Análise de Sistema de Gerenciamento de Pedidos

Um estudo técnico-científico sobre a aplicação de padrões de projeto em um sistema de gerenciamento de pedidos Java.

Contexto e Autores

Relatório Técnico-Científico

Análise detalhada de um sistema de gerenciamento de pedidos.

Foco na aplicação dos padrões de projeto **State**, **Strategy** e **Singleton**.

Avaliação da conformidade com os princípios **SOLID**.

Equipe de Autores:

- 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)

Introdução: A Importância dos Padrões

Desenvolvimento de Sistemas Complexos

Necessidade de organização, manutenibilidade e escalabilidade do código.

Padrões de Projeto: Soluções Reutilizáveis

Respostas consolidadas para problemas recorrentes em OO.

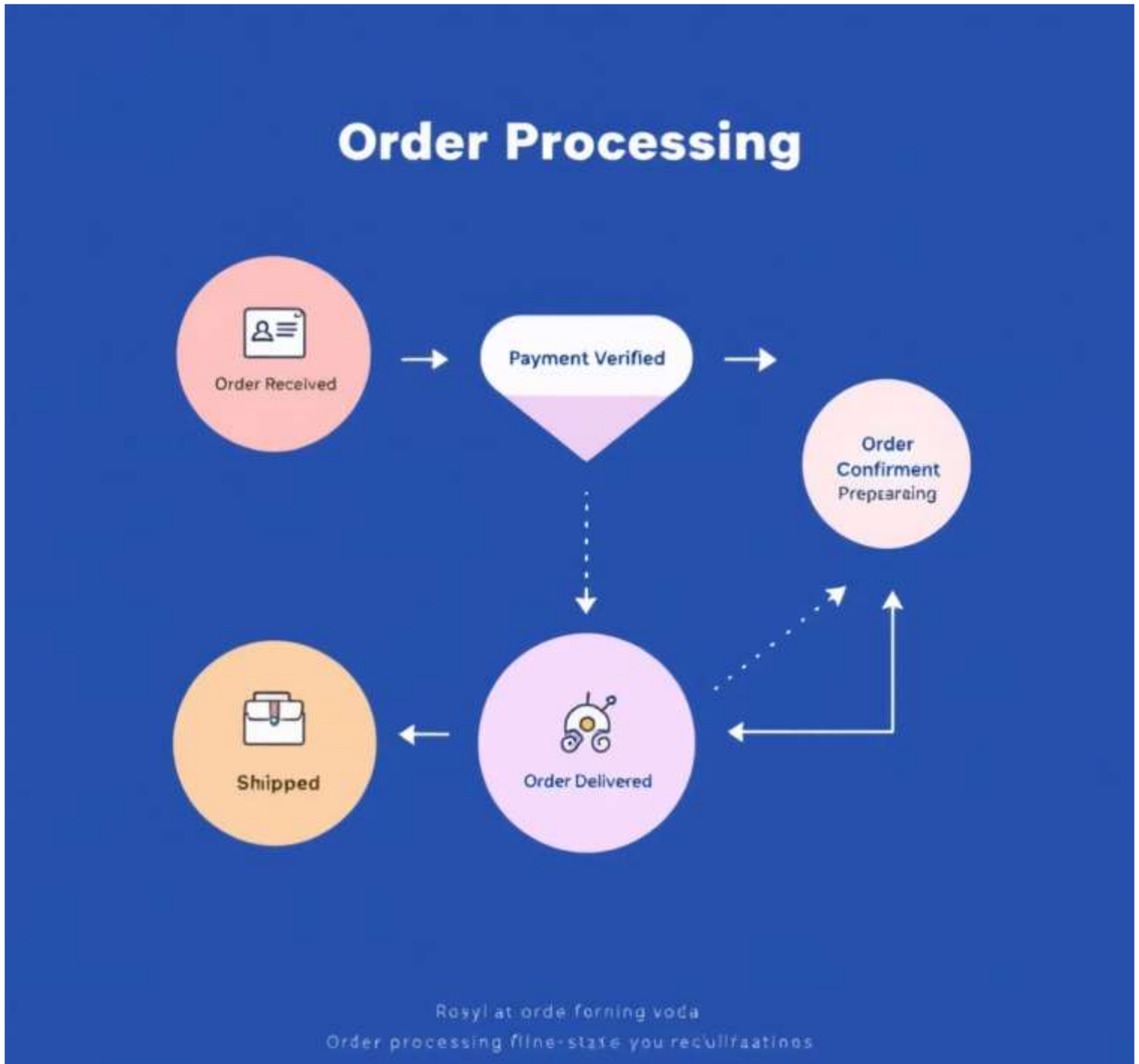
Foco: Sistema de Food Delivery

Implementação em Java com State, Strategy e Singleton.

Relevância do Estudo

Demonstrar a melhoria da qualidade arquitetural em sistemas corporativos.

Padrão State: Gerenciamento do Ciclo de Vida



Comportamento Variável

- Objeto altera comportamento com mudança de estado interno.
- Transições suaves e controladas.

Aplicação no Sistema

- Gerencia estados do pedido: Aceito, Preparando, Em Entrega, Entregue.

- Reduz complexidade ciclomática.
- Elimina estruturas condicionais complexas (if/else aninhados).

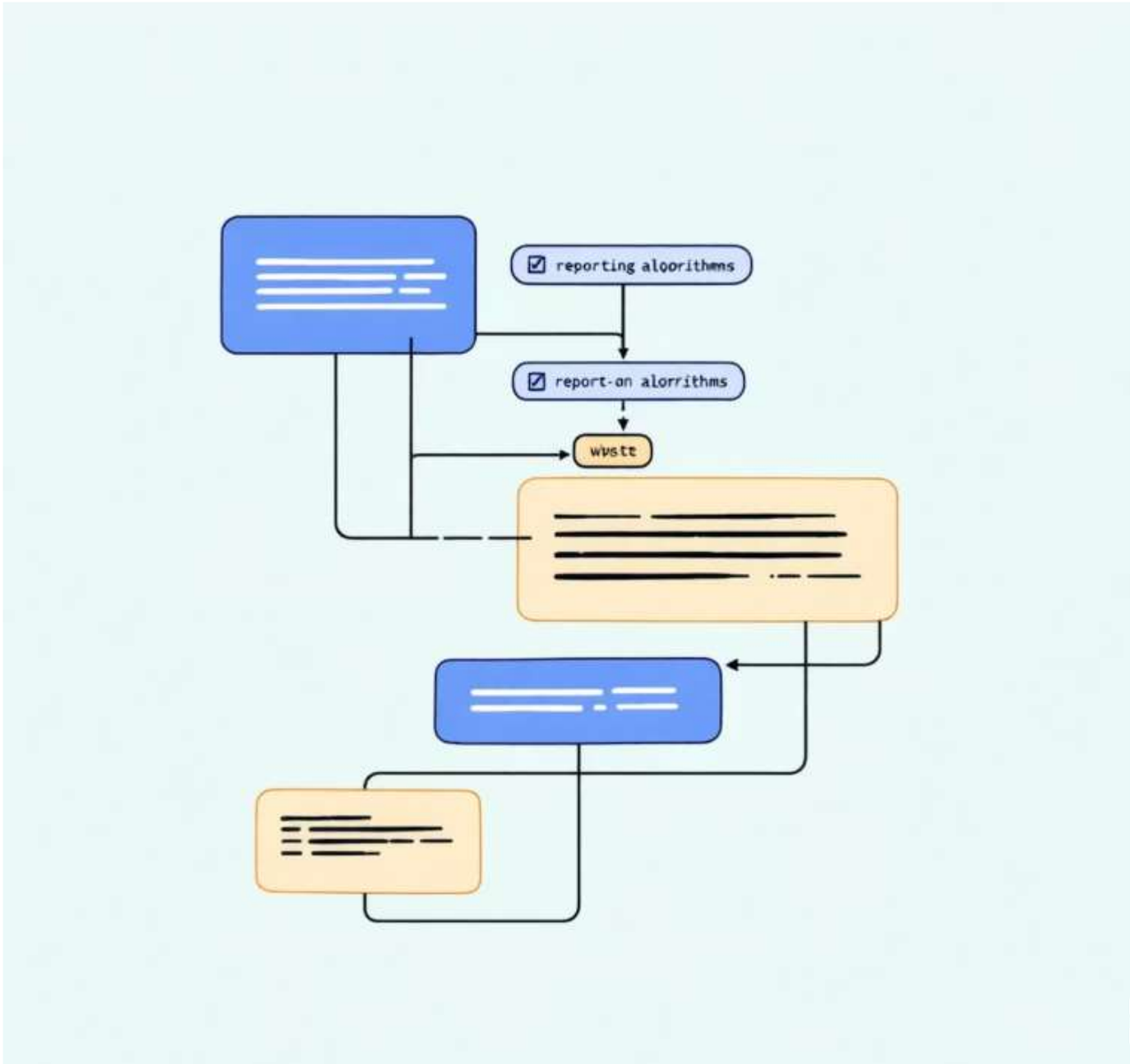
Padrão Strategy: Flexibilidade na Geração de Relatórios

Algoritmos Intercambiáveis

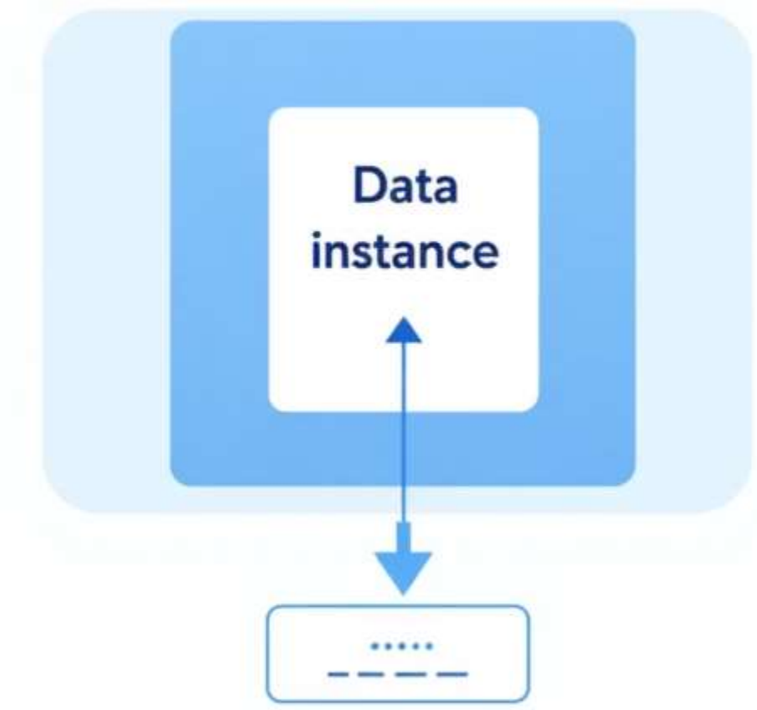
- Define uma família de algoritmos.
- Encapsula cada um, tornando-os intercambiáveis.

Aplicação no Sistema

- Geração de relatórios detalhados ou simplificados.
- Permite seleção em tempo de execução.
- Alta flexibilidade e separação de responsabilidades.



Padrão Singleton: Gerenciamento Centralizado de Dados



Instância Única Global

- Garante apenas uma instância da classe.
- Fornece um ponto de acesso global a ela.

Aplicação no Sistema

- Classe **CentralDeDados** armazena informações da aplicação.
- Simplifica o acesso global aos dados.

Desafios Potenciais

- Pode dificultar testes unitários.
- Introduz acoplamento global.

Princípios SOLID: A Base de um Código Robusto

Avaliação do sistema com base nos princípios SOLID para promover um código de fácil manutenção.

SRP - Single Responsibility Principle

Uma classe deve ter apenas uma razão para mudar.

OCP - Open/Closed Principle

Entidades de software abertas para extensão, fechadas para modificação.

LSP - Liskov Substitution Principle

Subtipos devem ser substituíveis por seus tipos base.

ISP - Interface Segregation Principle

Clientes não devem ser forçados a depender de interfaces que não utilizam.

DIP - Dependency Inversion Principle

Depender de abstrações, não de implementações concretas.

Metodologia de Análise

01

Revisão Sistemática do Código-Fonte

Exame de 15 classes Java, organizadas em 3 pacotes principais.

02

Identificação dos Padrões de Projeto

Mapeamento da aplicação dos padrões State, Strategy e Singleton.

03

Avaliação da Conformidade SOLID

Verificação da aderência aos princípios de design.

04

Critérios de Avaliação

- Coesão das classes e métodos
- Acoplamento entre componentes
- Extensibilidade do sistema
- Conformidade com POO
- Tratamento de exceções

Resultados e Análise

Padrão State

Máquina de estados finita.

Interface **IpedidoEstado** e implementações concretas.

Redução de complexidade ciclomática.

Padrão Strategy

Geração de relatórios flexível.

Interfaces **RelatorioStrategy**.

Alta flexibilidade e separação de responsabilidades.

Padrão Singleton

Classe **CentralDeDados**.

Ponto único de acesso aos dados.

Pode dificultar testes e aumentar acoplamento.

Princípios SOLID

Boa aderência, especialmente SRP e OCP.

Promove manutenção e robustez.

Considerações Finais e Próximos Passos

Arquitetura Resultante:

- Flexível
- De fácil manutenção
- Extensível

Contribuição dos Padrões:

- Redução da complexidade
- Melhor organização do código

Trabalhos Futuros:

- Validações mais robustas
- Persistência em banco de dados
- Adoção de injeção de dependência para reduzir acoplamento do Singleton