



THE  
DEVELOPER'S  
CONFERENCE

# O Futuro em construir APIs com .NET6

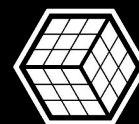
**Trilha Arquitetura .NET**  
**Júnior Porfirio**

## wois

- Sou do interior de SP (Campinas).
- Desenvolvedor .NET desde 2005.
- Arquiteto de Software na CI&T.

Blog - [juniorporfirio.medium.com](https://juniorporfirio.medium.com)

Github / Twitter / LinkedIn - [juniorporfirio](#)

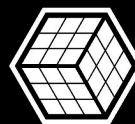


THE  
DEVELOPER'S  
CONFERENCE

# Agenda

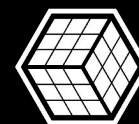
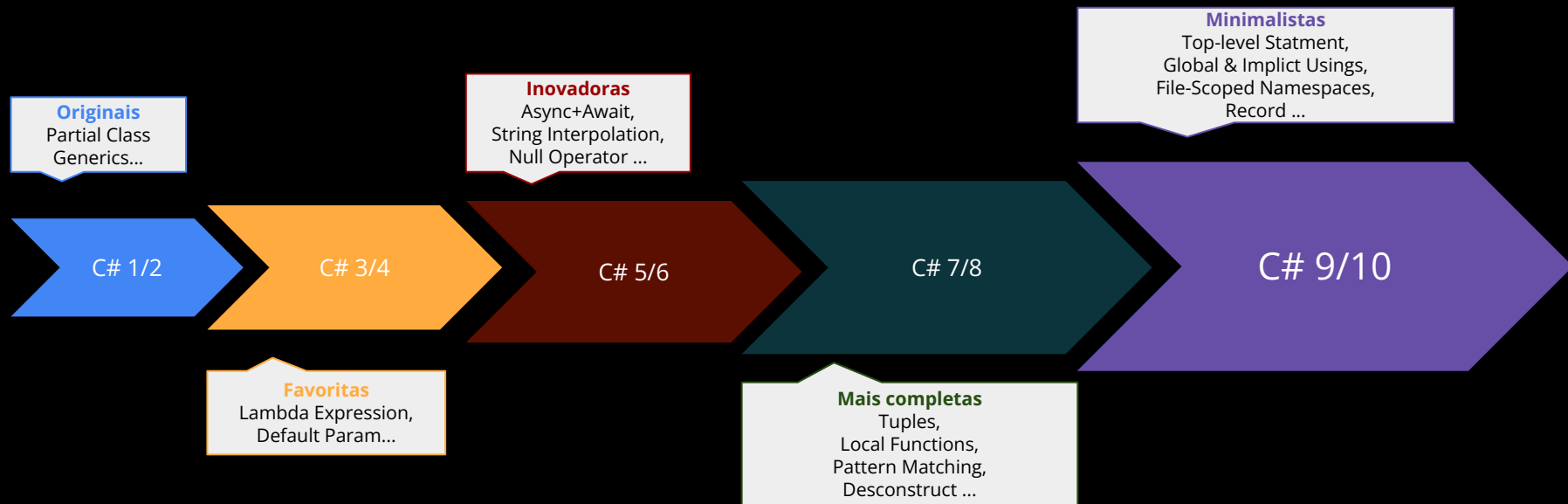


- O futuro minimalista
  - Features C# 10
- Minimal Apis .NET 6
- Benefícios
- Demos
  - OpenApi Swagger
  - Validações
  - Testes



THE  
DEVELOPER'S  
CONFERENCE

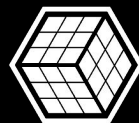
# O futuro minimalista



THE  
DEVELOPER'S  
CONFERENCE

# Global Usings

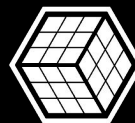
```
Using.cs*  [icon] X
TDCFutureTalks
1  [icon] global using Microsoft.EntityFrameworkCore;
2  global using MinimalApis.Validators;
3  global using System.ComponentModel.DataAnnotations;
```



THE  
DEVELOPER'S  
CONFERENCE

# Implicit Usings

```
<Project Sdk="Microsoft.NET.Sdk.Web">  
  <PropertyGroup>  
    <TargetFramework>net6.0</TargetFramework>  
    <Nullable>enable</Nullable>  
    <ImplicitUsings>enable</ImplicitUsings>  
  </PropertyGroup>  
  
  <ItemGroup>  
    <Using Include="Microsoft.EntityFrameworkCore" />  
    <Using Include="MinimalApis.Validators" />  
    <Using Include="System.ComponentModel.DataAnnotations" />  
  </ItemGroup>  
</Project>
```



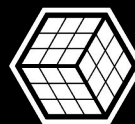
THE  
DEVELOPER'S  
CONFERENCE

# File-Scoped Namespaces

```
namespace TDCFutueTalks; // Não necessita colocar os braços para o namespace

1 referência
interface ITalkRepository
{
    1 referência
    List<Talk> All();
}

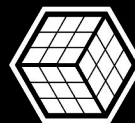
0 referências
internal class TalkRepository : ITalkRepository
{
    1 referência
    public List<Talk> All()
    {
        // Implementação Aqui
        return new ();
    }
}
```



THE  
DEVELOPER'S  
CONFERENCE

# Records

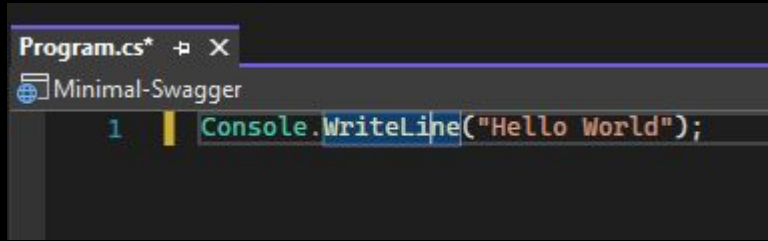
```
//class Customer
//{
//    public string Id { get; set; }
//    public string FirstName { get; set; }
//    public string LastName { get; set; }
//}
0 referências
record Customer(Guid Id, string FirstName, string LastName);
0 referências
record struct Document(string Id, string name, DateOnly Birthday);
```



THE  
DEVELOPER'S  
CONFERENCE

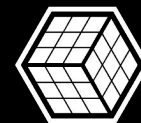


# Top-level Statement



A screenshot of a code editor window. The title bar shows 'Program.cs\*' with a minus, maximize, and close button. Below the title bar, there's a tab labeled 'Minimal-Swagger'. The code editor shows a single line of C# code: `1 Console.WriteLine("Hello World");`. The line number '1' is on the left, and a yellow cursor is positioned at the start of the code. The text 'Console.WriteLine' is highlighted with a blue selection box.

```
1 Console.WriteLine("Hello World");
```

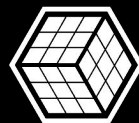


THE  
DEVELOPER'S  
CONFERENCE

# Minimal Apis



Aplicações sem cerimônias



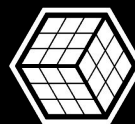
THE  
DEVELOPER'S  
CONFERENCE

# Minimal Apis

dotnet new web

- Sem Startup
- WebHost simples
- Sem Controller
- **3 linhas de código !**

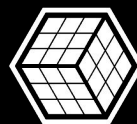
```
var app = WebApplication.Create();  
app.MapGet("/", () => "Hello World!");  
app.Run();
```



THE  
DEVELOPER'S  
CONFERENCE

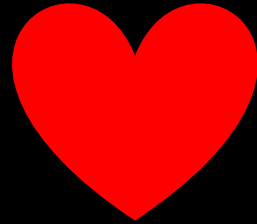
# Benefícios

- Implementação rápida para iniciantes
- Fácil de manter, um codebase menor (minimalismo)
- Não faz uso do MVC (controllers, filters, roteamentos, ...)
- Melhor performance ( Em média mais de 7 milhões RPS)
- Pode escalar de acordo com sua necessidade.
- Ideal para aplicações microservices / FaaS

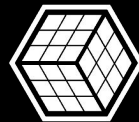


THE  
DEVELOPER'S  
CONFERENCE

# Minimal Apis



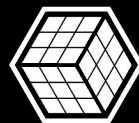
Show me code



THE  
DEVELOPER'S  
CONFERENCE

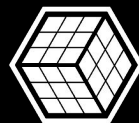
# Referências

- [minimalapis.github.io](https://minimalapis.github.io)
- [MinimalApis.Validators Package](#)
- [MinimalApis.Extensions Package](#)
- [Minimal APIs at a glance](#)
- [Introdução Minimal Apis](#)
- [Código Demonstração](#)



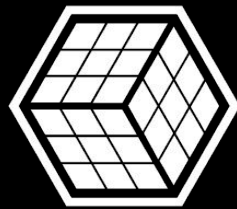
THE  
DEVELOPER'S  
CONFERENCE

# Dúvidas ?



THE  
DEVELOPER'S  
CONFERENCE

# Obrigado !



THE  
DEVELOPER'S  
CONFERENCE