

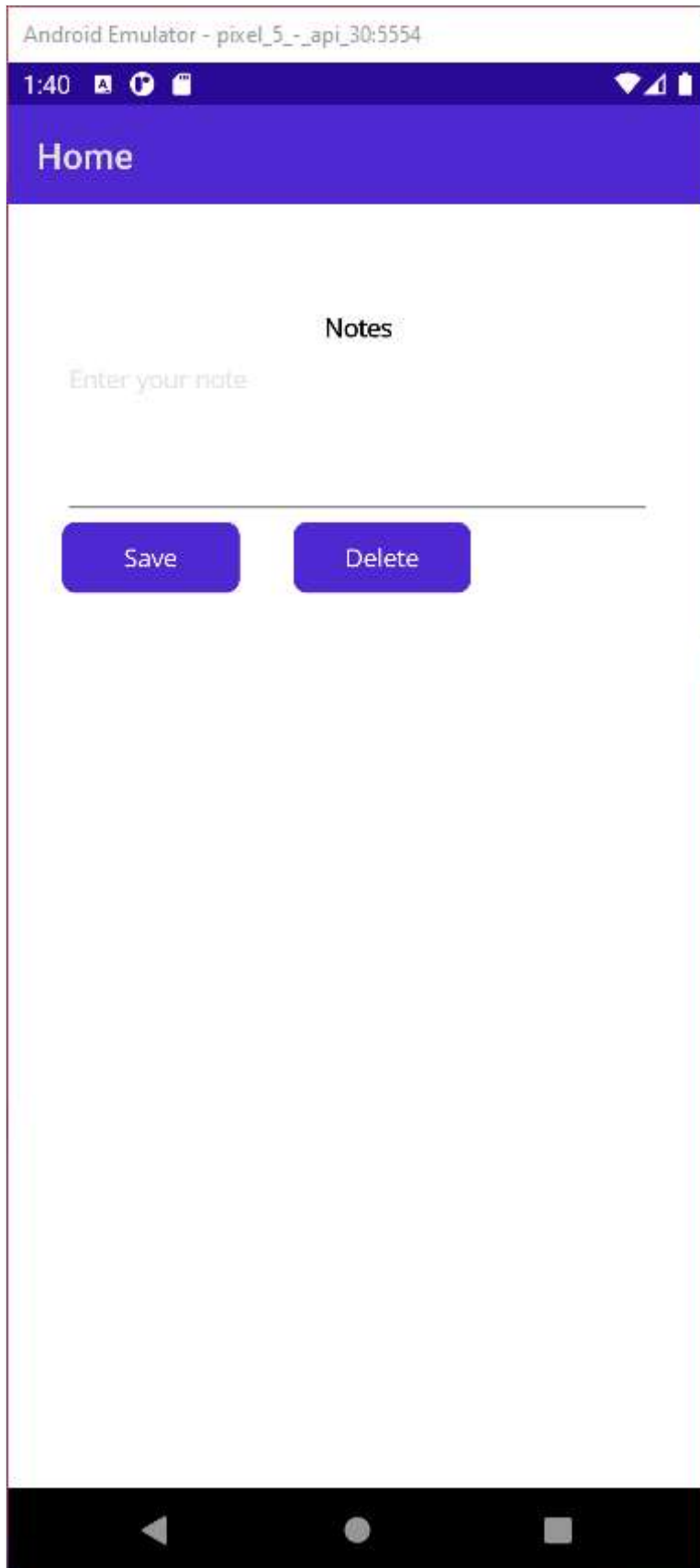
✓ 100 XP 

Exercício: criar sua primeira página em XAML

10 minutos

Os engenheiros da concessionária de energia visitam regularmente os clientes para realizar reparos de dispositivos e outras tarefas de manutenção elétrica. Parte do aplicativo permite que o engenheiro faça anotações sobre a visita. Ele exibe um editor simples que o engenheiro pode usar para inserir informações e salvá-las.

No Android, o aplicativo tem esta aparência:



Você foi solicitado a adicionar mais alguns recursos a esta página. Antes de começar, você precisa entender como a página foi criada, portanto, examine o código-fonte. Observe que a interface do usuário foi criada inteiramente usando o código C#. Embora essa abordagem funcione, ela mescla o código responsável pelo layout com o código que controla a maneira como a interface do usuário funciona. Você percebe que, em pouco tempo, há um risco de

que os dois aspectos do aplicativo sejam bloqueados juntos, dificultando a manutenção futura e possivelmente tornando o aplicativo mais frágil à medida que mais recursos são adicionados. Você decide separar o design da interface do usuário da lógica da interface do usuário extraíndo o código C# que define o layout do aplicativo e substituindo-o por uma página XAML.

Este módulo usa o SDK do .NET 8.0. Verifique se você tem o .NET 8.0 instalado executando o seguinte comando em seu terminal de comando preferencial:

CLI do .NET

```
dotnet --list-sdks
```

Uma saída semelhante ao seguinte exemplo aparece:

Console

```
6.0.317 [C:\Program Files\dotnet\sdk]  
7.0.401 [C:\Program Files\dotnet\sdk]  
8.0.100 [C:\Program Files\dotnet\sdk]
```

Verifique se uma versão que começa com 8 está listada. Se nenhum estiver listado ou o comando não for encontrado, [instale o SDK do .NET 8.0 mais recente](#).

Revisar o aplicativo existente

1. Clone o repositório [GitHub](#) para este exercício localmente no seu computador.

⚠ Observação

Recomendamos clonar ou baixar o conteúdo do exercício para um caminho de pasta curto, como C:\dev, para evitar que os arquivos gerados pelo build excedam o comprimento máximo do caminho.

2. Vá para a pasta **exercise1** na cópia local do repositório.
3. Abra o arquivo de solução **Notes.sln** do Visual Studio nesta pasta ou na pasta no Visual Studio Code.
4. Na janela do **Gerenciador de Soluções**, expanda o projeto do **Notas**, expanda o arquivo **MainPage.xaml** e abra o arquivo **MainPage.xaml.cs**.

5. Revise a classe **MainPage** definida neste arquivo. O construtor contém o seguinte código que cria a interface do usuário:

C#

```
public MainPage()
{
    var notesHeading = new Label() { Text = "Notes", HorizontalOptions =
    LayoutOptions.Center, FontAttributes = FontAttributes.Bold };

    editor = new Editor() { Placeholder = "Enter your note", HeightRequest =
    100 };
    editor.Text = File.Exists(_fileName) ? File.ReadAllText(_fileName) :
    string.Empty;

    var buttonsGrid = new Grid() { HeightRequest = 40.0 };
    buttonsGrid.ColumnDefinitions.Add(new ColumnDefinition() { Width = new
    GridLength(1.0, GridUnitType.Auto) });
    buttonsGrid.ColumnDefinitions.Add(new ColumnDefinition() { Width = new
    GridLength(30.0, GridUnitType.Absolute) });
    buttonsGrid.ColumnDefinitions.Add(new ColumnDefinition() { Width = new
    GridLength(1.0, GridUnitType.Auto) });

    var saveButton = new Button() { WidthRequest = 100, Text = "Save" };
    saveButton.Clicked += OnSaveButtonClicked;
    Grid.SetColumn(saveButton, 0);
    buttonsGrid.Children.Add(saveButton);

    var deleteButton = new Button() { WidthRequest = 100, Text = "Delete" };
    deleteButton.Clicked += OnDeleteButtonClicked;
    Grid.SetColumn(deleteButton, 2);
    buttonsGrid.Children.Add(deleteButton);

    var stackLayout = new VerticalStackLayout
    {
        Padding = new Thickness(30, 60, 30, 30),
        Children = { notesHeading, editor, buttonsGrid }
    };

    this.Content = stackLayout;
}
```

A interface do usuário é composta por um `VerticalStackLayout` contendo um `Label`, `Editor` e uma `Grid` com três colunas. A primeira coluna contém o controle **saveButton**, a segunda é um espaçador e a terceira coluna tem o controle **deleteButton**.

O diagrama a seguir ilustra a estrutura da interface do usuário:



Observe que a classe **MainPage** também contém métodos de manipulação de eventos para os botões e algum código que inicializa o controle `Editor`. Esse código não é diferenciado da definição da interface do usuário.

6. Compile e execute o aplicativo no Windows, só para ver a aparência dele. Para compilar e executar o aplicativo, selecione **F5**.
7. Feche o aplicativo e retorne ao Visual Studio ou ao Visual Studio Code quando terminar.

Criar uma versão XAML da interface do usuário

1. Abra o arquivo **MainPage.xaml**. A marcação nesta página representa uma página de conteúdo MAUI vazia:

XML

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="Notes.MainPage">

  </ContentPage>
```

2. Adicione um controle `VerticalStackLayout` à página de conteúdo:

XML

```
<ContentPage ...>
  <VerticalStackLayout Margin="30,60,30,30">

  </VerticalStackLayout>
</ContentPage>
```

3. Adicione um controle `Label` ao `VerticalStackLayout`. Defina as propriedades **Text**, **HorizontalTextAlignment** e **FontAttributes** desse controle, conforme mostrado abaixo:

XML

```
<ContentPage ...>
  <VerticalStackLayout ...>
    <Label Text="Notes"
           HorizontalOptions="Center"
           FontAttributes="Bold" />
  </VerticalStackLayout>
</ContentPage>
```

4. Adicione um controle `Editor` ao `VerticalStackLayout`:

XML

```
<ContentPage ...>
  <VerticalStackLayout ...>
    <Label .../>

    <Editor x:Name="editor"
            Placeholder="Enter your note"
            HeightRequest="100" />
  </VerticalStackLayout>
</ContentPage>
```

5. Adicione um `Grid` filho ao arquivo `VerticalStackLayout`. Essa `Grid` deve ter três colunas; a primeira e a terceira são dimensionadas automaticamente, enquanto a segunda tem uma largura de 30:

XML

```
<ContentPage ...>
  <VerticalStackLayout ...>
    <Label .../>

    <Editor .../>

    <Grid ColumnDefinitions="Auto, 30, Auto">

      </Grid>
    </VerticalStackLayout>
</ContentPage>
```

6. Adicione um `Button` à primeira coluna da `Grid` filha. Este é o botão **Salvar**:

XML

```
<ContentPage ...>
  <VerticalStackLayout ...>
    <Label .../>

    <Editor .../>

    <Grid ...>
      <Button Grid.Column="0"
        Text="Save"
        WidthRequest="100"
        Clicked="OnSaveButtonClicked" />
    </Grid>
  </VerticalStackLayout>
</ContentPage>
```

7. Adicione outro `Button` à terceira coluna da `Grid` filha. Este é o botão **Excluir**:

XML

```
<ContentPage ...>
  <VerticalStackLayout ...>
    <Label .../>

    <Editor .../>

    <Grid ...>
      <Button ... />

      <Button Grid.Column="2"
        Text="Delete"
        WidthRequest="100"
        Clicked="OnDeleteButtonClicked" />
    </Grid>
  </VerticalStackLayout>
</ContentPage>
```

Remover o código de layout do arquivo code-behind

1. Na janela do **Gerenciador de Soluções**, expanda o nó **MainPage.xaml** e abra o arquivo **MainPage.xaml.cs**.
2. Remova o campo **Editor** da classe **MainPage**.
3. No arquivo **MainPage.xaml.cs**, no construtor **MainPage**, remova todo o código que cria os elementos da interface do usuário e substitua-o por uma chamada para o método **InitializeComponent**. Adicione o código que verifica se o arquivo usado para armazenar

as anotações existe e, se esse for o caso, leia o conteúdo e preencha o campo **Text** do controle **Editor**. O construtor terá esta aparência:

C#

```
public partial class MainPage : ContentPage
{
    string _fileName = Path.Combine(FileSystem.AppDataDirectory,
    "notes.txt");

    public MainPage()
    {
        InitializeComponent();

        if (File.Exists(_fileName))
        {
            editor.Text = File.ReadAllText(_fileName);
        }

        ...
    }
}
```

4. No menu **Compilar**, selecione **Recompilar solução**. Verifique se o aplicativo é compilado sem erros.
5. Execute o aplicativo. Ele deve funcionar exatamente como antes.
6. Se você tiver tempo, implante e execute o aplicativo usando o emulador do Android. A interface do usuário do aplicativo deve ser semelhante à exibida na imagem no início deste exercício.

Unidade seguinte: Extensões de marcação XAML

Continuar >