

TableView

Transcrição

A partir de agora faremos alterações na aba "Editar" colocando elementos similares aos de "Usuário", incluindo a borda que fizemos por meio do `Frame`. Porém, para serem editáveis de fato, os controles deverão funcionar como um formulário.

Primeiramente criaremos um container com mesmo padrão utilizado na aba "Usuário", e controles parecidos. Pausando a aplicação, em `MasterView.xaml` vemos que para cada aba há um `ContentPage`, e dentro do `ContentPage` referente a "Usuário", temos um `Frame` e um `StackLayout`. Para fins de praticidade, copiaremos estes códigos para dentro do `ContentPage` referente a "Editar".

```
<ContentPage Title="Editar">
    <Frame OutlineColor="Silver"
    VerticalOptions="CenterAndExpand"
    Margin="15">
        <StackLayout VerticalOptions="Center" HorizontalOptions="Center">
            </StackLayout>
        </Frame>
    </ContentPage>
```

Feito isto, rodaremos a aplicação para verificarmos como esta alteração aparece. Há um `Frame` que contém um `StackLayout`, o qual neste momento não conseguimos ver. Para começarmos a colocar os controles da edição de perfil do usuário, é necessário utilizar campos de coleta de dados.

Como visto no [primeiro curso](https://cursos.alura.com.br/course/xamarin-aplicativos-mobile-com-visual-studio-parte-1) (<https://cursos.alura.com.br/course/xamarin-aplicativos-mobile-com-visual-studio-parte-1>), existe um controle que é um container próprio para ser usado como "pai" de controles de formulários.

Este componente do Xamarin Forms funciona como agregador de formulários, e se chama `TableView`, sendo que uma de suas funções é coletar dados. Ele possui uma raiz (`TableRoot`) com a qual é possível a criação de distintas seções para cada uma das partes do formulário.

Tendo como exemplo um formulário de cadastro de currículos, pode-se ter um `TableView` com uma seção para cadastro de dados pessoais, outra para formação acadêmica, outra para experiências profissionais, e assim por diante.

Neste caso, usaremos apenas uma seção para colocarmos uma legenda chamada `TableSection`, com título "Dados Pessoais". Para a entrada de dados, utilizaremos uma célula dentro de um `TableView` (`EntryCell`). Colocaremos uma legenda externa, como um `Label`, mas temos a opção de incluirmos uma interna, visível enquanto o usuário não digitar naquele campo, o `Placeholder`:

```
<ContentPage Title="Editar">
    <Frame OutlineColor="Silver"
    VerticalOptions="CenterAndExpand"
    Margin="15">
        <StackLayout VerticalOptions="Center" HorizontalOptions="Center">
            <TableView>
                <TableRoot>
```

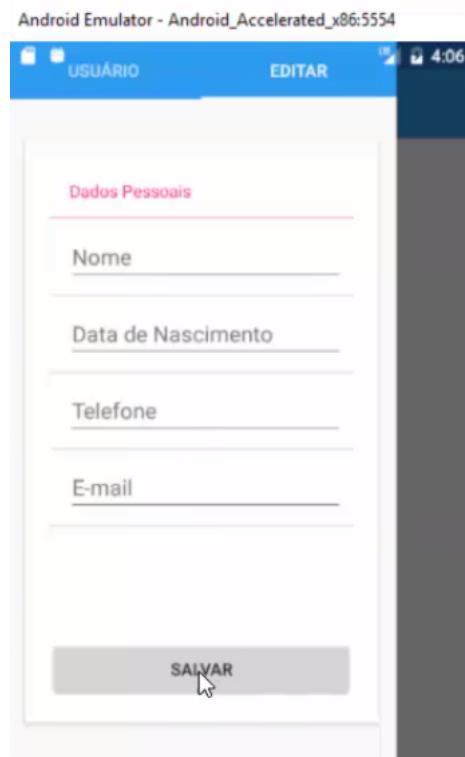
```
<TableSection Title="Dados Pessoais">
    <EntryCell Placeholder="Nome"></EntryCell>
</TableSection>
</TableRoot>
</TableView>
</StackLayout>
</Frame>
</ContentPage>
```

Vamos verificar como isto se apresenta ao usuário rodando a aplicação. Tudo foi implementado da maneira como gostaríamos, portanto continuaremos acrescentando mais campos neste `TableView`. Falta colocarmos "data de nascimento", "telefone" e "e-mail", após o qual rodaremos novamente a app.

Incluiremos um botão para salvarmos os dados editados. Pausaremos a aplicação e, logo após o fechamento da tag `<TableView>`, criaremos o controle `<Button>`, com texto "Salvar":

```
<ContentPage Title="Editar">
    <Frame OutlineColor="Silver"
VerticalOptions="CenterAndExpand"
Margin="15">
    <StackLayout VerticalOptions="Center" HorizontalOptions="Center">
        <TableView>
            <TableRoot>
                <TableSection Title="Dados Pessoais">
                    <EntryCell Placeholder="Nome"></EntryCell>
                    <EntryCell Placeholder="Data de Nascimento"></EntryCell>
                    <EntryCell Placeholder="Telefone"></EntryCell>
                    <EntryCell Placeholder="E-mail"></EntryCell>
                </TableSection>
            </TableRoot>
        </TableView>
        <Button Text="Salvar"></Button>
    </StackLayout>
</Frame>
</ContentPage>
```

Ao rodarmos a aplicação mais uma vez, obteremos esse resultado:



Falta dar uma função para este botão, sendo assim, implementaremos as funcionalidades do formulário.