

BIENVENIDOS  
AL CURSO:

# **ASP.NET Core Web Applications: Fundamentals**

SESIÓN 01





01

Repaso general de ASP.NET Core 5.0

---

02

Principios SOLID y MVC Pattern  
(Model, View and Controller).

---

03

ASP.NET Core 5 (MVC)

---

04

Visual Studio 2019 (soluciones, proyectos)  
e IIS (despliegue interno y externo).

---

05

NET Core CLI, NuGet, Git y GitHub  
(consola, gestor de paquetes y versiones).

---

## ÍNDICE



**.NET**

.NET es una plataforma de desarrollo gratuita, multiplataforma y de código abierto para crear muchos tipos diferentes de aplicaciones.

Con .NET, puede usar varios idiomas, editores y bibliotecas para crear para web, móvil, escritorio, juegos e IoT.

<https://dotnet.microsoft.com/>



# .NET



## Web

Build web apps and services for Windows, Linux, macOS, and Docker.



## Mobile

Use a single codebase to build native mobile apps for iOS, Android, and Windows.



## Desktop

Create beautiful and compelling desktop apps for Windows and macOS.



## Microservices

Create independently deployable microservices that run on Docker containers



## Gaming

Develop 2D and 3D games for the most popular desktops, phones, and consoles.



## Machine Learning

Add vision algorithms, speech processing, predictive models, and more to your apps.



## Cloud

Consume existing cloud services, or create and deploy your own.



## Internet of Things

Make IoT apps, with native support for the Raspberry Pi and other single-board computers.



## .NET Framework

Es un componente de software desarrollado por Microsoft que puede ser instalado en **Windows**. Provee soluciones pre-codificadas comunes de los programas y gestiona la ejecución de programas escritos para este Framework.

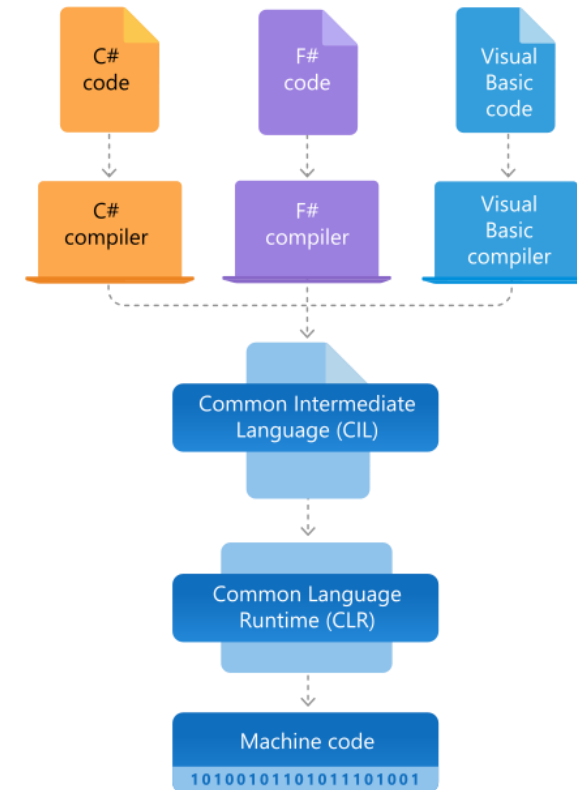
<https://docs.microsoft.com/es-es/dotnet/framework/get-started/index#Introducing>



.NET  
Frameworks

# .NET Framework

- **Common Language Runtime (CLR)** es el motor de ejecución que controla las aplicaciones en ejecución. Proporciona servicios como administración de subprocesos, recolección de elementos no utilizados, seguridad de tipos, control de excepciones y mucho más.
- **La biblioteca de clases** proporciona un conjunto de API's y tipos de funcionalidad común. Proporciona tipos para cadenas, fechas, números, etc. La biblioteca de clases incluye API's para leer y escribir archivos, conectarse a bases de datos, dibujar y mucho más.

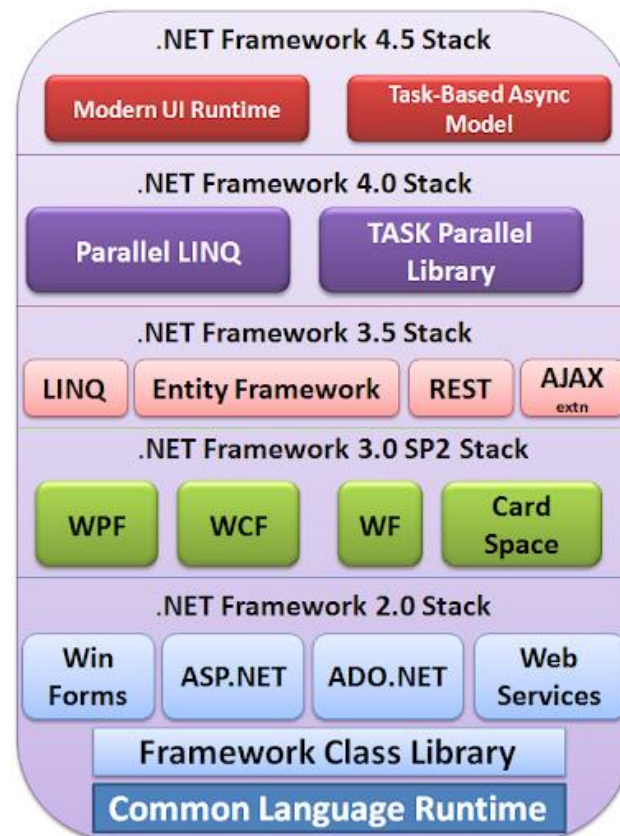




# .NET Framework



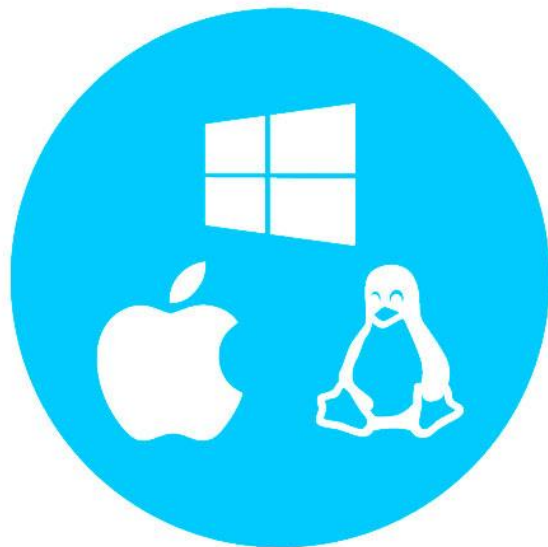
Versión actual  
NET Framework 4.8





# NET 5

Framework Open Source desarrollado por Microsoft para la creación de aplicaciones web y disponible para versiones **Linux**, **Windows** y **Mac**. Integrado con contenedores (Docker).



.NET 5

## Welcome to .NET

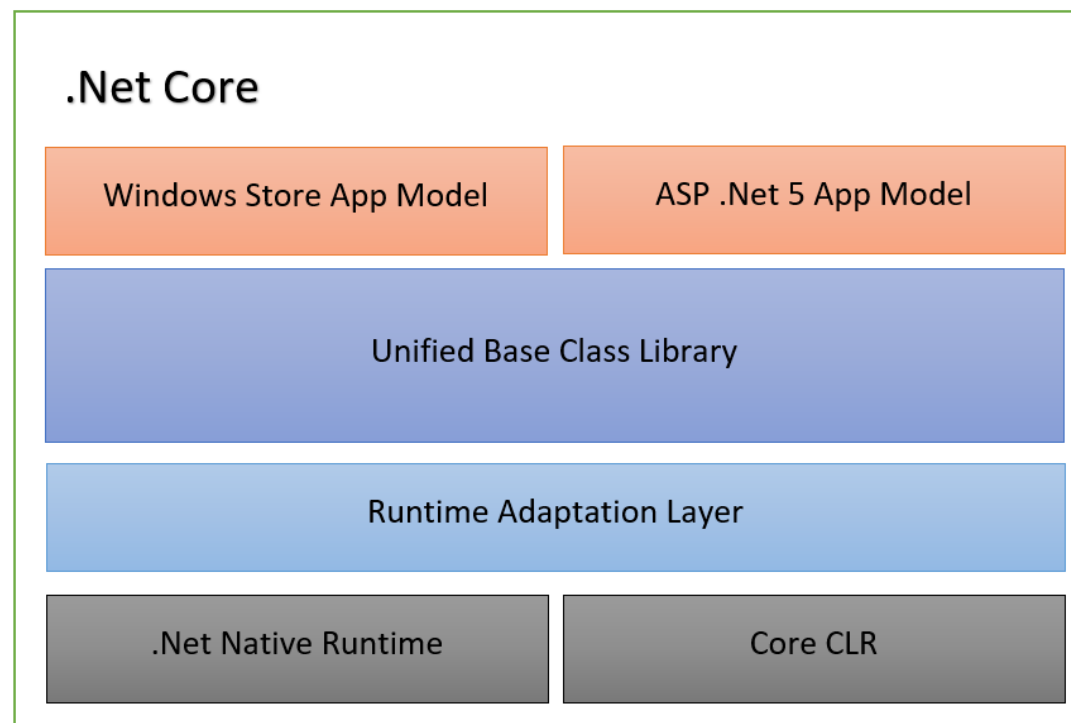
<https://dotnet.microsoft.com/>





## .NET Core

- **Core CLR** es Common Language Runtime optimizado para implementaciones multiplataforma y basadas en la nube. Esto, junto con .Net Native Runtime, constituye la base de todas las plataformas basadas en .Net.
- **.Net Native Runtime** También conocido como Managed Runtime, .Net Native Runtime contiene las bibliotecas basadas en Windows nativas. Esto también contiene la compilación Ahead Of Time (AOT) en lugar de la compilación Just In Time (JIT) de antwihle. Esto mejora el rendimiento de las aplicaciones.





## Ventajas de .NET Framework

- Si es un desarrollador de .NET que necesita compilar y lanzar algo rápido y no tiene tiempo para aprender .NET Core, .NET Framework es su elección. .NET Core tiene una curva de aprendizaje ligeramente superior.
- Si mantiene y actualiza aplicaciones .NET existentes, .NET Framework es su elección. La portabilidad de una aplicación .NET existente a una aplicación de .NET Core requiere cierto trabajo.
- .NET Framework es lo que es. Se supone que la versión actual de .NET Framework, 4.8, es la última versión de .NET Framework. No habrá más nuevas versiones de .NET Framework planeadas en el futuro.



## Ventajas de .NET 5

- Si va a crear una nueva aplicación y tiene la opción entre .NET Core y .NET Framework, .NET 5 es el camino a seguir.
- .NET 5 es compatible con WPF y Windows Forms, también admite el desarrollo cruzado entre UWP, WPF y Windows Forms. Esto proporciona a los desarrolladores la flexibilidad para incorporar interfaces modernas de UWP en Windows Forms y WPF.
- .NET 5 es más adecuado para necesidades multiplataforma. Las aplicaciones de .NET Core son compatibles con Windows, Linux y macOS. El popular editor de código abierto de Microsoft, Visual Studio Code, es compatible con Windows, Linux y macOS. VS Code admite las necesidades modernas de los editores de código, incluidos IntelliSense y la depuración. La mayoría de los editores de terceros, como Sublime, Emacs y VI, trabajan con .NET 5.

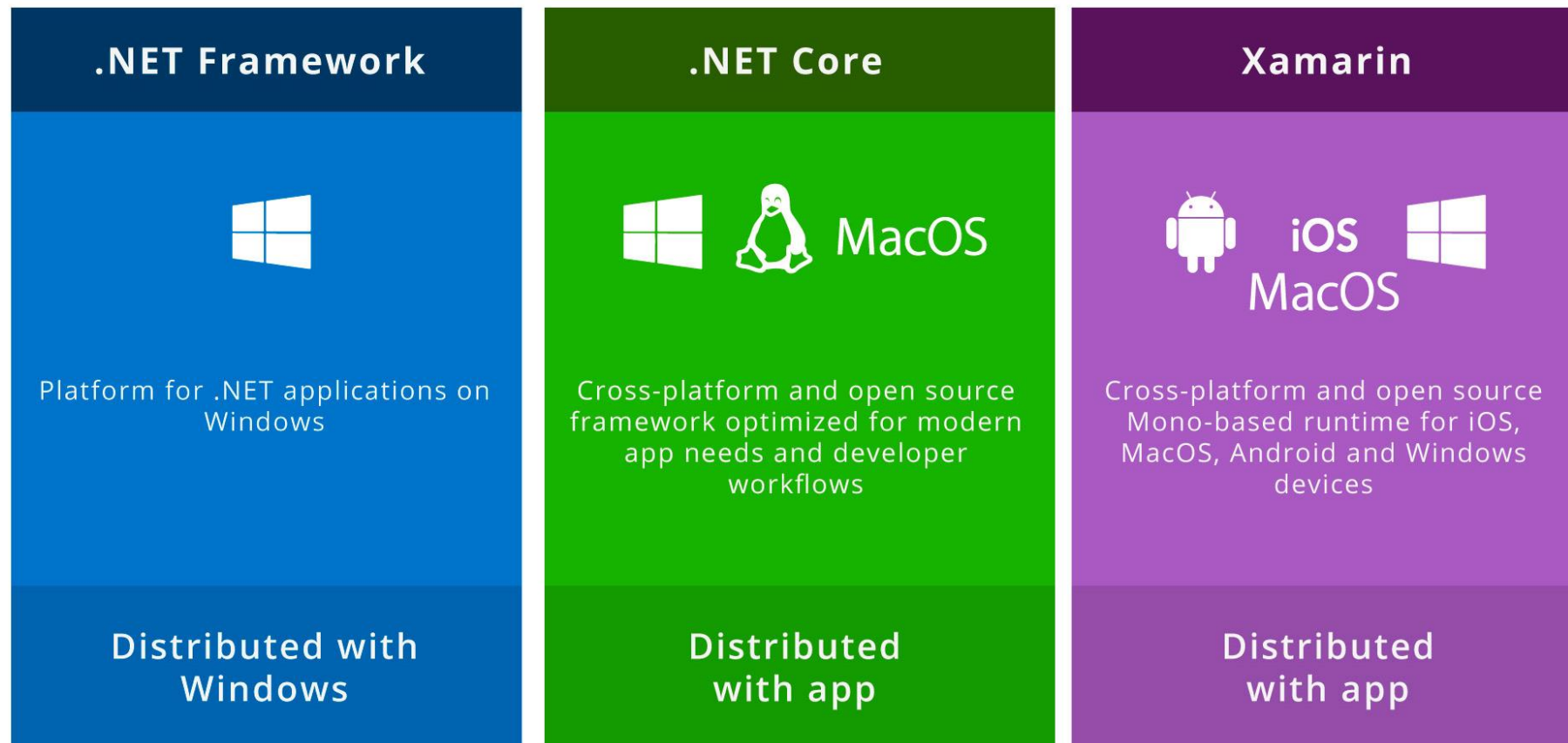


## Ventajas de .NET 5

- La arquitectura de microservicios se admite en .NET 5, que permite que los servicios multiplataforma funcionen con .NET Core, incluidos los servicios desarrollados con .NET Framework, Java, Ruby u otros.
- Los contenedores son las máquinas virtuales de hoy en día. La modularidad, el peso ligero y la flexibilidad de .NET 5 facilitan la implementación de aplicaciones de .NET 5 en contenedores. Los contenedores se pueden implementar en cualquier plataforma, nube, Linux y Windows. .NET 5 funciona bien con Docker y Azure Kubernetes Service.
- El rendimiento y la escalabilidad fueron las dos principales áreas de enfoque clave cuando se desarrolló .NET 5. .NET 5 y ASP.NET Core 5 son los marcos web de mejor rendimiento según algunos puntos de referencia.

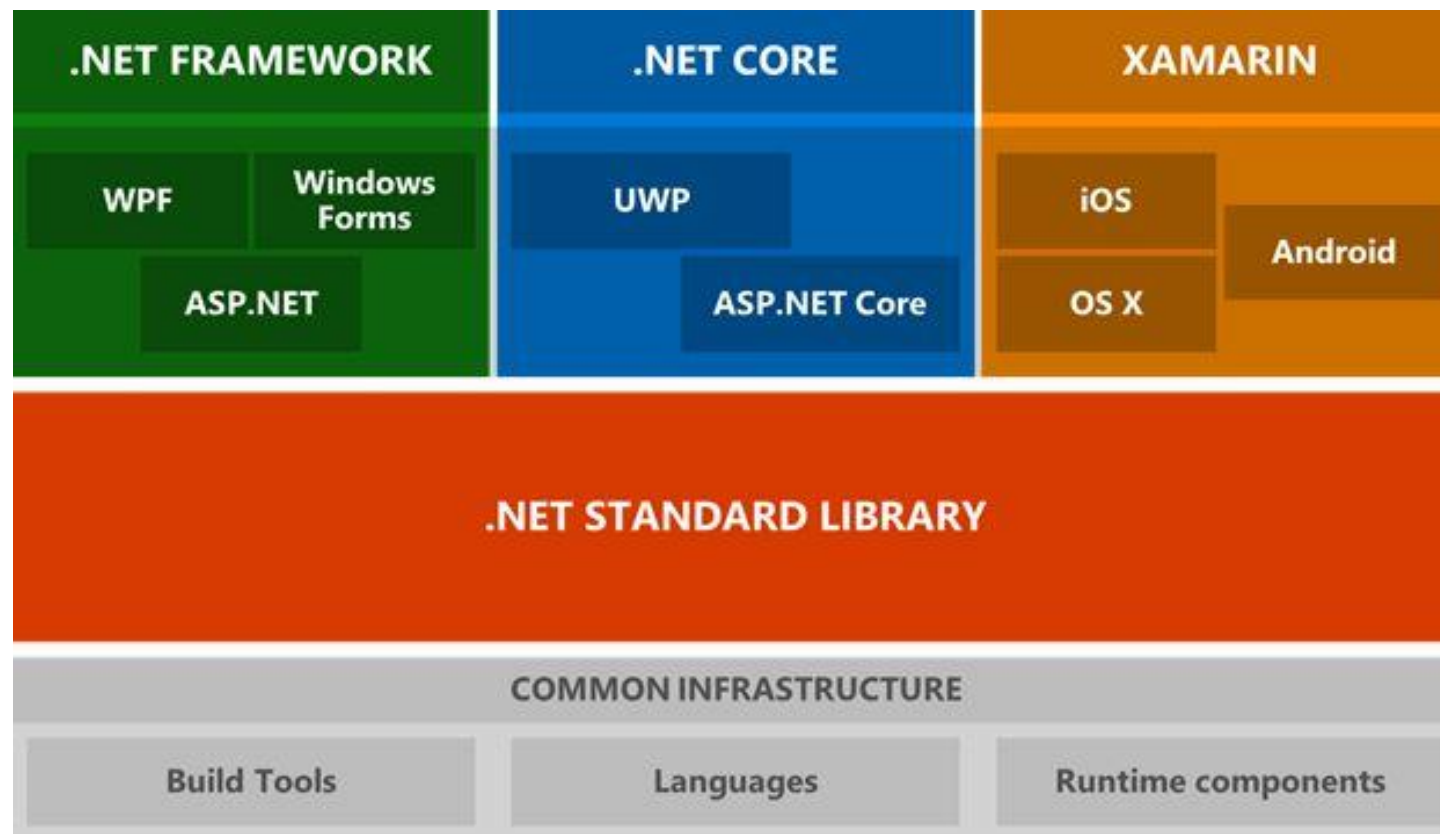


## El Ecosistema NET





# El Ecosistema NET





Openness  
Community  
Rapid innovation

.NET Compiler Platform ("Roslyn") LLILC MVVM Light Toolkit MSBuild  
ASP.NET MVC ASP.NET Core IdentityManager  
.NET SDK for Hadoop MEF Kudu Cecil  
.NET Micro Framework Mono Mailkit ASP.NET AJAX Control Toolkit Xamarin.Auth  
Mimekit Umbraco NuGet Cake Open Live Writer  
WorldWide Telescope ASP.NET Web Pages Microsoft Azure SDK for .NET Couchbase Lite for .NET  
ASP.NET SignalR ASP.NET Web API WCF  
Entity Framework Open XML SDK Xamarin SDK IdentityServer  
Microsoft Azure WebJobs SDK OWIN Authentication Middleware  
Microsoft Web Protection Library ASP.NET Web API Prism System.Drawing  
Orchard CMS ProtoBuild Xamarin.Mobile Salesforce Toolkits for .NET  
Orleans

Features

Protection

Licenses  
Copyrights  
Trademarks  
Patents



Practices

Mentorship  
Governance  
Feedback  
Co-ordination



Visibility

Media  
Branding  
Events



Support

Hosting  
Code signing  
CLA Management  
Swag



Repaso general de ASP.NET Core 5.0



# Entonces, ¿qué elegir?

Use .NET 5 para la aplicación de servidor cuando:

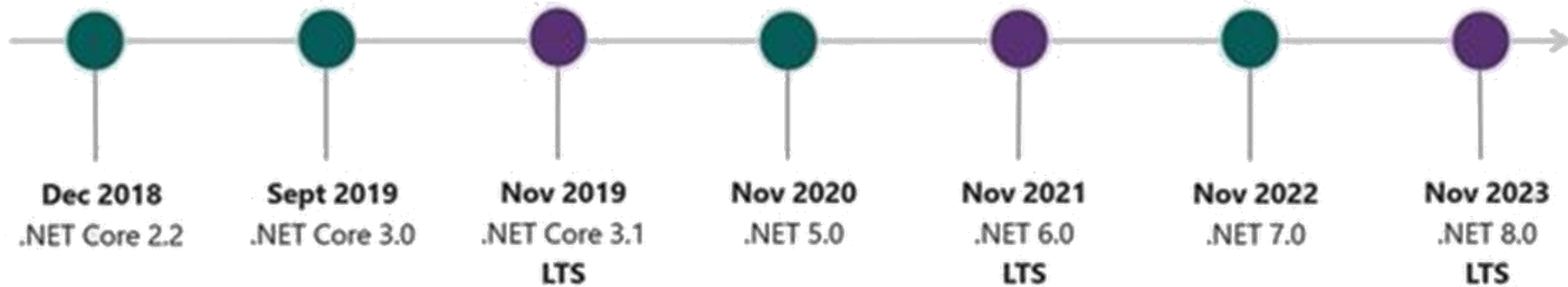
- Tenga necesidades multiplataforma.
- Tenga los microservicios como objetivo.
- Vaya a usar contenedores de Docker.
- Necesite sistemas escalables y de alto rendimiento.
- Necesite versiones de .NET en paralelo por aplicación.

Use .NET Framework para su aplicación de servidor cuando:

- La aplicación usa actualmente .NET Framework (la recomendación es extender en lugar de migrar).
- La aplicación usa bibliotecas .NET de terceros o paquetes NuGet que no están disponibles para .NET 5.
- La aplicación usa tecnologías de .NET que no están disponibles para .NET 5.
- La aplicación usa una plataforma que no es compatible con .NET 5.



## El futuro de .NET



- .NET Schedule
  - Major releases every year, LTS for even numbered releases
  - Predictable schedule, minor releases if needed



# El futuro de .NET

.NET – A unified platform

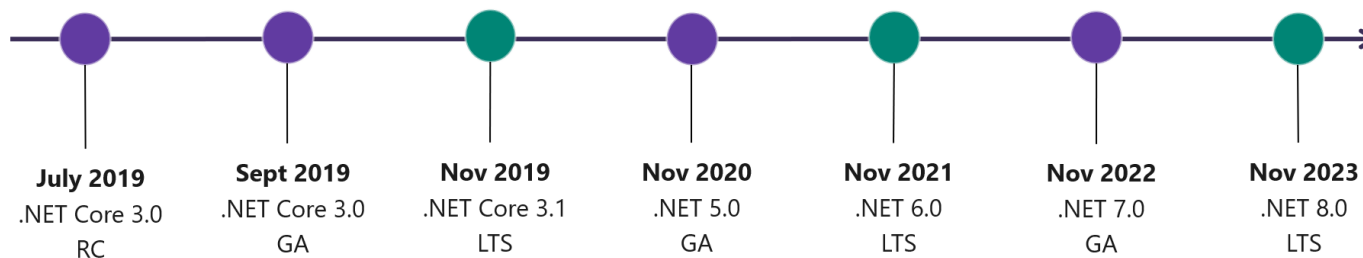


■ Repaso general de ASP.NET Core 5.0



# El futuro de .NET

## .NET Schedule



- .NET Core 3.0 release in September
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

<https://devblogs.microsoft.com/dotnet/introducing-net-5/>

## Principios SOLID

**S**

- Single Responsibility Principle

**O**

- Open/Closed Principle

**L**

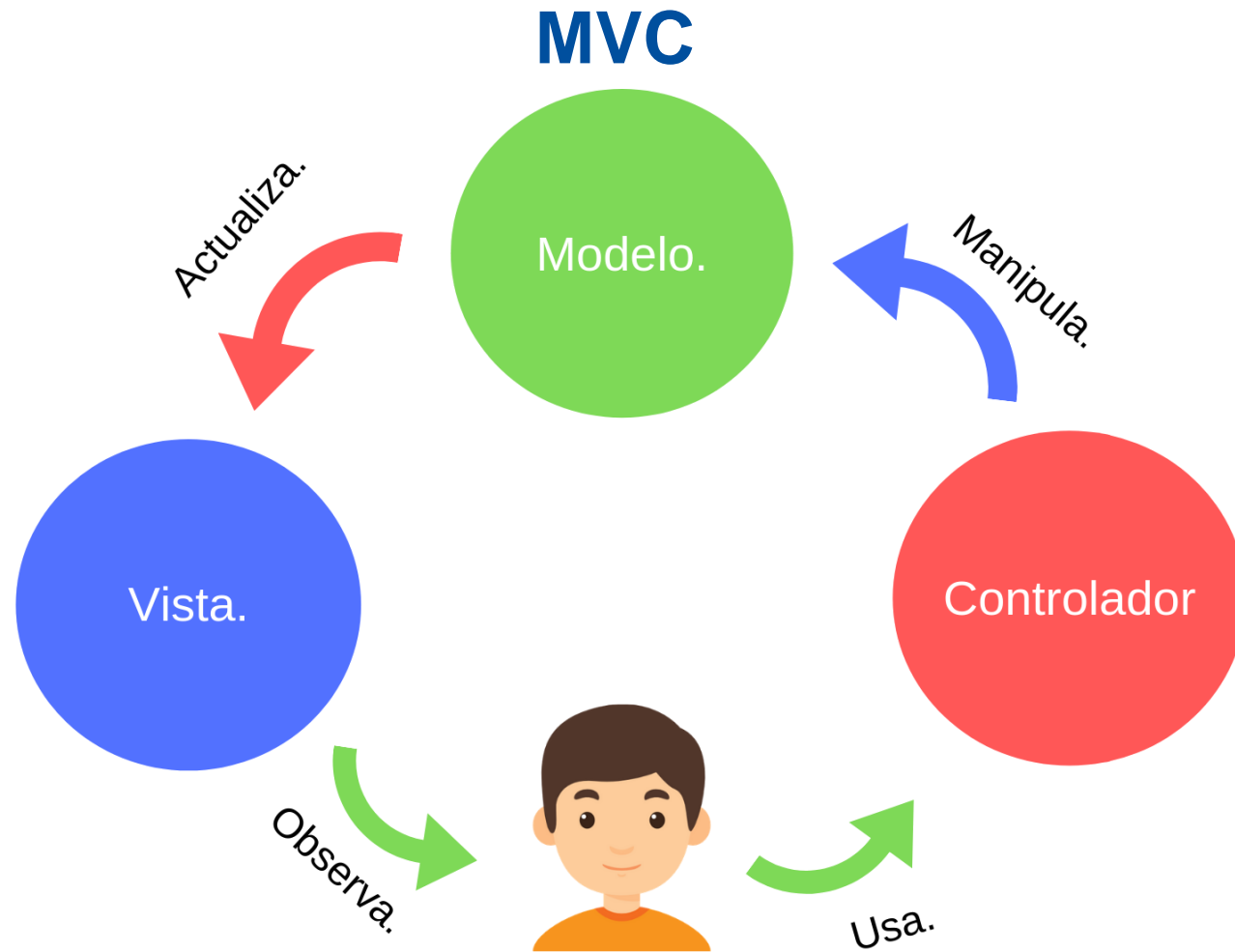
- Liskov Substitution Principle

**I**

- Interface Segregation Principle

**D**

- Dependency Inversion Principle

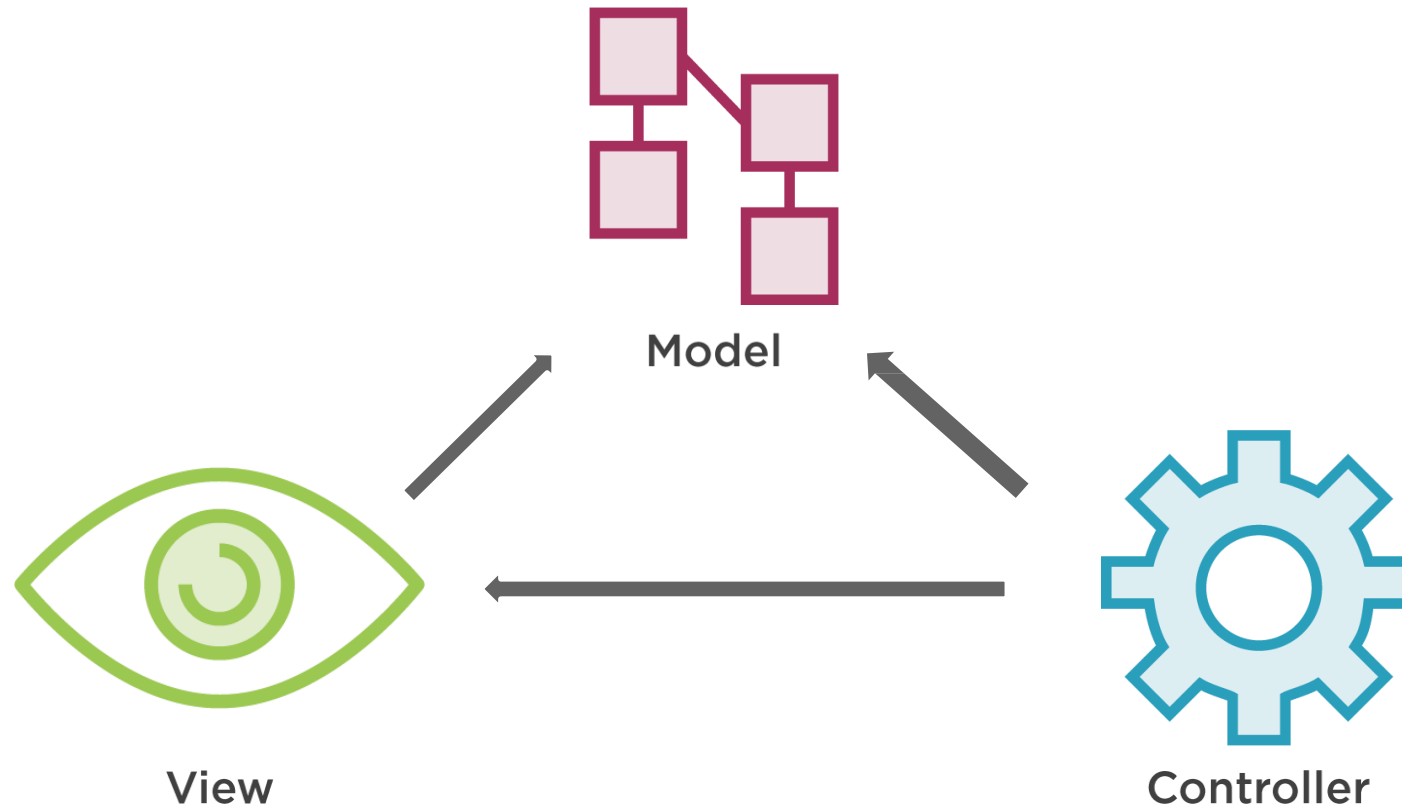


## Patrón MVC

El patrón de arquitectura del controlador de vista de modelos (MVC) separa una aplicación en tres grupos de componentes principales: modelos, vistas y controladores. Este patrón permite lograr la separación de responsabilidades. Con este patrón, las solicitudes del usuario se enrutan a un controlador que se encarga de trabajar con el modelo para realizar las acciones del usuario o recuperar los resultados de consultas. El controlador elige la vista para mostrar al usuario y proporciona cualquier dato de modelo que sea necesario.

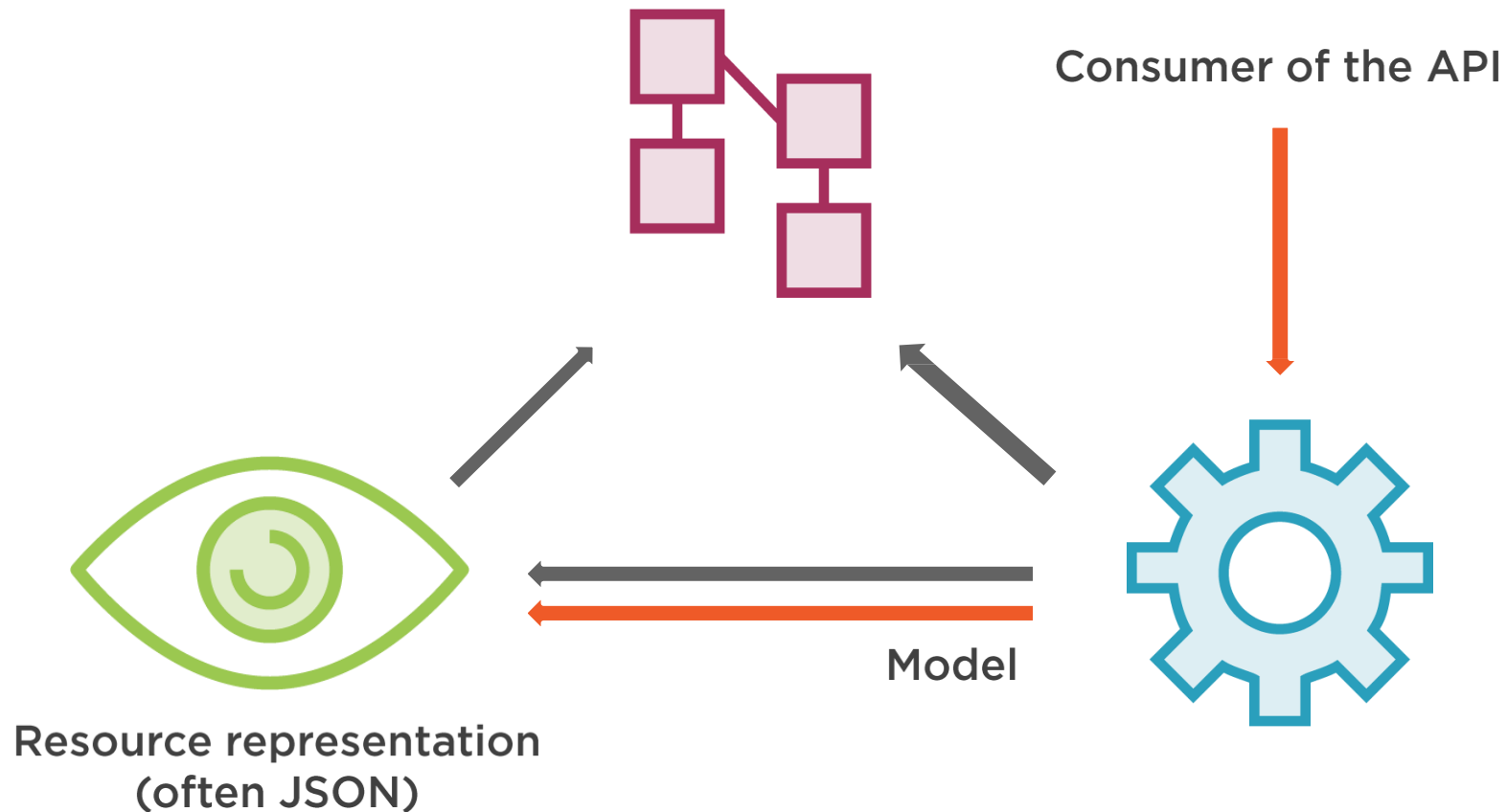
Con esta delineación de responsabilidades es más sencillo escalar la aplicación, porque resulta más fácil codificar, depurar y probar algo (modelo, vista o controlador) que tenga un solo trabajo. Es más difícil actualizar, probar y depurar código que tenga dependencias repartidas entre dos o más de estas tres áreas. Por ejemplo, la lógica de la interfaz de usuario tiende a cambiar con mayor frecuencia que la lógica de negocios. Si el código de presentación y la lógica de negocios se combinan en un solo objeto, un objeto que contenga lógica de negocios deberá modificarse cada vez que cambie la interfaz de usuario. A menudo esto genera errores y es necesario volver a probar la lógica de negocio después de cada cambio mínimo en la interfaz de usuario.

## Patrón Model-View-Controller (API)





## Patron Model-View-Controller (API)







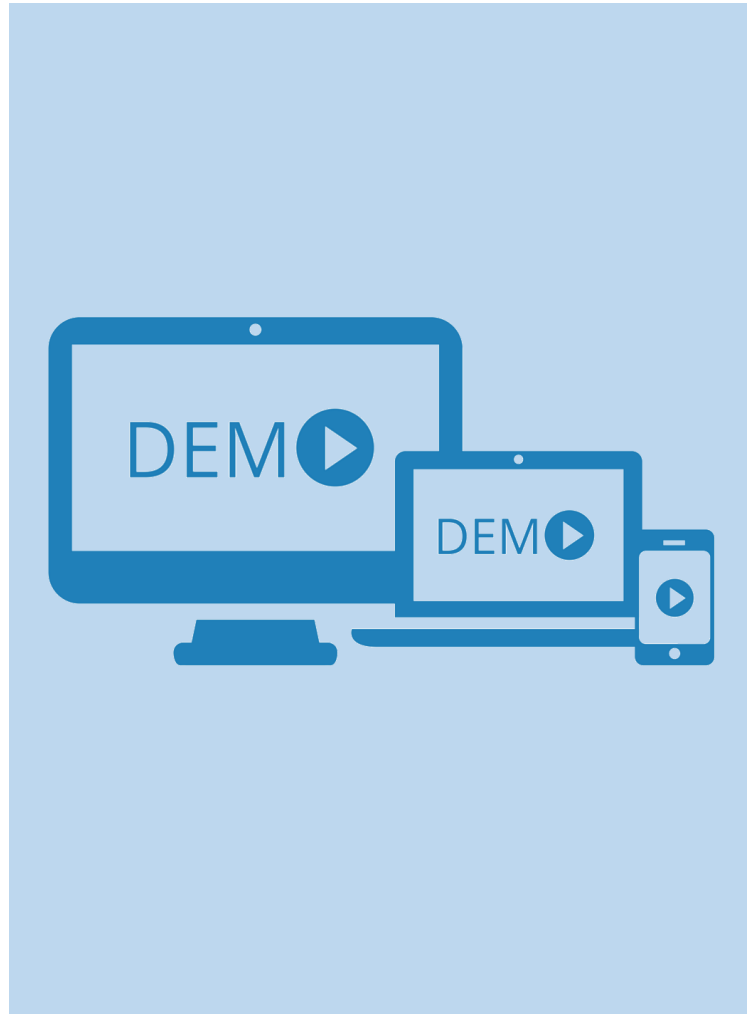
# ASP.NET Core 5 (MVC)

El framework de ASP.NET Core MVC es un framework de presentación ligero, de código abierto y con gran capacidad de prueba, que está optimizado para usarlo con ASP.NET Core.

ASP.NET Core MVC ofrece una manera basada en patrones de crear sitios web dinámicos que permitan una clara separación de intereses. Proporciona control total sobre el mercado, admite el desarrollo controlado por pruebas (TDD) y usa los estándares web más recientes.

- Enrutamiento
- Enlace de modelos
- Validación de modelos
- Inserción de dependencias
- Filtros
- Áreas
- API web
- Capacidad de prueba
- Motor de vistas de Razor
- Vistas fuertemente tipadas
- Asistentes de etiquetas
- Componentes de vista

<https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-5.0>



Visual Studio 2019 (soluciones, proyectos)  
e Internet Information Services ( IIS)  
(despliegue interno y externo).



Es una extensión de Visual Studio que permite gestionar librerías en NET. Sus herramientas cliente permiten producir y consumir paquetes.

<https://www.nuget.org/>





**NPM**

**NPM = Node Package  
Manager**



- Task Runners
- Ejecución en node.js
- Plugins
- Escribir las tareas en javascript
- Integrados con Visual Studio





NET Core CLI, NuGet, Git y GitHub (consola, gestor de paquetes y versiones).



# GRACIAS

POR SU PREFERENCIA