



UNIVERSIDAD DE COLIMA

Universidad de Colima
Facultad de Telemática
Programación para Web

Académico:
Damián Reyes Pedro

Alumno
Valdez Gutierrez Aldo Eduardo

**Act 7. Investigación Teórica .Net Core (Visual Studio
2022, Blazor, Razor y C#)**

Colima, Col; a 12 de febrero de 2024

Importancia:

La importancia de la arquitectura de desarrollo .Net Core radica en su capacidad para ofrecer un entorno de desarrollo integrado (IDE) como Visual Studio, que simplifica y acelera el proceso de desarrollo de software. Además, tiene una gran importancia al integrar o poder integrarse cosas como Blazor en conjunto con C# esta integración es de gran importancia ya que Blazor ayuda uniendo el desarrollo web con C# en lugar del comúnmente usado JavaScript ayudando esto debido a que el lenguaje C# tiene un enfoque en la legibilidad y la facilidad de uso, lo que lo hace un lenguaje elegante y fácil de aprender. Su sintaxis limpia y su estructura orientada a objetos facilitan la escritura y comprensión del código.



Ventajas

Entre las ventajas que nos encontramos están las siguientes:

Compatibilidad: la arquitectura de desarrollo .Net Core es compatible con múltiples plataformas como Windows, Linux, macOS

Desarrollo Web Moderno: la arquitectura Blazor como se dijo anteriormente integra C# al desarrollo web lo que nos evita usar JavaScript

Reducción de la complejidad: como ya no se usan más lenguajes de programación se reduce la complejidad de necesitar lenguajes adicionales o frameworks

Unificación de Bibliotecas: .Net Core, Blazor y C# son totalmente compatibles entre si junto con sus bibliotecas siendo compatible Blazor con casi todas las infraestructuras y bibliotecas existentes de .Net

Amplio Soporte: Al ser todas estas tecnologías integradas por Microsoft o propietarias de esta cuenta con una gran cantidad de soporte así mismo como una gran cantidad de desarrolladores teniendo una gran cantidad de documentación, código y herramientas.

Seguridad mejorada: la arquitectura de .Net Core ya cuenta con características de seguridad integradas que sin mucho problema podemos integrarlas en nuestro software

Desventajas

Curva de aprendizaje: para gente que no está adaptada al desarrollo en .Net Core o en Visual Studio junto con Blazor puede encontrar problemas para adaptarse a usar estas tecnologías debido a que estos son del ecosistema de Microsoft

Dependencia de Microsoft: al ser todo esto integrado por Microsoft puede existir la dependencia de que si Microsoft llega a aumentar sus costos o algo similar al ser dependiente de este puede causarnos problemas

Dependencia a JavaScript: Aunque Blazor elimina en su mayoría la necesidad de escribir código JavaScript aun así sigue necesitando de este en varios casos

Tiempos de carga: Al usar .Net Core junto con Blazor tenemos unos tiempos de carga mas altos debido a que ya que necesitamos ejecutar .Net en el navegador a diferencia de una combinación JavaScript con HTML

Costos de licenciamiento: Las tecnologías básicas en si son gratuitas sin embargo existen costos para utilidades mas avanzadas como por ejemplo Visual Studio en su versión Profesional y Enterprise

Conclusión:

Con esta investigación puedo decir que la integración de estas tecnologías es una idea muy interesante debido a que al ser todo integrado correctamente por Microsoft es algo muy confiable y seguro, así como sus numerosas ventajas, como su amplia compatibilidad, soporte, reducción de la complejidad lo hacen muy buena opción pero a su vez al ser todo de Microsoft también es una desventaja ya que nos hace dependiendo de esta empresa sin embargo es en general un buen entorno de desarrollo todo junto esto ya que en ciertos casos nos puede ayudar a facilitar ciertas cosas

