**Rascunho Angular Aplicação**

passos realizados para montar a aplicação:

Criar uma pasta que guardara o backend e frontend da aplicação;

dentro da pasta, criar uma nova pasta de backend que guardará as configurações:

- node\_modules, que contem os arquivos json, tais como package.json, json-server, e o arquivo contendo os dados, no caso o bd.json. Os comandos utilizados foram: npm init -y - responsavel por criar o package.json- npm i json-server - responsavel por criar o json-server(?)

Dentro do backend foi criado um arquivo 'db.json' que guarda os dados dos objetos da aplicação, como o id, name e price do objeto 'product'.

Dentro da pasta package.json foi criado um script chamado 'start' responsavel por iniciar o arquivo db.json. O comando utilizado foi ' json-server --watch db.json --port 3001'. Esse comando quer dizer que o json-server ficara monitorando o arquivo db.json na porta 3001.

Para inicar a API foi executado o comando 'npm start' no promt.

Os testes nessa API pode ser feita utilizando o postman:

- Basta copiar a url que o comando acima tras;

- Definir o método (POST, GET, PUT ou PATCH) e realizar as operações.

O GET é utilizado para "trazer" os dados do arquivo db.json. O POST é utilizado para inserir dados novos. O PUT é utilizado para alterar o objeto inteiro. O PATCH é utilizado para alterar algum dado específico do objeto, basta especificar o id do objeto na url.

TypeScript é um superset do JavaScript, uma linguagem orientada a objetos e fortemente tipada, compilado por JS.

Tudo necessario para criação de modulos e componentes do angular, é feito pelo CLI (command line interface). Para cria-lo basta executar o comando npm i -g @angular/cli

Para criação do frontend utilizando angular, foi executado o comando ng new frontend --minimal no promt. Para inicializar foi utilizado o comando npm start

o main.ts aponta para o app.module que aponta para o app.component

layout utilizado na aplicação: ng add @angular/material

para adicionar os componentes de header basta executar o comando ng g c components/template/header na pasta frontend

para criar os componentes de header, footer e nav foi utilizado o comando ng g c components/template/header ou footer ou nav

o componente view guarda e gerencia telas inteiras, ou seja, o conteudo do corpo da pagina. Comando para criar uma view: ng g c views/home ou qualquer outro nome.

Routes cria link entre componentes.

A ligação de eventos e componentes, tambem chamado bindings , é atraves do arquivo TS. O binding para atributos usa o '[ ]', ja para evento é '( )'.

quanto um atributo dor opcional é usado o '?'. ex: id?:

Pesquisar o que é o singleton

Para criar um http client é necessario importar o HttpClientModule, no arquivo app.module.ts

toway db utiliza o '[( )]' para referenciar um elemento escrito na pagina para o banco de dados

criar componentes simples: https://material.angular.io/guide/schematics

Decorator

TAREFA: fazer alterações na tabela e página