

Blazor, la nueva apuesta de Microsoft



Autor: Kaking Choi.

Ingeniero de sistemas y computación, con especialidad en Ingeniería de software, maestría en Ingeniería de sistema mención gerencia, estudiante de doctorado en gestión de proyectos, Certificado en desarrollador 5 estrella platinum.

Empecemos por conocer qué es un Web Assembly. Es un formato de código binario portable, para la ejecución íntegra en navegador de scripts del lado del cliente. Se trata de un lenguaje de bajo nivel. Lo interesante es que un desarrollador puede escribir código en un mismo lenguaje, tanto del lado del cliente como del lado del servidor. Por ejemplo, un programador experto en C# puede programar en ambos ambientes sin tener que combinar lenguajes.

Esto se logra de la siguiente manera: las páginas en la primera petición descargan todas las librerías, estas establecen un túnel de comunicación con el servidor y cada petición se hace a través de socket.

Blazor es la implementación de Microsoft.NET para generar ensamblados, podríamos decir que es todo un Framework para construir páginas web bajo el concepto de SPA utilizando C# como lenguaje base, Razor page y HTML que se ejecutan en el navegador. Esto habilita una nueva generación de programadores Full Stack bajo una sola plataforma estable y de alta productividad como lo es punto NET.

Para empezar a desarrollar en esta novedosa plataforma es necesario tener un entorno de desarrollo con la última versión de .NET Framework Core 3.0 y las plantillas de Blazor, las cuales existen dos: 1) Blazor Server App y 2) Blazor WebAssembly, donde esta última es la que funciona totalmente en el navegador.

Blazor provee todos los beneficios de un framework de UI en el lado de cliente (y opcionalmente en el servidor):

- A. Modelo de componentes UI
- B. Routing
- C. Layout
- D. Formularios y validación
- E. Inyección de dependencias
- F. Interoperabilidad con javascript

- G. Renderizado de UI en servidor
- H. Live-reloading durante el desarrollo
- I. Junto a los beneficios que otorgan otros frameworks, encontramos otros propios de Blazor, como son:
 - a. Uso de librerías en .NET Estándar 2.0 o superior
 - b. Reducción del tamaño del payload para disminuir los tiempos de descarga
 - c. Cacheado de .NET Runtime y ensamblados
 - d. Funcionamiento sin necesidad de ejecutar .NET en el servidor
- J. Por último, se pueden destacar los siguientes beneficios de utilizar .NET en un navegador:
 - a. Estabilidad y consistencia
 - b. Uso de lenguajes modernos e innovadores
 - c. Desarrollo con herramientas líderes, como la familia de productos Visual Studio
 - d. Velocidad y escalabilidad
 - e. Aprovechamiento de skills existentes para un desarrollo full-stack
 - f. Amplia compatibilidad con navegadores (es compatible con los navegadores líderes de escritorio y móvil)

Para finalizar, podemos observar las infinitas posibilidades que nos trae este tipo de desarrollo de software teniendo en cuenta que aún mantiene de forma nativa la interoperabilidad con JavaScripts. De tener éxito y buena acogida, esto podría marcar el futuro del desarrollo WEB y crear una revolución en los paradigmas actuales de programación.

Nos preguntamos, ¿cuál será el impacto de la implementación de esta tecnología para los desarrolladores? ¿Es una oportunidad para aquellos desarrolladores que dominaron el desarrollo WinForm?

Les dejo estas reflexiones.

Ing. Kaking Choi. M.A, P.I.S