

# Introdução à Plataforma .NET

Uma Plataforma Moderna e Unificada para Desenvolvimento

*Apresentação introdutória sobre .NET*



## O que é .NET?

- Plataforma de desenvolvimento gratuita e open-source
- Multiplataforma: Windows, Linux, macOS
- Múltiplas linguagens: C#, F#, VB.NET
- Desenvolvida pela Microsoft desde 2000
- Atualmente na versão .NET 8 (LTS)

# Evolução Histórica

## .NET Framework (2002-2019)

- Apenas Windows
- Versões 1.0 até 4.8
- Focado em aplicações desktop e web

## .NET Core (2016-2020)

- Multiplataforma
- Open-source
- Versões 1.0 até 3.1

## .NET 5+ (2020-presente)

- Unificação das plataformas
- .NET 5, 6, 7, 8, 9...
- Versões LTS a cada 2 anos
- Performance aprimorada

# Principais Características

## Performance

- Compilação JIT otimizada
- Garbage Collection eficiente
- AOT (Ahead-of-Time) compilation

## Segurança

- Type safety
- Memory management automático
- Recursos de segurança integrados

# Principais Características (cont.)

## Multiplataforma

- Windows, Linux, macOS
- Containers Docker
- Cloud-native

## Ecossistema Rico

- NuGet Package Manager
- Milhares de bibliotecas
- Ferramentas robustas

# Linguagens Suportadas

## C#

- Linguagem principal
- Orientada a objetos
- Sintaxe moderna e limpa
- Amplamente adotada

## F#

- Linguagem funcional
- Ideal para análise de dados
- Programação matemática

## VB.NET

- Visual Basic evoluído
- Sintaxe familiar
- Integração completa

## PowerShell

- Scripts e automação
- Administração de sistemas
- Cross-platform

# Tipos de Aplicações



## Desktop

- WPF, WinUI, MAUI
- Aplicações nativas



## Web

- ASP.NET Core
- Blazor (WebAssembly)
- APIs REST e GraphQL



## Mobile

- .NET MAUI
- Xamarin (legacy)

# Tipos de Aplicações (cont.)

## Cloud & Microservices

- Azure Functions
- Containers Docker
- Kubernetes

## Games

- Unity Engine
- MonoGame

## IA & Machine Learning

- ML.NET
- TensorFlow.NET



# Ferramentas de Desenvolvimento

## IDEs

- Visual Studio
- Visual Studio Code
- JetBrains Rider
- Visual Studio for Mac

## CLI Tools

- .NET CLI
- dotnet command
- Scaffolding tools
- Package management

# ASP.NET Core - Web Development

## Características Principais

- **High Performance:** Um dos frameworks web mais rápidos
- **Modular:** Middleware pipeline configurável
- **Dependency Injection:** Injeção de dependência nativa
- **Configuration:** Sistema flexível de configuração

## Tipos de Projetos

- **Web API:** APIs REST/GraphQL
- **MVC:** Model-View-Controller
- **Blazor:** UI interativa com C#
- **Razor Pages:** Desenvolvimento rápido de páginas

# Entity Framework Core

## ORM Moderno e Poderoso

- **Code First:** Modelos definem o banco
- **Database First:** Scaffold do banco existente
- **LINQ:** Consultas type-safe
- **Migrations:** Versionamento do schema
- **Múltiplos provedores:** SQL Server, MySQL, PostgreSQL, SQLite, etc.

```
var users = await context.Users
    .Where(u => u.IsActive)
    .OrderBy(u => u.Name)
    .ToListAsync();
```

# Performance Benchmarks

## .NET está entre os frameworks mais rápidos

- **TechEmpower Benchmarks:** Consistentemente no top 10
- **Startup time:** Muito rápido, especialmente com AOT
- **Memory usage:** Eficiente com GC moderno
- **Throughput:** Milhões de requests por segundo

## Otimizações Contínuas

- Cada versão traz melhorias significativas
- Compilação AOT reduz cold start
- Profile-Guided Optimization (PGO)

# Ecosystem NuGet

## Package Manager Oficial

- +400.000 pacotes disponíveis
- Gestão de dependências automática
- Versionamento semântico
- Restauração automática

## Pacotes Populares

- Newtonsoft.Json / System.Text.Json
- AutoMapper
- Serilog
- FluentValidation
- MediatR

# Deployment e DevOps

## Containers

- Docker oficial
- Imagens otimizadas
- Multi-stage builds

## Cloud

- Azure App Service
- AWS Lambda
- Google Cloud Run

## CI/CD

- GitHub Actions
- Azure DevOps
- Docker Hub
- Automated testing

## Monitoring

- Application Insights
- Logging integrado
- Health checks

# .NET MAUI - Multiplataforma

## Multi-platform App UI

- Um código, múltiplas plataformas
- Windows, macOS, iOS, Android
- Performance nativa
- Acesso às APIs específicas da plataforma

## Vantagens

- Compartilhamento de código
- UI nativa em cada plataforma
- Desenvolvimento unificado
- Tooling integrado

# Quando Usar .NET?

## ✓ Ideal para:

- Aplicações enterprise
- APIs de alta performance
- Sistemas críticos
- Equipes Microsoft-centric
- Projetos de longo prazo

## ⚠ Considerar alternativas:

- Projetos muito simples
- Equipes 100% open-source
- Constraints específicos de infraestrutura



# Aprendendo .NET

## Recursos Oficiais

- [docs.microsoft.com/dotnet](https://docs.microsoft.com/dotnet)
- Microsoft Learn
- .NET Documentation

## Cursos e Certificações

- Microsoft Certified: Azure Developer
- Pluralsight, Udemy
- YouTube: .NET Channel

## Comunidade

- Stack Overflow

# Roadmap .NET

## Próximos Lançamentos

- .NET 9 (Novembro 2024) - STS
- .NET 10 (Novembro 2025) - LTS
- Releases anuais em novembro

## Focos Futuros

- Performance contínua
- Cloud-native features
- AI/ML integration
- Developer productivity

# Exemplo Prático - Hello World API

```
// Program.cs
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/", () => "Hello World!");
app.MapGet("/user/{id}", (int id) =>
    new { Id = id, Name = $"User {id}" });

app.Run();
```

## Recursos demonstrados:

- Minimal APIs: Sintaxe concisa
- Route parameters: Binding automático
- JSON serialization: Automática

# Conclusão

.NET é uma plataforma **madura, performática e versátil**

## ✨ Principais Vantagens

- Ecossistema robusto e ativo
- Performance excepcional
- Tooling de primeira classe
- Suporte corporativo da Microsoft
- Comunidade global forte

## 🚀 Perfeito para

Desenvolvimento moderno de aplicações escaláveis e mantíveis

# Obrigado!

## Perguntas?

### Recursos para começar:

- [dotnet.microsoft.com](https://dotnet.microsoft.com)
- [github.com/dotnet](https://github.com/dotnet)
- [docs.microsoft.com/dotnet](https://docs.microsoft.com/dotnet)

*Vamos construir algo incrível com .NET! 🚀*