

Atividade: Desenvolver um CRUD de uma lista de compras.

Os passos a seguir é uma sugestão do projeto

Passo 1: Criação do Projeto em TypeScript com MongoDB

- Crie uma pasta para o projeto;
- Inicialize o projeto com npm e instale dependências;

```
npm init -y
```

Inicializa um novo projeto Node.js: Cria um arquivo package.json na raiz do seu projeto. Esse arquivo armazena metadados sobre o projeto, como o nome, versão, descrição e, mais importante, as dependências. O -y é um flag que responde "sim" automaticamente a todas as perguntas, criando um package.json básico.

```
npm install express mongoose cors body-parser
```

Instala as dependências principais:

- **express:** Um framework minimalista e flexível para criar aplicações web em Node.js. Ele oferece funcionalidades como roteamento, middleware e gerenciamento de requisições HTTP.
- **mongoose:** Um ORM (Object-Relational Mapper) para MongoDB. Ele permite que você interaja com o banco de dados MongoDB usando objetos JavaScript, facilitando a criação, leitura, atualização e deleção de dados.
- **cors:** Permite que aplicações feitas em diferentes domínios se comuniquem entre si, resolvendo problemas de CORS (Cross-Origin Resource Sharing). Isso é essencial para aplicações que envolvem front-ends e back-ends separados.
- **body-parser:** Analisa o corpo das requisições HTTP, extraindo dados de formatos como JSON e URL-encoded. É fundamental para receber dados enviados por formulários HTML ou APIs REST.

```
npm install --save-dev typescript ts-node @types/cors @types/express @types/node @types/mongoose
```

Instala as dependências de desenvolvimento:

- **@types/cors, @types/express, @types/node, @types/mongoose:** São arquivos de definição de tipos (type definitions) para TypeScript. Eles fornecem informações sobre as APIs dessas bibliotecas para o TypeScript, permitindo que o compilador verifique os tipos e ofereça autocompletar e outras funcionalidades de desenvolvimento.
- **typescript:** Um superconjunto tipado do JavaScript. Ele adiciona tipos estáticos ao JavaScript, ajudando a prevenir erros e melhorar a qualidade do código.

- **ts-node**: Uma ferramenta que permite executar arquivos TypeScript diretamente, sem a necessidade de compilar para JavaScript antes.

- Crie o arquivo `tsconfig.json`::

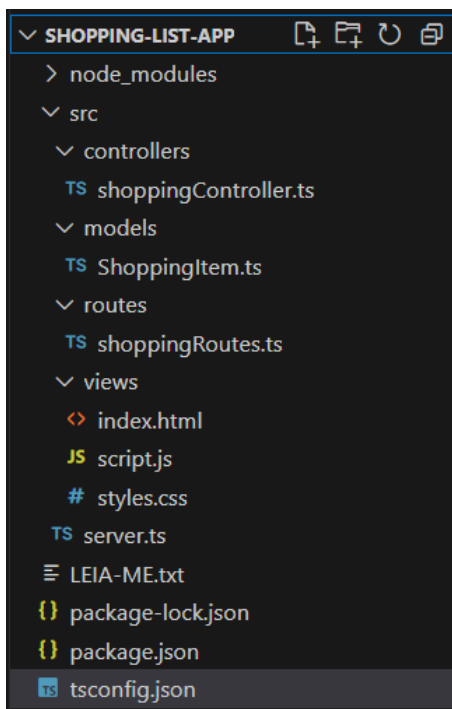
```
npx tsc -init
```

Isso gerará um arquivo `tsconfig.json`. Certifique-se de que as opções abaixo estejam habilitadas no arquivo:

```
"target": "es6",  
"moduleResolution": "node",  
"outDir": "./dist",  
"rootDir": "./src",  
"esModuleInterop": true,  
"strict": true,
```

Passo 2: Estrutura de Diretórios

Crie a seguinte estrutura de diretórios e arquivos:



Este é uma sugestão de projeto.

Você deve criar os controllers, os models, a rota, as views, além do server.ts.

Passo 3 – Criando o Banco de Dados no MongoDB Compass

Crie o banco de dados no MongoDB

Sugestão:

- Nome do banco de dados: shopping-list
- Nome da coleção: shoppingitems

Passo 4 – Rodar o Projeto

Inicie o servidor com o comando:

```
npx ts-node src/server.ts:
```

O comando `npx ts-node src/server.ts` executa o arquivo `server.ts` (que está localizado na pasta `src`) usando o TypeScript diretamente, sem a necessidade de uma etapa de compilação prévia. Isso é muito útil durante o desenvolvimento, pois permite que você veja as alterações no seu código em tempo real.

Lista de Compras

Nome do produto	Valor do produto	Adicionar Produto
Bala - R\$ 9.90		<button>Editar</button> <button>Excluir</button>
Chocolate Milka - R\$ 15.00		<button>Editar</button> <button>Excluir</button>
Finni - R\$ 5.00		<button>Editar</button> <button>Excluir</button>
Coca-Cola - R\$ 5.50		<button>Editar</button> <button>Excluir</button>