

Documentação

WISHLIST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alameda Barão de Limeira, 539 – São Paulo/SP |  |  |  | sp.senai.br |

Sumário

1. [Resumo 2](#_Toc3882857)

[Objetivos](#_Toc3882858)

1. [Descrição do projeto 2](#_Toc3882859)

[Resumo do projeto](#_Toc3882860)

1. [Modelagem de Software 3](#_Toc3882861)

[Modelo Lógico](#_Toc3882862)

[Modelo Físico](#_Toc3882863)

[Modelo Conceitual](#_Toc3882864)

[Cronograma](#_Toc3882865)

1. [Back-End 7](#_Toc3882866)

[Passo a passo de utilização](#_Toc3882867)

1. [Funcionalidades 12](#_Toc3882868)

[Web](#_Toc3882869)

[Mobile](#_Toc3882870)

1. [Protótipos 13](#_Toc3882871)

[Web](#_Toc3882872)

[Mobile](#_Toc3882873)

1. [Front-End 14](#_Toc3882874)
2. [Mobile 15](#_Toc3882875)
3. [Arquitetura do Projeto 16](#_Toc3882876)
4. [Referências 17](#_Toc3882877)

[Links](#_Toc3882878)

[Livros](#_Toc3882879)

Resumo

Objetivos

O objetivo deste documento é reunir as informações estruturais necessárias para o banco de dados e para a aplicação desenvolvida, bem como suas requisições.

Descrição do projeto

O projeto consiste no controle e utilização de um sistema de lista de desejos, reunindo as informações referentes aos usuários do sistema e a lista em si.

Resumo do projeto

Até o momento foi realizada a modelagem e estruturação do banco de dados.

Em seguida foi desenvolvido o código do sistema para que seja possível realizar requisições ao banco de dados através de uma ferramenta de simulação consumindo uma API.

Neste caso foi utilizado o Postman.

Modelagem de Software

A modelagem de software é a definição de todas as instruções que serão necessárias para sua implantação, visando aplicação atual e também ampliação.

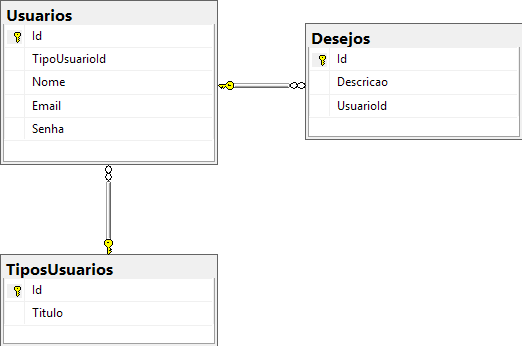
Modelo Lógico

O modelo lógico é estrutura das entidades com seus atributos (colunas da tabela), exibindo as chaves primárias e as chaves estrangeiras, além de ilustrar qual chave estrangeira está relacionada a qual entidade.



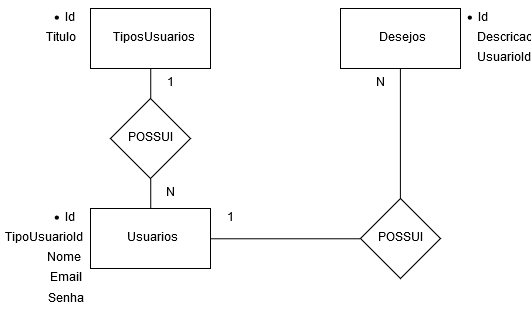
Modelo Físico

O modelo físico é o diagrama com as entidades e seus relacionamentos, criado a partir da estruturação do banco de dados via SQL Server.



Modelo Conceitual

O modelo conceitual é a forma simplificada da estrutura do banco de dados, ilustrando as entidades com seus atributos, as relações entre elas e suas cardinalidades. É o primeiro passo na criação de um banco de dados.



Cronograma

|  | Dia 1 | Dia 2 | Dia 3 |
| --- | --- | --- | --- |
| **MODELAGEM DE BANCO DE DADOS** |  |  |  |
| **DESENVOLVIMENTO DA API** |  |  |  |
| DESENVOLVIMENTO DA APLICAÇÃO WEB |  |  |  |

**Trello**:

https://trello.com/b/sZkroexy/senaiwishlistdesafio

Back-End

O código do sistema foi desenvolvido através de uma API utilizando o Microsoft Visual Studio.

API é um conjunto de padrões e instruções estabelecidos para utilização do software, definindo as requisições e as respostas seguindo o protocolo HTTP, neste caso expresso no formato JSON, para que seja possível acessar o sistema em diversos dispositivos distintos sem a preocupação com a linguagem que será utilizada por estes.

Além disso, foi utilizado o estilo de arquitetura REST.

**API** – Application Programming Interface – Interface de Programação de Aplicativos.

**HTTP** – Hypertext Transfer Protocol – Protocolo de Transferência de Hipertexto.

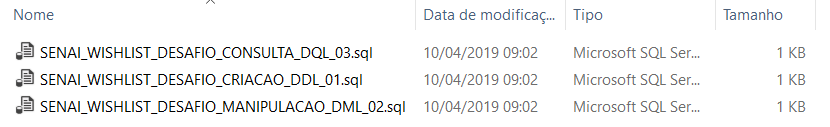
**JSON** – JavaScript Object Notation – Notação de Objetos JavaScript.

**REST** – Representational State Transfer – Interface de Programação de Aplicativos.

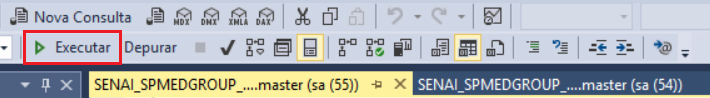
Passo a passo de utilização

**Executando o banco de dados:**

1. Abrir scripts do banco de dados

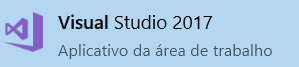


1. Executar cada script (DDL, DML, DQL) através do botão destacado.

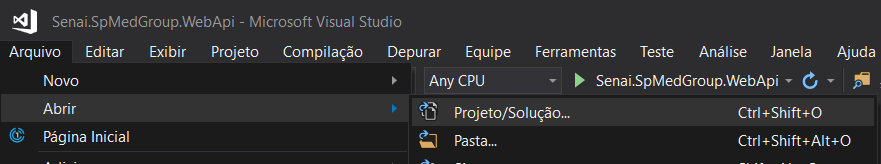


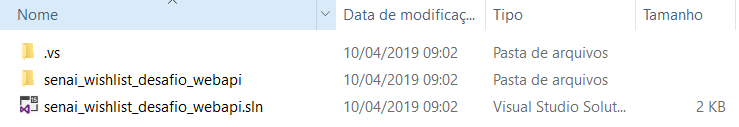
**Abrindo a solução:**

1. Abrir o Visual Studio 2017



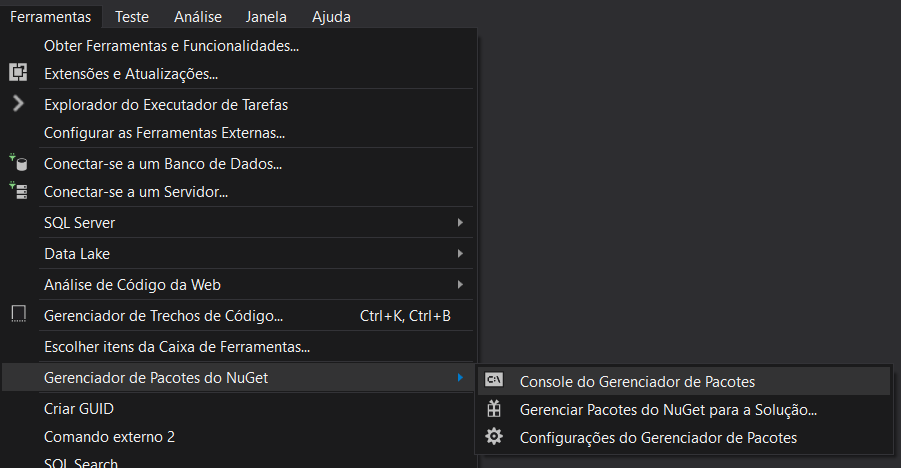
1. Abrir o projeto





**Se houver alteração no banco de dados:**

1. Selecionar o Console do Gerenciador de Pacotes do NuGet através da aba ferramentas



1. Executar o comando:

Scaffold-DbContext "Data Source=.\SqlExpress; Initial Catalog=SENAI\_WISHLIST\_DESAFIO; user id=sa; pwd=132;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Domains -ContextDir Context -Context WishlistContext –force

O comando acima cria as pastas Context e Domains com as classes necessárias de acordo com o banco de dados.

Comando: Scaffold-DbContext

String de conexão: "Data Source=.\SqlExpress; Initial Catalog=SENAI\_WISHLIST\_DESAFIO; user id=sa; pwd=132;"

Provedor: Microsoft.EntityFrameworkCore.SqlServer

Pasta onde ficarão as classes dos domínios: -OutputDir Domains

Pasta onde ficarão as classes dos contextos: -ContextDir Context

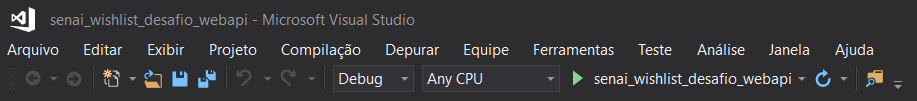
Contexto: -Context WishlistContext

Persistência na alteração: –force

Obs.: verificar se o nome do servidor (.\SqlExpress), o usuário (user id) e senha (pwd) corresponde com o acesso do servidor do banco de dados.

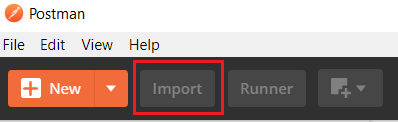
**Executando o projeto:**

1. Executar a solução através do botão destacado no Visual Studio

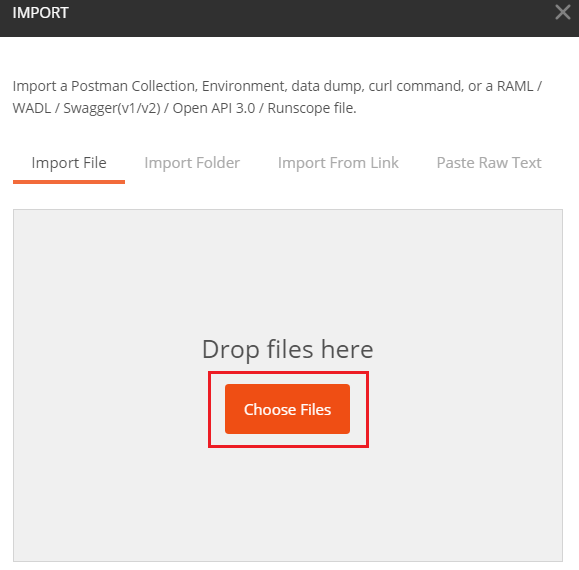


**Importando a coleção de requisições para o Postman:**

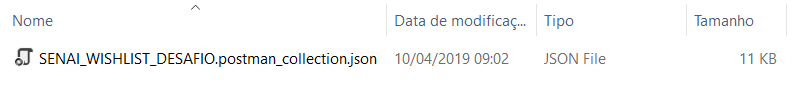
1. Importar através do botão destacado



1. Escolher o arquivo através do botão destacado ou arrastar o arquivo para a janela

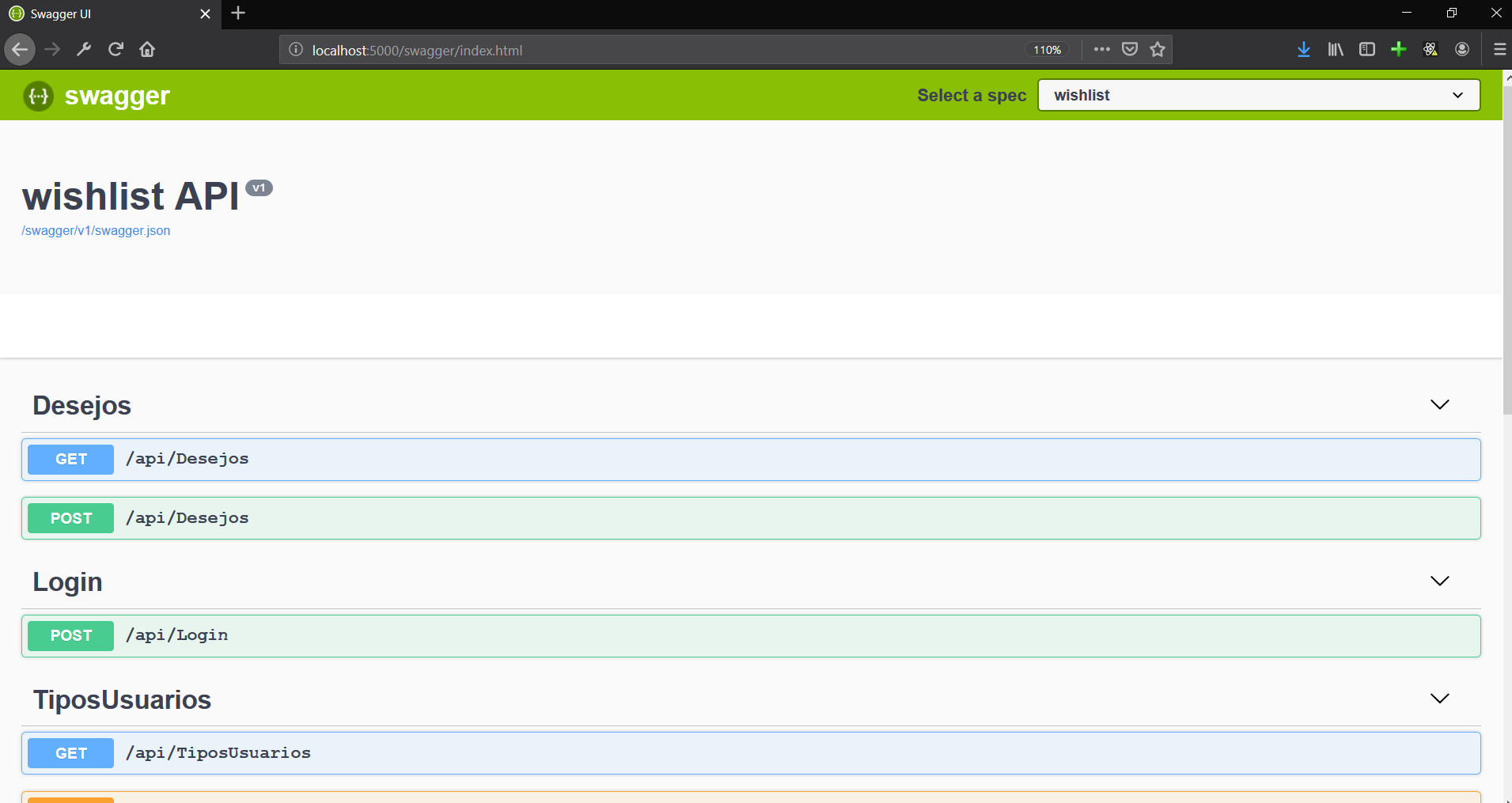


1. Escolher o arquivo



**Exibindo a documentação pelo Swagger:**

1. Com o projeto em execução, acesse o endereço <http://localhost:5000/swagger/index.html>



Funcionalidades

Web

1. O administrador poderá listar todos os tipos de **usuários**;
2. O administrador poderá cadastrar qualquer tipo de **usuário (administrador ou comum)**;
3. O administrador poderá editar qualquer tipo de **usuário (administrador ou comum)**;
4. O administrador poderá listar todos os **usuários**;
5. O administrador poderá cadastrar um **usuário (administrador ou comum)**;
6. O administrador poderá editar um **usuário (administrador ou comum)**;
7. Qualquer usuário autenticado poderá listar todos os desejos;
8. Qualquer usuário autenticado poderá cadastrar um desejo;

Protótipos

Web

Mobile

Front-End

Mobile

Arquitetura do Projeto

Referências

Links

Livros