando cada subclasse a implementar sua própria lógica de cálculo. Isso eliminaria a necessidade de verificações condicionais com `instanceof`, tornando o código mais limpo e aderente ao princípio do Aberto/Fechado (OCP).

No que diz respeito à interação com o usuário e persistência, o diagrama e a implementação atual dependem fortemente de entradas via console e armazenamento volátil em memória. Em um cenário acadêmico mais avançado ou profissional, recomenda-se o desacoplamento da camada de visão (`Input/Output`) da camada de modelo. Adicionalmente, é fundamental implementar um robusto tratamento de exceções para a entrada de dados. Por exemplo, no caso de atributos numéricos como `Item.peso` (que espera um `float`), o sistema deve ser capaz de capturar e tratar erros se o usuário digitar valores não numéricos, como "50g", evitando falhas no processamento e guiando o usuário a fornecer o formato correto. Por fim, a validação de dados, como a data de validade (atualmente tratada como `String`), deveria ser evoluída para tipos temporais nativos do Java (`LocalDate`), garantindo maior robustez no controle de qualidade dos itens perecíveis da lanchonete.