

# Aula 5: O que usar? Quando usar?

Introdução ao Ecossistema .NET &  
Documentação

# Objetivos

1. .NET X .NET Framework : Qual escolher para minha aplicação?
  2. O que usar para cada tipo de aplicação?
-

## Aula 5

.NET X .NET Framework : Qual escolher para minha aplicação?

Introdução ao Ecossistema .NET &  
Documentação



# Qual implementação escolher?



Use .NET quando:

- Necessitar de plataforma cruzada
  - Há direcionamento de microsserviços
    - Azure Service Fabric
    - Azure App Service
  - Uso de Docker
    - Hospedagem em infraestrutura Linux ou Windows
    - Azure Kubernetes Service
  - Alto desempenho e sistemas escalonáveis
  - Necessidade de versões .NET diferentes por aplicação
-

# Qual implementação escolher?



Use .NET Framework quando:

- Não houver necessidade de migração para .NET
- Bibliotecas de terceiros ou pacotes NUGET não são disponíveis para .NET
- Necessidade de tecnologias não disponíveis para .NET
  - ASP.NET WebForms
  - ASP.NET Web Pages não estão incluídos no ASP.NET Core
  - Implementação do lado servidor para WCF
  - Workflow Foundation, Workflow Services e Workflow Data Services
- Quando o projeto não suporta determinada linguagem para o caso de Visual Basic e F#



# ASP.NET Core X ASP.NET 4.x

ASP.NET Core	ASP.NET 4.x
Construído para windows, Linux e macOS	Para Windows
Criar aplicações com interface via Razor Pages, MVC, Web API, SignalR	Uso de Web Forms, SignalR, MVC, Web Api, WebHooks ou Web Pages
Múltiplas versões por máquina	Única versão por máquina
Diversas ferramentas de desenvolvimento	Foco no Visual Studio com C#, F# e VB
Maior Desempenho que ASP.NET 4.x	Bom desempenho no geral
Uso do .NET Core runtime	Uso do .NET Framework Runtime



# Aula 5

## O que usar para cada tipo de aplicação?

Introdução ao Ecossistema .NET &  
Documentação

# Aplicações Desktop

- ★ Universal Windows Apps
  - ★ Windows Presentation Foundation ( WPF)
  - ★ Windows Forms
  - ★ Xamarin.MAC
-





# Aplicações baseadas em containers

AZURE: Plataforma de nuvem completa que pode hospedar aplicativos e simplificar o desenvolvimento de novos aplicativos

- ★ Para aplicações monolíticas > Azure App Service
  - ★ Aplicações com arquiteturas em camadas -> Azure Kubernetes Service (AKS)
  - ★ Microserviços: AKS ou Azure Web Apps para Containers
  - ★ Funções Serverless e event handlers : Azure Functions
  - ★ Processamento Batch de larga escala: Azure Batch
-



# Outros serviços Azure para desenvolvedores

- ★ Azure SQL : Versão em nuvem do SQL Server
  - ★ Azure Cosmos DB: Banco de dados NoSQL com escalabilidade e de API compatível com MongoDB
  - ★ Azure Blob Storage: Armazenamento em nuvem de arquivos, prezando pela escalabilidade, redundância de dados e disponibilidade.
  - ★ Azure Key Vault: Permite o armazenamento em nuvem de segredos de aplicações, como connection strings e e chaves de APIs.
  - ★ Cognitive Services: Coleção de serviços em nuvem que permite a adição de Inteligência Artificial para aplicações, com funcionalidades como reconhecimento de falar, compreensão de linguagens e detecção de anomalias.
-



# Aprendizagem de máquina com .NET



## ★ ML.NET

- Framework de aprendizagem de máquina de plataforma cruzada e open source
- Criação de modelos de aprendizagem de máquina a partir de interface gráfica ( Model Builder) ou por meio do ML.NET CLI
- Integração com outros frameworks famosos como TensorFlow, ONNX, INFER.NET





# Aprendizagem de máquina com .NET



## Tipos de predições com ML.NET:

- Classificação e categorização de objetos
- Predição de valores contínuos
- Detecção de anomalias
- Recomendações
- Análise de sentimento





# Desenvolvimento de Jogos

- ★ Unity : Plataforma de desenvolvimento de jogos 3D e em tempo real, disponível para Windows e macOS.
- ★ MonoGame : Framework multiplataforma, open-source com suporte para .NET Core e .NET 4.5
- ★ Godot : Game engine gratuita e open-source, para desenvolvimento de jogos usando C#
- ★ Stride: Plataforma para desenvolvimento de jogos 2D/3D gratuita, multiplataforma, open source e com editor robusto e gráficos realistas.
- ★ Wave Engine: Mecanismo gratuito para desenvolvimento gráfico voltado para negócios e indústria. Constrói soluções de alta qualidade 3D e 2D.
- ★ FlatRedBall: Plataforma de desenvolvimento de jogos 2D, contruído em cima do MonoGame
- ★ CRYENGINE: Framework para desenvolvimento de jogos usando .NET e C#. Disponível para Windows e Linux.



# Internet das Coisas ( IoT)

- ★ Bibliotecas específicas para integração de aplicações com hardware especializado como sensores, dispositivos LCD, conversores analógico-digital
    - System.Device.Gpio
    - lot.Device.Bindings
  - ★ Suportado em muitas versões do Linux que suportam ARM/ARM64 e Windows IoT Core
  - ★ .NET NanoFramework
    - Plataforma gratuita e open-source que permite o desenvolvimento de aplicações com C# acoplados a dispositivos.
    - É suportado pela .NET Foundation.
    - Utilização do Visual Studio
-

# Internet das Coisas ( IoT)



## Meadow

- Plataforma IoT e full-stack que permite construir soluções focadas para indústria de forma segura e escalável
- Integrado com Azure
- Uso no Visual Studio ou VSCode



# Para saber mais

[Microserviços do .NET. Arquitetura de aplicativos .NET em contêineres | Microsoft Docs](#)

[Azure para desenvolvedores .NET | Microsoft Docs](#)

[Documentação do .NET | Microsoft Docs](#)

---