



UNIVERSIDAD DE COLIMA

Facultad de telemática.

Carrera: Ingeniería de software.

Materia: Programación web.

Profesor: Pedro Damián Reyes.

ACTIVIDAD 7

Alumna: Castrejón Mejía Evelyn Fernanda.

Semestre y grupo: 4°F

.NET Core es la plataforma de desarrollo de Microsoft más moderna, de código fuente abierto, multiplataforma y de alto rendimiento para la creación de todo tipo de aplicaciones.

.NET Core, a diferencia del .NET Framework, no tiene este tipo de dependencia del sistema Windows y es modular, usando el sistema de paquetes NuGet, gracias al cual, recibiremos las diversas actualizaciones de .NET Core, a diferencia de .NET Framework que se actualiza a través de Windows Update.

Esta arquitectura modular 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.

Al ser multiplataforma, no depende de cosas específicas del sistema operativo, como sucede con el .NET Framework.

Más allá de la edición de código, el IDE de Visual Studio reúne diseñadores gráficos, compiladores, herramientas de finalización de código, control de código fuente, extensiones y muchas más características en un solo lugar.

- Ventajas

- Multiplataforma: .NET Core es multiplataforma, lo que significa que puede ejecutarse en diferentes sistemas operativos, como Windows, Linux y macOS.
- Código abierto: .NET Core es de código abierto, lo que permite a los desarrolladores acceder al código fuente y contribuir a la plataforma.
- Alto rendimiento: .NET Core es conocido por su alto rendimiento. El marco de trabajo se ha optimizado para lograr un alto rendimiento incluso en cargas de trabajo intensivas.
- Seguridad: .NET Core tiene características de seguridad incorporadas, como la validación de entrada, la mitigación de ataques de inyección de SQL y la protección contra ataques CSRF.
- Escalabilidad: .NET Core es escalable, lo que significa que puede manejar cargas de trabajo crecientes sin sacrificar el rendimiento.

- Desventajas

- Curva de aprendizaje: Los desarrolladores que no tienen experiencia previa en .NET pueden enfrentar una curva de aprendizaje pronunciada para aprender a trabajar con la plataforma.
- Biblioteca de terceros: Aunque la biblioteca de .NET Core es extensa, todavía no es tan grande como algunas otras bibliotecas de terceros, lo que puede limitar las opciones de los desarrolladores.

- **Tamaño del archivo:** Las aplicaciones .NET Core pueden requerir un tamaño de archivo más grande en comparación con otras tecnologías, lo que puede afectar la velocidad de descarga y la velocidad de inicio.
- **Problemas de compatibilidad:** La compatibilidad puede ser un problema al cambiar de versiones anteriores de .NET a .NET Core. Las bibliotecas y herramientas de terceros pueden no ser compatibles, lo que puede generar problemas al actualizar a .NET Core.

Blazor: Es un marco de interfaz de usuario web del lado cliente de naturaleza similar a plataformas de front-end de JavaScript como Angular o React. Blazor controla las interacciones del usuario y representa las actualizaciones necesarias de la interfaz de usuario. Blazor no se basa en un modelo de solicitud-respuesta. Las interacciones del usuario se controlan como eventos que no están en el contexto de una solicitud HTTP concreta.

Las aplicaciones Blazor constan de uno o más componentes raíz que se representan en una página HTML.

- Crea interfaces de usuario completamente interactivas con C# en lugar de JavaScript.
- Comparta la lógica de aplicación del lado cliente y servidor escrita con .NET.
- Represente la interfaz de usuario como HTML y CSS para la compatibilidad con todos los exploradores, incluidos los móviles.
- Realice la integración con plataformas de hospedaje modernas, como Docker.
- Cree aplicaciones híbridas de escritorio y móviles con .NET y Blazor.

El uso de .NET para el desarrollo web en el lado cliente ofrece las siguientes ventajas:

- Escribe el código en C# en lugar de JavaScript.
- Aprovechamiento del ecosistema .NET existente de bibliotecas .NET.
- Uso compartido de la lógica de aplicación en el servidor y el cliente.
- Beneficios de rendimiento, confiabilidad y seguridad de .NET.
- Mantenga la productividad en Windows, Linux o macOS con un entorno de desarrollo, como Visual Studio o Visual Studio Code.
- Compile sobre un conjunto común de lenguajes, marcos y herramientas que son estables, completos y fáciles de usar.

Razor y c#

Razor es una sintaxis de marcado para insertar código basado en .NET en páginas web. La Razor sintaxis consta de Razor marcado, C# y HTML. Los archivos que contienen Razor generalmente tienen una .cshtml extensión de archivo.

Razor admite C# y usa el símbolo para realizar la @ transición de HTML a C#. Razor evalúa las expresiones de C# y las representa en la salida HTML.

Cuando un @ símbolo va seguido de una Razor palabra clave reservada, pasa al Razormarcado específico. De lo contrario, pasa a HTML sin formato.

C# (pronunciado "si sharp" en inglés) es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET.

Los programas de C# se ejecutan en .NET, un sistema de ejecución virtual denominado Common Language Runtime (CLR) y un conjunto de bibliotecas de clases. CLR es la implementación de Microsoft del estándar internacional Common Language Infrastructure (CLI). CLI es la base para crear entornos de ejecución y desarrollo en los que los lenguajes y las bibliotecas funcionan juntos sin problemas.

El código fuente escrito en C# se compila en un lenguaje intermedio (IL) que guarda conformidad con la especificación de CLI. El código y los recursos de IL, como los mapas de bits y las cadenas, se almacenan en un ensamblado, normalmente con una extensión .dll. Un ensamblado contiene un manifiesto que proporciona información sobre los tipos, la versión y la referencia cultural.

Opinión personal a manera de conclusión: En resumen, .NET Core es importante porque permite a los desarrolladores construir aplicaciones multiplataforma de alto rendimiento y escalables con una amplia variedad de bibliotecas y herramientas disponibles. Además, como es de código abierto y tiene compatibilidad con .NET Framework es atractivo para desarrolladores y empresas. Es una útil para el desarrollo de aplicaciones modernas y escalables en múltiples plataformas. Con su capacidad para funcionar en Windows, Linux y macOS, .NET Core es una plataforma de desarrollo de software versátil que puede ayudar a los desarrolladores a crear aplicaciones de alta calidad y seguras en cualquier plataforma.

Bibliografía

Alcolea, C. D. (02 de 03 de 2023). *openwebinars*. Obtenido de openwebinars:
<https://openwebinars.net/blog/que-es-net-core/>

learn.microsoft. (02 de 03 de 2023). *learn.microsoft*. Obtenido de learn.microsoft:
<https://learn.microsoft.com/es-es/dotnet/architecture/blazor-for-web-forms-developers/architecture-comparison>

learn.microsoft. (02 de 03 de 2023). *learn.microsoft*. Obtenido de learn.microsoft:
<https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>

learn.microsoft. (02 de 03 de 2023). *learn.microsoft*. Obtenido de learn.microsoft.:
<https://learn.microsoft.com/es-es/aspnet/core/blazor/?view=aspnetcore-7.0>

learn.microsoft. (02 de 03 de 2023). *learn.microsoft*. Obtenido de learn.microsoft.:
<https://learn.microsoft.com/es-es/aspnet/core/mvc/views/razor?view=aspnetcore-7.0>