

Macoratti.net Xamarin Studio - Apresentando Xamarin.Forms



No artigo de hoje vou apresentar os conceitos básicos sobre o **Xamarin.Forms** da plataforma **Xamarin** como uma alternativa para a criação de **Interfaces de Usuários** multiplataforma para dispositivos móveis usando a linguagem C# e o .NET Framework.

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.

As aplicações escritas utilizando Xamarin.Forms são capazes de utilizar qualquer um dos recursos da API ou da plataforma subjacente, tais como **CoreMotion**, **PassKit**, e **StoreKit** no iOS; **NFC** e **Google Play** no Android; e **Tiles** no Windows Phone. Isso também significa que é possível criar aplicações que terão partes da sua interface de usuário criada com **Xamarin.Forms** enquanto outras partes são criadas usando o kit de ferramentas UI nativo.

As aplicações Xamarin.Forms são arquitetadas na mesma forma que as aplicações multiplataformas tradicionais sendo que a abordagem mais comum é usar **Portable Libraries** (*Livrarias Portáteis*) ou **Shared Projects** (*Projetos Compartilhados*) e então criar aplicações para plataformas específicas que irão consumir código compartilhado.

Existem duas técnicas para criar interfaces de usuários (UI) no Xamarin.Forms :

1. Criar UI com código fonte usando a API fornecida pelo Xamarin.Forms;
2. Usar a **XAML (Extensible Application Markup Language)** que é uma linguagem

de marcação declarativa da Microsoft que é usada para descrever interfaces. A interface do usuário é definida em um arquivo **XAML** usando a sintaxe XML, enquanto que o comportamento em tempo de execução é definido em um arquivo *code-behind* separado;

A vantagem de atingir várias plataformas com uma única linguagem de programação vem da capacidade de compartilhar código entre as aplicações. Antes do código pode ser compartilhado, um aplicativo deve ser estruturado para essa finalidade e o padrão de arquitetura **MVVM**, ajuda o desenvolvedor a separar o código para uma view de uma plataforma específica, o código que precisa interagir com a API da plataforma e o **Model e View-Model** da plataforma independente.

A parte da aplicação que é independente de plataforma pode ser isolada e, em um contexto do **Visual Studio ou Studio-Xamarin**, pode ser colocada em um projeto separado da seguinte forma:

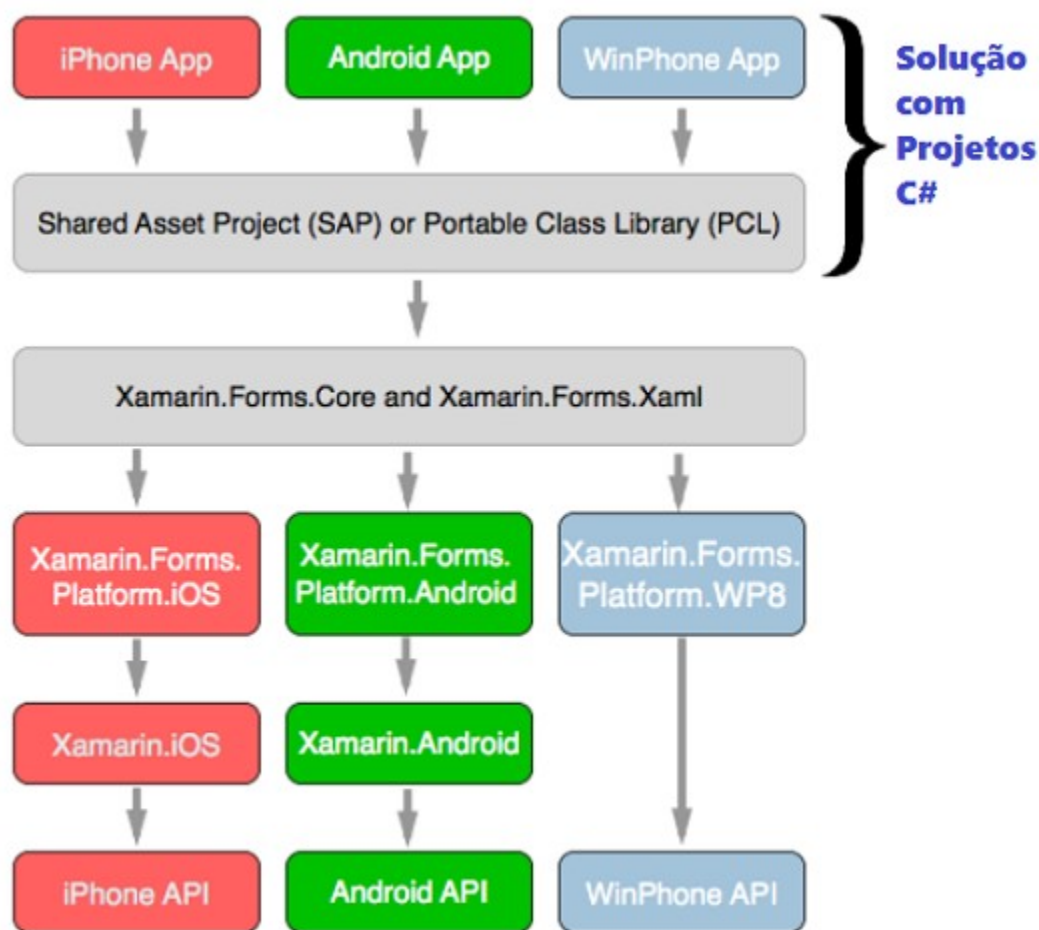
- 1- **Projeto de recurso compartilhado - Shared Asset Project (SAP)** - que consiste simplesmente em arquivos de código e outros ativos acessíveis a partir de outros projetos;
- 2- **Livraria de Classe Portável - Portable Class Library (PCL)** - que inclui todo o código comum em uma biblioteca de vínculo dinâmico (DLL) que pode então ser referenciada a partir de outros projetos;

Os Projetos Compartilhados (Shared Asset Projects) permitem que você escreva código que é compartilhado entre vários projetos alvo, incluindo aplicações Xamarin. Eles suportam diretivas de compilador para que você possa incluir condicionalmente código específico de plataforma para ser compilado em um subconjunto dos projetos que fazem referência ao projeto compartilhado.	Portable Class Libraries(PCLS) ou Bibliotecas de Classes Portátil permitem que você escreva código e produza bibliotecas que podem ser compartilhadas entre múltiplas plataformas incluindo Xamarin.iOS , Xamarin.Android e Windows Phone . As PCLS podem ser criadas tanto no Xamarin Studio como no Visual Studio, e em seguida serem referenciadas em cada projeto específico da plataforma para permitir que o código seja facilmente compartilhado.
--	--

Independentemente do método utilizado, este código comum tem acesso à biblioteca de classes do .NET Framework, para que possa executar operações de *I/O com arquivos, lidar com a globalização, acessar web services, decompor XML, etc.*

Isso significa que você pode criar uma única solução no Visual Studio que contém quatro projetos C# direcionados para as três principais plataformas móveis (*todos com acesso a um SAP comum ou projeto PCL*), ou você pode usar o **Xamarin Studio** para atingir dispositivos [iPhone](#) e [Android](#).

Dessa forma uma aplicação Xamarin.Forms é composta de **três projetos separados** para cada uma das três plataformas, com um **quarto projeto que contém o código comum**. Veja abaixo o diagrama representando esse cenário:



Os projetos das 3 plataformas na aplicação **Xamarin.Forms** são pequenos e constituídos muitas vezes apenas de código de inicialização.

O projeto de ativo compartilhado - **Shared Asset Project**, ou o projeto de livreria de classe portátil - **Portable Class Library** - contém a maior parte do código da aplicação

incluindo o código da interface com o usuário.

As bibliotecas **Xamarin.Forms.Core** e **Xamarin.Forms.Xaml** implementam a **API Xamarin.Forms**. Dependendo da plataforma, a biblioteca **Xamarin.Forms.Core** usa uma das bibliotecas da plataforma **Xamarin**.

Essas bibliotecas são na maior parte uma coleção de classes chamadas *renderizadores* que transformam os objetos da interface do usuário Xamarin.Forms na interface do usuário específica da plataforma.

O Xamarin.Forms também dá suporte ao **XAML** (*pronuncia-se "zammel"*) da Microsoft como uma linguagem de marcação de uso geral para instanciar e inicializar objetos. Abaixo temos um exemplo de código XAML :

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
              x:Class="HelloXamarinFormsWorldXaml.StackLayoutExample1"
              Padding="20">

  <StackLayout Spacing="10">

    <Label Text="Parar"
           BackgroundColor="Red"
           Font="20" />

    <Label Text="Devagar"
           BackgroundColor="Yellow"
           Font="20" />

    <Label Text="Siga"
           BackgroundColor="Green"
           Font="20" />

  </StackLayout>
</ContentPage>
```

No entanto, apesar da enorme ajuda que o **Xamarin.Forms** pode lhe oferecer na criação de aplicações para plataforma independente, ele não é um substituto completo para a programação da API nativa. Se a sua aplicação depende fortemente de recursos da API nativas, como determinados tipos de controles ou widgets, então você pode precisar usar o **Xamarin.iOS**, o **Xamarin.Android**, ou a **API do Windows Phone**.

Provavelmente você também vai precisar usar as APIs nativas para aplicações que

necessitam de [gráficos vetoriais ou interação ao toque sensível](#). A versão atual do Xamarin.Forms não está completamente pronta para oferecer esses recursos a nível avançado.

Por outro lado, Xamarin.Forms é ótimo para prototipagem ou para criar aplicações de linha de negócio de forma rápida. E depois de ter feito isso, você pode simplesmente achar que você pode continuar usando os recursos do Xamarin.Forms para construir o aplicativo inteiro.

Mesmo se você começar a construir uma aplicação com Xamarin.Forms e, em seguida, implementar grandes partes dela com APIs da plataforma, você estará fazendo isso dentro de uma estrutura que permite que você compartilhe o código e que oferece formas estruturadas para fazer visuais específicos para cada plataforma.

Como saber quando usar **Xamarin.Forms** ou [Xamarin.iOS](#) ou [Xamarin.Android](#) ?

A plataforma Xamarin.Forms é mais indicada para:

- Aplicativos de entrada de dados;
- Protótipos e provas de conceito;
- Aplicativos que exigem pouca funcionalidade específica da plataforma;
- Apps onde o código compartilhado é mais importante do que a UI;

Já usar [Xamarin.iOS](#) ou [Xamarin.Android](#) é mais adequado para :

- Aplicativos que requerem interações especializadas;
- Aplicativos com design específico e especializado;
- Aplicativos que usam muitas APIs específicas da plataforma;
- Apps onde a UI personalizada é mais importante que o código compartilhado;

Na [próxima parte do artigo](#) veremos como criar interfaces com o usuário usando XAML e a linguagem C#.

[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](#) 
- [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](#)