

**Desenvolvimento de Sistemas de Software**

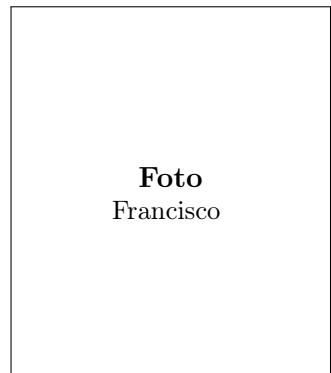
*Sistema de Gestão de Restaurantes Fast-Food*

**Projeto Prático - Grupo TP-28**

**URL do Repositório:**

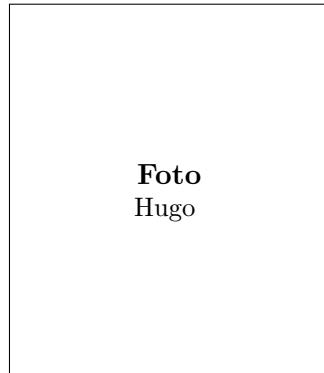
<https://github.com/marcosevegrand/DSS2526>

## **Identificação dos Elementos do Grupo**



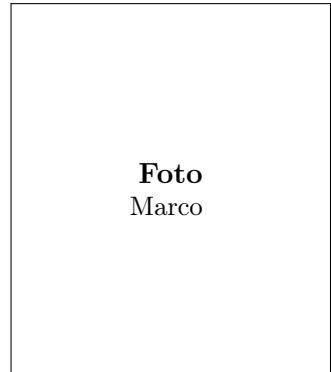
**Foto**  
Francisco

**Francisco Martins**  
A106902



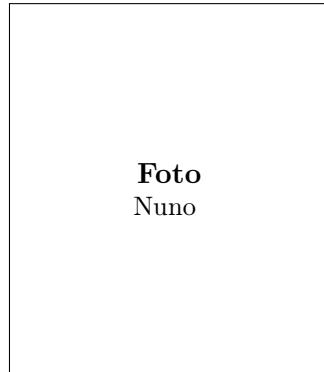
**Foto**  
Hugo

**Hugo Soares**  
A107293



**Foto**  
Marco

**Marco Sevegrand**  
A106807



**Foto**  
Nuno

**Nuno Rebelo**  
A107372

# Conteúdo

<b>1 Descrição dos Resultados Obtidos</b>	<b>2</b>
<b>2 Análise de Requisitos</b>	<b>2</b>
2.1 Modelação de Domínio . . . . .	2
2.2 Diagramas de Casos de Uso . . . . .	3
<b>3 Modelação Conceptual</b>	<b>3</b>
3.1 Diagramas de Classe e de Sequência . . . . .	3
<b>4 Solução Implementada</b>	<b>4</b>
4.1 Arquitetura, Componentes e Pacotes . . . . .	4
<b>5 Manual de Utilização</b>	<b>5</b>
5.1 Subsistema de Venda (Terminal do Cliente) . . . . .	5
5.2 Subsistema de Produção (Cozinha) . . . . .	5
5.3 Subsistema de Gestão (Administração) . . . . .	6
<b>6 Conclusão</b>	<b>6</b>

# 1 Descrição dos Resultados Obtidos

O sistema desenvolvido para a gestão de restaurantes permite a coordenação centralizada de uma cadeia de fast-food, ligando o momento da escolha do cliente ao processo de confecção na cozinha. Através de uma análise cuidada das necessidades operacionais, estruturámos a aplicação para garantir que o fluxo de informação entre o balcão e a produção seja contínuo e organizado.

Um dos resultados mais significativos foi a separação do sistema em três subsistemas distintos, mas complementares, cada um com a sua própria fachada especializada. O subsistema de venda (*VendaFacade*) gera a interação direta com o cliente, permitindo a visualização do catálogo e a personalização de produtos. O subsistema de produção (*ProducaoFacade*) organiza as tarefas nas estações de trabalho, calculando tempos de espera reais. Por fim, o subsistema de gestão (*GestaoFacade*) oferece as ferramentas administrativas para o controlo de pessoal, stocks e configuração do sistema.

Do ponto de vista arquitetural, o projeto destaca-se pelo isolamento das responsabilidades. A lógica de persistência, baseada em SQL, está completamente separada da lógica de negócio através de objetos de acesso a dados (DAOs). Cada fachada é independente e utiliza apenas os DAOs necessários para o seu subsistema, eliminando dependências desnecessárias. Esta decisão permite que o sistema lide com volumes maiores de informação sem perder desempenho e facilita futuras atualizações na estrutura de dados ou na interface de utilizador.

## 2 Análise de Requisitos

A análise focou-se na identificação clara dos processos de negócio e na forma como as entidades se relacionam no contexto de vários restaurantes.

### 2.1 Modelação de Domínio

O modelo de domínio identifica conceitos centrais como *Pedido*, *Estacao*, *Funcionario* e *Ingrediente*. As relações estabelecidas permitem que o sistema saiba, por exemplo, quais os ingredientes necessários para cada *Produto* e em que *Estacao* devem ser processados.

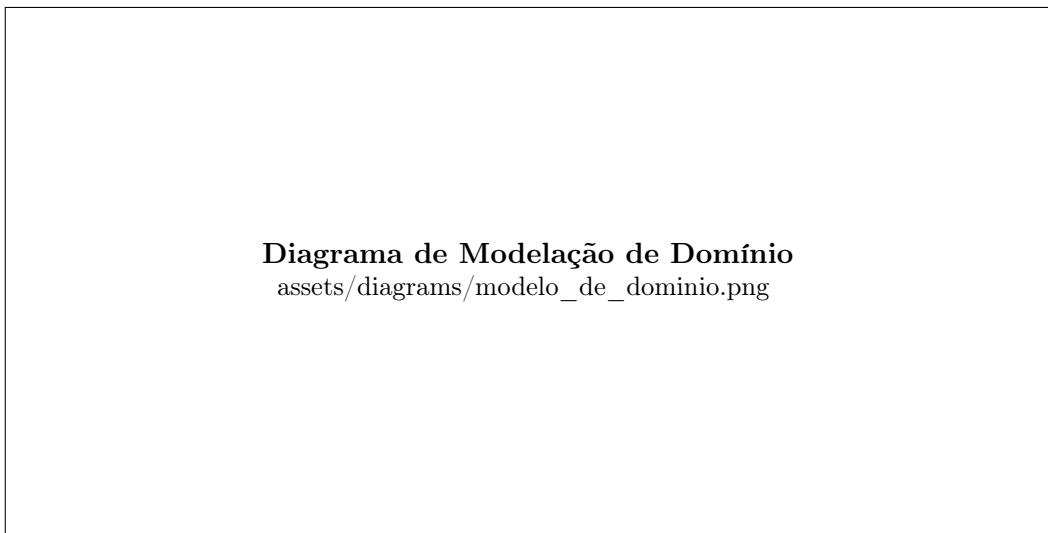


Figura 1: Representação das entidades e relações do domínio.

## 2.2 Diagramas de Casos de Uso

Estes diagramas ilustram as interações entre os atores (Cliente, Cozinheiro, Gestor) e o sistema.

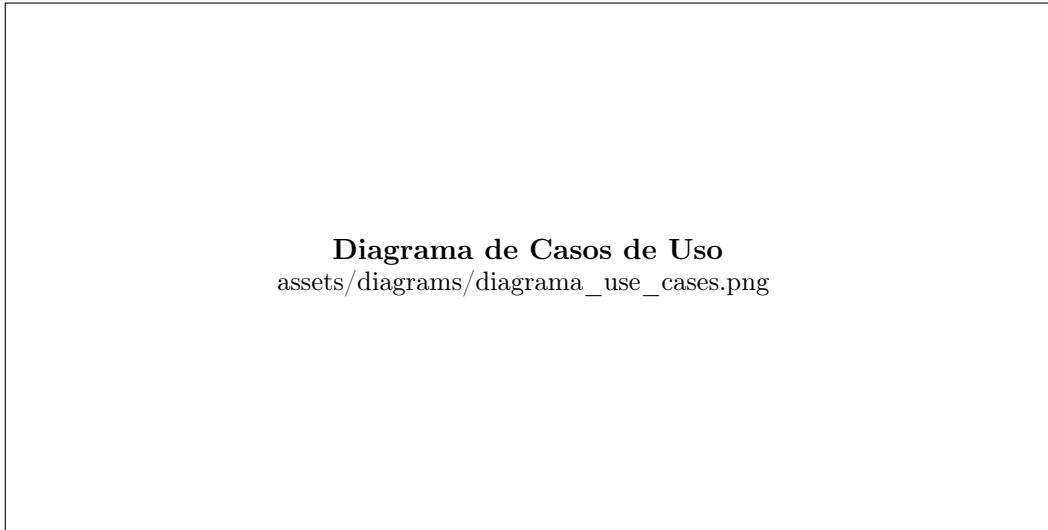


Figura 2: Casos de uso principais do sistema.

## 3 Modelação Conceptual

A transição da análise para a solução técnica baseou-se na definição de como as classes colaboram entre si para cumprir as regras de negócio.

### 3.1 Diagramas de Classe e de Sequência

A modelação conceptual define a estrutura das fachadas especializadas (*VendaFacade*, *ProducaoFacade*, *GestaoFacade*) e como estas utilizam as entidades de domínio e os DAOs para processar pedidos. O foco recai sobre a coordenação entre os controladores de interface e a lógica centralizada em cada serviço.

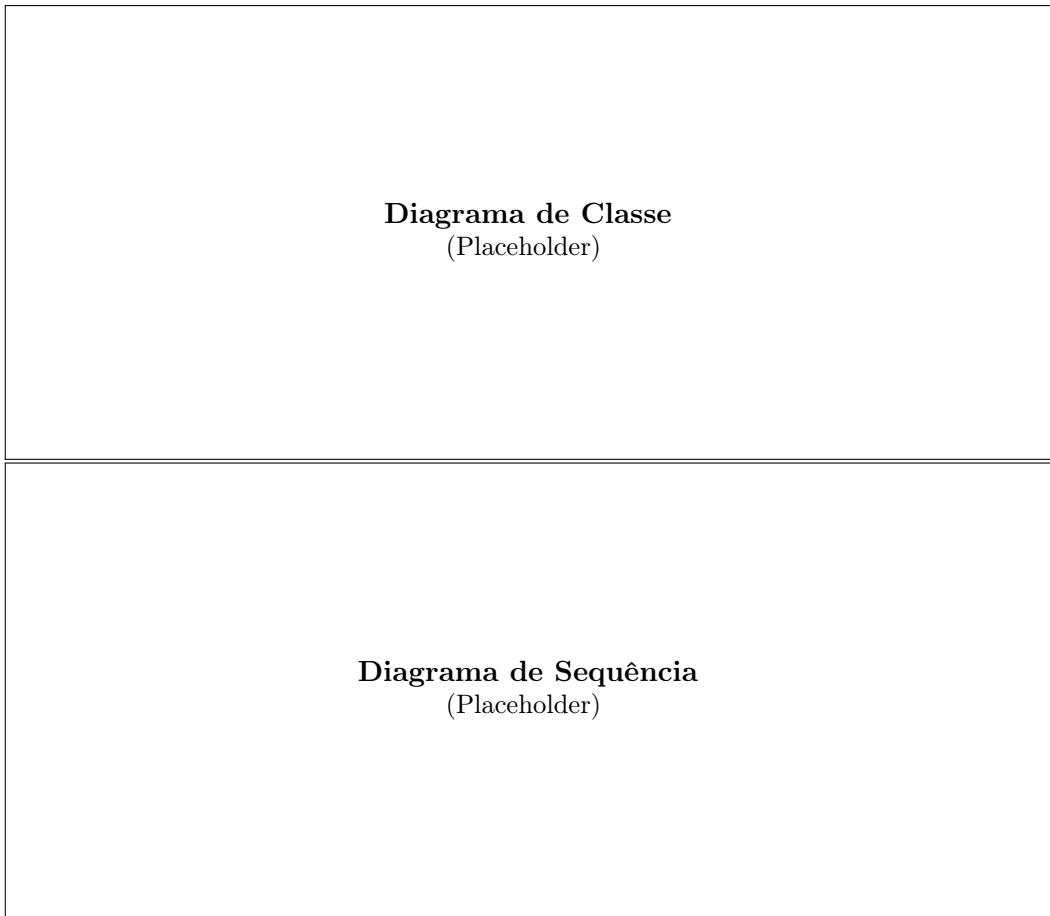


Figura 3: Estrutura da solução implementada.

## 4 Solução Implementada

A implementação reflete as decisões tomadas durante o planeamento, utilizando uma arquitetura em camadas bem definida.

### 4.1 Arquitetura, Componentes e Pacotes

O código está organizado em pacotes que refletem a sua função:

- *dss2526.ui* — Apresentação (interfaces de utilizador para cada subsistema)
- *dss2526.service* — Lógica de negócio, dividida em três sub-pacotes:
  - *dss2526.service.venda* — VendaFacade e operações de venda
  - *dss2526.service.producao* — ProducaoFacade e operações de produção
  - *dss2526.service.gestao* — GestaoFacade e operações de gestão
- *dss2526.data* — Persistência de dados através de DAOs e implementações (SQL)
- *dss2526.domain* — Entidades de domínio e abstrações
- *dss2526.app* — Configuração e inicialização da aplicação

Cada fachada especializada implementa a sua interface correspondente (*IVendaFacade*, *IProducaoFacade*, *IGestaoFacade*), oferecendo um contrato claro das operações disponíveis. As fachadas utilizam o padrão Singleton para garantir uma única instância durante a execução da aplicação.

A persistência de dados é gerida por DAOs especializados (IngredienteDAO, RestauranteDAO, ProdutoDAO, PedidoDAO, etc.), que abstraem os detalhes de acesso à base de dados SQL. Cada fachada utiliza apenas os DAOs necessários para o seu subsistema, promovendo baixo acoplamento.

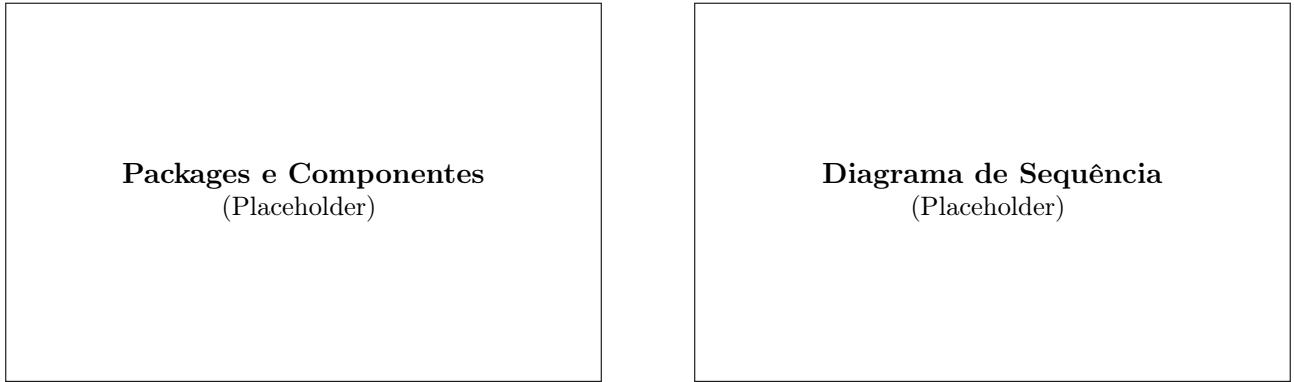


Figura 4: Detalhamento da solução implementada.

## 5 Manual de Utilização

O manual descreve o percurso típico de cada utilizador dentro do sistema.

### 5.1 Subsistema de Venda (Terminal do Cliente)

O cliente interage com o terminal de venda para gerir os seus pedidos. As principais funcionalidades são:

- Visualizar os pedidos ativos do restaurante
- Iniciar um novo pedido
- Consultar os itens disponíveis no catálogo
- Adicionar itens ao pedido
- Remover itens do pedido
- Finalizar o pedido e escolher o local de pagamento
- Receber um recibo com o número do pedido e o tempo estimado de entrega
- Consultar as estatísticas do restaurante

A *VendaFacade* coordena todas estas operações, gerenciando o fluxo de criação e manipulação de pedidos até à finalização.

## 5.2 Subsistema de Produção (Cozinha)

O funcionário na cozinha utiliza este subsistema para gerir a produção dos pedidos. O funcionário escolhe o restaurante e a sua estação específica. O sistema apresenta as tarefas a realizar. Ao concluir um item, o estado é atualizado em tempo real, permitindo que o cliente seja informado quando o pedido está pronto para entrega.

A *ProducaoFacade* gerencia:

- Distribuição de tarefas por estação
- Atualização de estados de produção
- Cálculo de tempos reais de processamento
- Sincronização com o subsistema de venda

## 5.3 Subsistema de Gestão (Administração)

O gestor utiliza este subsistema para gerenciar os aspectos operacionais e administrativos do restaurante. Pode monitorizar o inventário de ingredientes, gerir os funcionários e suas permissões, configurar catálogos e menus, e acompanhar o desempenho do restaurante.

A *GestaoFacade* oferece:

- Controlo de stocks e ingredientes
- Gestão de funcionários e definição de permissões
- Configuração de catálogos e menus
- Monitorização de pedidos e relatórios operacionais

## 6 Conclusão

O projeto desenvolvido demonstra como uma arquitetura bem planeada facilita a gestão de fluxos operacionais complexos. Ao separar a interface da lógica e esta dos dados através de fachadas independentes e DAOs especializados, conseguimos criar um sistema onde a rapidez no atendimento é suportada por uma organização eficiente da produção.

A estrutura em três subsistemas especializados, cada um com a sua própria fachada e responsabilidades bem definidas, garante que o sistema está preparado para ser utilizado em ambientes reais, oferecendo uma base estável para o crescimento da cadeia de restaurantes. O padrão de DAOs para persistência baseado em SQL assegura a escalabilidade e a facilidade de manutenção futura.