

Macoratti.net Xamarin.Forms - Apresentando os principais recursos



No artigo de hoje vou apresentar os principais recursos usados na criação de uma aplicação **Xamarin.Forms**.



Nota: Quando este artigo foi escrito o Xamarin era uma ferramenta paga, mas em março de 2016 ela foi adquirida pela Microsoft e agora esta integrada ao Visual Studio de forma gratuita.

Você pode fazer o download do Xamarin neste link : <https://www.xamarin.com/download>

O **Xamarin.Forms**, lançado em 2014, é uma plataforma que roda no **Xamarin** e foi originalmente criada como um **kit de ferramentas UI** que permite aos desenvolvedores criar facilmente interfaces de usuário que podem ser compartilhadas entre **Android**, **iOS** e **Windows Phone**.

O **Xamarin.Forms** foi escrito na linguagem C#, e, permite a prototipagem rápida de aplicações que podem evoluir ao longo do tempo para aplicações complexas. Como os aplicativos **Xamarin.Forms** são aplicativos nativos, eles não têm as limitações de outros kits de ferramentas.

Usando a API **Xamarin.Forms**, você pode criar uma única base de código para sua interface de usuário, e plataforma **Xamarin** irá traduzi-lo automaticamente para os elementos de interface do usuário nativo para cada plataforma que você está alvejando.

A API Xamarin.Forms é conhecida como uma biblioteca de abstração de interface que funciona como uma **Biblioteca de Classes Portátil (PCL)**, situando-se nas plataformas de destino e criando elementos de interface do usuário que o aplicativo precisa. A vantagem do Xamarin.Forms é que ele permite criar interfaces de aplicativo de forma rápida e com código compartilhado.

Neste artigo vou apresentar os principais recursos do **Xamarin.Forms**.

Principais Recursos

A API do Xamarin.Forms esta dividida em 4 áreas principais:

1. **Pages** - o Xamarin.Forms representa uma única tela em sua aplicação. São semelhantes ao conceito de **Activity** do Android, ou a uma **Page** no Windows Phone, ou a um controle **View** no iOS;
2. **Layouts** - É um subtipo especializado de View e atua como um container para outros Layouts e Views. Os subtipos de layouts contém lógica que é específica para organizar as views filhas de certa forma;
3. **Views** - referem-se a controles ou **widgets** em outras plataformas. Correspondem a elementos de interface do usuário como *labels*, *buttons*, *campos textos*, etc.;
4. **Cells** - Esta classe é um elemento especializado que é usado para itens em uma lista ou tabela. Ele descreve como cada item na lista deverá ser desenhado;

1 - Pages

Uma página ocupa a tela inteira na maioria dos casos e age da mesma forma como uma **Page** no Windows Phone e como um **UIViewController** no iOS; no Android ela se parece mais com um **Resource.Layout** e não como uma **Activity**.

Nota: Uma Activity é uma classe que gerencia a UI. Todo aplicativo Android inicia com uma Activity.



ContentPage



MasterDetailPage



NavigationPage



TabbedPage



CarouselPage

Existem 5 tipos de **Pages** :

Tipo	Descrição
ContentPage	Contém uma única view.
MasterDetailPage	Uma página que tem dois painéis para a página. Normalmente, o painel mestre irá conter os um menu com o detalhe do conteúdo.
NavigationPage	Uma página que contém uma barra de navegação. As páginas são mantidas em uma pilha e podemos navegar entre elas. A barra de navegação pode manipular os botões de ambos os lados da barra, bem como um título.
TabbedPage	Uma página de container. O TabbedPage atua como um recipiente que contém as páginas de conteúdo associados a cada guia.

CarouselPage	Uma página que permite uma movimentação rápida para exibir outras views.
---------------------	--

2 - Layouts

A classe **Layout** no Xamarin.Forms é um subtipo de view especializado, que atua como um recipiente para outros **Layouts ou Views**. Ela normalmente contém a lógica para definir a posição e o tamanho dos elementos filhos em aplicações Xamarin.Forms.

Existem 7 diferentes tipos de Layouts:



Layout	Descrição
ContentView	Usado como classe base para views definidas pelo usuário. Trata um único elemento.
Frame	Um objeto Pai que contém um único objeto filho que pode ser enquadrado.
ScrollView	Uma views capaz de ser muito maior do que a tela do dispositivo. Se o conteúdo não exceder o tamanho da tela, o deslocamento não tem efeito e atua como um modo de exibição padrão.
AbsoluteLayout	Uma view que posiciona layouts filhos em posições especificadas usando âncoras para definir a colocação e o tamanho.
Grid	Um layout que contém várias views dispostas em linhas e colunas (como seria encontrada em um visualizador de imagens da câmera).
RelativeLayout	Um layout que posiciona os elementos relativos aos outras usando restrições.
StackLayout	Um dos layouts mais usados. Ele posiciona os elementos filho em uma linha. O layout lida com os limites dos filhos, portanto, quaisquer limites definidos pelo usuário são substituídos.

3 - Views

O Xamarin.Forms usa a palavra View para se referir a objetos visuais, como botões, etiquetas ou caixas de entrada de texto - o que pode ser mais comumente conhecido como controles de widgets. Estes elementos de interface do usuário são normalmente subclasses de View.



Nota: Emulador Android.

Existem atualmente 19 tipos diferentes de Views :

View	Descrição
ActivityIndicator	Usado para mostrar o progresso de tempo (durante uma atividade como download, cálculos, etc.)
BoxView	Permite desenhar caixa sólidas coloridas.
Button	Um botão padrão.

DatePicker	Produz um controle para obter a data específico para a plataforma.
Editor	Um controle editor de texto multilinha
Entry	Um controle de entrada de texto de única linha.
Image	Trata uma imagem a partir da plataforma host. Estes devem ser colocados em locais onde normalmente estão os gráficos (<i>Resources/Drawable no Android, Resource no iOS, e Assets no Windows Phone</i>).
Label	Exibe texto que não pode ser alterado. Pode se estender por várias linhas.
ListView	Exibe uma coleção de dados em uma lista vertical
OpenGLView	É usado para jogos ou código que usam gráficos de forma intensa.
Picker	Um controle giratório que permite escolher a partir de uma lista de itens.
ProgressBar	Parecido como a view ActivityIndicator .
SearchBar	Um controle de entrada usado para realizar buscas.
Slider	Um método de deslizamento para a entrada de um valor controlado de forma linear.
Stepper	Uma view que permite a movimentação entre valores definidos pelo usuário usando os botões + e - .
Switch	Uma view de alternância.
TableView	Esta mais próximo de um TableView Android que possui linhas de células.(<i>Não é o mesmo que o UITableView do iOS</i>)
TimePicker	Similar ao DatePicker , para obter a hora.
WebView	Uma view que permite exibir HTML gerado a partir de uma app ou usando um recurso online.

4 - Cells

Uma **Cell** (célula) é um elemento especializado usado para itens em uma tabela e descreve como cada item em uma lista deve ser desenhado. Uma **Cell** deriva de **Element**, a partir do qual o **VisualElement** também deriva. No entanto, uma **Cell** não é um elemento visual, ele apenas descreve um modelo para criar um elemento visual. A classe **Xamarin.Forms.Cell** fornece a classe base e as capacidades para todas as células Xamarin.Forms. As células são elementos concebidos para serem adicionado aos controles **ListView** ou **TableView**.

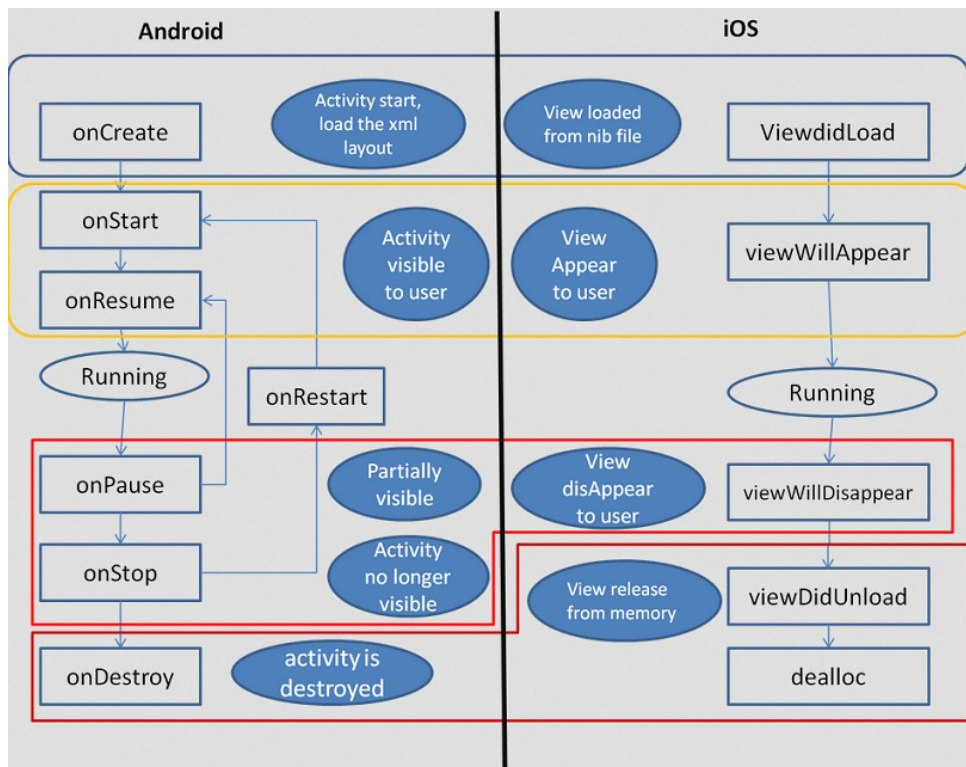


Nota: Emulador Android.

Existem 4 tipos de Cells:

Cell	Descrição
EntryCell	Uma célula contendo uma Label e um campo de entrada de texto de linha única
SwitchCell	Uma célula que contém uma Label e um interruptor liga/desliga.
TextCell	Uma célula contendo um campo de texto primário e um secundários.
ImageCell	Uma célula de texto que também contém uma imagem

Para concluir abaixo temos uma figura que exhibe o ciclo de vida de uma aplicação **Android** e de uma aplicação **iOS**:



Acessada a partir de <http://i.stack.imgur.com/Jn6MZ.png> em dezembro de 2015.

Podemos então, resumir o processo no seguinte fluxo:

1. **Iniciar;**
2. **Tornar a Interface do Usuário (UI) visível;**
3. **Tratar eventos;**
4. **Encerrar;**

Aguarde mais artigos sobre assuntos relevantes em relação ao **Xamarin e Xamarin.Forms**.

Porque o amor ao dinheiro é a raiz de toda a espécie de males; e nessa cobiça alguns se desviaram da fé, e se traspassaram a si mesmos com muitas dores.

1 Timóteo 6:10

[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 - Video 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](#) **NEW**

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#](#)
- [Super DVD C# - Recursos de aprendizagens e vídeo aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti .net](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW
- [Macoratti .net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | Twitter](#)
- **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# \(vídeo aula\)](#)

[José Carlos Macoratti](#)