



Neste artigo vou mostrar como podemos explorar a experiência de navegação que o Xamarin Forms oferece com base no tipo de página utilizada, iniciando com a navegação hierárquica.

O [Xamarin Forms](#) fornece um diferente número de experiências de navegação dependendo do tipo de página que usamos em nossas aplicações.



Curso de Xamarin Forms Vídeo Aulas

Desenvolva para Android, iOS e Windows Phone

Na figura abaixo vemos os principais tipos de página que podemos usar no Xamarin Forms:



ContentPage



MasterDetailPage



NavigationPage



TabbedPage



CarouselPage

Temos dois principais tipos de navegação entre páginas usados no Xamarin Forms:

- **Navegação Hierárquica** : Permite ao usuário se mover para baixo em uma pilha de páginas e depois subir novamente pelos níveis;
- **Navegação não Hierárquica ou Modal** : É uma tela de interrupção que requer uma ação do usuário, e pode ser descartada por um botão **Cancel**;

Apresentando a Navegação Hierárquica

A classe **NavigationPage** fornece a experiência de navegação hierárquica onde o usuário pode navegar entre as páginas para frente e para trás.

Esta classe implementa a navegação em uma pilha de páginas (*LIFO- last in, first out*) onde a última página a entrar é a primeira a sair.

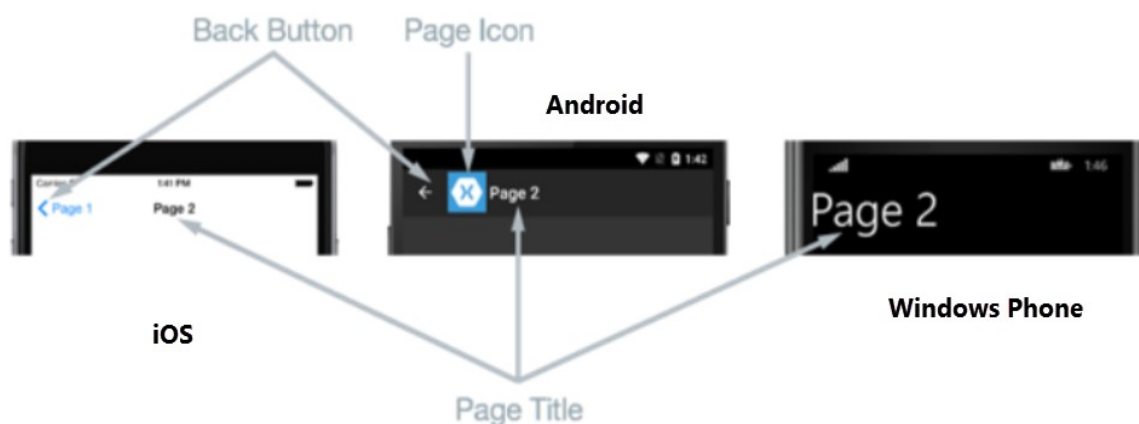
Para passar de uma página para outra, uma aplicação irá empurrar uma nova página para a pilha de navegação, onde se tornará a página ativa:



Para retornar à página anterior, o aplicativo puxará a página atual da pilha de navegação e a nova página mais alta se tornará a página ativa



Na navegação hierárquica a classe **NavigationPage** é usada para navegar através de uma pilha de objetos **ContentPage** onde os principais componentes usados em cada plataforma é mostrado abaixo:



Fluxo da Navegação Hierárquica

Vamos supor um cenário onde temos duas páginas: **Pagina1.xaml** e **Pagina2. xaml**

A seguir veremos o fluxo da navegação hierárquica usado para navegar entre as páginas:

1 - A primeira página adicionada a uma pilha de navegação é referida como a página raiz do aplicativo :

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

Isso faz com que a instância **ContentPage** de **Pagina1** seja empurrada para a pilha de navegação, onde ela se torna a página ativa e a página raiz da aplicação.

2 - Para navegar para **Pagina2**, é necessário invocar o método **PushAsync** na propriedade **Navigation** da página atual :

```
Async void OnNextPageButtonClicked (objeto remetente, EventArgs e)
{
    await Navigation.PushAsync (new Pagina2());
}
```

Isso faz com que a instância de **Pagina2** seja empurrada para a pilha de navegação, onde ela se torna a página ativa

A página ativa pode ser extraída da pilha de navegação pressionando o botão **Voltar** no dispositivo, independentemente de se tratar de um botão físico no dispositivo ou de um botão na tela.

Para retornar programaticamente à página original, a instância **Pagina2** deve invocar o método **PopAsync**:

```
Async void OnPreviousPageButtonClicked (objeto remetente, EventArgs e)
{
    await Navigation.PopAsync ();
}
```

Assim como os métodos **PushAsync** e **PopAsync**, a propriedade de navegação de cada página também fornece um método **PopToRootAsync** :

Esse método remove todas, exceto a página raiz da pilha de navegação, tornando a página raiz do aplicativo a página ativa:

```
Async void OnRootPageButtonClicked (objeto remetente, EventArgs e)
{
    await Navigation.PopToRootAsync ();
}
```

Observe que os métodos para navegação são expostos pela propriedade **Navigation**.

Vejamos a seguir um exemplo prático mostrando a navegação hierárquica.

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_Apps** 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.)*

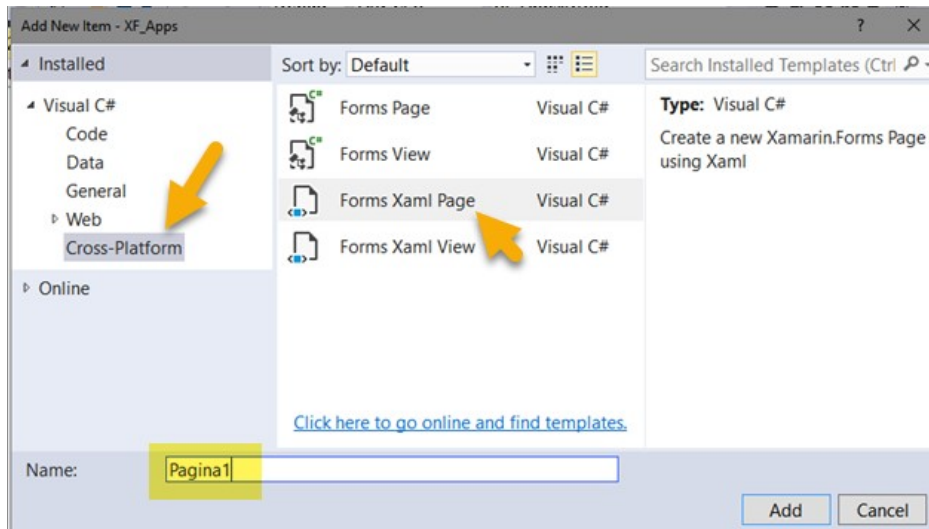
O projeto comum possui a classe **App.cs** que irá conter o código compartilhado e que vamos usar neste artigo.

Não esqueça de atualizar o Xamarin Forms acionando o menu **Tools** e as opções **Nuget Package Manager -> Manage Nuget Packages for Solution**;

A seguir clique em **Update** e selecione o item **Xamarin Forms** e clique em **Install**.

[Criando as páginas para Navegação : Pagina1.xaml e Pagina2.xaml](#)

No menu **Project** clique em **Add New Item** e a seguir em **Cross Platform** e em **Forms Xaml Page** e informe o nome **Pagina1**:



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

Inclua o código abaixo na **Pagina1** :

```
<?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_Apps.AppNavegacao1.Views.Pagina1"
    Title="Página 1">

    <StackLayout Padding="10">
        <Image Source="http://www.macoratti.net/Imagens/animais/cao1.jpg"
            Aspect="AspectFit"
            WidthRequest="300"
            HeightRequest="300"/>
        <Button Text="Ir para Pagina2" Clicked="Button_Clicked" />
    </StackLayout>

</ContentPage>
```

No evento **Clicked** do Button inclua o código que vai permitir navegar para a **Pagina2** usando o método **PushAsync**:

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

A seguir defina o código XAML da **Pagina2**:

```
<?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_Apps.AppNavegacao1.Views.Pagina2"
    Title="Página 2">

    <StackLayout Padding="10">
        <Image Source="http://www.macoratti.net/Imagens/animais/cao2.jpg"
            Aspect="AspectFit"
            WidthRequest="300"
            HeightRequest="300"/>
        <Button Text="Voltar para Página 1" Clicked="btn_Clicked"/>
    </StackLayout>
```

</ContentPage>

No evento **Clicked** do Button temos o código que permite retornar para a **Página1** usando o método **PopAsync**:

```
private async void btn_Clicked(object sender, EventArgs e)
{
    await Navigation.PopAsync();
}
```

Podemos também navegar de volta à **Página1** usando o botão de retorno da barra de navegação:

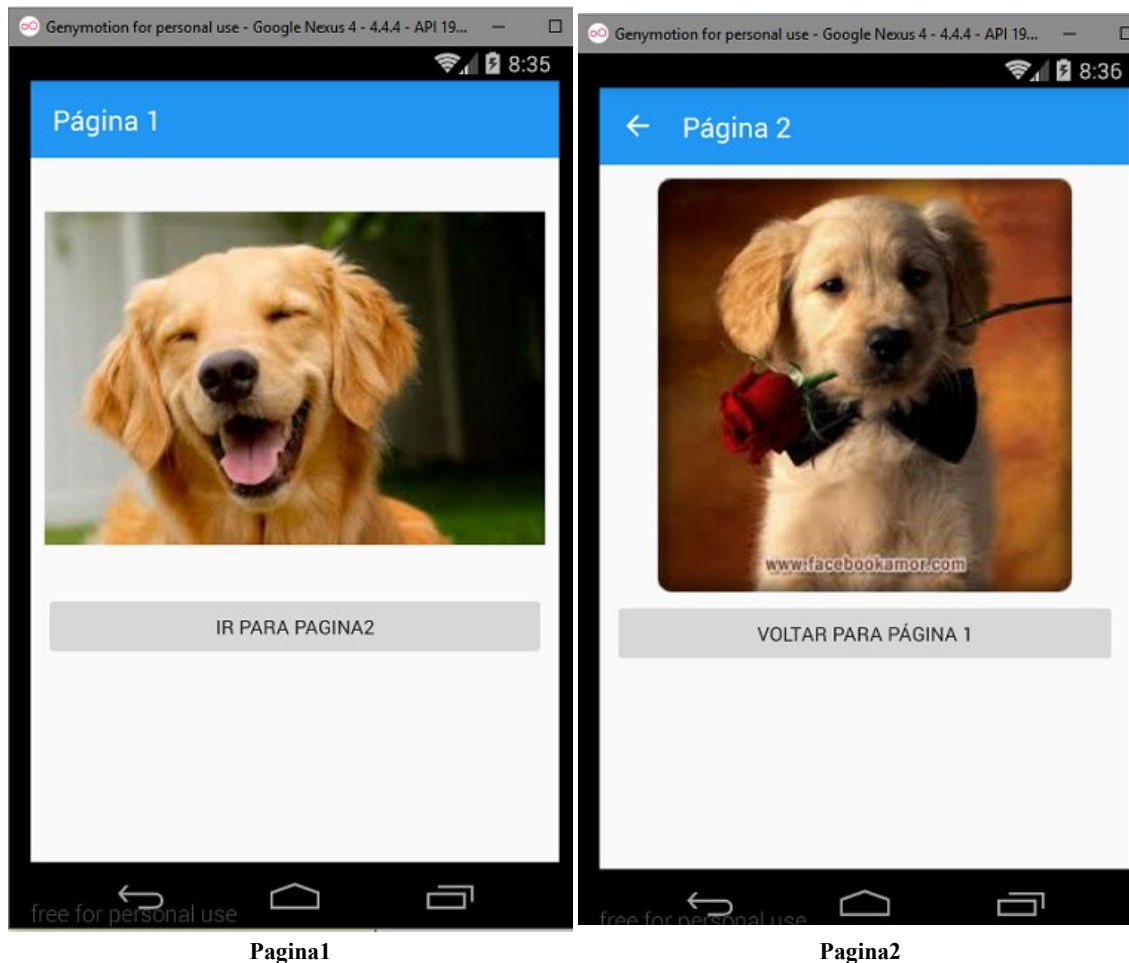
Note que definimos em cada página um título usando a propriedade **Title**. Isso é muito importante para identifica a página na navegação.

O código das páginas é bem simples e usa um **StackLayout** e uma view **Image** que exibe uma imagem na página.

A seguir abra o arquivo **App.cs** e inclua o código abaixo definindo a página raiz da aplicação usando **NavigationPage**:

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

Executando a solução usando o emulador Android [GenyMotion](#) iremos obter o seguinte resultado:



Na [próxima parte do artigo](#) vou apresentar a navegação modal.

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

Mas, ó homem, quem és tu, que a Deus replicas? Porventura a coisa formada dirá ao que a formou: Por que me fizeste assim?

Ou não tem o oleiro poder sobre o barro, para da mesma massa fazer um vaso para honra e outro para desonra?

Romanos 9:20,21

[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#](#) (vídeo aula)
- https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_binding_basics/

[José Carlos Macoratti](#)