

Bootcamp Desenvolvedor Tech PAN

Desafio

Módulo 2	Angular
----------	---------

Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Criar componentes, *templates* e pipes.
- ✓ Comunicar com o *back end*.
- ✓ Definir roteamento.
- ✓ Usar serviços.
- ✓ Usar a biblioteca Angular Material.

Enunciado

Construa, utilizando Angular, uma aplicação para pedido de comida, que vamos chamar de *ngFood*. Tal aplicação deve ter duas telas (cada uma com sua rota).

A tela inicial, **Cardápio**, exibe a lista de produtos e seus respectivos preços. Ao clicar em um produto, ele deve ser adicionado ao pedido atual. Um produto pode ser adicionado repetidas vezes ao pedido, aumentando a quantidade do mesmo. Esta tela também deve:

- Mostrar a quantidade de itens adicionados e o valor total do pedido atual.
- Ter um link para a tela **Pedido**.
- Exibir a categoria acima do produto quando ele não for da mesma categoria do anterior na lista (produtos da mesma categoria estão em sequência).

A tela **Pedido** exibe cada produto adicionado ao pedido atual e a respectiva quantidade, bem como o valor total do pedido. Esta tela deve ter botões para:

- Voltar para a tela **Cardápio**.

- Limpar o pedido (excluir todos os produtos).
- Concluir o pedido (não faz nada, apenas exiba o botão).

Cardápio		
	VER PEDIDO	R\$ 0,00
SANDUICHES DE BOI		
Hamburguer		R\$ 13,90
X - Burger		R\$ 15,20
X - Egg Burger		R\$ 15,80
X - Bacon		R\$ 16,20
X - Egg Bacon		R\$ 16,70
Laçador		R\$ 16,70
X - Tudo		R\$ 17,70
SANDUICHES DE FRANGO		
Frango Light		R\$ 18,30

Tela **Cardápio**

Cardápio		
	VER PEDIDO	R\$ 73,70
Tilápia c/ fritas		
		R\$ 59,90
Contra Filé c/ fritas		
		R\$ 69,90
Picanha c/ fritas		
		R\$ 79,90
CREME DE AÇAÍ		
Creme de açaí (300g)		
		R\$ 13,90
Creme de açaí (500g)		
		R\$ 16,90
Creme de açaí (1kg)		
		R\$ 29,90
SUCOS		
Suco de laranja natural (300 ml)		
		R\$ 6,50
Jarra de suco de laranja natural (700 ml)		
		R\$ 12,90

Tela **Cardápio** após adicionar produtos

Pedido		
Hamburguer	1 × R\$ 13,90	
X - Tudo	2 × R\$ 17,70	
Refrigerante Coca-Cola (2 litros)	1 × R\$ 10,50	
Creme de açaí (300g)	1 × R\$ 13,90	
Total		R\$ 73,70
<div> <div>Incluir mais itens</div> <div>Limpar</div> <div>Concluir</div> </div>		

Pedido		
Seu pedido está vazio.		
Total		R\$ 0,00
<div> <div>Incluir mais itens</div> <div>Limpar</div> <div>Concluir</div> </div>		

*Tela **Pedido****Tela **Pedido** após limpar*

Acima temos exemplos da interface da aplicação em diferentes situações. Sua aplicação não precisa ter exatamente a mesma aparência do exemplo dado, basta ter o comportamento especificado.

No entanto, você deve cumprir às seguintes exigências:

- A lista de produtos deve ser obtida por meio de uma requisição HTTP a um *back end* fictício, baseado em *json-server*. Execute-o por meio do comando `json-server --watch cardapio.json`. O arquivo `cardapio.json` será fornecido pelo professor no fórum de avisos. Note que o *endpoint* `/cardapio` retorna um *array* onde cada objeto possui os campos `categoria`, `descricao` e `preco`.
- Use ao menos um componente da biblioteca Angular Material.
- Use um serviço para armazenar e gerenciar os dados do pedido atual (que são globais).

Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Crie o projeto e adicione a dependência para Angular Material.
2. Configure o módulo principal da aplicação, adicionando *imports* para os módulos necessários, como HttpClientModule, módulos do Angular Material etc.
3. Crie os componentes para as duas telas.
4. Configure rotas para cada tela.
5. Implemente o carregamento de dados (lembre-se de executar o *back end*).
6. Implemente as funcionalidades descritas.

Dicas

- Preocupe-se primeiro com a funcionalidade, deixe a aparência para o final.
- Nas interfaces demonstradas, eu utilizei os seguintes componentes do Angular Material: Toolbar, List, Button, Badge e Icon.
- Defina uma interface Typescript descrevendo o formato dos objetos retornados pelo *back end*.
- O json-server pode ser instalado com o comando `npm install -g json-server`. Para executar, use o comando `json-server --watch nomeDoArquivo.json`.