**Introdução ao .Net (DotNet)**

**Introdução ao .Net**

É uma plataforma diversificada que permite a construção de sistemas e aplicações.

C# é uma linguagem de programação usada no .Net.

É uma plataforma de desenvolvimento e multiplataforma (Windows, Linux, macos)

Link para o readme <https://dotnet.microsoft.com/en-us/learn>

<https://time.graphics/pt/line/291016>

<https://learn.microsoft.com/pt-br/dotnet/core/introduction>

**História do .Net**

O objetivo da criação do .Net era competir com o java

**Linha do tempo do .Net**

1998 foi quando o desenvolvimento do .net começou.

Em 05/01/2022 teve a primeira versão do .Net 1.0 e do C# e foi lançado o visual studio no mesmo ano.

O mono é uma implementação multiplataforma também do .Net, porém um pouco mais voltada para linux.

Todo o esforço da Microsoft agora é voltado para o desenvolvimento do .Net Core que é uma continuação do .Net só que multiplataforma.

**.Net framework (legado) e .Net**

O .Net framework ela só funciona em Windows e já o .Net core ou só .Net é a versão mais atual que funciona multiplataforma.

A versão.net core ela praticamente foi reescrita do zero, e foi feita com a finalidade de funcionar em multiplataformas.

O .net framework é apenas para a plataforma Windows.

**Versões .net framework**

Ele acabou na versão 4.8, foi sua última versão.

**Versões .net**

Para evitar confusão entre os nomes no .net a Microsoft resolveu tirar o nome core e deixar apenas .net.

Da versão 3.1 para a versão 5 teve algumas melhorias, porém é a mesma coisa.

**Compilador .net e seu funcionamento**

O compilador faz o trabalho de traduzir o que nós intendemos para que o computador entenda. Ele é um programa de computador que realiza essa conversão.

Programamos em uma linguagem de alto nível (C#, python, java) e o computador entende em uma linguagem de baixo nível (assembly, binario). O que o computador entende é: 0 1, ligado/desligado, passa corrente/não passa corrente, sim/não.

**O compilador do .Net**

Ele tem algumas especialidades que ele faz para que o código seja executado de fato. No processo de compilação ele gera um código intermediário IL Code, ele não depende de uma máquina ou de uma arquitetura específica para executar, e vai ser gerado .exe e dll que são os arquivos onde estão armazenados o código intermediário, e ai, ele precisará passar por mais um processo para ser executado que é o JIT compiler que vai pegar os arquivos .exe e dll, vai compilar novamente e ai sim ele vai converter para a arquitetura especifica da máquina.

**Compilador e Transpilador**

**O compilador**, ele realiza a conversão de linguagem de alto nível para baixo nível.

**O transpilador**, é a conversão de uma linguagem ou implementação para outra. A sua saída permanece em linguagem de alto nível.

Exemplo: Typescript para javascript

**Linguagem compilada** são linguagens onde o código fonte é traduzido para o código de máquina.

C#, Java

**Linguagem interpretada** são linguagens que fazem a leitura e interpretação diretamente do código fonte.

Javascript, php.