



Introducción a Blazor

Héctor Uriel Pérez Rojas



Comunidad



<https://www.facebook.com/groups/blazoresespanol>



<https://bit.ly/2BISs1u>



Curso – Introducción a Blazor



<https://www.udemy.com/course/curso-blazor-gratis/>



<https://bit.ly/2W4RUtl>





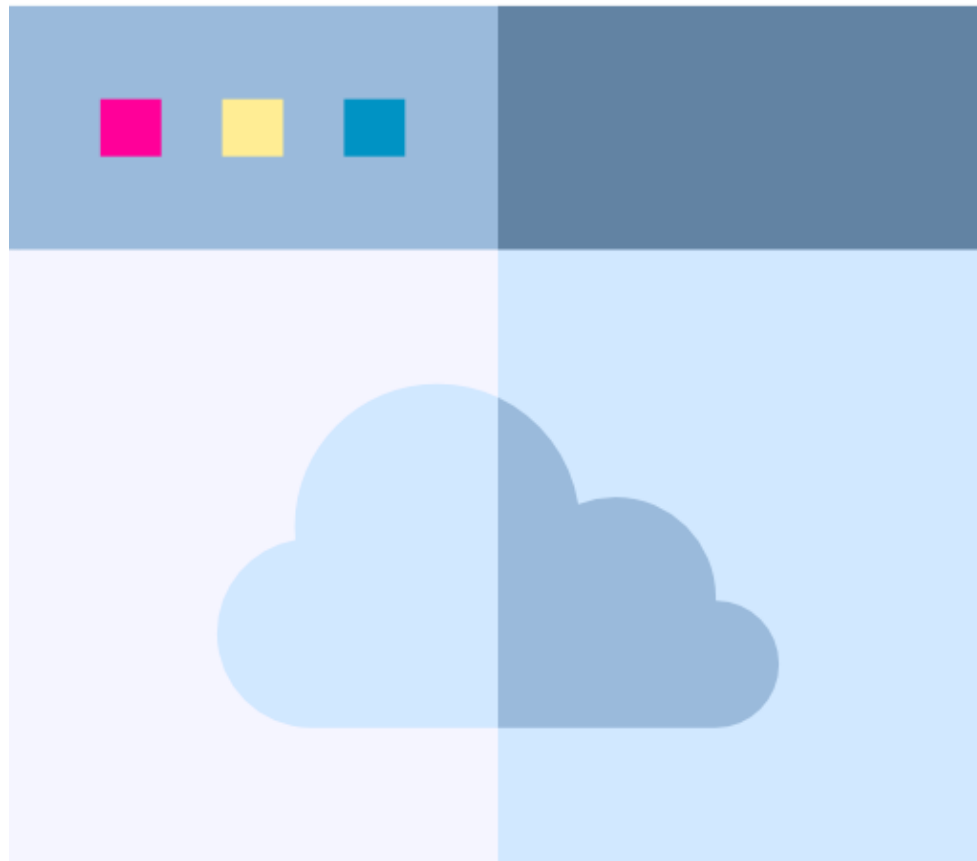
Las aplicaciones web hoy en día



Java, .NET, Node.js

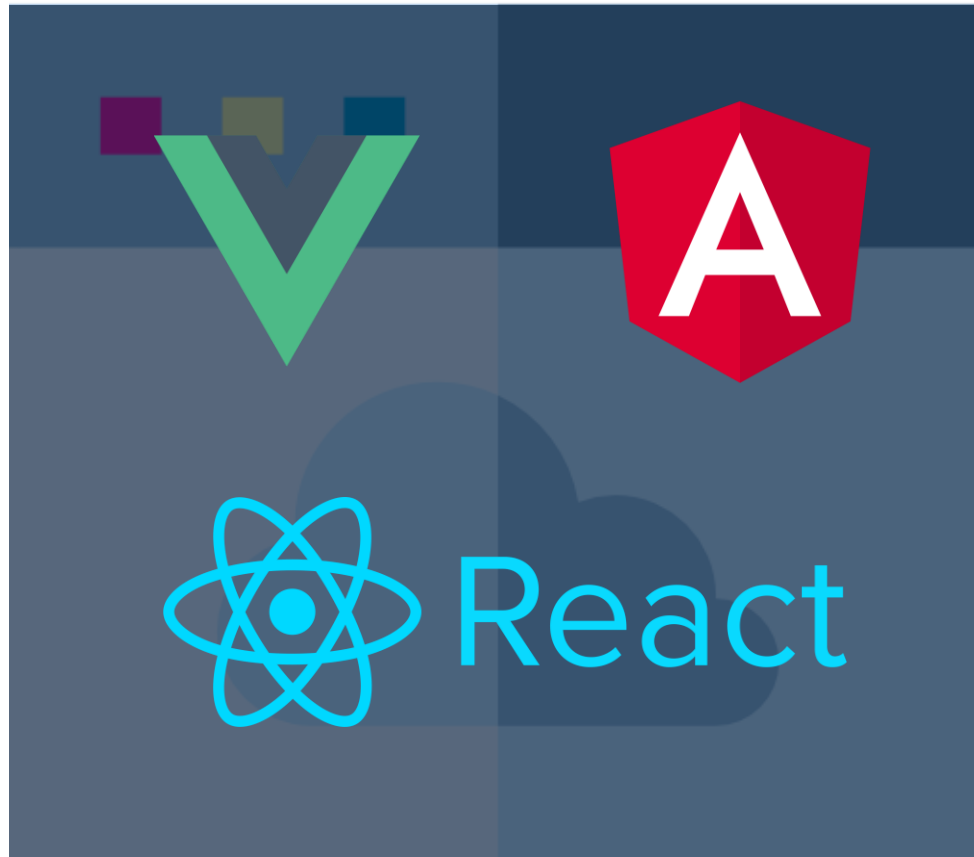


Las aplicaciones web hoy en día



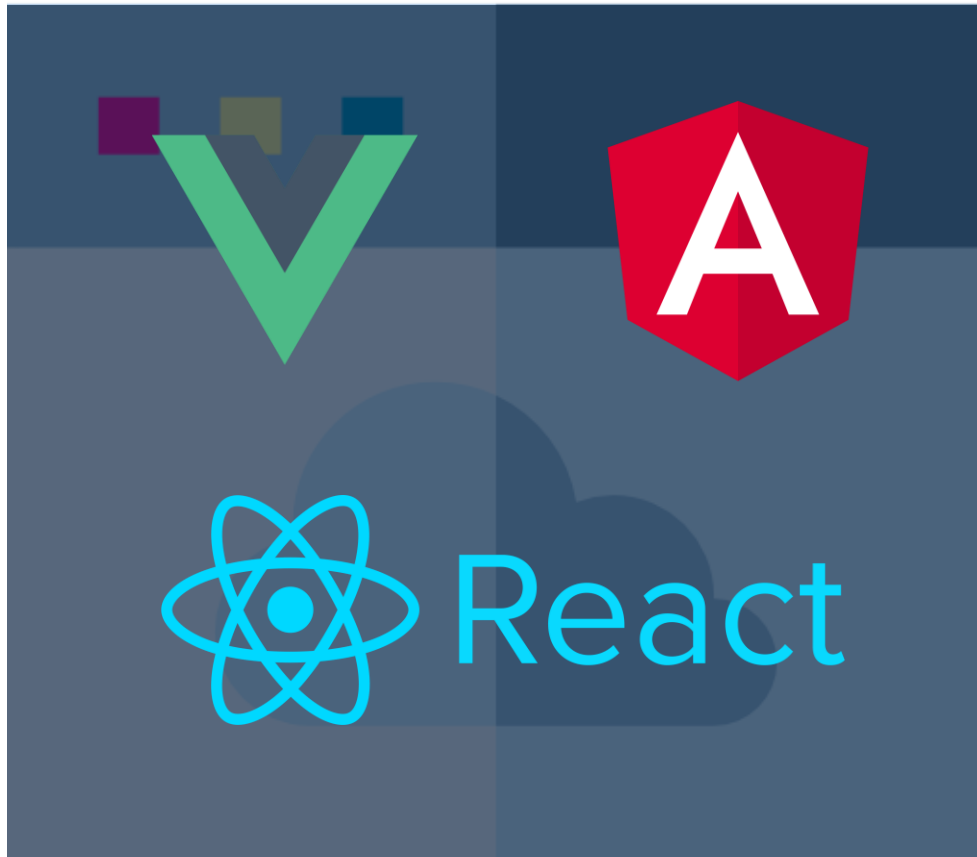


Las aplicaciones web hoy en día



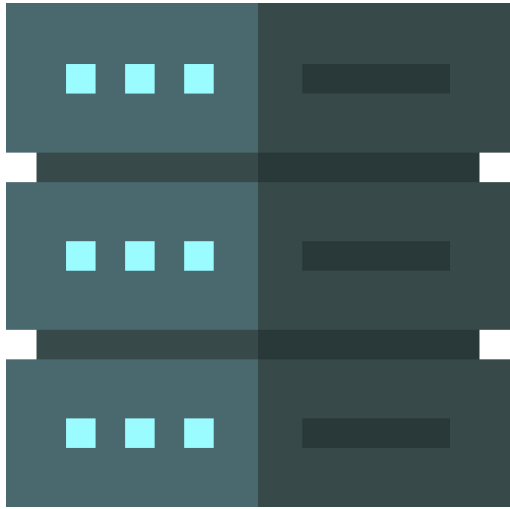


Las aplicaciones web hoy en día

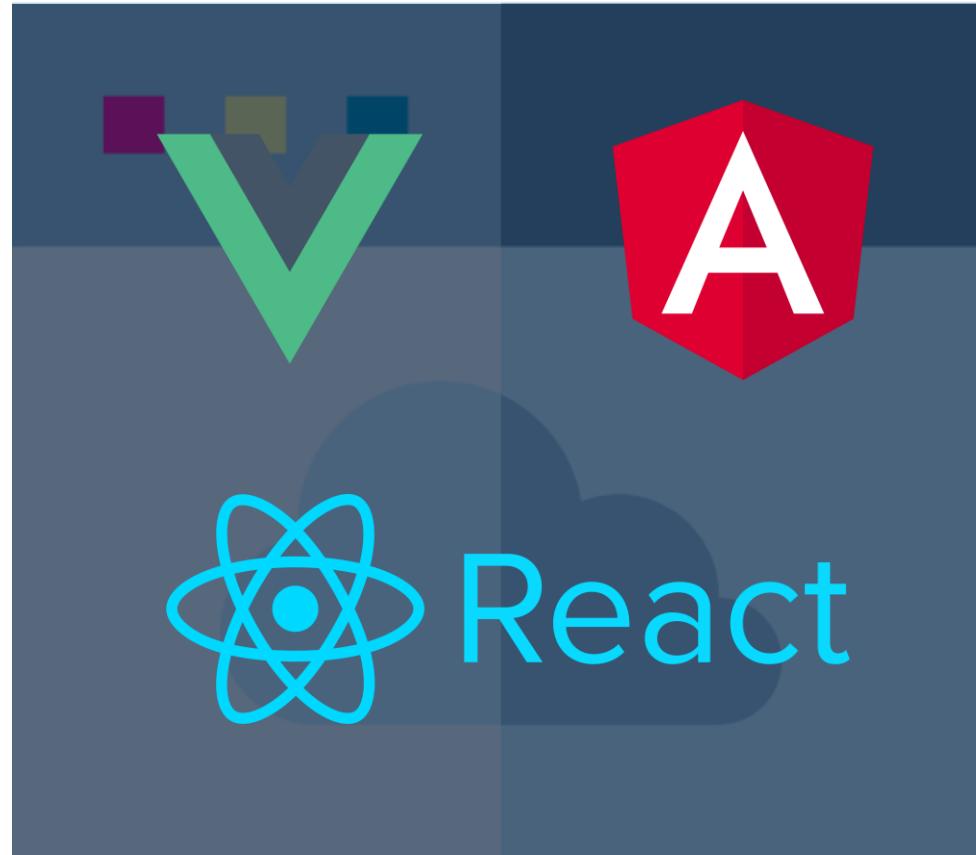




Las aplicaciones web hoy en día



Java, .NET, Node.js



@ WebAssembly (WASM)

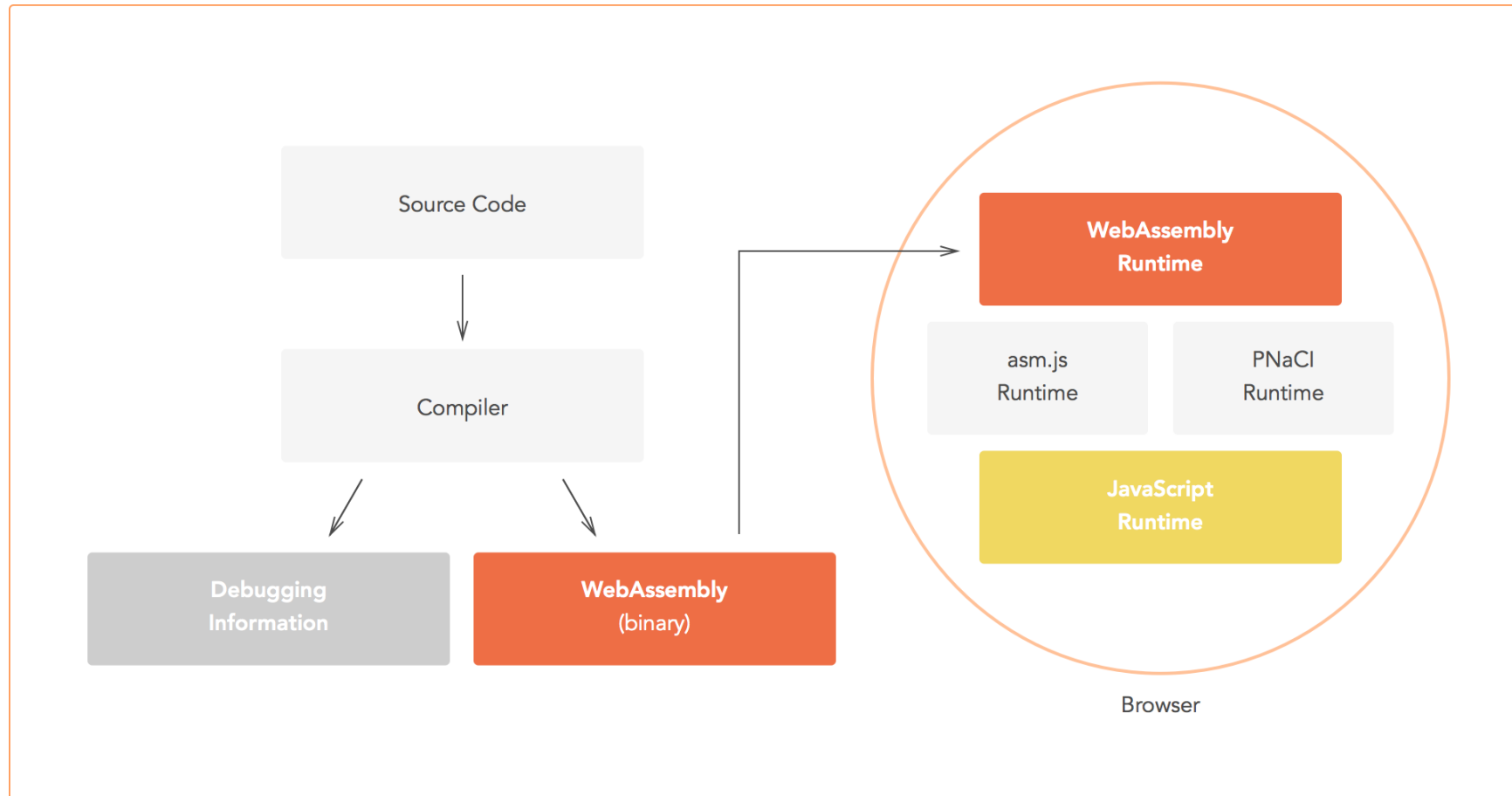




WebAssembly (WASM)

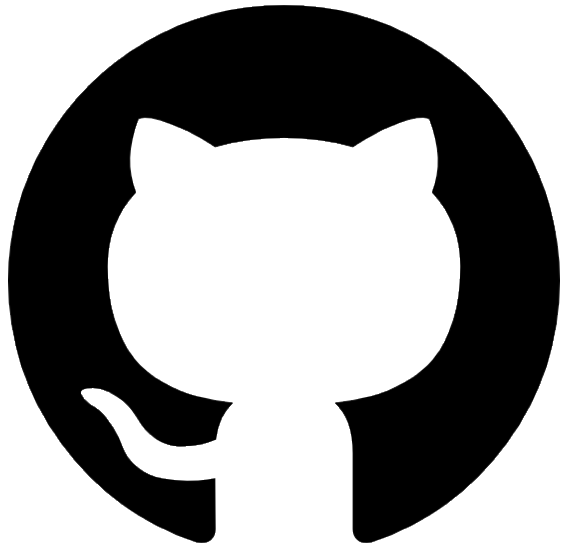
```
1 00000000: 0061 736d                ; WASM_BINARY_MAGIC
2 00000004: 0100 0000                ; WASM_BINARY_VERSION
3 ; section "Type" (1)
4 00000008: 01                        ; section code
5 00000009: 00                        ; section size (guess)
6 0000000a: 02                        ; num types
7 ; type 0
8 0000000b: 60                        ; func
9 0000000c: 00                        ; num params
10 0000000d: 00                        ; num results
11 ; type 1
12 0000000e: 60                        ; func
13 0000000f: 00                        ; num params
14 00000010: 01                        ; num results
15 00000011: 7f                        ; i32
16 00000009: 08                        ; FIXUP section size
17 ; section "Function" (3)
18 00000012: 03                        ; section code
19 00000013: 00                        ; section size (guess)
20 00000014: 02                        ; num functions
21 00000015: 00                        ; function 0 signature index
22 00000016: 01                        ; function 1 signature index
23 00000013: 03                        ; FIXUP section size
24 ; section "Table" (4)
25 00000017: 04                        ; section code
26 00000018: 00                        ; section size (guess)
27 00000019: 01                        ; num tables
28 ; table 0
```

WebAssembly (WASM)



Demo - WebAssembly

Blazor WebAssembly



.NET
Core3.0



WebAssembly (WASM)



@ WebAssembly (WASM)



WebAssembly (WASM)





¿Qué es Blazor?

Blazor es una plataforma de trabajo (framework) para la creación de interfaces de usuario web interactivas del lado cliente con .NET



¿Qué es Blazor?

- Framework de desarrollo para Single Page Applications
- Te permite crear interfaces de usuario web interactivas utilizando C# en lugar de Javascript.
- Las aplicaciones Blazor, se componen de componentes web reutilizables utilizando C#, HTML y CSS.
- El código del cliente y servidor son escritos en C#, lo que permite compartir código y bibliotecas.
- Es una característica de ASP.NET.

¿Dónde se puede usar Blazor?

Podemos utilizar Blazor del lado del cliente en cualquier navegador que cuente con soporte **WebAssembly**.



¿Dónde se puede usar Blazor?

WebAssembly - OTHER

Usage	% of all users
Global	91.39%

WebAssembly or "wasm" is a new portable, size- and load-time-efficient format suitable for compilation to the web.

[illegible]



¿Porqué utilizar Blazor?

Soportado por la mayoría de navegadores modernos, incluidos dispositivos móviles

Se utiliza código C#, en lugar de Javascript

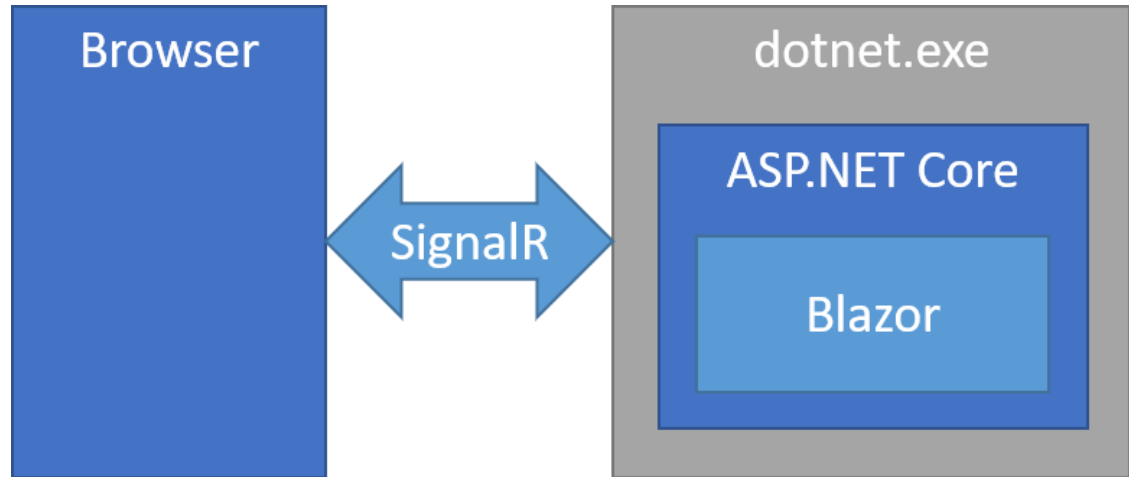
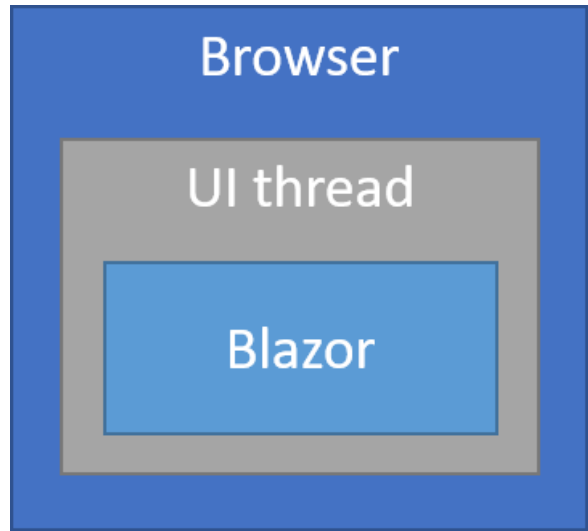
Se pueden reutilizar bibliotecas escritas en .NET

Casi rendimiento nativo

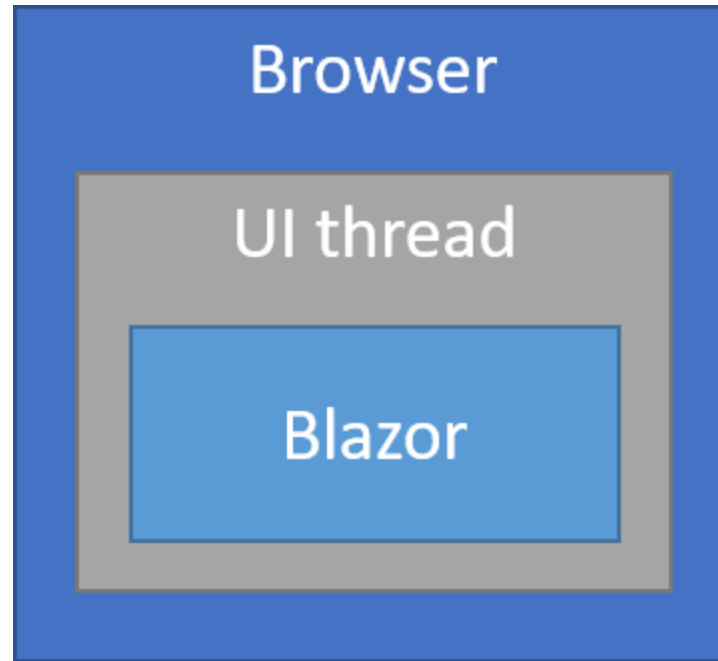
Los mejores IDE's de desarrollo



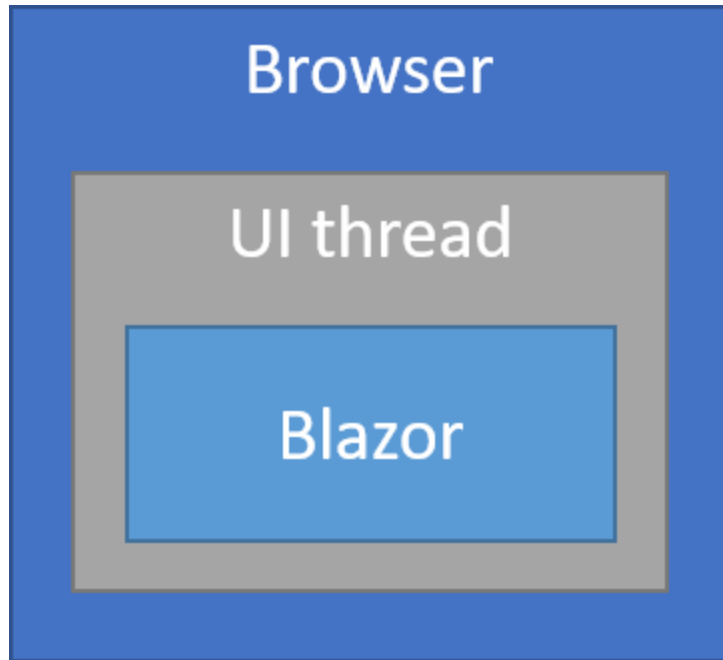
Modelos de Hospedaje Blazor



Blazor client-side



Blazor client-side



Todos los recursos son descargados al navegador

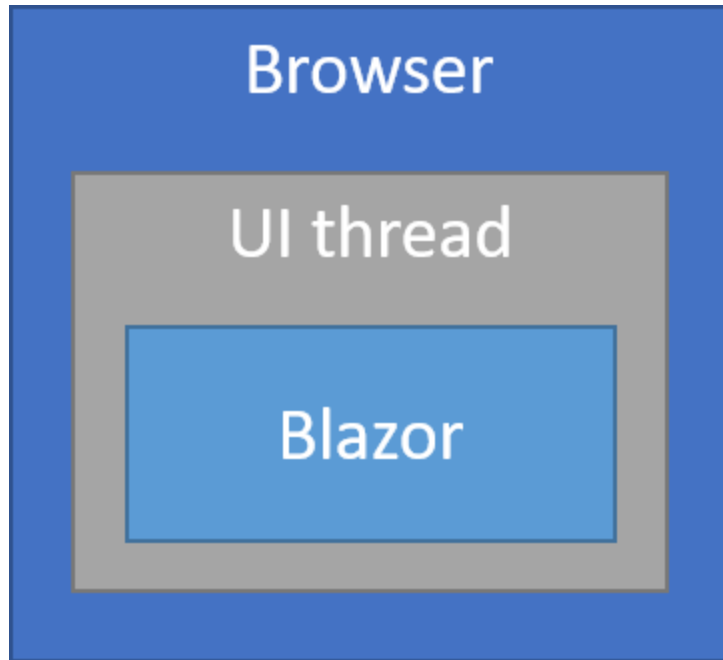
- HTML, CSS y Javascript
- Archivos de la aplicación (Bibliotecas .NET Standard)
- .NET Runtime

No es necesaria una conexión a internet, una vez descargados los archivos

Se ejecuta en WebAssembly



Blazor client-side



BENEFICIOS:

- Rendimiento casi nativo
- Se puede trabajar fuera de línea
- No se necesita de un servidor
- Se utilizan recursos del lado del cliente
- Se puede ejecutar en todos los navegadores modernos



Blazor client-side

Resumen:

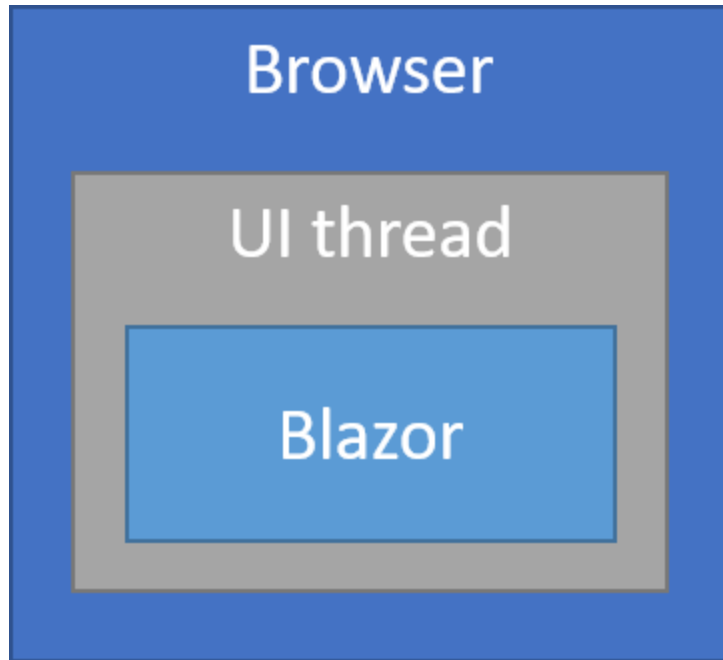
Ejecutamos código .NET en el navegador

Necesitamos WebAssembly

Ventaja: Altamente escalable

Desventaja: Descarga del runtime .NET + DLLs

Blazor client-side

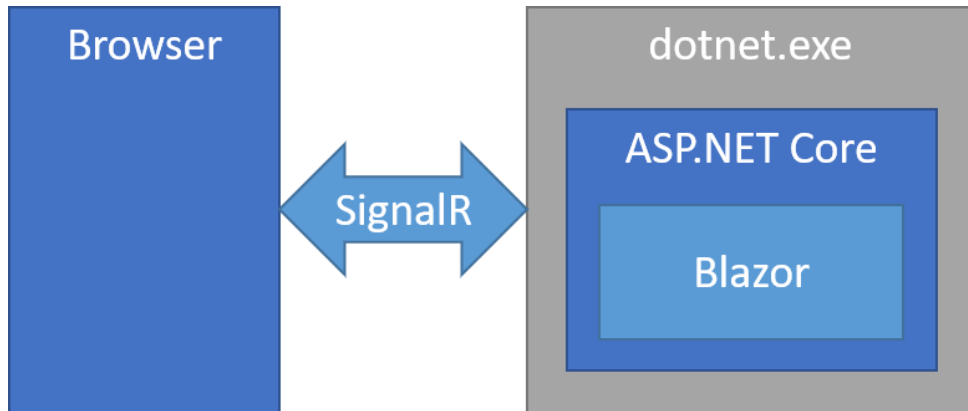


CONTRAS:

- Restringido a las capacidades del navegador
- El navegador hace todo el trabajo
- Toma más tiempo la descarga de los recursos
- No es buena opción para contraseñas
- No es compatible con navegadores antiguos



Blazor server-side



Los recursos son procesados por el servidor, y no se descargan bibliotecas .NET Standard, ni el .NET Runtime.

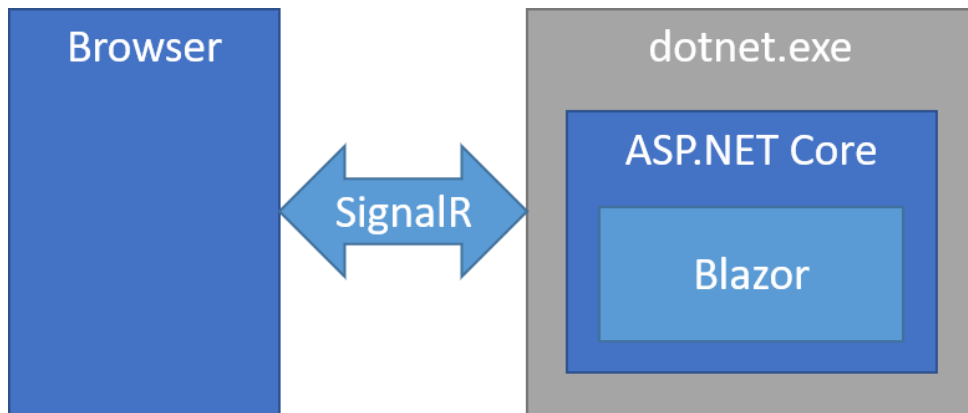
Utilizado para clientes “ligeros”

Conexión en tiempo real a través de Web Sockets

Acceso al framework ASP.NET completo



Blazor server-side

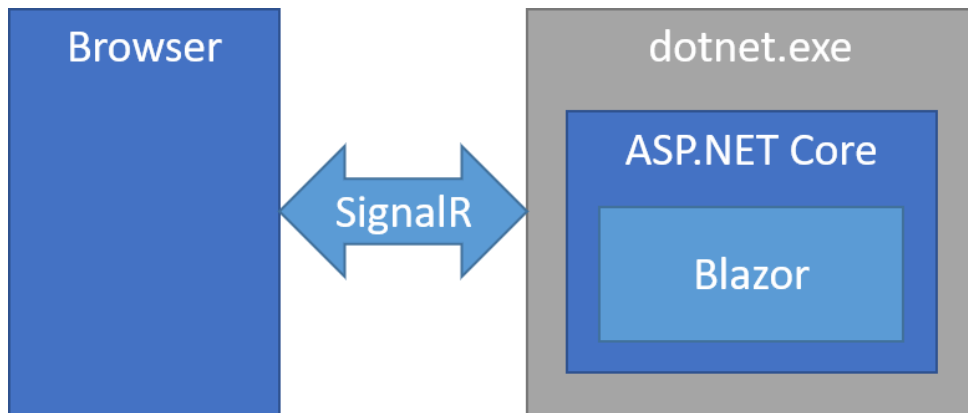


BENEFICIOS:

- Una descarga pequeña
- Acceso al Framework ASP.NET completo
- WebAssembly no es necesario
- Acceso a bases de datos y almacenamiento seguro de claves



Blazor server-side



CONTRAS:

- No hay soporte fuera de línea
- Se necesita un servidor ASP.NET Core
- Es un poco menos rápido que Blazor WebAssembly



Demo – Modelos de Hospedaje Blazor

Demos de Blazor

Preguntas comunes de **Blazor**

¿Se recomienda su uso en
producción?



Preguntas comunes de **Blazor**

¿Es soportado en
exploradores viejitos?



Preguntas comunes de **Blazor**

¿Y si muere como
Silverlight?



Preguntas comunes de **Blazor**

¿Y si sé Javascript, para
qué Blazor?





¡Gracias!