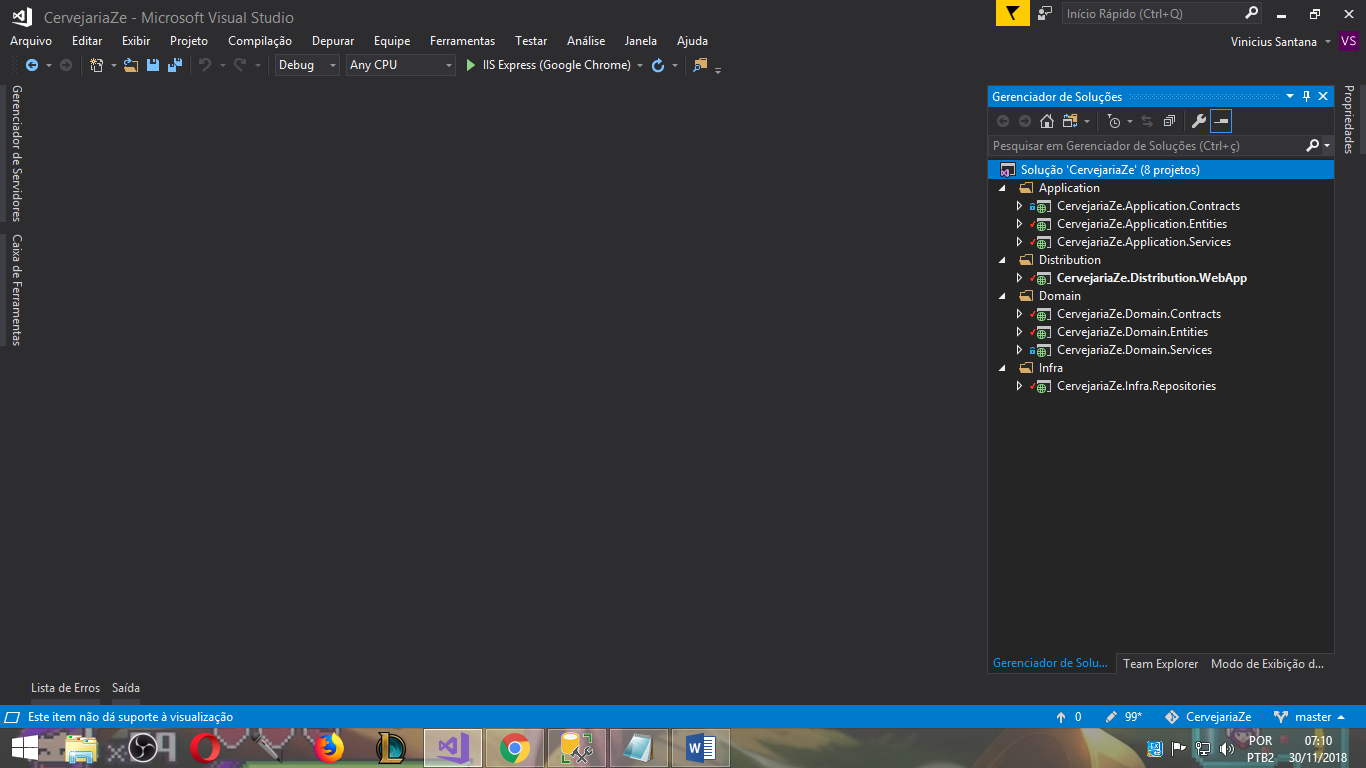
Documento do Projeto

**Arquitetura:**



A organização das pastas é similar a arquitetura Onion onde cada pasta significa uma camada da “cebola” que tem sua função:

Distribution – Camada Web onde se encontra o projeto WEB API onde os controllers delegam o serviço que cada método vai enviar seus dados.

* WebApp - projeto REST responsável pela comunicação com o projeto FrontEnd externo feito em Angular por requisição HTTP.

Application – Camada de Serviço que faz regra de negócio relacionada à camada Web.

* Contracts – projeto onde se encontra os contratos de serviço da Application.
* Entities – projeto onde se encontra as entidades de entrada e saída de dados.
* Services – projeto onde se encontra a implementação dos serviços que fazem a regra de negócio da camada Web.

Domain – Camada de Serviço que recebe os dados da camada Application e faz a regra de negócio para serem enviados para o repositório.

* Contracts – projeto onde se encontra os contratos de serviço e repositório da camada Domain e Infra.
* Entities – projeto onde se encontra as entidades de repositório.
* Services – projeto onde se encontra a implementação dos serviços que fazem a regra de negócio do repositório.

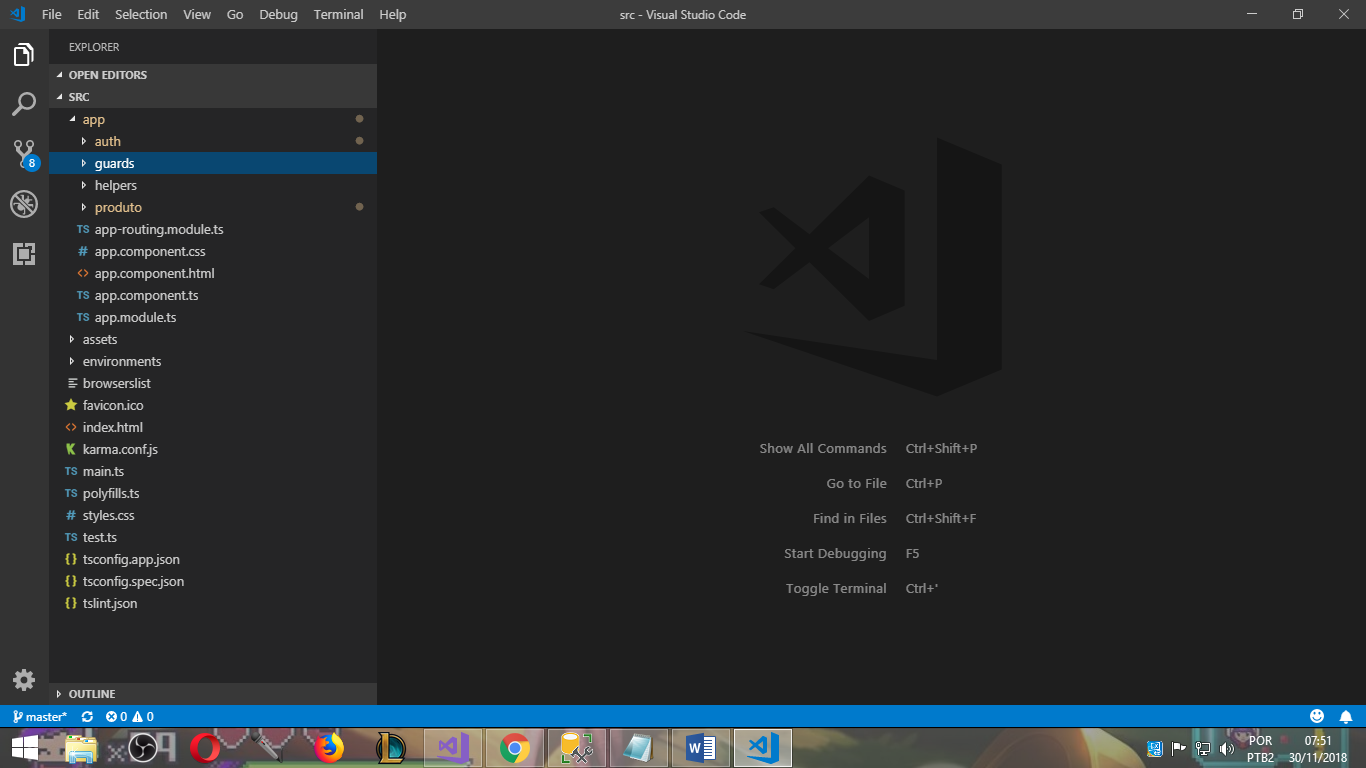
Infra – Camada responsável por persistir os dados na base de dados, fazer o mapeamento das tabelas e criação do contexto.

* Repositories – projeto onde se encontra as classes de contexto, mapeamento e implementação dos repositórios.

Obs.: A arquitetura não pode ser considerada Onion pois ela não implementa a injeção de dependência e inversão de controle, mas segue o princípio das referências verticais

A camada de infra enxerga a camada de Domain, a camada de Domain enxerga a camada de Application e a mesma enxerga a camada de Distribution.

**Arquitetura WEB:**



No front foi utilizado o Angular 2, projeto criado pelo angular CLI.

Auth: Encontra-se o componente responsável pela autenticação do usuário ao sistema.

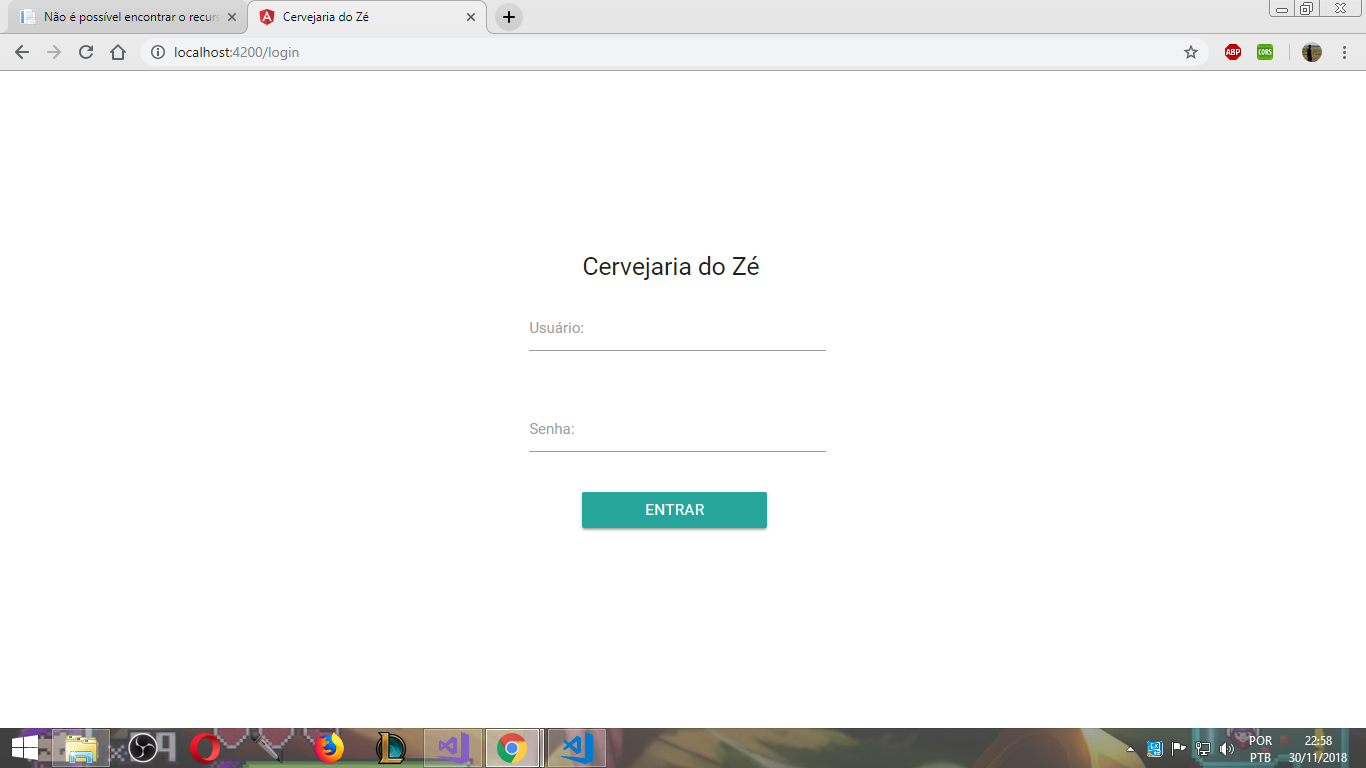
Guards: Encontra-se o serviço responsável por verificar se o token do usuário não expirou e manter ele autenticado.

Helpers: Encontra-se o serviço JWT onde o mesmo vai interceptar a requisição e adicionar o token de autenticação para acessar as funcionalidades permitidas.

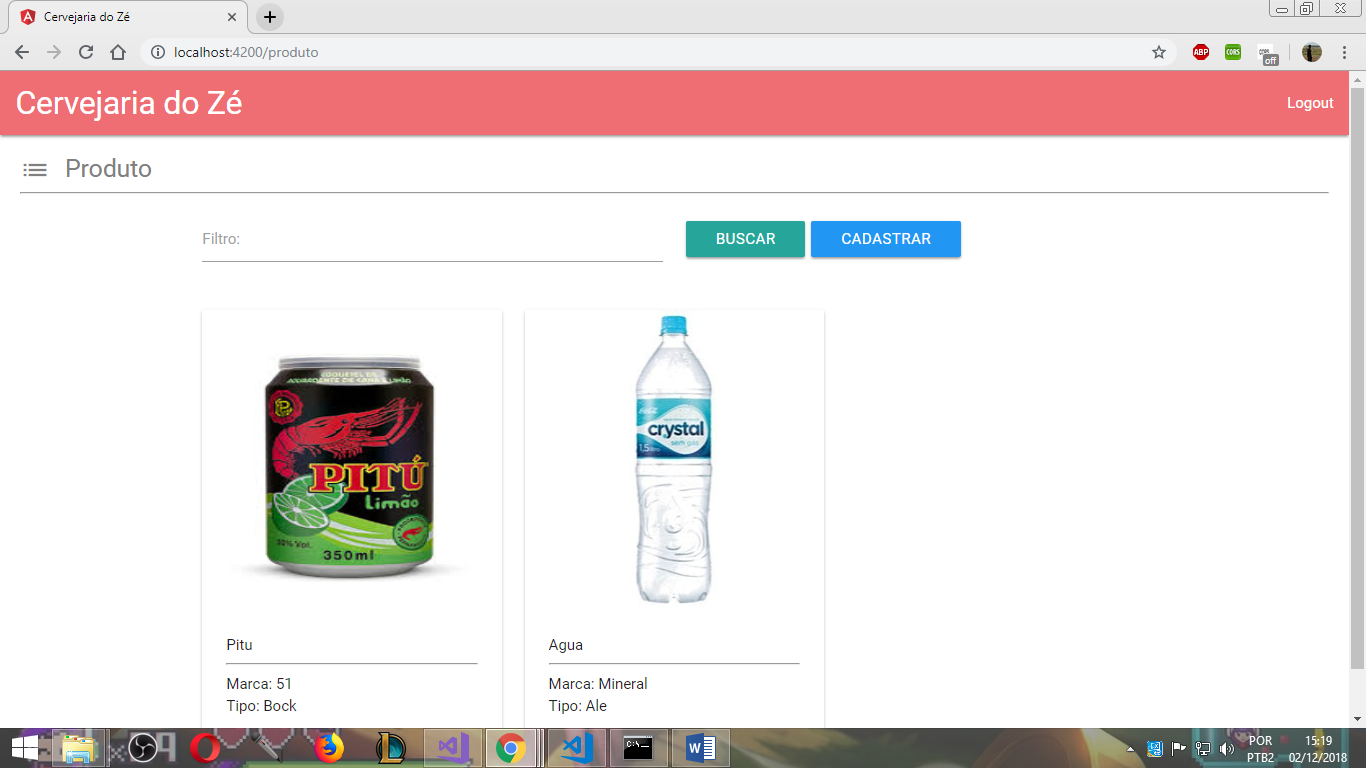
Produto: Encontra-se o componente de listagem e na pasta filha Cadastrar se encontra o componente de cadastro de produtos.

Telas

**Login:**



**Listar e Filtrar Produto:**



**Cadastrar Produto:**

