

Aula 2: .NET: implementações e tecnologias

Introdução ao Ecossistema .NET &
Documentação



Objetivos

1. Desenvolvimento multiplataforma com .NET: contexto e conceitos
 2. Entendendo .NET Framework
 3. Entendendo .NET / .NET Core
 4. Entendendo Xamarin
-

Aula 2

Desenvolvimento multiplataforma com .NET: contexto e conceitos

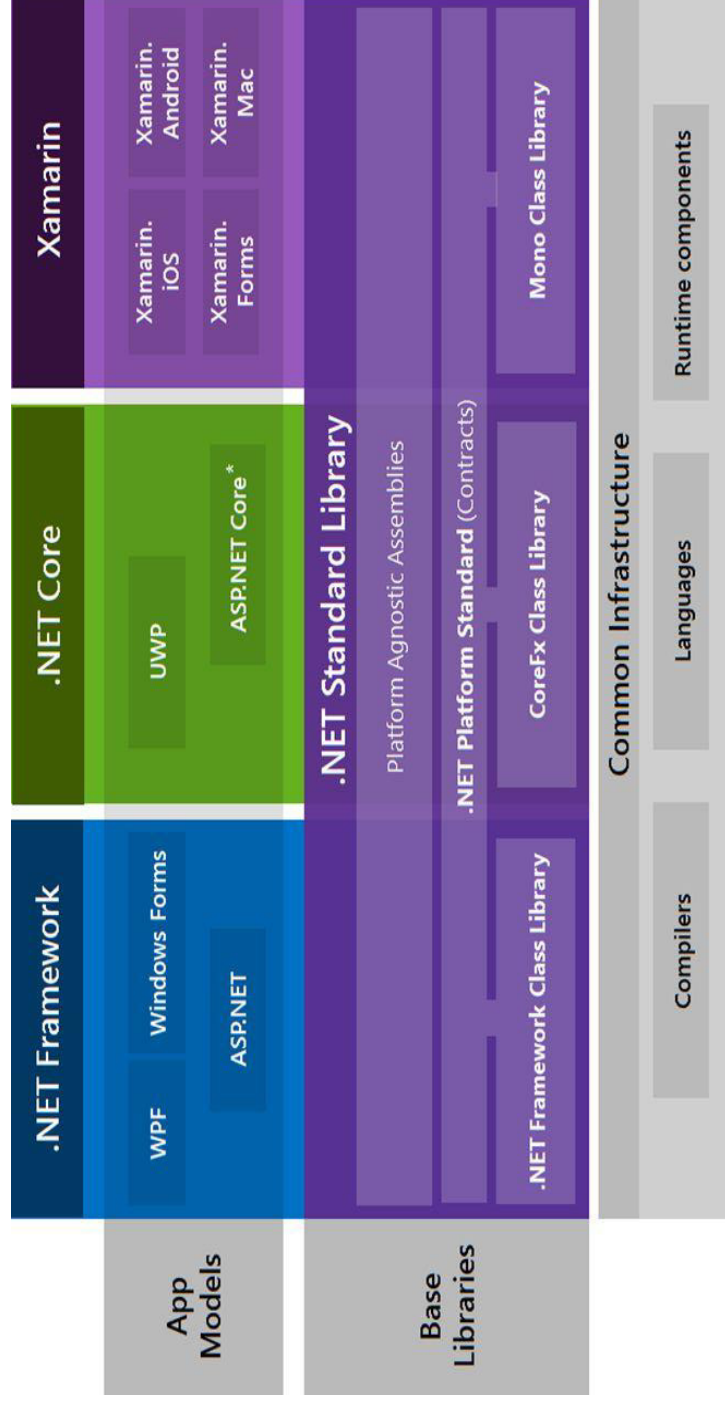
Introdução ao Ecossistema .NET & Documentação

A plataforma .NET é gratuita?

→ Open-source e gratuito - > [.NET is open source on GitHub | .NET \(microsoft.com\)](https://github.com/dotnet/core?ref=dotnet.microsoft.com)

- ♦ Distribuição livre;
 - ♦ Acesso ao código-fonte;
 - ♦ Permissão para criação de trabalhos derivados;
 - ♦ Integridade do autor do código-fonte;
 - ♦ Não discriminação contra pessoas ou grupos;
 - ♦ Não discriminação contra áreas de atuação;
 - ♦ Distribuição da licença;
 - ♦ Licença não específica a um produto;
 - ♦ Licença não restritiva a outros programas;
 - ♦ Licença neutra em relação à tecnologia.
-

Abordagem multiplataforma



Fonte: .NET Blog(2016)

Especificação das APIs que são comuns às implementações
.NET -> [.NET Standard | Common APIs across all .NET implementations \(microsoft.com\)](https://docs.microsoft.com/en-us/dotnet/standard/compatibility/compatibility-matrix)

Version: <input type="text" value=".NET Standard 2.0"/>		Available APIs: 32,638 of 37,118	
.NET Implementation	Version Support		
.NET and .NET Core	× 1.0 × 1.1 ✓ 2.0 ✓ 2.1 ✓ 2.2 ✓ 3.0 ✓ 3.1 ✓ 5.0 ✓ 6.0		
.NET Framework	× 4.5 × 4.5.1 × 4.5.2 × 4.6 ✓ 4.6.1 ✓ 4.6.2 ✓ 4.7 ✓ 4.7.1 ✓ 4.7.2 ✓ 4.8		
Mono	× 4.6 ✓ 5.4 ✓ 6.4		
Xamarin.iOS	× 10.0 ✓ 10.14 ✓ 12.16		
Xamarin.Android	× 7.0 ✓ 8.0 ✓ 10.0		
Universal Windows Platform	× 8.0 × 8.1 × 10.0 ✓ 10.0.16299 ✓ TBD		
Unity	✓ 2018.1		

Aula 2

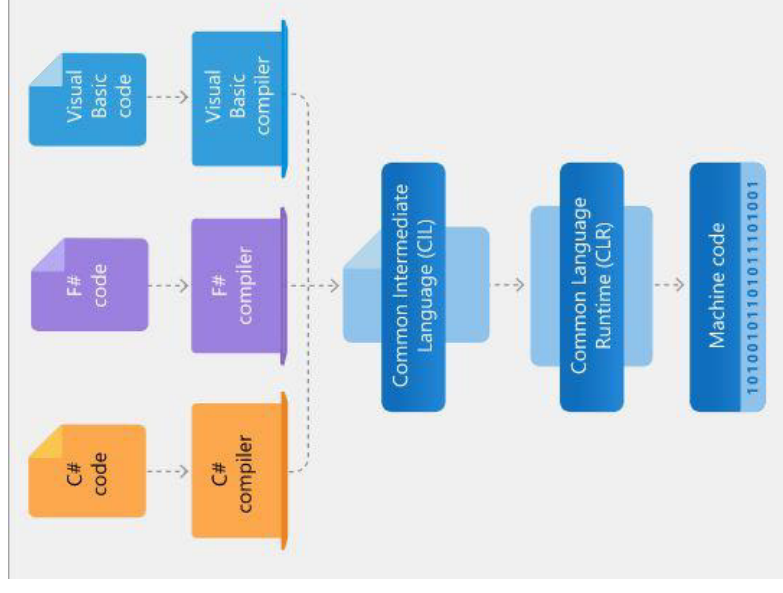
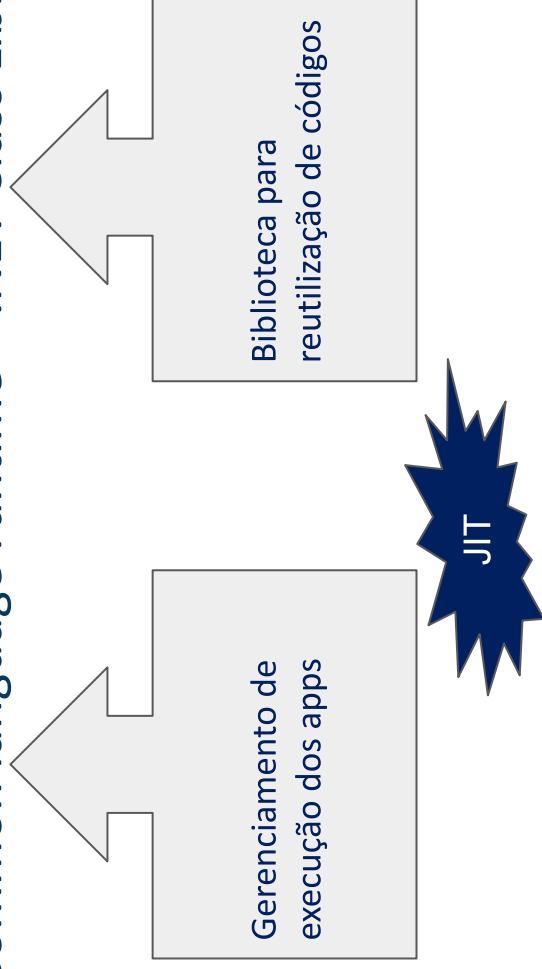
Entendendo .NET Framework

Introdução ao Ecossistema .NET &
Documentação



O que é .NET Framework?

- Ambiente de desenvolvimento e execução focado em Windows
- Common language runtime + .NET Class Library





Características principais

- Gerenciamento de memória
- Sistema de tipos comum - > Tipos definidos pelo próprio framework
- Grande e específicas bibliotecas de classes utilitárias
- Interoperabilidade entre linguagens
- Compatibilidade de versões
- Uso do .NET Standard

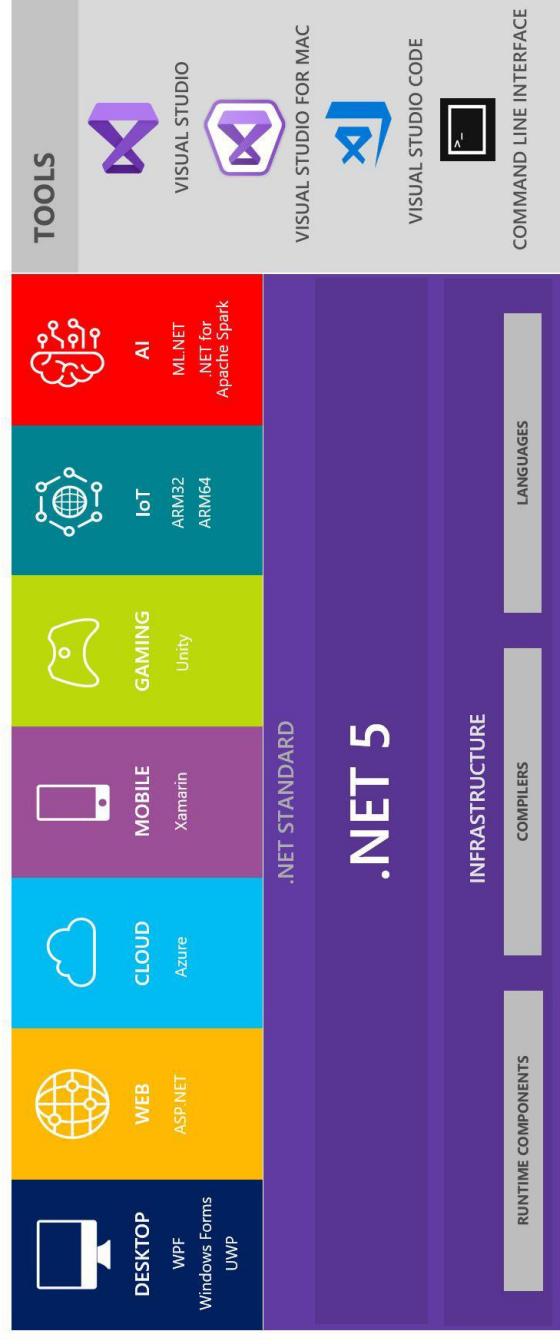
ASP.NET
ADO.NET
WCF
WPF

Aula 2

Entendendo .NET/.NET Core

Introdução ao Ecossistema .NET & Documentação

.NET – A unified platform

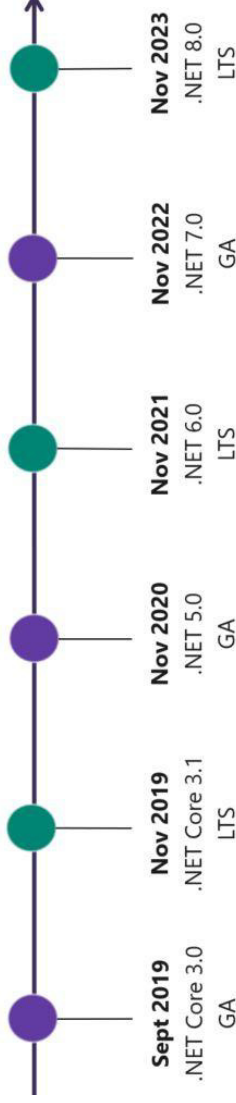


Fonte: .NET Blog(2019)



Sobre .NET Core e .NET

- ➔ Plataforma cruzada : Linux, Windows e Mac OS
- ➔ Produzir um produto com base de código única
- ➔ Open Source -> MIT e Apache 2
- ➔ .NET = .NET Core + .NET Framework + Xamarin + Mono
- ➔ Aquisição facilitada a partir do NUGET
- ➔ Modular e com melhor desempenho - > [Performance Improvements in .NET 5 - .NET Blog](https://blogs.msdn.microsoft.com/dotnet/2017/08/24/performance-improvements-in-net-5/)
- ➔ Ciclos de lançamentos mais rápidos





Sobre ASP.NET Core

- ➔ Construção de aplicações web, IoT apps e aplicações backend para mobile
- ➔ Funcionalidades para construção de aplicações com renderização no servidor
 - ◆ Uso do MVC -> ASP.NET Core MVC
 - Uso de Padrão MVC (Model-View-Controller)
 1. Model: Representa o estado da aplicação, encapsulando lógica de negócio.
 2. View: Responsável por apresentar dados através de interface.
 3. Controller: Componentes que lidam com a interação do usuário a partir da View e trabalham com a model para apresentar respostas para o usuário.



Sobre ASP.NET Core

- Construção de aplicações web, IoT apps e aplicações backend para mobile
 - Funcionalidades para construção de aplicações com renderização no servidor
 - ◆ Uso do MVC -> ASP.NET Core MVC
 - Uso de Padrão MVC (Model-View-Controller)
 - Model Binding
 - Mapeamento automático de dados de requisições HTTP para parâmetros de métodos de ação
 - Model Validation : Validação automática no lado do cliente e servidor
-



Sobre ASP.NET Core

- ➔ Construção de aplicações web, IoT apps e aplicações backend para mobile
- ➔ Funcionalidades para construção de aplicações com renderização no servidor
 - ◆ Uso do MVC -> ASP.NET Core MVC
 - ◆ Razor Pages e Razor markup
 - Modelo baseado em páginas -> simplificado
 - Interface e lógica de negócio são separados, mas dentro da página.
 - sintaxe de marcação para inserir código baseado em .NET em páginas WEB = Razor markup + C# + HTML



Sobre ASP.NET Core

- Construção de aplicações web, IoT apps e aplicações backend para mobile
 - Funcionalidades para construção de aplicações com renderização no servidor
 - ◆ Uso do MVC -> ASP.NET Core MVC
 - ◆ Razor Pages e Razor markup
 - ◆ Tag Helpers
 - Permite que o código do lado do servidor participe da criação e renderização de elementos HTML em arquivos Razor
 - Experiência de desenvolvimento com HTML mais amigável
 - IntelliSense no ambiente para sintaxe HTML e Razor
 - Manutenção de código usando somente informações do servidor
-



Sobre ASP.NET Core

- Construção de aplicações web, IoT apps e aplicações backend para mobile
 - Funcionalidades para construção de aplicações com renderização no cliente
 - ◆ Blazor
 - Framework para construir aplicativos do lado do cliente, que são executados direto no navegador com **WebAssembly(WASM)** usando C#.
 - Uso da interoperabilidade com Javascript para lidar com manipulação da DOM e chamadas de API.
-



Sobre ASP.NET Core

- Construção de aplicações web, IoT apps e aplicações backend para mobile
 - Funcionalidades para construção de aplicações com renderização no cliente
 - ◆ Blazor
 - ◆ Integração com frameworks como Angular, React e Bootstrap
-

Aula 2

Entendendo Xamarin

Introdução ao Ecossistema .NET &
Documentação

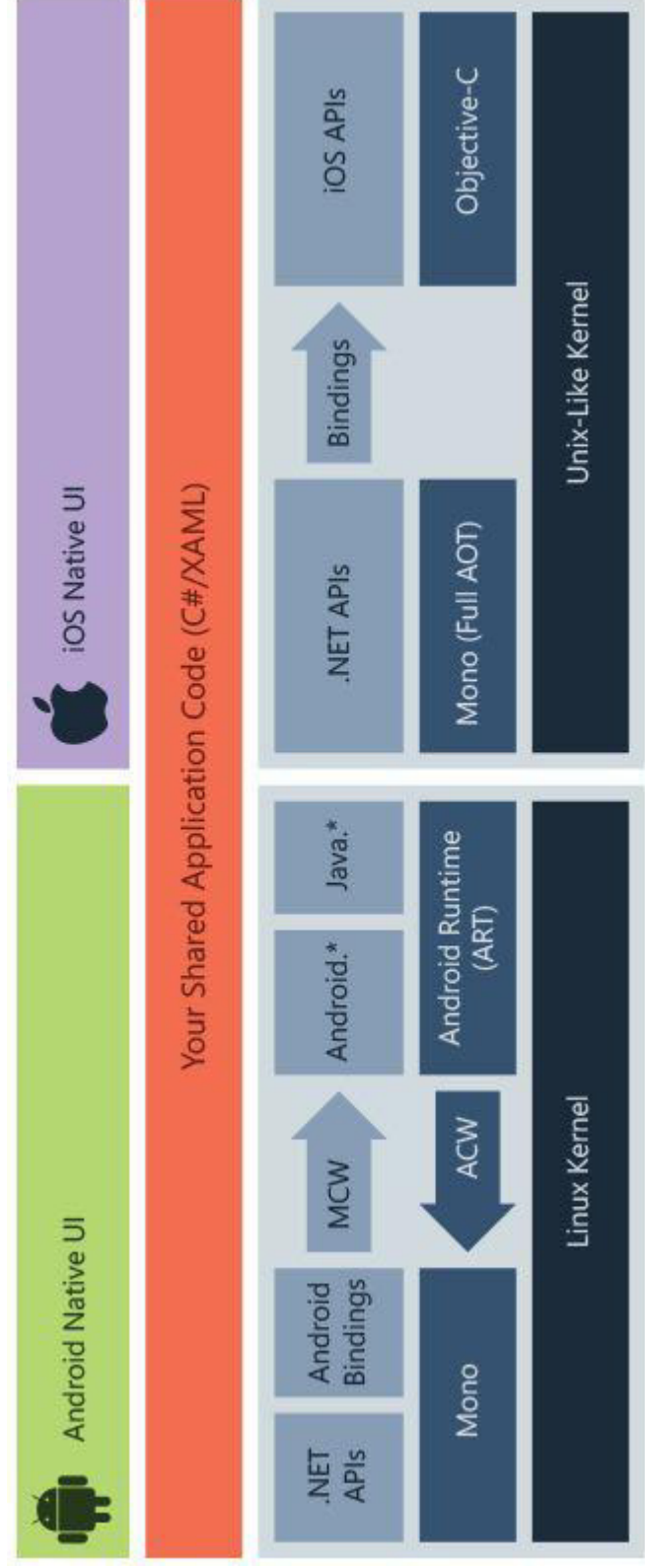


Sobre Xamarin

- Plataforma para desenvolvimento de aplicações mobile modernas com alto desempenho
 - Baseado no projeto Mono, implementação open source baseada no .NET Framework
 - Criação de interface nativa em cada plataforma(Android, iOS, MacOS e Windows apps) e código de lógica compartilhado
 - Possui todos os benefícios já citados da plataforma .NET
-



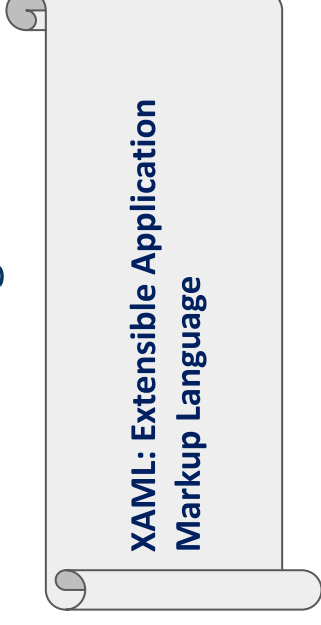
Como funciona o Xamarin?





Xamarin.Forms

- ★ Framework open source para desenvolvimento de interfaces para o usuário
- ★ Aplicativos Android, iOS e Windows com única base de código
- ★ Criação de UI com XAML e lógica com C#
- ★ Uso de biblioteca Xamarin.Essentials
 - informações de dispositivos
 - sistema de arquivos
 - acelerômetro
 - bloqueio de tela...

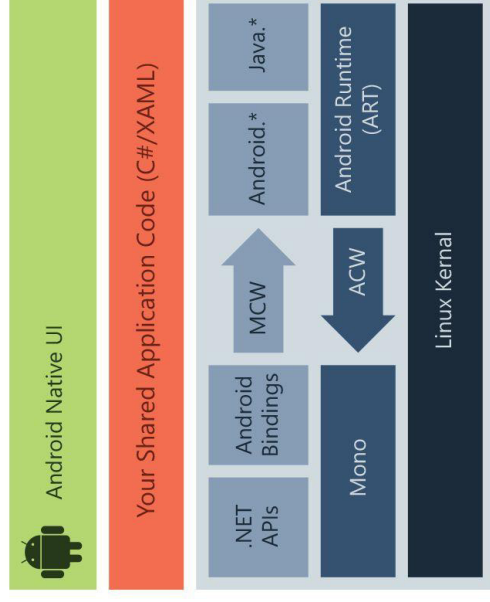


```
XAML
<StackPanel>
  <Button Content="Click Me" />
</StackPanel>
```



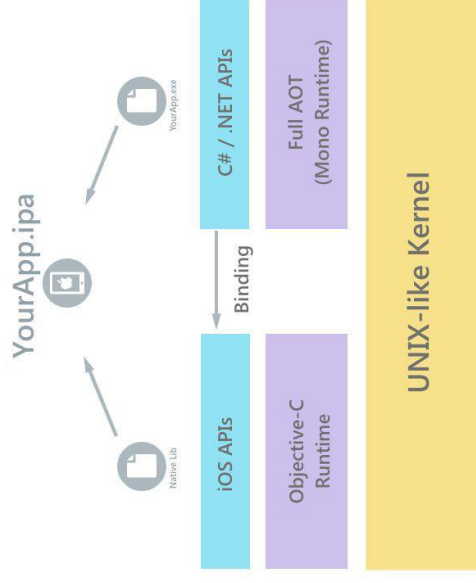
Xamarin.Android

- ★ Compilação de C# para LI (linguagem intermediária),
que por sua vez é compilado para assembly nativo no momento
da execução
- ★ Executado no ambiente Mono + ART





- ★ Compilação total de C# para código assembly nativo
- ★ Executado no ambiente Mono + AOT(ahead of time)
- ★ Restrição de segurança: sem permissão para execução de código gerado dinamicamente.



Para saber mais

[Tudo o que você precisa saber sobre as licenças de projetos open source | by Diego Martins de Pinho | Training Center | Medium](#)

[.NET is open source on GitHub | .NET](#)

[.NET Standard | Common APIs across all .NET implementations](#)

[Performance Improvements in .NET 5 - .NET Blog \(microsoft.com\)](#)

[Introduction to ASP.NET Core Blazor | Microsoft Docs](#)
