

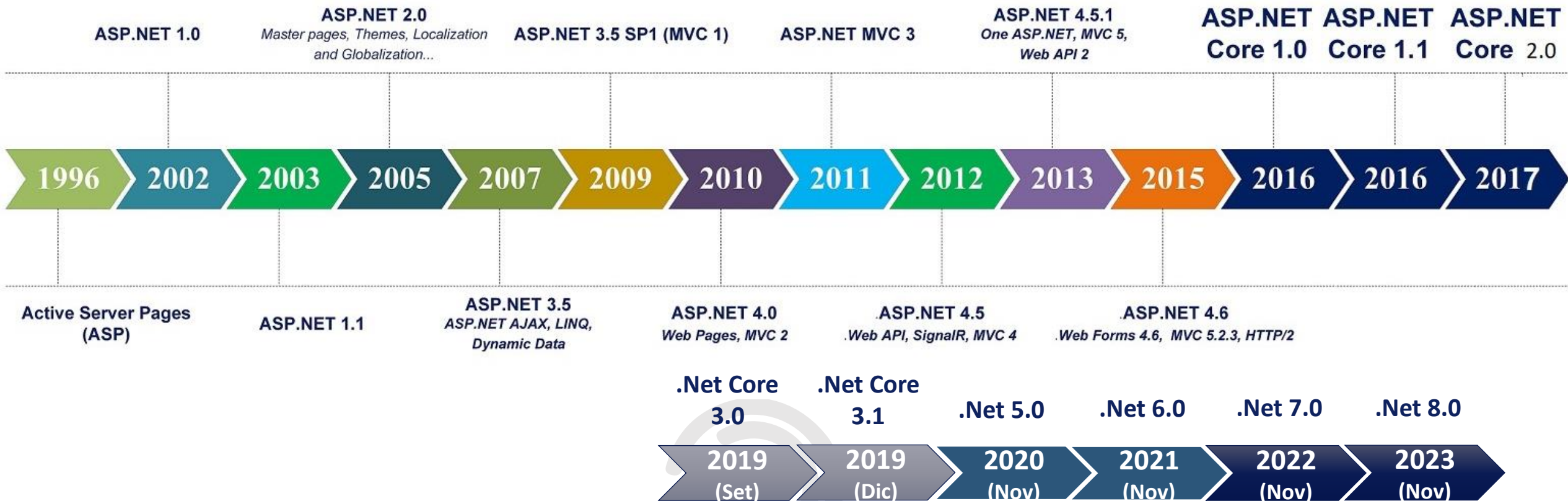


INTRODUCCIÓN A ASP.NET USANDO RAZOR PAGES

INTRODUCCIÓN

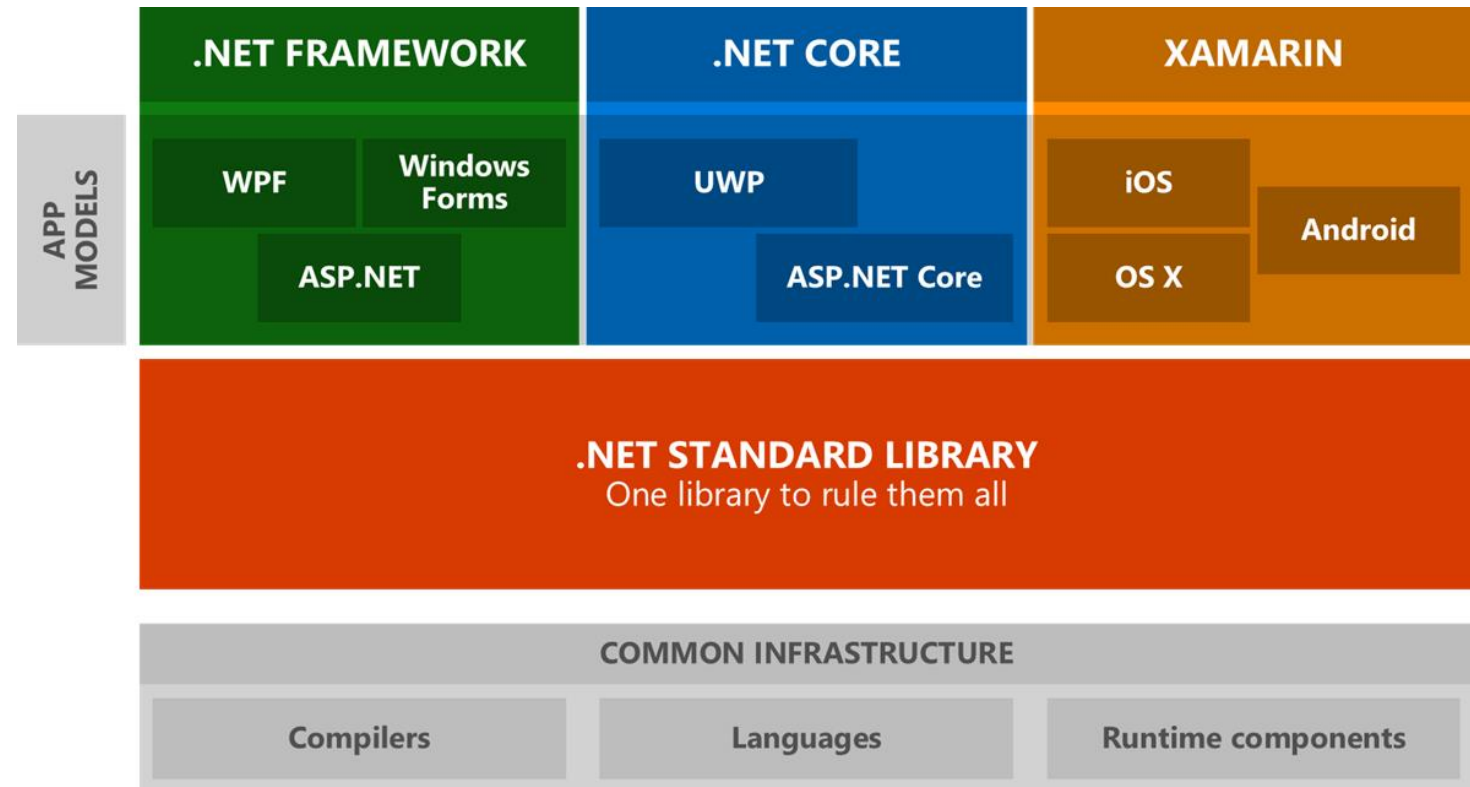


EVOLUCIÓN DE ASP.NET



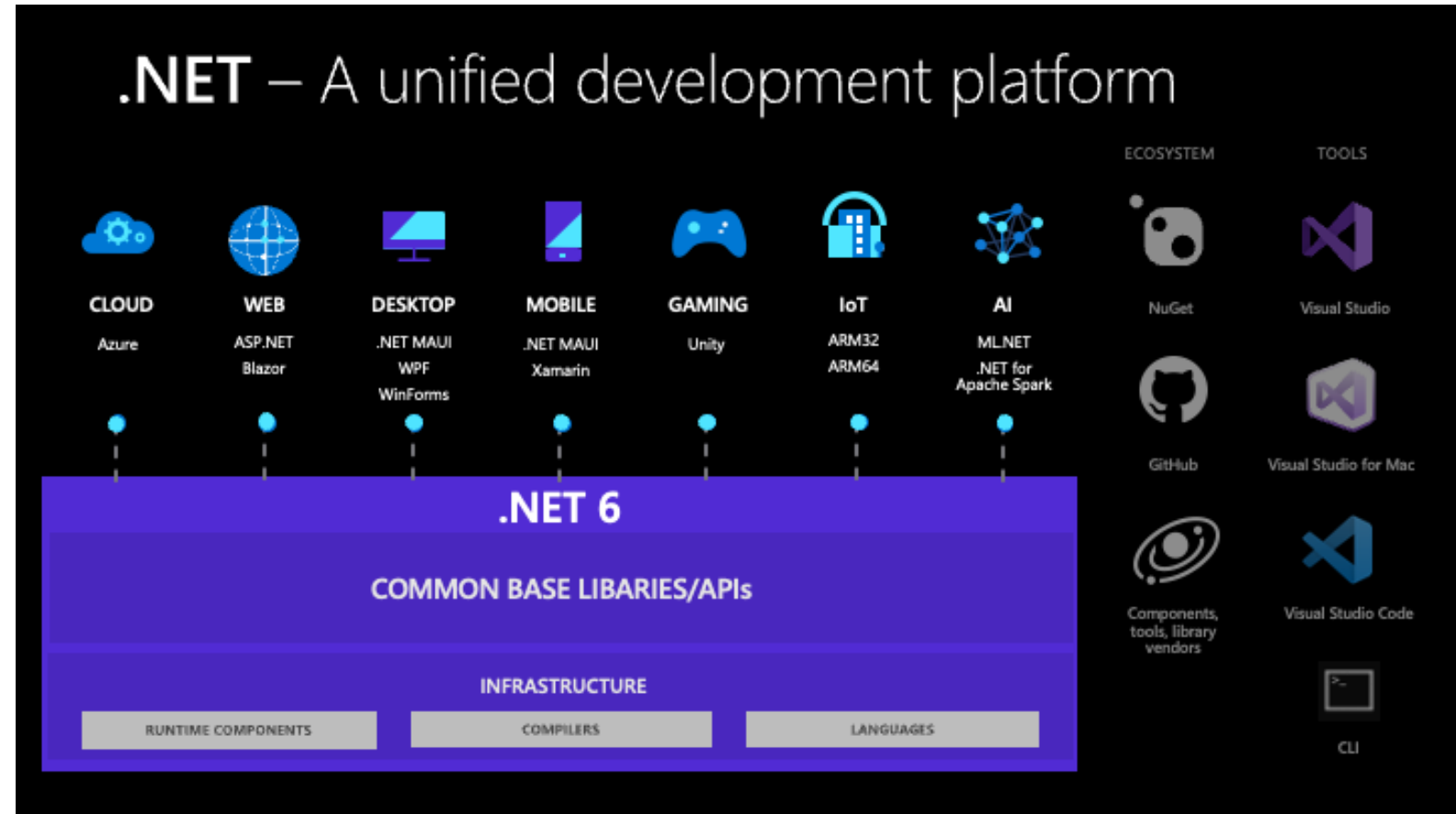
LA PLATAFORMA .NET

- .NET es una **plataforma de desarrollo** compuesta por herramientas, lenguajes de programación y bibliotecas para crear diferentes tipos de aplicaciones.
- La plataforma base proporciona componentes que se aplican a todos los diferentes tipos de aplicaciones. Los marcos adicionales, como ASP.NET, amplían .NET con componentes para crear tipos específicos de aplicaciones.



LA PLATAFORMA .NET

- Los lenguajes de programación C#, F# y Visual Basic.
- Bibliotecas base para trabajar con cadenas, fechas, archivos E/S y más.
- Editores y herramientas para Windows, Linux, macOS y Docker.



ASP.NET

- ASP.NET amplía la plataforma .NET con herramientas y bibliotecas específicamente para **crear aplicaciones web**.
- Estas son algunos elementos que ASP.NET agrega a la plataforma .NET:
 - ✓ **Marco base para procesar solicitudes web en C # o F #**
 - ✓ **Sintaxis de plantillas de páginas web** , conocida como Razor, para crear páginas web dinámicas con C#.
 - ✓ **Bibliotecas para patrones web comunes** , como Model View Controller (MVC)
 - ✓ **Sistema de autenticación** que incluye bibliotecas, una base de datos y páginas de plantilla para manejar los inicios de sesión, incluida la autenticación multifactor y la autenticación externa con Google, Twitter y más.
 - ✓ **Extensiones de editor** para proporcionar resaltado de sintaxis, finalización de código y otras funciones específicas para desarrollar páginas web.



ASP.NET Core

- ASP.NET Core es un marco multiplataforma de código abierto y de alto rendimiento que tiene como finalidad compilar aplicaciones modernas conectadas a Internet y habilitadas para la nube.
- Con ASP.NET Core se puede hacer lo siguiente:
 - ✓ Compilar servicios y aplicaciones web, aplicaciones de Internet de las cosas (IoT) y back-ends móviles.
 - ✓ Usar sus herramientas de desarrollo favoritas en Windows, macOS y Linux.
 - ✓ Efectuar implementaciones locales y en la nube.
 - ✓ Ejecutar en .Net Core.



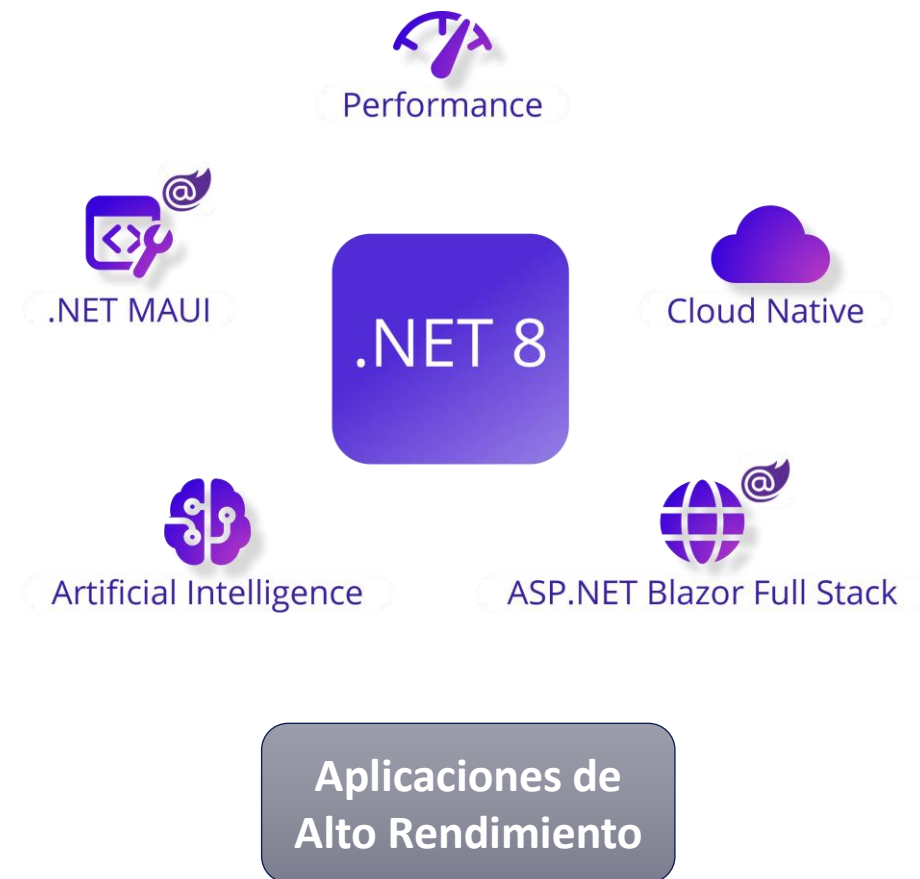
CARACTERÍSTICAS DE .NET Core

- Tiene una arquitectura modular, que permite que podamos incluir únicamente lo necesario que requieren nuestras aplicaciones, haciendo que tengan un peso menor a la hora de ser desplegadas y que sea más sencilla su actualización a través de updates de **NuGet**.
- Es multiplataforma y viene con soporte para su uso con contenedores **Docker**.
- Alto rendimiento. Se ha desarrollado desde cero y se le ha dado una alta importancia a esta característica.
- Asincronía con el uso de async/await. Se ha implementado este patrón en todas las librerías comunes para mejorar el rendimiento en las llamadas I/O.
- Es Open Source.



.NET 8

- Marco que pone un fuerte énfasis en garantizar que las aplicaciones sean rápidas, eficientes y se ejecuten sin problemas en una amplia gama de plataformas.
- Entre sus principales características:
 - **Reducción de los tiempos de inicio:** los usuarios finales experimentarán un inicio de aplicación más rápido, mejorando la experiencia general del usuario.
 - **Mejora de la paralelización:** permite a los desarrolladores escribir código más eficiente y escalable para aprovechar al máximo las arquitecturas multinúcleo modernas.
 - **Optimización de la Gestión de Recursos:** permite que las aplicaciones de .NET 8 sean más eficientes en el uso de la memoria y los recursos del sistema.
 - **Perfilado avanzado:** Las herramientas de perfilado de .NET 8 ofrecen una vista detallada del rendimiento de la aplicación, lo que permite a los desarrolladores identificar y resolver fácilmente problemas de rendimiento.



Programación con ASP.NET usando sintaxis Razor

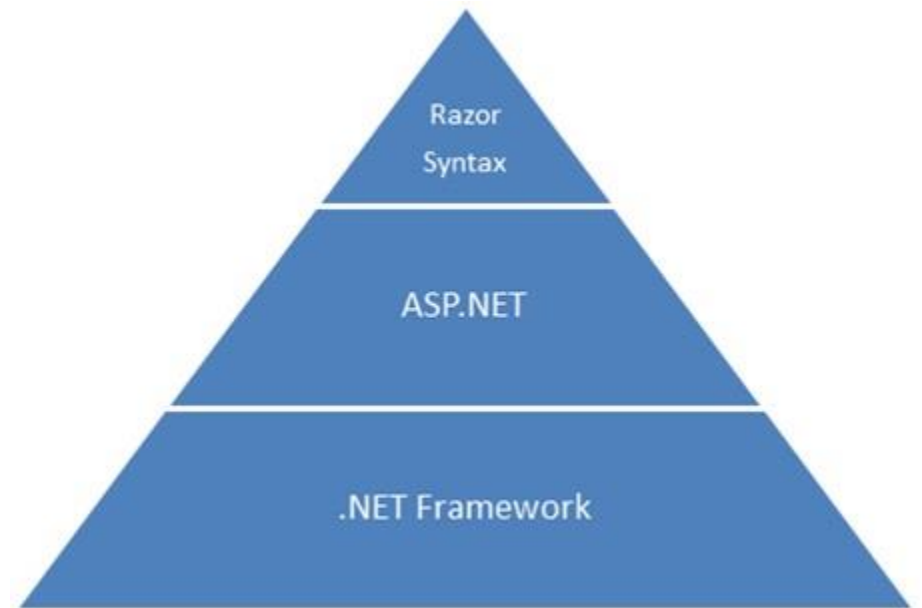
- Razor pages es la propuesta de ASP.NET Core MVC para facilitar la creación de contenido web de una forma más cercana al mundo transversal Back, permitiendo programar **parte de la lógica** en **C#** y el aspecto de **vistas** con **html, css, js** y notaciones propias.
- Razor Pages facilita la programación de escenarios centrados en páginas y hace que resulte más productiva y sencilla que con controladores y vistas.
- Utiliza rutas, modelos, viewmodels, actionresult, tag helpers, etc.
- Soporta el desarrollo multiplataforma: Windows, Linux, Mac.
- Razor pages es una abstracción de MVC y de Webforms e implementa el patrón de diseño MVVM.

Programación con ASP.NET usando sintaxis Razor

- La sintaxis de Razor es una sintaxis de programación simple para incrustar en una página web, código basado en servidor.
- En una página web que utiliza la sintaxis de Razor, hay dos tipos de contenido:
 1. **Contenido del cliente:** marcado HTML (elementos), información de estilo como CSS, script de cliente como JavaScript y texto sin formato.
 2. **Código del servidor:** la sintaxis de Razor permite agregar código de servidor al contenido de del cliente, ejecutándose primero el código de servidor en la página, antes de enviar la página al navegador.
- Lo más importante es que el código del servidor puede crear dinámicamente contenido de cliente, es decir puede generar HTML u otro contenido durante la marcha y luego enviarlo al navegador junto con algún HTML estático que pueda contener la página)

Programación con ASP.NET usando sintaxis Razor

- Las páginas web ASP.NET que incluyen la sintaxis de Razor tienen una extensión de archivo especial (**.cshtml** o **.vbhtml**).
- La sintaxis de Razor brinda todas las ventajas de ASP.NET, fácil de aprender y la vez permite crear aplicaciones más productivas
- Permite crear sitios web sofisticados, ya que puede usar marcos de trabajo disponibles en ASP.NET y .NET Framework.



Creación de una Aplicación web ASP.Net con Razor Pages

Directorio, que contiene los archivos razor.

Archivos razor compartidos por la aplicación.

Archivos razor con un archivo de C#.

Archivos razor especiales no navegables.

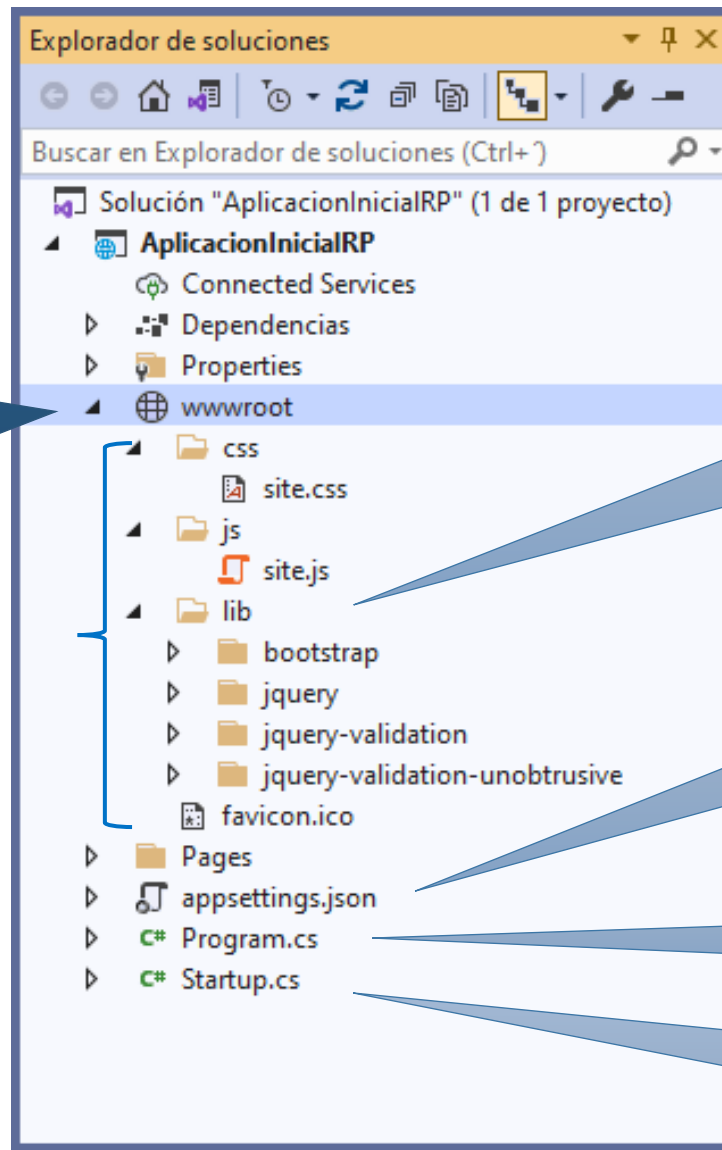
The screenshot shows the 'Explorador de soluciones' (Solution Explorer) window for a project named 'Solución "AplicacionInicialRP" (1 de 1 proyecto)'. The 'AplicacionInicialRP' folder is expanded, showing subfolders: 'Connected Services', 'Dependencias', 'Properties', 'wwwroot', and 'Pages'. The 'Pages' folder is selected and expanded, showing a 'Shared' subfolder containing '_Layout.cshtml' and '_ValidationScriptsPartial.cshtml'. Below 'Shared' are '_ViewImports.cshtml' and '_ViewStart.cshtml'. Further down are 'Error.cshtml', 'Index.cshtml', and 'Privacy.cshtml', each with a corresponding C# file (e.g., 'Error.cshtml.cs', 'Index.cshtml.cs', 'Privacy.cshtml.cs'). At the bottom are 'appsettings.json', 'Program.cs', and 'Startup.cs'. Blue callout boxes with brackets point to specific parts of the structure: one to the 'Pages' folder, one to the 'Shared' folder, one to the C# files associated with the Razor pages, and one to the '_ViewImports.cshtml' and '_ViewStart.cshtml' files.

Aplicación web ASP.Net con Razor Pages

Directorio, que contiene los archivos estáticos de la aplicación web. Contiene:

- Hojas de estilo.
- Archivos JavaScript.
- Imágenes
- Otros recursos estáticos de la aplicación.

Estos archivos son reconocidos en **Startup.cs** con la línea de código: **app.UseStaticFiles()**: lo que permite usar archivos estáticos en la aplicación web.



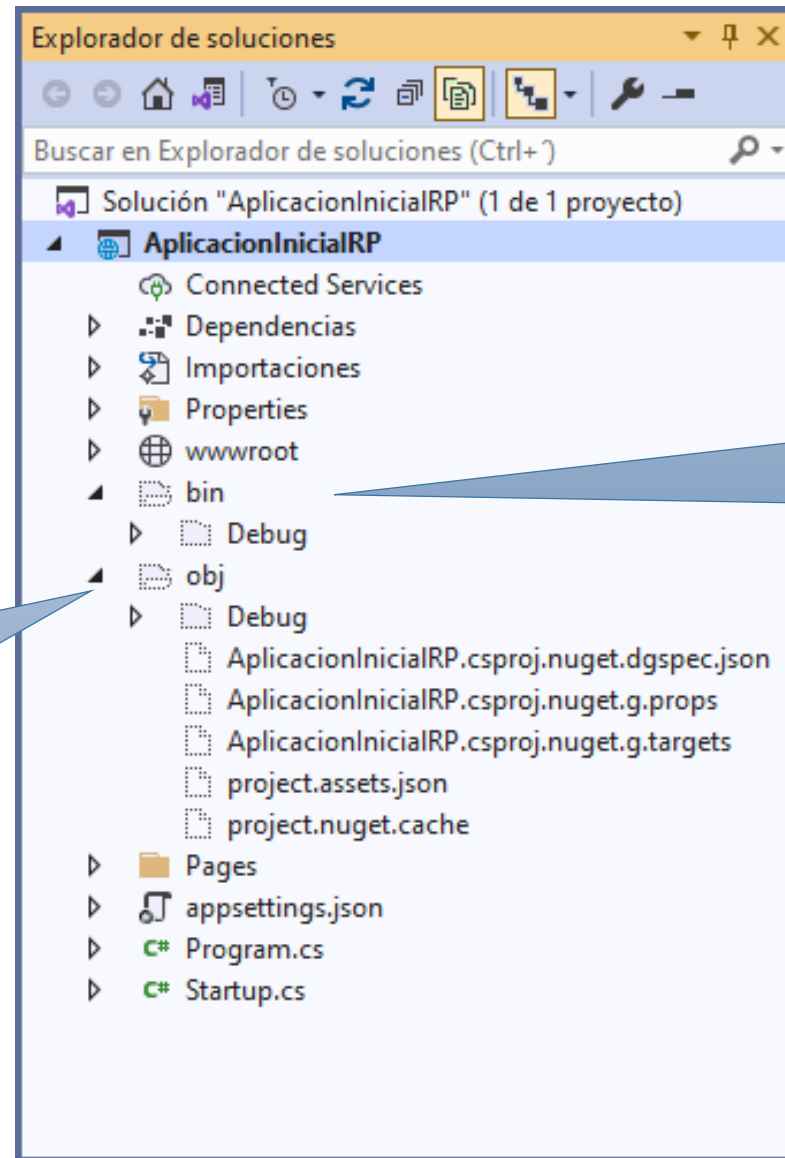
Contiene librerías de terceros: bootstrap, jquery, jquery-validation.

Para los ajustes de la configuración global de la aplicación.

Punto de entrada para la aplicación de Razor Pages.

Donde se configura la canalización de solicitudes hacia la aplicación.

Aplicación web ASP.Net con Razor Pages



Para almacenar archivos de objetos temporales y otros archivos empleados para crear el binario final durante el proceso de compilación.

Ubicación de salida predeterminada para los archivos binarios generados cuando la aplicación es compilada. Debug y Release.

REFERENCIAS

- Manual de ASP.NET Web Pages Using the Razor Syntax.
- <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>
- <https://docs.microsoft.com/es-es/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0>
- <https://openwebinars.net/blog/que-es-net-core>