

Revisão

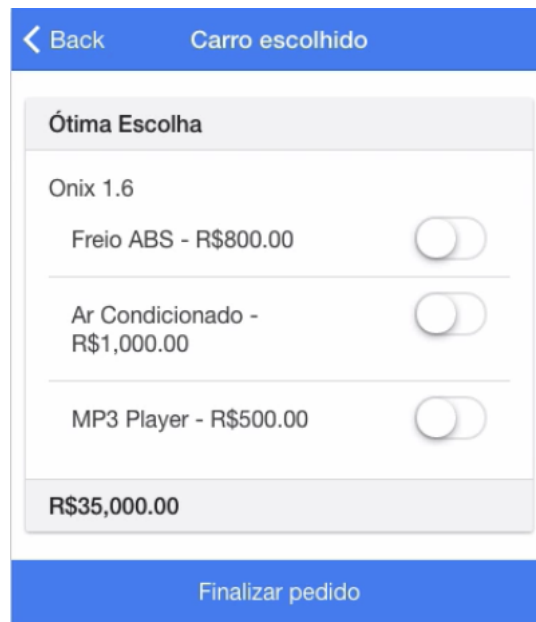
Revisão

Essa é a nossa última aula e aproveitaremos para fazer uma revisão do que vimos até agora. Depois que instalamos o Ionic, construímos uma aplicação que era apenas a tela da lista só com os nomes dos carros. Para isso, usamos as tags do Ionic, como a `<ion-header>` para o cabeçalho, a `<ion-footer>` para o rodapé, e a `<ion-content>` para conteúdo, que também foi usada em outras telas. Depois de acrescentados os preços, a tela inicial ficou assim:

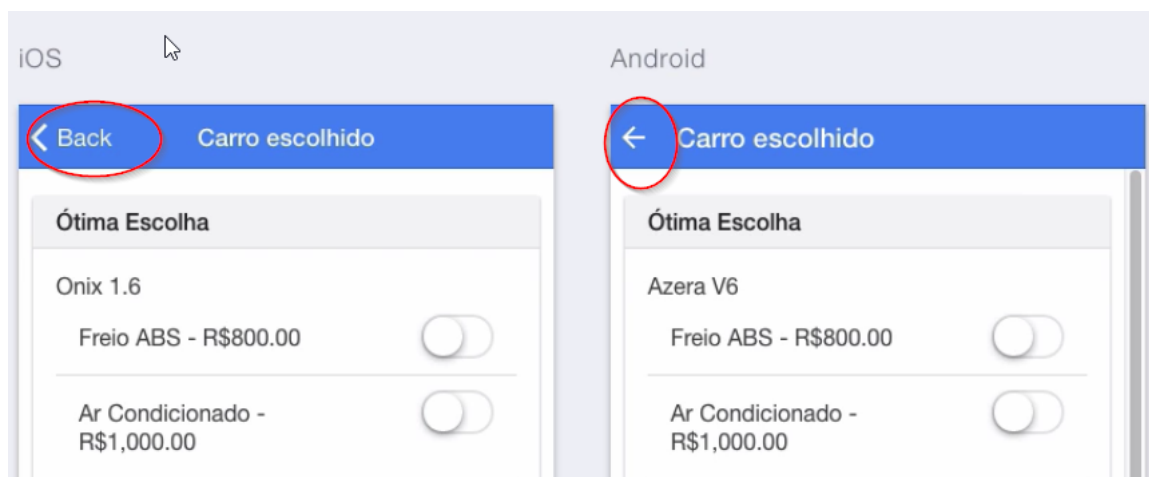
Alura car
Azera V6 - R\$85,000.00
Onix 1.6 - R\$35,000.00
Fiesta 2.0 - R\$52,000.00
C3 1.0 - R\$22,000.00
Uno Fire - R\$11,000.00
Sentra 2.0 - R\$53,000.00
Astra Sedan - R\$39,000.00
Vectra 2.0 Turbo - R\$37,000.00
Hilux 4x4 - R\$90,000.00

Depois, vimos que era necessário que a aplicação tivesse uma navegação, então precisamos fazer uma refatoração no código, tirando a lista do `index` e criando um `template` para ela. Fizemos um arquivo separado para os *templates*, bem como para os *controllers*, para os serviços e para as *views*.

Depois que criamos a navegação, que permite que passemos de uma tela para outra, continuamos explorando os componentes do Ionic, como o card de acessórios. Embora ele não seja exatamente do Ionic, ele faz parte de sua documentação, que vimos bastante no curso. Como já mencionado, a documentação é uma ferramenta importante para contruir componentes interessantes dentro do Ionic.



Depois, vimos a `tag` de `view` para agrupar as `views` de listagem, de carro escolhido e, depois, de envio do pedido. Uma das grandes vantagens das `tags` do Ionic é que automaticamente criado um padrão entre iOS e Android. Um exemplo disso é o botão "Back" do iOS, que no Android é apenas uma seta, mas ambos vêm da `tag` `<ion-nav-back-button>`.



Vimos também a navegação por rota do Angular e a configuração do preço na lista de acessórios por *data binding*, que altera um objeto (nesse caso o rodapé do card) quando alteramos outro (a seleção do acessório). Na tela seguinte criamos um formulário (outro componente que tiramos da documentação do Ionic) para os dados do comprador, e colocamos pop-ups: um para quando o formulário está preenchido, outro para quando não está.

Pudemos também testar a aplicação no celular, e vimos que podemos mandá-la para o celular do nosso cliente, seja ele Android ou iOS. Lembrando que a grande vantagem do Ionic é seu código único para os dois sistemas.



Assim, construímos uma aplicação do começo ao fim usando o Ionic como *framework*, com a ajuda do Angular, que é outro *framework* bem útil. Finalizamos o curso por aqui, torcendo para que você tenha gostado. Até o próximo!