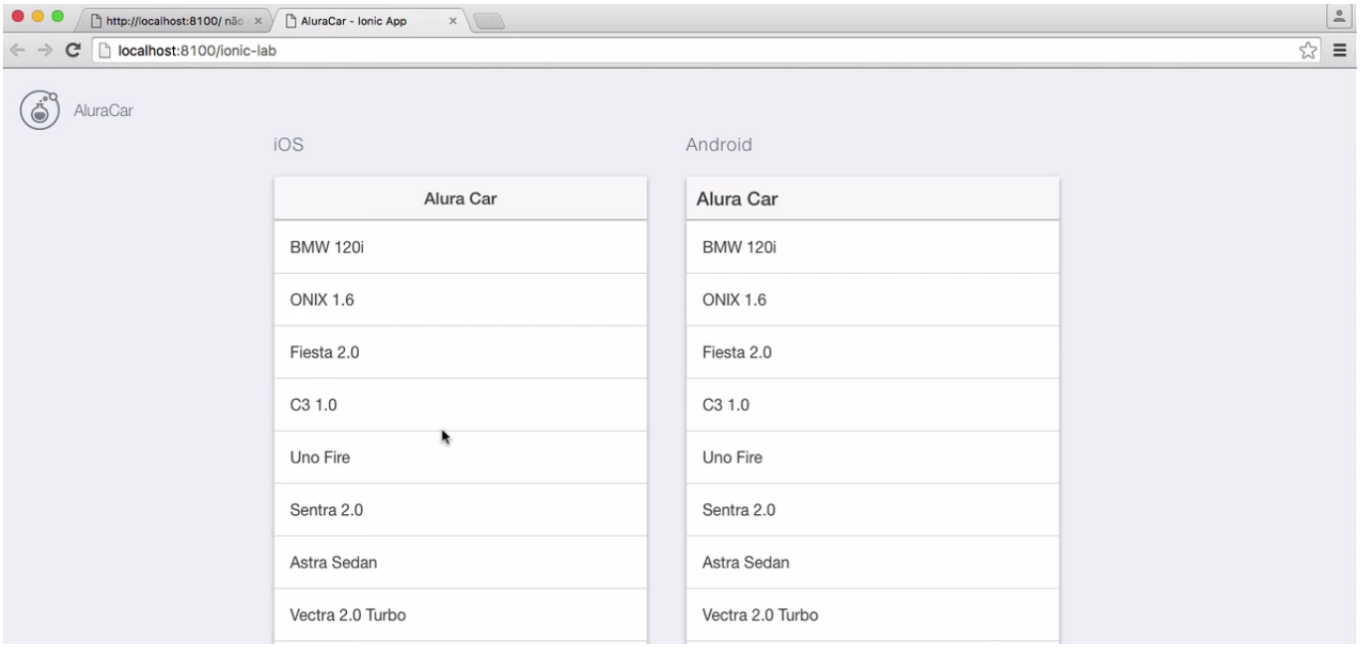
02

Separando lógica e visualização em camadas

Vamos dar continuidade na construção da aplicação mobile para a concessionária. Atualmente, ela está assim:



Teremos que incluir mais dois modelos na listagem. Podemos adicionar usando a tag .

```
<ion-content>
  <ion-list>
    <ion-item>BMW 120i </ion-item>
    <ion-item>ONIX 1.6 </ion-item>
    <ion-item>Fiesta 2.0 </ion-item>
    <ion-item>C3 1.0 </ion-item>
    <ion-item>Uno Fire </ion-item>
    <ion-item>Sentra 2.0 </ion-item>
    <ion-item>Astra Sedan </ion-item>
    <ion-item>Vectra 2.0 Turbo </ion-item>
    <ion-item>Hilux 4x4 </ion-item>
    <ion-item>Montana Cabine dupla </ion-item>
    <ion-item>Outlander 2.4 </ion-item>
    <ion-item>Fusca 1500 </ion-item>

  </ion-list>
</ion-content>
```

Se todas as vezes que precisarmos incluir um item, tivermos que adicionar uma nova linha, nosso código ficará muito grande - dificultando futuramente, a manutenção. Por isso, iremos conhecer outra ferramenta poderosa que é o Angular. Já recomendamos que você faça o curso de Angular da plataforma da Alura.

Agora, vamos retirar a lista da `view` e movê-la para uma `controller` . Para isto, criaremos outro arquivo `js` . Primeiramente, o que faremos é chamar o `angular` .

```
angular.module()
```

Podemos conferir o nome do módulo da nossa aplicação no `index.html` ou no `app.js`. Veremos que em ambos encontraremos a referência a `starter`. Iremos usar o mesmo e depois, criar a `controller`.

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

});
```

Chamamos o `controller` de `ListagemController` e a `function()` recebeu como injeção de dependência o `$scopo`.

Em seguida, iremos chamar o arquivo como `controller.js`. Vamos adicionar o arquivo no `index.html`. Na parte sobre JS, localizada acima da lista, adicionaremos:

```
    <!-- your app´s js -->
    <script src="js/app.js"></script>
    <script src="js/controller.js"></script>
  </head>
  <body ng-app="starter">
```

Até aqui não veremos nenhuma alteração na tela.

Voltaremos ao `controller.js` e criaremos a variável `$scope.listaDeCarros`, que logo irá receber um *array* de carros.

No arquivo `index.html`, e informaremos que a `view` irá receber a `controller`.

```
body ng-app="starter" ng-controller="ListagemController"

    <ion-pane>
      <ion-header-bar class="bar-stable">
        <h1 class="title">Alura Car </h1>
      </ion-header-bar>
```

Agora, como o Angular pode nos ajudar com a lista de carros? Nós repetimos vários vezes a tag `<ion-header-bar>`. Podemos otimizar o código, criando uma regra para a diretiva. Para sinalizar que queremos repeti-la, usaremos a tag do Angular `ng-repeat`. Especificaremos que queremos repetir que queremos um `carro` na `listaDeCarros`, que criamos no `controller.js`.

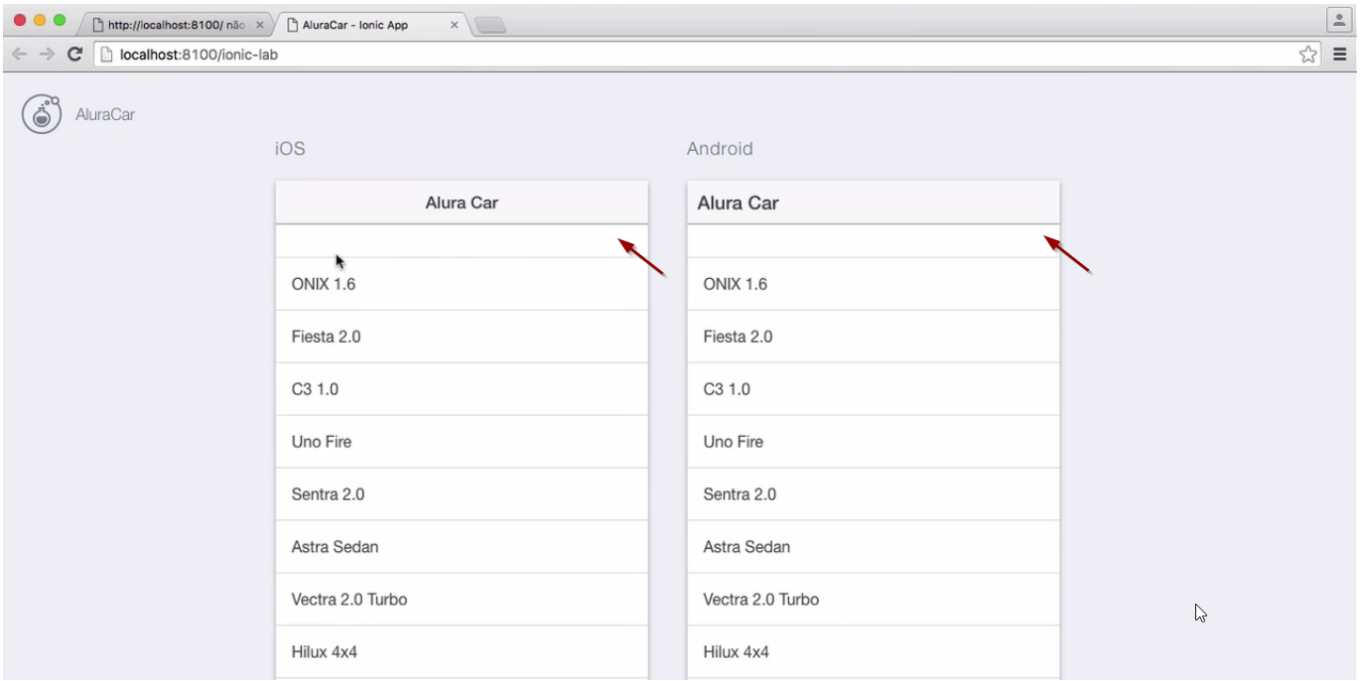
```
<ion-content>
  <ion-list>
    <ion-item ng-repeat="carro in listaDeCarros" >

    </ion-item>

    <ion-item>BMW 120i </ion-item>
    <ion-item>ONIX 1.6 </ion-item>
    <ion-item>Fiesta 2.0 </ion-item>
    <ion-item>C3 1.0 </ion-item>
    <ion-item>Uno Fire </ion-item>
    <ion-item>Sentra 2.0 </ion-item>
    <ion-item>Astra Sedan </ion-item>
    <ion-item>Vectra 2.0 Turbo </ion-item>
    <ion-item>Hilux 4x4 </ion-item>
```

```
    <ion-item>Montana Cabine dupla </ion-item>
    <ion-item>Outlander 2.4 </ion-item>
    <ion-item>Fusca 1500 </ion-item>

  </ion-list>
</ion-content>
```

Em seguida, removeremos os carros da lista e vamos passá-los para o `controller.js`. Começaremos movendo o `BMW 120i`.

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

    $scope.listaDeCarros = ['BMW 120i'];
});
```

Se visualizarmos a nossa aplicação como está agora, veremos que aparece um item vazio.



Vamos usar um recurso forte do Angular que é o Data Binding. Adicionaremos as chaves (`{}`) e dentro o (`carro`).

```
<ion-content>
  <ion-list>
    <ion-item ng-repeat="carro in listaDeCarros" >
      {{carro}}
    </ion-item>
```

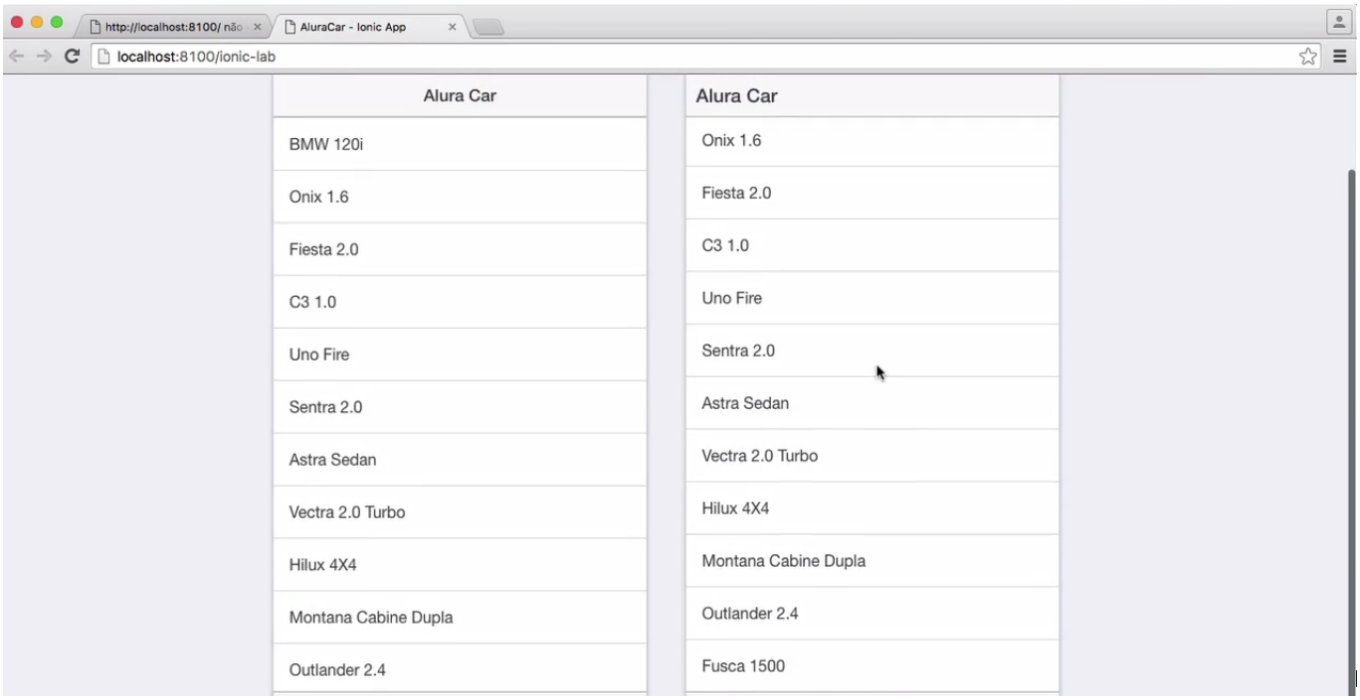
Depois de salvar, veremos como está o nosso aplicativo.

Conseguimos visualizar o primeiro item. Faz sentido usarmos o `controller`. Agora, podemos mover todos os itens do `index.html`, que ficará apenas com o item `carro`, para o `controller.js`.

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

```
    $scope.listaDeCarros = ['BMW 120i', 'Onix 1.6', 'Fiesta 2.0', 'C3 1.0', 'Uno Fire', 'Sentra 2.0'
});
```

A lista estará completa tanto no Android, como no iOS.



Conhecemos nesta aula o poder do Angular, usamos o **Data Binding** e o **Controller**, além de deixarmos mais limpo o `index.html` e nossa aplicação rodando mais rápido.