

## Botão de finalizar pedido

### Botão de finalizar pedido

Da aula anterior, já temos a lista de acessórios com a modificação dos preços. O que acrescentaremos agora é a finalização do pedido. O cliente já vê o carro escolhido, o preço com os acessórios e agora criaremos uma terceira tela - uma nova *view* -, a partir do nosso editor de texto. Nele, já criamos uma pasta exclusiva para templates, que será usada agora. Criaremos um novo arquivo chamado `finalizarpedido.html`.

Esse arquivo é um HTML que estruturaremos tendo em vista que ele será uma *view*. Já criamos duas *views* anteriormente: a `listagem.html` e a `carroescolido.html`. Elas tem três *tags* em comum: `<ion-view>`, `<ion-nav-title>` e `<ion-content>`. Assim, a nossa nova *view* seguirá a mesma estrutura básica. O código ficará assim:

```
<ion-view>
  <ion-nav-title>Estamos quase lá</ion-nav-title>
  <ion-content>

  </ion-content>

</ion-view>
```

Agora, ao testarmos a aplicação e adicionarmos os acessórios desejados, vemos que ainda não aparece nenhuma indicação dessa tela nova que estamos criando. Como é um aplicativo mobile e não um site em que o usuário poderia editar a URL, precisamos dar uma indicação. Então colocaremos um link na HTML. Mas como faremos isso?

O link deverá ser colocado na tela anterior, de carros escolhidos, para que o usuário acesse a nova tela. Ele será inserido abaixo do card de acessórios, que até agora tem o seguinte código:

```
<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">
        {{carroEscolhido.nome}}

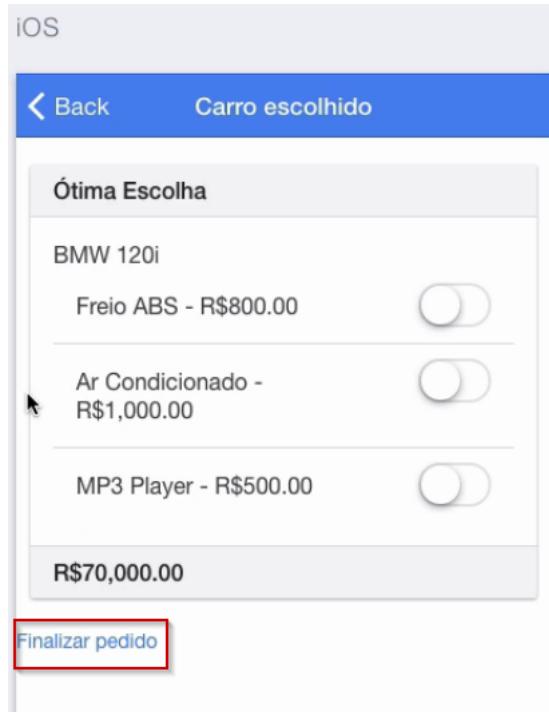
      <ul class="list">
        <li class="item item-toggle" ng-repeat="acessorio in listaDeAcessorios">
          {{acessorio.nome}} - R{{acessorio.preco | currency }}
          <label class="toggle toggle-assertive">
            <input type="checkbox" ng-model='isMarcado' ng-click="mudou(acessorio, isMarcado)">
            <div class="track">
              <div class="handle"></div>
            </div>
          </label>
        </li>
      </ul>
    </div>
```

```
</div>
<div class="item item-divider">
  R{{carroEscolhido.preco | currency }}
</div>
</div>
```

O que vamos acrescentar tem a tag `<a>`, que preencheremos com "Finalizar pedido", ficando assim:

```
<a>Finalizar pedido</a>
```

Ao voltarmos para o navegador, veremos que ainda não está com a aparência de uma aplicação mobile.



Para melhorar isso, usaremos um CSS com três classes. Então a tag que era `<a>` ficará:

```
<a class="button button-positive button-full">Finalizar pedido</a>
```

E a aparência no navegador ficará assim:



Ao clicarmos no botão recém-criado, veremos que ele reage, mas não nos leva à tela seguinte. Isso acontece porque ainda não indicamos a ação do botão, nem que rota ele deve seguir.

Vamos criar a rota para ele no arquivo de rotas, o `routes.js`. Por enquanto ele está assim:

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

$urlRouterProvider.otherwise('listagem');

$stateProvider

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

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

})
})
```

Agora acrescentaremos logo abaixo a rota para a tela de finalização, com o trecho:

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

})
```

Como ainda não criamos uma `controller`, é isso que faremos agora no arquivo `controllers.js`.

```
//...
$scope.mudou = function(acessorio, isMarcado) {

    if (isMarcado) {
        $scope.carroEscolhido.preco =
            $scope.carroEscolhido.preco + acessorio.preco;
    } else {
        $scope.carroEscolhido.preco =
            $scope.carroEscolhido.preco - acessorio.preco;
    }
};

});

angular.module('starter')
.controller('FinalizarPedidoController', function(){

});
```

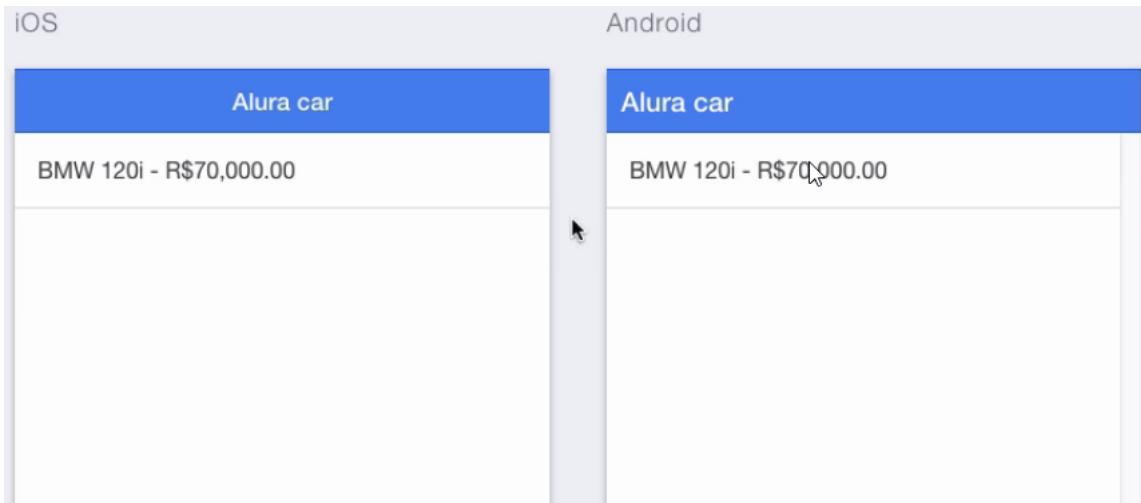
O `controller` que adicionamos tem dois argumentos: `FinalizarPedidoController` e `function()`. Salvo este arquivo, podemos voltar à tela que estávamos criando.

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

Apesar de já termos a `URL`, o `template` e o `controller`, ao testarmos no navegador, o botão ainda não nos leva à terceira tela. Isso acontece porque o link ainda não está chamando a rota. Assim como fizemos com a rota `carroescolhido`, acrescentaremos ao link o `href` com a rota que acabamos de criar:

```
<a class="button button-positive button-full" href="#/finalizarpedido/">Finalizar pedido</a>
```

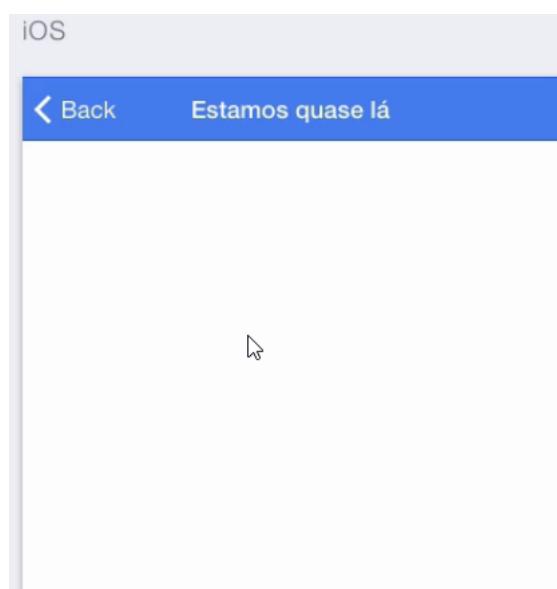
Após salvar e testar no navegador, chegaremos à seguinte tela:



Essa é a tela inicial, que não é à qual planejamos chegar. Reavaliando o nosso link, vemos que, embora a rota `finalizarpedido` seja a correta, o programa não a está encontrando e nos redireciona para a rota padrão `listagem`. Portanto, algo deve estar errado no link que fizemos.

```
<a class="button button-positive button-full" href="#/finalizarpedido/">Finalizar pedido</a>
```

Comparando a rota com o link, vemos que há uma barra ( / ) a mais ao final deste. Embora seja um erro de rota do Angular, nos deparamos com ele no Ionic. Ao removê-la, somos levados à terceira tela:



Como já chegamos ao nosso objetivo, podemos encerrar a aula. Até a próxima!