

Relatório Técnico-Científico

Análise de Sistema de Gerenciamento de Pedidos Utilizando Padrões de Projeto

Clara Gabryellen P. Aderno, Igor dos S. Vieira, Jhon Luiz S. Santos, Natan C. Da Silva, Rayan S. Silva

Sistemas de Informação – Faculdade de Excelência (UNEX)

`gabryellenpaixao@gmail.com, igorvieira1445@gmail.com,
devjhon01010@gmail.com, natancorreiatr@gmail.com,
rayansantos725@gmail.com`

Abstract. *This technical-scientific report presents a detailed analysis of an order management system developed in Java, focusing on the application of the State, Strategy and Singleton design patterns. The study demonstrates how the combination of these patterns provides flexibility, maintainability, and extensibility to the system, in addition to evaluating its compliance with the SOLID principles. The system was analyzed through code review, pattern identification and evaluation of criteria such as cohesion, coupling and extensibility.*

Resumo. *Este relatório apresenta uma análise detalhada de um sistema de gerenciamento de pedidos desenvolvido em Java, com foco em na aplicação dos padrões de projeto State, Strategy e Singleton. O estudo demonstra como a combinação desses padrões proporciona flexibilidade, manutenibilidade e extensibilidade ao sistema, além de avaliar a conformidade com os princípios SOLID. O sistema foi analisado por meio de revisão de código, identificação de padrões e avaliação de critérios como coesão, acoplamento e extensibilidade.*

1. Introdução

O desenvolvimento de sistemas complexos exige abordagens que promovam a organização, a manutenibilidade e a escalabilidade do código. Nesse contexto, os padrões de projeto surgem como soluções consolidadas para problemas recorrentes no desenvolvimento orientado a objetos. Este relatório analisa um sistema de gerenciamento de pedidos para estabelecimentos de *food delivery*, implementado utilizando a linguagem de programação Java, que utiliza três padrões de projeto fundamentais: State para controle do ciclo de vida dos pedidos, Strategy para geração de relatórios e Singleton para gerenciamento centralizado de dados. A relevância do estudo reside na demonstração prática de como a aplicação sistemática de padrões de projeto pode melhorar significativamente a qualidade arquitetural de sistemas corporativos.

2. Fundamentação Teórica

1. Padrão State: O padrão State permite que um objeto altere seu comportamento quando seu estado interno muda. No sistema analisado, esse padrão é utilizado para gerenciar o ciclo de vida dos pedidos, que envolve transições entre estados como **Aceito**, **Preparando**, **Em Entrega** e **Entregue**.
2. Padrão Strategy: O padrão Strategy define uma família de algoritmos, encapsula cada um e os torna intercambiáveis. No sistema, ele é aplicado na geração de relatórios, permitindo que estratégias detalhadas e simplificadas sejam selecionadas em tempo de execução.
3. Padrão Singleton: O padrão Singleton garante que uma classe tenha apenas uma instância e fornece um ponto de acesso global a ela. No sistema, a classe **CentralDeDados** utiliza esse padrão para armazenar todas as informações da aplicação.
4. Princípios SOLID: O sistema também foi avaliado com base nos princípios SOLID, que promovem a criação de um sistema robusto e de fácil manutenção.

3. Metodologia

A análise foi conduzida por meio de revisão sistemática do código-fonte, identificação dos padrões de projeto aplicados e a avaliação da conformidade com os princípios SOLID. Foram examinadas 15 classes Java organizadas em três pacotes principais. Os critérios de avaliação incluíram:

1. Coesão das classes e métodos
2. Acoplamento entre componentes
3. Extensibilidade do sistema
4. Conformidade com princípios de orientação a objetos
5. Tratamento de exceções

4. Resultados e Análise

1. Implementação do Padrão State: O sistema implementa uma máquina de estados finita para gerenciar o ciclo de vida dos pedidos. A interface **IPedidoEstado** define o contrato para todos os estados, e cada estado (como **AceitoPedido**, **PreparandoEstado**, etc) implementa suas próprias transições. Essa abordagem reduziu significativamente a complexidade ciclomática, eliminando estruturas condicionais complexas.
2. Implementação do Padrão Strategy: A geração de relatórios foi implementada usando o padrão Strategy, com interfaces como **RelatorioStrategy** e implementações concretas para relatórios detalhados e simplificados. Isso permitiu alta flexibilidade e separação de responsabilidades.
3. Implementação do Padrão Singleton: A classe **CentralDeDados** implementa o Singleton para garantir um único ponto de acesso aos dados. Embora isso

simplifique, o acesso global, identificou-se que a abordagem pode dificultar testes unitários e introduzir acoplamento.

4. Conformidade com os Princípios SOLID: O sistema demonstrou boa aderência aos princípios SOLID, especialmente no que diz respeito ao Princípio de Responsabilidade Única (SRP) e ao Princípio Aberto/Fechado (OCP).

5. Considerações finais

O desenvolvimento do programa com a aplicação dos padrões State, Strategy e Singleton resultou em uma arquitetura flexível, de fácil manutenção e extensível. A análise mostrou que os padrões contribuíram para a redução da complexidade e para a melhor organização do código. Como trabalhos futuros, sugere-se a implementação de validações mais robustas, persistência em banco de dados e a adoção de injeção de dependência para reduzir o acoplamento introduzido pelo Singleton.