Exercício: Adicionar comportamento à sua página XAML

10 minutos

Você modificou anteriormente o aplicativo Notas para mover o layout da interface do usuário do código C# para XAML. Agora você está pronto para adicionar os seguintes recursos à página:

- Suporte à personalização da cor da fonte e da cor de fundo do rótulo, dos botões e do controle do editor. Dessa forma, é fácil ajustar o aplicativo para torná-lo mais acessível para usuários que exigem uma interface de usuário de alto contraste.
- Ajuste da altura do controle do Editor no Android e no iOS. Ao executar no Windows, esse controle tem largura suficiente para permitir que o usuário insira uma quantidade razoável de texto antes de rolar. Em um Android ou um iPhone, a largura mais estreita resulta na rolagem ocorrendo mais rapidamente, portanto, é vantajoso fornecer mais espaço vertical.

Usar um recurso estático em XAML

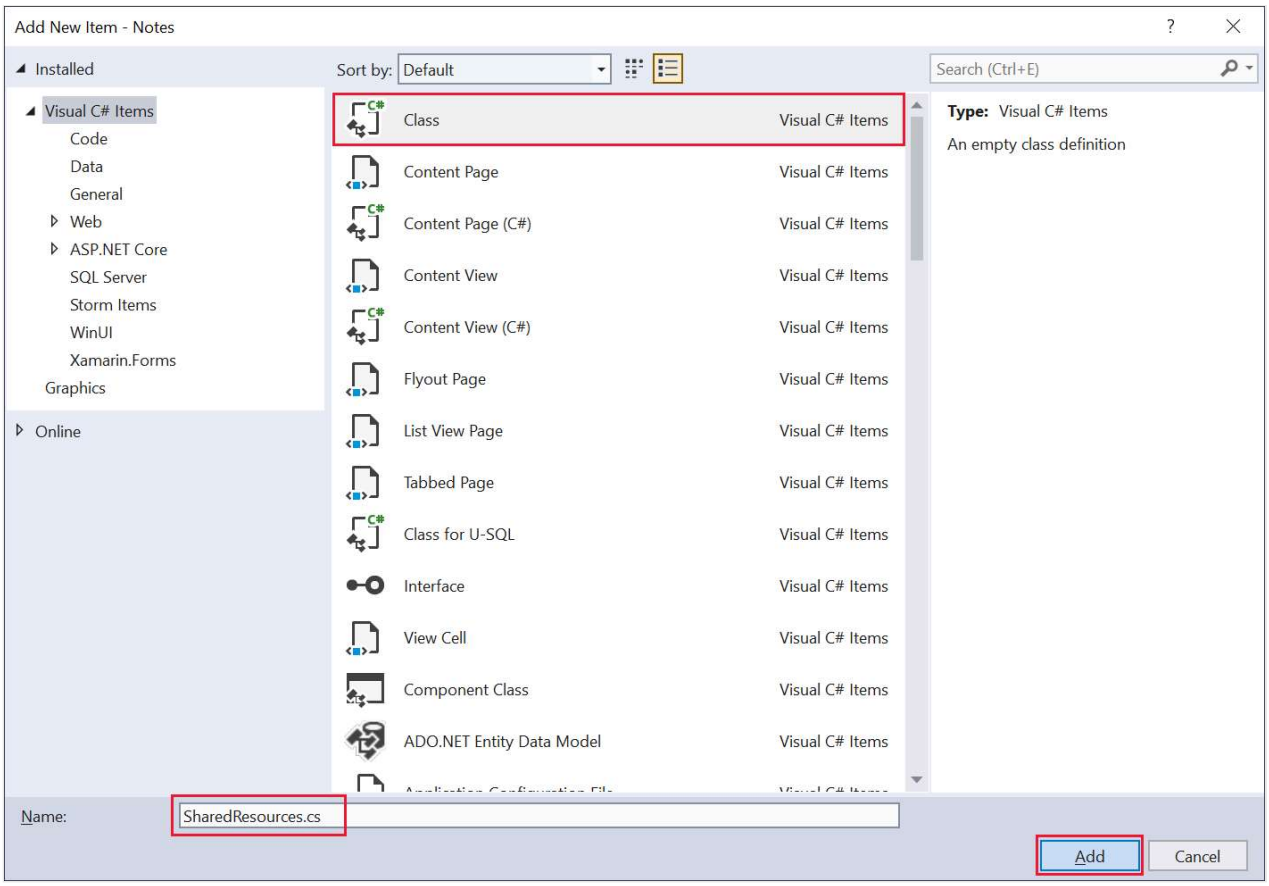
Você criará uma classe estática para manter os valores da cor da fonte e da cor de fundo do aplicativo. Em seguida, você usará a extensão de marcação `x:Static` para ler esses valores da classe e aplicá-los à marcação XAML para os controles na página.

1. No Visual Studio, retorne ao aplicativo Notas que você editou no exercício anterior.

   > **Observação**
   >
   > Uma cópia funcional do aplicativo está disponível na pasta **exercise2** no repositório de exercícios clonado no início do exercício anterior.

2. Na janela **Gerenciador de Soluções**, clique com o botão direito do mouse no projeto **Notas**, selecione **Adicionar**, depois selecione **Classe**.

3. Na caixa de diálogo **Adicionar Novo Item**, verifique se o modelo **Classe** está selecionado. Nomeie o novo arquivo de classe **SharedResources.cs** e selecione **Adicionar**:



4. No arquivo **SharedResources.cs**, substitua as diretivas `using` pelo código a seguir e marque a classe **SharedResources** como `static`:

C#

```
namespace Notes;

static class SharedResources
{

}
```

5. Adicione o `static readonly` campo **FontColor** à classe **SharedResources**. Atualmente, esse campo fornece um valor que corresponde ao azul, mas você pode modificá-lo usando qualquer combinação válida de valores RGB:

C#

```
static class SharedResources
{
    public static readonly Color FontColor = Color.FromRgb(0, 0, 0xFF);
}
```

6. Adicione um segundo `static readonly` campo chamado **BackgroundColor** e defina-o como uma cor de sua escolha:

C#

```
static class SharedResources
{
    ...
    public static readonly Color BackgroundColor = Color.FromRgb(0xFF, 0xF0,
0xAD);
}
```

7. Abra o arquivo **MainPage.xaml**.

8. Adicione a seguinte declaração de namespace XML ao elemento `ContentPage`, antes do atributo `x:Class`. Essa declaração coloca as classes no namespace **Notas** de C# no escopo na página XAML:

XML

```
<ContentPage ...
             xmlns:notes="clr-namespace:Notes"
             x:Class="Notes.MainPage"
             ...>
```

9. Adicione o atributo `TextColor` mostrado no código a seguir ao controle `Label`. Essa marcação usa a extensão de marcação `x:Static` para recuperar os valores armazenados nos campos `static` da classe **SharedResources**:

XML

```
<Label Text="Notes"
    HorizontalOptions="Center"
    FontAttributes="Bold"
    TextColor="{x:Static Member=notes:SharedResources.FontColor}" />
```

10. Use a extensão de marcação `x:Static` para definir os atributos `TextColor` e `BackgroundColor` os controles `Editor` e `Button`. A marcação concluída para seu arquivo **MainPage.xaml** deve ter esta aparência:

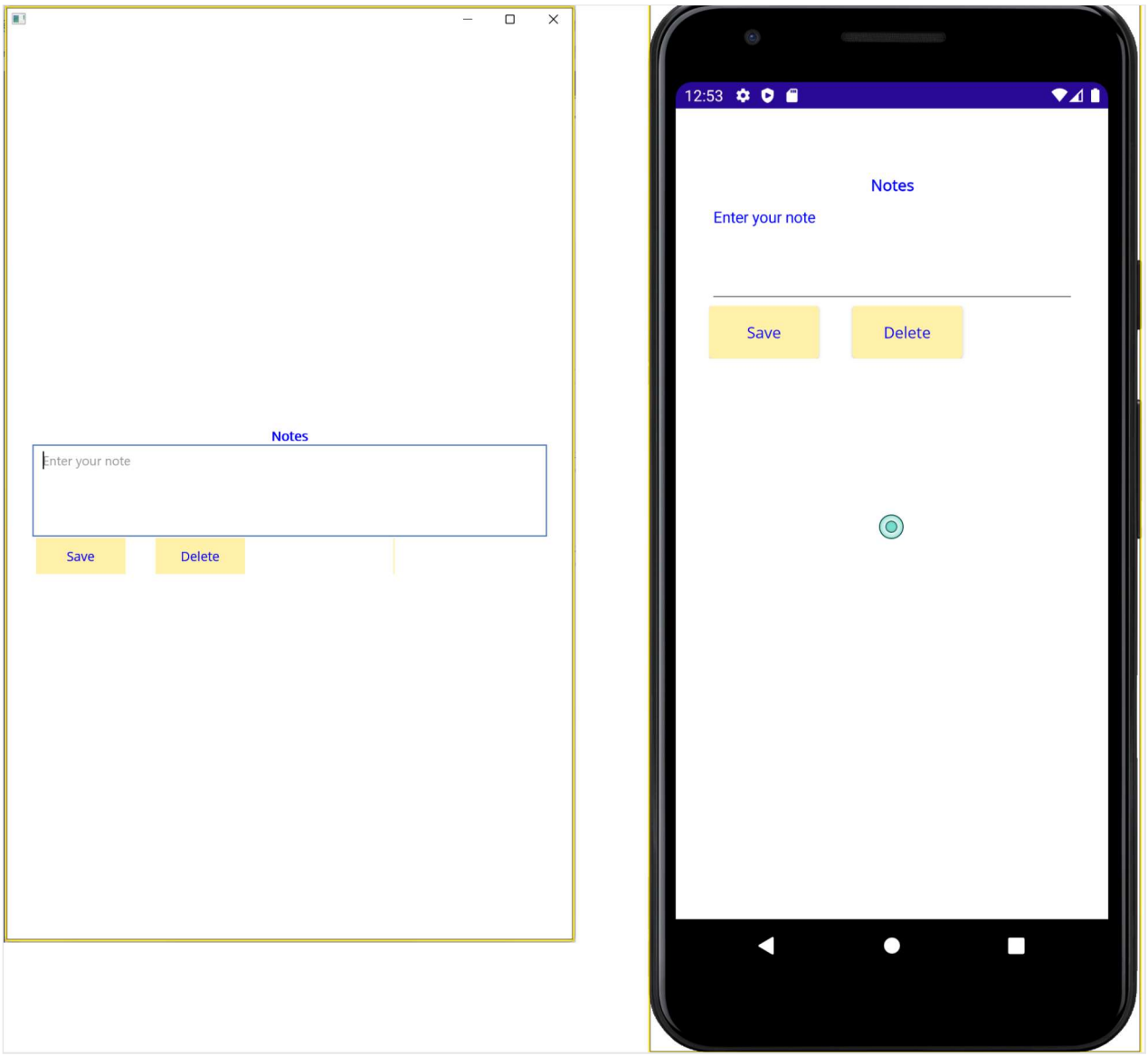
XML

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:notes="clr-namespace:Notes"
             x:Class="Notes.UIPage">

    <VerticalStackLayout Padding="30,60,30,30">
        <Label Text="Notes"
```

```
            HorizontalOptions="Center"
            FontAttributes="Bold"
            TextColor="{x:Static Member=notes:SharedResources.FontColor}" />

        <Editor x:Name="editor"
                Placeholder="Enter your note"
                HeightRequest="100"
                TextColor="{x:Static
Member=notes:SharedResources.FontColor}"/>

        <Grid Grid.Row="2" ColumnDefinitions="Auto,30,Auto">

            <Button Grid.Column="0"
                    Text="Save"
                    WidthRequest="100"
                    TextColor="{x:Static
Member=notes:SharedResources.FontColor}"
                    BackgroundColor="{x:Static
Member=notes:SharedResources.BackgroundColor}"
                    Clicked="OnSaveButtonClicked" />

            <Button Grid.Column="2"
                    Text="Delete"
                    WidthRequest="100"
                    TextColor="{x:Static
Member=notes:SharedResources.FontColor}"
                    BackgroundColor="{x:Static
Member=notes:SharedResources.BackgroundColor}"
                    Clicked="OnDeleteButtonClicked" />
        </Grid>
    </VerticalStackLayout>

</ContentPage>
```

> **Observação**
>
> Esse código XAML contém a repetição da marcação que define as propriedades `TextColor` e `BackgroundColor`. O XAML deixa você definir recursos que podem ser aplicados globalmente em um aplicativo usando um dicionário de recursos no arquivo **App.xaml**. Descrevemos essa técnica em um módulo posterior.

11. Recompile o aplicativo e execute-o usando o Windows. Verifique se as cores correspondem às especificadas na classe **SharedResources**. Se você tiver tempo, também tente executar o aplicativo usando o Android Emulator:



12. Volte para o Visual Studio quando terminar.

Adicionar a personalização de específica da plataforma

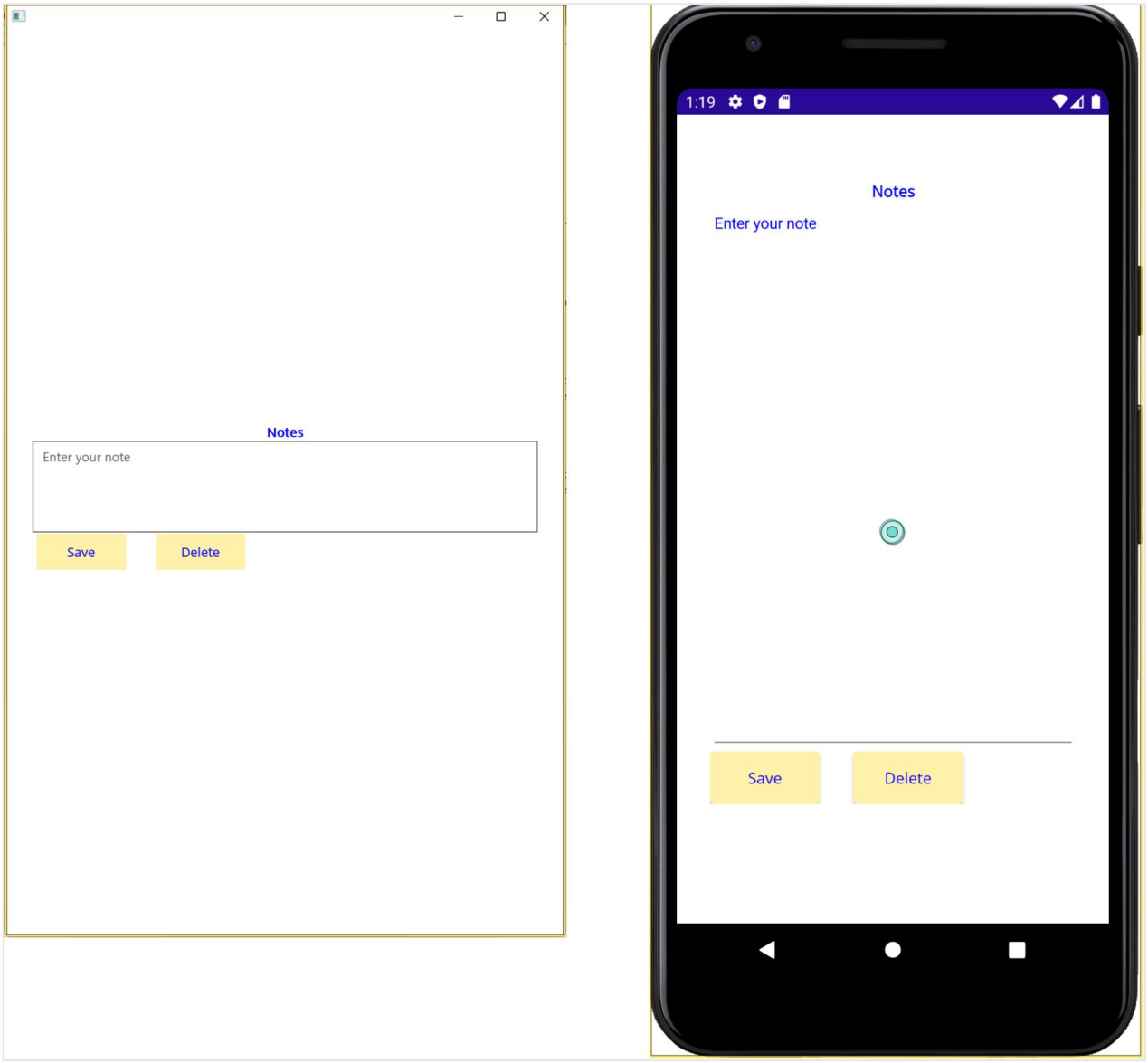
1. Abra o arquivo **MainPage.xaml** no Visual Studio.
2. Localize a definição do controle `Editor` e modifique o valor da propriedade **HeightRequest**, conforme mostrado no seguinte exemplo:

XML

```
<Editor x:Name="editor"
        ...
        HeightRequest="{OnPlatform 100, Android=500, iOS=500}"
        .../>
```

Essa marcação define a altura padrão do controle como 100 unidades, mas aumenta para 500 no Android.

3. Recompile o aplicativo e execute-o usando o Windows e depois o Android. O aplicativo deve ter esta aparência em cada plataforma:



Unidade seguinte: Resumo

Continuar >