**Filmes e Séries**

App feito em Xamarin utilizando banco de dados e com recursos Shell

**Prints:**

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

**Criando o projeto:**

* Crie o projeto FilmesSeries

**Instalando o pacote do sqlite:**

* No Gerenciador de Soluções, selecione o projeto FilmesSeries, clique com o botão direito do mouse e selecione Gerenciar Pacotes do NuGet…:
* Procure o pacote: sqlite-net-pcl
* Instale o pacote

**Criando e estruturando as classes e as páginas do projeto**

* Para criar as pastas:
  + No Gerenciador de Soluções, selecione o projeto FilmesSeries, clique com o botão direito do mouse e posicione em Adicionar e depois clique em Nova Pasta.
  + Crie as pastas:
    - Models
    - Views
    - Controls
* Para criar as classes:
  + No Gerenciador de Soluções, selecione a pasta Models, clique com o botão direito do mouse e posicione em Adicionar e depois clique em Nova Classe.
  + Escolha Classe Vazia e Digite o nome da classe.
  + Crie as classes:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

* Para criar as páginas:
  + No Gerenciador de Soluções, selecione o projeto FilmesSeries, clique com o botão direito do mouse e posicione em Adicionar e depois clique em Novo Arquivo.
  + Escolha Forms e depois escolha XAML do ContentPage do Forms
  + Digite a nome e clique em Novo
  + **Crie a página: AppShell**
  + As páginas seguintes devem ser criadas na pasta Views:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

**Importando imagens**

Para utilizar imagens no projeto, precisamos salvá-las nas pastas específicas do iOS e do Android.

**iOS:** Clique com o botão direito em Resources que está no projeto .iOS -> Adicionar-> Arquivos existentes... -> Localize a imagem desejada e clique em adicionar.

**Android:** Clique com o botão direito em Drawable que está no projeto .Android -> Adicionar-> Arquivos existentes... -> Localize a imagem desejada e clique em adicionar.

**Importe as imagens para as pastas mostradas acima**

|  |  |
| --- | --- |
| iOS | Android |
|  |  |

**Classes**

As classes presentes na pasta Models definirão as tabelas e o Banco de Dados, em todas as classes relativas às tabelas a propriedade ID é marcada com os atributos PrimaryKey e AutoIncrement para garantir que cada instância no banco de dados SQLite.NET terá uma ID exclusiva fornecida pelo SQLite.NET.

**Classe Filmes**

* Abra a classe Filmes:

using System;

using SQLite;

namespace FilmesSeries.Models

{

public class Filmes

{

**[PrimaryKey, AutoIncrement]**

**public int IDFilme { get; set; }**

**public string NomeFilme { get; set; }**

**public string GeneroFilme { get; set; }**

**public int DuracaoFilme { get; set; }**

**public string PlataformaFilme { get; set; }**

**public string** **URLCapaFilme { get; set; }**

}

}

**Classe Series**

* Abra a classe Series:

using System;

using SQLite;

namespace FilmesSeries.Models

{

public class Series

{

**[PrimaryKey, AutoIncrement]**

**public int IDSerie { get; set; }**

**public string NomeSerie { get; set; }**

**public string GeneroSerie { get; set; }**

**public int TemporadasSerie { get; set; }**

**public string PlataformaSerie { get; set; }**

**public string URLCapaSerie { get; set; }**

}

}

**Classe Plataformas**

* Abra a classe Plataformas:

using System;

using SQLite;

namespace FilmesSeries.Models

{

public class Plataformas

{

**[PrimaryKey, AutoIncrement]**

**public int IDPlataforma { get; set; }**

**public string NomePlataforma { get; set; }**

**public double** **ValorAssinaturaPlataforma { get; set; }**

**public string** **URLLogoPlataforma { get; set; }**

}

}

**Classe BancoDeDados**

* Abra a classe BancoDeDados

using System;

**using System.Collections.Generic;**

**using System.Threading.Tasks;**

**using SQLite;**

namespace FilmesSeries.Models

{

public class BancoDeDados

{

**readonly SQLiteAsyncConnection banco\_de\_dados;**

**public BancoDeDados(string dbPath)**

**{**

**banco\_de\_dados = new SQLiteAsyncConnection(dbPath);**

**//criação das tabelas**

**banco\_de\_dados.CreateTableAsync<Filmes>().Wait();**

**banco\_de\_dados.CreateTableAsync<Series>().Wait();**

**banco\_de\_dados.CreateTableAsync<Plataformas>().Wait();**

**}**

**// ---------------- Listagem**

**public Task<List<Filmes>> GetFilmes()**

**{**

**return banco\_de\_dados.Table<Filmes>().ToListAsync();**

**}**

**public Task<List<Series>> GetSeries()**

**{**

**return banco\_de\_dados.Table<Series>().ToListAsync();**

**}**

**public Task<List<Plataformas>> GetPlataformas()**

**{**

**return banco\_de\_dados.Table<Plataformas>().ToListAsync();**

**}**

**// ---------------- Searchbar**

**public Task<List<Filmes>> GetFilmeNome(string nome)**

**{**

**return banco\_de\_dados.Table<Filmes>().Where(i => i.NomeFilme.ToLower().Contains(nome.ToLower())).ToListAsync();**

**}**

**public Task<List<Series>> GetSerieNome(string nome)**

**{**

**return banco\_de\_dados.Table<Series>().Where(i => i.NomeSerie.ToLower().Contains(nome.ToLower())).ToListAsync();**

**}**

**public Task<List<Plataformas>> GetPlataformaNome(string nome)**

**{**

**return banco\_de\_dados.Table<Plataformas>().Where(i => i.NomePlataforma.ToLower().Contains(nome.ToLower())).ToListAsync();**

**}**

**// ---------------- Salvar**

**public Task<int> SalvarFilme(Filmes f)**

**{**

**if (f.IDFilme != 0)**

**{**

**return banco\_de\_dados.UpdateAsync(f);**

**}**

**else**

**{**

**return banco\_de\_dados.InsertAsync(f);**

**}**

**}**

**public Task<int> SalvarSerie(Series s)**

**{**

**if (s.IDSerie != 0)**

**{**

**return banco\_de\_dados.UpdateAsync(s);**

**}**

**else**

**{**

**return banco\_de\_dados.InsertAsync(s);**

**}**

**}**

**public Task<int> SalvarPlataforma(Plataformas p)**

**{**

**if (p.IDPlataforma != 0)**

**{**

**return banco\_de\_dados.UpdateAsync(p);**

**}**

**else**

**{**

**return banco\_de\_dados.InsertAsync(p);**

**}**

**}**

**// ---------------- Excluir**

**public Task<int> ApagarFilme(Filmes f)**

**{**

**return banco\_de\_dados.DeleteAsync(f);**

**}**

**public Task<int> ApagarSerie(Series s)**

**{**

**return banco\_de\_dados.DeleteAsync(s);**

**}**

**public Task<int> ApagarPlataforma(Plataformas p)**

**{**

**return banco\_de\_dados.DeleteAsync(p);**

**}**

}

}

* Esta classe contém código para criar o banco de dados e ler, gravar e excluir dados do banco de dados. O código usa APIs SQLite.NET assíncronas que movem operações de banco de dados para threads de segundo plano. Além disso, o construtor BancoDeDados usa o caminho para o arquivo de banco de dados como um argumento. Esse caminho será fornecido pela classe App na próxima etapa.

**PAREI BEM AQUI**

**App.xaml**

* Edite a programação

<?xml version="1.0" encoding="utf-8" ?>

<Application xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.App">

<Application.Resources>

**<ResourceDictionary>**

**<Style x:Key="lblTitulos" TargetType="Label">**

**<Setter Property="VerticalOptions" Value="Center"/>**

**<Setter Property="HorizontalTextAlignment" Value="Center"/>**

**<Setter Property="FontAttributes" Value="Bold"/>**

**<Setter Property="FontSize" Value="16"/>**

**<Setter Property="Margin" Value="10,10,10,0"/>**

**</Style>**

**<Style x:Key="lblTitulosFilmes" TargetType="Label" BasedOn="{StaticResource lblTitulos}">**

**<Setter Property="TextColor" Value="#3D689B"/>**

**</Style>**

**<Style x:Key="lblTitulosSeries" TargetType="Label" BasedOn="{StaticResource lblTitulos}">**

**<Setter Property="TextColor" Value="#588333"/>**

**</Style>**

**<Style x:Key="lblTitulosPlataformas" TargetType="Label" BasedOn="{StaticResource lblTitulos}">**

**<Setter Property="TextColor" Value="DimGray"/>**

**</Style>**

**<Style x:Key="txt" TargetType="Entry">**

**<Setter Property="TextColor" Value="Black"/>**

**<Setter Property="VerticalOptions" Value="Center"/>**

**<Setter Property="HorizontalTextAlignment" Value="Center"/>**

**<Setter Property="FontAttributes" Value="Bold"/>**

**<Setter Property="FontSize" Value="18"/>**

**<Setter Property="BackgroundColor" Value="WhiteSmoke"/>**

**<Setter Property="Margin" Value="15,0,15,0"/>**

**</Style>**

**<Style x:Key="cbb" TargetType="Picker">**

**<Setter Property="TextColor" Value="Black"/>**

**<Setter Property="VerticalOptions" Value="Center"/>**

**<Setter Property="HorizontalTextAlignment" Value="Center"/>**

**<Setter Property="FontAttributes" Value="Bold"/>**

**<Setter Property="FontSize" Value="18"/>**

**<Setter Property="BackgroundColor" Value="WhiteSmoke"/>**

**<Setter Property="Margin" Value="15,0,15,0"/>**

**</Style>**

**<Style x:Key="btn" TargetType="Button">**

**<Setter Property="CornerRadius" Value="10"/>**

**<Setter Property="TextColor" Value="White"/>**

**<Setter Property="FontAttributes" Value="Bold"/>**

**</Style>**

**<Style x:Key="btnFilmes" TargetType="Button" BasedOn="{StaticResource btn}">**

**<Setter Property="BackgroundColor" Value="#3D689B"/>**

**</Style>**

**<Style x:Key="btnSeries" TargetType="Button" BasedOn="{StaticResource btn}">**

**<Setter Property="BackgroundColor" Value="#588333"/>**

**</Style>**

**<Style x:Key="btnPlataformas" TargetType="Button" BasedOn="{StaticResource btn}">**

**<Setter Property="BackgroundColor" Value="DimGray"/>**

**</Style>**

**<Style x:Key="btnApagar" TargetType="Button" BasedOn="{StaticResource btn}">**

**<Setter Property="BackgroundColor" Value="DarkRed"/>**

**</Style>**

**</ResourceDictionary>**

</Application.Resources>

</Application>

**App.xaml.cs**

* Edite a programação

using System;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

**using Filmes.Models;**

**using System.IO;**

namespace FilmesSeries

{

public partial class App : Application

{

**static BancoDeDados banco\_de\_dados;**

**public static BancoDeDados Banco\_de\_dados**

**{**

**get**

**{**

**if (banco\_de\_dados == null)**

**{**

**banco\_de\_dados = new BancoDeDados(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), "BDFilmesSeries.db3"));**

**}**

**return banco\_de\_dados;**

**}**

**}**

public App()

{

InitializeComponent();

**MainPage = new AppShell();**

}

protected override void OnStart()

{

}

protected override void OnSleep()

{

}

protected override void OnResume()

{

}

}

}

* Esse código define uma propriedade Banco\_de\_dados que cria uma nova instância de BancoDeDados, passando o nome de arquivo do banco de dados como o argumento para o construtor BancoDeDados. A vantagem de criar o banco de dados dessa forma é que uma conexão de banco de dados individual criada é mantida aberta enquanto o aplicativo é executado, evitando, portanto, o trabalho de abrir e fechar o arquivo de banco de dados cada vez que uma operação de banco de dados é realizada.

**OBS:** Apague a página MainPage.xaml, pois ela não será mais necessária.

**FlyoutHeader.xaml**

Crie na pasta Controls uma página do tipo Content View chamada FlyoutHeader.xaml que será o header do menu e onde vamos exibir o logo do app.

<?xml version="1.0" encoding="UTF-8" ?>

<ContentView

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="Filmes.Controls.FlyoutHeader"

**HeightRequest="100"**

>

**<Grid BackgroundColor="White">**

**<Image Aspect****="AspectFit" Source="logofilmesseries.png"/>**

**</Grid>**

</ContentView>

**AppShell.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<**Shell**

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:controls="clr-namespace:FilmesSeries.Controls"

xmlns:views="clr-namespace:FilmesSeries.Views"

x:Class="FilmesSeries.AppShell">

**<Shell.Resources>**

**<Style x:Key="BaseStyle"**

**TargetType="Element">**

**<Setter Property="Shell.ForegroundColor"**

**Value="White" />**

**<Setter Property="Shell.TitleColor"**

**Value="White" />**

**<Setter Property="Shell.DisabledColor"**

**Value="#B4FFFFFF" />**

**<Setter Property="Shell.UnselectedColor"**

**Value="#95FFFFFF" />**

**</Style>**

**<Style TargetType="ShellItem"**

**BasedOn="{StaticResource BaseStyle}" />**

**<Style x:Key="FilmesShell"**

**TargetType="Element"**

**BasedOn="{StaticResource BaseStyle}">**

**<Setter Property="Shell.BackgroundColor" Value="#3D689B" />**

**</Style>**

**<Style x:Key="SeriesShell"**

**TargetType="Element"**

**BasedOn="{StaticResource BaseStyle}">**

**<Setter Property="Shell.BackgroundColor" Value="#588333" />**

**</Style>**

**<Style x:Key****="PlataformasShell"**

**TargetType="Element"**

**BasedOn="{StaticResource BaseStyle}">**

**<Setter Property="Shell.BackgroundColor" Value="DimGray" />**

**</Style>**

**</Shell.Resources>**

**<Shell.FlyoutHeader>**

**<controls:FlyoutHeader />**

**</Shell.FlyoutHeader>**

**<FlyoutItem Route="filmesseries"**

**FlyoutDisplayOptions="AsMultipleItems">**

**<ShellContent Route="filmes"**

**Style="{StaticResource FilmesShell}"**

**Title="Filmes"**

**Icon="filmes.png">**

**<views:PagFilmes />**

**</ShellContent>**

**<ShellContent Route="series"**

**Style="{StaticResource SeriesShell}"**

**Title="Séries"**

**Icon="series.png">**

**<views:PagSeries />**

**</ShellContent>**

**<ShellContent Route="plataformas"**

**Style="{StaticResource PlataformasShell}"**

**Title="Plataformas"**

**Icon="plataforma.png">**

**<views:PagPlataformas />**

**</ShellContent>**

**</FlyoutItem>**

**</Shell>**

Esse código define o menu flyout para a exibição das opções que o usuário pode abrir e para utilizar esse tipo de estrutura temos que mudar para Shell.

**AppShell.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Views;

using Xamarin.Forms;

namespace FilmesSeries3M

{

public partial class AppShell : **Shell**

{

**public Dictionary<string, Type> Routes { get; private set; } = new Dictionary<string, Type>();**

**void RegisterRoutes()**

**{**

**Routes.Add("filmesdetalhes", typeof(PagFilmesDetalhes));**

**Routes.Add("seriesdetalhes", typeof(PagSeriesDetalhes));**

**Routes.Add("palataformasdetalhes", typeof(PagPlataformasDetalhes));**

**foreach (var item in Routes)**

**{**

**Routing.RegisterRoute(item.Key, item.Value);**

**}**

**}**

public AppShell()

{

InitializeComponent();

}

}

}

**PagFilmes.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<ContentPage

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.Views.PagFilmes"

**Title="Filmes"**

**BackgroundColor="#3D689B**">

**<ContentPage.ToolbarItems>**

**<ToolbarItem Text="+" Clicked="AdicionarFilme" />**

**</ContentPage.ToolbarItems>**

**<ScrollView >**

**<StackLayout>**

**<SearchBar x:Name="FilmeSearchBar" TextChanged="OnTextChanged" BackgroundColor="WhiteSmoke"/>**

**<ListView x:Name="listView" Margin="2" RowHeight="100" ItemSelected="OnListViewSelected">**

**<ListView.ItemTemplate>**

**<DataTemplate>**

**<ViewCell>**

**<Grid Padding="10"**

**RowDefinitions="Auto,Auto"**

**ColumnDefinitions="70,Auto">**

**<Image**

**Grid.RowSpan="2"**

**Source="{Binding URLCapaFilme}"**

**Aspect="AspectFit"**

**HeightRequest="63"**

**WidthRequest="60"**

**/>**

**<Label**

**Grid.Column="1" Grid.Row="0"**

**Text="{Binding PlataformaFilme}"**

**FontSize="14"**

**FontAttributes="Italic"**

**VerticalOptions="Center"**

**/>**

**<Label**

**Grid.Column="1" Grid.Row="1"**

**Text="{Binding NomeFilme}"**

**FontSize="22"**

**FontAttributes="Bold"**

**VerticalOptions="Start"**

**HorizontalOptions="Center"**

**/>**

**</Grid>**

**</ViewCell>**

**</DataTemplate>**

**</ListView.ItemTemplate>**

**</ListView>**

**</StackLayout>**

**</ScrollView>**

</ContentPage>

**PagFilmes.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Models;

using Xamarin.Forms;

namespace FilmesSeries3M.Views

{

public partial class PagFilmes : ContentPage

{

public PagFilmes()

{

InitializeComponent();

}

**protected async override void OnAppearing()**

**{**

**base.OnAppearing();**

**listView.ItemsSource = await App.Banco\_de\_dados.GetFilmes();**

**}**

**async void AdicionarFilme(object sender, EventArgs e)**

**{**

**await Navigation.PushAsync(new PagFilmesDetalhes { BindingContext = new Filmes() });**

**}**

**async void OnListViewSelected(object sender, SelectedItemChangedEventArgs e)**

**{**

**if (e.SelectedItem != null)**

**{**

**await Navigation.PushAsync(new PagFilmesDetalhes { BindingContext = e.SelectedItem as Filmes });**

**}**

**}**

**async void OnTextChanged(object sender, TextChangedEventArgs e)**

**{**

**var texto = FilmeSearchBar.Text;**

**listView.ItemsSource = await App.Banco\_de\_dados.GetFilmeNome(texto);**

**}**

**}**

}

**PagFilmesDetalhes.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<ContentPage

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.Views.PagFilmesDetalhes"

**Title="Detalhes dos Filmes"**

**BackgroundColor="#3D689B">**

**<ScrollView >**

**<StackLayout Margin="2" BackgroundColor="WhiteSmoke">**

**<Label**

**Text="Nome"**

**Style="{StaticResource lblTitulosFilmes}"**

**/>**

**<Entry**

**x:Name="txtNome"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Gênero"**

**Style="{StaticResource lblTitulosFilmes}"**

**/>**

**<Entry**

**x:Name="txtGenero"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Duração (minutos)"**

**Style="{StaticResource lblTitulosFilmes}"**

**/>**

**<Entry**

**x:Name="txtDuracao"**

**Keyboard="Numeric"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Onde assistir"**

**Style="{StaticResource lblTitulosFilmes}"**

**/>**

**<Picker**

**x:Name="pikerPlataformaFilmes"**

**Style="{StaticResource cbb}"**

**/>**

**<Label**

**Text="Imagem"**

**Style="{StaticResource lblTitulosFilmes}"**

**/>**

**<Grid**

**RowDefinitions="35"**

**ColumnDefinitions="\*,0.2\*">**

**<Entry**

**x:Name="txtURLCapaFilme"**

**Grid.Row="0" Grid.Column="0"**

**Style="{StaticResource txt}"**

**Margin="15,0,0,0"**

**/>**

**<Button**

**Grid.Row="0" Grid.Column="1"**

**ImageSource="refresh.png"**

**BackgroundColor="WhiteSmoke"**

**Clicked="VisualizarCapaFilme"**

**Margin="0,0,15,0"**

**/>**

**</Grid>**

**<Image**

**x:Name="imgFilme"**

**Aspect="AspectFit"**

**HeightRequest="200"**

**Margin="5"**

**/>**

**<Grid Padding="10"**

**RowDefinitions="35"**

**ColumnDefinitions="\*,\*,\*">**

**<Button**

**Grid.Column="0"**

**Text="Editar"**

**Style="{StaticResource btnFilmes}"**

**Clicked="Editar"**

**/>**

**<Button**

**Grid.Column="1"**

**Text="Salvar"**

**Style="{StaticResource btnFilmes}"**

**Clicked="SalvarFilme"**

**/>**

**<Button**

**Grid.Column="2"**

**Text="Excluir"**

**Style="{StaticResource btnApagar}"**

**Clicked="ExcluirFilme"**

**/>**

**</Grid>**

**</StackLayout>**

**</ScrollView>**

</ContentPage>

**PagFilmesDetalhes.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Models;

using Xamarin.Forms;

namespace FilmesSeries3M.Views

{

public partial class PagFilmesDetalhes : ContentPage

{

public PagFilmesDetalhes()

{

**CarregaPlataformas();**

**InitializeComponent();**

**StatusEntry(true);**

}

**async void CarregaPlataformas()**

**{**

**var nomesplataformas = await App.Banco\_de\_dados.GetPlataformas();**

**foreach (var itemnome in nomesplataformas)**

**{**

**pikerPlataformaFilmes.Items.Add(itemnome.NomePlataforma.ToString());**

**}**

**}**

**public void StatusEntry(bool status)**

**{**

**txtNome.IsReadOnly = status;**

**txtGenero.IsReadOnly = status;**

**txtDuracao.IsReadOnly = status;**

**pikerPlataformaFilmes.IsEnabled = !status;**

**txtURLCapaFilme.IsReadOnly = status;**

**}**

**public void Editar(object sender, EventArgs e)**

**{**

**StatusEntry(false);**

**}**

**async void SalvarFilme(object sender, EventArgs e)**

**{**

**var f = (Filmes)BindingContext;**

**f.NomeFilme = txtNome.Text;**

**f.GeneroFilme = txtGenero.Text;**

**f.DuracaoFilme = Convert.ToInt16(txtDuracao.Text);**

**f.PlataformaFilme = pikerPlataformaFilmes.SelectedItem.ToString();**

**f.URLCapaFilme = txtURLCapaFilme.Text;**

**await App.Banco\_de\_dados.SalvarFilme(f);**

**await Navigation.PopAsync();**

**}**

**async void ExcluirFilme(object sender, EventArgs e)**

**{**

**var f = (Filmes)BindingContext;**

**await App.Banco\_de\_dados.ApagarFilme(f);**

**await Navigation.PopAsync();**

**}**

**public void VisualizarCapaFilme(object sender, EventArgs e)**

**{**

**imgFilme.Source = txtURLCapaFilme.Text;**

**}**

**protected override void OnAppearing()**

**{**

**base.OnAppearing();**

**var f = (Filmes)BindingContext;**

**txtNome.Text = f.NomeFilme;**

**txtGenero.Text = f.GeneroFilme;**

**txtDuracao.Text = Convert.ToString(f.DuracaoFilme);**

**pikerPlataformaFilmes.SelectedItem = f.PlataformaFilme;**

**txtURLCapaFilme.Text = f.URLCapaFilme;**

**imgFilme.Source = f.URLCapaFilme;**

**}**

**}**

}

**PagSeries.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<ContentPage

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.Views.PagSeries"

**Title="Séries"**

**BackgroundColor="#588333">**

**<ContentPage.ToolbarItems>**

**<ToolbarItem Text="+" Clicked="AdicionarSerie" />**

**</ContentPage.ToolbarItems>**

**<ScrollView >**

**<StackLayout>**

**<SearchBar x:Name="SerieSearchBar" TextChanged="OnTextChanged" BackgroundColor="WhiteSmoke"/>**

**<ListView x:Name="listView" Margin="2" RowHeight="100" ItemSelected="OnListViewSelected">**

**<ListView.ItemTemplate>**

**<DataTemplate>**

**<ViewCell>**

**<Grid Padding="10"**

**RowDefinitions="Auto,Auto"**

**ColumnDefinitions="70,Auto">**

**<Image**

**Grid.RowSpan="2"**

**Source="{Binding URLCapaSerie}"**

**Aspect="AspectFit"**

**HeightRequest="63"**

**WidthRequest="60"**

**/>**

**<Label**

**Grid.Column="1" Grid.Row="0"**

**Text="{Binding PlataformaSerie}"**

**FontSize="14"**

**FontAttributes="Italic"**

**VerticalOptions="Center"**

**/>**

**<Label**

**Grid.Column="1" Grid.Row="1"**

**Text="{Binding NomeSerie}"**

**FontSize="22"**

**FontAttributes="Bold"**

**VerticalOptions="Start"**

**HorizontalOptions="Center"**

**/>**

**</Grid>**

**</ViewCell>**

**</DataTemplate>**

**</ListView.ItemTemplate>**

**</ListView>**

**</StackLayout>**

**</ScrollView>**

</ContentPage>

**PagSeries.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Models;

using Xamarin.Forms;

namespace FilmesSeries3M.Views

{

public partial class PagSeries : ContentPage

{

public PagSeries()

{

InitializeComponent();

}

**protected async override void OnAppearing()**

**{**

**base.OnAppearing();**

**listView.ItemsSource = await App.Banco\_de\_dados.GetSeries();**

**}**

**async void AdicionarSerie(object sender, EventArgs e)**

**{**

**await Navigation.PushAsync(new PagSeriesDetalhes { BindingContext = new Series() });**

**}**

**async void OnListViewSelected(object sender, SelectedItemChangedEventArgs e)**

**{**

**if (e.SelectedItem != null)**

**{**

**await Navigation.PushAsync(new PagSeriesDetalhes { BindingContext = e.SelectedItem as Series });**

**}**

**}**

**async void OnTextChanged(object sender, TextChangedEventArgs e)**

**{**

**var texto = SerieSearchBar.Text;**

**listView.ItemsSource = await App.Banco\_de\_dados.GetSerieNome(texto);**

**}**

**}**

}

**PagSeriesDetalhes.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<ContentPage

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.Views.PagSeriesDetalhes"

**Title="Detalhes das Séries"**

**BackgroundColor="#588333">**

**<ScrollView >**

**<StackLayout Margin="2" BackgroundColor="WhiteSmoke">**

**<Label**

**Text="Nome"**

**Style="{StaticResource lblTitulosSeries}"**

**/>**

**<Entry**

**x:Name="txtNome"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Gênero"**

**Style="{StaticResource lblTitulosSeries}"**

**/>**

**<Entry**

**x:Name="txtGenero"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Número de Temporadas"**

**Style="{StaticResource lblTitulosSeries}"**

**/>**

**<Entry**

**x:Name="txtTemporadas"**

**Keyboard="Numeric"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Onde assistir"**

**Style="{StaticResource lblTitulosSeries}"**

**/>**

**<Picker**

**x:Name="pikerPlataformaSeries"**

**Style="{StaticResource cbb}"**

**/>**

**<Label**

**Text="Imagem"**

**Style="{StaticResource lblTitulosSeries}"**

**/>**

**<Grid**

**RowDefinitions="35"**

**ColumnDefinitions="\*,0.2\*">**

**<Entry**

**x:Name="txtURLCapaSerie"**

**Grid.Row="0" Grid.Column="0"**

**Style="{StaticResource txt}"**

**Margin="15,0,0,0"**

**/>**

**<Button**

**Grid.Row="0" Grid.Column="1"**

**ImageSource="refresh.png"**

**BackgroundColor="WhiteSmoke"**

**Clicked="VisualizarCapaSerie"**

**Margin="0,0,15,0"**

**/>**

**</Grid>**

**<Image**

**x:Name="imgSerie"**

**Aspect="AspectFit"**

**HeightRequest="200"**

**Margin="5"**

**/>**

**<Grid Padding="10"**

**RowDefinitions="35"**

**ColumnDefinitions="\*,\*,\*">**

**<Button**

**Grid.Column="0"**

**Text="Editar"**

**Style="{StaticResource btnSeries}"**

**Clicked="Editar"**

**/>**

**<Button**

**Grid.Column="1"**

**Text="Salvar"**

**Style="{StaticResource btnSeries}"**

**Clicked="SalvarSerie"**

**/>**

**<Button**

**Grid.Column="2"**

**Text="Excluir"**

**Style="{StaticResource btnApagar}"**

**Clicked="ExcluirSerie"**

**/>**

**</Grid>**

**</StackLayout>**

**</ScrollView>**

</ContentPage>

**PagSeriesDetalhes.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Models;

using Xamarin.Forms;

namespace FilmesSeries3M.Views

{

public partial class PagSeriesDetalhes : ContentPage

{

public PagSeriesDetalhes()

{

**CarregaPlataformas();**

**InitializeComponent();**

**StatusEntry(true);**

}

**async void CarregaPlataformas()**

**{**

**var nomesplataformas = await App.Banco\_de\_dados.GetPlataformas();**

**foreach (var itemnome in nomesplataformas)**

**{**

**pikerPlataformaSeries.Items.Add(itemnome.NomePlataforma.ToString());**

**}**

**}**

**public void StatusEntry(bool status)**

**{**

**txtNome.IsReadOnly = status;**

**txtGenero.IsReadOnly = status;**

**txtTemporadas.IsReadOnly = status;**

**pikerPlataformaSeries.IsEnabled = !status;**

**txtURLCapaSerie.IsReadOnly = status;**

**}**

**public void Editar(object sender, EventArgs e)**

**{**

**StatusEntry(false);**

**}**

**async void SalvarSerie(object sender, EventArgs e)**

**{**

**var s = (Series)BindingContext;**

**s.NomeSerie = txtNome.Text;**

**s.GeneroSerie = txtGenero.Text;**

**s.TemporadasSerie = Convert.ToInt16(txtTemporadas.Text);**

**s.PlataformaSerie = pikerPlataformaSeries.SelectedItem.ToString();**

**s.URLCapaSerie = txtURLCapaSerie.Text;**

**await App.Banco\_de\_dados.SalvarSerie(s);**

**await Navigation.PopAsync();**

**}**

**async void ExcluirSerie(object sender, EventArgs e)**

**{**

**var s = (Series)BindingContext;**

**await App.Banco\_de\_dados.ApagarSerie(s);**

**await Navigation.PopAsync();**

**}**

**public void VisualizarCapaSerie(object sender, EventArgs e)**

**{**

**imgSerie.Source = txtURLCapaSerie.Text;**

**}**

**protected override void OnAppearing()**

**{**

**base.OnAppearing();**

**var s = (Series)BindingContext;**

**txtNome.Text = s.NomeSerie;**

**txtGenero.Text = s.GeneroSerie;**

**txtTemporadas.Text = Convert.ToString(s.TemporadasSerie);**

**pikerPlataformaSeries.SelectedItem = s.PlataformaSerie;**

**txtURLCapaSerie.Text = s.URLCapaSerie;**

**imgSerie.Source = s.URLCapaSerie;**

**}**

**}**

}

**PagPlataformas.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<ContentPage

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.Views.PagPlataformas"

**Title="Plataformas"**

**BackgroundColor="DimGray">**

**<ContentPage.ToolbarItems>**

**<ToolbarItem Text="+" Clicked="AdicionarPlataforma" />**

**</ContentPage.ToolbarItems>**

**<ScrollView >**

**<StackLayout>**

**<SearchBar x:Name="PlataformaSearchBar" TextChanged="OnTextChanged" BackgroundColor="WhiteSmoke"/>**

**<ListView x:Name="listView" Margin="2" RowHeight="100" ItemSelected="OnListViewSelected">**

**<ListView.ItemTemplate>**

**<DataTemplate>**

**<ViewCell>**

**<Grid Padding="10"**

**RowDefinitions="Auto,Auto"**

**ColumnDefinitions="70,Auto">**

**<Image**

**Grid.RowSpan="2"**

**Source="{Binding URLLogoPlataforma}"**

**Aspect="AspectFit"**

**HeightRequest="63"**

**WidthRequest="60"**

**/>**

**<Label**

**Grid.Column="1" Grid.Row="0"**

**Text="{Binding ValorAssinaturaPlataforma}"**

**FontSize="14"**

**FontAttributes="Italic"**

**VerticalOptions="Center"**

**/>**

**<Label**

**Grid.Column="1" Grid.Row="1"**

**Text="{Binding NomePlataforma}"**

**FontSize="22"**

**FontAttributes="Bold"**

**VerticalOptions="Start"**

**HorizontalOptions="Center"**

**/>**

**</Grid>**

**</ViewCell>**

**</DataTemplate>**

**</ListView.ItemTemplate>**

**</ListView>**

**</StackLayout>**

**</ScrollView>**

</ContentPage>

**PagPlataformas.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Models;

using Xamarin.Forms;

namespace FilmesSeries3M.Views

{

public partial class PagPlataformas : ContentPage

{

public PagPlataformas()

{

InitializeComponent();

}

**protected async override void OnAppearing()**

**{**

**base.OnAppearing();**

**listView.ItemsSource = await App.Banco\_de\_dados.GetPlataformas();**

**}**

**async void AdicionarPlataforma(object sender, EventArgs e)**

**{**

**await Navigation.PushAsync(new PagPlataformasDetalhes { BindingContext = new Plataformas() });**

**}**

**async void OnListViewSelected(object sender, SelectedItemChangedEventArgs e)**

**{**

**if (e.SelectedItem != null)**

**{**

**await Navigation.PushAsync(new PagPlataformasDetalhes { BindingContext = e.SelectedItem as Plataformas });**

**}**

**}**

**async void OnTextChanged(object sender, TextChangedEventArgs e)**

**{**

**var texto = PlataformaSearchBar.Text;**

**listView.ItemsSource = await App.Banco\_de\_dados.GetPlataformaNome(texto);**

**}**

}

}

**PagPlataformasDetalhes.xaml**

* Edite a programação

<?xml version="1.0" encoding="UTF-8" ?>

<ContentPage

xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="FilmesSeries3M.Views.PagPlataformasDetalhes"

**Title="Plataformas"**

**BackgroundColor="DimGray">**

**<ScrollView BackgroundColor="DimGray">**

**<StackLayout Margin="1" BackgroundColor="WhiteSmoke">**

**<Label**

**Text="Nome"**

**Style="{StaticResource lblTitulosPlataformas}"**

**/>**

**<Entry**

**x:Name="txtNome"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Valor da assinatura mensal"**

**Style="{StaticResource lblTitulosPlataformas}"**

**/>**

**<Entry**

**x:Name="txtAssinatura"**

**Keyboard="Numeric"**

**Style="{StaticResource txt}"**

**/>**

**<Label**

**Text="Imagem"**

**Style="{StaticResource lblTitulosPlataformas}"**

**/>**

**<Grid**

**RowDefinitions="35"**

**ColumnDefinitions="\*,0.2\*">**

**<Entry**

**x:Name="txtURLLogoPlataforma"**

**Grid.Row="0" Grid.Column="0"**

**Style="{StaticResource txt}"**

**Margin="15,0,0,0"**

**/>**

**<Button**

**Grid.Row="0" Grid.Column="1"**

**ImageSource="refresh.png"**

**BackgroundColor="WhiteSmoke"**

**Clicked="VisualizarLogoPlataforma"**

**Margin="0,0,15,0"**

**/>**

**</Grid>**

**<Image**

**x:Name="imgPlataforma"**

**Aspect="AspectFit"**

**HeightRequest="200"**

**Margin="5"**

**/>**

**<Grid Padding="10"**

**RowDefinitions="35"**

**ColumnDefinitions="\*,\*,\*">**

**<Button**

**Grid.Column="0"**

**Text="Editar"**

**Style="{StaticResource btnPlataformas}"**

**Clicked="Editar"**

**/>**

**<Button**

**Grid.Column="1"**

**Text="Salvar"**

**Style="{StaticResource btnPlataformas}"**

**Clicked="SalvarPlataforma"**

**/>**

**<Button**

**Grid.Column="2"**

**Text="Excluir"**

**Style="{StaticResource btnApagar}"**

**Clicked="ExcluirPlataforma"**

**/>**

**</Grid>**

**</StackLayout>**

**</ScrollView>**

</ContentPage>

**PagPlataformasDetalhes.xaml.cs**

* Edite a programação

using System;

using System.Collections.Generic;

using FilmesSeries3M.Models;

using Xamarin.Forms;

namespace FilmesSeries3M.Views

{

public partial class PagPlataformasDetalhes : ContentPage

{

public PagPlataformasDetalhes()

{

InitializeComponent();

**StatusEntry(true);**

}

**public void StatusEntry(bool status)**

**{**

**txtNome.IsReadOnly = status;**

**txtAssinatura.IsReadOnly = status;**

**txtURLLogoPlataforma.IsReadOnly = status;**

**}**

**public void Editar(object sender, EventArgs e)**

**{**

**StatusEntry(false);**

**}**

**async void SalvarPlataforma(object sender, EventArgs e)**

**{**

**var p = (Plataformas)BindingContext;**

**p.NomePlataforma = txtNome.Text;**

**p.ValorAssinaturaPlataforma = Convert.ToDouble(txtAssinatura.Text);**

**p.URLLogoPlataforma = txtURLLogoPlataforma.Text;**

**await App.Banco\_de\_dados.SalvarPlataforma(p);**

**await Navigation.PopAsync();**

**}**

**async void ExcluirPlataforma(object sender, EventArgs e)**

**{**

**var p = (Plataformas)BindingContext;**

**await App.Banco\_de\_dados.ApagarPlataforma(p);**

**await Navigation.PopAsync();**

**}**

**public void VisualizarLogoPlataforma(object sender, EventArgs e)**

**{**

**imgPlataforma.Source = txtURLLogoPlataforma.Text;**

**}**

**protected override void OnAppearing()**

**{**

**base.OnAppearing();**

**//Esse código armazena uma instância c, que representa um único filme, no BindingContext da página.**

**var p = (Plataformas)BindingContext;**

**txtNome.Text = p.NomePlataforma;**

**txtAssinatura.Text = Convert.ToString(p.ValorAssinaturaPlataforma);**

**txtURLLogoPlataforma.Text = p.URLLogoPlataforma;**

**imgPlataforma.Source = p.URLLogoPlataforma;**

**}**

}

}