Ferramentas de desenvolvimento para aplicativos móveis: Xamarim

Felipe Regino, José Olímpio Negrão, Rogério Lázaro Jr.

Universidade Federal da Bahia – Avenida Adhemar de Barros, Salvador, Bahia, Brasil

[liperegino@gmail.com](mailto:liperegino@gmail.com); [joseolimpionegrao@gmail.com](mailto:joseolimpionegrao@gmail.com); [rogeriolazarojr@gmail.com](mailto:rogeriolazarojr@gmail.com);

**Abstract.** This document refers to the description activity of the development tool Xamarim, used for mobile applications creation. On this document, questions about Xamarim’s features will be answered, as described on the activity document, and other complementary information about the experience of first-time development with Xamarim tools.

**Resumo.** Este documento refere-se à atividade de descrição da ferramenta de desenvolvimento Xamarim, utilizada para criação de aplicativos mobile. Neste documento serão respondidas perguntas sobre características do Xamarim conforme descrito em documento da atividade, e complementos sobre experiências com primeiros desenvolvimentos na sua ferramenta.

# 1. Introdução à Atividade

Abaixo, seguem respostas sobre os questionamentos especificados no documento de atividade, publicado no dia 29 de janeiro de 2016, no endereço <http://disciplinas.dcc.ufba.br/MATC89/EquipesFerramentasHibridas>.

## 1.1. Nome, empresa que mantêm ou possui os direitos da ferramenta

O nome da ferramenta é Xamarim, criado e mantido pela Xamarim Inc.. A iniciativa possui cinco investidores. São eles a Charles River Venture, Floodgate Inc., Ignition, LeadEdge Capital e a Insight Venture Partners

## 1.2. Um pouco sobre o Xamarim

O Xamarim foi desenvolvido com o intuito de oferecer uma ferramenta nativa capaz de otimizar o ciclo de criação de aplicativos móveis desde o planejamento até a fase de testes e monitoramento, integrado em uma só ferramenta, de forma que a experiência de desenvolvimento seja a melhor possível para seus usuários desenvolvedores.

## 1.3. Descrevendo a arquitetura do Xamarim

O Xamarin utiliza a arquitetura MONO, um ambiente executado em paralelo com uma máquina virtual Android Runtime, ambos rodando numa base Linux Kernel

De forma simplificada, existem as ACW (Android Callable Wrappers), que são as ligações entre a interface nativa do Java (Java Native Interface), e que são utilizadas no momento em que o Android Runtime (ART) invoca códigos gerenciados. Dessa forma, os métodos virtuais podem ser sobrescritos e as interfaces Java, implementadas.

De forma análoga, existem os MCW (Managed Callable Wrappers) que são usados no momento em que o código gerenciado precisa utilizar estruturas do Android, sendo este o responsável pela conversão entre o código gerado pela ferramenta e pela chamada dos tipos relacionados a interface Java nativa. Não menos importantes, são as subclasses do tipo supracitado, uma vez que são nelas onde as Activities estão definidas.

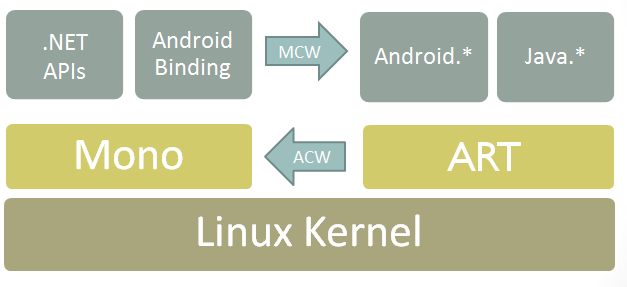
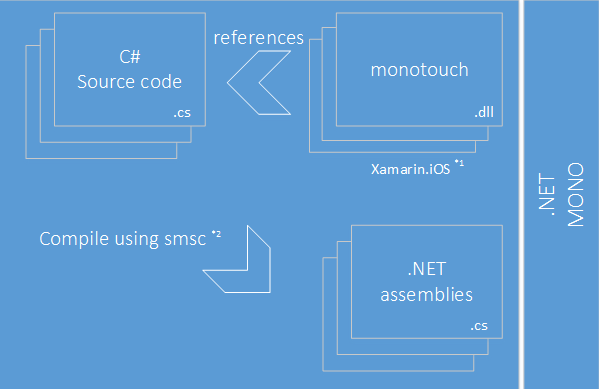
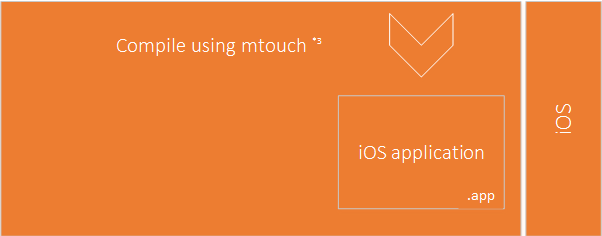


Figura 1. Arquitetura do Xamarin para Android

O código C gerado pelo Xamarim referenci aas DLLs do MonoTouch e esse código é compilado pelo iOS compiler (SMSC), o qual gera um código intermediário, que ainda não é executado pelo iOS. Para ser executado pelo iOS, o código tem que ser unido com bibliotecas .NET, feito por uma ferramenta chamada Mtouch, pertencente ao SDK. Uma vez unidas, se tornam um aplicativo iOS executável.





**Figura 2. Arquitetura do Xamarin para iOS**

## 1.4. Quais são as modalidades de licença do Xamarim?

A IDE da ferramenta, o Xamarim Studio, é livre para download. No entanto, o Xamarim possui modalidades de pagamento, contendo três pacotes:

* Indie ($25/mês): Para uso individual, contendo o básico da ferramenta de desenvolvimento, isto é, a IDE Xamarim Studio com a possibilidade de distribuição para dispositivos como diretamente para app stores como Google Play , Apple App Store, dentre outras funcionalidades.
* Business ($999/ano): Inclui as funcionalidades do pacote Indie, e adiciona a integração da ferramenta com o Visual Studio, suporte da companhia via e-mail, e funções de negócio como bancos de dados e outros.
* Entreprise ($1899/ano): Inclui o pacote Business, e adiciona outros benefícios como acesso prioritário a patches para corrigir bugs da ferramenta, gerenciamento de contas, suporte para iniciação de projetos com engenheiros da Xamarim Inc.

## 1.5. Quais as linguagens de desenvolvimento que o Xamarim usa?

O Xamarim usa uma única linguagem de código, o C#, pois é uma linguagem que é familiar de muitos desenvolvedores e que possui muitas similaridades com Java, Swift e Objective-C, o que facilita na codificação dos apps.

## 1.6. O Xamarim possui IDE?

Sim. Existe a possibilidade de utilizar o Xamarim Studio, da própria empresa, ou utilizar a integração do Xamarim para o Visual Studio 2015.

## 1.7. O Xamarim Studio contempla todo o ciclo de desenvolvimento de um app?

O Xamarim Studio foi desenvolvido para otimizar todo o ciclo de desenvolvimento, desde o planejamento, passando pela fase de testes e finalizando com a ditribuição do app. Não se faz necessária a utilização de outras IDEs ou plugins.

## 1.8. O Xamarim possui seu próprio emulador para testes?

Sim. Graças ao Xamarim Test Cloud, existe a possibilidade de testar um app em até 2000 dispositivos diferentes utilizando Android ou iOS. E caso não exista algum dispositivo dentro do banco de dispositivos, a Xamarim Inc. aceita pedidos para adição destes.

## 1.9. O Xamarim possui a funcionalidade de drag-drop para design de telas?

Sim. Tanto no Xamarim Studio como na integração com o Visual Studio, existe a função de design de telas nativa, contendo um leque extenso de funcionalidades.

## 1.10. Para quais plataformas/SOs é possível desenvolver no Xamarim?

Atualmente o Xamarim suporta o desenvolvimento para Android, Windows Phone, iOS e também para Mac, através da função de portabilidade TouchDraw.

## 1.11. Quais as opções de instalação do aplicativo?

A aplicativo possui um executável assistente, que gerencia o download e instalação dos componentes necessários para o pleno funcionamento das IDEs. São estes: o Xamarin, que é o núcleo contendo a linguagem de programação, APIs e todos os aspectos da ferramenta, o Xamarin Studio, que é a IDE desenvolvida pela empresa utilizando o Xamarin, e o GTK (Graphical user Interface Kit) para utilização das visões de design dentro do Xamarin Studio e suas propriedades.

A utilização do Visual Studio é opcional e não está dentro do pacote disponibilizado. Da mesma forma, o Android SDK e o NDK devem ser baixados no repositório do Google, enquanto o Java SDK (com o JRE embutido) devem ser baixados no repositório da Sun.

## 1.12. De qual abordagem o Xamarim se aproxima mais (Web Approach/Interpreted Approach/Cross Compiled Approach/Hybrid Approach)?

Pode se considerar que o Xamarim tenha uma abordagem de compilação cruzada (cross-compiled). Utilizando a funcionalidade denominada Xamarim.Forms, um código feito em C#, com suas páginas e elementos, são mapeados com identidades próprias do Xamarim.Forms. No momento de execução, esses elementos são transformados em elementos específicos de cada plataforma para total acessibilidade. Exemplo: uma entrada de texto na IDE seria uma Xamarim.Forms Entry. Numa execução do app para Android, esse tipo Entry é mapeada para um Edit Text. Da mesma forma, para iOS ela é mapeada para um UITextView, e para Windows Phone, para uma TextBox.

## 1.13. Quais as APIs suportadas pelo Xamarim?

Abaixo segue uma tabela resumida das APIs suportadas pelo Xamarin nos sitemas operacionais Android e iOS. Não foram encontradas referências de APIs para Windows Phone.

Tabela 1. Lista de APIs suportadas pelo Xamarim no Android

|  |  |
| --- | --- |
| **ANDROID** | **Descrição e versões suportadas** |
| Android.\* | Contém classes de recursos usados ​​por aplicativos incluídos na plataforma e  define permissões de aplicativos para os recursos do sistema, como o bluetooth (versões suportadas 2.x, 3.0), .database, responsáveis pelo gerenciamento das bases de dados, recursos de hardware como câmera e microfones, localização, sistema operacional, ferramentas de aceleração gráfica, segurança entre outros. |
| Dalvik.Annotation | Define uma anotação usado ser usado dentro da anotação TestInfo. |
| Dalvik.Bytecode | Informações sobre opcodes Dalvik. |
| Java.\* | Diversas funções originadas do java, Segurança, criptografia,  gerência de elementos de entrada e saída, tipos de variáveis e gerencia de bancos de dados. |
| Nunit | Framework  para testes unitários. |
| OpenTK.Audio | Ligações para o sistema de áudio OpenAL |
| OpenTK.Graphics | Métodos referentes a funções OpenGL, suporte para OpenGLES 1.0/ OpenGLES 1.1 / OpenGLES 2.0/OpenGLES 3.0 |
| OpenTK.Platform | Gerencia a interface para uma janela de jogos. |
| Org.Apache | Métodos para gerência das funções do APACHE |
| Org.W3c | Gerencia de métodos relacionados a World Wide Web. |
| Org.Xml | Métodos de gerência de XML (eXtensible Markup Language) |
| System | Gerencia funções do sistema Android, como Zona de hora, e tipos de variáveis. |

Tabela 2. Lista de APIs suportadas pelo Xamarim no iOS

|  |  |
| --- | --- |
| **IOS** | **Descrição e versões suportadas** |
| Accelerate | Fornece código de alto desempenho para a manipulação de dados de imagem. |
| Accounts | Fornece classes para armazenamento de credenciais de contas de redes sociais e UIs para autenticação. |
| AddressBook | Fornece acesso ao centro de 'Contactos' contatos banco de dados. |
| AdSupport | Fornece um identificador de dispositivo, apenas para ser usado para servir anúncios. |
| AssetsLibrary | Fornece acesso a fotos e vídeos gerenciados pelo sistema. |
| AudioToolbox | Oferece aulas de nível médio para gravação e reprodução de áudio. |
| AudioUnit | Proporciona acesso de áudio de baixo nível, permitindo o acesso a unidades de áudio e cadeias de processamento de áudio. |
| AVFoundation | Fornece classes de alto nível para a gravação e reprodução de áudio. |
| AVKit | Proporciona vistas padrão para a captura de mídia e reprodução do conteúdo visual. |
| CloudKit | Fornece acesso a iCloud. |
| Contacts | API que manipula os contatos do dispotivo. |
| CoreAnimation | Fornece acesso à estrutura de animação subjacente que faz o UIKit funcionar. |
| CoreAudioKit | Oferece visão genérica que pode ser estendido para fornecer interfaces de usuário para as unidades de áudio. |
| CoreBluetooth | Fornece classes para a comunicação Bluetooth. |
| CoreData | Fornece serialização objeto gráfico e persistência. |
| CoreFoundation | Fornece acesso a  APIS CocoaTouch de baixo nivel. |
| CoreGraphics | Fornece acesso a  APIS C-based 2D graphics. |
| CoreImage | Fornece processamento de imagem acelerada por hardware. |
| CoreLocation | Fornece classes para rastreamento e monitoramento local, as regiões (incluindo iBeacons), e geocodificação. |
| CoreMedia | Fornece classes de baixo nível para a gestão e reprodução de áudio e vídeo. |
| CoreMidi | Fornece classes para interagir com o subsistema MIDI. |
| CoreMotion | Permite o acesso a dados de movimento, incluindo a aceleração do dispositivo e orientação, altitude, bússola, e pedômetro. |
| The CoreServices | Fornece classes de baixo nível relativos à rede com CFNetwork  (métodos necessários para acessar as configurações de proxy globais do sistema  e resolver uma lista de proxies para usar para se conectar a um URL.) |
| CoreSpotlight | Permite que os aplicativos para adicionar dados para o índice de pesquisa do sistema. |
| CoreTelephony | Fornece classes que fornecem informações sobre o provedor de rede celular, bem como informações sobre a chamada atual. |
| CoreText | Fornece classes de baixo nível para colocar para fora e processamento de texto. |
| CoreVideo | Fornece classes de baixo nível para a manipulação de buffers de vídeo. |
| Dialog | Criação simplificada de interfaces de usuário. |
| Dialog.Utilities | Funções de utilitário de propósito geral que são usados ​​por  MonoTouch.Dialog UIs, mas também podem ser utilizados por outras aplicações. |
| EventKit | Fornece classes para manipular o calendário do sistema. |
| ExternalAccessory | Fornece classes para comunicação com acessórios do aparelho. |
| Foundation | Fornece classes para comunicar com os acessórios ligados ao dispositivo.  Oferece aulas de nível mais baixo no mundo do Objective-C. Muitas vezes,  essas classes têm equivalentes na Biblioteca de classes .NET Base. |
| GameKit | Fornece uma comunicação fácil entre dispositivos iPhone / iPod Touch. |
| GameplayKit | Fornece classes para de alto nível a mecânica do jogo-playing como pathfinding, motores de regras, e os adversários AI. |
| GLKit | Fornece uma estrutura para configurar e utilizar o OpenGL. |
| HealthKit | Fornece armazenamento seguro de dados e recuperação de informação relacionada com a saúde. |
| HomeKit | Fornece controle e acesso a dispositivos de automação doméstica. |
| iAd | iAd Plataforma de Publicidade. |
| ImageIO | Fornece classes para entrada e saída de imagem. |
| JavaScriptCore | Fornece classes que avaliam o código JavaScript, analisar JSON, e permitem a manipulação do ambiente JS. |
| LocalAuthentication | Fornece suporte para TouchID. |
| MapKit | Fornece mapas incorporáveis, anotações e direções. |
| MediaAccessibility | Fornece suporte para legendas de vídeo. |
| MediaPlayer | Fornece classes de alto nível para reprodução padrão de áudio ou vídeo. |
| MediaToolbox | Fornece classes de baixo nível para trabalhar com torneiras de processamento de áudio. |
| MessageUI | Fornece uma interface de usuário padrão para composição e-mail. |
| Metal | Fornece renderização 3D acelerada por GPU e computação de dados em paralelo. |
| ModelIO | Gerencia Métodos usados por engines 3D |
| MonoTouch | Definições globais do MonoTouch |
| MultipeerConnectivity | Fornece conexões (WiFi / Bluetooth) de mensagens e de dados peer-to-peer locais. |
| NetworkExtension | Fornece classes para configurar e se conectar a servidores VPN. |
| NotificationCenter | Fornece classes para criar e atualizar os widgets. |
| ObjCRuntime | Fornece classes para a ligação de tipos de Objective-C com o Mono gerenciado tempo de execução. |
| OpenTK | Métodos referentes a funções OpenGL, suporte para OpenGLES 1.1 / OpenGLES 2.0/OpenGLES 3.0 |
| PassKit | Fornece classes para criação e manipulação de passes, incluindo o pagamento e autorização passa. |
| Photos | Fornece classes para manipular ativos de foto e vídeo geridos pela aplicação Fotografias, incluindo iCloud fotos. |
| PushKit | Fornece registro de eficiência energética para as notificações de VoIP. |
| QuickLook | Fornece capacidades de visualização de documentos externos. |
| ReplayKit | Oferece gravação de usuário de aplicações. |
| SafariServices | Fornece a capacidade de adicionar itens à lista de leitura Safari. |
| SceneKit | Fornece alto nível, baseados em cena gráfico gráficos 3D. |
| Security | Fornece acesso ao key chain. |
| Social | Fornece para compor e compartilhar dados em redes sociais. |
| SpriteKit | Fornece uma biblioteca de gráficos 2D, animação e física. |
| StoreKit | Fornece compras de conteúdo dentro das aplicações. |
| System | Gerencia funções do sistema IOS, como Zona de hora, e tipos de variáveis. |
| Twitter | Fornece composição, pedido, e os resultados de mensagens do Twitter. |
| UIKit | Fornece as classes interface principal Xamarin.iOS. |
| VideoToolbox | Fornece classes de baixo nível para codificação e decodificação de vídeo |
| WatchConnectivity | Métodos referentes ao apple watch |
| WatchK | Fornece a capacidade de programar o relógio Apple. |
| WebKit | Fornece controle de navegador da Web. |

## 1.14. Quais os sensores que podem ser utilizados pelo Xamarim?

Abaixo listamos os sensores suportados pelo Xamarim categorizados por sistema.

**Disponíveis para IOS e ANDROID:**

* Acelerômetro
  + Sensor responsável por prover a força aplicada ao dispositivo.
* Luminosidade
  + Sensor de luminosidade, responsável por aferir a intensidade de luz no ambiente.
* Gravidade
  + Retorna a intensidade e o vetor da força gravitacional.
* Temperatura
  + Retorna a temperatura ambiente.
* Detector de Passos.
  + Detecta os passos do usuário.
* Giroscópio
  + Retorna a orientação do dispositivo via campo gravitacional.

**Apenas disponíveis para Android.**

* Frequência cardíaca
  + Sensor biométrico, retorna a frequência batimentos cardíacos.
* Campo magnético
  + Sensor que retorna informações sobre os campos magnéticos identificados.

### Barômetro

* + Retorna a pressão ambiente.
* Proximidade
  + Identifica corpos próximos ao aparelho por infravermelho.
* Higrômetros
  + Retorna a umidade do ar.

**Apenas disponíveis para iOS.**

* Magnetômetro
  + Usado para medir a intensidade, dire(c)ção e sentido de campos magnéticos em sua proximidade.
* Altímetro.
  + Retorna altitude do dispositivo.

## 1.15. Cite dois exemplos de aplicativos context-awareness que podem ser desenvolvidos com Xamarim, e justifique com as informações das tabelas de API e sensores informados acima

Atualmente podemos listar o AeroScout, um aplicativo feito em parceria com a Cisco e a Tallahassee Memorial Healthcare (THC), que através de banco de dados, mapas em planta baixa disponibilizados pelo motor de contexto da Cisco denominada Cisco Mobility Services Engine (MSE), sensores de GPS e câmera para análise de tags, possibilita a busca de instrumentos e materiais necessários nos depósitos mais próximos para aplicações de urgência, como no caso de cirurgias de emergência em hospitais, somente tomando como parâmetro a ação que o usuário fará. Este processo elimina significativamente a aquisição de materiais armazenados em múltiplos locais, mostrando informações referentes ao estado de uso e prontidão, como temperatura, tempo de armazenamento, grau de periculosidade, dentre outros. Todo esse processo é monitorado e atualizado a cada 10 minutos e qualquer modificação fora do padrão gera um alerta para um funcionário especializado para que este verifique o problema e solucione.

Também existe um projeto de desenvolvimento de aplicativo sensível ao contexto na empresa Appear, de origem sueca, no qual eles utilizam um motor de contexto da própria empresa, denominada Appear Context Engine (ACE), onde ele possui regras para determinar, através de uma busca de informação sensível/confidencial, onde e quando esta informação pode ser obtida, e para quais pessoas é permitido, evitando o vazamento de informações dos funcionários de empresas aéreas. O aplicativo também oferecerá módulos de sincronização de informações, gerenciamento de dispositivos e aquisição e disposição de informações, todos baseados nas regras e estruturas da ACE.

## 1.16. Cite dois exemplos de aplicativos context-awareness que NÃO podem ser desenvolvidos com Xamarim, e justifique com as informações das tabelas de API e sensores informados acima

Atualmente ocorreu um grande avanço nas tecnologias que auxiliam o mercado agrônomo, e dentro desta esfera, faz-se necessário a medição constante de condições meteorológicas e/ou atmosféricas. Estas medições são feitas através de barômetros e higrômetros, sensores que, no escopo da plataforma, só são suportados no Android. Desta forma, aplicações móveis como o AgroSmart, que monitora e notifica os usuários sobre condições terrestres e atmosféricas para o plantio, não podem ser implementadas para o sistema iOS, visto que não há suporte da plataforma Xamarim para estes sensores.

Outra aplicação que provavelmente não será capaz de ser reproduzida/implementada pelo dispositivo móvel, especificamente no iOS, seria o Cardiógrafo. Esta aplicação foi criada para aferir o ritmo cardíaco dos usuários de wearables, gerando assim dados que são dispostos em gráficos de perfis pessoais, podendo ser acompanhados diariamente por amigos ou profissionais da área médica. No entanto, Devido ao fato de o Xamarim não dar suporte ao sensor de batimentos cardíacos (heartbeat sesor) no IOS, torna-se inviável a implementação desse tipo de app na plataforma.

# 2. Referências

Schmidt, A. “Context-Aware Computing,” <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/context-aware-computing-context-awareness-context-aware-user-interfaces-and-implicit-interaction>, acessado no dia 27 de fevereiro de 2016.

Vieira, V., Souza, D., Carolina Salgado, A. e Tedesco, P. Uso e Representa Uso e “Representa ção de Contexto ão de Contexto em Sistemas Computacionais” <http://www.cin.ufpe.br/~ggc/IC/referencias/Minicurso-Contexto2006.pdf> , acessado no dia 27 de fevereiro de 2016.

Xamarin Inc., “Android Hardware API Reference” <https://developer.xamarin.com/api/namespace/Android.Hardware/> Acessado no dia 23 de fevereiro de 2016.

Xamarin, Inc “Mobile App Development & App Creation Software”. <https://xamarin.com> Acessado no dia 19 de fevereiro de 2016.

Guthrie, G. (2016), “Microsoft to acquire Xamarim and empower more developers to build apps on any device”

<https://blogs.microsoft.com/blog/2016/02/24/microsoft-to-acquire-xamarin-and-empower-more-developers-to-build-apps-on-any-device/> Acessado no dia 08 de março de 2016.

Schultz, B., ComputerWorld.com, (2009), “Context-aware mobility: What is it and how will it change the business world?” <http://www.computerworld.com/article/2525703/mobile-wireless/context-aware-mobility--what-is-it-and-how-will-it-change-the-business-world-.html>, acessado no dia 08 de março de 2016.

Wilde, E., (2013) “How does Xamarim iOS a.k.a. MonoTouch work” <http://blogs.edwardwilde.com/2013/04/13/how-does-xamarin-ios-aka-monotouch-work/> , acessado no dia 09 de março de 2016.

“AgroSmart: Cultivo Inteligente”, <http://www.agrosmart.com.br> , acessado no dia 23 de março de 2016