

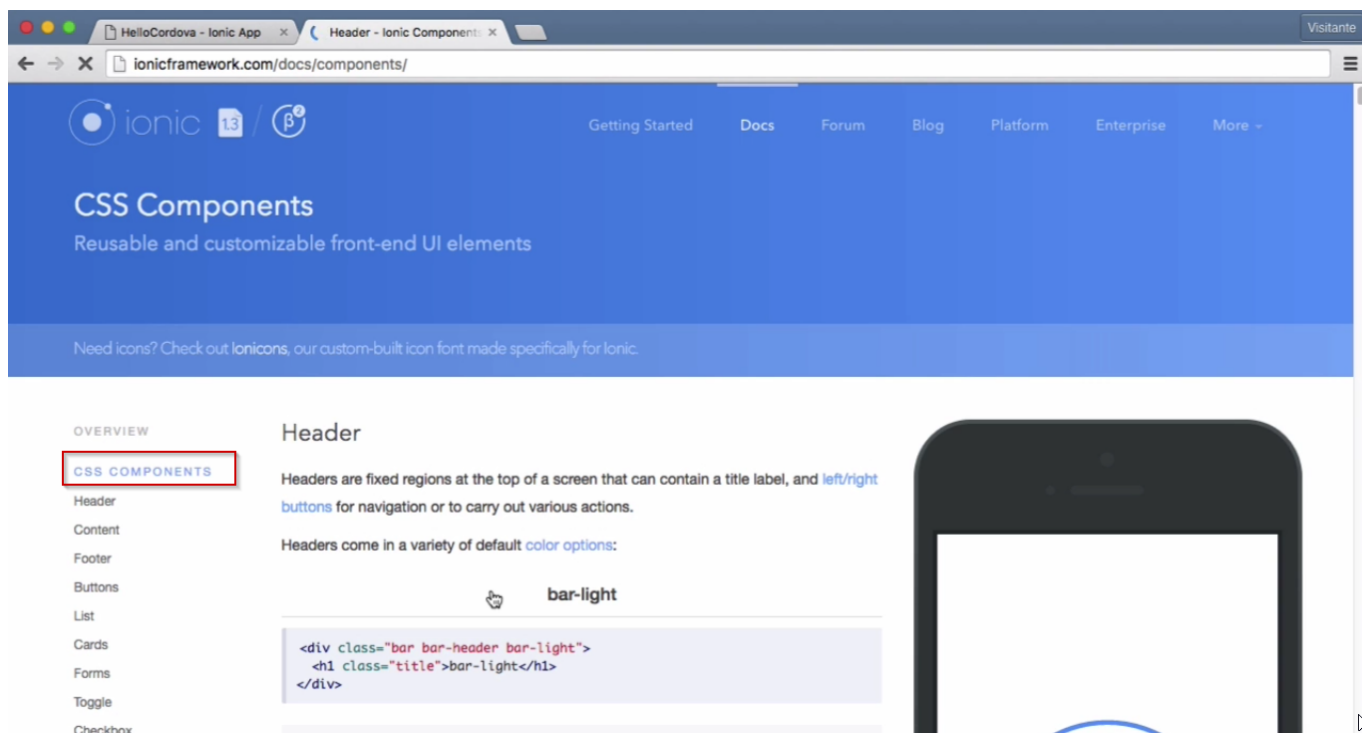
## Avançando no Aplicativo

### Avançando no aplicativo

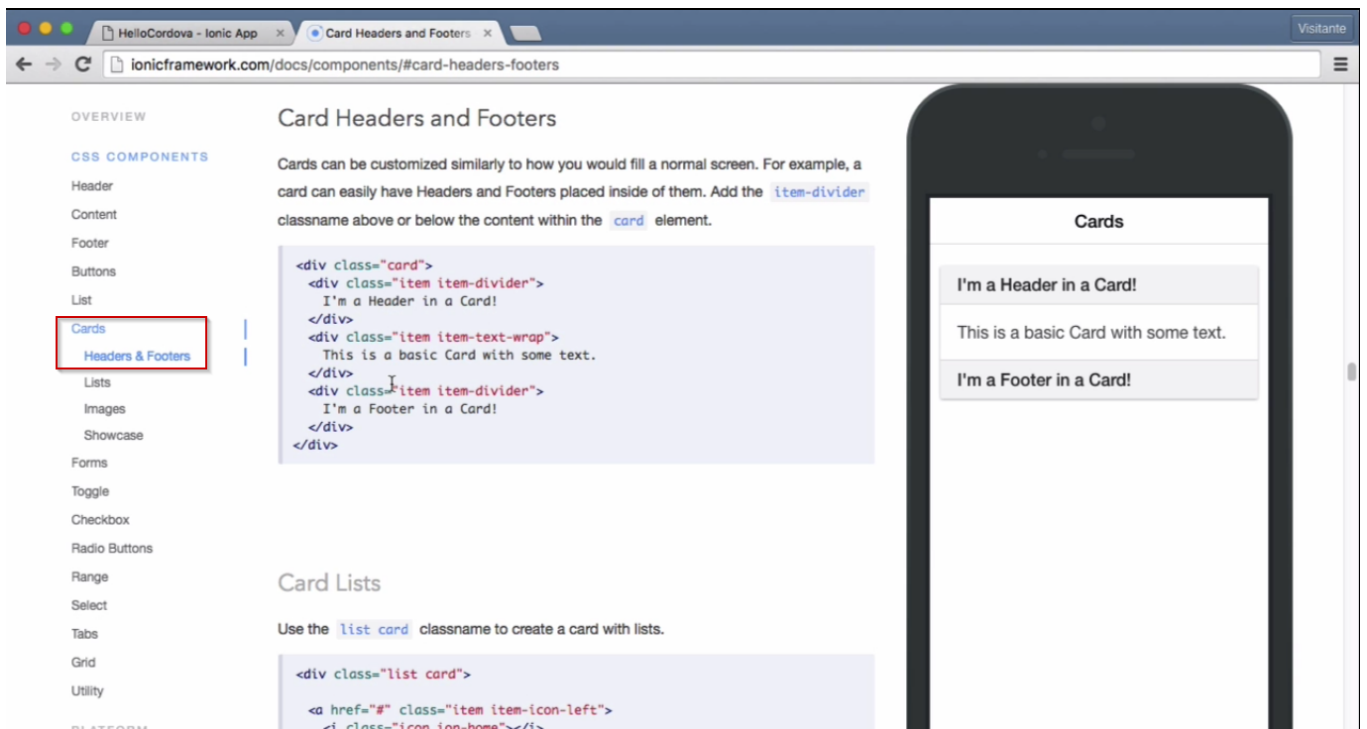
Temos a estrutura de navegação pronta, podemos avançar com o nosso aplicativo. Agora, construiremos a tela de detalhamento do pedido, na qual o usuário poderá selecionar os acessórios.

Lembre-se de que estamos guardando nossas *View* na pasta `templates`. Criaremos um novo arquivo que receberá o nome: `carroescolhido.html`.

Em seguida, vamos consultar outra parte interessante da documentação do Ionic, a seção referente aos **componentes**. Você poder acessá-la [aqui \(http://ionicframework.com/docs/components/\)](http://ionicframework.com/docs/components/).



Nós iremos clicar na opção "Cards", localizada no menu da esquerda. Nesta parte - bastante parecida a das tags -, queremos ver como funciona o `Headers` & `Footers` (cabeçalho e rodapé).



Iremos ver como podemos implementar o Card. Usaremos o cartão no nosso aplicativo. Nós já vimos que apenas copiar o código do site não funciona, logo iremos aproveitar apenas a parte que nos interessa.

De volta ao editor de texto, adicionaremos no arquivo `carroescolhido.html` a tag `<ion-view>`. Em seguida, incluiremos o título `Carro escolhido`.

```
<ion-view>
  <ion-nav-title>Carro escolhido</ion-nav-title>

</ion-view>
```

Depois, vamos inserir o conteúdo usando a tag `<ion-content>`. Começaremos a trazer a parte que pesquisamos na documentação a partir do pai, usando o `class="card"`.

```
<ion-view>
  <ion-nav-title>Carro escolhido</ion-nav-title>
  <ion-content>
    <div class="card">

    </div>

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

E para que seja possível chegarmos na tela de acessório ao clicarmos em um dos carros, teremos que criar uma nova rota no arquivo `routes.js`. Nele já encontraremos uma rota para `listagem`.

```
angular.module('starter')
.config(function($stateProvider, $urlRouterProvider){

  $urlRouterProvider.otherwise('listagem');
```

```
$stateProvider

.state('listagem'){
  url : '/listagem',
  templateUrl : 'template/listagem.html',
  controller: 'ListagemController'
});
```

Agora, criaremos para os Carros escolhidos , logo abaixo.

```
.state('carroescolhido', {
  url : '/carroescolhido',
  templateUrl: 'templates/carroescolhido.html.'

})
```

Mas não temos ainda a controller , iremos criá-la então, no arquivo controllers.js . Iremos nomeá-la como CarrosEscolhidoController . Após a ListagemController , adicionaremos o seguinte trecho.

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

});
```

Em seguida, adicionaremos a controller na rota.

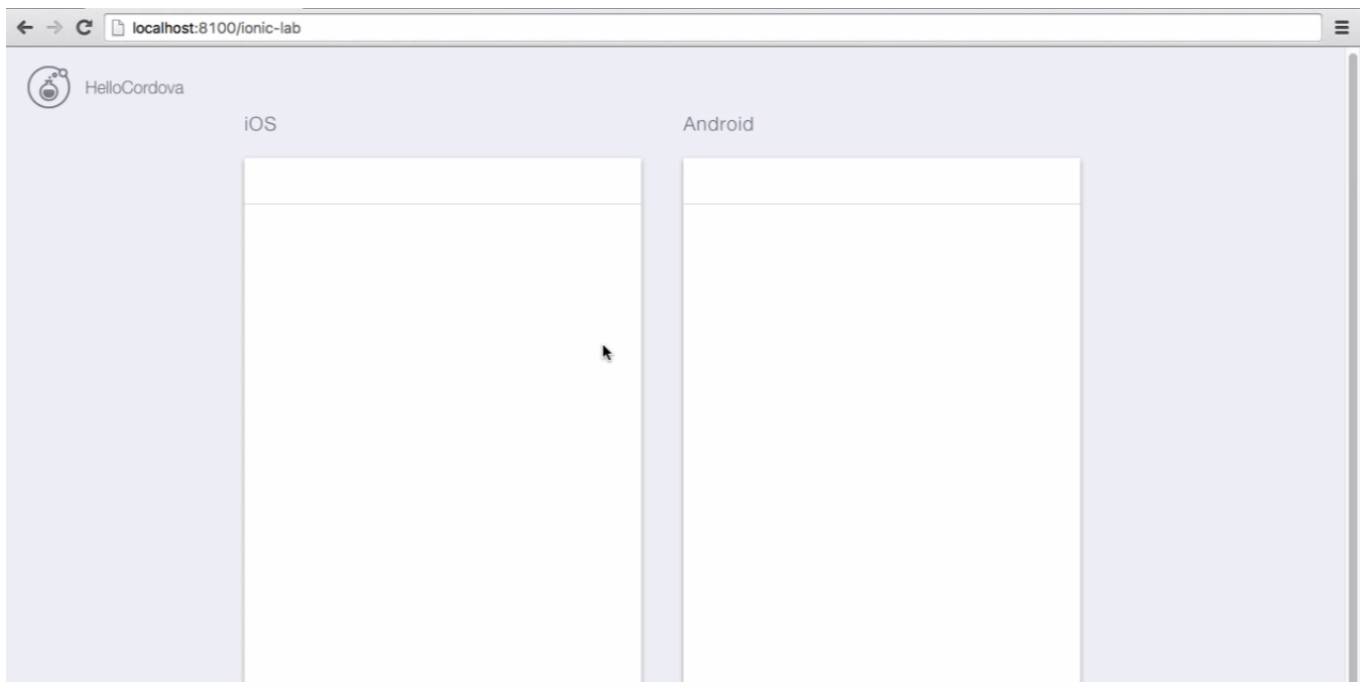
```
.state('carroescolhido', {
  url : '/carroescolhido'.
  templateUrl: 'templates/carroescolhido.html.',
  controller: 'CarroEscolhidoController'

})
```

Porém, se testarmos a aplicação no browser, veremos que não acontece nada quando clicamos nos carros. Afinal, não adicionamos nenhuma ação, nós criamos lista de itens estáticos. Nosso objetivo é que ao clicarmos nos carros, o usuário seja enviado para nossa view. Faremos isto fazendo alterações no arquivo listagem.html , adicionando href .

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

Mas se testarmos a aplicação no browser, veremos que desapareceram as listas.

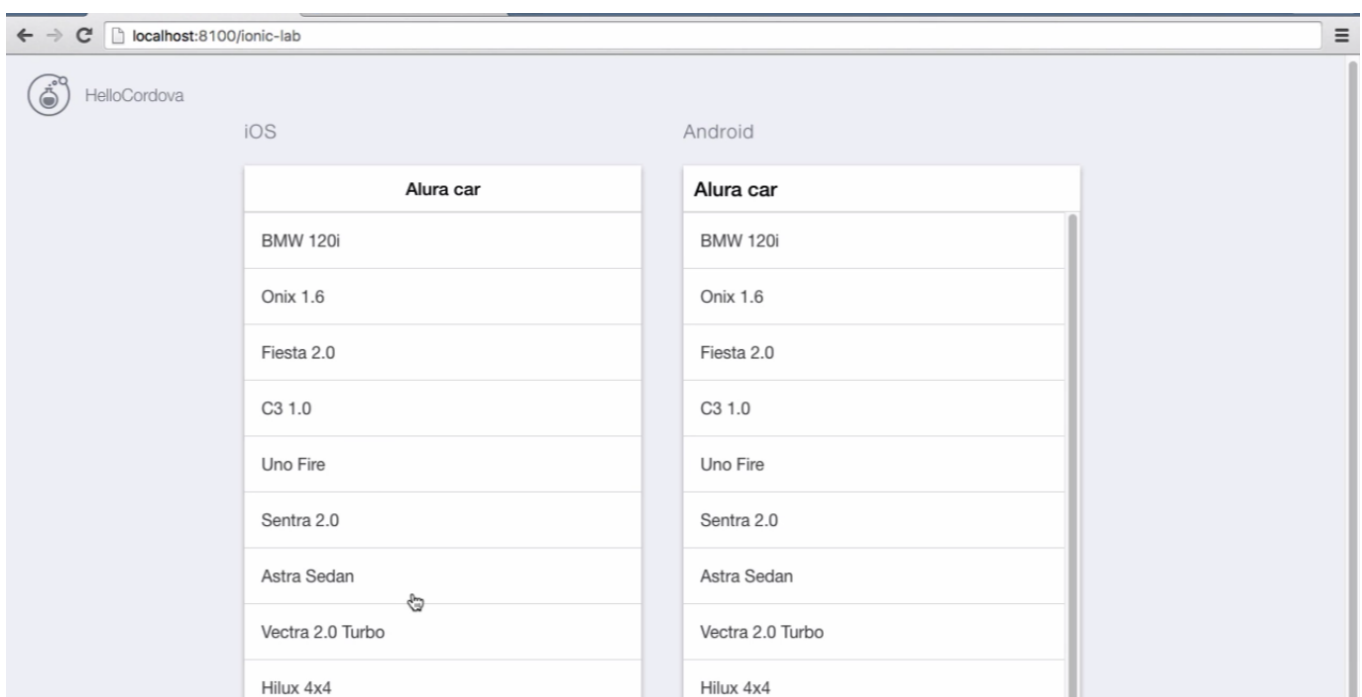


Usaremos as ferramentas de desenvolvimento do Google Chrome para entender o que está acontecendo. Na aba console, veremos que temos um erro de Syntaxe no arquivo `routes.js`. O problema ocorreu porque havíamos adiciona ; após a rota da `listagem`.

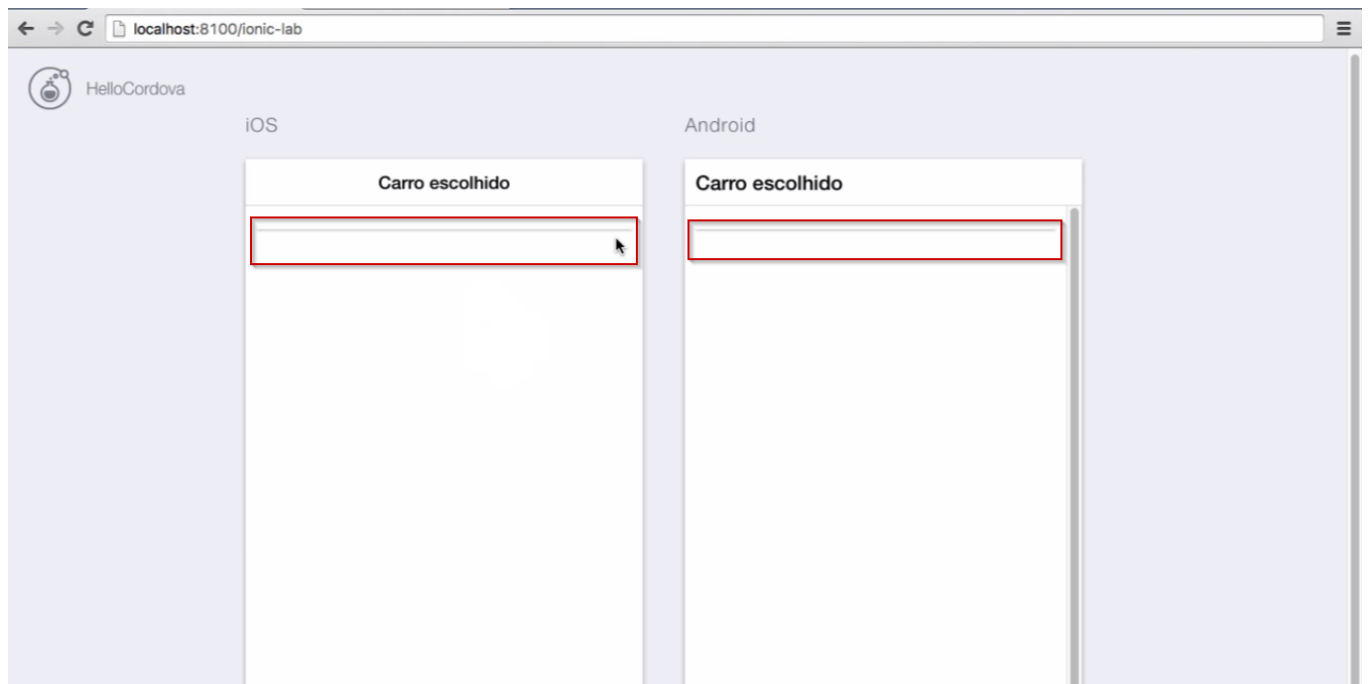
```
$stateProvider

.state('listagem'){
  url : '/listagem',
  templateUrl : 'template/listagem.html',
  controller: 'ListagemController'
});
```

Após retirarmos o ; , podemos fazer um novo teste.



Temos todos os itens da listas e agora, eles são links. Se clicarmos em algum carro, veremos a navegação.



Os títulos ficarão diferentes em cada sistema operacional e os cards estarão centralizados - ficaram pequenos, mas funcionam. Mais adiante, continuaremos a trazer a documentação para a aplicação.