CENTRO PAULA SOUZA

FATEC MAUÁ

Tecnólogo em Desenvolvimento de Software Multiplataforma

JOÃO PEDRO SASSI GRANADO  
LUCAS MENDONÇA DA SILVA JUNIOR  
FELIPE DOUGLAS DOS SANTOS FELIX

DOCUMENTAÇÃO - GOLDCS

Mauá

2023

JOÃO PEDRO SASSI GRANADO  
LUCAS MENDONÇA DA SILVA JUNIOR  
FELIPE DOUGLAS DOS SANTOS FELIX

DOCUMENTAÇÃO - GOLDCS

Documentação do projeto GoldCS apresentado no Tecnólogo de Desenvolvimento de Software Multiplataforma da Fatec Mauá, orientado pelos professores Prof. Humberto Luiz de Toledo e Ivan Carlos pavão das disciplinas Desenvolvimento Web II e Engenharia de Software II.

Mauá

2023

ÍNDICE DE ILUSTRAÇÕES

[Figura 1: Exemplo figura colchão 4](#_Toc136516208)

[Figura 2: Arquitetura Back-End 7](#_Toc136516209)

[Figura 3: Logo GoldCS 8](#_Toc136516210)

[Figura 4: Casos de Uso 9](#_Toc136516211)

[Figura 5: Sequência - Login 10](#_Toc136516212)

[Figura 6: Sequência - Inserir pedido 10](#_Toc136516213)

[Figura 7: Sequência - Buscar pedido por ID 11](#_Toc136516214)

[Figura 8: Sequência - Inserir estoque 11](#_Toc136516215)

[Figura 9: Sequência - Gerar PDF pedido 12](#_Toc136516216)

[Figura 10: Sequência - Enviar Email 12](#_Toc136516217)

[Figura 11: Diagrama de Classes 13](#_Toc136516218)

[Figura 12: Modelagem DB 14](#_Toc136516219)

SUMÁRIO

[1. INTRODUÇÃO 4](#_Toc136516266)

[2. MODELO DE NEGÓCIOS 5](#_Toc136516267)

[3. BACK-END 6](#_Toc136516268)

[3.1. Tecnologias 6](#_Toc136516269)

[3.2. Arquitetura 6](#_Toc136516270)

[4. FRONT-END 8](#_Toc136516271)

[4.1. Logo 8](#_Toc136516272)

[4.2. Tecnologias 8](#_Toc136516273)

[5. UML 9](#_Toc136516274)

[5.1. Casos de Uso 9](#_Toc136516275)

[5.2. Diagramas de Sequência 10](#_Toc136516276)

[5.2.1. Login 10](#_Toc136516277)

[5.2.2. Inserir pedido 10](#_Toc136516278)

[5.2.3. Buscar pedido por ID 11](#_Toc136516279)

[5.2.4. Inserir estoque 11](#_Toc136516280)

[5.2.5. Gerar PDF do pedido 12](#_Toc136516281)

[5.2.6. Enviar Email 12](#_Toc136516282)

[5.3. Diagrama de Classes 13](#_Toc136516283)

[6. BANCO DE DADOS 14](#_Toc136516284)

[6.1. Modelagem do Banco 14](#_Toc136516285)

[BIBLIOGRAFIA 15](#_Toc136516286)

# INTRODUÇÃO

Este documento contém a documentação do GoldCS – Gold Colchões Software, sistema web responsável por fazer pedidos e controle de estoque das lojas de Colchão.

O projeto contém definições do Back-End, Front-End, além das modelagens UML e do banco de dados.

Figura : Exemplo figura colchão



Fonte: Portal Undraw, 2023

# Modelo de negócios

O modelo de negócios escolhido foi a franquia de loja Gold Colchões. Como diz o nome, a loja é pautada em vendas de colchões e outros produtos correlatos, como Cama, Travesseiros, Estofados e outros.

O atual modelo de Pedidos da loja é totalmente via papel, o que inviabiliza a expansão, além de gerar confusão e diversos erros manuais. Com isso em mente, criamos o sistema apresentado com a proposta de facilitar o controle de pedidos e estoque, além de possibilitar a expansão das operações.

O sistema, por se tratar de uma abordagem Web, poderia ter ido por diversos caminhos. O escolhido pela nossa equipe foi dividir a aplicação entre Front-End e Back-end.

# Back-end

O Back-End foi feito em base de uma Web API, fazendo com que ele possa ser versátil para estar em diversos clientes, desacoplando a camada visual com a camada lógica.

## Tecnologias

A principal tecnologia foi a plataforma .NET, onde pautamos a nossa ASP NET Core WEB API. Foi utilizado o DotNet 6, com a especificação do Swagger para documentação.

Além do C# como linguagem, diversos pacotes do Nuget foram usados, como o BCrypt para criptografia de senhas e o Fluent Validation para validação.

Outra tecnologia utilizada foi o PHP, que foi usado como um middleware entre a camada do front end e API e que disponibiliza os endereços do Brasil. Como esta API é pública, utilizamos o PHP para capturar a requisição e customizá-la de modo a não trazer informações desnecessárias e não popular o banco em vão.

Para autenticação e autorização, foi utilizado a tecnologia Bearer JWT, com Tokens. Para o Deploy na plataforma Render, foi utilizado um DockerFile.

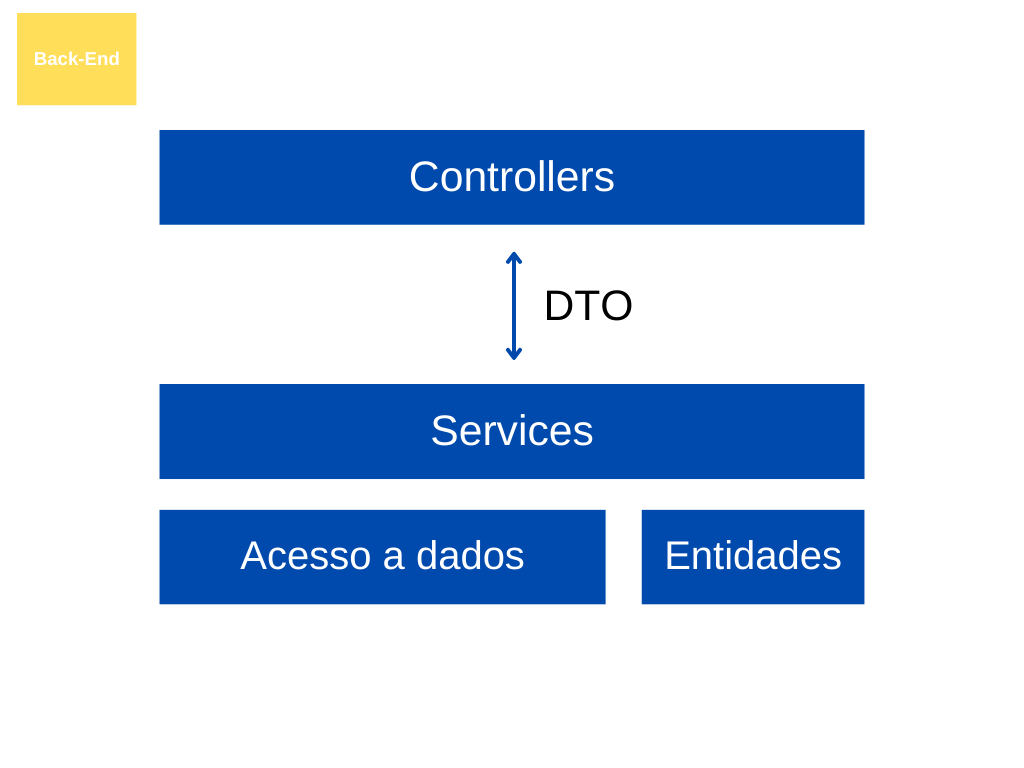
## Arquitetura

Arquitetura do back-end seguiu uma base clássica, contando com algumas camadas:

* Controllers: Essa camada é responsável por receber e tratar as requisições HTTP;
* Services: Essa camada é responsável por implementar as regras de negócio da aplicação;
* Repositories: Essa camada é responsável por implementar as lógicas de acesso a dados;

Além disso, utilizamos a técnica de DTO para fazer a transferência entre dados, deixando as entidades monitoradas pelo Entity Framework desacopladas das requisições.

Figura : Arquitetura Back-End



#### Fonte: Própria autoria, 2023

A documentação da API está disponível no Swagger nos seguintes endereços web:

* [Swagger UI (railway.app)](https://goldcs-production-api.up.railway.app/swagger/index.html)
* [Swagger UI (goldcsapi.onrender.com)](https://goldcsapi.onrender.com/swagger/index.html)

# front-end

Por termos feito uma Web API, o cliente escolhido foi uma solução JS, que faz as requisições HTTP para o back-end

## Logo

O logo, desenvolvido e utilizado:

Figura : Logo GoldCS



#### Fonte: Própria autoria, 2023

## Tecnologias

Para o Front-End utilizamos o ReactJS, junto de diversas bibliotecas. Algumas das principais foram:

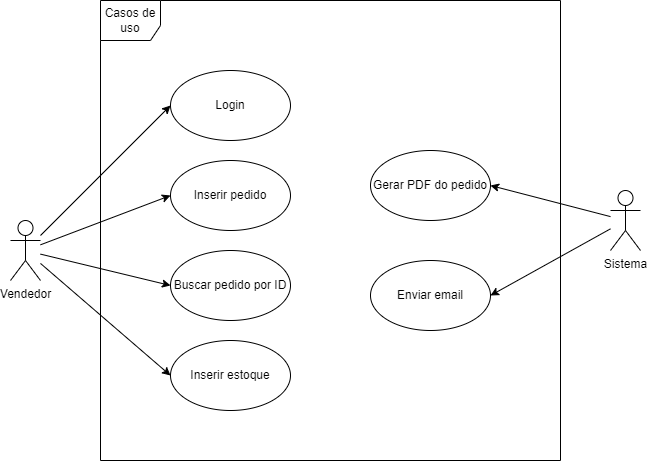
* React-Router-Dom: Fazer o roteamento das páginas
* Toastify: Mensagens do sistema

# uml

Os diagramas UML foram feitos usando o software Draw.io.

## Casos de Uso

Figura : Casos de Uso

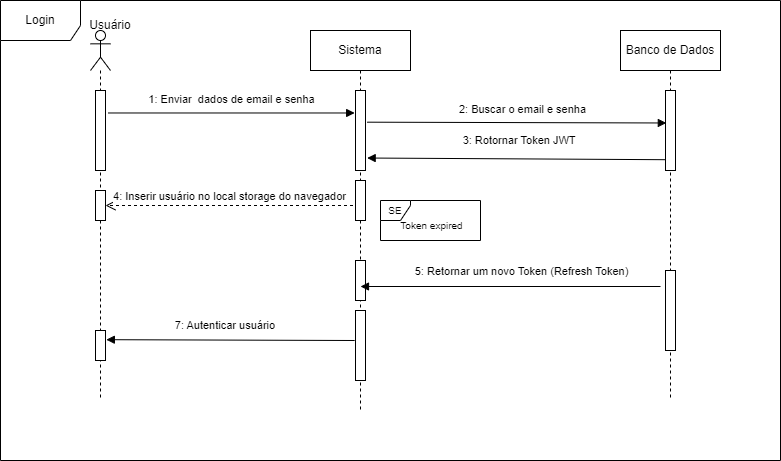


Fonte: Própria autoria, 2023

## Diagramas de Sequência

### Login

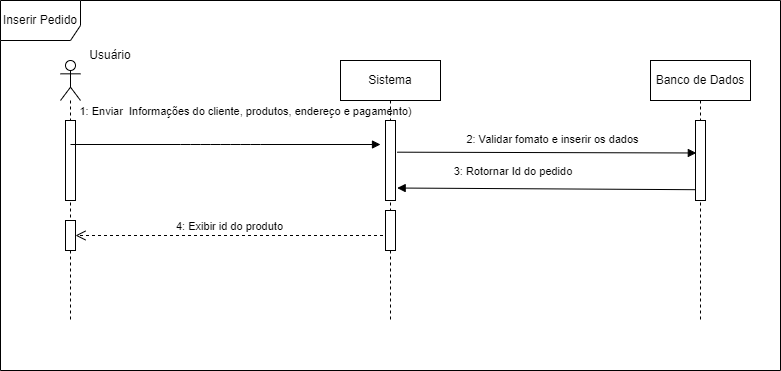
Figura : Sequência - Login



#### Fonte: Própria autoria, 2023

### Inserir pedido

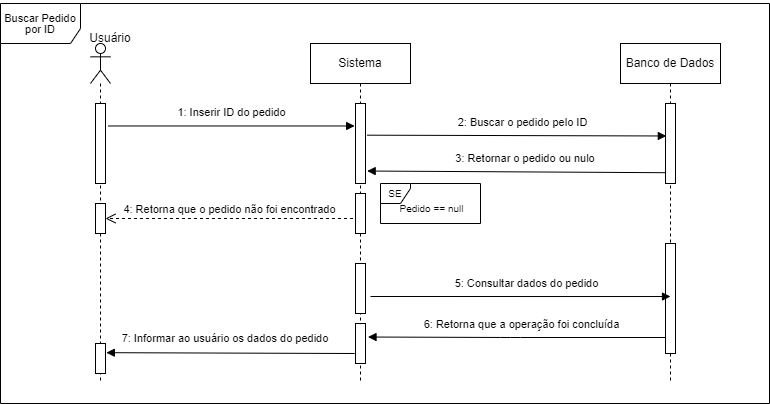
Figura : Sequência - Inserir pedido



#### Fonte: Própria autoria, 2023

### Buscar pedido por ID

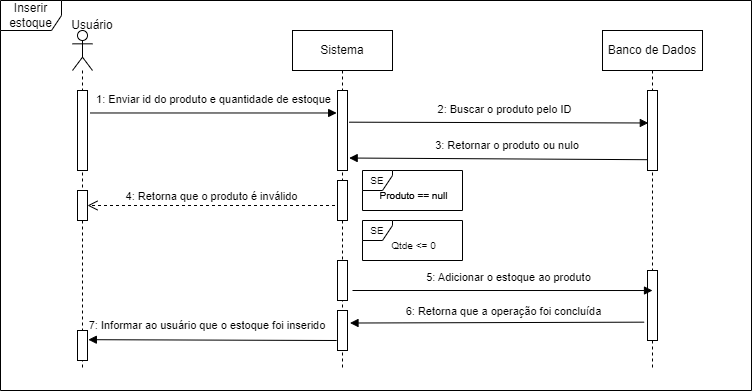
Figura : Sequência - Buscar pedido por ID



#### Fonte: Própria autoria, 2023

### Inserir estoque

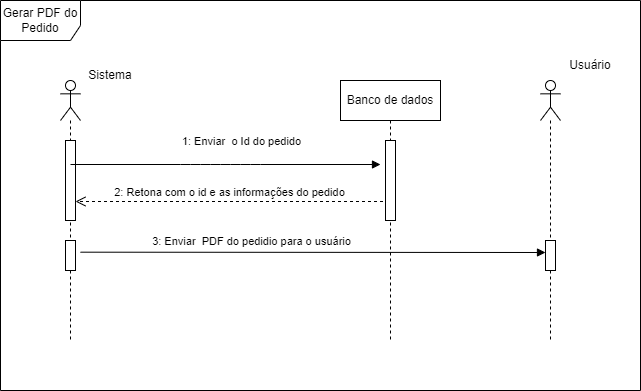
Figura : Sequência - Inserir estoque



#### Fonte: Própria autoria, 2023

### Gerar PDF do pedido

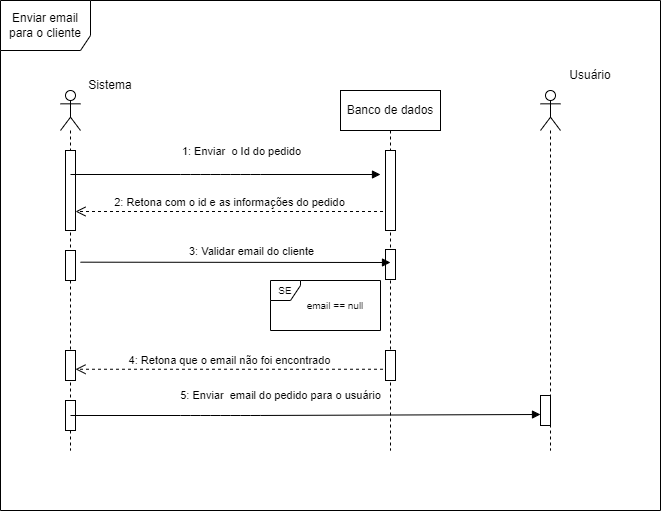
Figura : Sequência - Gerar PDF pedido



#### Fonte: Própria autoria, 2023

### Enviar Email

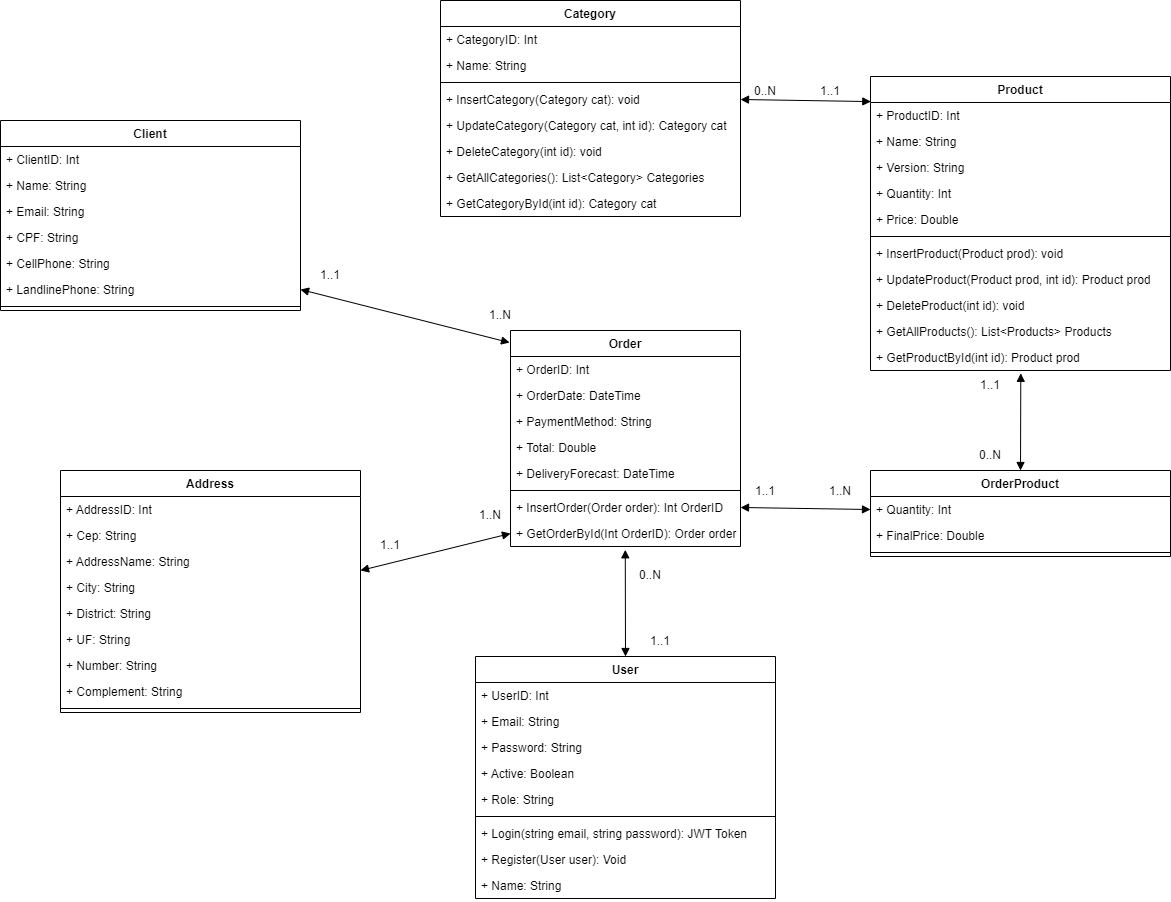
Figura : Sequência - Enviar Email



#### Fonte: Própria autoria, 2023

## Diagrama de Classes

Figura : Diagrama de Classes



#### Fonte: Própria autoria, 2023

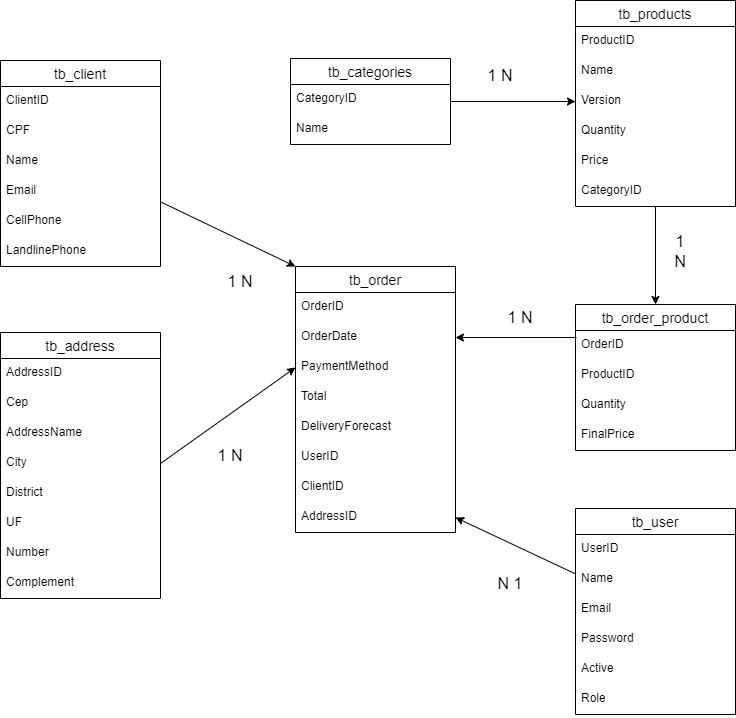
# Banco de dados

O banco utilizado inicialmente havia sido o SQL Server, mas com a realidade do Deploy, a solução foi migrada para o PostgreSQL. Por ter sido utilizado a abordagem de code first, utilizando das migrações para criação e modelagem do banco baseando-se nas nossas classes, o impacto foi mínimo.

Para tal utilizamos o Entity Framework ferramenta ORM da plataforma DotNet responsável por mapear o banco e externalizar a camada de acesso a dados.

## Modelagem do Banco

Figura : Modelagem DB



#### Fonte: Própria autoria, 2023

# BIBLIOGRAFIA

Este documento não se utilizou de bibliografia externa.