

FIA/P GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

Xamarin

PROFº. FLÁVIO MORENI

•Na aula anterior

- Pendências
- Navegação
- Formulário
- Binding (Conceito)

•Aula de Hoje

- Binding (programando)

Formulário

(Binding)

Xamarin

- **Binding** (ligação, conexão, amarração)
 - Permite ligar dados de diversas fontes sem que seja necessária a utilização de código, como exemplo de Data Binding pode ser a ligação da propriedade **Text** de uma **TextBox** a um atributo de um objeto Model.



- **BindingContext**

- É a propriedade de um objeto **BindableObject** o qual é possível atribuir propriedades que serão ligadas pelo conceito de **Binding**.

Code .cs

```
var label = new Label ();  
label.BindingContext = new {Name = "John Doe", Company = "Xa"};  
label.SetBinding (Label.TextProperty, "Name");
```

Xaml

```
<ContentPage.BindingContext>  
    <exemploModels:SaldoModelViewModel/>  
</ContentPage.BindingContext>
```

Exercício 1

Binding do Investidor

Xamarin

- Dados do Investidor

View



A mobile application interface for entering investor data. The screen is titled "Dados Investidor". It contains several input fields: "Nome Investidor:" with a text box containing "Nome Investidor", "CPF:" with a text box containing "CPF", "e-mail:" with a text box containing "nome@mail.com", and "Telefone" with a text box containing "(11)99999-8888". Below these is a "Patrimônio:" section with a horizontal slider. Underneath is a "Perfil Investidor" section with a dropdown menu showing "Moderado". At the bottom right is a "Gravar" button. A blue arrow points from the "Nome Investidor" text box to the corresponding property in the Model class.

Model

```
public class InvestidorModel
{
    public int IdInvestidor { get; set; }
    public String NomeInvestidor { get; set; }
    public String Cpf { get; set; }
    public String Email { get; set; }
    public String Telefone { get; set; }
    public String Perfil { get; set; }
    public double ValorPatrimonio { get; set; }
    public DateTime DataNascimento { get; set; }

    public IList<string> perfis { get; set; }
}
```



Xamarin

- **Passo 1** - Definir qual é o objeto (**Model**) que ficará associado ao contexto da nossa página (*DadosInvestidorPage*) de investidor.

```
➡ public InvestidorModel investidoModel { get; set; }  
  
public DadosInvestidorPage()  
{  
    InitializeComponent();  
    ➡ investidoModel = new Business.InvestidorBusiness().Consultar(IdUsuario);  
    ➡ BindingContext = investidoModel;  
    ButtonGravar.Clicked += ButtonGravarClicked;  
}
```

Xamarin

- **Passo 2** - Associar os componentes (apenas os **Entry**) visuais ao modelo do ***BindingContext***.




```
<Label Text="Nome:"></Label>
<Entry x:Name="EntryNome" Text="{Binding NomeInvestidor}" Placeholder="Nome do Investidor"></Entry>
<Label Text="Cpf:"></Label>
<Entry x:Name="EntryCpf" Text="{Binding Cpf}" Placeholder="9999999999"></Entry>
<Label Text="e-mail:"></Label>
<Entry x:Name="EntryEmail" Text="{Binding Email}" Placeholder="nome@dominio.com.br"></Entry>
<Label Text="Telefone:"></Label>
<Entry x:Name="EntryTelefone" Text="{Binding Telefone}" Placeholder="11 99999-9999"></Entry>
```

Xamarin

• Passo 3 - Binding Mode

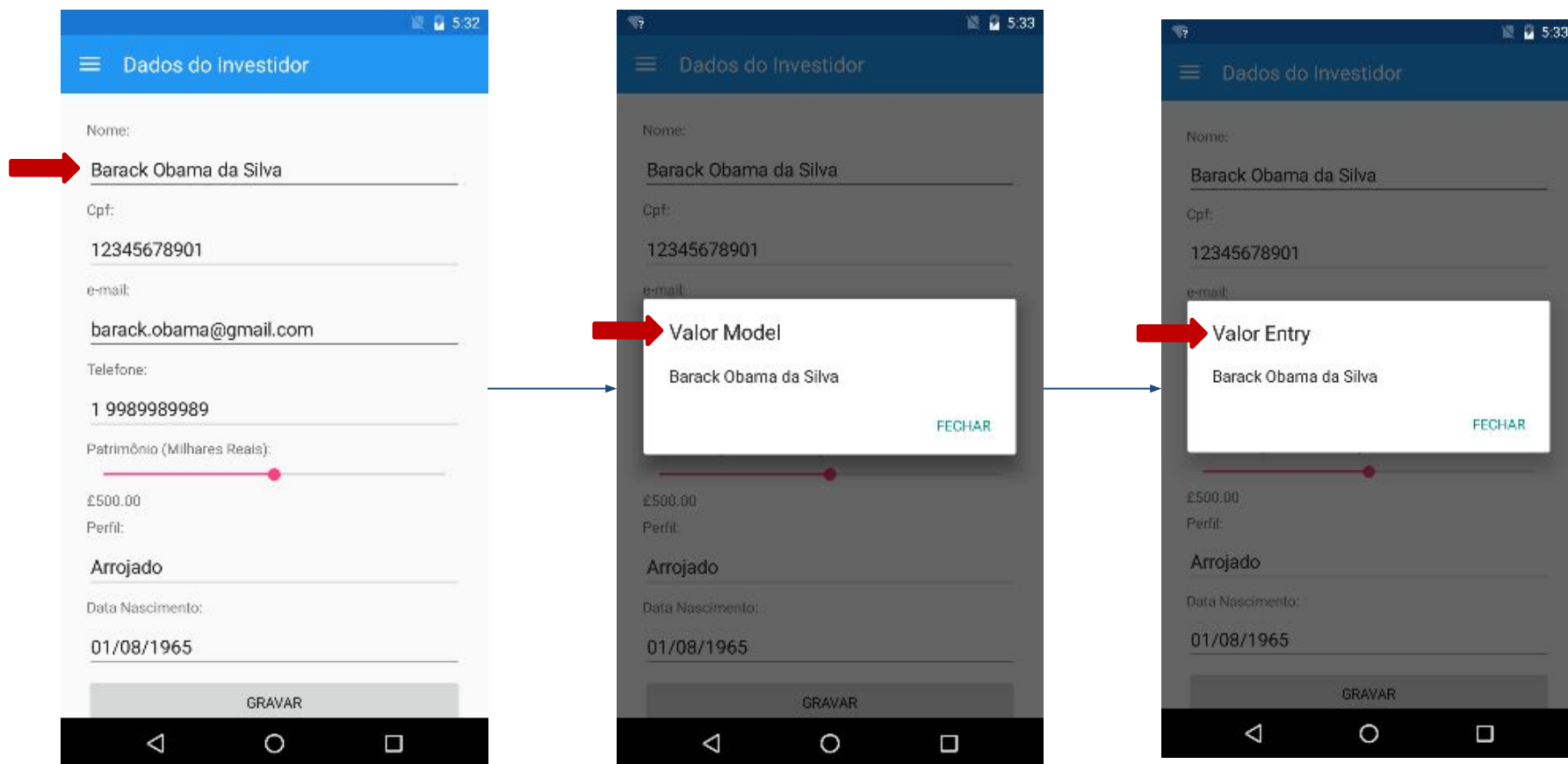
- Adicionar o binding mode “**twoway**” para propagar as alterações efetuadas nos objetos ligados pelo binding.

```
<Label Text="Nome:"></Label>
<Entry x:Name="EntryNome" Text="{Binding NomeInvestidor, Mode=TwoWay}" Placeholder="Nome do investidor" />
<Label Text="Cpf:"></Label>
<Entry x:Name="EntryCpf" Text="{Binding Cpf, Mode=TwoWay}" Placeholder="99999999" />
<Label Text="e-mail:"></Label>
<Entry x:Name="EntryEmail" Text="{Binding Email, Mode=TwoWay}" Placeholder="e-mail" />
<Label Text="Telefone:"></Label>
<Entry x:Name="EntryTelefone" Text="{Binding Telefone, Mode=TwoWay}" Placeholder="Telefone" />
```



Xamarin

• Passo 4 - Testando



Xamarin

- **Passo 5 - DatePicker e Slider**
 - DatePicker

```
<Label Text="Data Nascimento:"></Label>  
<DatePicker x:Name="EntryDataNascimento" Date="{Binding DataNascimento}">
```



- Slider e Label

```
<Label Text="Patrimônio (Milhares Reais):"></Label>  
→ <Slider x:Name="SliPatrimonio" Minimum="0" Maximum="1000"  
Value="{Binding ValorPatrimonio, Mode=TwoWay}"></Slider>
```

```
<Label x:Name="LabelPatrimonio" Text="{Binding Value, Mode=TwoWay}"  
BindingContext="{x:Reference SliPatrimonio}"></Label>
```





Xamarin

- **Passo 6** - Componente **Picker** não aceita **Binding**
 - **Solução 1** - O componente **Picker** não é um componente que aceita **Binding**. Assim para recuperar um elemento selecionado pelo usuário ou adicionar elementos na lista é preciso acessar suas propriedades por meio de código .cs.
 - **Solução 2** - Utilizar um componente customizado com o nome de “**BindablePicker**”. Esse componente foi adicionado no projeto na pasta **Views\Controls**.



Xamarin

- **Passo 6 - Usando o BindablePicker**

- Importar o pacote Views.Controls

```
xmlns:Controls="clr-namespace:FiapInvest.Views.Controls"
```

- Usando o **BindablePicker**

```
<Label Text="Perfil:"></Label>  
<Controls:BindablePicker x:Name="PickerPerfil"  
    ➡ItemsSource="{Binding perfis}"  
    ➡SelectedItem="{Binding Perfil, Mode=TwoWay}" />
```

Observação: a propriedade **perfis** não foi carregada no model do **Investidor**. Precisamos incluir uma linha em nossa classe **InvestidorService** para carregar esse atributo

Xamarin

- **Passo 7** - Carregando os perfis

```
public InvestidorModel Consultar(int idInvest)
{
    // Mock para Consultar Investidor
    InvestidorModel Investidor = new InvestidorModel()
    {
        IdInvestidor = 1,
        NomeInvestidor = "Barack Obama",
        Cpf = "12345678901",
        Email = "barack.obama@gmail.com",
        DataNascimento = new System.DateTime(1965, 8, 1),
        Perfil = "Arrojado",
        ValorPatrimonio = 500,
        Telefone = "1 9989989989"
    };

    ➔ Investidor.perfis = this.ListarPerfis();
    return Investidor;
}
```



- Hands On - Codificando

A screenshot of a mobile application interface. At the top, there is a blue header bar with a white hamburger menu icon and the text 'Dados do Investidor'. Below the header, the form contains several input fields: 'Nome:' with a text input containing 'Nome do Investidor'; 'Cpf:' with a text input containing '99999999999'; 'e-mail:' with a text input containing 'nome@dominio.com.br'; 'Telefone:' with a text input containing '11 99999-9999'; 'Patrimônio (Milhares):' with a range slider; 'Perfil:' with a dropdown menu showing 'Selecione o Perfil'; and 'Data Nascimento:' with a date picker showing '16/05/2017'. At the bottom of the form is a grey button labeled 'GRAVAR'. The entire form is set against a light grey background with a subtle grid. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Finalizar o código do botão **gravar**, enviando os dados da página para as camadas **Business e Services**.





- **Na aula de hoje**

- Binding (programando)

- **Próxima Aula**

- Prova Semestral



Copyright © 2017 Profº. Flávio Moreni

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).