

## Ajustes no Card

### Ajustes no Cards

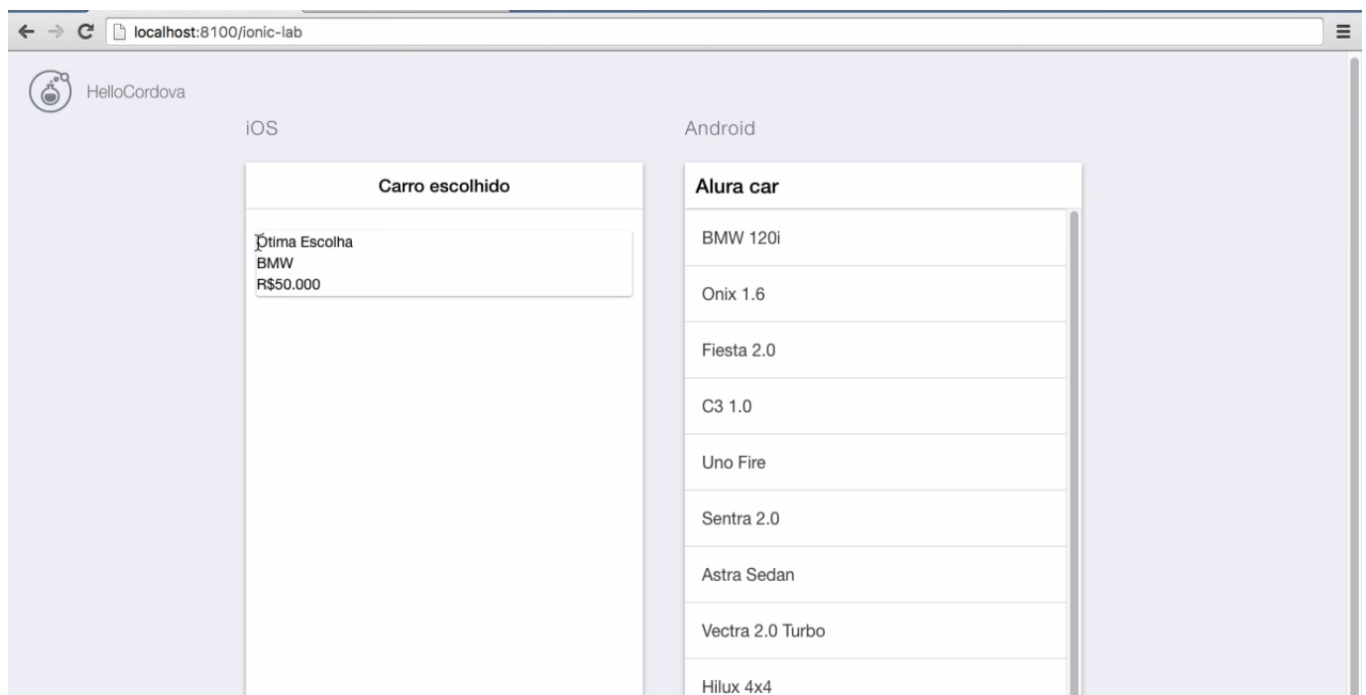
Nós já temos nossa listagem e a navegação para a segunda *View*. Temos também o componente CSS que é o card, ainda sem nenhum conteúdo. Nós iremos continuar desenvolvê-lo. Na documentação, veremos que ele terá um cabeçalho, um "miolo" preenchido com texto e o rodapé.

No arquivo `carroescolhido.html`, temos a `div` pai que é a do `card`. Iremos incluir as três `div`s filhas.

```
<ion-view>
  <ion-nav-title>Carro escolhido</ion-nav-title>
  <ion-content>
    <div class="card">
      <div>
        Ótima Escolha
      </div>
      <div>
        BMW
      </div>
      <div>
        R$50.000
      </div>
    </div>

  </ion-content>
</ion-view>
```

Vamos ver como conferir como ficou o resultado.

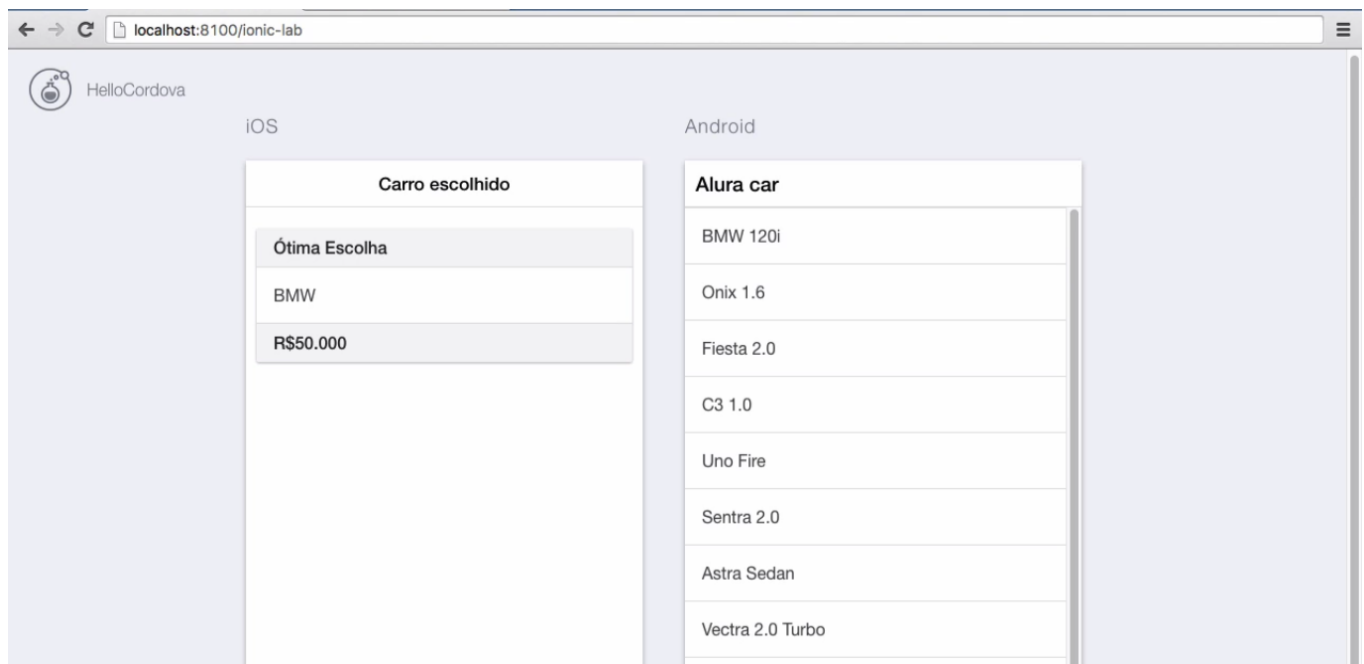


Ainda existem diferenças com o exemplo mostrado na documentação porque não adicionamos o CSS ainda. Vamos usar o CSS disponível da documentação.

```
<ion-view>
  <ion-nav-title>Carro escolhido</ion-nav-title>
  <ion-content>
    <div class="card">
      <div class="item item-divider">
        Ótima Escolha
      </div>
      <div class="item item-text-wrap">
        BMW
      </div>
      <div class="item item-divider">
        R$50.000
      </div>
    </div>

  </ion-content>
</ion-view>
```

Já veremos a diferença...



Porém, perceberemos que nesta tela, não temos a opção de voltar à lista de carros. No Ionic, temos uma tag específica que será adicionada na barra de navegação, disponível no `index.html`.

Abaixo do fechamento do `<head>`, encontraremos a tag `<ion-nav-view>`.

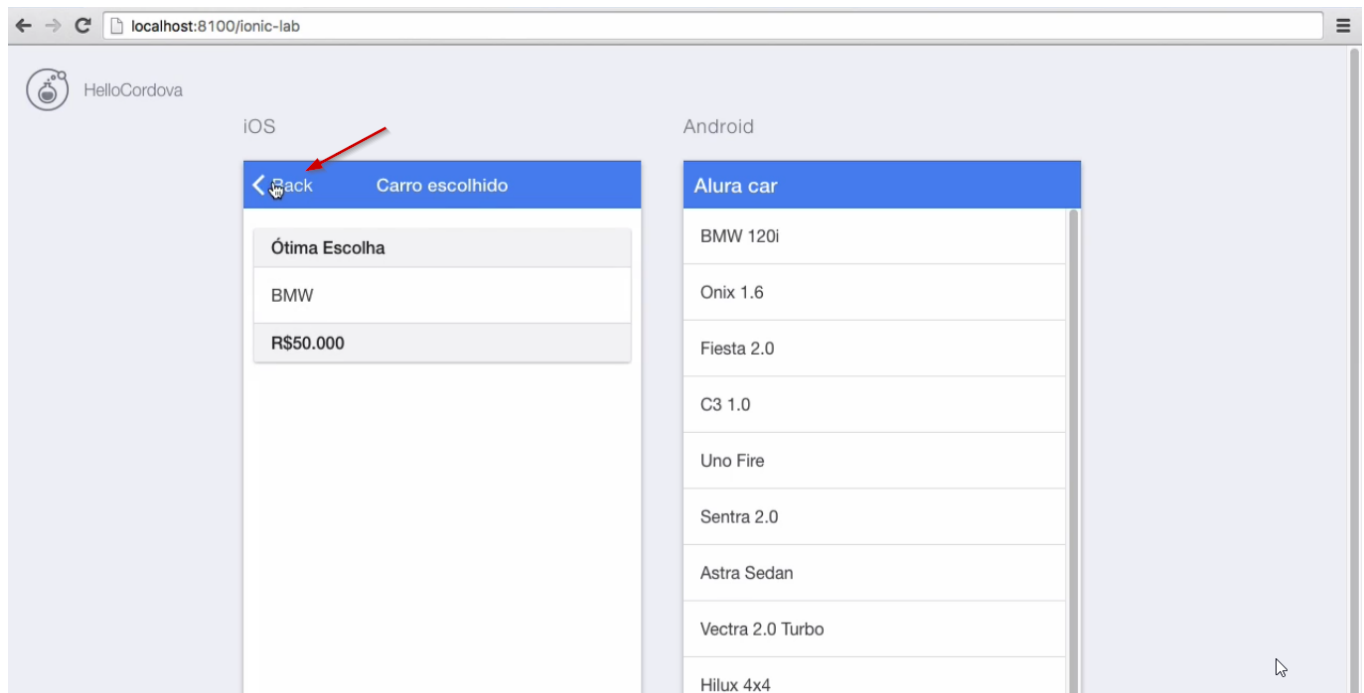
```
<body ng-app="starter" ng-controller="ListagemController">
  <ion-pane>
    <ion-nav-view>
      <ion-nav-bar>
        <ion-nav-back-button></ion-nav-back-button>
      </ion-nav-bar>
    </ion-nav-view>
```

```
//...
```

Por padrão, o botão é branco, assim como o fundo. Mas já conseguiremos retornar da tela do Carro escolhido para a lista de carros. Para destacá-lo em relação ao nome da aplicação, alteraremos a cor do título. Para fazer isto, adicionaremos um CSS no `<ion-nav-bar>`.

```
<ion-nav-bar class="bar bar-positive">
  <ion-nav-back-button></ion-nav-back-button>
</ion-nav-bar>
```

Já veremos a barra com a cor de fundo azul.



Podemos ver a palavra "Back" com destaque no iOS, enquanto vemos uma flechinha branca no Android. Nós sabemos como implementar a tag nos dois sistemas operacionais.

No entanto, se clicarmos em um item como Onix, por exemplo, irá aparecer que escolhemos a BMW. O mesmo acontecerá se clicarmos em qualquer outro carro. Se analisarmos o `carroescolhidos.html`, veremos que o texto está estático. Ele não recebe nada da página anterior e também não passa o carro selecionado no `listagem` para a tela de `carroescolhido`. Como podemos resolver isto?

No `listagem.htm`, temos um `href` que nos envia para o `#/carroescolhido/`, mas ele não é recebido no outro arquivo. Precisamos de alguma forma informar o carro que queremos e o que foi escolhido. Podemos fazer isto, adicionando uma pequena alteração no arquivo `routes.js`. Adicionaremos o parâmetro `carro` na `url`:

```
.state('carroescolhido',{
  url : '/carroescolhido/:carro',
  templateUrl: 'templates/carroescolhido.html',
  controller: 'CarroEscolhidoController'
})
```

Nós alteramos a rota, mas o link ainda não a conhece. Vamos passar o objeto `carro` como parâmetro na `href` do `listagem.html` :

```
<ion-view>
  <ion-nav-title>Alura car</ion-nav-title>
  <ion-content>
    <ion-list>
      <ion-item ng-repeat="carro in listaDeCarros" href="#/carroescolhido/{{carro}}">
        {{carro}}
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>
```

Agora, o `carroescolhido.html` continua estático. Como conseguimos pegar a variável `carro` que está sendo mandada como parâmetro? Vamos fazer alguns ajustes no arquivo `controller.js` . Sabemos que estamos trabalhando com `CarroEscolhidoController` . No curso de Angular, vimos que teremos que usar o `$stateParams` . E para manipular o escopo do Angular, iremos injetá-lo também.

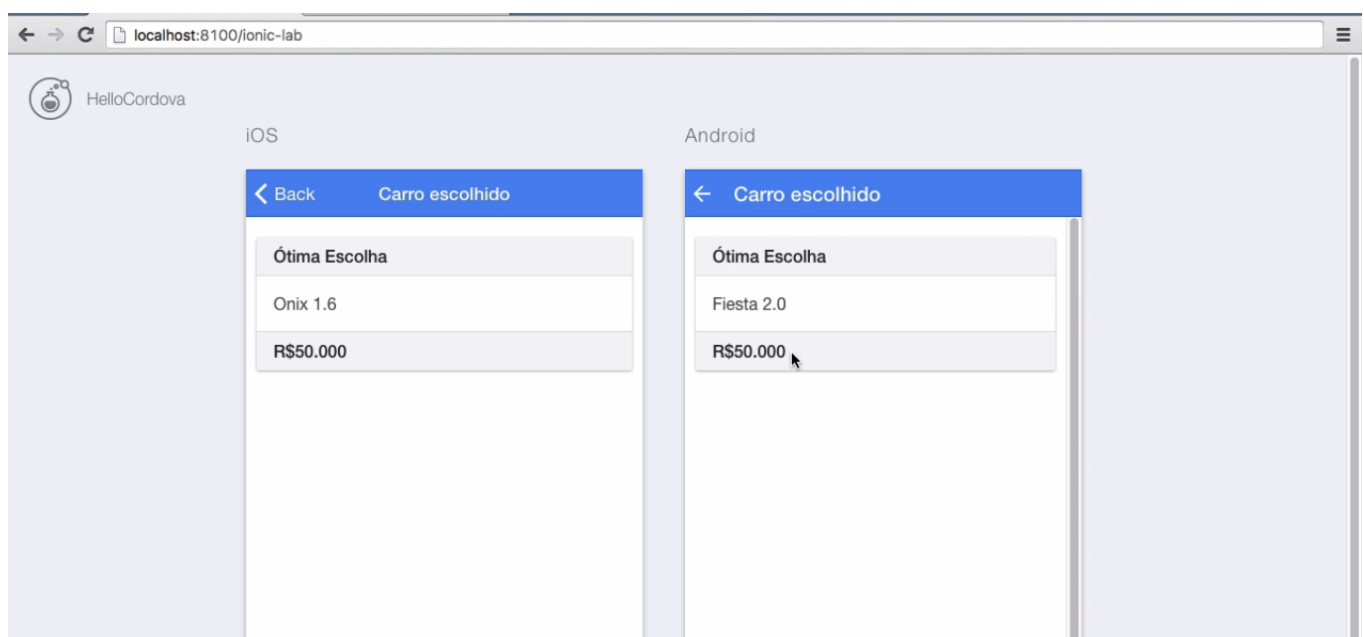
```
angular.module('starter')
.controller('CarroEscolhidoController', function($stateParams, $scope){

  $scope.carroEscolhido = $stateParams.carro;

});
```

Precisamos criar a variável `carroEscolhido` para usar na *View*. E onde tínhamos o modelo de carro estático ( `BMW` ), vamos substituir pelo `carroEscolhido` .

Se testarmos a aplicação no navegador, veremos que ao clicarmos nos modelos dos carros, a tela "Carro escolhido" terá as informações corretas.



Agora, mandamos da rota e recebemos na outra tela. A nossa navegação está funcionando muito bem.

Até a próxima!