

UNIVERSIDAD DE CUENCA

ASP.NET CORE

PROGRAMACIÓN WEB





ASP.NET

Equipo de Trabajo

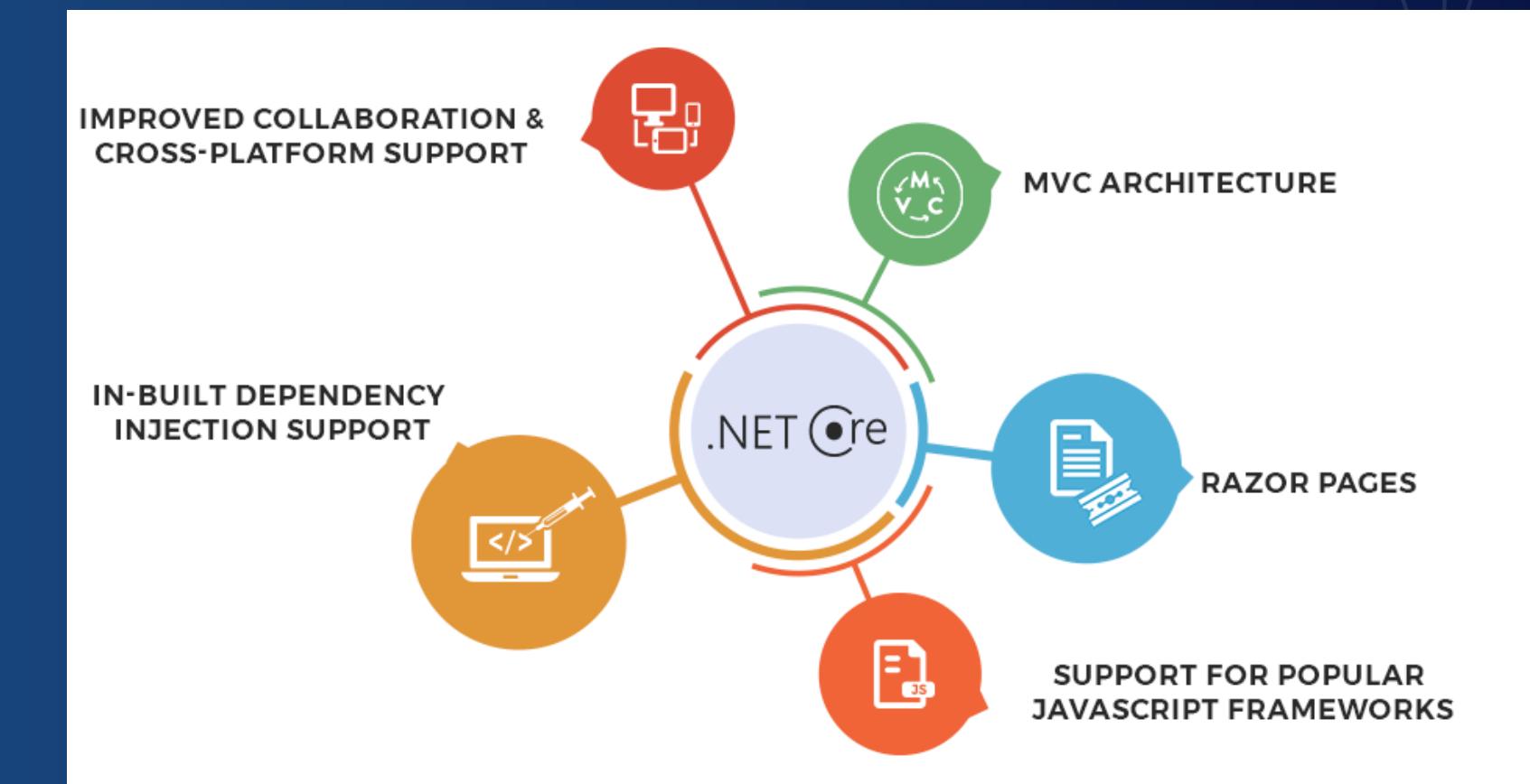




ASP.NET

CONTENIDO

1. INTRODUCCIÓN AL FRAMEWORK
2. ARQUITECTURA DEL FRAMEWORK
3. LENGUAJE Y ECOSISTEMA
4. REQUISITOS PARA INSTALACIÓN
5. PRINCIPALES CARACTERÍSTICAS
6. CASOS DE USO
7. COMPARACIÓN CON OTROS FRAMEWORKS
8. FUTURO Y ACTUALIZACIONES





01

INTRODUCCIÓN AL FRAMEWORK





Que es .NET?

.NET es una plataforma de desarrollo creada por Microsoft que se utiliza para construir y ejecutar aplicaciones en diversas plataformas, incluyendo Windows, macOS, Linux, Android, iOS y más. .NET proporciona un entorno consistente y unificado para desarrollar aplicaciones de escritorio, web, móviles, juegos y la nube.

.NET utiliza el lenguaje de programación C#, aunque también es compatible con otros lenguajes como F# y Visual Basic. Además, ofrece un amplio conjunto de bibliotecas y herramientas que facilitan el desarrollo, depuración y despliegue de a



Componentes .NET

.NET Core: Es la versión multiplataforma y de código abierto de .NET que se puede ejecutar en Windows, macOS y Linux.

ASP.NET: Es un marco de trabajo para construir aplicaciones web y servicios en la nube.

.NET Framework: Es la versión original de .NET que se ejecuta solo en Windows.

Blazor: Es una tecnología para construir interfaces de usuario interactivas del lado del cliente usando C# en lugar de JavaScript.

Xamarin: Es una extensión de .NET que permite el desarrollo de aplicaciones móviles para Android y iOS utilizando el lenguaje C# y la biblioteca de clases .NET.

Entity Framework: Es un marco de trabajo de mapeo objeto-relacional (ORM) que facilita el trabajo con bases de datos en aplicaciones .NET.



¿Qué es ASP.NET Core?

ASP.NET

ASP.NET Core es un marco de trabajo de desarrollo web de código abierto y multiplataforma desarrollado por Microsoft. Está diseñado para permitir a los desarrolladores construir aplicaciones web modernas y escalables que puedan ejecutarse en diferentes plataformas, como Windows, macOS y Linux.

ASP.NET Core es parte integral del ecosistema .NET y se integra bien con herramientas y servicios como Visual Studio, Azure y Entity Framework Core.

Historia

ASP.NET

- Microsoft introdujo Active Server Pages en diciembre de 1996, permitiendo el uso de scripts y componentes en HTML.
- Tras el lanzamiento de IIS en 1997, Microsoft investigó un nuevo modelo de aplicación para mejorar ASP.
- Anders y Guthrie desarrollaron el modelo inicial y el prototipo llamado XSP en Java.



Historia

ASP.NET

- Con el uso de Common Language Runtime (CLR), XSP se implementó en C# y fue conocido como Project Cool, luego renombrado a ASP+.
- Despues de varias versiones, el 5 de enero de 2002 se liberó ASP.NET como parte de .NET con la versión 1.0.



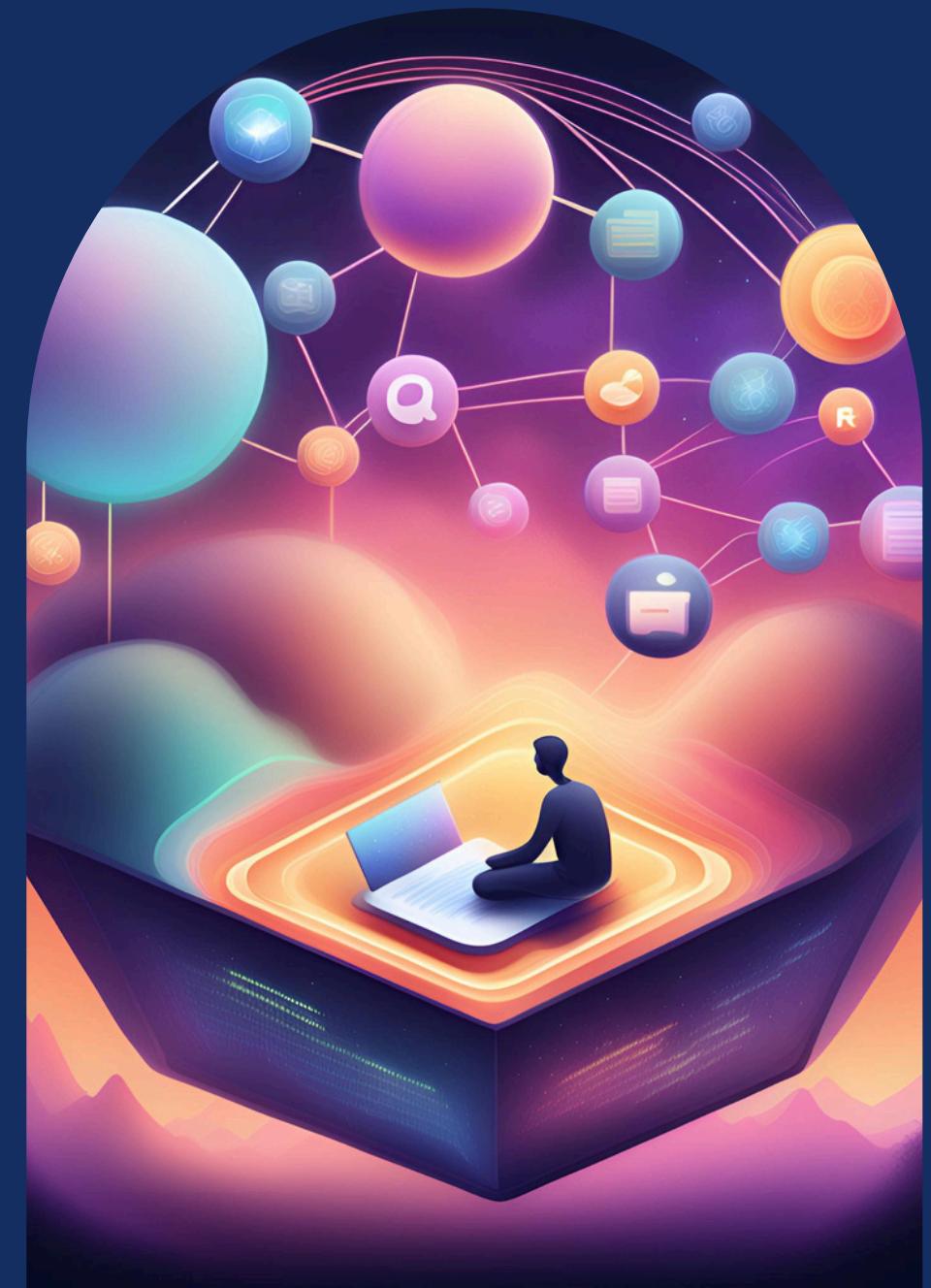


Common Language Runtime

- Compilador Just-in-time ligero: MSIL a lenguaje de máquina nativo.
- El código copilado es transformado dentro de un lenguaje intermedio llamado Microsoft Intermediate Language (MSIL o IL).
- Un integer en Visual Basic .Net o un int en C# son convertidos a un mismo tipo de dato .NET, conocido como Int32.
- El IL creado es el mismo para todos los lenguajes.
- El assembly es compilado en el programa .NET.
- En assembly contiene el IL con la información adicional llamada metadata.
- Se usa el IL Disassembler (ildasm.exe) para ver el IL dentro del assembly.

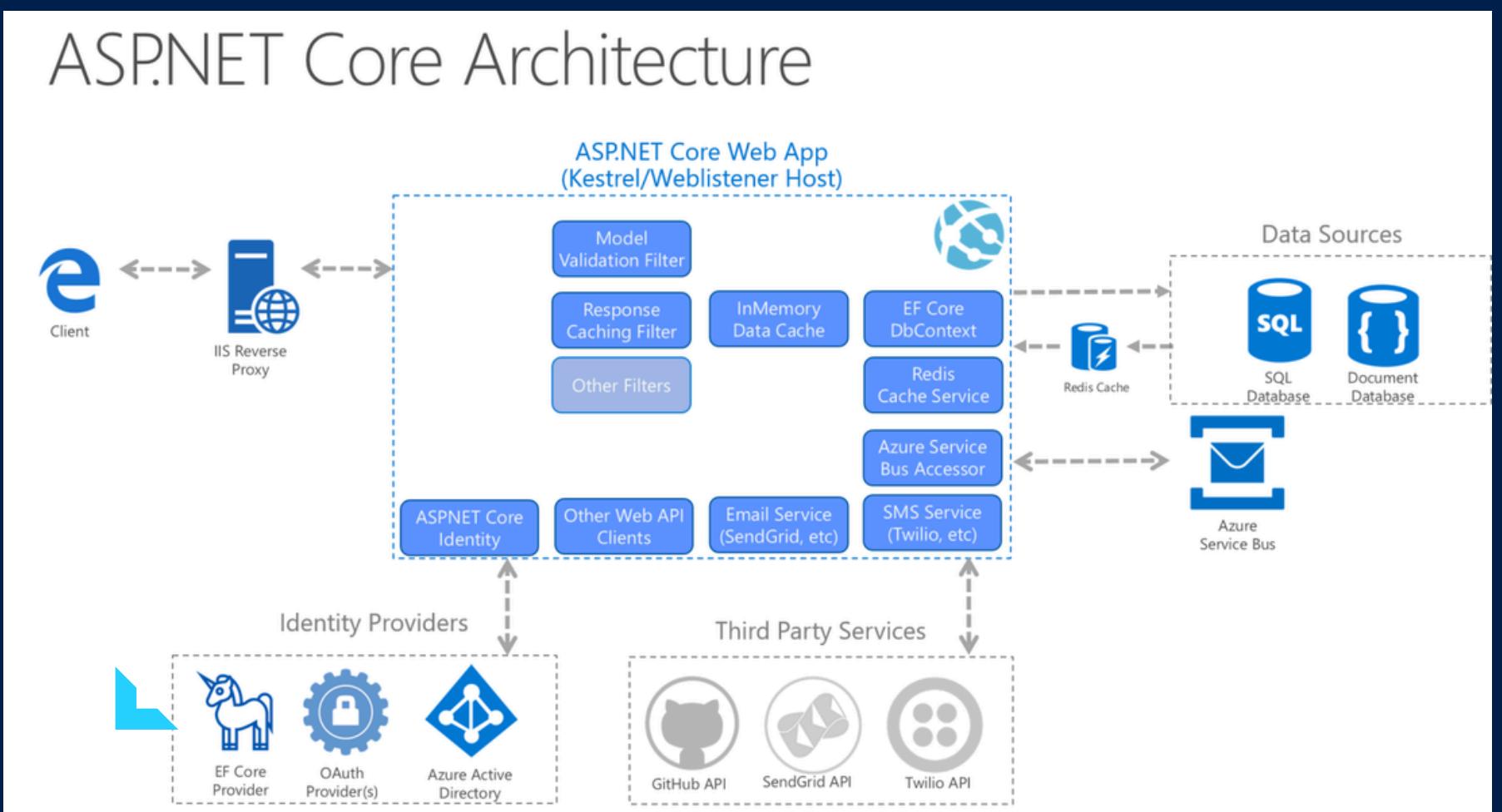
02

ARQUITECTURA



ARQUITECTURA EN APLICACIONES MONOLÍTICAS

Una aplicación monolítica es completamente independiente en su comportamiento. Aunque puede interactuar con otros servicios o almacenes de datos, su núcleo opera dentro de su propio proceso y se implementa como una única unidad.



Una aplicación monolítica es completamente independiente en su comportamiento.

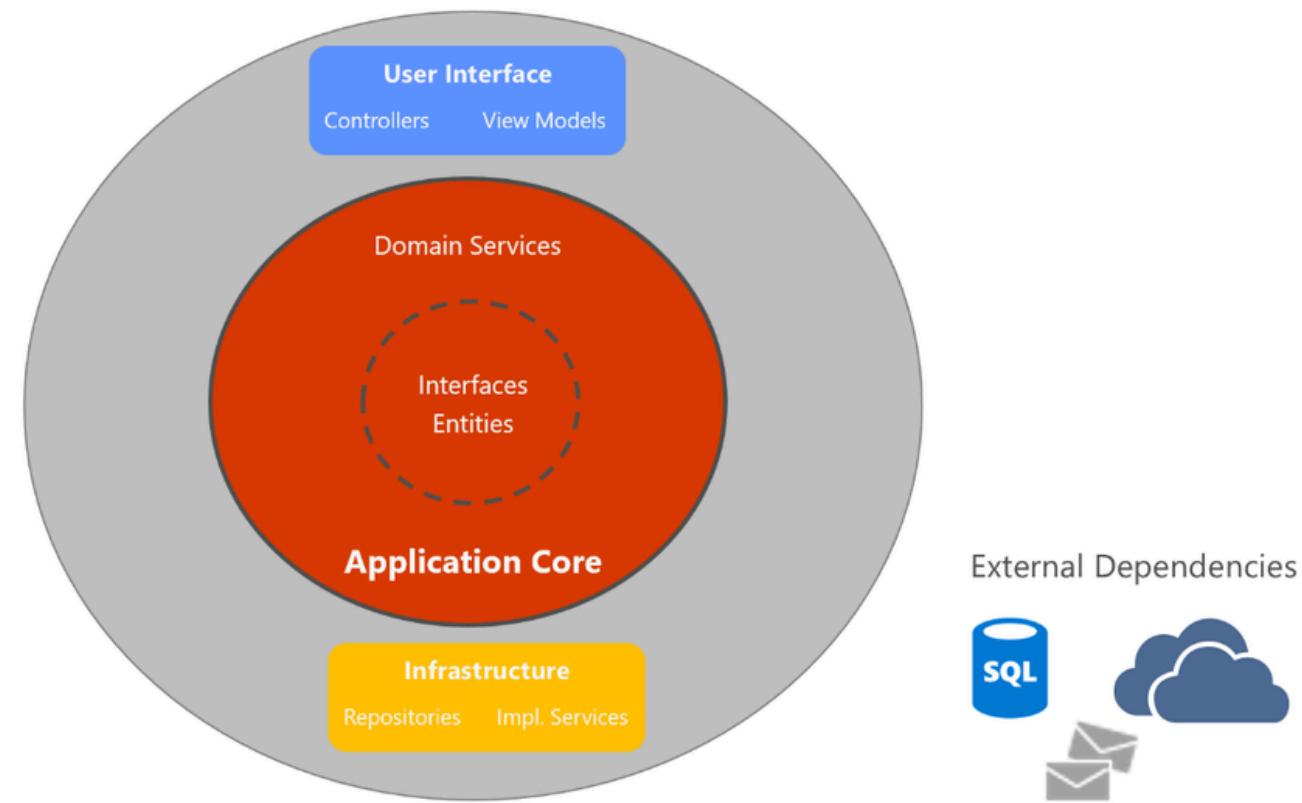
Aunque puede interactuar con otros servicios o almacenes de datos, su núcleo opera dentro de su propio proceso y se implementa como una única unidad. Para escalar horizontalmente, la aplicación completa se duplica en varios servidores o máquinas virtuales.



ARCHITECTURA

Arquitectura Limpia

Clean Architecture Layers (Onion view)



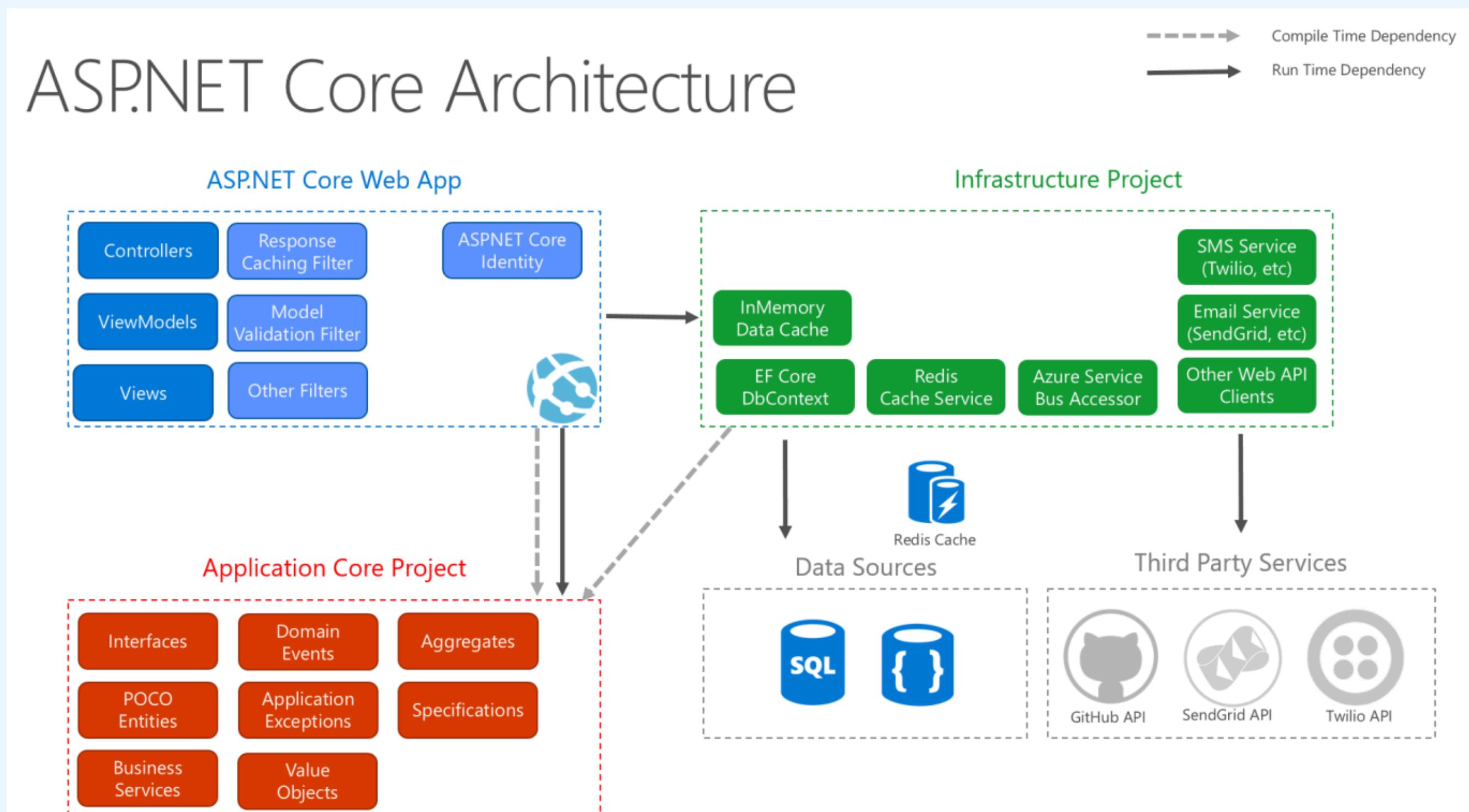
Esta arquitectura coloca el modelo de la lógica de negocios y aplicación en el centro de la aplicación. El acceso a los datos y otros aspectos de infraestructura dependen de la lógica del negocio.

En este diagrama, las dependencias fluyen hacia el círculo más interno, donde se ubica el núcleo de la aplicación.

Este núcleo no tiene dependencias de otros niveles. Las entidades e interfaces se encuentran en el centro, y los servicios de dominio, que implementan interfaces del núcleo, están alrededor. Fuera del núcleo, las capas de interfaz de usuario e infraestructura dependen del núcleo, pero no entre sí.

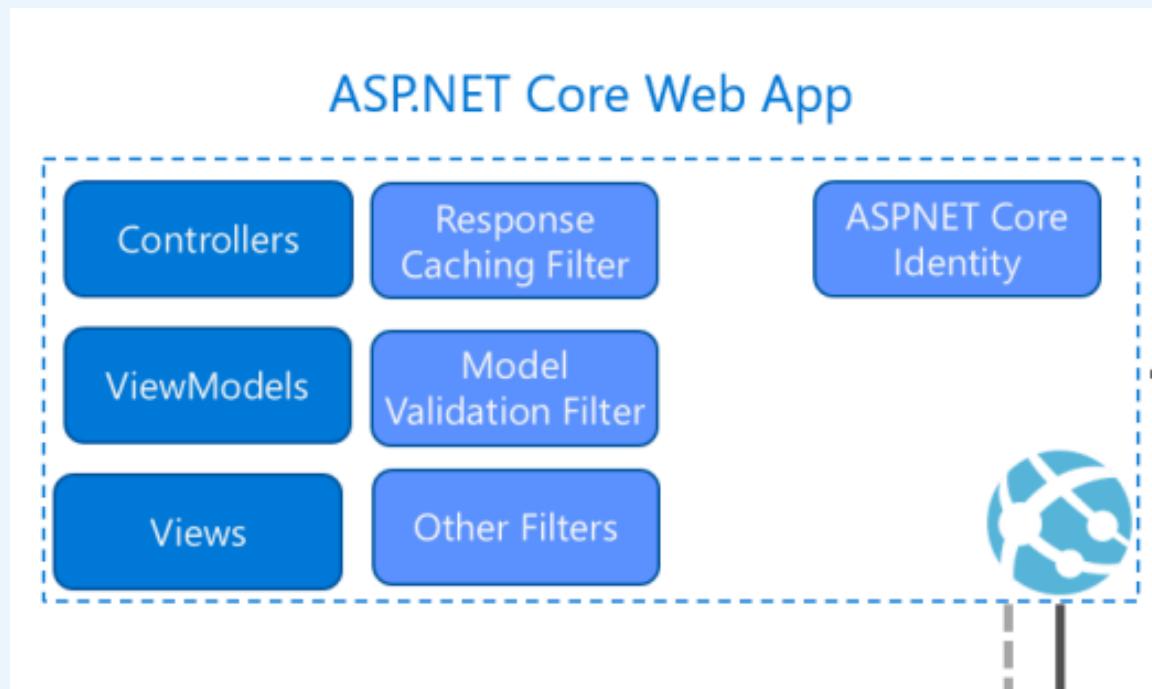


ARCHITECTURA ASP.NET CORE





ARCHITECTURA ASP.NET CORE

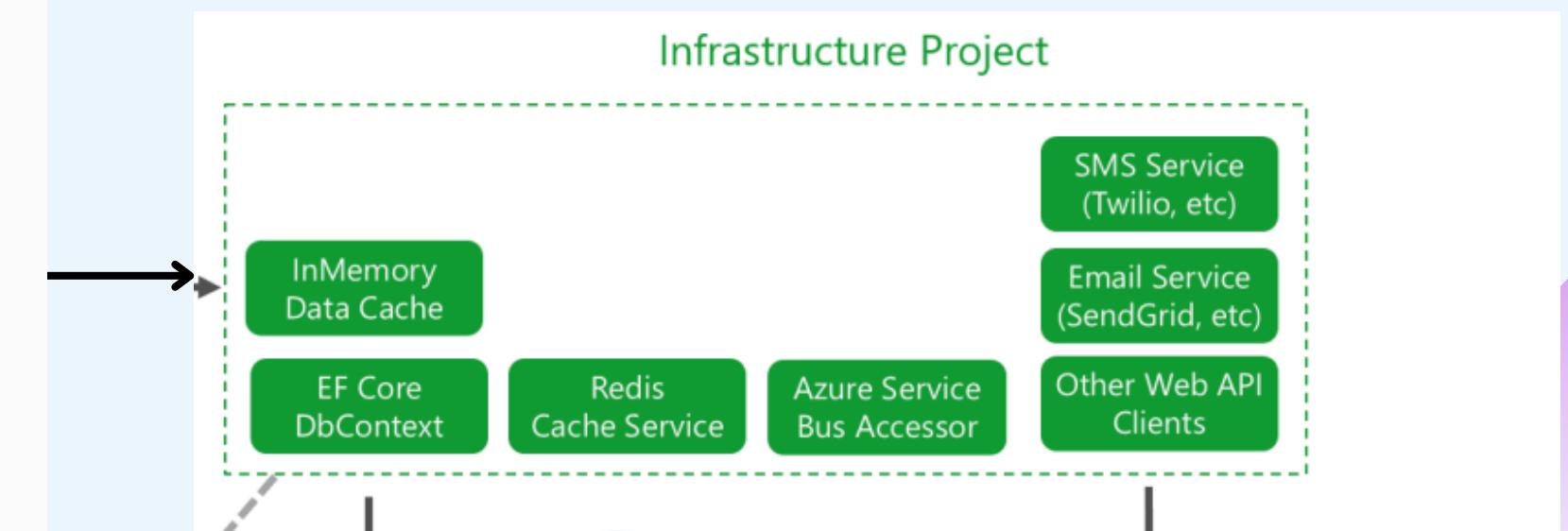


ASP.NET Core Web App

- Componentes como “Controllers,” “ViewModels,” “Views” y “Other Filters.”
- Estos están conectados a un cuadro central llamado “ASP.NET Core Middleware Pipeline,” que representa el flujo de datos a través de varios middleware.

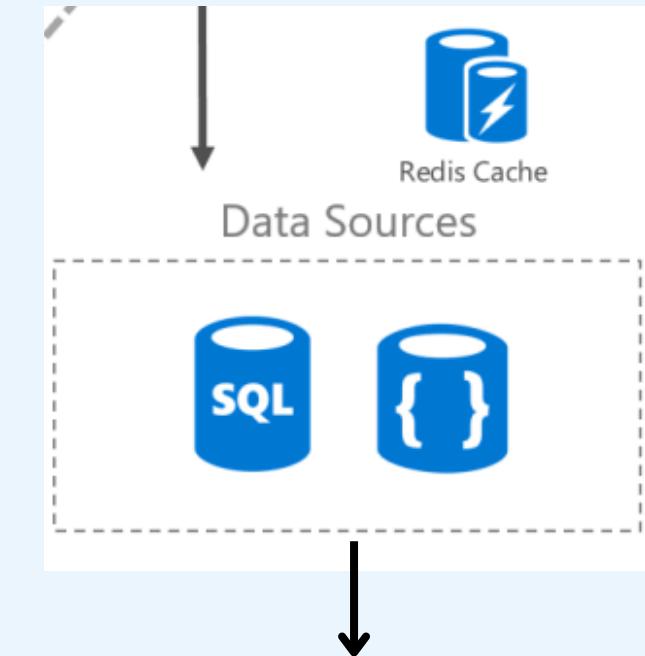
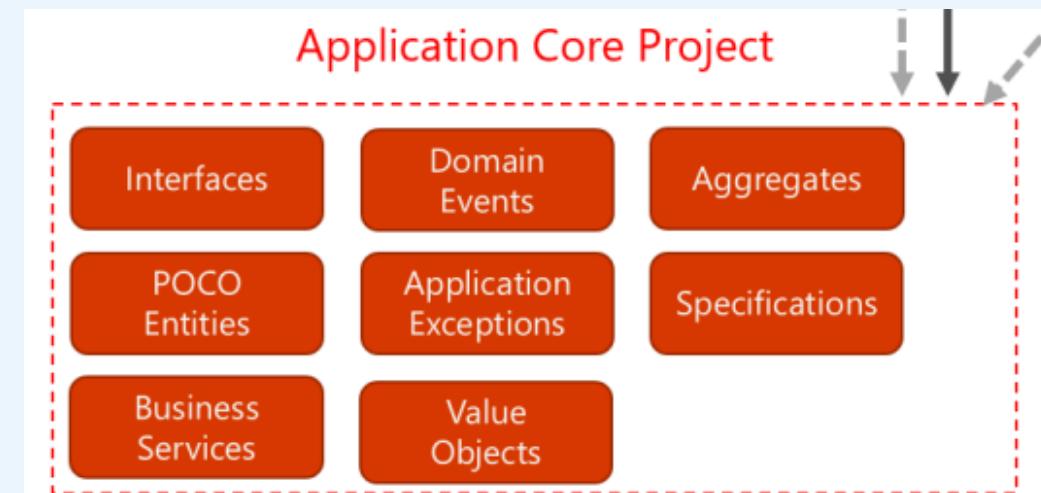
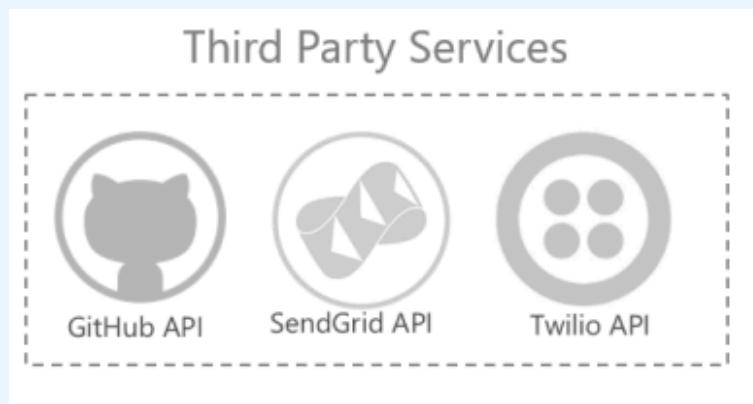
Infrastructure Project

- Servicios como “SMS Service” (Twilio), “Email Service” (SendGrid), “In-Memory Cache,” “Redis Cache,” y “EF Core DBContexts” que se conectan a fuentes de datos como SQL Server.
- Entity Framework (EF) Core.





ARCHITECTURA ASP.NET CORE



Third Party Services

Se representan servicios externos como “GitHub API” y “SendGrid API” que pueden integrarse en una aplicación ASP.NET Core.

Application Core Project

Es donde reside la lógica de negocio fundamental de tu aplicación ASP.NET Core

- Application Domain Projects
- Application Specification Projects
- Infrastructure Projects.

Data Sources

Son fuentes de datos que una aplicación utiliza para almacenar y recuperar información.





03

LENGUAJE Y ECOSISTEMA





LENGUAJE

Principal Lenguaje Utilizado:

ASP.NET Core utiliza principalmente C#. C# es un lenguaje de programación moderno, orientado a objetos y desarrollado por Microsoft, que se ejecuta en el framework .NET.



x ● ○
● ○ +
+ + ×

Características del lenguaje



- Tipado Estático y Seguro: Ayuda a detectar errores en tiempo de compilación.
- Programación Orientada a Objetos: Facilita la organización y reutilización del código.
- Rendimiento: C# y .NET Core están optimizados para rendimiento y pueden competir con lenguajes como Java en aplicaciones de alta carga.
- Interoperabilidad: Puede interactuar fácilmente con otros lenguajes y tecnologías, gracias a .NET.



ASP.NET

Lenguajes

- Además de C#, ASP.NET Core también es compatible con otros lenguajes de programación, aunque C# es el más utilizado y recomendado.
- Los otros dos lenguajes principales que pueden utilizarse con ASP.NET Core son: Visual Basic y F#

F#

- Sintaxis ligera

```
let names = [ "Ana"; "Felipe"; "Emilia" ]  
  
for name in names do  
    printfn $"Hello {name}"
```

Visual Basic

- Lenguaje Orientado a objetos

```
Dim names As New List(Of String){  
    "Ana",  
    "Felipe",  
    "Emilia"  
}  
  
For Each name In names  
    Console.WriteLine($"Hello {name}")  
Next
```

Ecosistema y Herramientas

- Bibliotecas y Extensiones
- Herramientas de Desarrollo
- Integración con Otros Servicios





Bibliotecas y Extensiones

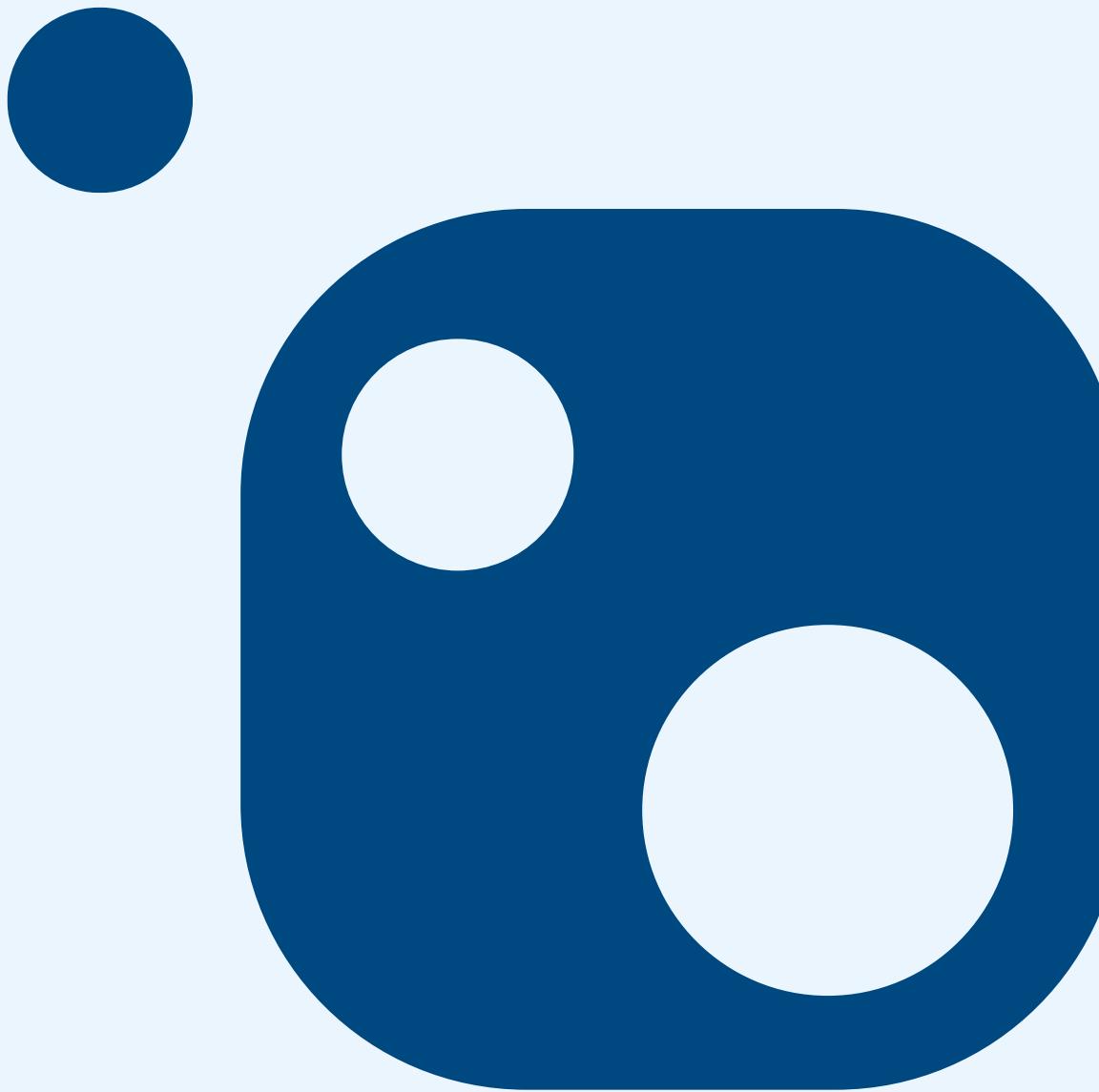
ASP.NET CORE CUENTA CON UN AMPLIO ECOSISTEMA DE BIBLIOTECAS Y PAQUETES NUGET QUE EXTIENDEN SU FUNCIONALIDAD. COMO POR EJEMPLO:

- Entity Framework Core para ORM
- Serilog para logging
- AutoMapper para mapeo de objetos.





ECOSISTEMA



Gestión de Paquetes

ASP.NET CORE UTILIZA NUGET PARA LA
GESTIÓN DE PAQUETES. NUGET
PERMITE INSTALAR, ACTUALIZAR Y
GESTIONAR DEPENDENCIAS DE
MANERA EFICIENTE.





ECOSISTEMA

Swagger

SWAGGER ES UN CONJUNTO DE HERRAMIENTAS Y ESPECIFICACIONES
QUE AYUDAN A DISEÑAR, CONSTRUIR, DOCUMENTAR Y CONSUMIR
APIS RESTFUL DE MANERA EFICIENTE.





ECOSISTEMA



Swagger

SWAGGER OFRECE UNA INTERFAZ DE USUARIO WEB INTERACTIVA, LLAMADA SWAGGER UI, QUE PERMITE A LOS DESARROLLADORES Y USUARIOS EXPLORAR Y PROBAR LOS ENDPOINTS DE UNA API DIRECTAMENTE DESDE EL NAVEGADOR.



ESTO HACE QUE SEA FÁCIL VER CÓMO FUNCIONA LA API Y PROBAR DIFERENTES LLAMADAS SIN NECESIDAD DE ESCRIBIR CÓDIGO ADICIONAL.



Herramientas de Desarrollo

VISUAL STUDIO Y VISUAL STUDIO CODE SON LAS HERRAMIENTAS MÁS UTILIZADAS PARA EL DESARROLLO CON ASP.NET CORE. ESTAS IDES OFRECEN INTEGRACIÓN COMPLETA CON .NET CORE COMO POR EJEMPLO:

- IntelliSense
- Debugging
- Refactorización





ECOSISTEMA

Integración con Otros Servicios

ASP.NET CORE SE INTEGRA FÁCILMENTE CON
SERVICIOS DE AZURE COMO:

- Azure SQL Database
- Azure Storage
- Azure App Services





ASP.NET

04

REQUISITOS PARA INSTALACIÓN





Software Necesario

- .NET SDK: Debe instalarse el SDK de .NET Core, disponible en la página oficial de Microsoft.
- IDE: Visual Studio 2019 o posterior (para una experiencia completa) o Visual Studio Code (para una opción más ligera).





REQUISITOS

Dependencias del Sistema

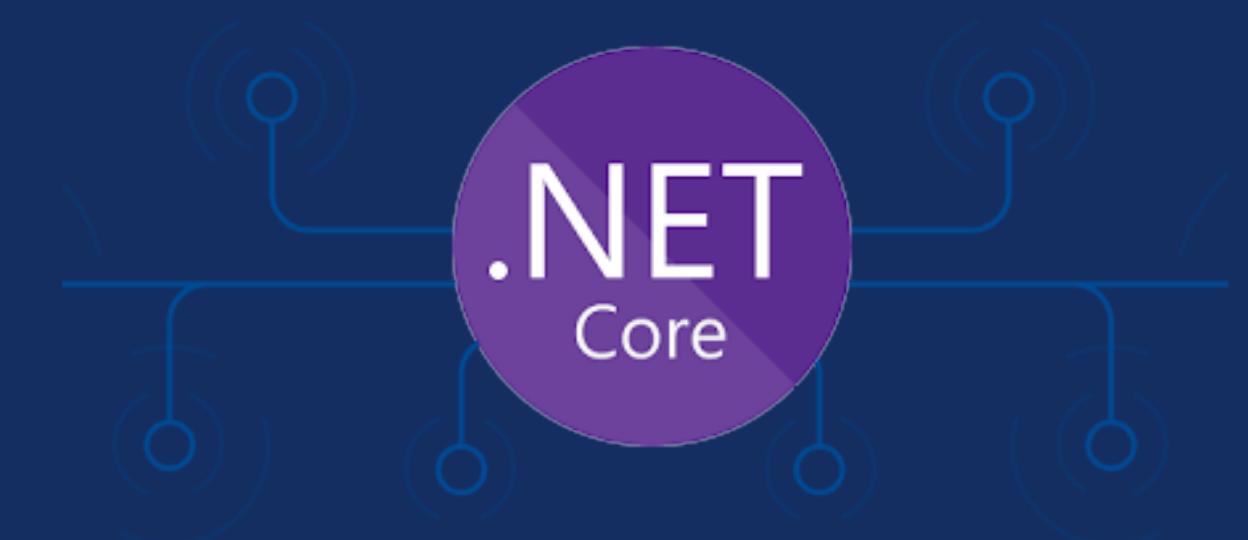


No hay dependencias adicionales significativas para empezar, pero dependiendo del proyecto, pueden ser necesarias bases de datos como SQL Server u otros.

05

PRINCIPALES CARACTERÍSTICAS

ASP.NET



Lenguajes de Programación



CARACTERÍSTICAS

Soporte para múltiples lenguajes:

- Principalmente C# y VB.NET.

Flexibilidad:

- Elección del lenguaje según preferencias del desarrollador
 - Integración con otras tecnologías .NET.

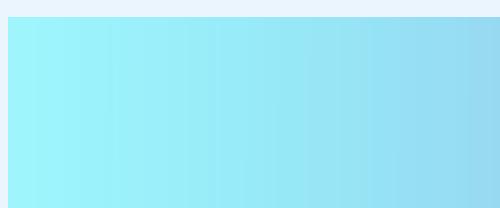
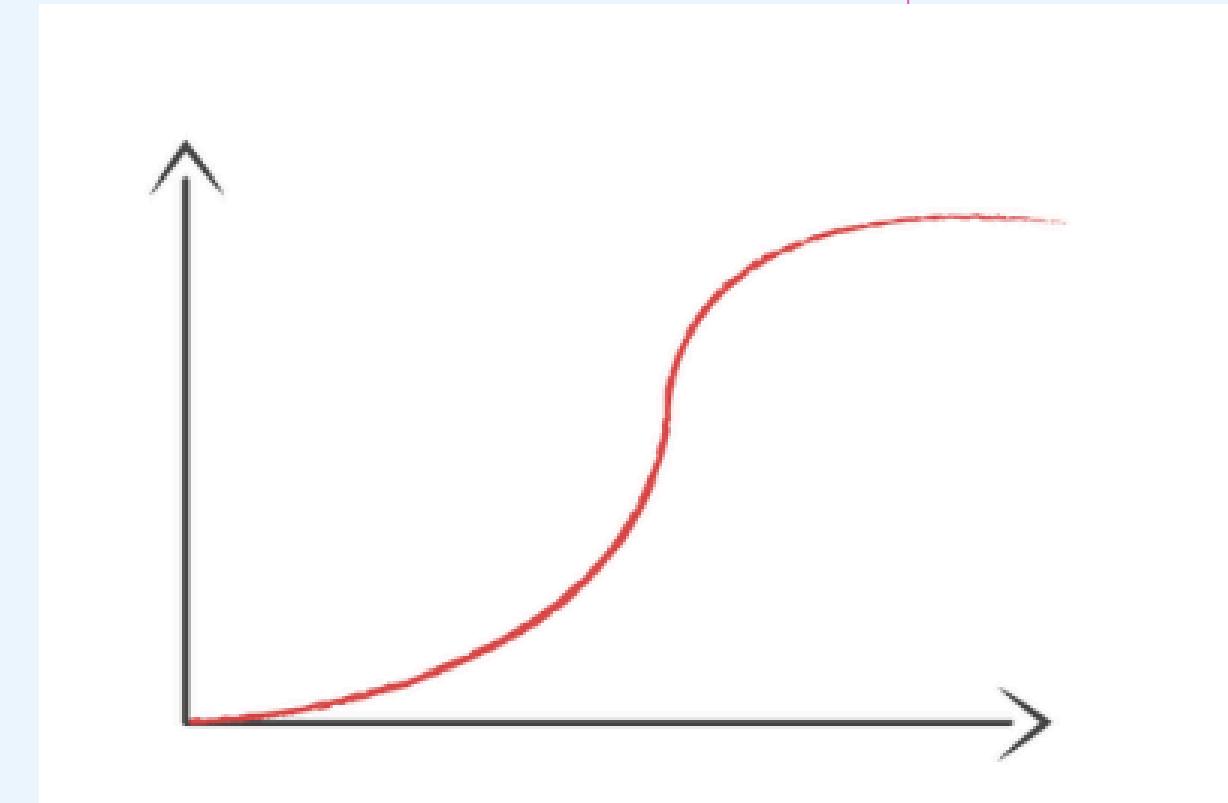


Curva de Aprendizaje



CARACTERÍSTICAS

- Se considera un framework de desarrollo web de curva de aprendizaje moderada a alta.
- Familiaridad con C# - necesario aprender el lenguaje.
- Recursos para aprender:
 - Documentación amplia, tutoriales, libros, etc.



Modelo de Programación Unificado



CARACTERÍSTICAS

ASP.NET MVC:

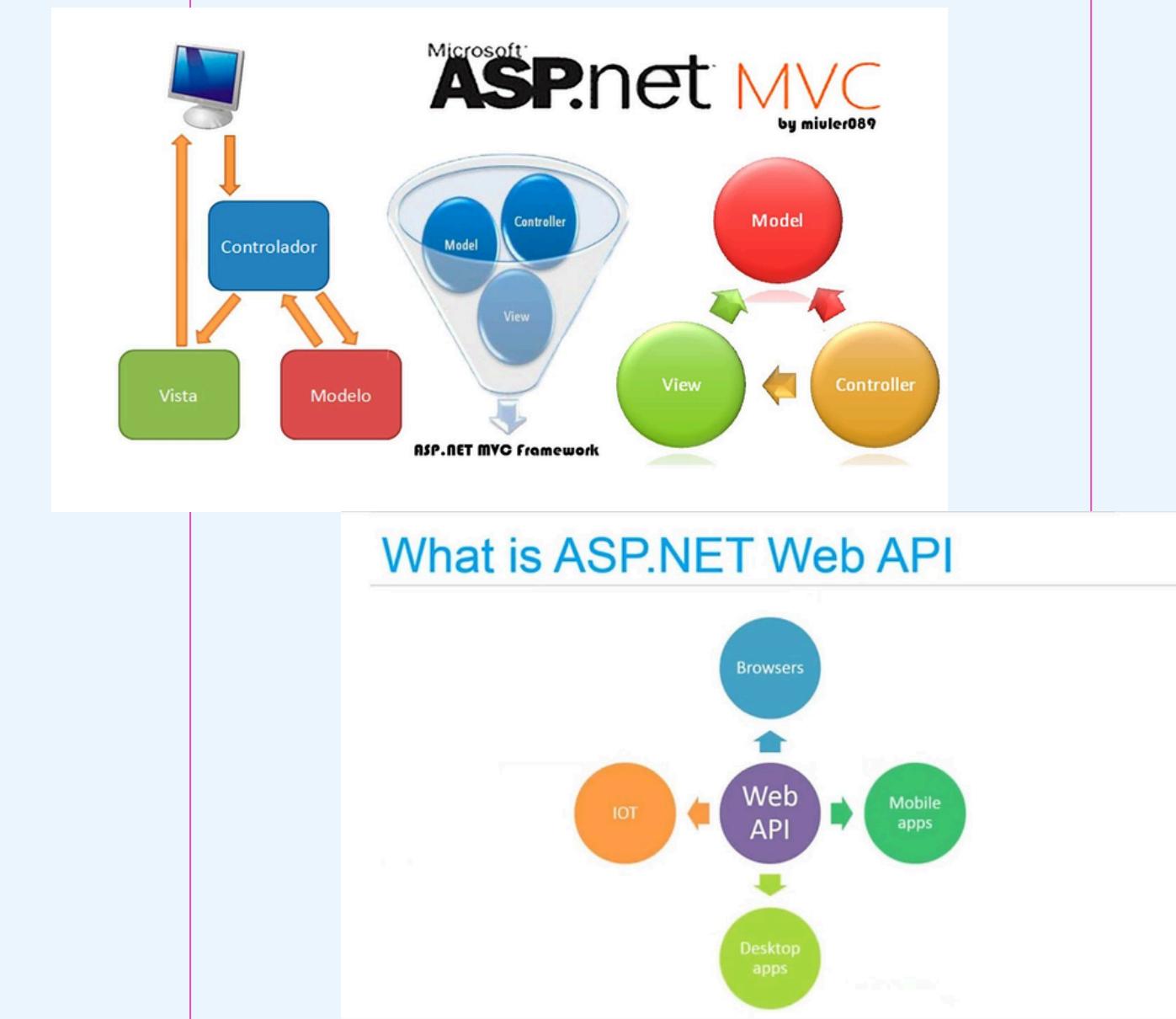
- Patrones de diseño Model-View-Controller.
- Separación de preocupaciones.
- Modularidad y facilidad de pruebas.

ASP.NET Web API:

- Creación de API RESTful.
- Consumo desde aplicaciones web y móviles.

ASP.NET Web Forms:

- Desarrollo basado en formularios.
- Controles de servidor.
- Manejo automático del estado.



Desempeño y Escalabilidad

Alto rendimiento:

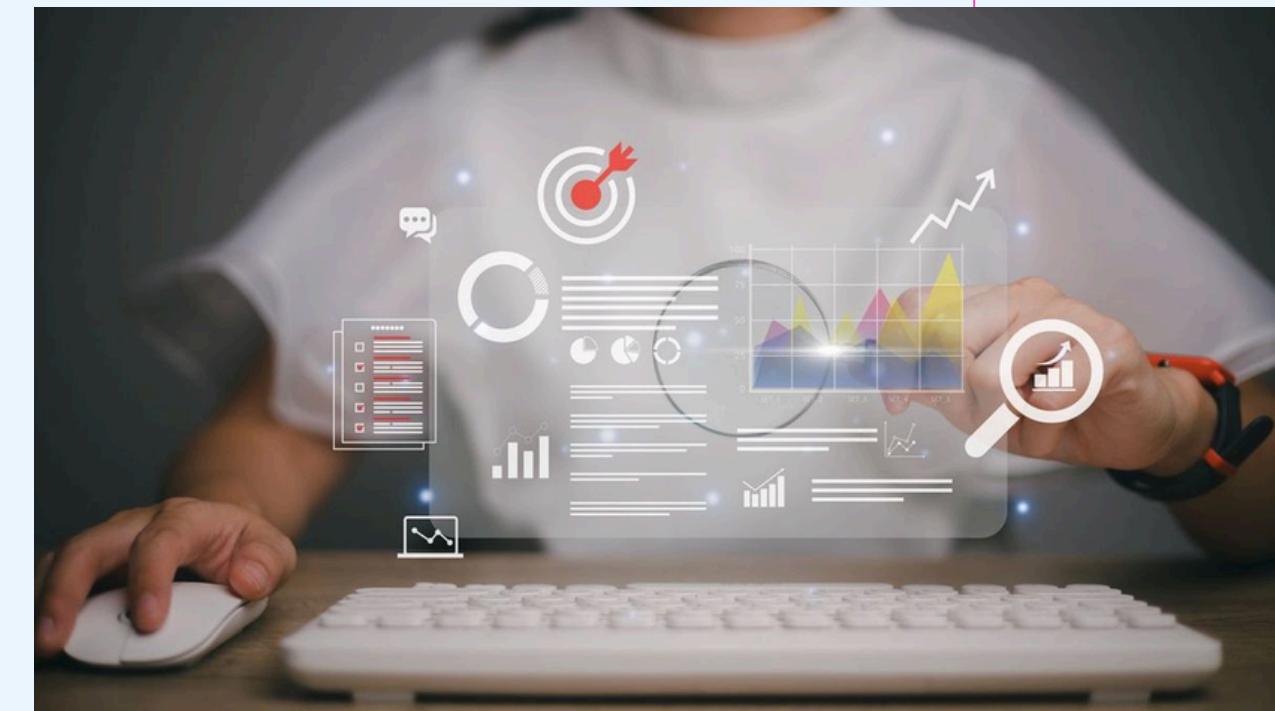
- Optimización de código.
- Manejo eficiente de recursos.

Escalabilidad:

- Soporte para grandes volúmenes de tráfico.
- Capacidad de manejar múltiples usuarios simultáneos.



CARACTERÍSTICAS



Integración con Visual Studio



CARACTERÍSTICAS

Herramientas avanzadas:

- Depuración integrada.
- Diseño y gestión de proyectos.

Productividad:

- Acceso a múltiples plantillas y ejemplos.
- Integración con servicios en la nube.



Seguridad



CARACTERÍSTICAS

Autenticación y autorización:

- Múltiples métodos de autenticación (Forms, Windows, OAuth).

Protección contra ataques:

- XSS (Cross-Site Scripting).
- CSRF (Cross-Site Request Forgery).

XSS VS CSRF

Feature	Cross-Site Scripting (XSS)	Cross-Site Request Forgery (CSRF)
User Interaction	Does not require user interaction beyond visiting a compromised site.	Typically involves the user performing an action.
Session Requirement	Does not require an active session with the targeted site.	Requires the user to have an active session with the targeted site.
Trust Exploitation	Exploits the trust a user has in a website.	Exploits the trust a website has in a user's browser.
Scope of Actions	Can execute arbitrary scripts.	Restricted to actions that a user can perform.
Attack Initiation	Requires a vulnerability in the website.	Requires the user to access a malicious page or click a link.
Data Access	Can send and receive HTTP requests and responses.	Can only send HTTP requests and cannot view the response.

Administración del Estado



CARACTERÍSTICAS

Estado de sesión:

- Mantenimiento de datos específicos del usuario.

Estado de aplicación:

- Mantenimiento de datos compartidos entre usuarios.

Cookies:

- Almacenamiento de información persistente del usuario.

Característica	Estado de sesión	Estado de aplicación	Cookies
Propósito	Mantener datos específicos del usuario	Mantener datos compartidos entre usuarios	Almacenar información persistente del usuario
Alcance	Interacción individual del usuario	Toda la aplicación	A través de múltiples interacciones
Persistencia	Temporal	Persistente	Persistente
Ubicación	Servidor	Servidor	Cliente
Ejemplo	Detalles del pedido del cliente en una panadería	Libro de recetas compartido en una panadería	Información de la tarjeta de fidelización del cliente

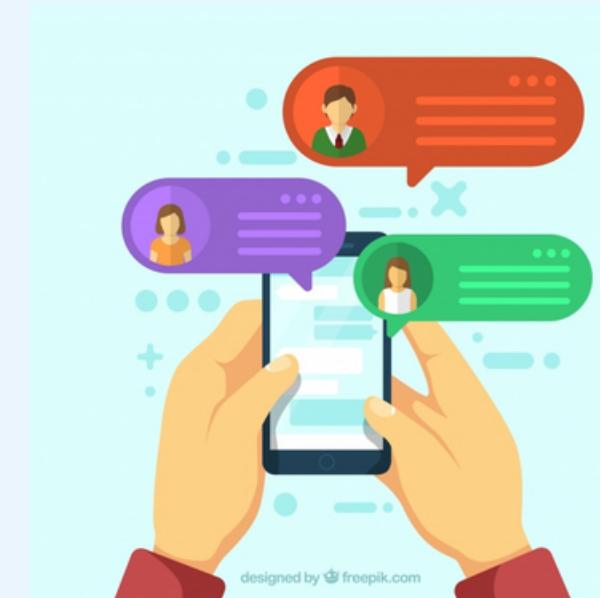
Soporte para AJAX



CARACTERÍSTICAS

Interactividad:

- Actualización parcial de páginas.
- Mejora de la experiencia del usuario.



Receptividad:

- Respuesta rápida a las interacciones del usuario.
- Carga dinámica de contenido.



Bibliotecas y Frameworks Adicionales

ASP.NET Identity:

- Gestión de identidades y roles.

ASP.NET SignalR:

- Comunicación en tiempo real.
- Soporte para WebSockets.



CARACTERÍSTICAS



**ASP.NET Core
Identity**

User Registration
Login and Logout

ASP.NET SignalR



Despliegue y Configuración Simplificados

Herramientas integradas:

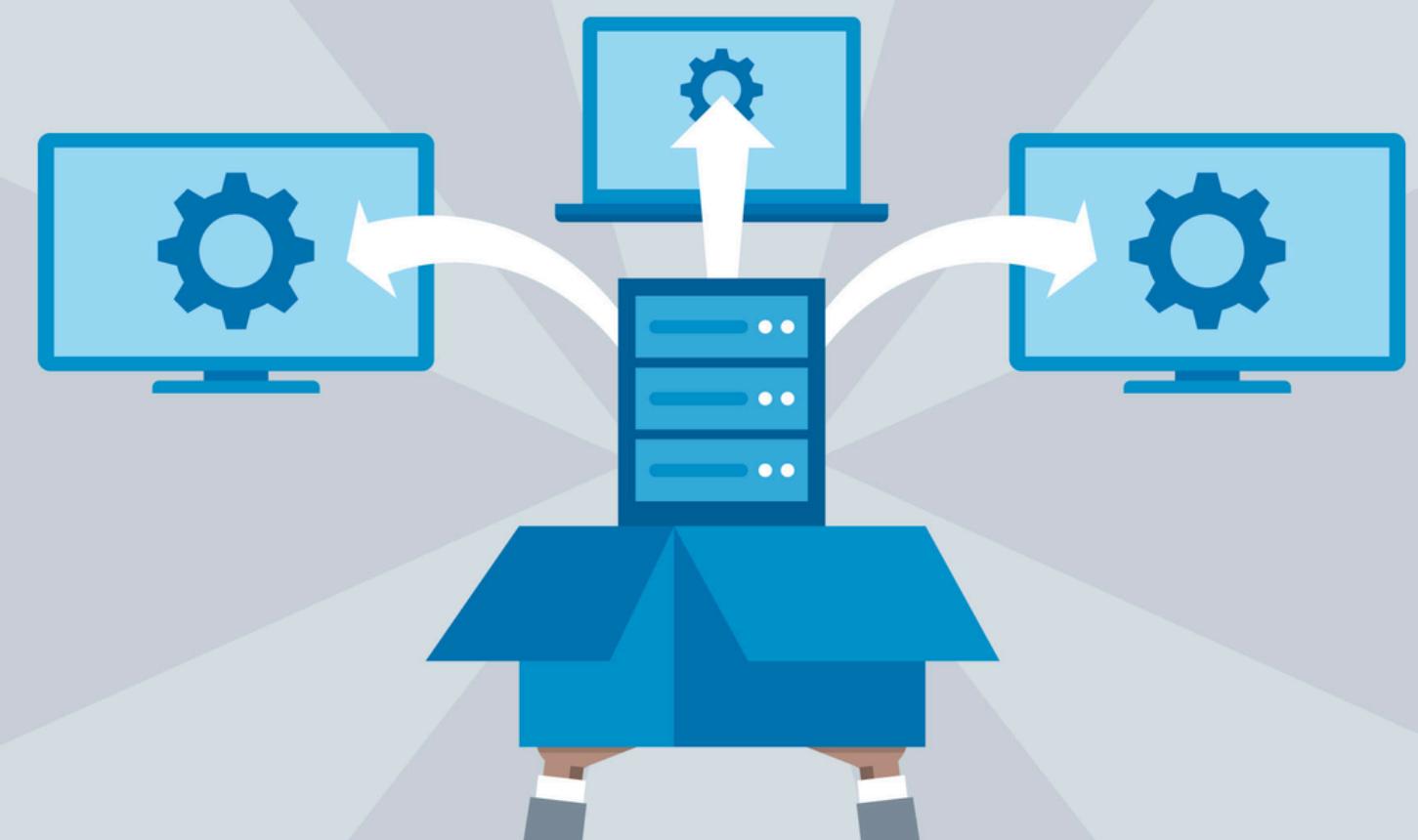
- Publicación directa desde Visual Studio.

Opciones de configuración:

- Configuración basada en XML.
- Facilidad para ajustar parámetros de despliegue.



CARACTERÍSTICAS



Compatibilidad con .NET Core



CARACTERÍSTICAS

Multiplataforma:

- Compatible con Windows, macOS y Linux.



Flexibilidad y opciones de despliegue:

- Posibilidad de elegir el entorno de ejecución.
- Mayor control sobre la infraestructura.

06

CASOS DE USO



ASP.NET



STACK OVERFLOW



CASOS DE USO

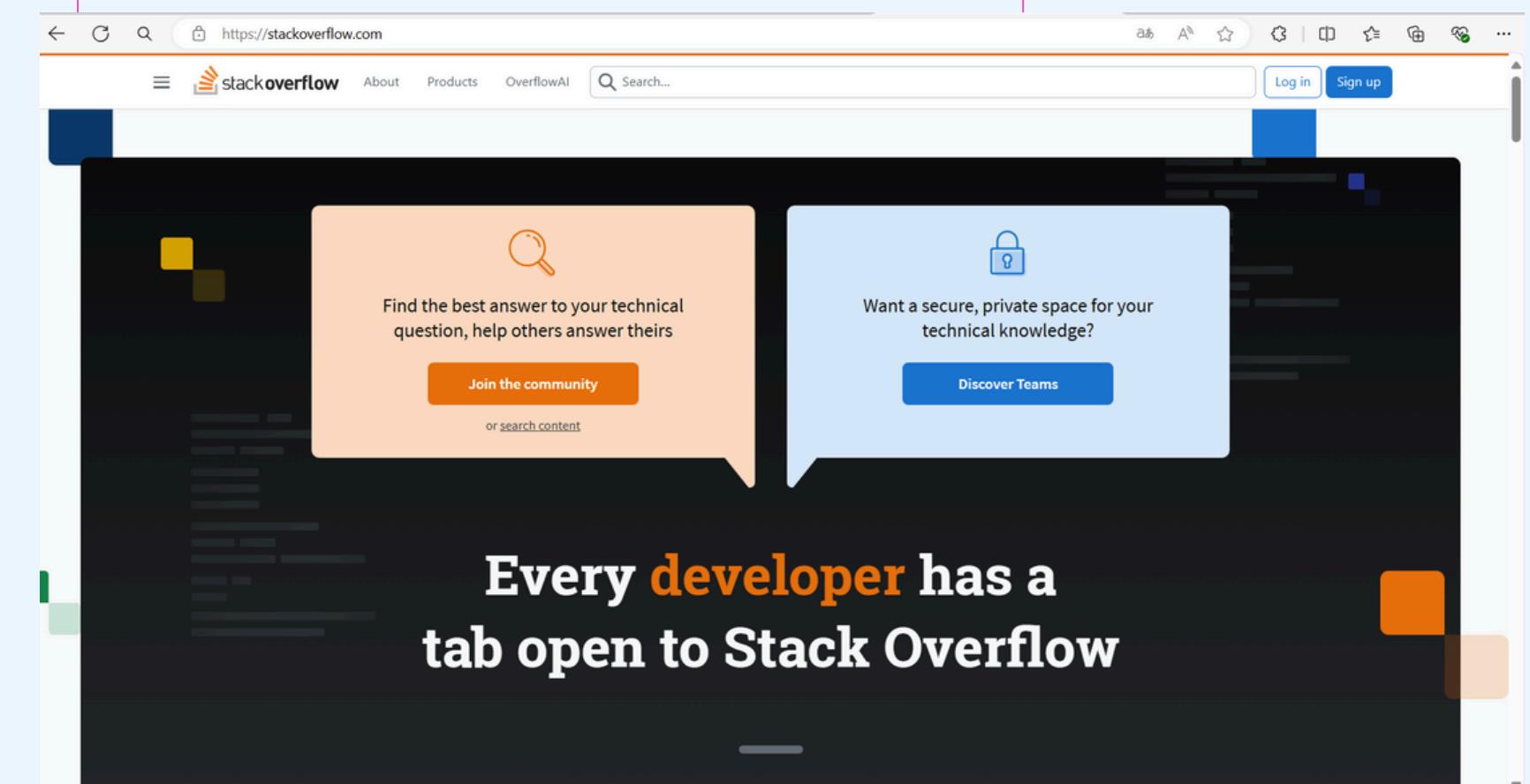
- Uno de los sitios de preguntas y respuestas más grandes y populares para desarrolladores.

- **Características Implementadas:**

- Administración de usuarios y roles.
- Interfaz de usuario dinámica con AJAX.
- API para integración con otros servicios.

- **Beneficios:**

- Alto rendimiento y escalabilidad.
- Seguridad robusta.
- Fácil mantenimiento y actualización.



MICROSOFT



CASOS DE USO

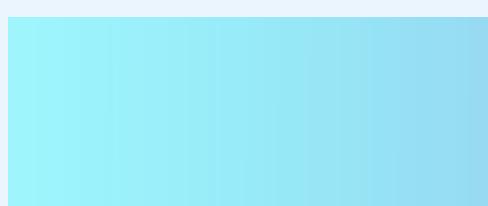
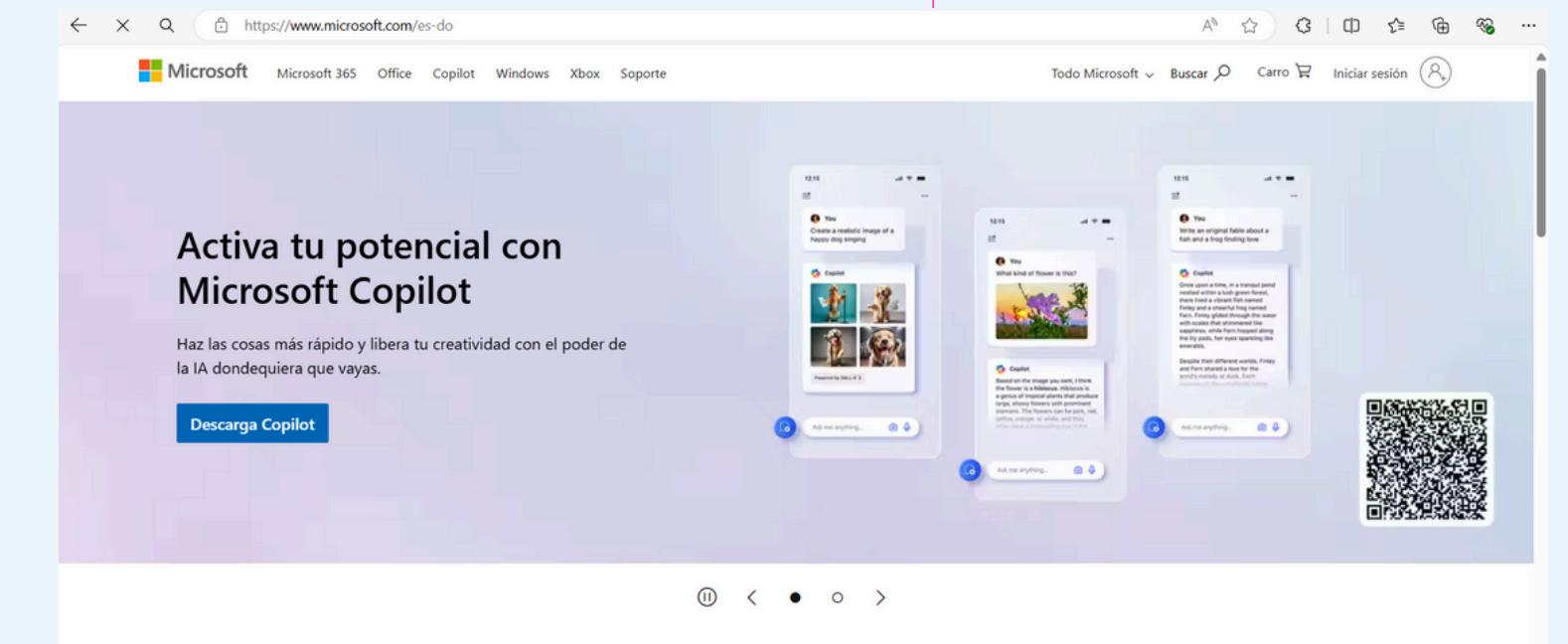
- Utiliza ASP.NET para varios de sus productos y servicios, incluyendo el sitio web principal de Microsoft.

Características Implementadas:

- Gran cantidad de contenido dinámico.
- Integración con múltiples servicios de Microsoft.
- Altos estándares de seguridad.

Beneficios:

- Integración perfecta con otros servicios de Microsoft.
- Alta disponibilidad y escalabilidad.
- Soporte y mantenimiento continuo.



GODADDY



CASOS DE USO

Utiliza ASP.NET para administrar su plataforma de alojamiento web y servicios de dominio.

Características Implementadas:

- Gestión de dominios y alojamiento.
- Panel de control de usuarios dinámico.
- Integración con múltiples servicios de terceros.

Beneficios:

- Fiabilidad y tiempo de actividad elevado.
- Escalabilidad para soportar millones de usuarios.
- Facilidad de integración y mantenimiento.

The screenshot shows the GoDaddy website at https://www.godaddy.com/es/offers/brand/new. The top navigation bar includes links for Dominos, Sitios web y hosting, Correo electrónico, Seguridad, Marketing, Precios, Ayuda, and Iniciar sesión. A search bar says 'Escribe el dominio que deseas' with a 'Buscar dominios' button. A promotional banner for 'Páginas web + Marketing' features a woman named Lucinda Yrene La Morena smiling, with the text 'Crea tu sitio web gratis.' and a 'Comienza gratis' button. Below the banner, it says 'Sin necesidad de tarjetas de crédito.' To the right, there's a sidebar with a green box titled 'NUEVO' containing the text 'Dominios sugeridos por IA' and 'Usa la IA para encontrar los nombres perfectos para tu dominio y tu empresa.', along with a 'Búsqueda de dominios IA' button and a 'Contáctanos' button.

- El gigante de la tecnología utiliza ASP.NET para su tienda en línea y sistemas internos.

Características Implementadas:

- Catálogo de productos dinámico.
- Sistema de gestión de pedidos y clientes.
- Integración con servicios de pago y logística.

Beneficios:

- Escalabilidad para manejar picos de tráfico.
- Seguridad en transacciones.
- Experiencia de usuario mejorada con AJAX.



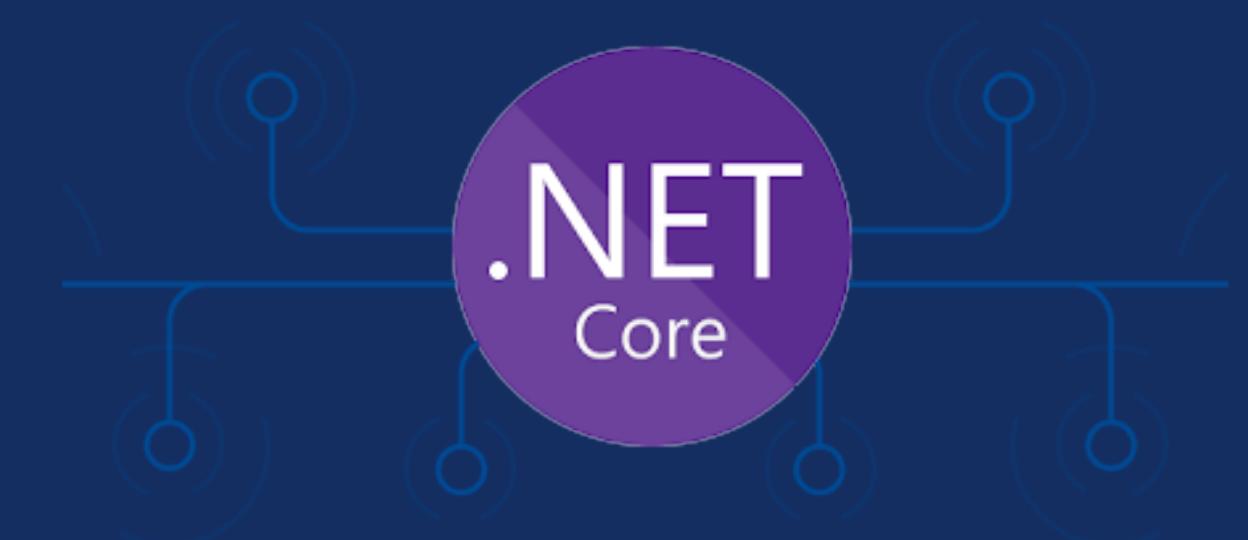
CASOS DE USO

The screenshot shows the Dell Support website. At the top, there's a navigation bar with the Dell Technologies logo, a search bar, and links for 'Iniciar sesión', 'Contáctenos', and 'EC/ES'. Below the header, a main banner features a photograph of a workspace with a computer monitor, keyboard, and a small potted plant. The text 'Bienvenido a soporte de Dell' is displayed above a paragraph about personalized support for all needs. Two buttons, 'Iniciar sesión' and 'Crear una cuenta', are visible. Below the banner, a section titled '¿Cómo lo podemos ayudar?' contains three cards: 'Controladores y diagnósticos' (with links to 'Actualizar los controladores' and 'Analizar hardware'), 'Solicitudes de servicio' (with a link to 'Visitar las solicitudes de servicio'), and 'Servicios de soporte' (with a link to 'Administrar servicios de soporte'). A blue call-to-action button is partially visible at the bottom right.



07

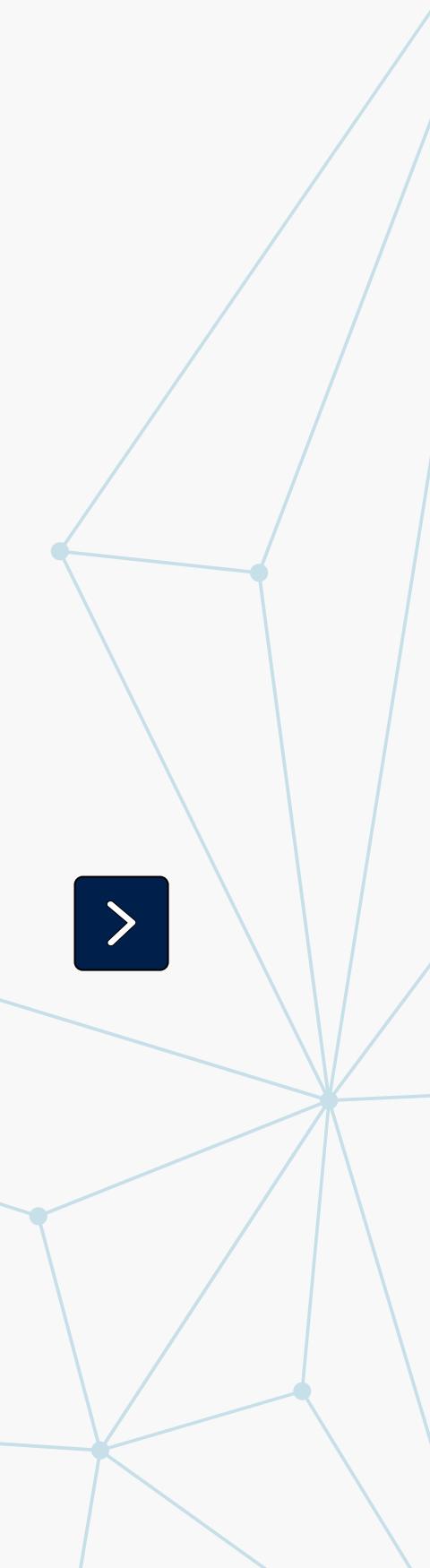
COMPARACIÓN CON OTROS FRAMEWORKS



¡La elección del framework es importante!



Cada uno con sus pros y contras, y diseñados en base a distintas necesidades y preferencias de los desarrolladores.





Evaluando aspectos como el lenguaje de programación, la plataforma, el rendimiento, la flexibilidad, la escalabilidad, las herramientas disponibles, etc., facilitando así la elección del más adecuado para tu próximo proyecto web.

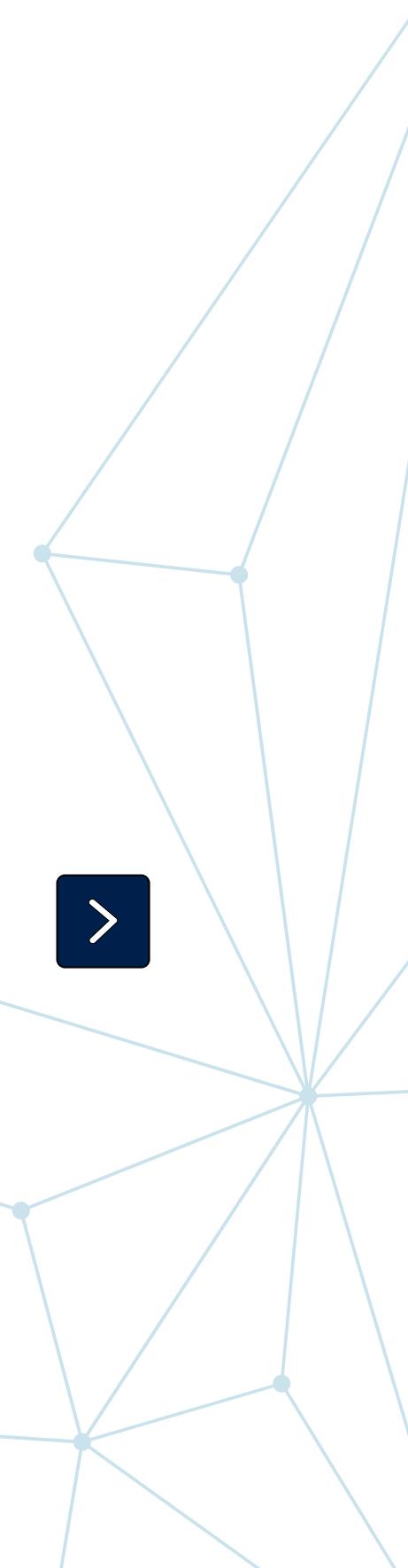
Definiciones



Marco multiplataforma de código abierto y de alto rendimiento que tiene como finalidad compilar aplicaciones conectadas a Internet y habilitadas para la nube.



Marco de aplicación web WSGI ligero . Está diseñado para que comenzar sea rápido y fácil, con la capacidad de escalar a aplicaciones complejas.





- Marco para crear aplicaciones del lado del servidor Node.js escalables y eficientes, proporciona una arquitectura de aplicaciones lista para usar.



- Marco de aplicación web con una sintaxis elegante proporciona una estructura y un punto de partida para crear aplicaciones, sin preocuparse por los detalles.



- Marco web Python de alto nivel que fomenta un desarrollo rápido, se encarga de gran parte de las molestias del desarrollo web, sin necesidad de reinventar la rueda.

Lenguaje y Plataforma

ASP.NET Core



C#

PYTHON

TYPESCRIPT / JS

PHP

PYTHON

.NET

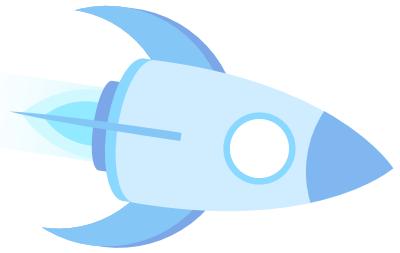
PYTHON

NODE.JS

PHP

PYTHON





Flexibilidad

ASP.NET Core

Altamente modular, permite incluir solo componentes necesarios.

Flask

Minimalista, sin imponer estructuras rígidas, permitiendo organizar y estructurar el código con libertad.

nest

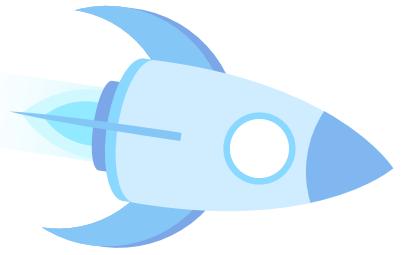
Sistema modular similar a Angular, soporta inyección de dependencias facilitando la creación de apps flexibles y mantenibles.

Laravel

Estructura modular que facilita la adición o eliminación de componentes según sea necesario.

django

Usa el patrón de diseño MVT, aunque no todos los plugins de terceros son compatibles, por lo que es un poco menos flexible.



Seguridad

ASP.NET Core

Flask

nest

Laravel

django

Autenticación y Autorización

ASP.NET Core Identity

Flask Security

Librería Passport.js

guards y providers

Sistema Integrado

Protección CSRF

Integrado

No integrado: Flask WTF

Bibliotecas adicionales: csrf

Integrado en el framework

Integrado en el framework:
CsrfViewMiddleware

Protección XSS

Integrado

Flask configura Jinga2 automáticamente.

Bibliotecas adicionales

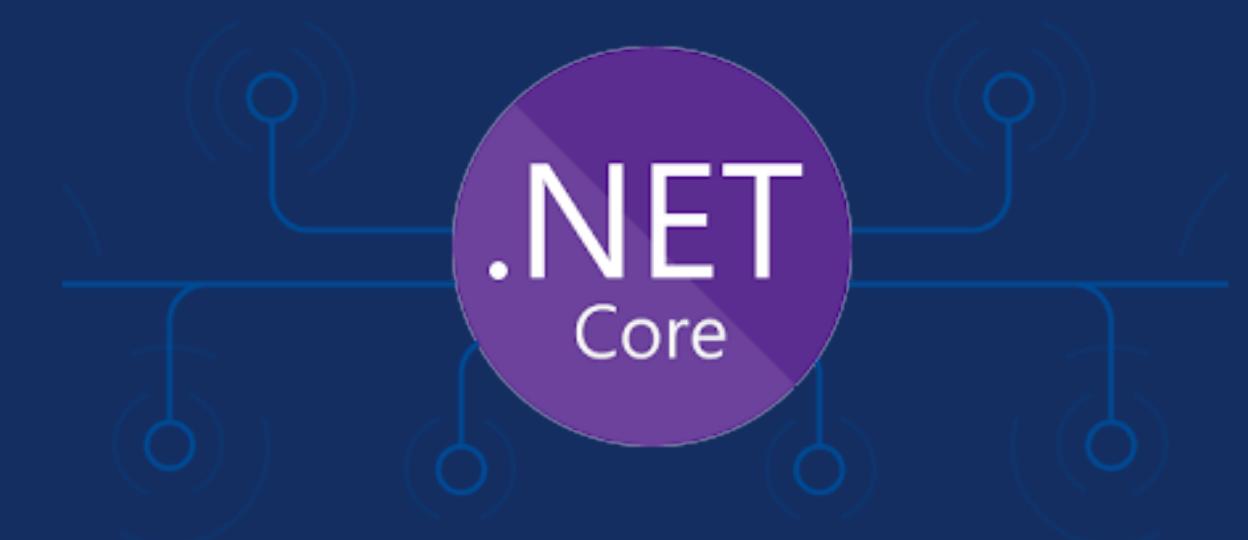
Integrado en el framework

Integrado en el framework

08

FUTURO Y ACTUALIZACIONES

ASP.NET



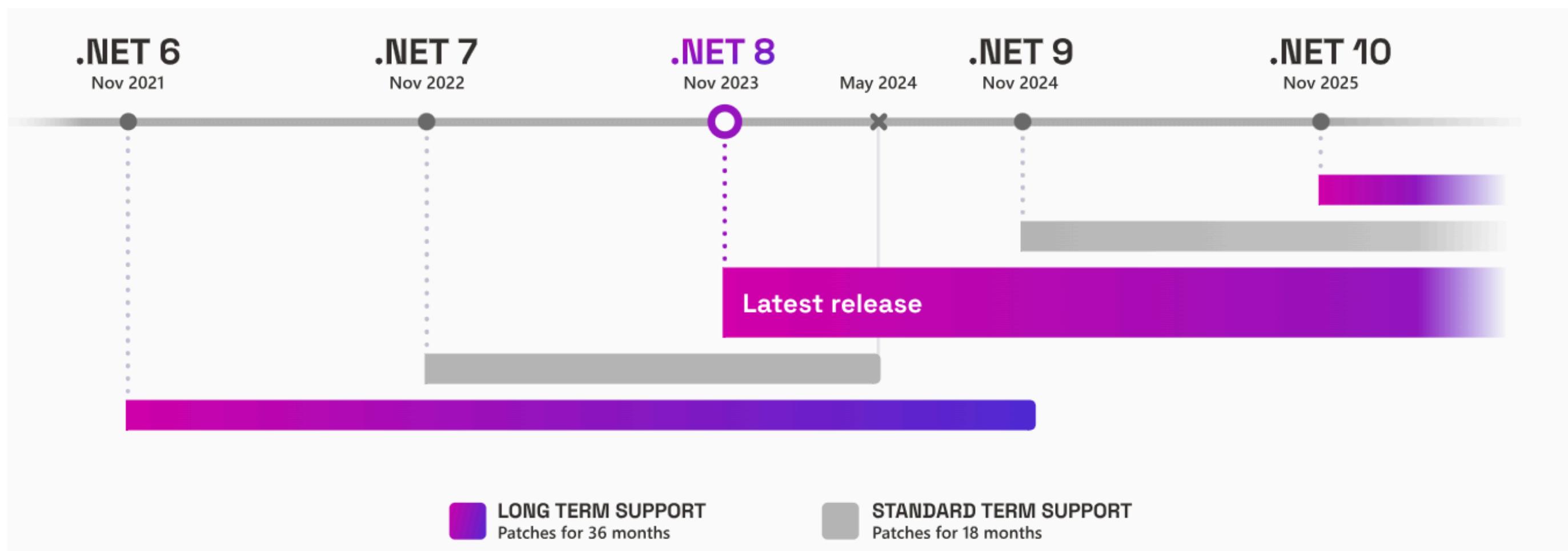
.NET desde su comienzo en 2002 se ha convertido en una de las plataformas más utilizadas y avanzadas en la industria del desarrollo.

Gracias a su capacidad para evolucionar y adaptarse a las nuevas tendencias y necesidades en el mundo de la tecnología.

En la actualidad está desempeñando un papel importante en la implementación de tecnologías emergentes como el Internet de las cosas, la inteligencia artificial y la realidad aumentada.



Cada año se publica una nueva versión principal de .NET en noviembre, lo que permite a los desarrolladores, la comunidad y las empresas planear sus planes de desarrollo.



Las versiones numeradas pares son versiones LTS y las versiones numeradas impares son versiones STS.



ASP.NET

GRACIAS POR
SU ATENCION



ASP.NET