



Neste artigo vou mostrar como passar informações entre as páginas de uma aplicação Xamarin Forms usando o VS 2015 e a linguagem C#: Passando objetos complexos entre páginas.

No [artigo anterior](#) apresentei os conceitos da navegação hierárquica no Xamarin Forms e como passar informações entre páginas.



Curso de Xamarin Forms Vídeo Aulas

Desenvolva para Android, iOS e Windows Phone

Este artigo mostra como passar informações de objetos complexos entre páginas e neste caso, como no abordado no artigo anterior, podemos usar duas formas de passar informações:

- [Passar as informações via construtor](#)
- [Passar as informações via BindingContext](#)

Vamos ver na prática dada uma dessas abordagens.

Recursos usados:

- [Visual Studio Community 2015](#)

Nota: Baixe e use a versão Community 2015 do VS ela é grátis e é equivalente a versão Professional.

Criando um projeto no VS 2015 com Xamarin

Abra o [Visual Studio Community 2015](#) e clique em **New Project**;

Selecione Visual C#, o template **Cross Platform** e a seguir **Blank App (Xamarin.Forms Portable)**;

Informe o nome **XF_Navegacao2** e clique no botão OK;

Ao clicar no botão OK, será criada uma solução contendo 4 projetos. *(Dependendo do seu ambiente pode haver alguma variação nos projetos.)*

O projeto comum(projeto Portable) possui a classe **App.cs** que irá conter o código compartilhado e que vamos usar neste artigo.

Criando a classe Contato

Vamos criar a classe Contato que será o nosso modelo de domínio e que conterà as informações que desejamos tratar.

No menu **Project** clique em **Add Class** e informe o nome **Contato**.

A seguir inclua o código abaixo nesta classe:

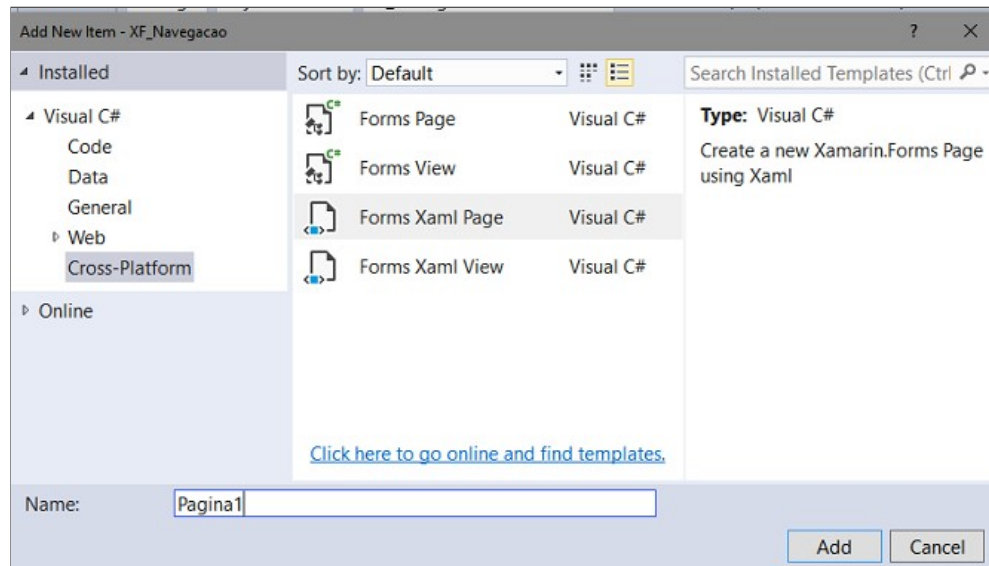
```
public class Contato
{
    public string Nome { get; set; }
    public int Idade { get; set; }
    public string Cargo { get; set; }
    public string Pais { get; set; }
}
```

Criando as páginas no projeto Portable

Vamos criar 2 páginas no projeto Portable : **Pagina1.xaml e Pagina2.xaml**.

Selecione o projeto Portable e no menu **Project** clique em **Add New Item**;

Clique em **Cross Platform** e selecione **Forms Xaml Page** informando o nome **Pagina1**;



Repita o procedimento e crie a página **Pagina2.xaml**.

Antes de continuar abra o arquivo **App.cs** e defina página raiz da pilha de navegação:

```
public App()
{
    MainPage = new NavigationPage(new Pagina1());
}
```

Na página **Pagina1.xaml** defina o seguinte código XAML :

```
<?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="XF_Navegacao2.Pagina1"
    Title="Página 1">

    <StackLayout>
        <Entry x:Name="txtNome" Placeholder="Informe o nome" />
        <Entry x:Name="txtIdade" Placeholder="Informe a idade" />
        <Entry x:Name="txtCargo" Placeholder="Informe o cargo" />
        <Entry x:Name="txtPais" Placeholder="Informe o país" />
        <Button x:Name="btnEnviar" Text="Enviar" Clicked="btnEnviar_Clicked" />
    </StackLayout>
</ContentPage>
```

Nesta página temos uma 4 views **Entry** definidas com o nome **txtNome,txtIdade, txtCargo e txtPais** onde o usuário vai informar as respectivas informações.

Temos também definido um evento **Click** do botão de comando **btnEnviar_Clicked**.

Passando informações para a outra página usando o BindingContext

Ao clicar no botão **Avançar** o evento **Clicked** será disparado e ele deverá navegar para página **Pagina2** e passar as informações digitadas para essa página.

Para fazer isso inclua o evento **Clicked**, no arquivo code-behind **Pagina1.xaml.cs** o seguinte código:

```
private async void btnEnviar_Clicked(object sender, EventArgs e)
{
    var contato = new Contato
    {
        Nome = txtNome.Text,
        Idade = Convert.ToInt32(txtIdade.Text),
        Cargo = txtCargo.Text,
        Pais = txtPais.Text
    };

    var pagina2 = new Pagina2();
    pagina2.BindingContext = contato;
    await Navigation.PushAsync(pagina2);
}
```

Observe que estamos criando uma instância da classe **Contato** e preenchendo os dados do objeto **contato** com as informações recebidas na view Entry.

A seguir criamos uma instância da página **Pagina2** e atribuímos ao seu **BindingContext** o objeto **contato** com as informações preenchidas.

A seguir usando o método **PushAsync()** navegamos para a página **Pagina1**.

Agora vamos definir o código da página **Pagina2** iniciando com o código XAML:

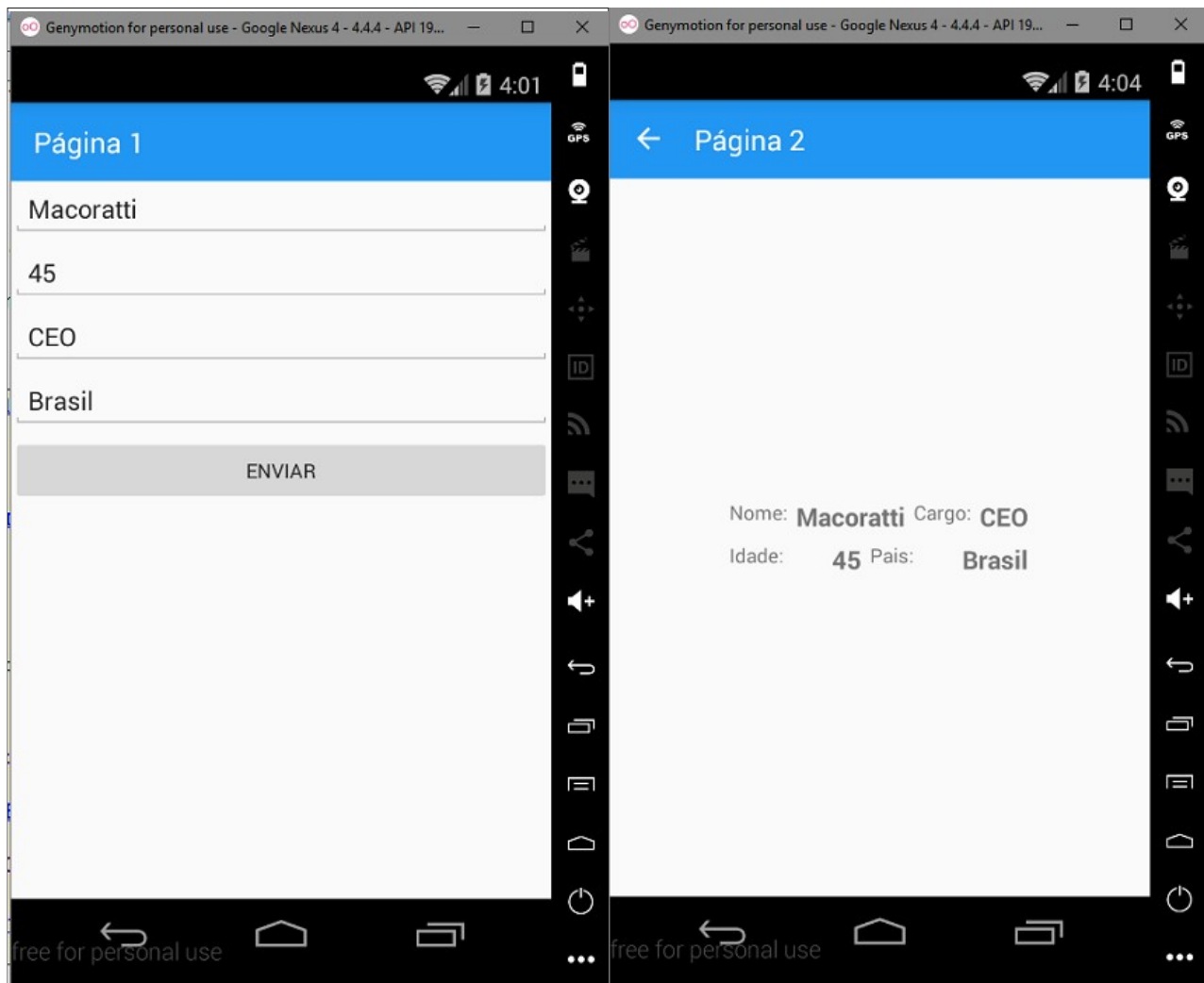
```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="XF_Navegacao2.Pagina2"
    Title="Página 2">

    <StackLayout HorizontalOptions="Center" VerticalOptions="Center">
        <StackLayout Orientation="Horizontal">
            <Label Text="Nome:" HorizontalOptions="FillAndExpand" />
            <Label Text="{Binding Nome}" FontSize="Medium" FontAttributes="Bold" />
            <Label Text="Cargo:" HorizontalOptions="FillAndExpand" />
            <Label Text="{Binding Cargo}" FontSize="Medium" FontAttributes="Bold" />
        </StackLayout>
        <StackLayout Orientation="Horizontal">
            <Label Text="Idade:" HorizontalOptions="FillAndExpand" />
            <Label Text="{Binding Idade}" FontSize="Medium" FontAttributes="Bold" />
            <Label Text="Pais:" HorizontalOptions="FillAndExpand" />
            <Label Text="{Binding Pais}" FontSize="Medium" FontAttributes="Bold" />
        </StackLayout>
    </StackLayout>

</ContentPage>
```

Nesta página estamos usando as informações do **BindingContext** e exibindo as informações na página.

Executando o projeto iremos obter:



Note que a propriedade **Title** das páginas define o título no cabeçalho da página e que o botão de retorno da **Página2** é exibido permitindo ao usuário retornar à página anterior.

Poderíamos ter obtido o mesmo resultado passando as informações via construtor como fizemos no artigo anterior.

Pegue o código das páginas aqui: [📁 XF_Navegacao2.zip](#)

Em outro artigo vou mostrar como passar informações entre a página Modal e a página que a chamou.

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Vídeo Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) **NEW**

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#)

Gostou ?  [Compartilhe no Facebook](#)  [Compartilhe no Twitter](#)

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Seção C# do site Macoratti.net](#)
- [Super DVD C#](#)
- [Super DVD Visual Basic](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [An Introduction to Xamarin.Forms - Xamarin](#)
- <https://www.visualstudio.com/pt-br/features/xamarin-vs.aspx>
- <https://xamarin.com/starter>
- [Xamarim Studio - Desenvolvimento Multiplataforma com C# \(Android, iOS e Windows\)](#)
- [Xamarim - Criando Apps com o Visual Studio e C#](#)
- https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_binding_basics/

[José Carlos Macoratti](#)