

E-Commerces
Documento de Arquitetura de Software
Versão 1.0

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

Revision History

Date	Version	Description	Author
23/10/2020	1.0	Primeira versão da documentação	Joseph Weber

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

Sumário

1.	Introdução	4
1.1	Objetivo	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
1.4	Visão Geral	4
2.	Visão de Caso de Uso e Requisitos	4
3.	Requisitos e Restrições arquiteturais	5
4.	Visão Lógica	5
4.1	Visão Geral No sistema e-commerces o servidor e o cliente possuem os seguintes papéis:	6
4.2	Pacotes de projeto com significado arquitetônico	6
4.3	Realizações de Caso de Uso	7
5.	Visão do Processo	7
6.	Visão de Implantação	8
7.	Qualidade	13

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

Documento de Arquitetura de Software

1. Introdução

1.1 Objetivo

Esse documento de arquitetura de software tem como objetivo apresentar e documentar a arquitetura de um Sistema de E-commerce, o primeiro trabalho avaliativo de Projeto e Arquitetura de Software do curso de Engenharia de Software da PUCRS. Para esta solução, foi utilizada uma arquitetura cliente-servidor, conforme falaremos abaixo.

1.2 Escopo

O Sistema tem como objetivo centralizar em uma única plataforma todas as compras feitas por um cliente, realizando o cadastro de pedidos dentro desta aplicação, facilitando assim o acompanhamento de suas compras online.

1.3 Definições, Acrônimos e Abreviações

E-Commerce: Sistema de comércio online.

API: Application Programming Interface

CRUD: Create, Read, Update e Delete

PUCRS: Pontifícia Universidade Católica do Rio Grande do Sul

1.4 Visão Geral

Este documento apresenta toda a descrição detalhada do projeto arquitetônico realizado pelos autores para a solução do Sistema, mostrando os diagramas realizados para representar a arquitetura, codificação, base de dados, interação do Sistema e requisitos.

2. Visão de Caso de Uso e Requisitos

Neste projeto utilizamos os casos de uso para representar o usuário e as ações possíveis dentro do Sistema em dois modos: logado e não logado. Também existe uma representatividade do Administrador.

Referência	Descrição
US01	O usuário precisa logar para ter acesso às funcionalidades do sistema
US02	O usuário deve estar autenticado através de um token
US03	O usuário deve poder se cadastrar no sistema
US04	O banco de dados deve ser H2
US05	O administrador deve ter permissão para cadastrar um e-commerce
US06	O sistema deve simular importação de pedidos para usuário
US07	Deve ser possível consultar todos os e-commerces que o sistema dá suporte

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

US08	Deve ser possível filtrar os pedidos do cliente por e-commerce e opcionalmente, pelo status do pedido
US09	Deve ser possível consultar detalhadamente as informações do produtos comprados do pedido
US10	O sistema deve consultar os pedidos das compras realizadas, por e-commerce, por data, por agilidade e por cumprimento de entrega
US11	O sistema deve se baseado em um arquitetura cliente-servidor
US12	O frontend da aplicação deve ser em React Native
US13	O backend da aplicação deve ser em Spring

3. Requisitos e Restrições arquiteturais

Nesta sessão serão apresentados os requisitos de software utilizado na arquitetura:

Requisito	Solução
Linguagem	Spring boot 2 (API Backend), React Native (Frontend)
Plataforma	Mobile, IOS ou Android
Segurança	Spring Security e JWT Token
Persistência	H2 (Banco de dados embutido)

Requisitos Funcionais:

- Cadastrar um e-commerce;
- Cadastrar os pedidos dos clientes (deverá ser “simulado” um mecanismo de importação de pedidos em aberto do cliente a partir de um e-commerce pré-cadastrado);
- Consultar pedidos do cliente, por e-commerce (estes pedidos deveria ser atualizados, “buscando” de cada ecommerce a informação do status do seu pedido – simular este comportamento.)
- Gerar relatórios de compras do cliente, por e-commerce, por data, por agilidade de entrega, por cumprimento de prazo de entrega.

Requisitos Não-Funcionais:

- Autenticação com criptografia;

4. Visão Lógica

A Arquitetura cliente-servidor é uma abordagem que separa os processos em módulos independentes que interagem entre si, o cliente e o servidor, permitindo o compartilhamentos e troca de mensagens afim de ter um maior benefício de cada parte.

O servidor fica responsável por realizar disponibilizar recursos e informações necessárias, faz o processamento das solicitações realizadas por máquinas clientes e responde conforme suas necessidades.

O Cliente é o responsável por iniciar e terminar a conversação com os servidores, realizando solicitações de entrada e obtendo como retorno as saídas.

Segue abaixo imagem de demonstração:

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

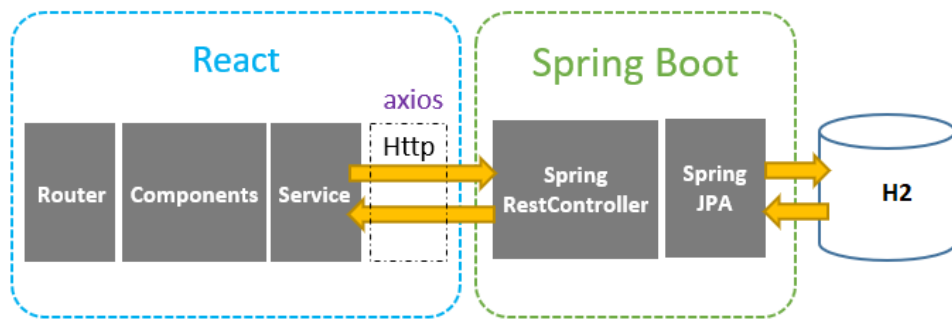


Figura 1 Arquitetura Cliente-servidor

4.1 Visão Geral

No sistema e-commerce o servidor e o cliente possuem os seguintes papéis:

Servidor:

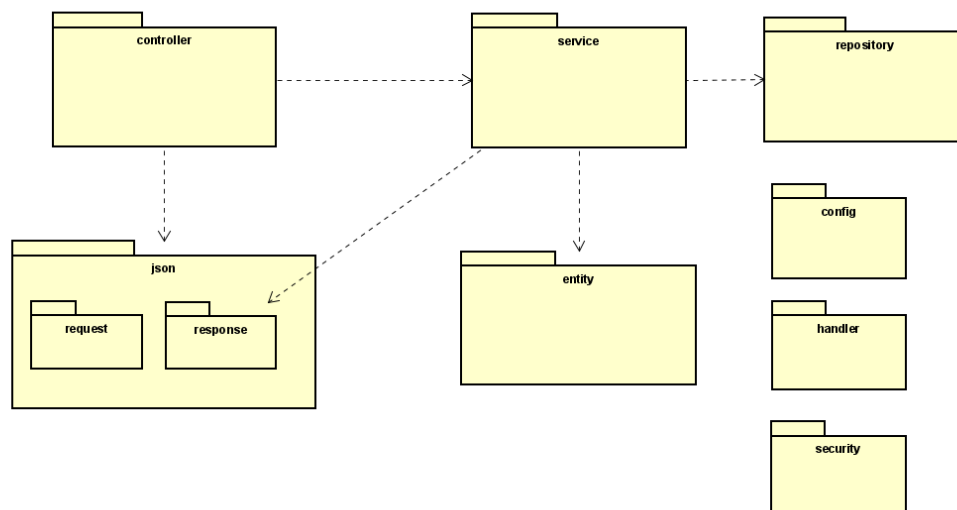
- Processa solicitações de cadastro de usuários e as responde se foi realizado;
- Processa solicitação de cadastro de e-commerce e as responde se foi realizado;
- Processa solicitação de cadastro de pedidos e as responde se foi realizado;
- Processa solicitação de consulta de e-commerce e responde com as consultas realizadas;
- Processa solicitação de filtros e responde com os filtros realizados;

Cliente:

- Envia solicitação de cadastramento de usuário e mostra o retorno do servidor;
- Envia solicitação de cadastro de e-commerce e mostra o retorno do servidor;
- Envia solicitação de cadastro de pedidos e mostra o retorno do servidor;
- Envia solicitação de consultas e mostra o retorno do servidor;
- Envia solicitação de filtro e mostra o retorno do servidor;

4.2 Pacotes de projeto com significado arquitetônico

Nesta sessão será apresentada brevemente como foi modelada os pacotes do projeto:



E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

Figura 2 Diagrama de Pacotes

4.3 Realizações de Caso de Uso

Nesta sessão será apresentada o diagrama de casos de uso, representando a interação do usuário com a aplicação, mostrando as interações de caso de uso que o usuário é envolvido durante o fluxo da aplicação:

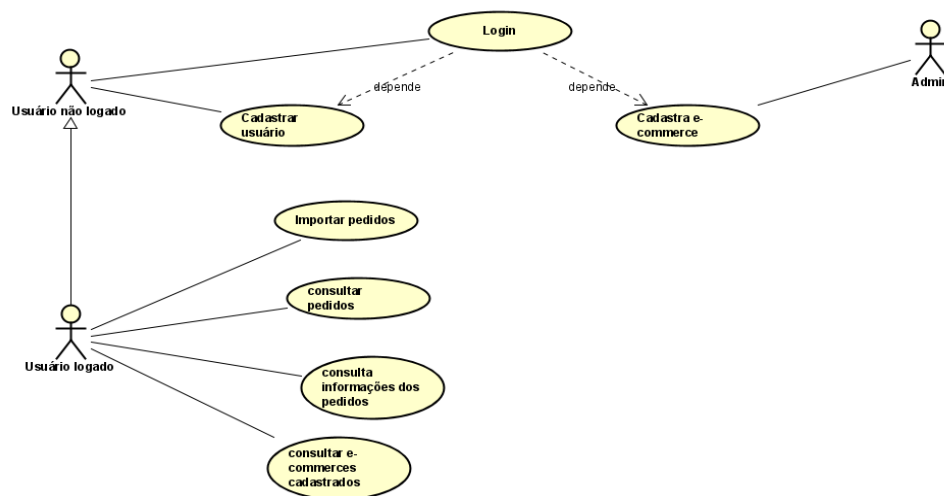


Figura 3 Diagrama de Casos de Uso

5. Visão do Processo

A arquitetura é baseada em padrão API-REST, ou seja, um conjunto de definições a serem utilizadas para a criação de web services.

Utilizando Como base o a arquitetura de Cliente-Servidor e Rest-API, criamos uma API em backend utilizando Springboot 2, que permite a comunicação com uma camada de persistência h2, (banco de dados embutido no springboot) e realiza o response no formato JSON para ser consumido no frontend codificado em React Native.

O Cliente (frontend), também envia request no formato JSON para ser processados pelo Servidor (backend) e assim é realizado a comunicação do cliente com o server.

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

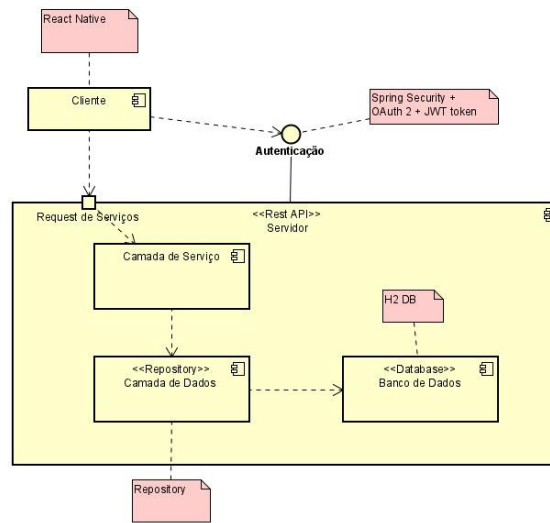


Figura 4 Diagrama de Componentes

6. Visão de Implantação

Nesta sessão será apresentada como foi pensada a modelagem da Implantação do sistema, conforme imagem abaixo:

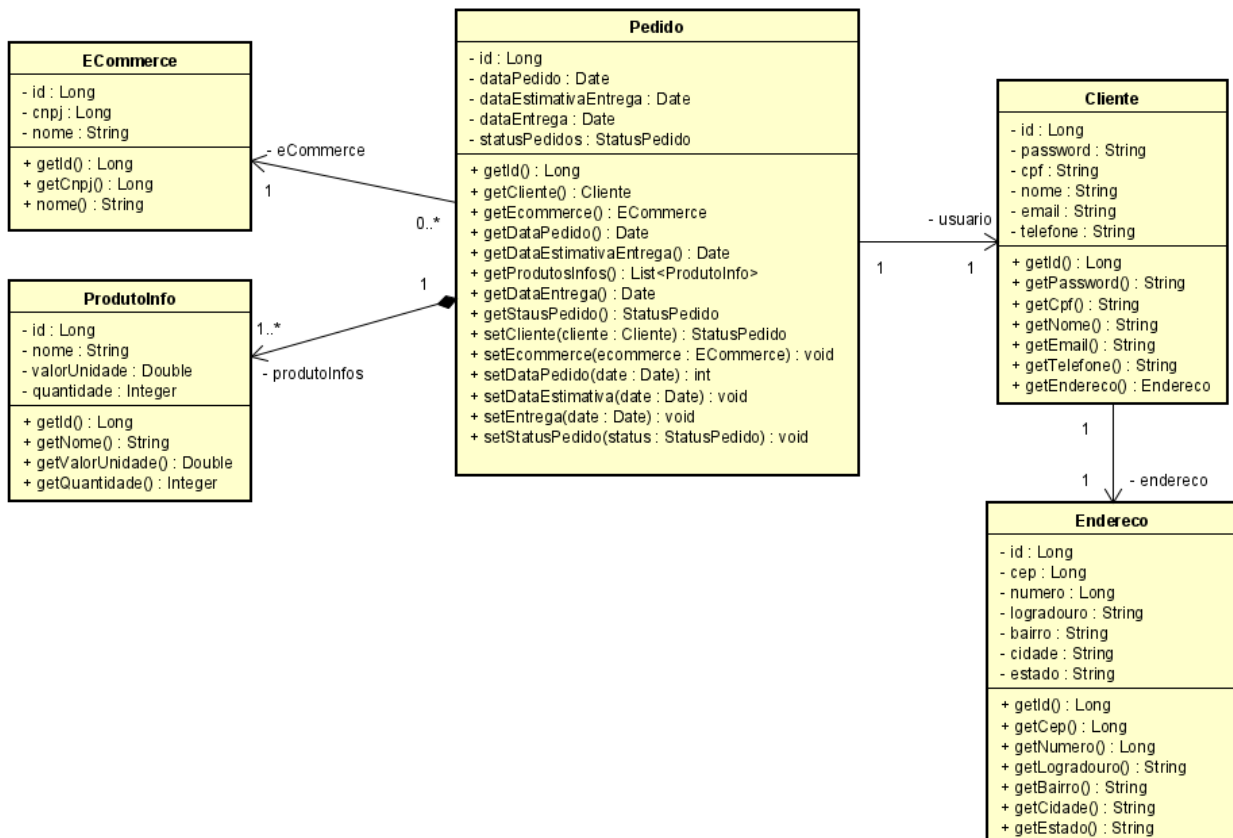


Figura 5 Diagrama de Classes

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

Classe Pedido:

- Responsável por pelas informações do pedido, possuindo o e-commerce que foi realizado a compra, as informações do produto que foi comprado e o status do pedido e suas principais funções;

Classe ProdutoInfo:

- Armazena todas as informações do Produto, como a quantidade, valor e o nome e suas principais funções;

Classe E-commerce:

- Armazena todas as informações do e-commerce, como nome e cnpj e suas principais funções;

Classe Cliente:

- Armazena as informações dos clientes, como nome, cpf, endereço e também suas principais funções

Classe Endereço:

- Representa o endereço do Sistema.

Projeto foi dividido em Front-end e Back-End

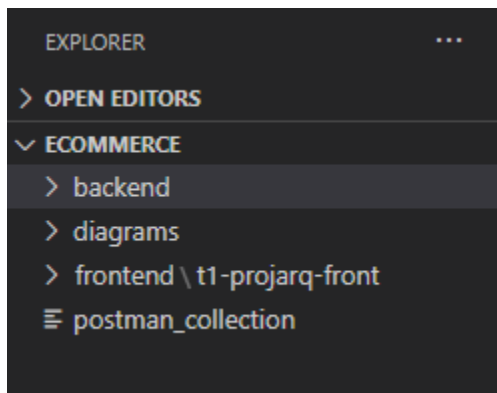
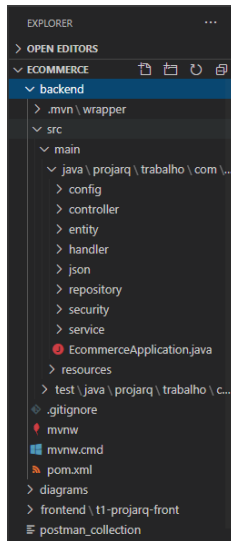


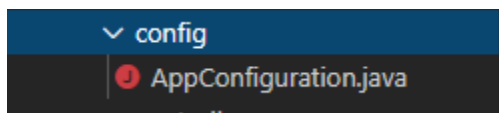
Figura 6 VSCode - Backend Frontend

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

No Back-End possuímos as seguintes divisões:

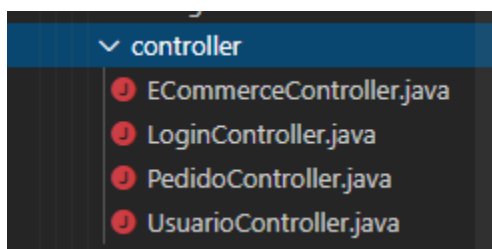


Config:



- Classes de configurações iniciais do sistema, como localidade;

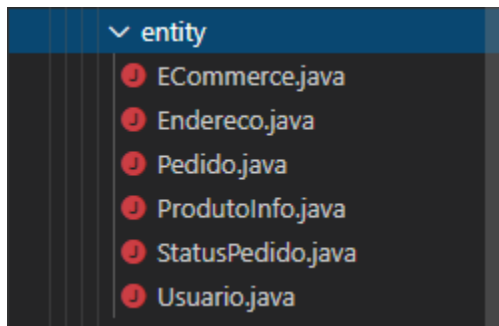
Controller:



- Possui as classes que permitem as requisições da API

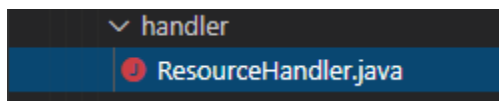
E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

Entity:



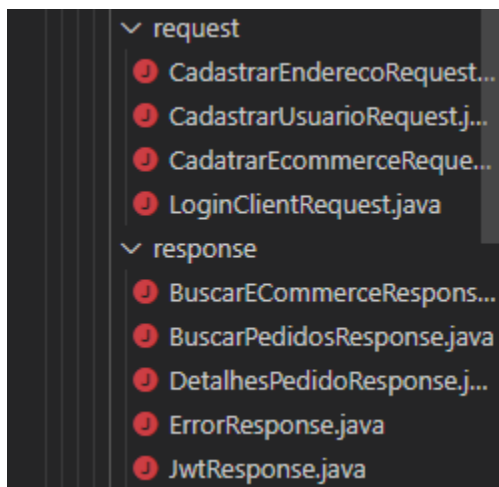
- Armazenas as informações dos objetos principais;

Handler:



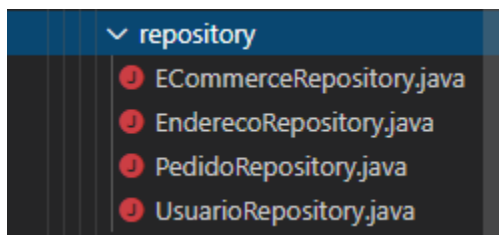
- Possui a classe de manipulação.

Json:



- Possui as classes para request e reponse da aplicação:

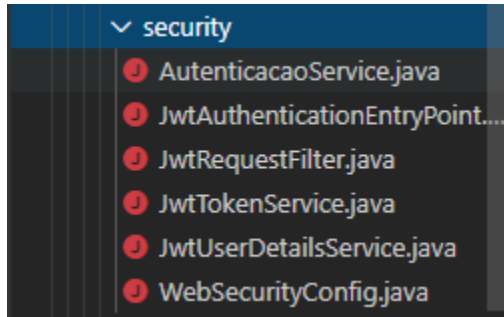
Repository:



E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

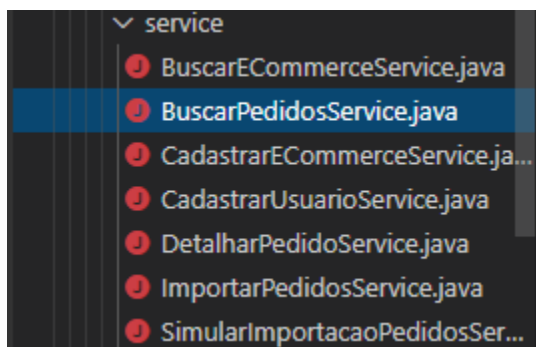
- Possui as classes para os CRUD do sistema.

Security:



- Possui as classes de autenticação (segurança) do usuário;

Service:



- Manter os serviços do sistema;

Bando de Dados H2:

H2 é um banco de dados embutido, portando o mecanismo de acesso ao banco de dados roda junto com a aplicação, ao em invés de ter uma outra aplicação que deva se comunicar para acessar um banco de dados.

Script do banco gerado pelo H2 dentro do contexto hibernate na aplicação Spring

E-Commerces	Versão: 1.0
Documento de Arquitetura de Software	Data: 23/10/2020
Joseph Weber	

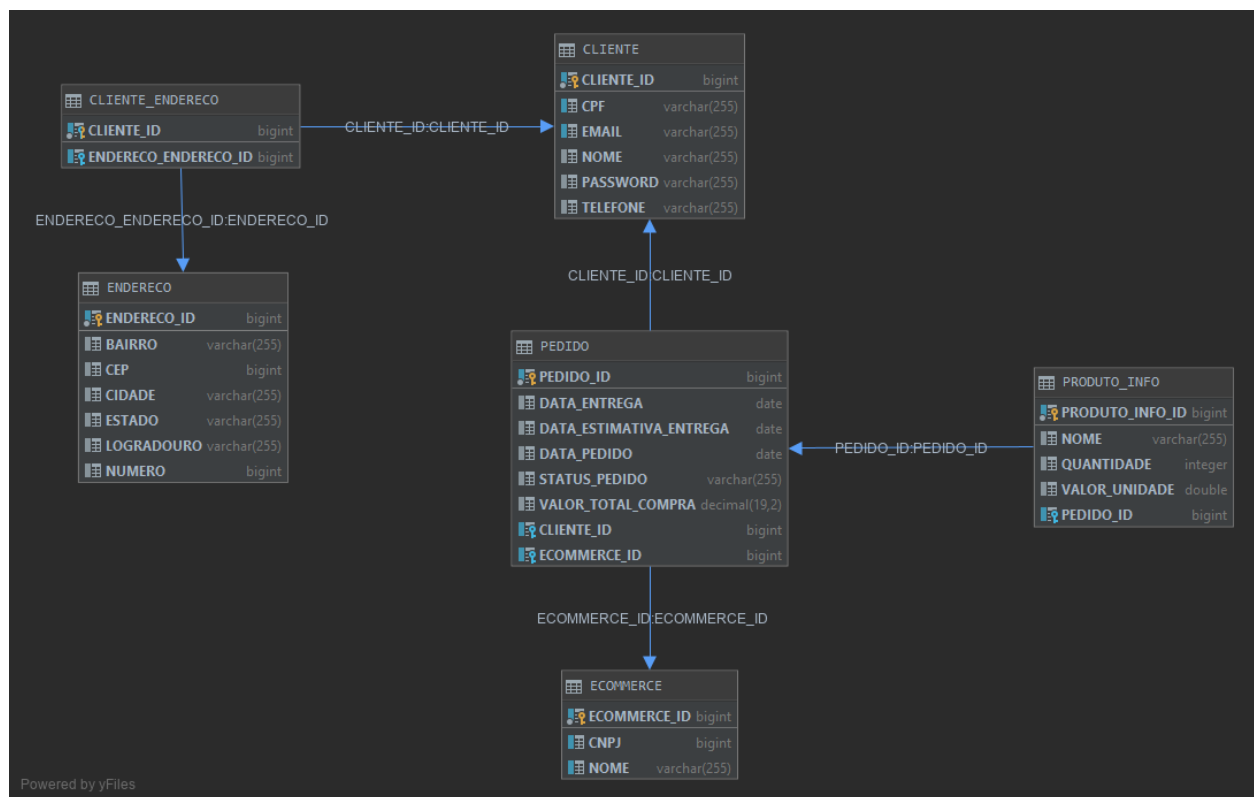


Figura 7 Banco de Dados

7. Qualidade

Nesta sessão apresentaremos alguns atributos de qualidade:

Item	Descrição	Solução
Segurança	Sistema possui autenticação de usuário criptografada.	Spring Security e JWT Token
Interoperabilidade	Sistema possui comunicação entre Front e Back-end	Padrão Rest-API, Spring boot 2, React Native e Axios