

## Bootcamp Desenvolvedor Tech PAN

### Trabalho Prático

Módulo 2	Angular
----------	---------

#### Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Criar uma aplicação Angular.
- ✓ Criar componentes e *templates*.
- ✓ Utilizar *data/event bindings*.
- ✓ Utilizar diretivas estruturais.
- ✓ Estilizar componentes.

#### Enunciado

Utilizando o Angular, construa um componente que permita a seleção de opções de uma lista predefinida. Tal componente deve ser configurável e receber como propriedades:

- Um título, exibido no topo do componente (*string*).
- A lista de opções disponíveis, na forma de um *array* de *strings*.
- A quantidade máxima de escolhas permitidas (*number*).

A seguir temos um exemplo de uso do componente, assumindo que seu seletor é **app-selecao**, em uma aplicação hipotética de pedido de comida.

```
<h2>Monte seu açaí</h2>
<app-selecao titulo="Tamanho"
  [opcoes]="['300 ml', '500 ml']"
  [escolhaAte]="1">
</app-selecao>
<app-selecao titulo="Acréscimos"
  [opcoes]="['Leite condensado', 'Banana', 'Granola', 'Morango',
    'Leite em pó', 'Côco', 'Castanha de caju']"
  [escolhaAte]="3">
</app-selecao>
```

Com o exemplo acima, espera-se que a aplicação resultante possua interface semelhante a exibida abaixo:

Monte seu açaí

Tamanho

☐ 300 ml
 ☐ 500 ml

Acréscimos

Escolha até 3

☐ Leite condensado
 ☐ Banana
 ☐ Granola
 ☐ Morango
 ☐ Leite em pó
 ☐ Côco
 ☐ Castanha de caju

A) Interface no estado inicial.

Monte seu açaí

Tamanho

☐ 300 ml
 ☒ 500 ml

Acréscimos

Escolha até 3 (1 selecionado)

☐ Leite condensado
 ☐ Banana
 ☐ Granola
 ☒ Morango
 ☐ Leite em pó
 ☐ Côco
 ☐ Castanha de caju

B) Interface após selecionar uma opção.

Monte seu açaí

Tamanho

☐ 300 ml
 ☒ 500 ml

Acréscimos

Escolha até 3 (3 selecionados)

☒ Leite condensado
 ☐ Banana
 ☐ Granola
 ☒ Morango
 ☒ Leite em pó
 ☐ Côco
 ☐ Castanha de caju

C) Interface após selecionar todas as opções permitidas.

Seu componente não precisa ter exatamente a mesma aparência do exemplo dado, mas precisa atender aos seguintes requisitos:

1. Quando é permitida apenas uma escolha, devem ser utilizados elementos *input* do tipo *radio* nas opções.
2. Quando são permitidas duas ou mais escolhas, devem ser utilizados elementos *input* do tipo *checkbox* nas opções. Além disso, devem ser exibidos abaixo do título o número máximo de escolhas e a quantidade de escolhas já feitas, por exemplo: “*escolha até 3 (2 selecionadas)*”.
3. Quando são permitidas múltiplas escolhas e já foram feitas todas as escolhas permitidas, as demais opções devem ficar desabilitadas (use a propriedade *disabled* do elemento *input*), conforme figura C.

## Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Crie uma aplicação Angular com o nome desejado.
2. Adicione na aplicação um componente `SelecaoComponent`.

3. Adicione as *input properties* necessárias no `SelecaoComponent`, conforme especificado.
4. Você precisará de alguma propriedade no seu componente para armazenar o estado da interface, ou seja, as opções selecionadas. Uma opção é usar um *array* de *strings*.
5. Implemente os comportamentos exigidos no seu componente conforme especificado, lembrando de tratar tanto o caso de seleção única quando seleção múltipla.
6. Use o componente `SelecaoComponent` ao menos duas vezes no componente principal. Inspire-se no exemplo de código dado.
7. Estilize a aplicação da maneira que desejar, mas defina pelo menos uma regra no CSS global (arquivo `styles.css`) e uma regra no CSS específico de seu componente.

### Dicas

- Comece com um HTML estático no *template* do seu componente e vá adicionando comportamentos dinâmicos aos poucos, testando a cada etapa.
- Você vai precisar do *decorator* `@Input` para declarar as propriedades que o componente deve receber.
- Elementos *input* do tipo *checkbox* ou *radio* possuem a propriedade *checked*, que pode ser usado para controlar se eles estão marcados ou não. Além disso, tais elementos disparam o evento *change* quando são modificados.
- É possível que você precise utilizar funções de *arrays*, tais como `indexOf`, `push`, `splice`, dentre outras. Consulte a documentação das mesmas no MDN Web Docs em caso de dúvidas.