

Curso de Xamarin.Forms – Macoratti .net

Exercício 1

Crie uma aplicação Xamarin.Forms que acesse e exiba a imagem contida na seguinte URI

https://developer.xamarin.com/demo/IMG_2313.JPG

usando código C#. Defina o aspecto da imagem para preencher a tela do emulador sem distorcer a imagem.

Exercício 2

Repita o exercício 1 desta vez usando apenas código XAML.

Exercício 3

Repita o exercício 2 definindo a largura e altura da imagem igual a 200.

Exercício 4

Crie uma aplicação Xamarin.Forms que exiba um componente **ActivityIndicator** no centro da tela durante a carga da imagem URL : https://developer.xamarin.com/demo/IMG_2138.JPG

Após a carga da imagem a mesma deve ocupar a tela inteira e ocultar o ActivityIndicator.
(Desabilite o cache para a imagem)

Exercício 5

Incorpore o arquivo **paisagem.jpg** em anexo no projeto **Portable** e a seguir acesse e exiba esta imagem centralizada na tela do dispositivo usando código C# e código XAML.

Dicas:

Crie uma pasta no projeto Portable e copie a imagem para esta pasta.

Crie uma pasta para definir a classe que implementa a interface **IMarkupExtension**

Defina o namespace no arquivo XAML.

Exercício 6

Crie uma aplicação Xamarin.Forms para exibir as seguintes imagens locais :

iOS : **Icon-Small-40.png** Android: **icon.png** e WinPhone: **Assets/StoreLogo.png**

usando Código C# e também via código XAML. Centralize as imagens usando um StackLayout.

Curso de Xamarin.Forms – Macoratti .net

Exercício 7

Crie uma aplicação Xamarin.Forms usando código C# que exiba uma imagem centralizada no layout e adicione a essa imagem o evento de toque/clique de forma que ao clicar na imagem sua opacidade seja alterada para que o usuário perceba visualmente que o evento de toque foi acionado. Exiba também uma mensagem de alerta ao usuário usando o método **DisplayAlert()** conforme mostrado abaixo.

Configure a propriedade **NumbersOfTapsRequired** para dois toques.

url da imagem: https://developer.xamarin.com/demo/IMG_2188.JPG

Dica: Você deverá o gesto de toque à view usando a classe **TapGestureRecognizer**, depois tratar evento **Tapped** e adicionar o reconhecedor de toque à coleção **GestureRecognizers**.

DisplayAlert ("Aviso", "Você clicou na imagem", "OK");

Exercício 8

Repita o exercício 7 usando código XAML sem exibir o **DisplayAlert()** e sem configurar a propriedade **NumbersOfTapsRequired**.

url da imagem: https://developer.xamarin.com/demo/IMG_2188.JPG

Exercício 9

Crie uma aplicação Xamarin.Forms que exiba a imagem definida pela url :

<http://www.macoratti.net/Imagens/animais/cao3.jpg>

de forma circular usando o plugin :

- <https://github.com/jamesmontemagno/ImageCirclePlugin>

Dica:

- - Incluir nos projetos Android e iOS , após a inicialização do Xamarin Forms a seguinte linha de comando:
- **Xamarin.Forms.Init();//platform specific init
ImageCircleRenderer.Init();**
- - Incluir o seguinte namespace no arquivo XAML:
- xmlns:controls="clr-namespace:ImageCircle.Forms.Plugin.Abstractions;assembly=ImageCircle.Forms.Plugin.Abstractions"

Curso de Xamarin.Forms – Macoratti .net

Exercício 10

Criando um álbum de fotos com tela de fundo Preta.

Neste exercício vamos acessar e exibir imagens remotas usando a seguinte uri :

<http://lorempixel.com/1920/1080/city/{id}>

Aqui o **id** é um número deverá ficar entre 1 e 10.

Nosso objetivo é exibir as imagens centralizadas na tela e incluir no topo dois botões de comando a 5% do topo da página:

- 1- Botão á esquerda – Texto = **Anterior**, nome do evento **Clicked = Anterior_Clicked**
- 2- Botão à direita – Texto – **Próximo** , nome do evento **Clicked = Proximo_Clicked**
- 3- Definir duas imagens locais para exibir na propriedade **Image** dos dois botões. (Arquivos em anexo: anterior64.jpg e proximo64.jpg)

No centro do topo, e no meio dos botões, inclua um **ActivityIndicator**.

Implemente o código no arquivo code-behind que permita navegar e exibir as imagens exibindo, durante a carga das imagens o componente ActivityIndicator em ação:

Dicas: Ao clicar nos botões você deverá incrementar ou decrementar o id da imagem e carregar a imagem. Deverá verificar no código se o id é menor que 1 ou maior que 10.

Use o layout **AbsoluteLayout** para posicionar os botões no topo.

Para formatar a uri usa a função **String.Format**:

Exemplo: **String.Format("http://lorempixel.com/1920/1080/city/{0}", _Id)**

Ao carregar a imagem desabilite o cache da imagem.

Abaixo temos a figura da aplicação com os botões e a imagem exibida:

Curso de Xamarin.Forms – Macoratti .net

