

□ 10

Button da mensagem e revisão

Transcrição

Criaremos em nossa `view` de Agendamento o botão a ser clicado pelo usuário, em seguida, mostraremos os dados preenchidos. Como vimos, o `TableView` é um controle que ocupa todo o `ContentPage`, a raiz da nossa página.

Colocaremos um botão que aparecerá no fim da página, porém, com este `TableView` ocupando toda a página, precisaremos colocá-lo em outro container, o `StackLayout`.

Configuraremos o `Button`, que possui o texto "Agendar" com evento `Button_Clicked()`, disponível no *code behind*.

```
<StackLayout>
    <TableView...>
        <Button></Button>
    </StackLayout>
```

No arquivo `AgendamentoView.xaml.cs` acrescentaremos:

```
private void Button_Clicked(object sender, EventArgs e)
{
    DisplayAlert("Agendamento", "Nome: " + Nome, "OK");
}
```

A partir daí rodaremos a aplicação, testando as mudanças feitas digitando-se "fulano de tal" no campo referente a nome. Quando clicamos em "Agendar", uma mensagem de confirmação de agendamento aparece na tela. Significa que o `Binding` já está funcionando para as propriedades desta página.

Modificaremos o método `Button_Clicked()` para incluir os outros campos que forem preenchidos pelo usuário neste formulário. Vamos expandir o código, colocando um `@` antes de "Nome: [...]", para indicar que é possível colocar múltiplas linhas nesta `string`, as quais acrescentaremos dentro de um formato (`string.Format`) que será exibido ao usuário na mensagem.

Acrescentaremos também as posições dos parâmetros, e agora começaremos a preencher os parâmetros desta formatação: o primeiro é `Nome`, depois vem `Fone`, `Email`, `DataAgendamento`, que será formatado obedecendo o padrão `dia/mês/ano`, e `HoraAgendamento`.

Lembre-se que o mês fica `MM`, com letras maiúsculas; `mm` em minúsculas se referem a `minutos`!

```
private void Button_Clicked(object sender, EventArgs e)
{
    DisplayAlert("Agendamento",
    string.Format(
    @"Nome: {0}
    Fone: {1}
    E-mail: {2}
    Data Agendamento: {3}"))
```

```
Hora Agendamento:{4}",  
Nome, Fone, Email, DataAgendamento.ToString("dd/MM/yyyy"), HoraAgendamento), "OK");  
}
```

Vamos ver como esta mensagem se comporta na aplicação. Digitaremos "fulano de tal" no campo destinado ao nome, "1234-5678" no de telefone e, no e-mail, "fulano@gmail.com". A data será semana que vem, 9/12/2016 , e o horário será 4:00PM . Feito isto, clicaremos em "Agendar", e uma view é exibida, com todos os dados que acabamos de colocar:



Agora que conseguimos implementar a terceira página, faremos uma breve revisão do que foi visto: aprendemos a utilizar o `TableView` para criar uma tela de cadastro no Xamarin Forms. Nele, temos duas seções, ou `TableSections` , que denominamos "Seus Dados" e "Agendamento".

Para o preenchimento dos dados nos campos de entrada, utilizamos o `EntryCell` , adaptando o teclado por meio da propriedade `Keyboard` para telefone e e-mail, melhorando a experiência do usuário.

Para a parte de agendamento, vimos dois novos controles, `DatePicker` e `TimePicker` , selecionadores de data e hora, respectivamente. Para inseri-los em um `TableView` , tivemos que utilizar o `ViewCell` , que permite controles que normalmente não são possíveis em um `TableView` .

Por fim, aprendemos a programar o evento de um botão para armazenar os dados preenchidos anteriormente, apresentando-os ao usuário a partir de uma mensagem.

Mais adiante, veremos como enviá-los ao servidor, para o cadastramento destes dados. O curso é sobre Xamarin, ou aplicação móvel, mas veremos um pouco sobre o serviço ser acessado pelo Xamarin para envio destes dados. Não veremos a parte sobre servidores, e sim somente a parte de consumo de serviço disponível na internet.

Espero que estejam gostando do curso.