Macoratti.net Xamarin Android - Apresentando a arquitetura Xamarin.Android

Neste artigo vou apresentar uma visão geral sobre a arquitetura do **Xamarin.Android** do Framework Xamarin.

A plataforma **Xamarim** é uma solução da Microsoft, totalmente gratuita, para criar aplicações nativas nas 3 plataformas : **Android, iOS e Windows**, usando as IDE, Xamarim Studio ou o Visual Studio, e uma linguagem, a linguagem C#.

Além da plataforma Xamarim , a Xamarim possui mais duas ferramentas para o desenvolvimento de aplicações móveis multiplataforma:

1. Xamarim Test Cloud - Permite testar a sua aplicações em centenas de dispositivos, na nuvem, de forma automática;
2. Xamarim Insights - Permite monitorar a sua aplicação em tempo real detectando problemas de desempenho ou falhas;

O destaque é que agora podemos usar o Visual Studio ou o Xamarin Studio para o desenvolvimento de aplicações multiplataforma usando a linguagem C# criando aplicativos nativos dos Sistemas Operacionais Windows, Android e iOS.

Nota : Se você ainda não tem o Xamarin acesse o link e baixe aqui : https://www.xamarin.com/download

Como o nosso foco será o desenvolvimento para Android vou me ater apenas nos aspectos relacionados à plataforma Android.

Porque usar o Xamarin.Android ?

Antes de abordar a arquitetura do Xamarin.Android, vamos discutir primeiro a questão do porquê o Xamarin.Android é a nossa escolha.

Indo direto ao que interessa vamos relacionar as vantagens e também desvantagens em usar o Xamarin.Android:

1- Vantagens

a - Aproveitar o seu conhecimento e habilidade na plataforma .NET e na linguagem C#;
Você já investiu uma grande quantidade de tempo e recursos no aprendizado da linguagem C# e da plataforma .NET.
Se sua necessidade agora é desenvolver aplicativos para a plataforma Android porque não aproveitar todo o seu conhecimento e usar uma ferramenta grátis ganhando tempo e dinheiro ?

b - Reutilização de código em ambiente de desenvolvimento multiplataforma
Embora o Xamarin.Android não permita que você construa um único aplicativo que pode ser implantado para Android, iOS e WP8, ele lhe dá a capacidade de reutilizar grandes porções de sua base de código em todas essas plataformas.

Em geral, o código de interface de usuário e o código que lida com as capacidades do dispositivo tendem a ser escrito para cada plataforma, enquanto a lógica de cliente do serviço, validação do lado do cliente lado, cache de dados e armazenamento de dados do lado do cliente podem ser potencialmente ser reutilizados, poupando uma quantidade significativa de tempo.

c - O Xamarin é totalmente grátis e já vem integrado com o Visual Studio e também oferece a opção de uma instalação a parte como o Xamarin Studio
Tudo que você tem que fazer é fazer o download do Xamarin e escolher se deseja usá-lo integrado ao Visual Studio ou como uma ferramenta á parte no Xamarin Studio.

O Xamarin é grátis e o Visual Studio Community 2015 também é grátis.

2- Desvantagens

a - Tempo de espera para atualizações no Xamarin.Android
Há algum tempo de defasagem entre uma nova versão da plataforma Android e a liberação correspondente para a plataforma Xamarin.Android.

a - Desempenho e gerenciamento de memória
Em alguns casos, o Xamarin.Android aloca objetos Java e C# para alcançar alguns dos benefícios de desenvolver em C#/.NET para plataforma Android, e , isso tem um impacto tanto no consumo de memória como no desempenho da

execução da aplicação. Infelizmente, até o momento, não se têm nenhum dado objetivo para quantificar este impacto.

c - Tamanho do pacote de distribuição

Existe um número maior de bibliotecas *runtimes* que precisam ser distribuídas com uma aplicação Xamarin.Android, aumentando assim o tamanho do pacote gerado.

O que é o Mono e como uma aplicação Xamarin.Android roda ?

O Mono é uma implementação multiplataforma do compilador C#, e da **Common Language Runtime (CLR)** que é compatível com a plataforma .NET.

O **Mono CLR** foi portado para várias plataformas, incluindo o Android e muitas distribuições Linux (*BSD, OS X, Windows, Solaris e até mesmo alguns consoles de jogos como Wii, Xbox 360)*.

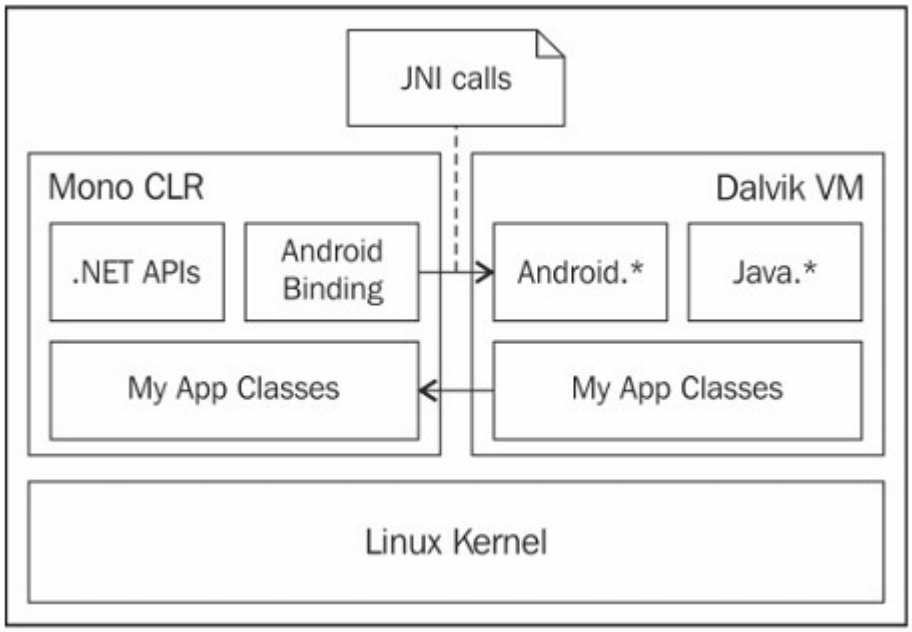
Além disso o Mono fornece um compilador estático que permite que as aplicações sejam compiladas para ambientes iOS e PS3.

Como já citei anteriormente, as aplicações rodam na máquina virtual Dalvik (Dalvik VM) e as aplicações **Mono** rodam no **Mono CLR**.

Isso posto, como uma aplicação Xamarin.Android é executada ?

Uma resposta simples é que ela usa tanto o Mono CLR como a Dalvik VM.

No diagrama abaixo temos uma descrição de como esses dois ambientes coexistem :



Como o **Mono CLR e a Dalvik VM** trabalham juntos em um aplicativo Xamarin.Android ?

A magia é realizada através de um conceito chamado **peer objects** e um framework chamado de **Java Native Interface (JNI).**

O **Java Native Interface (JNI)** é um framework que permite que um código não Java *(tais como C++ ou C#)* chame ou seja chamado por um código Java em execução dentro de uma JVM *(Java Virtual Machine)*.

Como você pode ver a partir do diagrama anterior, a JNI é um componente crítico na arquitetura Xamarin.Android.

Os **Peer objects** são um par de objetos constituídos de um objeto gerenciado que reside na Mono CLR e um objeto Java que reside na Dalvik VM e que trabalham em conjunto para desempenhar asa funções de uma aplicativo Xamarin.Android.

O Xamarin.Android é entregue com um conjunto de *assemblies* chamado de bibliotecas de vinculação Android. As classes nessas bibliotecas correspondem às classes Java no framework da aplicação Android, e os métodos relativos às classes de vinculação atuam como invólucros para chamar métodos correspondentes nas classes Java.

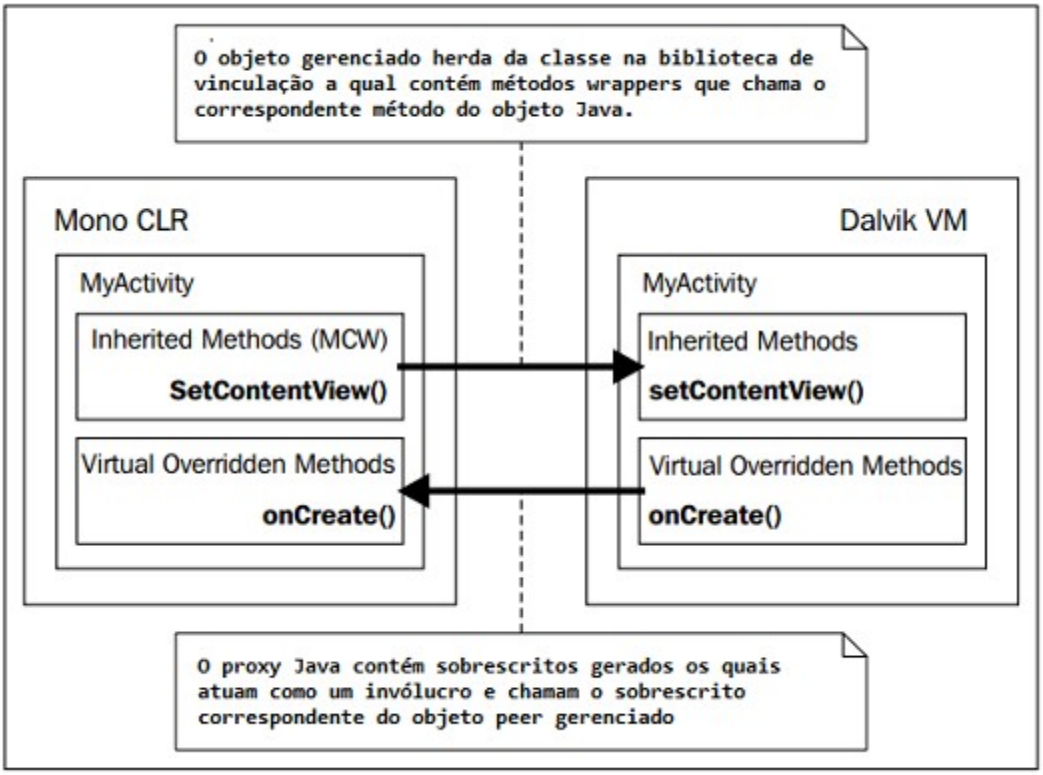
As classes de **Binding** referem-se aos **Managed Callable Wrappers (MCW)**, e, sempre que você criar uma classe C#, que herda uma dessas classes de ligação,

uma classe proxy Java correspondente é gerada em tempo de compilação. O proxy Java contém uma substituição gerada para cada método sobrescrito na classe C# e atua como um *wrapper* para chamar o método correspondente na classe C#.

A criação de **peer objects** pode ser iniciada a partir do Dalvik VM pela framework Android ou de dentro do Mono CLR pelo código que você escrever nos métodos sobrescritos.

Uma referência entre os dois objetos de mesmo nível é mantida por cada instância de um MCW e pode ser acessado através do da propriedade Android.Runtime.IJavaObject.Handle.

O diagrama a seguir mostra como os **objetos peers** colaboram:



Dessa forma, as partes centrais do Xamarin.Android são as vinculações para as APIs do Android o que nos leva ao seguinte:

• A API Android é se torna transparente para um desenvolvedor C# e permite ao desenvolvedor explorar a API usando o preenchimento de código e a documentação no IDE;
• Os desenvolvedores C# podem aproveitar a vasta gama de exemplos Java/Android e a documentação que pode ser facilmente transformada para

utilização com C# e Xamarin.Android.

Para poder criar aplicações Android usando o Xamarin.Android o desenvolvedor pode usar as seguintes ferramentas:

1. **Xamarin Studio** - O Xamarin Studio é uma versão personalizada do **MonoDevelop IDE**, que pode ser usado para desenvolver aplicações Android, iOS e OS X. O Xamarin Studio está disponível tanto para Windows como para OS X ;
2. **Visual Studio** - O Xamarin para o Visual Studio é um add-in que suporta o desenvolvimento de aplicativos Xamarin.Android;

Assim, os arquivos da solução e dos projetos criados e atualizados pelo Xamarin Studio são compatíveis com o Visual Studio, tornando mais fácil alternar entre os dois ambientes durante toda a duração de um projeto. Isso também permite à equipe de trabalho adotar a ferramenta que mais se adéqua ao ambiente usado.

Até mais...

Porque há um só Deus, e um só Mediador entre Deus e os homens, Jesus Cristo homem.

1 Timóteo 2:5

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

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 (@macorati) | Twitter**
- Xamarim - Desenvolvimento Multiplataforma com C# ... - Macoratti.net
- Xamarin - Apresentando Xamarin.Forms - Macoratti.net

- Xamarin.Forms - Olá Mundo - Criando sua primeira ... - Macoratti.net
- Xamarin.Forms - Olá Mundo - Anatomia da aplicação - Macoratti.net

---

José Carlos Macoratti