

# APRESENTANDO A PLATAFORMA XAMARIN

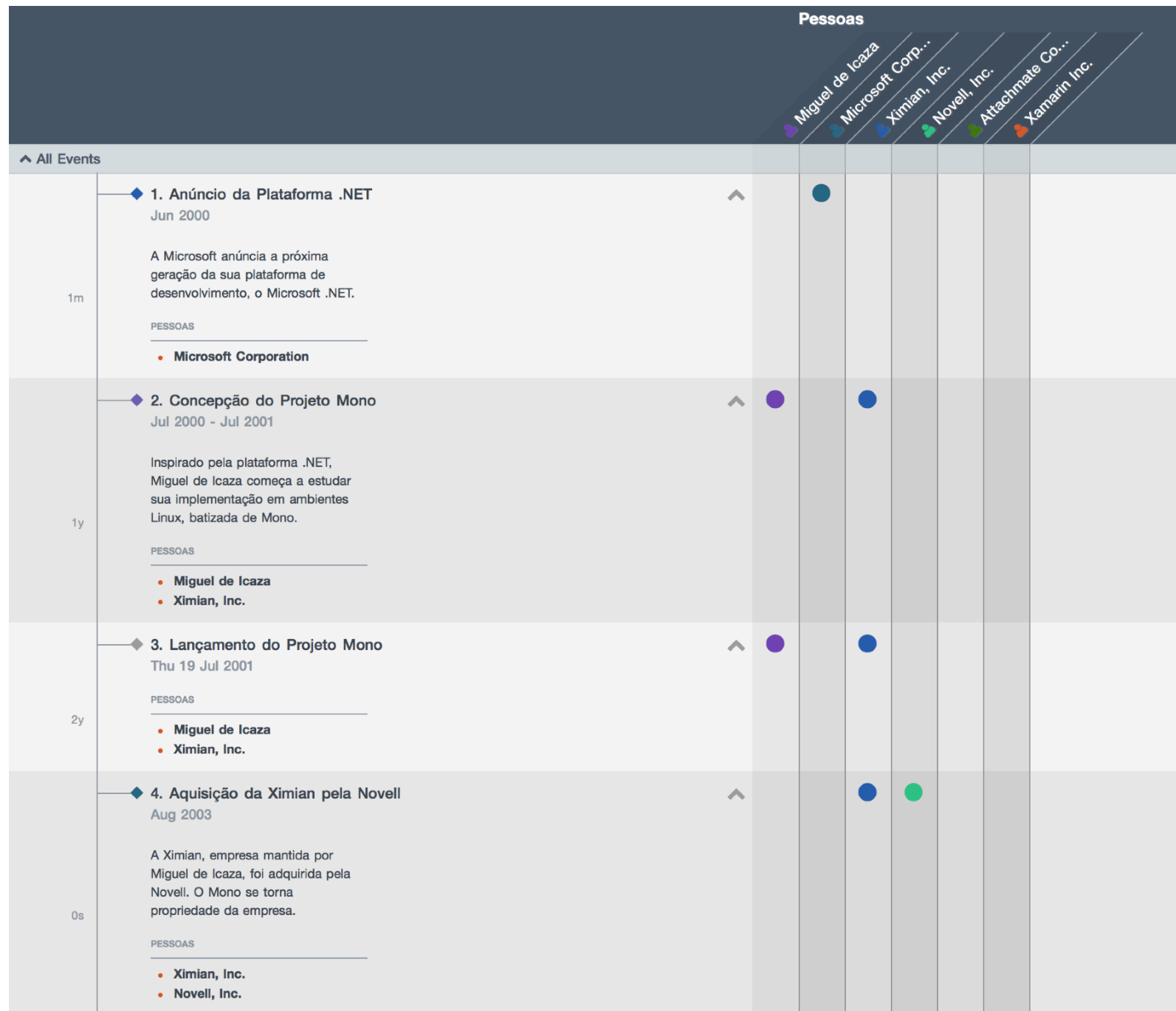
---

# HISTÓRIA

---



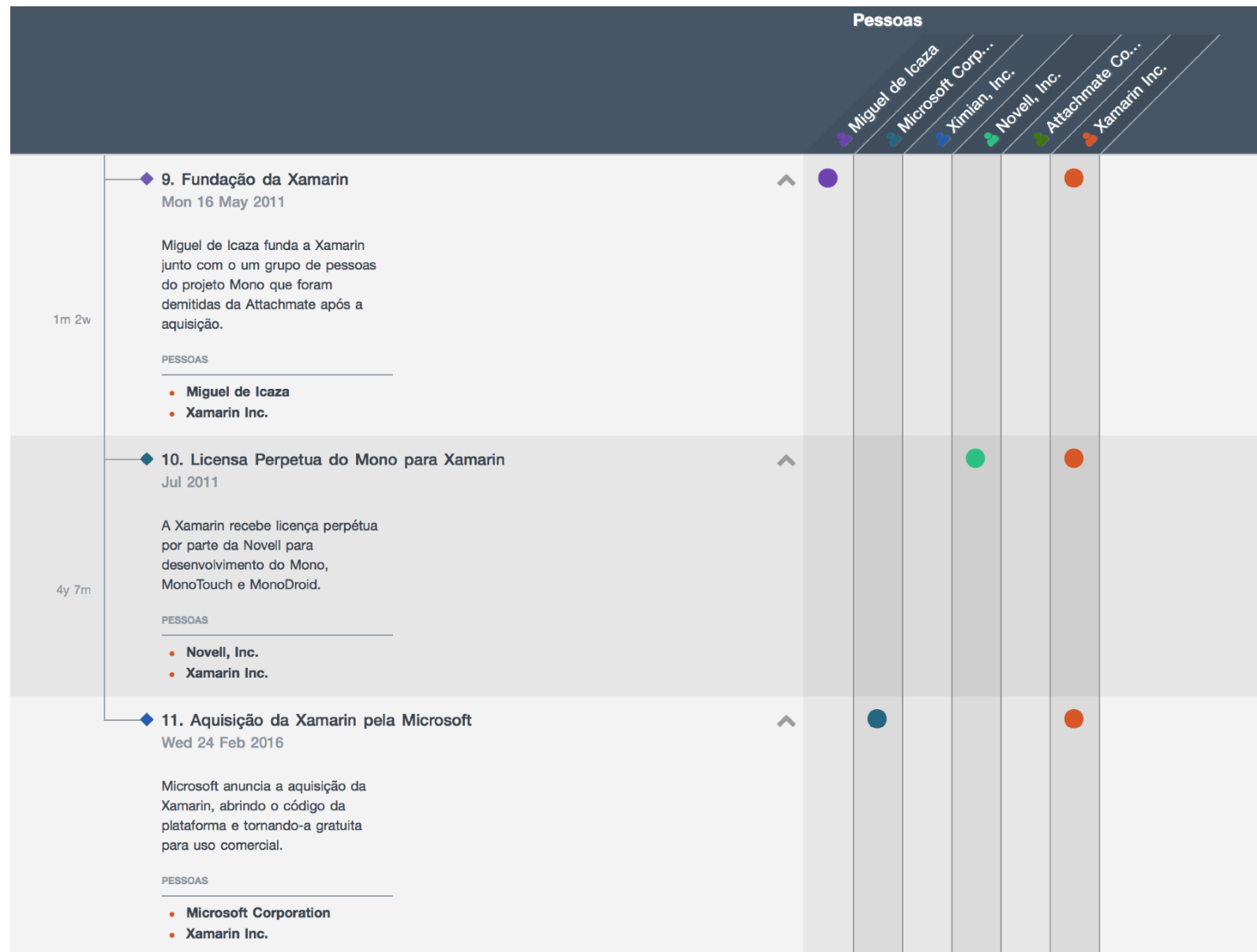
# Linha do tempo



# Linha do tempo



# Linha do tempo



# Xamarin Inc.

---

Oferecia a plataforma Xamarin como um produto comercial.

Cobrava mensalidade de seus desenvolvedores, sua principal barreira de entrada.

# Aquisição pela Microsoft

---

Anunciada em Março de 2016.

Tornou a plataforma Xamarin livre para uso comercial e abriu seu código fonte.

Esta investindo pesadamente na evolução da plataforma e de suas ferramentas.

# ARQUITETURA DO XAMARIN

---





# Visão geral

---

O Xamarin utiliza assemblies .NET que fazem binding com as bibliotecas nativas de cada plataforma que ele suporta.

Isso possibilita escrever um programa em .NET, com C# ou F#, que faz referência a essas bibliotecas.

O código IL gerado é compilado nativamente nas plataformas suportadas, produzindo código binário totalmente Nativo.

# Mono

---



Projeto livre e Open Source liderado pela Xamarin.

Criado por Miguel de Icaza e mantido inicialmente Xiamin.

# Mono

---

Implementação de código aberto multi-plataforma do .NET Framework.

CLR (Common Language Runtime)

Compiladores

Ferramentas

Base das implementações do Xamarin.

# COMPONENTES DA PLATAFORMA

---





# XAMARIN.IOS

---

# Visão Geral

---

Primeira peça do que conhecemos hoje como "Plataforma Xamarin".

Lançado pela Novell em 14/09/2009 com o nome MonoTouch.

Ponte entre o "mundo .NET" e o "mundo iOS".

Conjunto de assemblies .NET que fazem bind com os frameworks disponíveis no iOS.

# Arquitetura

---

Cada release do iOS tem uma versão do Xamarin.iOS que o acompanha.

Os frameworks são 100% mapeados e disponibilizados para uso com C#.



Mono Runtime

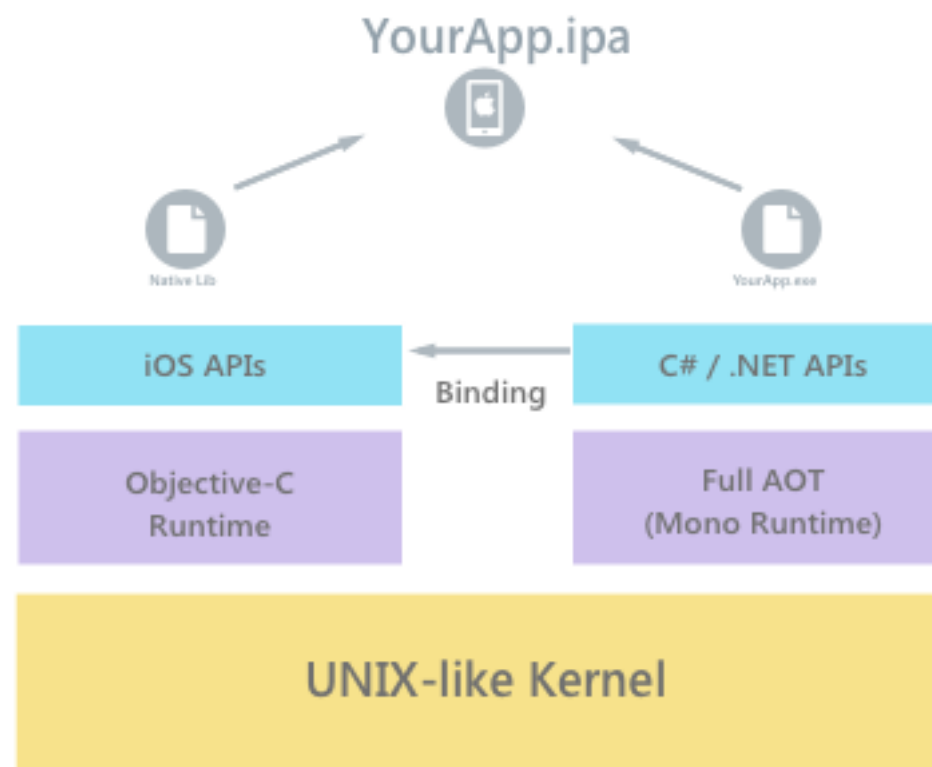
AOT

# Arquitetura

Mono  
Runtime

Cada IPA inclui uma versão pré-compilada do CLR.

Esse Runtime é responsável pela execução do código .NET





# Arquitetura

---

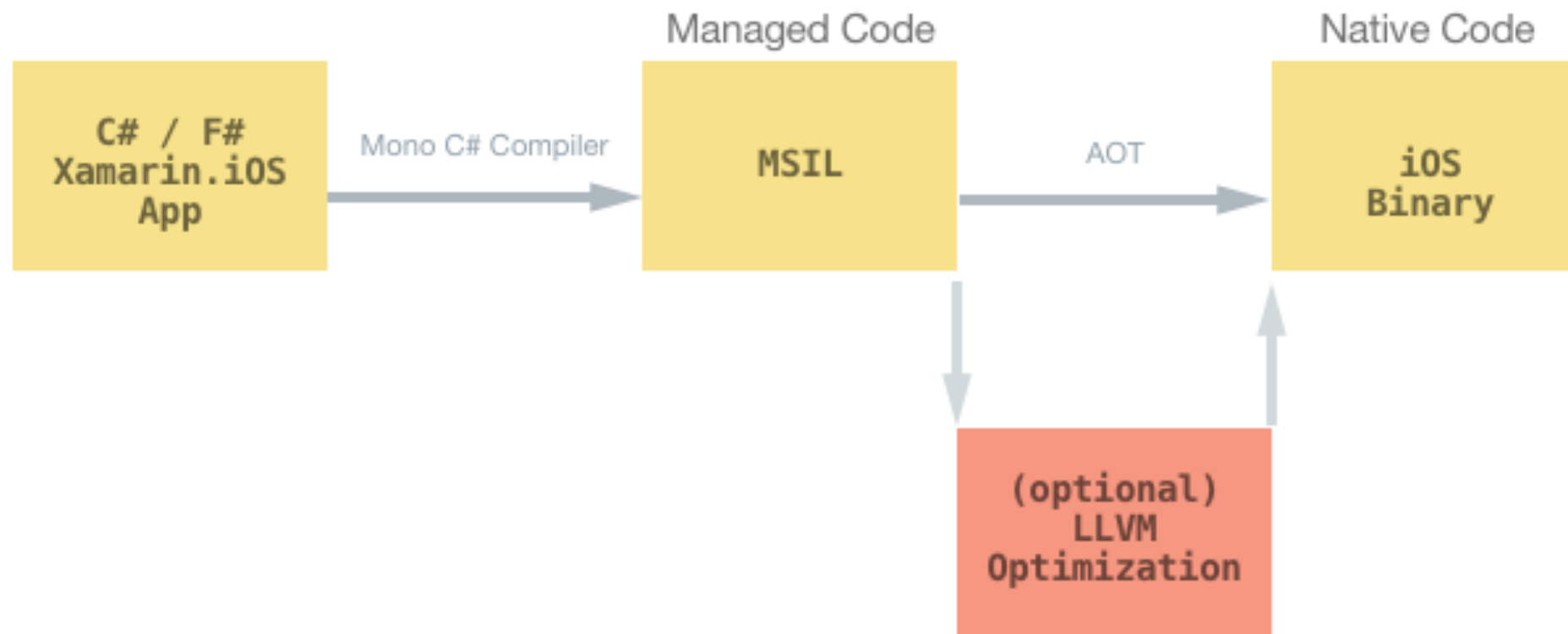


Sigla de "Ahead of Time Compilation".

Se contrapõe ao JIT do .NET para produzir código .NET executável em ambiente iOS.

Todo código .NET produzido para um programa iOS é compilado nativamente com AOT.

# Arquitetura



Possibilita superar as restrições de execução de código dinâmico da Apple para o iOS

# Mac Build Host

---

Apps iOS são compilados exclusivamente em ambiente macOS.

Apps iOS podem ser desenvolvidos em Windows com Visual Studio.

Porém é necessário um Mac com o Xcode e o Xamarin para compilar e executar os programas.



# XAMARIN.ANDROID

---

# Visão geral

---

Segunda peça da atual "Plataforma Xamarin".

Lançado pela Novell em 06/04/2011 suportando o Android 2.2.

Ponte entre o "mundo NET" e o "mundo Android".

Proporciona um ambiente de execução sobre o Kernel Unix do Android.

# Arquitetura

---

O Xamarin.Android usa o JNI para mapear código Android e Managed Code.



Android  
Callable  
Wrappers

Managed  
Callable  
Wrappers

# Arquitetura

---



## Android Callable Wrappers

Permite o Android chamar Managed Code

Usado quando algum código Android vai executar um código próprio do nosso API, escrito em C#.

**Exemplo:** tratamento de clique de um botão

# Arquitetura

---



## Managed Callable Wrappers

Permite o Managed Code chamar código Android.

Usado quando o código do app escrito em C# interage com um código nativo Android.

Gerencia o mapeamento de tipos entre o Android e o Managed Code.

### Exemplos:

Criando um componente visual do Android;

Implementando uma interface de um Listener Android.



# Arquitetura

---

Framework Java que possibilita conversão com código de outras plataformas.



JNI: Java Native Interfaces



# XAMARIN.FORMS

---

# Visão geral

---

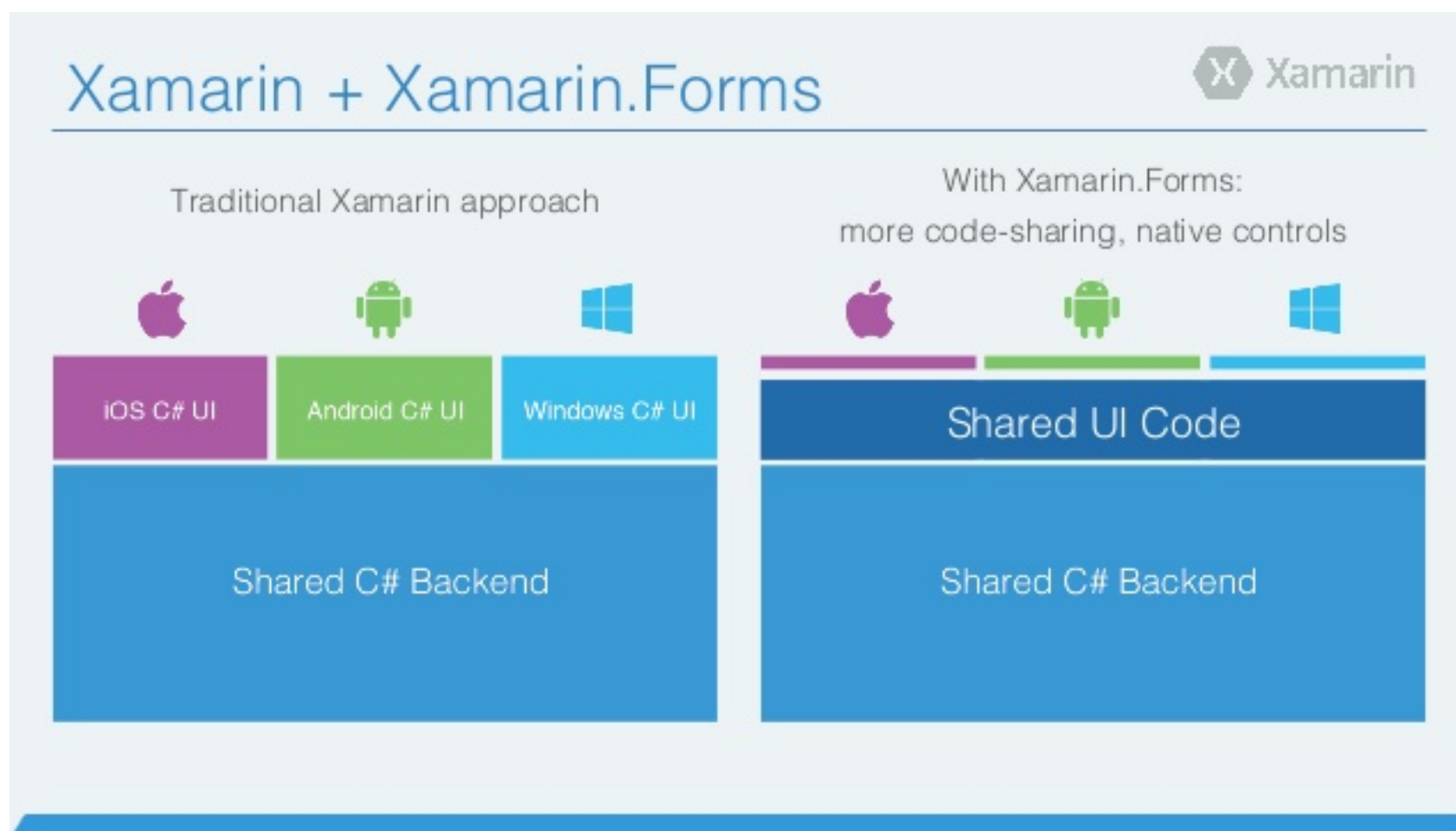
Uma biblioteca de abstração para criação de apps multi-plataforma com Xamarin.

Lançado em 28/05/14 com o Xamarin 3.

Construído sobre o Xamarin.iOS e Xamarin.Android.

Oferece uma forma unificada para construir a interface dos apps se aproveitando de seus recursos nativos.

# Visão geral



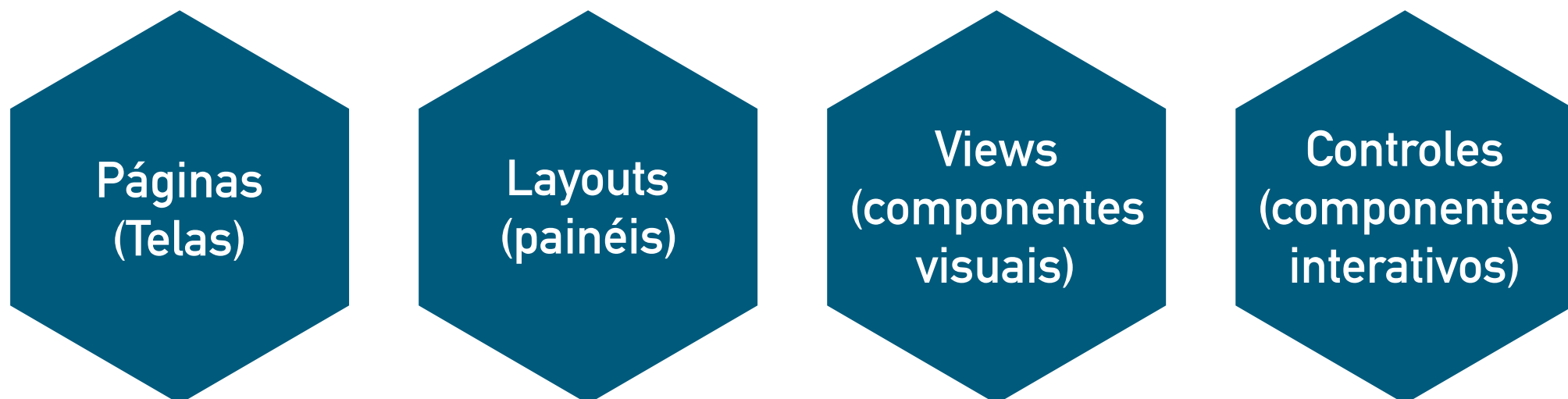
## Arquitetura

Sua utilização possibilita re-aproveitamento de até 95% de código entre plataformas

# Funcionamento

---

O Xamarin.Forms oferece uma série de componentes de alto nível como:



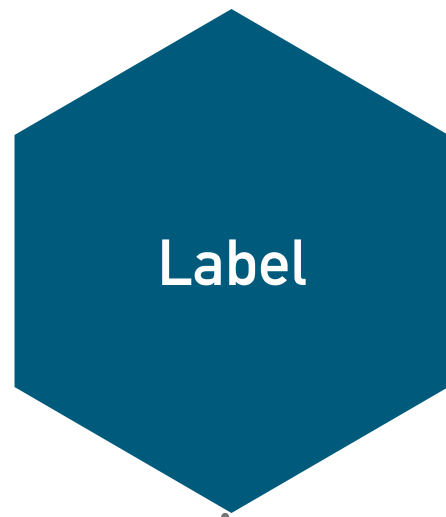
Um app é construído sobre esses componentes.

Durante sua execução esses componentes são renderizados com objetos nativos de cada plataforma.

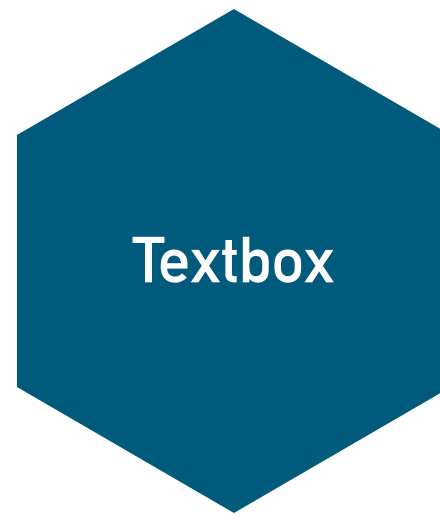
# Funcionamento

---

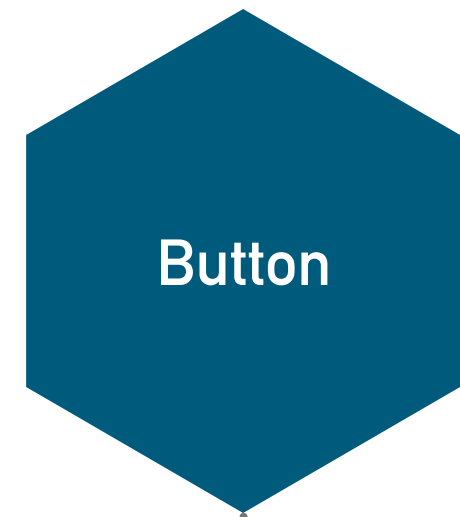
Exemplos:



Android: `TextView`  
iOS: `UILabel`



Android: `EditText`  
iOS: `UITextField`



Android: `Button`  
iOS: `UIButton`

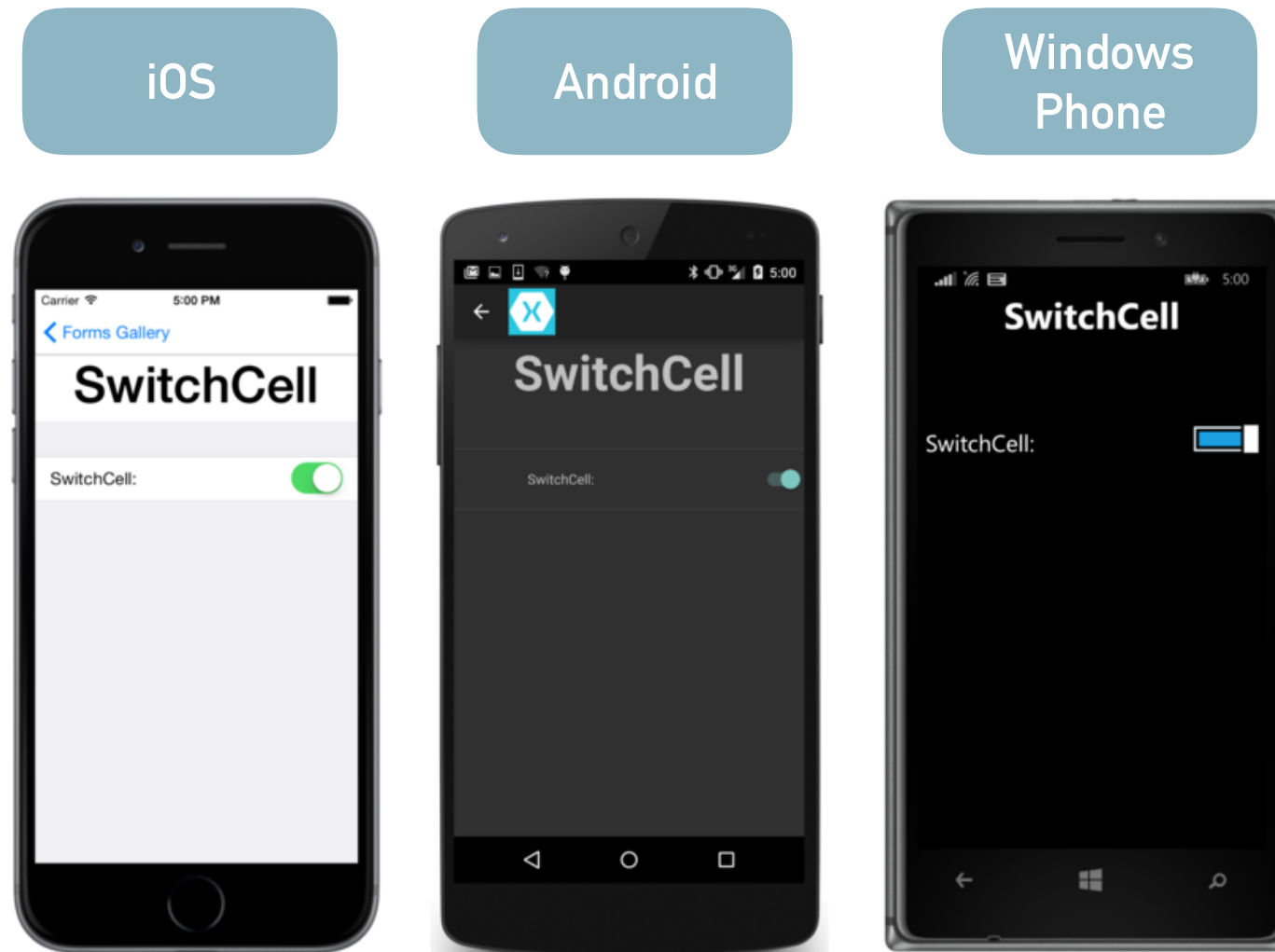
# Funcionamento

---

Há diversas formas de personalizar elementos próprios de cada plataforma.

Versões mais recentes permitem referenciar objetos próprios de cada plataforma.

# Exemplos

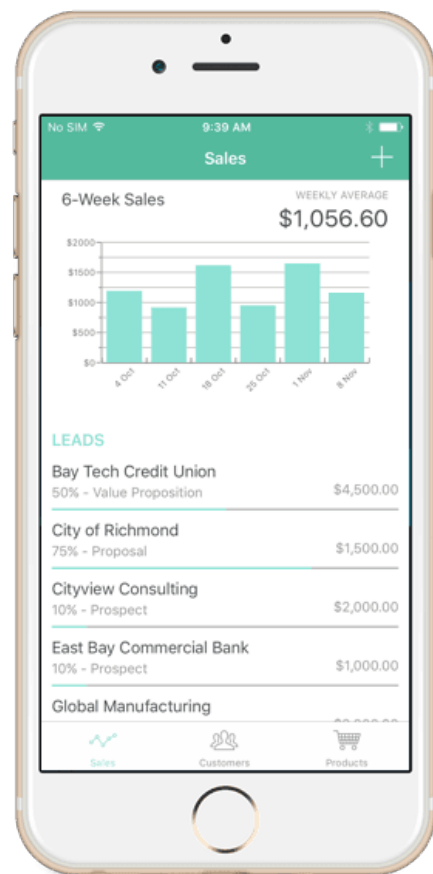


Renderização de um Switch em cada plataforma.

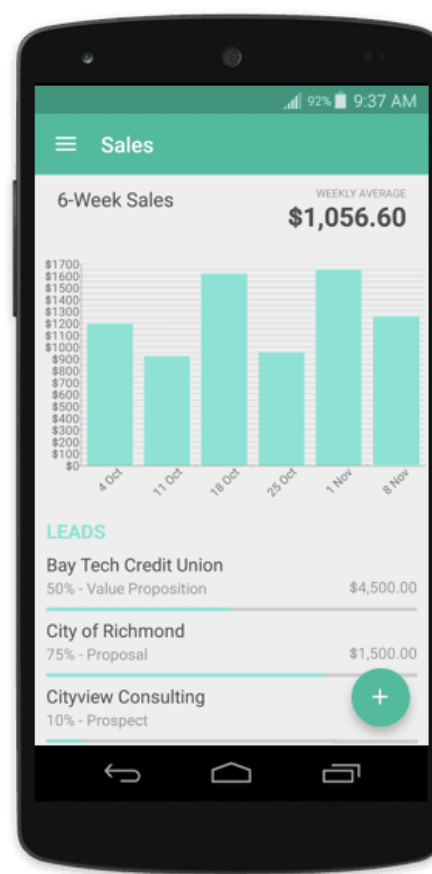


# Exemplos

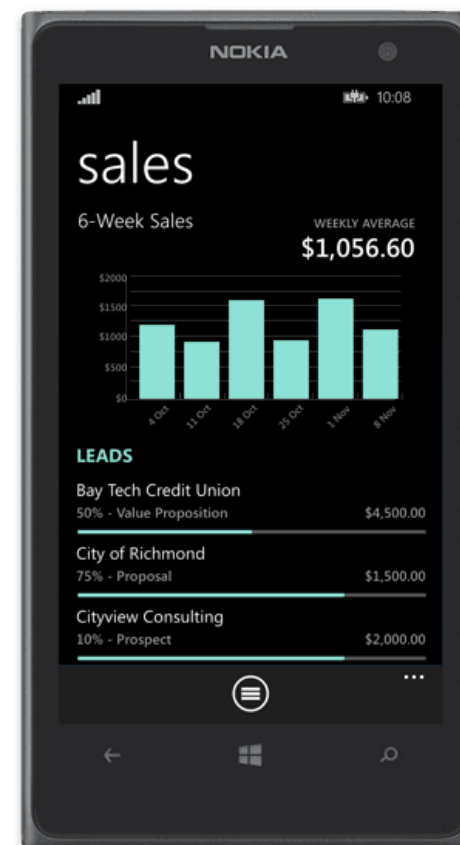
iOS



Android



Windows Phone



Interface complexa renderizada nativamente em cada plataforma.

# FERRAMENTAS DE DESENVOLVIMENTO

---

# Visão geral

---

Além das vantagens proporcionadas pelo compartilhamento de código, o Xamarin também se aproveita de um rico conjunto de ferramentas disponíveis para desenvolvimento na plataforma.

O Xamarin traz versatilidade para seus desenvolvedores que podem obter por trabalhar em PC's ou Macs.



# IDE's

---



Visual Studio 2017

O Xamarin se integra ao Visual Studio 2017 para PC's através do "Visual Studio Tools for Xamarin"

Essa ferramenta proporciona os seguintes recursos:

- Templates de Projetos do Xamarin para Android, iOS, UWP e Xamarin.Forms.
- Xamarin Workbooks & Inspector
- Xamarin Profiler
- Remote iOS Simulator

# IDE's - Visual Studio 2017

---



Visual Studio 2017

## Principais Vantagens

- Todos os recursos disponíveis no Visual Studio 2017
- É possível trabalhar em soluções que integrem outros tipos de projetos como ASP.NET ou WCF.
- Aproveita o ambiente dos desenvolvedores .NET

# IDE's - Visual Studio 2017

---



Visual Studio 2017

## Desvantagens

- É necessário um Mac disponível na rede para desenvolvimento iOS.
- A compilação e uso de recursos em projetos iOS tende a ser mais lenta.

# IDE's

---



Visual Studio for Mac

O Visual Studio for Mac é o rebranding do Xamarin Studio, ambiente de desenvolvimento multi-plataforma baseado no MonoDevelop.

O ambiente era destinado primariamente para o desenvolvimento de projetos Xamarin (iOS, Android e Forms), mas tem incorporado novas funcionalidades como o suporte .NET Core.

A tendência é que com o tempo é se equiparar em recursos com o Visual Studio para PC.

# IDE's

---



Visual Studio for Mac

## Principais Vantagens

- Desenvolvimento através do macOS com ferramenta adaptada aos paradigmas desse sistema.
- Oferece melhor performance para desenvolvimento iOS e macOS.
- Compatível com projetos .NET Core.



# IDE's

---



Visual Studio for Mac

## Desvantagens

- Não oferece todos os recursos disponíveis no Visual Studio para Windows.
- Incompatível com os plug-ins do Visual Studio para Windows.

# Disponibilidade

---

- Ambas versões do Visual Studio estão disponíveis para download gratuito do “Community Edition”, que contém todos os recursos fundamentais para desenvolvimento de apps.
- Alguns recursos como o Xamarin Profiler estão disponíveis somente para assinantes do MSDN.
- Há uma infinidade de recursos gratuitos para desenvolvedores através do Visual Studio Dev Essentials



# IDE's

---



JetBrains Rider

O Rider é uma IDE comercial alternativa ao Visual Studio para desenvolvimento .NET. Criado e mantido pela JetBrains, dona do ReSharper (plugin de produtividade muito popular para Visual Studio) e do IntelliJ IDEA (que é a base do Android Studio), traz uma solução completa para desenvolvimento de apps, inclusive Xamarin.

# IDE's

---



JetBrains Rider

## Principais Vantagens

- Multiplataforma.
- Compatível com o formato de projeto do Visual Studio.
- Inclui todos os recursos disponíveis no ReSharper

# IDE's

---



JetBrains Rider

## Desvantagens

- Não há licença para uso gratuito, o produto tem de ser adquirido.

# IDE's

---



Visual Studio Code

O Visual Studio Code é um editor de código multi-plataforma (Windows, macOS e Linux) open-source criado e mantido pela Microsoft. Embora carregue o nome Visual Studio, ele foi criado para competir diretamente com o Atom, Sublime e outros editores de código leves, estando mais próximo desses do que de seu irmão maior.

Baseado em um modelo de plugins, o Code hoje tem a maior comunidade de componentes para praticamente todas as linguagens e plataformas do mercado, incluindo o Xamarin.

# IDE's

---



Visual Studio Code

## Principais Vantagens

- Multiplataforma
- Open-source
- Extremamente leve e simples
- Rico ecossistema de componentes
- Desenvolvimento acelerado com atualizações mensais.

# IDE's

---



Visual Studio Code

## Desvantagens

- Para desenvolvimento Xamarin, oferece poucas ferramentas para desenvolvimento de interfaces.