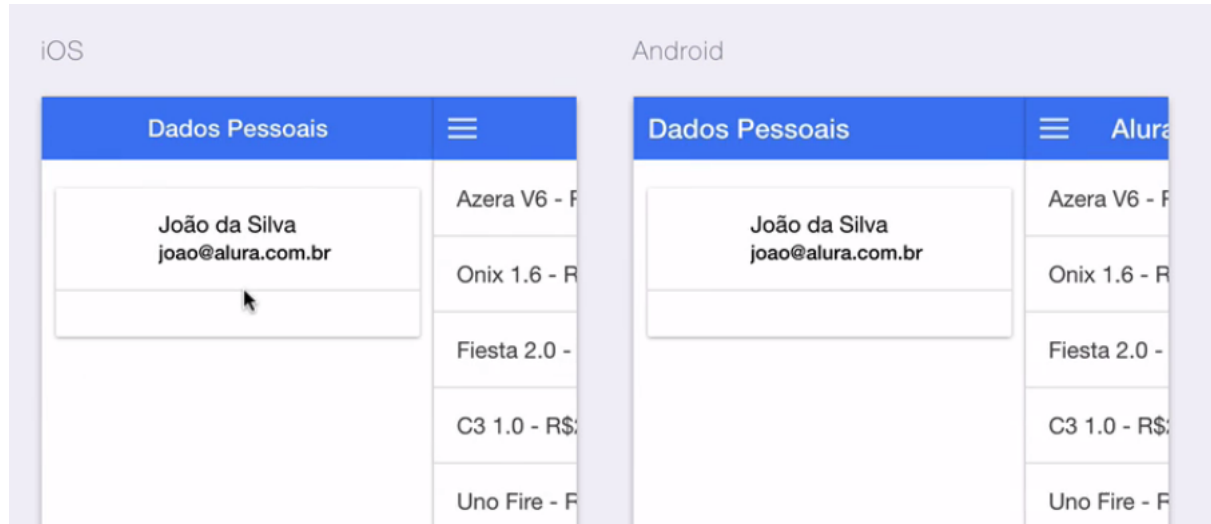


## Criação da nossa tela do perfil

### Transcrição

Agora, a nossa aplicação tem um menu na lateral.



Veja que conseguimos visualizar os dados do usuário (nome e e-mail). Mas tem outros dados básicos, como a data de nascimento e telefone, por exemplo, que recebemos na aplicação mobile, quando fazemos o login. O nosso cliente quer dar a opção para o usuário editar esses dados. Além de visualizá-los, poder alterar o nome e e-mail, caso eles tenham sido modificados.

Para isso, ele precisa de uma tela, em que colocaremos os dados. Criaremos um novo arquivo, que chamaremos de `perfil.html`, que será salvo na `pasat` `template.html`. Começaremos adicionando as tags `<ion-view>` e o `<ion-content>`. Dentro do conteúdo, vamos criar um componente de card semelhante ao que usamos no menu.

```
<ion-view>
  <ion-content>
    <div class="list card">
      <div class="item item-avatar">
        Usuário Logado
      </div>
      <div class="item item-body">
      </div>
    </div>

  </ion-content>

</ion-view>
```

Já adicionamos o conteúdo do cabeçalho (Usuário Logado). No `body`, vamos colocar os dados, adicionando `label`s para Data de Nascimento, Telefone e E-mail.

```
<div class="item item-body">
  <div class="list">
    <label class="item item-input">
```

```

    <span class="input-label">Data de Nascimento</span>
  </label>
  <label class="item item-input">
    <span class="input-label">Telefone</span>
  </label>
  <label class="item item-input">
    <span class="input-label">E-mail</span>
  </label>
</div>
</div>

```

Adicionaremos o `button` para criar o botão de `Editar`.

```

<button class="button button-full button-positive">Editar</button>

```

Nossa tela já estará assim:



Faltou adicionar o link para que ao clicarmos, seja chamada a nova tela. Precisaremos criar uma lista de menus e assim conseguir irmos a tela de perfil.

No arquivo `menu.html`, abaixo do componente `card`, vamos adicionar uma lista. Adicionaremos as tags `ion-list` e `ion-item`:

```

<ion-list>
  <ion-item href=>
    Perfil
  </ion-item>
</ion-list>

```

Precisamos criar a rota para a tela de Perfil. No arquivo `routes.js`, adicionaremos o `app.perfil`.

```

.state('app.perfil', {
  url: '/perfil',
  views: {

```

```

    'menuContent' : {
      templateUrl : 'templates/perfil.html'
    }
  }
})

```

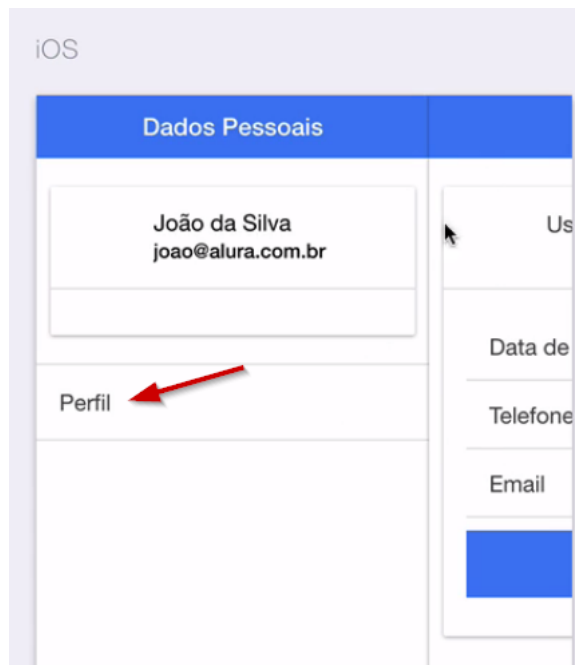
Agora, no `menu.html`, podemos chamar o `/perfil`.

```

<ion-list>
  <ion-item href="#/app/perfil">
    Perfil
  </ion-item>
</ion-list>

```

Após salvarmos as alterações, podemos testar. Já vemos o link de `Perfil`.



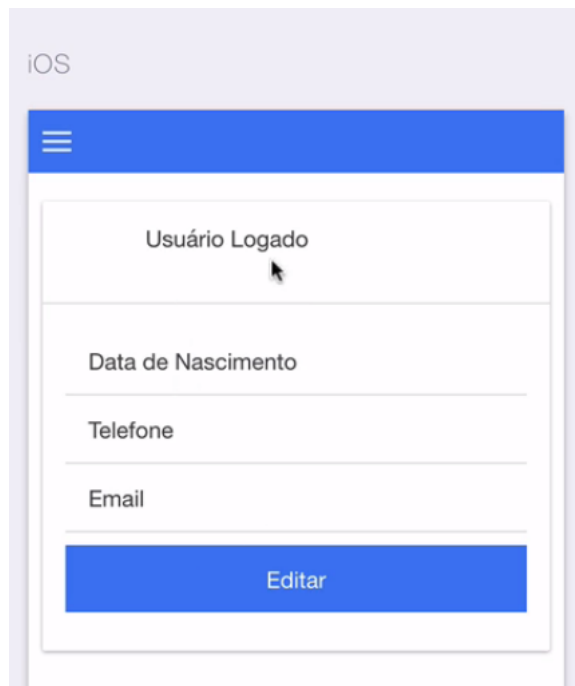
Ao lado, conseguimos visualizar a tela de perfil. Como nós não definimos que o menu deve fechar, ele permaneceu aberto simultaneamente com a mesma tela. Existe uma diretiva exclusiva para fecharmos o menu: `menu-close`. Vamos adicioná-la.

```

<ion-list>
  <ion-item menu-close href="#/app/perfil">
    Perfil
  </ion-item>
</ion-list>

```

Agora, após clicarmos em "Perfil", o menu será fechado e a tela de Perfil será aberta, que inclui o botão "Editar".



Nós agora criamos uma lista de menu. Podemos clicar em um link e acessar uma nova tela.

Até a próxima!