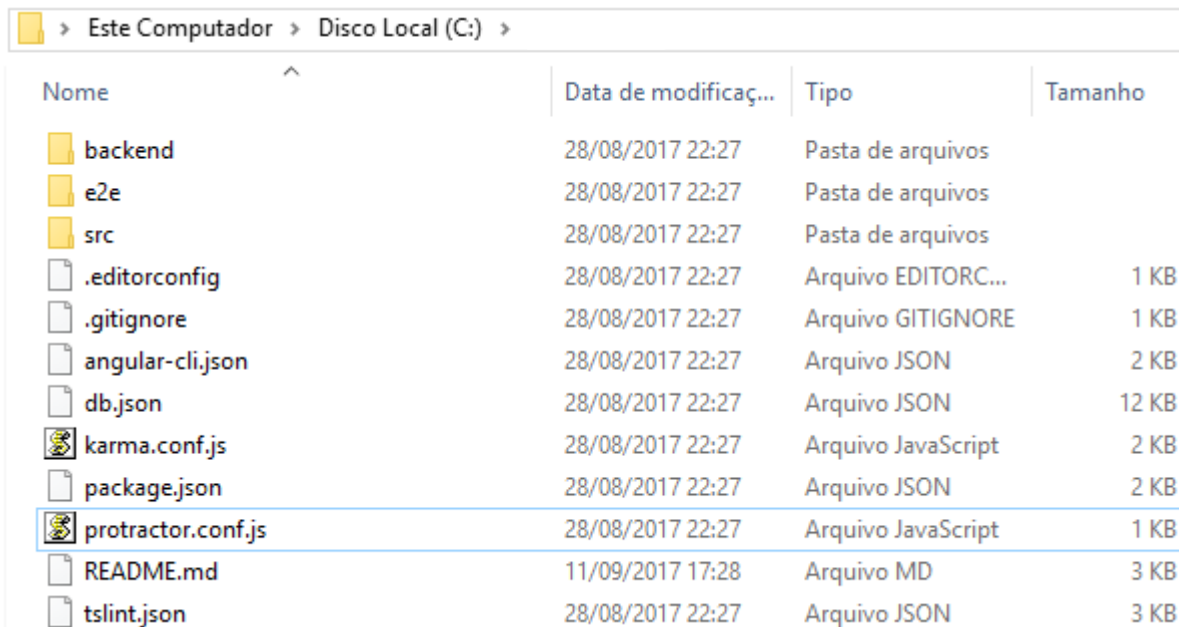


**LEIA COM ATENÇÃO AS INSTRUÇÕES ABAIXO**

Primeiramente você deve baixar o projeto em:

**<https://github.com/ProfessorTolentino/Cantina-FUMEC>**

Coloque o conteúdo desse arquivo dentro de uma pasta chamado “CantinaFUMEC”.



Nome	Data de modificaç...	Tipo	Tamanho
backend	28/08/2017 22:27	Pasta de arquivos	
e2e	28/08/2017 22:27	Pasta de arquivos	
src	28/08/2017 22:27	Pasta de arquivos	
.editorconfig	28/08/2017 22:27	Arquivo EDITORC...	1 KB
.gitignore	28/08/2017 22:27	Arquivo GITIGNORE	1 KB
angular-cli.json	28/08/2017 22:27	Arquivo JSON	2 KB
db.json	28/08/2017 22:27	Arquivo JSON	12 KB
karma.conf.js	28/08/2017 22:27	Arquivo JavaScript	2 KB
package.json	28/08/2017 22:27	Arquivo JSON	2 KB
protractor.conf.js	28/08/2017 22:27	Arquivo JavaScript	1 KB
README.md	11/09/2017 17:28	Arquivo MD	3 KB
tslint.json	28/08/2017 22:27	Arquivo JSON	3 KB

Entre na pasta com o prompt do node e execute a instalação dos pacotes com:

**npm install**

Você deve criar uma aplicação e-commerce contendo dashboard, e cadastros de categorias, produtos, usuários, pedidos realizados.

A aplicação consiste em listar dados do JSON-Server e cadastrar novos. O tema utilizado é <https://adminlte.io/themes/dev/AdminLTE/> você pode utilizá-lo para pegar peças html para a aplicação.

O arquivo db.json é o arquivo que contém os dados do json para manipular no JSON-Server, ele está na raiz do projeto. Se ainda não tiver conhecimento sobre JSON-Server, pode aprender em: [https://www.youtube.com/playlist?list=PLxlgFKtP\\_VyhoFh50mQDDMJWDNFatGIYn](https://www.youtube.com/playlist?list=PLxlgFKtP_VyhoFh50mQDDMJWDNFatGIYn)

A aplicação deve conter as seguintes funcionalidades.

**Tela gestão de usuários:**

- Listar usuários
- Cadastrar usuários
- Alterar usuários
- Excluir usuários

**Tela gestão de Categorias:**

- Listar categorias
- Cadastrar categorias
- Alterar categorias
- Excluir categorias

#### **Tela gestão de Produtos:**

- Listar produtos
- Cadastrar produtos
- Alterar produtos
- Excluir produtos
- Permitir incluir o produto no carrinho de compras.

#### **Tela carrinho de compras:**

- Permitir pesquisar e incluir um produto no carrinho de compras via id ou nome no mesmo campo.
- Permitir alterar quantidade de um produto.
- Exibir valor unitário do produto.
- Exibir valor total do produto. (Produto \* quantidade).
- Exibir valor total da compra.
- Permitir finalizar uma compra.

#### **Tela histórico de compras:**

- Listar todas as compras realizadas
- Permitir filtrar as compras pela data.

#### **Home page:**

- Incluir um Card que informe a quantidade total de Usuários cadastrados
- Incluir um Card que informe o valor total das vendas realizadas no dia atual.
- Incluir um Gráfico que informe o valor total das vendas diárias do mês no mês atual.

#### **1) Regras básicas de validação**

##### **Usuários:**

- Realizar a validação dos campos obrigatórios.
- Realizar a validação do CPF do usuário.
- Não permitir a inclusão de e-mails iguais.

##### **Categorias:**

- Realizar a validação dos campos obrigatórios.
- Não permitir categorias de nome repetido.

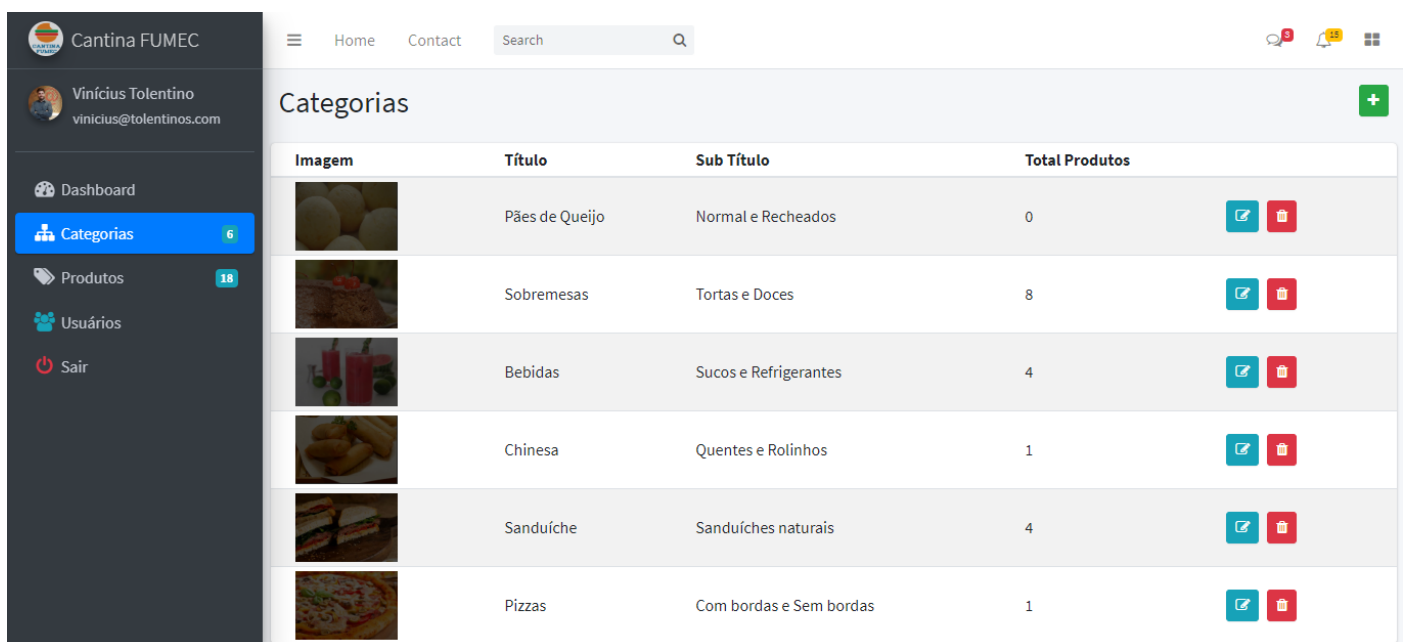
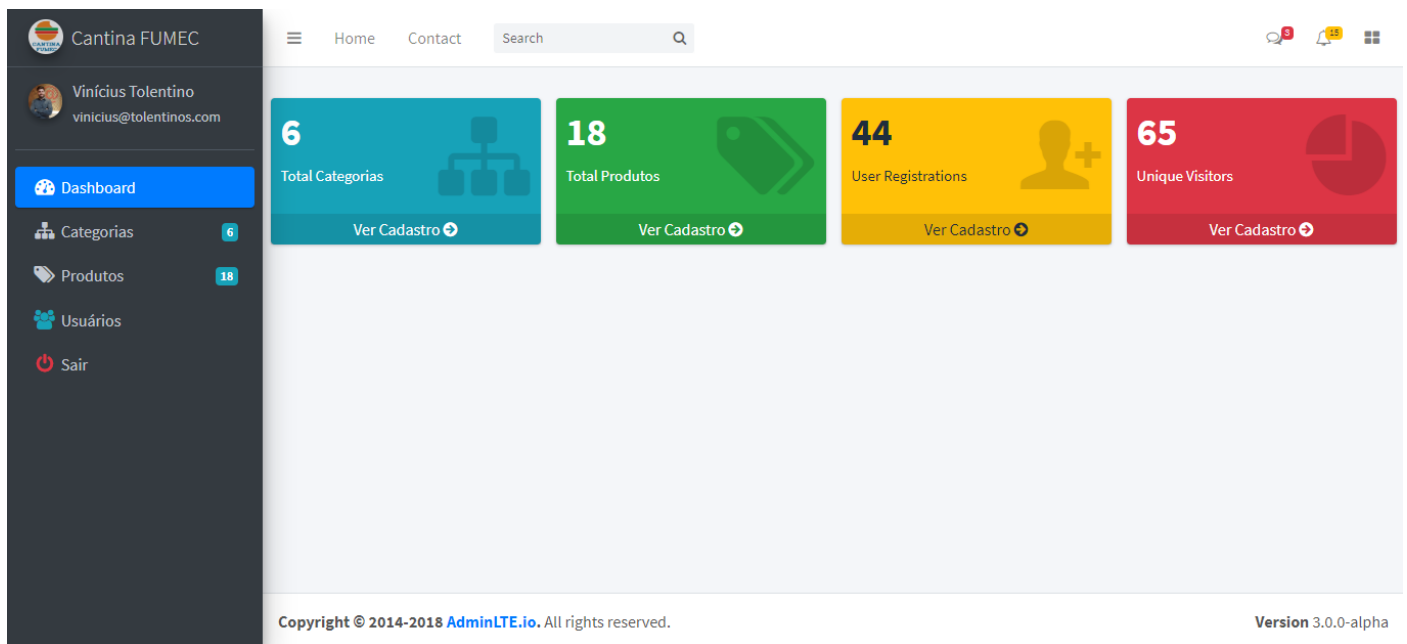
##### **Produtos:**

- Realizar a validação dos campos obrigatórios.
- O valor do produto deve ser maior que 0.
- Não permitir produtos de nome repetido.
- Deve ser gerado um código de barras para cada produto de acordo com o seu Id.

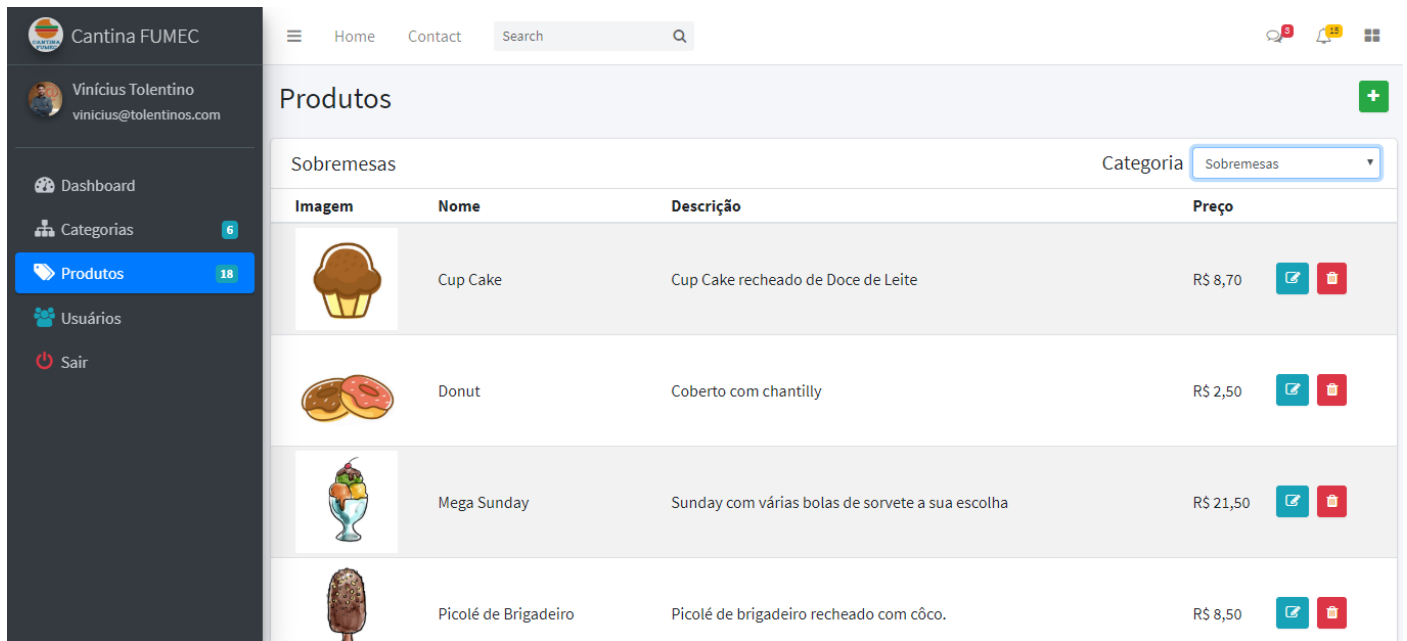
#### **Tarefas extras:**

- Incluir no Dashboard uma lista que informe os 10 produtos mais comprados e permitir incluir este produto no carrinho atual.
- Incluir uma lista que informe as 10 compras mais realizadas e permita incluir os itens desta compra no carrinho de compra atual.

A aplicação deve seguir o seguinte layout.



Deve seleccionar a categoria primeiro para poder listar os produtos.



Deve se seguir a seguinte estrutura dos dados, não é permitido alterar os campos para estrutura diferente dessas apresentadas abaixo.

```
Category {
    id: number;
    cardSubtitle: string;
    cardTitle: string;
    imagem: string;
    menu: Product[];
}
```

```
Product {
  id: string;
  imagePath: string;
  name: string;
  description: string;
  price: number;
}
```

```
User {
  uid: string;
  id: number;
  name: string;
  email: string;
  cpf:string;
  phoneNumber: string;
  password:string;
  photoUrl: string;
  location: string;
  points: number;
  settingsUser: { language: string, type: string };
}
```

```
CartItem {  
  constructor(public menuItem: Product,  
    public quantity: number = 1) { }  
  
  value(): number {  
    return this.menuItem.price * this.quantity  
  }  
}
```

```
Order {  
  id:string,  
  userEmail:string,  
  date: string,  
  value: number,  
  items: CartItem[]  
}
```

Sinta-se livre para realizar outras validações ou melhorias que acredite serem uteis para o bom funcionamento da aplicação.