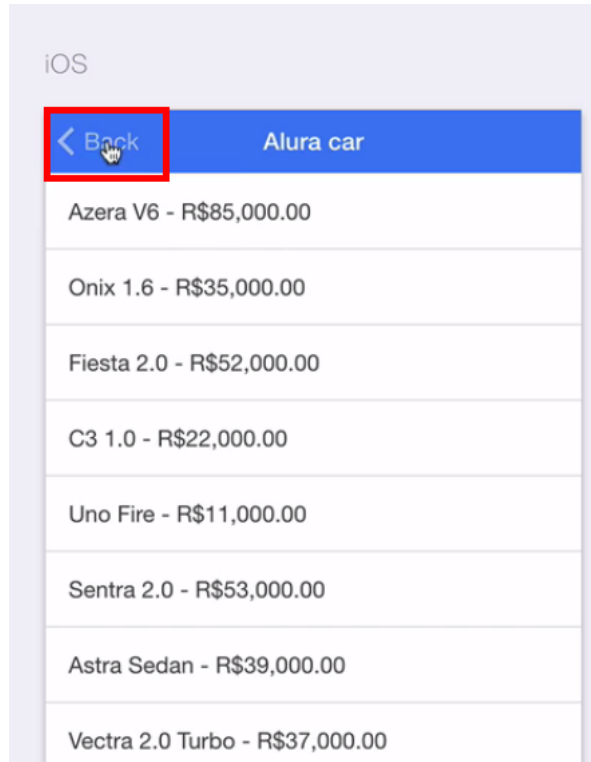


Criação do nosso menu lateral SideMenu

Transcrição

Vamos fazer o login da aplicação e ir para a listagem.



A navegação está permitindo ao usuário que ele volte para o login. Esse comportamento é bem estranho. O padrão é que ao fazermos o login, não poderemos mais voltar nesta tela. É possível fazer um logout, mas não é o caso que estamos falando.

Nós vamos trocar essa `nav-bar` e colocarmos um menu hambúrguer, que ao clicarmos ele abrirá ocupando metade da tela. Podemos adicionar conteúdos, como dados pessoais, componentes de cartões, uma lista de menus. Substituiremos o botão de voltar desta forma.

Começaremos a montar esse menu, que não será uma View comum, que terá as tags `<ion-view>` e `<ion-content>`. Mas ele é um template, por isso, salvaremos o `menu.html` na pasta `template`.

Ele será um conteúdo na lateral e a partir dele teremos a navegação no aplicativo. Usaremos as seguintes tags: `<ion-side-menus>`, `<ion-side-menu-content>` e `ion-side-menu>`.

```
<ion-side-menus>
  <ion-side-menu-content>

  </ion-side-menu-content>
  <ion-side-menu>

  </ion-side-menu>
</ion-side-menus>
```

Dentro da tag `<ion-side-menu-content>` , guardaremos a parte do cabeçalho, em que colocaremos a barra de navegação. Já na `<ion-side-menus>` , colocaremos o conteúdo do menu. Temos a liberdade de adicionar qualquer conteúdo nesta parte.

Em seguida, usaremos o `<ion-nav-bar>` e o `<ion-nav-view>` .

```
<ion-side-menus>
  <ion-side-menu-content>
    <ion-nav-bar>

    </ion-nav-bar>

    <ion-nav-view>

    </ion-nav-view>

  </ion-side-menu-content>
</ion-side-menu>

</ion-side-menus>
```

No `nav-view` , será onde a aplicação irá navegar, para isto adicionaremos `menuContent` . Então, nós iremos modificar a navegação criado no outro curso anterior. E no `nav-bar` adicionaremos os `buttons` da navegação.

```
<ion-side-menus>
  <ion-side-menu-content>
    <ion-nav-bar>
      <ion-nav-buttons side="left">
        <button></button>

      </ion-nav-buttons>
    </ion-nav-bar>

    <ion-nav-view name="menuContent"> </ion-nav-view>

  </ion-side-menu-content>
</ion-side-menu>

</ion-side-menus>
```

O `button` receberá algumas classes CSS, para aparecer o ícone do menu hambúrguer com os três traços.

```
<button class="button button-icon button-clear ion-navicon" menu-toggle="center"></button>
```

As quatro classes farão o estilo do botão. Adicionamos também o atributo `menu-toggle` para deixá-lo centralizado.

Agora, dentro dos botões temos a `nav-bar` e a `nav-view` . No `ion-side-menu` , usaremos um atributo para deixar o botão à esquerda. Com o `ion-content` , colocaremos o conteúdo.

```
<ion-side-menu side="left">
  <ion-content>
    Olá
  </ion-content>
</ion-side-menu>
```

Mas se salvarmos as alterações, veremos que o menu hambúrguer ainda não é visualizado. O botão "Back" aparece, porque não criamos a nossa rota. Como não construímos uma View normal, também não construiremos uma rota normal. Estamos trabalhando com uma View abstrata, porque ela não será uma View que acessaremos e que terá todo o planejamento de uma View. É a partir dela que teremos a nossa navegação. Então, precisamos declarar que vamos trabalhar com uma View abstrata e as rotas que usaremos com esse menu vão rodar em cima do componente `menuContent`.

No arquivo `routes.js`, precisarei declarar uma rota nova: `app`. Ela ficará posicionada acima da rota de `listagem`.

```
.state('app', {
  url : '/app',
  templateUrl : 'templates/menu.html',
  abstract: true
})
```

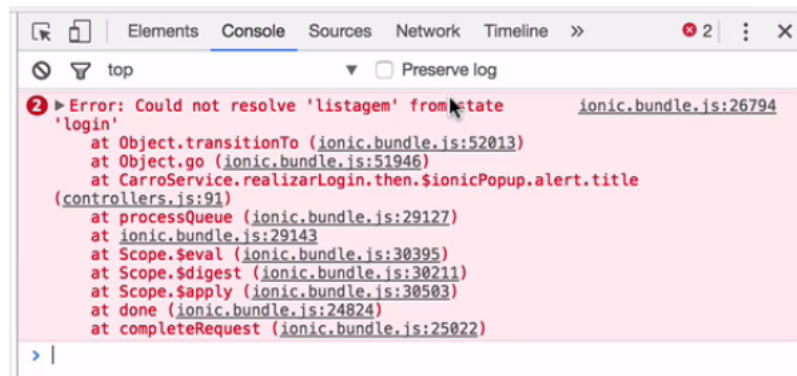
Com o atributo `abstract`, nós especificamos para o Ionic e para o Angular que a rota é abstrata. Precisaremos adicionar uma rota concreta para adicionar o menu. Faremos isto refatorando a rota concreta `listagem`. Atualmente ela está assim:

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

Começaremos concatenando o nome das rotas abstrata com a concreta. Teremos também um atributo chamado `views`, que passará a guardar a controller.

```
.state('listagem', {
  url : '/listagem',
  views : {
    `menuContent` : {
      templateUrl : 'templates/listagem.html',
      controller: 'ListagemController'
    }
  }
})
```

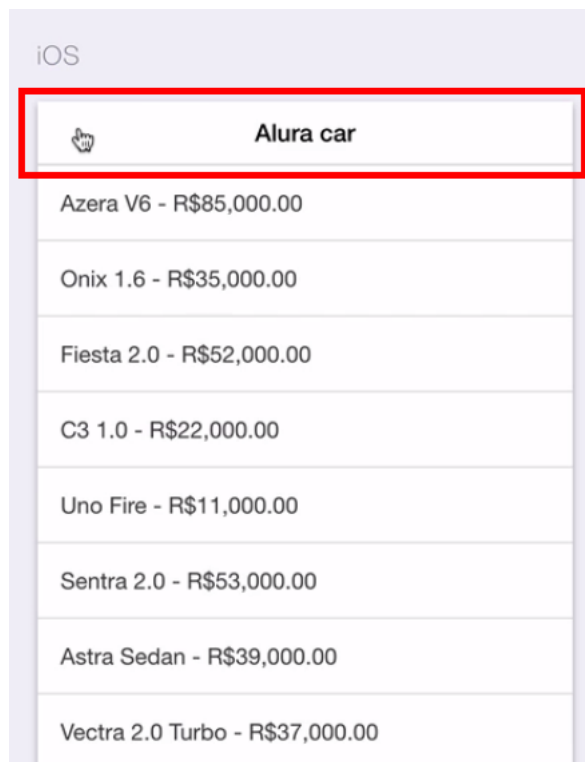
No entanto, se tentarmos fazer o login agora na nossa app, veremos que não conseguiremos. Vamos utilizar a "ferramenta de desenvolvedor" do Chrome para identificar o erro.



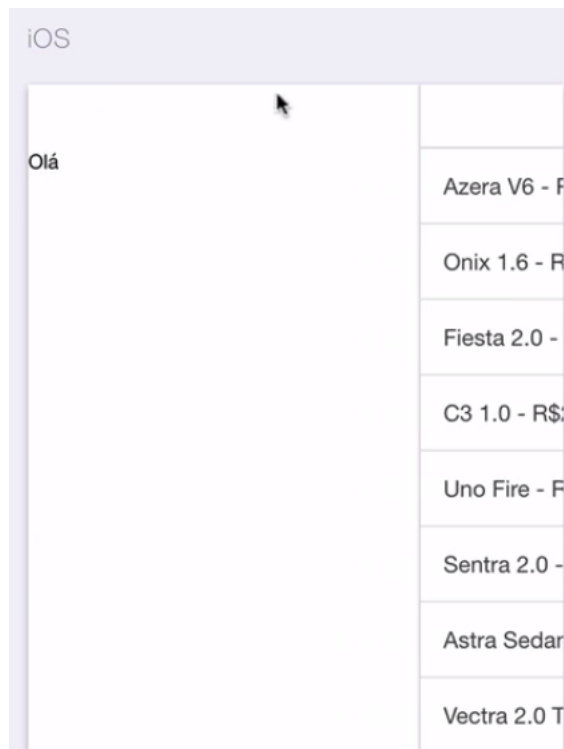
Vemos que uma mensagem de erro diz que não é possível resolver listagem a partir do state 'login'. Quando refatoramos a rota listagem e o botão "login", nos envia para listagem. Nós temos que mudar no controller quando clica no botão "Entrar". Vamos fazer as alterações no arquivo `controllers.js` :

```
CarroService.realizarLogin(dadosDoLogin).then(function(dados){
  $state.go('app.listagem');
}, function(erro){
  $ionicPopup.alert({
    title : 'Opa!',
    template : 'Email ou senha incorretos.'
  })
})
```

Após salvarmos, poderemos testar novamente.



Observe que o fundo do Alura Car está branco e tem um link imperceptível, mas que quando clicado, abrirá o menu.



O menu hambúrguer está ali, mas tanto a barra como o ícone são brancos, então, ficou difícil visualizá-los. Cometemos o mesmo erro no primeiro curso. Mas o menu já está funcionando. Em seguida, mudaremos a cor da barra para azul, assim o botão branco poderá aparecer.

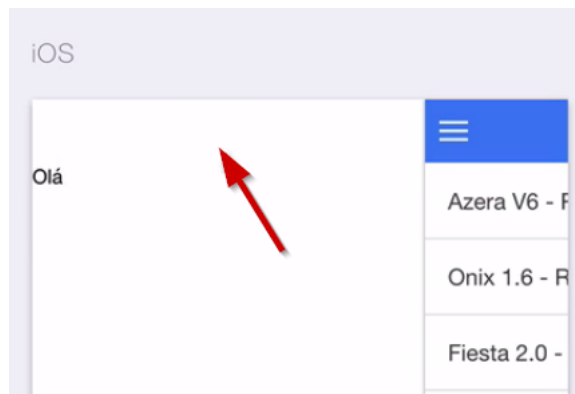
Dentro de `ion-nav-bar`, adicionaremos a classe `bar var-positive`.

```
<ion-nav-bar class="bar bar-positive">
  <ion-nav-buttons side="left">
    <button class="button button-icon button-clear ion-navicon" menu-toggle="center"></button>
  </ion-nav-buttons>
</ion-nav-bar>
```

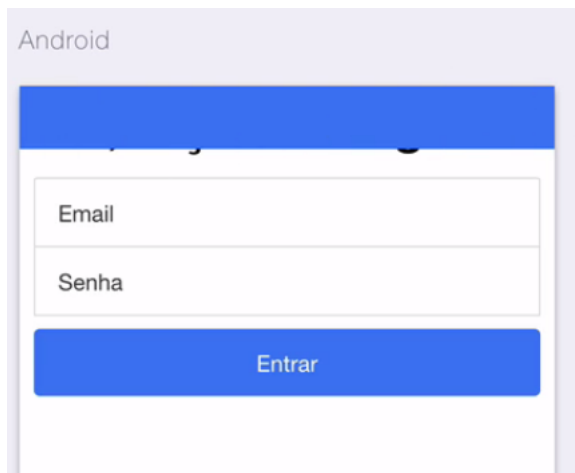
Após salvarmos, podemos ver o menu mais visível.



Mas ao abrirmos o menu, parece haver um espaço no topo.



Está ali a tag `<ion-nav-bar>` que adicionamos no `index` - ainda no primeiro curso. Estamos trabalhando com duas `<ion-nav-bar>` e nós queremos trabalhar apenas com a do menu. Podemos ver a `<nav-bar>` do `index.html` mais perceptível na tela do login.



Vamos retirar a tag do arquivo. Ela se encontra da tag `<body>`, localizada atualmente abaixo do fechamento da tag `<head>`.

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

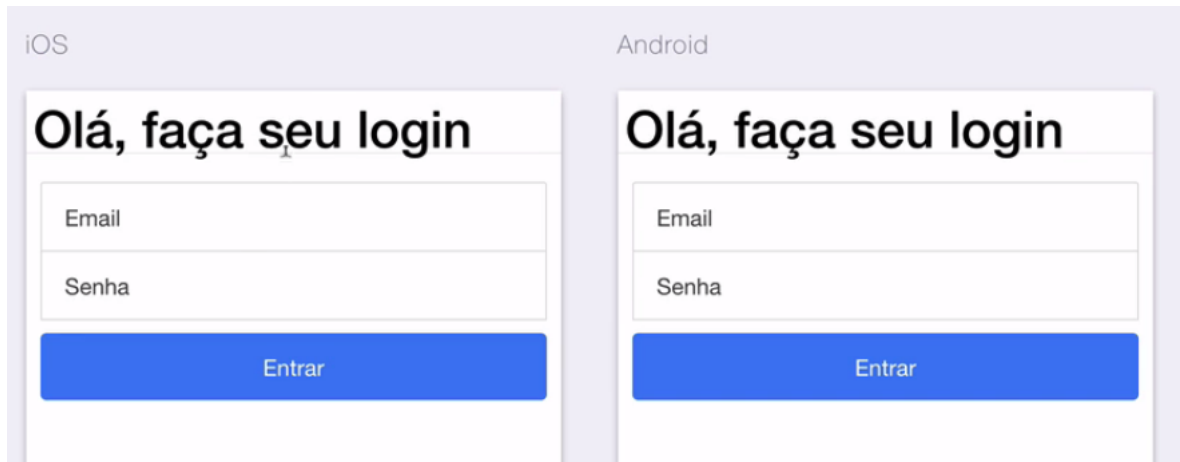
    <ion-footer-bar class="bar-stable">
      <h1 class="title">Rua Vergueiro, 3185</h1>
    </ion-footer-bar>
  </ion-pane>
</body>
</html>
```

Com a alteração, o trecho ficará assim:

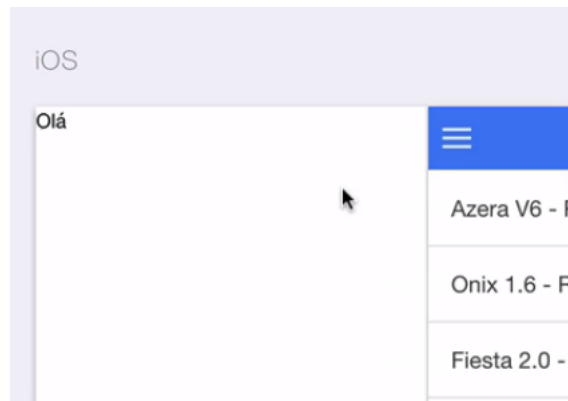
```
<body ng-app="starter" ng-controller="ListagemController">
  <ion-pane>
    <ion-nav-view>
      </ion-nav-view>

    <ion-footer-bar class="bar-stable">
```

```
<h1 class="title">Rua Vergueiro, 3185</h1>
</ion-footer-bar>
</ion-pane>
</body>
</html>
```



Observe que a barra que ficava acima tampando a frase já desapareceu. Veremos que ela sumiu também do menu.



Criamos a funcionalidade do nosso menu hambúrguer.