

# Introducción a C# & .NET



# .NET

¿En todos  
lados?



# Están listos?



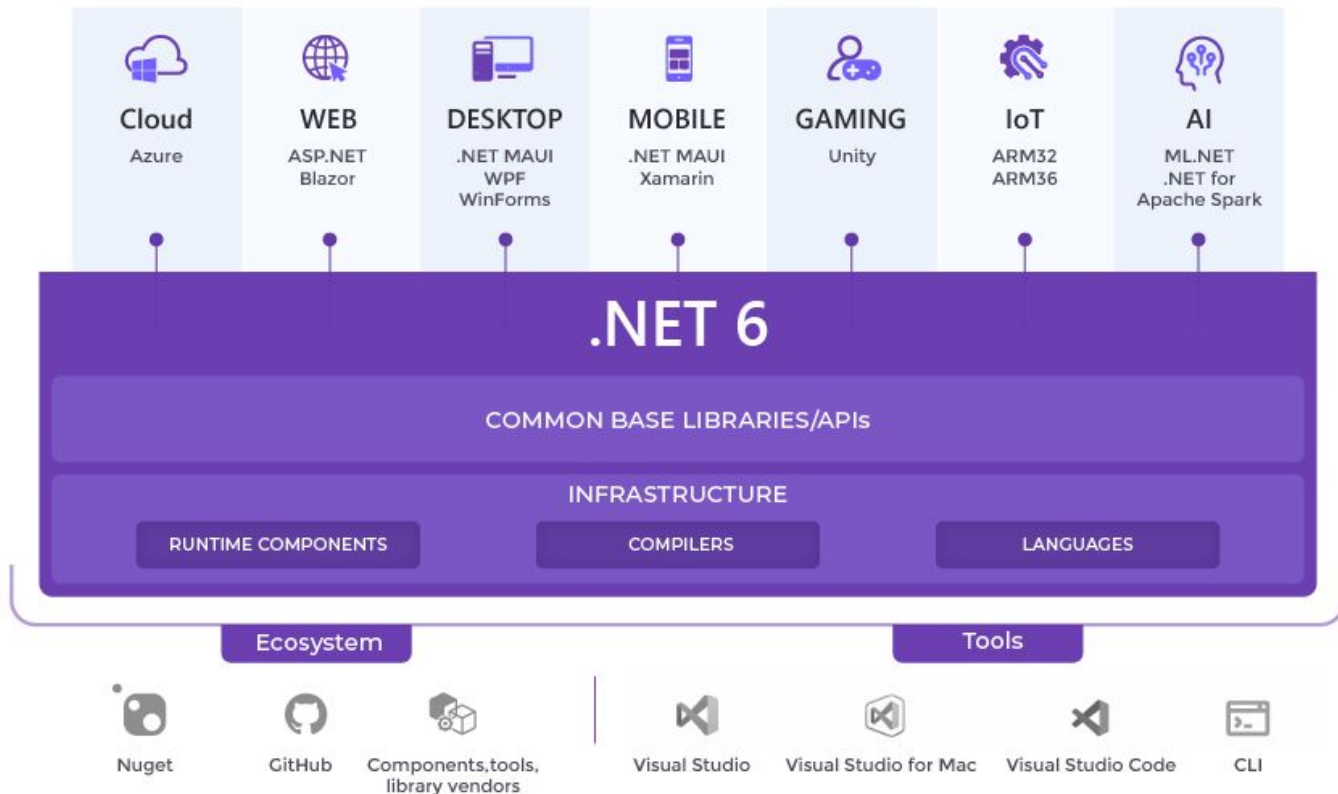


## QUÉ ES .NET?

.NET es una plataforma para desarrolladores de código abierto, multiplataforma y gratuita que incluye herramientas, lenguajes de programación y bibliotecas para crear diversos tipos de aplicaciones. La plataforma base proporciona componentes aplicables a todos los tipos de aplicaciones.

Con .NET, puede usar varios lenguajes, editores y bibliotecas para crear aplicaciones web, móviles, de escritorio, juegos, IoT, etc. Puedes utilizar marcos como ASP.NET para ampliar las capacidades con componentes específicos.









# **UN POCO DE SU HISTORIA**

## Cuando apareció .NET?

La plataforma .NET fue anunciada por primera vez por Microsoft en el año 2000 y su primera versión oficial, conocida como .NET Framework 1.0, fue lanzada el 13 de febrero de 2002.

Esta plataforma fue diseñada para ofrecer un entorno de desarrollo unificado para construir una variedad de aplicaciones de software, incluyendo aplicaciones web, aplicaciones de escritorio, servicios web y más.

C# está inspirado en C, C++ y Java. Por su lado, .NET permite trabajar con distintos lenguajes de programación. Esto lo logra







**¡Developers, developers,  
developers!**

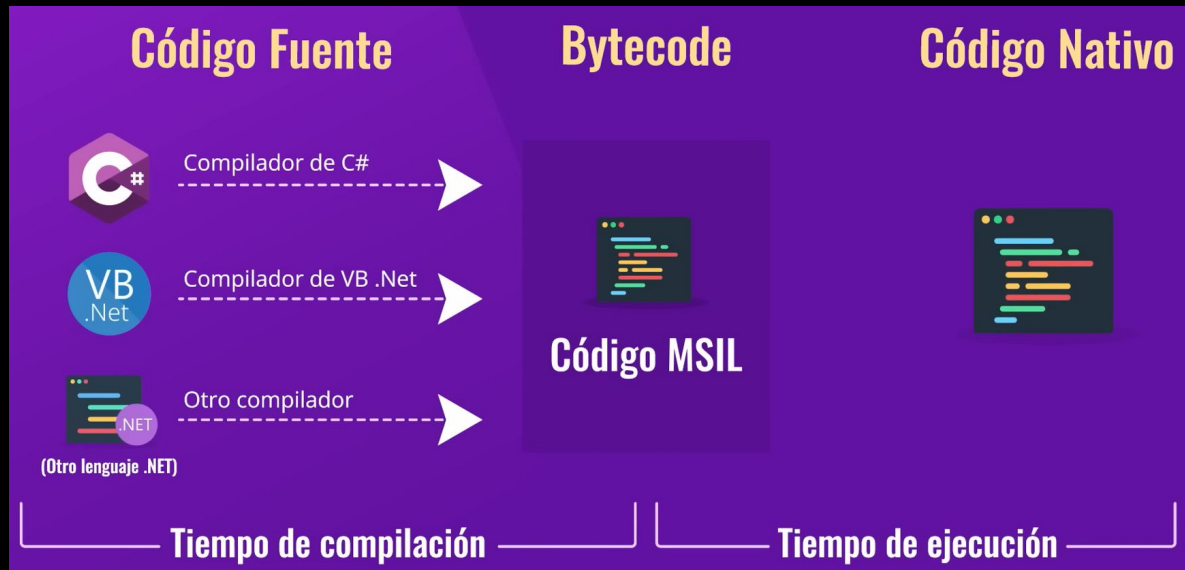


[Steve Ballmer at .NET presentation](#)

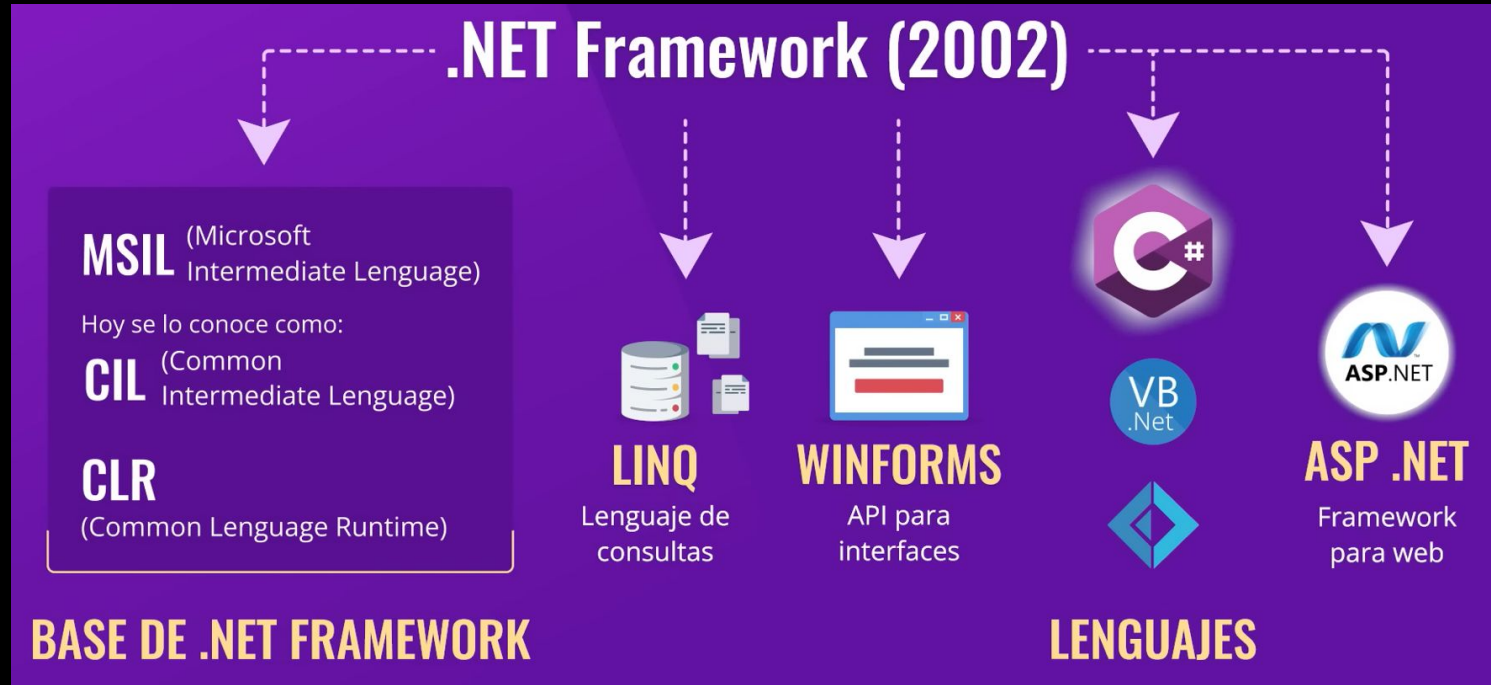


# .NET Framework

El código se compila a bytecode que es un lenguaje intermedio (MSIL - Microsoft Intermediate Language) y es en tiempo de ejecución cuando se lo traduce a código máquina, siendo casi lo mismo que hace Java.

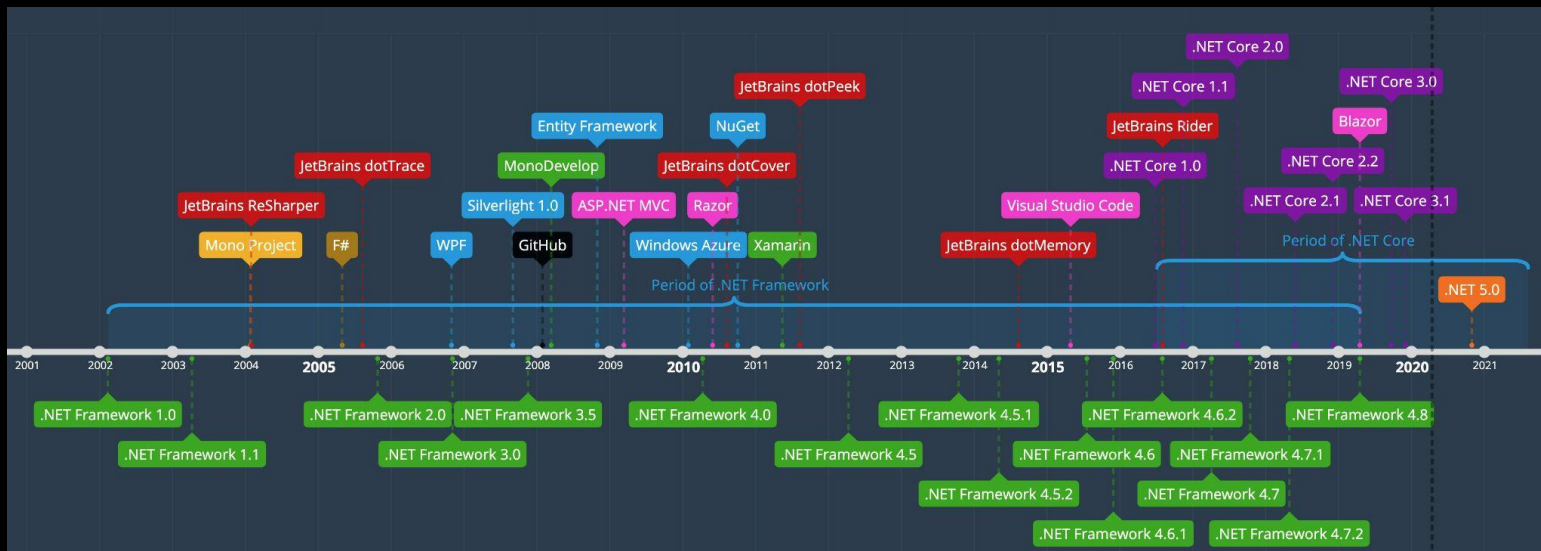


# .NET Framework





# Timeline de .NET Framework



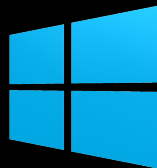


**Pero había un  
problema...**



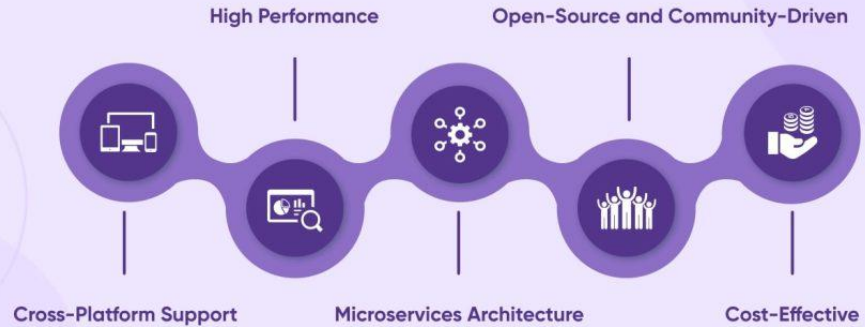
## **Pero había un problema...**

.NET Framework solo funcionaba sobre la plataforma  
Windows





## Advantages of .NET Core





# Diferencias entre .NET Core y .NET Framework

.NET CORE vs. .NET FRAMEWORK - COMPARISON	
.NET Core	.NET Framework
• Open-source	• Not open-source
• Cross-platform support	• For Windows only
• Even better performance	• Excellent performance
• Modular deployment	• Via Internet Information Server only
• Portable class libraries	• Built-in class libraries

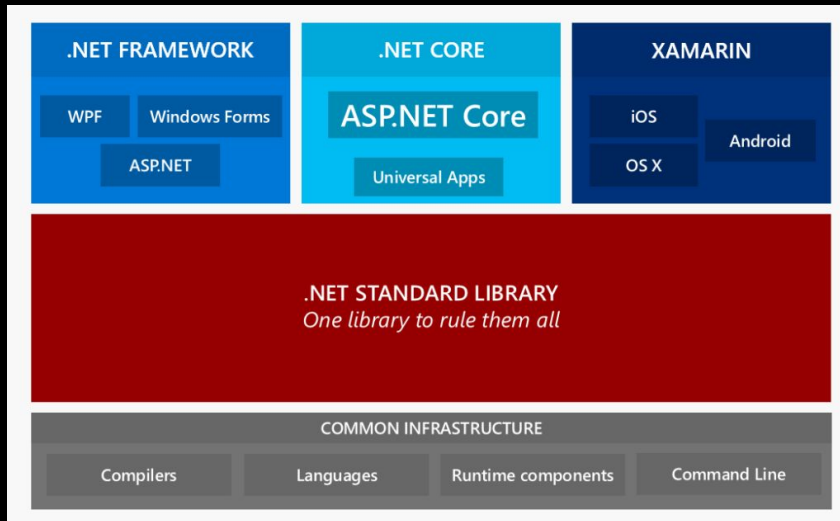




# .NET Standard

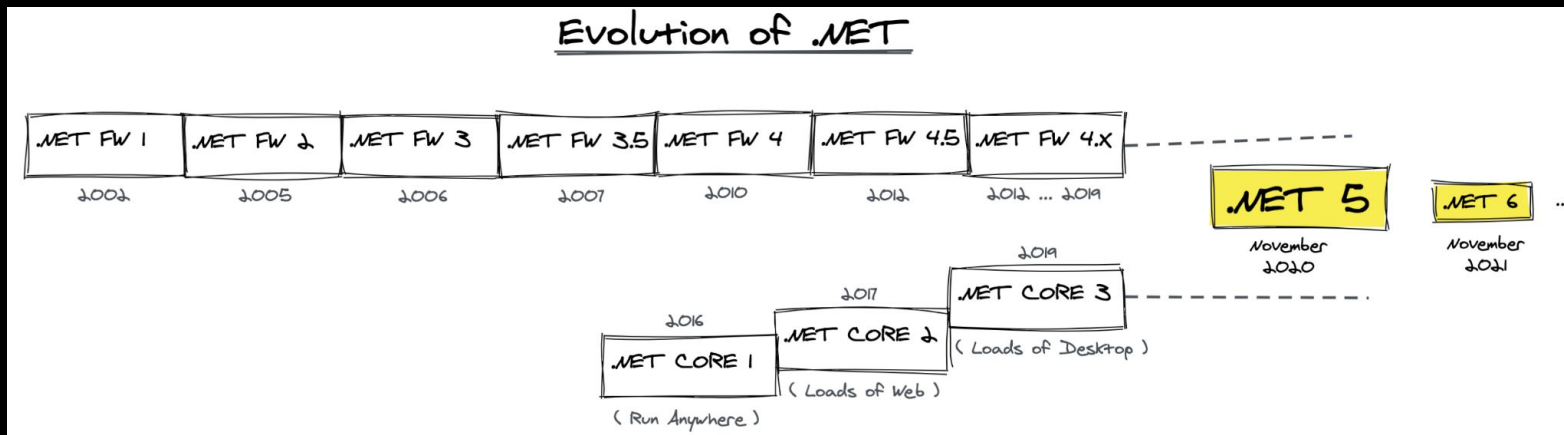
.NET Standard es una especificación que define un conjunto de APIs comunes para diferentes implementaciones de .NET.

Su propósito principal es establecer una interfaz común entre las diversas versiones y variantes de la plataforma .NET, permitiendo que las bibliotecas y los componentes escritos para .NET Standard sean compatibles con múltiples implementaciones de .NET, como .NET Framework, .NET Core y Xamarin, entre otros.



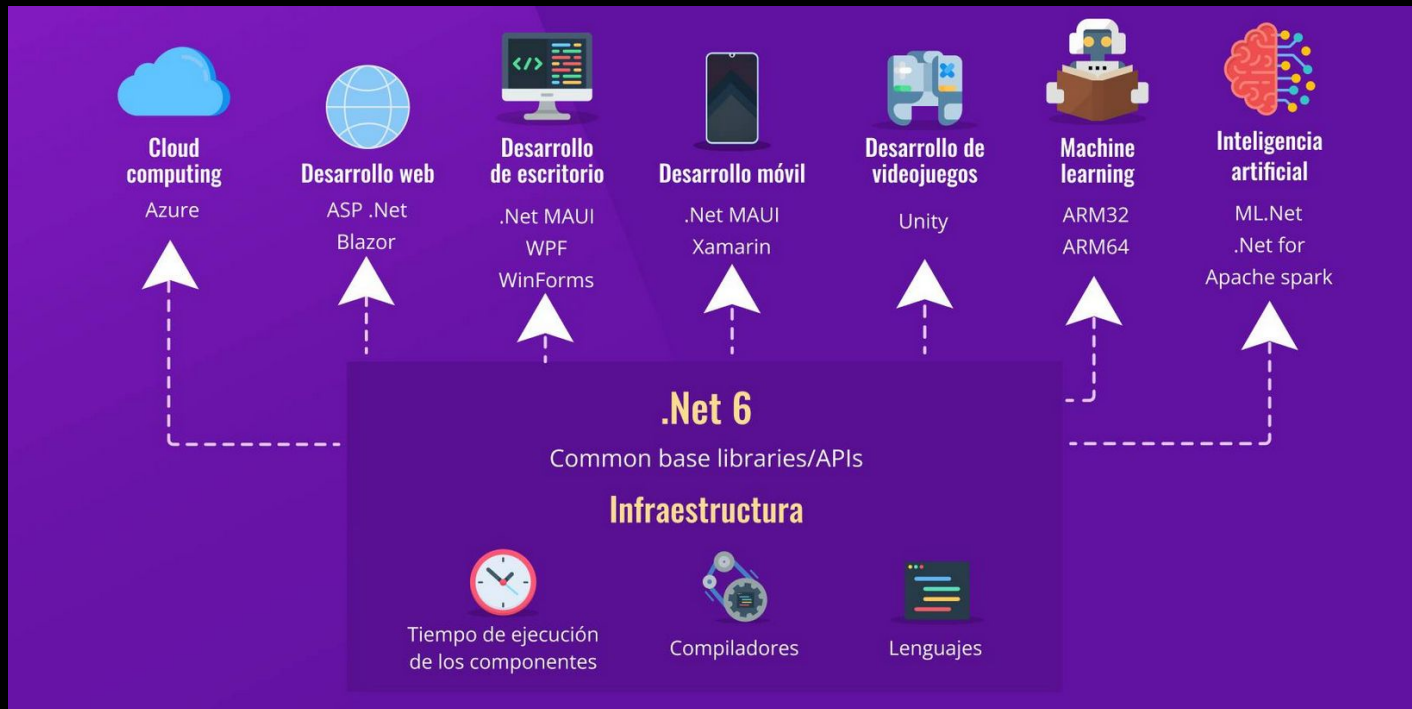


# La evolución de .NET y el futuro de la plataforma





# .NET





# ASP.NET Core

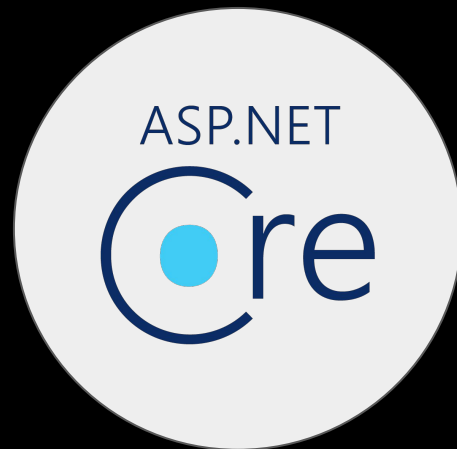


# Qué es ASP.NET Core?

ASP.NET Core es un marco de desarrollo web moderno, de alto rendimiento multiplataforma y de código abierto para la plataforma .NET.

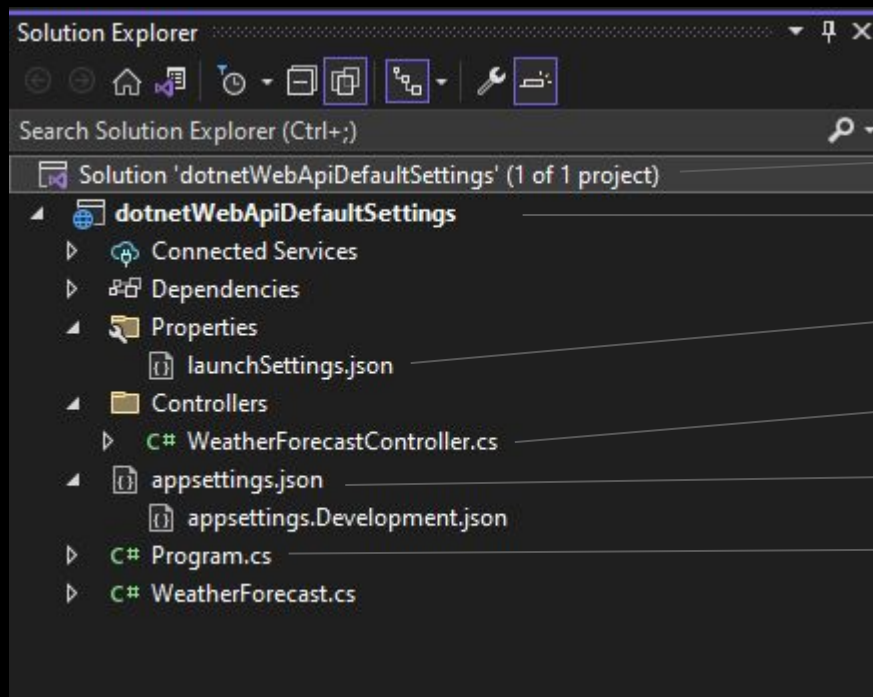
## Con ASP.NET Core podemos crear:

- Aplicaciones web usando el [patrón Modelo-Vista-Controlador](#) (MVC).
- Usar [Razor Pages](#) para la codificación de las vistas.
- [Blazor](#) para usar C# en el navegador junto con JavaScript. Comparte la lógica de aplicación del lado cliente y servidor escrita toda con .NET.
- Servicios backend como Web APIs, microservicios, serverless function, aplicaciones en tiempo real usando SignalR, etc.





# Estructura de un proyecto



Solución o workspace

Proyecto

Configuraciones para la ejecución de la aplicación

Controlador

Archivos de configuración de nuestra aplicación

Punto de entrada de nuestra aplicación



# Minimal APIs

Las Minimal APIs son un enfoque simplificado para construir rapidamente APIs HTTP con ASP.NET Core.

Puedes crear endpoints REST completamente funcionales con un código y configuración mínimas.

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();
app.UseHttpsRedirection();

var summaries = new[]
{
    "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering"
};

app.MapGet("/weatherforecast", () =>
{
    var forecast = Enumerable.Range(1, 5).Select(index =>
        new WeatherForecast
        (
            DateTime.Now.AddDays(index),
            Random.Shared.Next(-20, 55),
            summaries[Random.Shared.Next(summaries.Length)]
        ))
        .ToArray();
    return forecast;
})
.WithName("GetWeatherForecast");

app.Run();

internal record WeatherForecast(DateTime Date, int TemperatureC, string? Summary)
{
    public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
}
```



# **Iniciar un proyecto de Web API con .NET**





# Donde aprender .NET?



hdeleon.net



Maxi  
Programa



Nick  
Chapsas



[Roadmap de ASP.NET Core](#)

# Gracias!





**LINQ**



# Entity Framework