

Tecnologias para Mobilidade IONIC



Prof. Alessandro Aparecido Antonio

Prof. Arnaldo Martins Hidalgo Junior

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

- JSON (pronuncia-se "dienson").
- JSON é a evolução do XML, no sentido de ser uma forma muito prática de armazenar informações por meio de texto. JSON é basicamente um formato leve de troca de informações/dados, entre sistemas diferentes, sendo também muito mais fácil de ler, tanto para o ser humano quanto para a máquina.

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

Veja um exemplo de JSON:

- `{"id":1, "nome":"maria", "endereco":"R. Tal"}`

Veja abaixo como seria uma listagem de nomes e endereços:

- `{"id":1, "nome":"maria", "endereco":"R.Tal"},
{"id":2, "nome":"Pedro", "endereco":"R.XV"},
{"id":3, "nome":"Alex", "endereco":"R. Itu"}`

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

Fica mais fácil entender a estrutura de dados JSON quando tabulada, como na figura abaixo:

```
{
  "firstName": "Jonathan",
  "lastName": "Freeman",
  "loginCount": 4,
  "isWriter": true,
  "worksWith": ["Spantree Technology Group", "InfoWorld"],
  "pets": [
    {
      "name": "Lilly",
      "type": "Raccoon"
    }
  ]
}
```

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

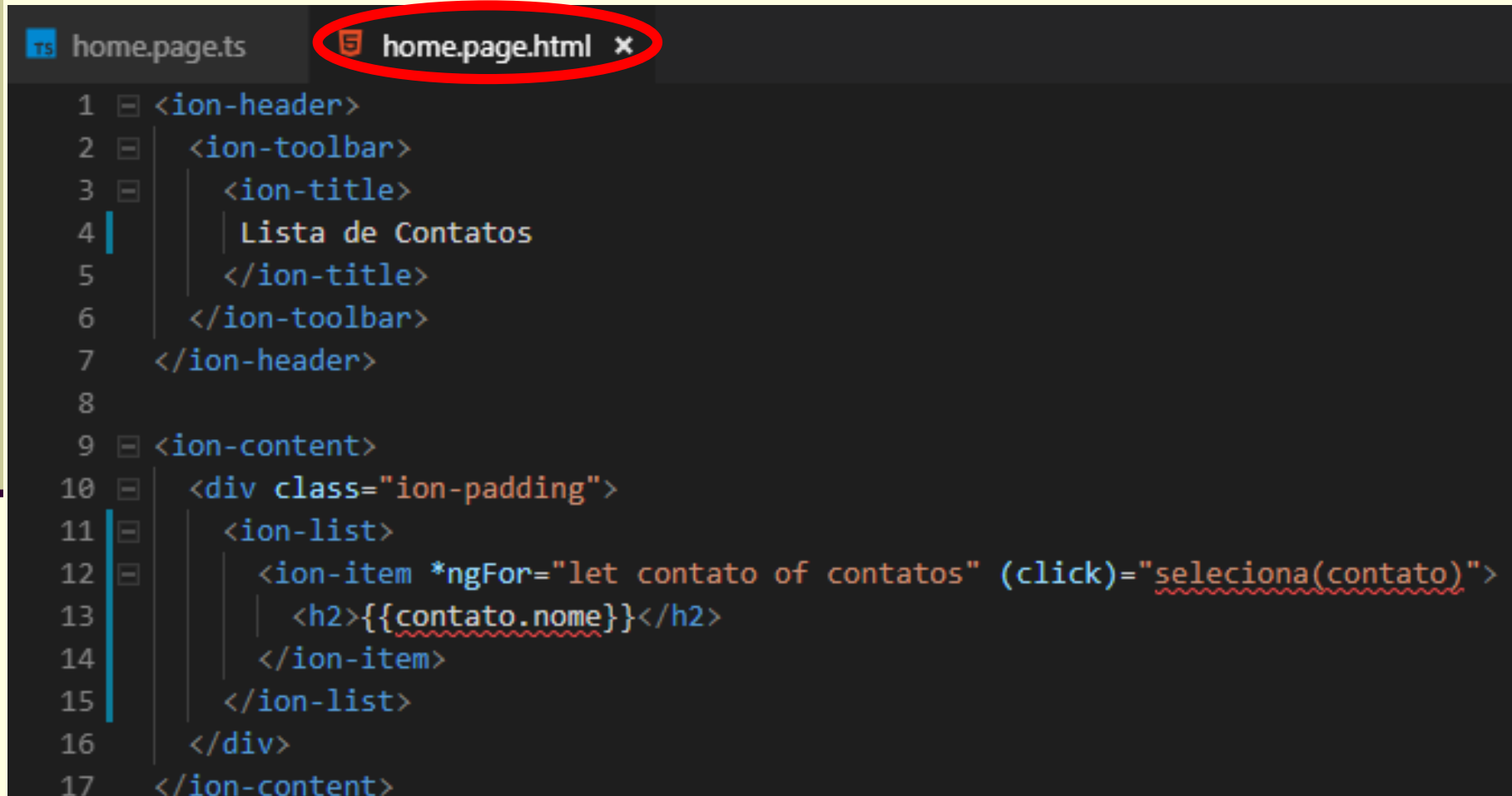
- 1) Crie a aplicação: **ionic start Etim_Contatos blank**
- 2) Em **src > app > home > home.page.ts** desenvolver o seguinte código referente a lógica da listagem:

```
8  export class HomePage {
9
10 // Definindo o vetor de contatos
11 private contatos: Array<object> = [];
12
13 constructor() {
14 // JSON com os contatos
15 this.contatos = [
16 {
17     nome: 'Ricardo',
18     telefone: '1234-5678'
19 },
```

```
20 {
21     nome: 'João',
22     telefone: '9874-5428'
23 },
24 {
25     nome: 'Maria',
26     telefone: '9823-2555'
27 },
28 {
29     nome: 'Fernanda',
30     telefone: '8457-2584'
31 },
32 {
33     nome: 'Anita',
34     telefone: '6899-2541'
35 }
36 ];
```

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

3) Agora na interface, em
src > app > home > home.page.html implemente:

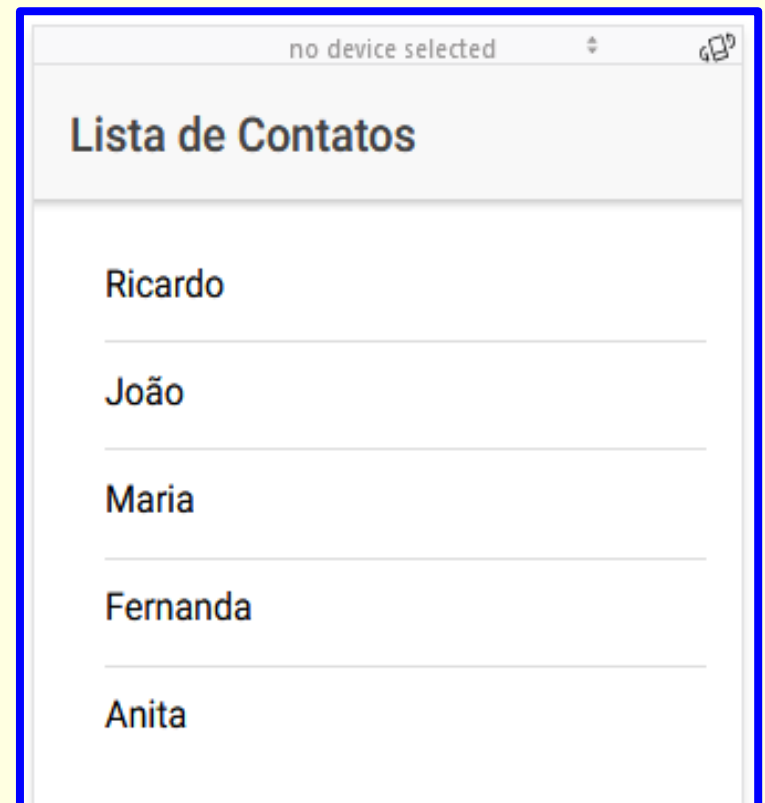


```
home.page.ts  home.page.html x
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>
4       Lista de Contatos
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content>
10  <div class="ion-padding">
11    <ion-list>
12      <ion-item *ngFor="let contato of contatos" (click)="seleciona(contato)">
13        <h2>{{contato.nome}}</h2>
14      </ion-item>
15    </ion-list>
16  </div>
17 </ion-content>
```

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS


```
<ion-list>
  <ion-item *ngFor="let contato of contatos" (click)="seleciona(contato)">
    <h2>{{contato.nome}}</h2>
  </ion-item>
</ion-list>
```

O ***ngFor** que é utilizado para realizar o loop do **array** de **contatos** onde cada iteração o objeto selecionado é retornado em uma variável local (contato). Já no título (h2) a partir do objeto contato acessamos o atributo "nome" do JSON (**contato.nome**).



JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

```
<ion-list>
  <ion-item *ngFor="let contato of contatos" (click)="seleciona(contato)">
    <h2>{{contato.nome}}</h2>
  </ion-item>
</ion-list>
```

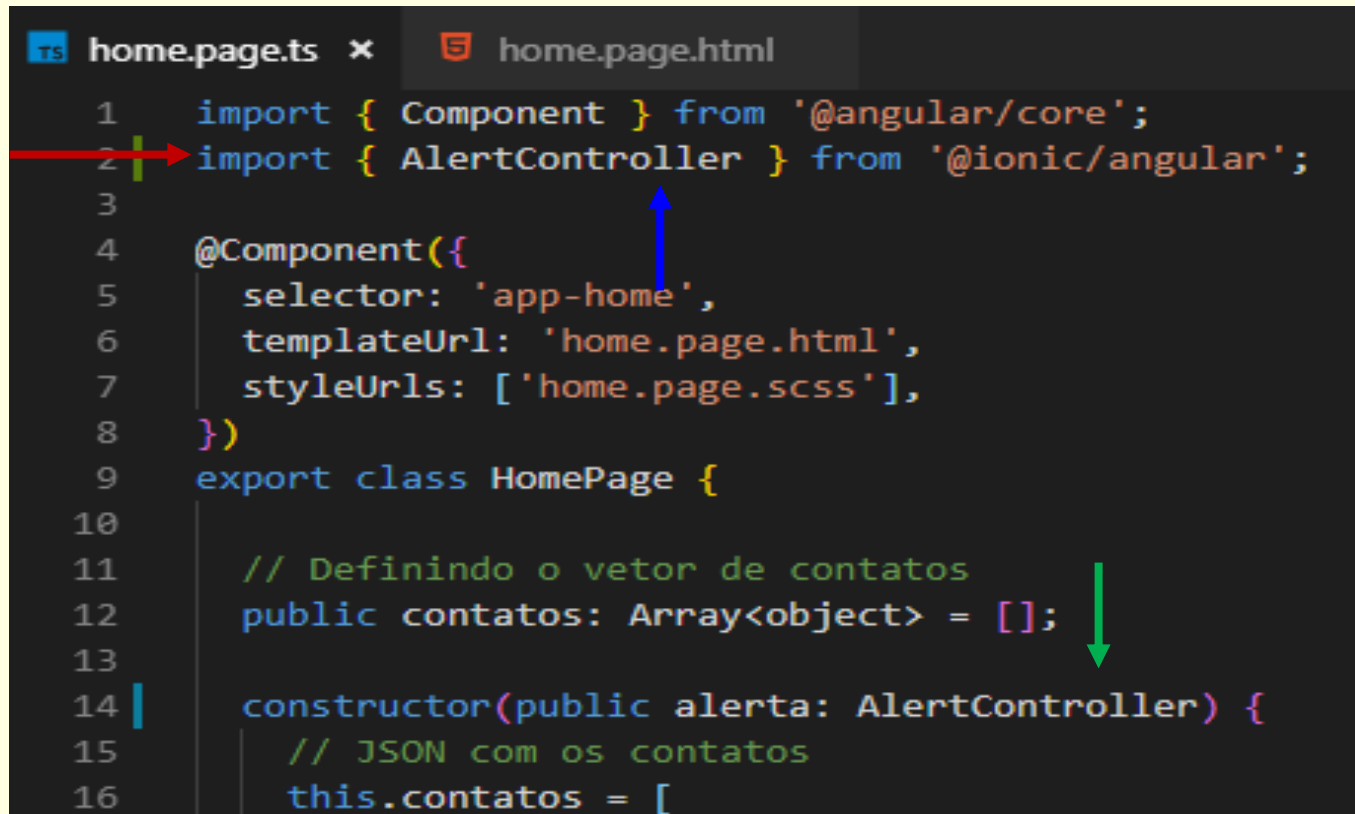


Observe no arquivo **home.html** o evento **(click)** no item, chamando a função **seleciona**, passando no parâmetro o objeto **contato**.

Agora no arquivo **home.ts** vamos implementar o método **seleciona** de modo que quando o usuário clicar no contato será exibido uma mensagem **(Alert)** contendo o número do telefone.

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

Mas antes é necessário realizar o *import* da classe **AlertController** e declarar no construtor também o objeto *alerta* para podermos utilizarmos posteriormente, veja:



```
TS home.page.ts x home.page.html
1 import { Component } from '@angular/core';
2 → import { AlertController } from '@ionic/angular';
3
4 @Component({
5   selector: 'app-home',
6   templateUrl: 'home.page.html',
7   styleUrls: ['home.page.scss'],
8 })
9 export class HomePage {
10
11   // Definindo o vetor de contatos
12   public contatos: Array<object> = [];
13
14   constructor(public alerta: AlertController) {
15     // JSON com os contatos
16     this.contatos = [
```

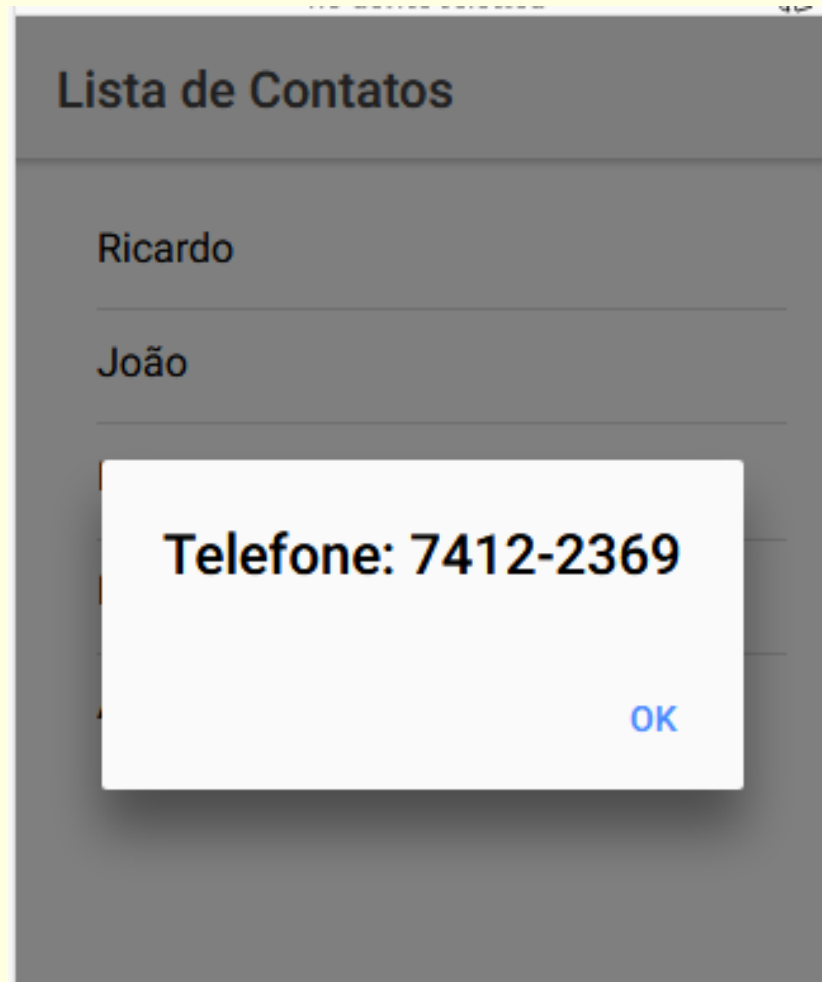
JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

4) Tudo preparado, basta criar a função:

```
41  async seleciona(contato) {  
42      const alert = await this.alerta.create({  
43          header: contato.nome,  
44          message: 'Telefone: ' + contato.telefone,  
45          buttons: ['OK']  
46      });  
47      await alert.present();  
48  }  
49
```

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

E para finalizar segue o print com o resultado final ao clicar em um contato:



JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

Se você deseja fazer uma lista, mas no estilo "dropdown" basta alterar a interface (**home.html**) como abaixo

```
9  <ion-content>
10  <div class="ion-padding">
11    <ion-item>
12      <ion-label>Contatos: </ion-label>
13      <ion-select>
14        <ion-select-option *ngFor="let contato of contatos" >{{contato.nome}}</ion-select-option>
15      </ion-select>
16    </ion-item>
17  </div>
18 </ion-content>
```

JSON - DESENVOLVENDO APLICATIVOS COM LISTAS

ATIVIDADE

- Criar um aplicativo em que o usuário possa selecionar um Estado e exiba a qual região pertence, qual o nome da Capital e o número de habitantes da Capital.