

Desarrollo de Web Api's con .NET

Blazor WebAssembly



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
ENCUENTRO REGIONAL BUENOS AIRES

**Centro de
e-Learning**

¿Qué vamos a ver hoy?

- Que es Blazor
- Nuestra primera APP usando Blazor

¿Qué es Blazor?

Es una plataforma de trabajo para la creación de interfaces de usuario web interactivas del lado cliente con .NET. Nos permite:

- Crear interfaces de usuario con C# en lugar de JavaScript.
- Compartir la lógica de aplicación del lado cliente y servidor escrita con .NET.
- Representar la interfaz de usuario como HTML y CSS para la compatibilidad con todos los exploradores, incluidos los móviles.
- Realizar la integración con plataformas de hospedaje modernas, como Docker.

Ventajas de usar Blazor

- Escribir 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.
- Mantener la productividad con Visual Studio en Windows, Linux y macOS.
- Compilar sobre un conjunto común de lenguajes, marcos y herramientas que son estables, completos y fáciles de usar.

Pensar en componentes

Las aplicaciones de Blazor se basan en componentes. Un componente es un elemento de la interfaz de usuario, como una página, un cuadro de diálogo o un formulario de entrada de datos.

- Definen la lógica de representación de la interfaz de usuario.
- Controlan acciones del usuario.
- Se pueden anidar y reutilizar.
- Se pueden compartir y distribuir como bibliotecas de clases de Razor o paquetes NuGet.

Pensar en componentes

Los componentes normalmente se escriben archivos de extensión “.razor”.

Nos permite cambiar entre marcado HTML y C# en el mismo archivo gracias a la compatibilidad de programación de IntelliSense en Visual Studio.

Blazor usa etiquetas HTML naturales para la composición de la interfaz de usuario.

Pensar en componentes

razor

```
<div class="card" style="width:22rem">
  <div class="card-body">
    <h3 class="card-title">@Title</h3>
    <p class="card-text">@ChildContent</p>
    <button @onclick="OnYes">Yes!</button>
  </div>
</div>

@code {
  [Parameter]
  public RenderFragment ChildContent { get; set; }

  [Parameter]
  public string Title { get; set; }

  private void OnYes()
  {
    Console.WriteLine("Write to the console in C#! 'Yes' button selected.");
  }
}
```

¿Cómo funciona?

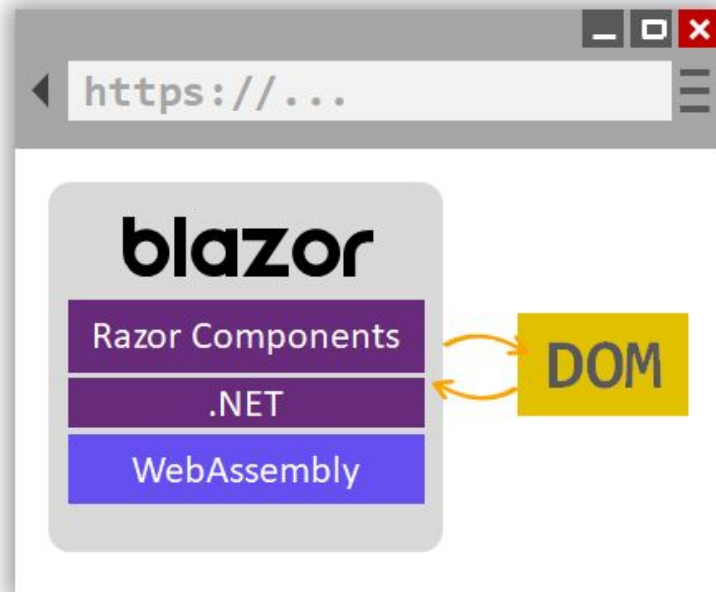
Cuando se compila y se ejecuta una aplicación de Blazor WebAssembly en un explorador:

1. Los archivos de código C# y los archivos de Razor se compilan en ensamblados de .NET.
2. Los ensamblados y el runtime de .NET se descargan en el explorador.
3. Blazor WebAssembly arranca el entorno de ejecución .NET y lo configura para cargar los ensamblados de la aplicación.

¿Cómo funciona?

El entorno de ejecución de Blazor WebAssembly emplea la interoperabilidad de JavaScript para gestionar la manipulación de DOM y las llamadas API del explorador.

[Documentación oficial](#)



¡¡A trabajar!!

