



Neste artigo vou mostrar como passar informações entre as páginas de uma aplicação Xamarin Forms usando o [VS 2015](#) e a [linguagem C#](#).

O Xamarin.Forms fornece várias experiências de navegação entre páginas, dependendo do tipo de página que estamos usando.



Curso de Xamarin Forms Vídeo Aulas

Desenvolva para Android, iOS e Windows Phone

De forma geral a navegação entre páginas no Xamarin Forms pode ser definida de duas formas:

- 1- **Hierárquico** - Permite ao usuário se mover para baixo em uma pilha de páginas e depois subir novamente pelos níveis, usando o botão UP ou Back.
- 2- **Modal** - É uma tela de interrupção que requer uma ação do usuário, e pode ser descartada por um botão [Cancel](#).
Exemplos: *Alertas, avisos, notificações, caixas de diálogo, etc.*

Na navegação hierárquica usamos a classe **NavigationPage** para navegar através de uma pilha de **ContentPages** e usamos os seguintes métodos da propriedade **Navigation** da página atual:

- **PushAsync**(new pagina) - navega para página indicada;
- **PopAsync**() - retorna para página anterior na pilha;

Além desses métodos, que são os mais utilizados, a propriedade **Navigation** possui ainda os seguintes métodos que podemos usar na navegação hierárquica:

- 1 - **PopToRootAsync** que extrai todas as páginas da pilha e acessa a página raiz. Exemplo :
Navigation.PopToRootAsync();
- 2 - **RemovePage** – remove da pilha a página especificada. Exemplo: **Navigation.RemovePage(pagina)**
- 3 - **InsertPageBefore** – inclui uma página na pilha antes da página especificada. Exemplo :
Navigation.InsertPageBefore(insertPage, beforePage)

Acontece que muitas vezes precisamos passar informações entre as páginas durante a navegação. Existem duas técnicas que podemos usar para realizar essa tarefa :

1. **Construtor da página** - Passar dados através do construtor da página;
2. **BindingContext** - Definir a propriedade BindingContext da nova página com os dados que desejamos passar;

Vejamos como realizar essas tarefas na prática primeiro usando a abordagem do construtor da página.

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_Navegacao** 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.*)

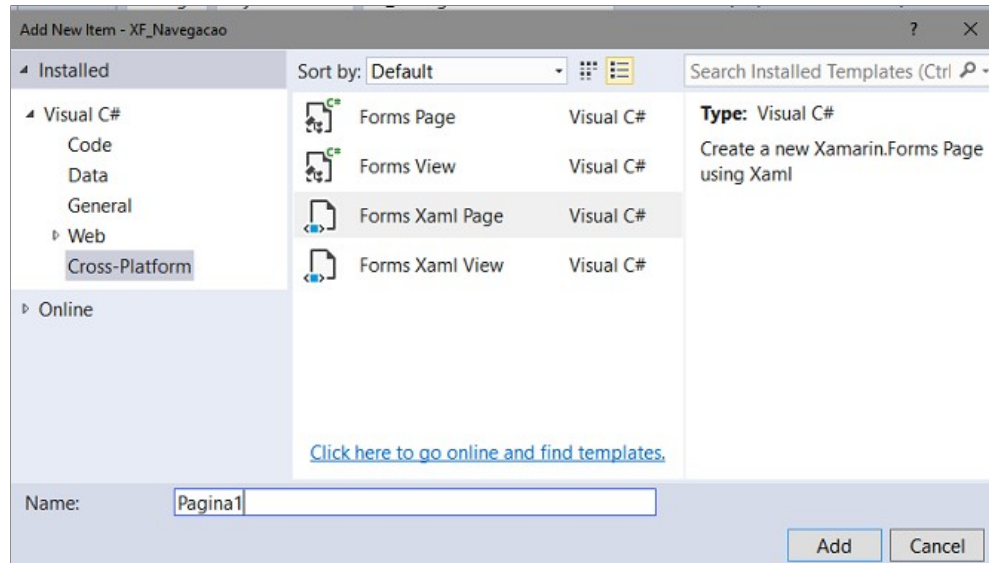
O projeto comum possui a classe **App.cs** que irá conter o código compartilhado e que vamos usar neste artigo.

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_Navegacao.Pagina1"
    Title="PÁGINA 1">
    <StackLayout HorizontalOptions="Center" VerticalOptions="Center">
        <Entry x:Name="txtNome" Placeholder="Informe o seu nome" />
        <Button Text="Avançar" Clicked="Button_Clicked" BackgroundColor="Lime" FontSize="25" />
    </StackLayout>
</ContentPage>
```

Nesta página temos uma view **Entry** definida com o nome **txtNome** onde o usuário vai informar o seu nome.

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

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

```
private async void Button_Clicked(object sender, EventArgs e)
{
    await Navigation.PushAsync(new Pagina2(txtNome.Text));
}
```

```
}
```

Observe que estamos passando o conteúdo do texto informando na view Entry, **txtNome**, no construtor da página **Pagina2()**.

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_Navegacao.Pagina2"
  Title="PÁGINA 2">

  <StackLayout HorizontalOptions="Center" VerticalOptions="Center">
    <Label Text="Bem Vindo" FontSize="Large" FontAttributes="Bold" TextColor="Navy" />
    <Label x:Name="lblNome" FontSize="Large" FontAttributes="Bold" TextColor="Black" />
  </StackLayout>
</ContentPage>
```

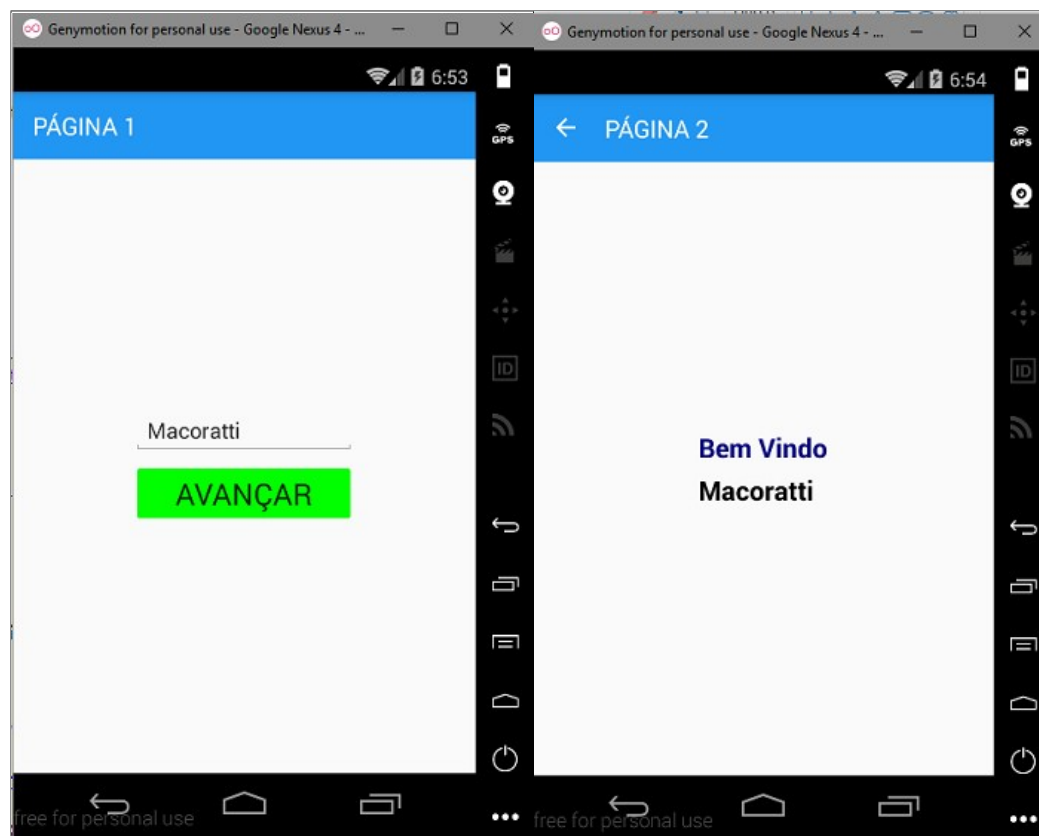
Nesta página temos duas views Labels, a primeira apresenta o texto : **Bem-Vindo**, e na segunda, identificada como **lblNome**, é onde vamos exibir o nome que foi informado na página **Pagina1**.

Para isso temos que definir o seguinte código no arquivo code-behind **Pagina2.xaml.cs**:

```
public Pagina2(string texto)
{
    InitializeComponent();
    lblNome.Text = texto;
}
```

Neste código recebemos no construtor uma string texto que atribuímos à Label **lblNome** da página **Pagina2**.

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.

Neste momento você poderá estar se perguntando como passar mais de uma informação ? Ou como passar dados complexos entre páginas ?

Na [próxima parte](#) do artigo vou abordar esse assunto.

*E, como Moisés levantou a serpente no deserto, assim importa que o Filho do homem seja levantado;
Para que todo aquele que nele crê não pereça, mas tenha a vida eterna.*

João 3:14,15

[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# Básico - 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>
- [Xamarin Studio - Desenvolvimento Multiplataforma com C# \(Android, iOS e Windows\)](#)
- [Xamarin - Criando Apps com o Visual Studio e C#](#)
- https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_binding_basics/

[José Carlos Macoratti](#)