

# **Implementação do Padrão de Projeto Facade no Sistema TiaLuDelivery**

Maciel Lopes Francisco, Allan Brito Baretto, Aeverton Santos de Oliveira, Carlos Henrique Santos de Carvalho, Daniel Barbosa dos Reis

Sistema de informação – Centro Universitário de Excelência (UNEX)  
Praça José Bastos, Itabuna, Bahia – Brazil

[maciel.francisco@ftc.edu.br](mailto:maciel.francisco@ftc.edu.br), [allan.brito013@gmail.com](mailto:allan.brito013@gmail.com),  
[danielreisemail@gmail.com](mailto:danielreisemail@gmail.com), [bulhoesaevertonbulhoes@gmail.com](mailto:bulhoesaevertonbulhoes@gmail.com),  
[henriquecarvalhosw@gmail.com](mailto:henriquecarvalhosw@gmail.com)

**Equipe Assunção – Unidade Itabuna**

## **Resumo**

Este relatório descreve a implementação do padrão de projeto Facade no sistema TiaLuDelivery, desenvolvido como parte da OAT 1.2 da disciplina de Programação Orientada a Objetos. O objetivo foi criar uma fachada que centraliza o acesso às operações principais do sistema (clientes, pedidos e cardápio), sem expor os detalhes internos de implementação. O código desenvolvido define a estrutura da classe SistemaPedidosFacade, que fornece métodos simplificados para interação com o sistema e serve de base para integração com outros padrões (Builder, Singleton e State).

## **Abstract**

This report describes the implementation of the Facade design pattern in the TiaLuDelivery system, developed as part of the OAT 1.2 assignment for the Object-Oriented Programming course. The main goal was to create a facade that centralizes access to the system's core operations (customers, orders, and menu) while hiding implementation details. The developed code defines the SistemaPedidosFacade class as a simplified interface and serves as a foundation for integrating other design patterns (Builder, Singleton, and State).

## **1. Introdução**

O projeto TiaLuDelivery tem como objetivo refatorar um sistema de delivery utilizando padrões de projeto orientados a objetos. A equipe Assunção, da unidade Itabuna, ficou responsável pela implementação do padrão Facade, cuja função é simplificar o acesso às funcionalidades principais do sistema. O padrão Facade permite que o sistema seja utilizado através de uma interface única, ocultando a complexidade interna e promovendo baixo acoplamento entre as classes.

## **2. Fundamentação Teórica**

O padrão Facade (do catálogo GoF) é um padrão estrutural que fornece uma interface unificada para um conjunto de interfaces em um subsistema. Ele define uma 'fachada' que simplifica o uso do sistema, facilitando a manutenção e o reuso de código. Em sistemas complexos, o Facade atua como intermediário entre camadas, reduzindo dependências e

simplificando interações — ideal para o contexto do TiaLuDelivery, que envolve múltiplos componentes (clientes, pedidos, cardápio).

### 3. Metodologia

A implementação foi realizada a partir do código-base fornecido no repositório GitHub da disciplina. Os passos seguidos foram:

1. Criação da branch itabuna/implementacao-facade.
2. Criação da classe SistemaPedidosFacade no pacote br.edu.unex.tiaLuDelivery.model.
3. Definição dos métodos públicos:
  - registrarNovoCliente(String nome, String telefone)
  - criarPedido(Customer cliente, List<OrderItem> itens)
  - listarCardapio()
  - exibirPedidosCliente(Customer cliente)
4. Cada método lança UnsupportedOperationException, pois depende das implementações de outras equipes.
5. Foi criado um teste simples na classe TiaLuCLI para verificar a integração e a instância da fachada.

### 4. Resultados

A classe SistemaPedidosFacade foi criada com sucesso e integrada ao projeto, permitindo que a camada de interface (CLI) utilize seus métodos sem acoplar-se diretamente às classes do modelo. A execução do teste resultou na mensagem:

[OK] Facade instanciado e acessível (método ainda não implementado).

Com isso, foi possível comprovar que o Facade está corretamente integrado e pronto para receber as implementações futuras de outras equipes.

### 5. Conclusão

A implementação do padrão Facade pela equipe Assunção atendeu aos requisitos propostos. A fachada criada fornece um ponto de entrada único para o sistema e estabelece a base para futuras expansões. Com o avanço das outras equipes (Singleton, Builder e State), o SistemaPedidosFacade servirá como camada intermediária de comunicação entre os subsistemas.

### Referências

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.  
Documentação da OAT 1.2 – Refatorando Código com Padrões de Projeto.  
Código-fonte do repositório TiaLuDelivery (GitHub da disciplina).