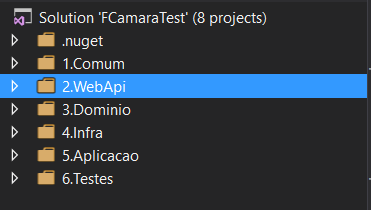
# Teste FCamara

Foi desenvolvido uma API para realização de Login com validação via TOKEN.

A duração do token está fixa de 1 minuto, pois foi determinada pelo cliente.

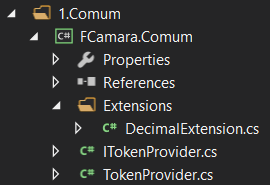
Segue abaixo o a arquitetura da solução do backend:



*Arquitetura da Solução*

Cada camada foi desenvolvida de acordo com a sua responsabilidade para facilitar a manutenção da aplicação.

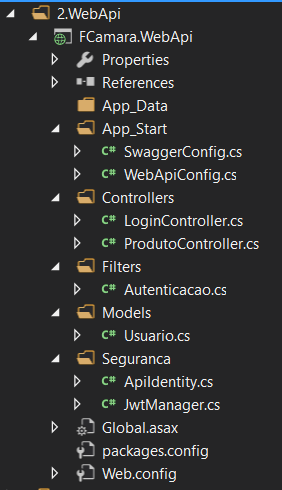
## Comum



*Camada Comum da Aplicação*

A camada comum da aplicação tem a função de prover/gerar o token de autenticação para acesso da aplicação.

## WebApi



*A camada de Web API da aplicação*

*Esse projeto é responsavel por atender todas as requisições HTTP da aplicação. No caso realizar o login e também obter os produtos cadastrados.*

*Para o teste da API utilizei uma ferramenta chamada POSTMAN*

* *Configurando a aplicação:*
  + *No Package Manager Console executar o comando* ***Update-Database*** *para executar as migrations.*
  + *Entrar em um programa para executar os scripts SQL tal como: SQL Server Management Studio e executar os scripts abaixo:*

insert into dbo.Usuarios (Nome, Login, Senha)

values ('FCamara Teste', 'fcamara', 'fcamara')

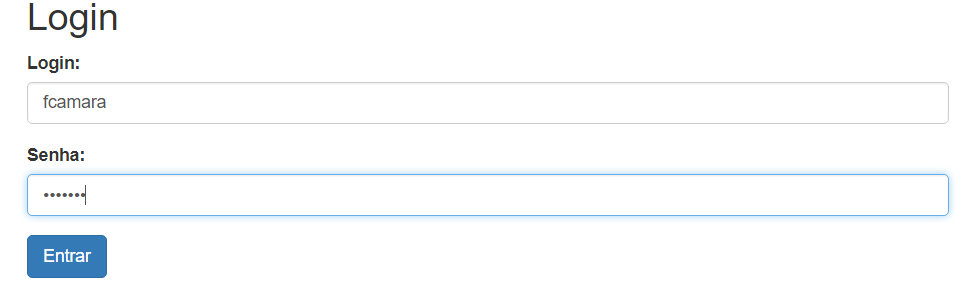
insert into dbo.Produtos (Nome, Descricao, Preco)

values ('Notebook Dell 14 pol', 'Intel Core i5, 8GB, 1TB', 2500.00)

insert into dbo.Produtos (Nome, Descricao, Preco)

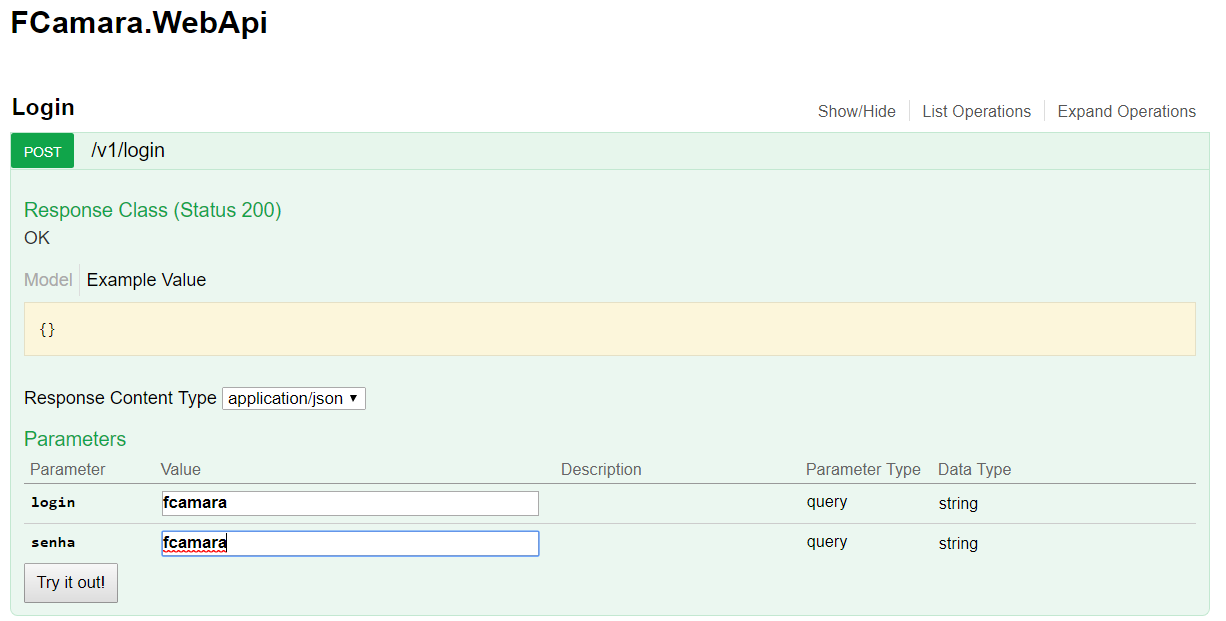
values ('Impressora Deskjet HP', 'Modelo J100', 279.5)

* *Testando a API via tela:*



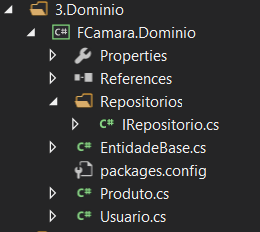
*Teste de login via localhost*

* *Testando a API via SWAGGER:*



*Teste via Swagger*

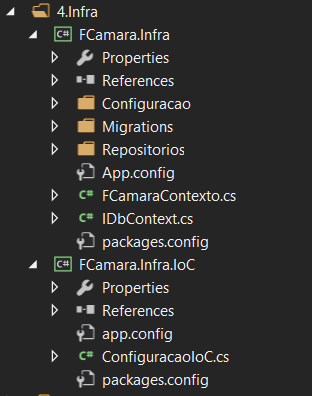
## Repositorio



Camada de Dominio

A camada de Dominio tem como objetivo representar todas as classes pertinentes a solução.

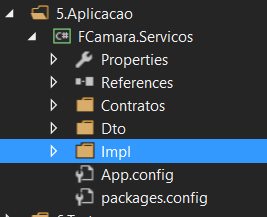
# Infraestrutura



Camada de Infraestrutura

A camada de infraestrutura representa nosso a relação entre a nossa aplicação e o banco de dados. É nela que aplicamos nossas alterações (migrations) e também configuramos nossos repositorios.

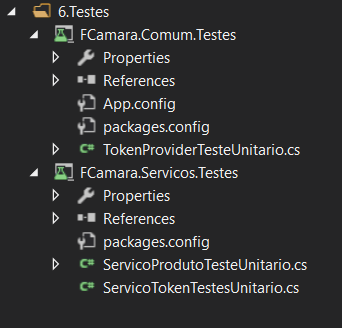
## Aplicação



Camada de Aplicação

A camada de Aplicação representa o serviços/trabalhos que aplicação irá exercer. Todos os métodos, contratos e objetos que irão se comunicar entre as outras camadas ficam nessa camada.

## Testes



Camada de Testes

Foram criados testes unitários para garantir a qualidade da aplicação, garantindo que todas as funcionalidades estejam funcionando corretamente. Foi utilizado o design pattern TDD (Test Driven Development).

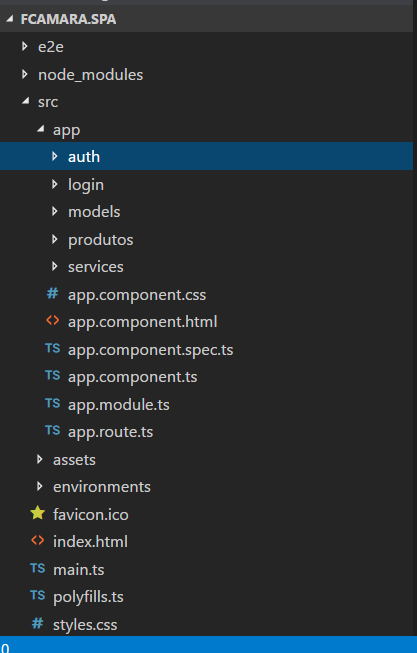
# SPA – Utilizando Angular JS

O client da aplicação foi realizado em Angular js.

Pré requisitos para utilização:

* Node js 6.9.0 ou superior
* NPM 3.0 ou superior

## Arquitetura da solução



* No diretorio client da aplicação via prompt de comando utilize o comando ng serve para iniciar a aplicação.