



# Blazor



# O que aprenderemos?

- O que é Blazor?
- Render Modes
- Componentes seu ciclo de vida
- Rotas e navegação
- Formulários
- Validação de formulários
- Scaffold de componentes e conexão com banco de dados
- Desenvolvimento de uma aplicação de Agendamento de consultas, com autenticação e autorização gerenciadas pelo identity.



# O que é Blazor?



# O que é Blazor?

“O Blazor é um framework para desenvolvimento Web, baseado em HTML, CSS e C# que ajuda você a criar aplicativos Web com mais rapidez. Com o Blazor, crie aplicativos Web usando componentes reutilizáveis que podem ser executados do cliente e do servidor para que você possa proporcionar ótimas experiências na Web.”





# Vantagens de utilizar o Blazor

- Escreva código em C#, o que pode melhorar a produtividade no desenvolvimento e manutenção de aplicativos.
- Aproveite o ecossistema .NET existente das bibliotecas .NET.
- Beneficie-se com o desempenho, confiabilidade e segurança do .NET.
- Produtividade Windows, Linux ou macOS



# Render Modes



# O que são os Render Modes?

Render Modes (modos de renderização), é uma nova feature do Blazor que chegou junto com a versão do .Net 8.

Com Render Modes, podemos escolher como nossas páginas e componentes serão renderizados.

Podemos realizar tais configurações nos seguintes níveis:

- Nível de projeto (global)
- Nível de página
- Nível de componente





# Modos de renderização

Nome	Descrição	Local de renderização	Interativo
Servidor estático	Renderização do lado do servidor estática (SSR estática)	Servidor	Não
Servidor interativo	Renderização interativa do lado do servidor (SSR interativo) usando Blazor Server.	Servidor	Sim
WebAssembly Interativo	Renderização interativa do lado do cliente (CSR) usando WebAssembly.	Cliente	Sim
Auto Interativo	SSR interativa usando Blazor Server inicialmente e em seguida, CSR em visitas subsequentes depois que o pacote blazor é baixado	Servidor, depois no cliente	Sim





# Render Modes na prática



# SSR - Renderização Estática

Por padrão, os componentes usam renderização estática no lado do servidor (SSR estático).

O componente é renderizado para o streaming de resposta e a interatividade não é habilitada.





# Stream Rendering

Stream Rendering é um SSR melhorado, ou seja, mais inteligente.

Com esse tipo de renderização, buscamos em nosso servidor aquilo que está pronto para ser carregado, enquanto o restante dos dados é baixado.

Dessa forma buscamos primeiro a nossa página e logo após, os dados.

Date	Temp. (C)	Temp. (F)	Summary
19/04/2024	-13	9	Cool
20/04/2024	3	37	Freezing
21/04/2024	48	118	Bracing
22/04/2024	-9	16	Sweltering
23/04/2024	-13	9	Hot

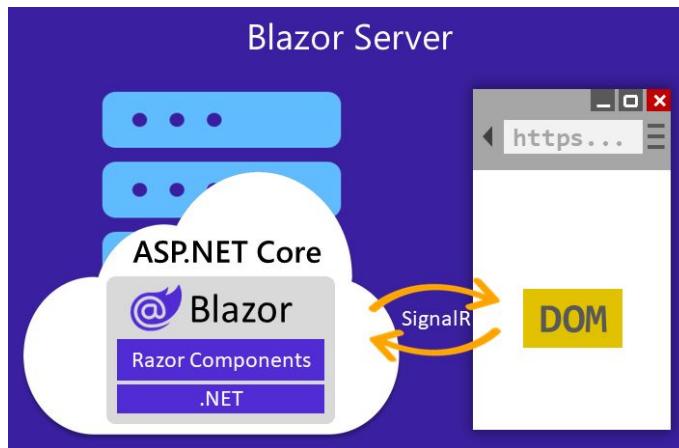




# SSR Interativo - InteractiveServer

A renderização interativa (SSR interativa) renderiza o componente interativamente no servidor usando Blazor Server.

As interações do usuário são tratadas por meio de uma conexão em tempo real com o navegador. A conexão é estabelecida quando o componente é renderizado.

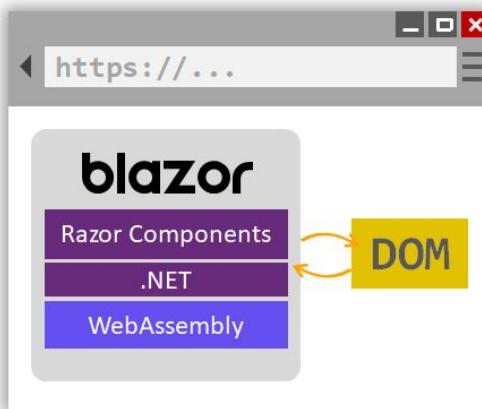




# CSR - Interactive WebAssembly

A renderização do lado do cliente (CSR) renderiza o componente interativamente no cliente usando Blazor WebAssembly.

O runtime do .NET e todas as dlls da nossa aplicação são baixados e armazenados em cache quando o componente do WebAssembly é renderizado inicialmente.





# Renderização Automática (Auto)

A Renderização Automática (Auto) determina como renderizar o componente em runtime. O componente é inicialmente renderizado do lado do servidor com interatividade utilizando o modelo de hospedagem Blazor Server. O runtime do .NET e as demais dlls são baixados para o cliente em segundo plano e armazenados em cache para que possam ser utilizados em visitas futuras.

Os componentes que utilizam o modo de renderização automática devem ser criados a partir de um projeto cliente separado que configure o host Blazor WebAssembly.





# Componentes Blazor



# Componentes Razor

Um componente é uma parte independente da interface do usuário com lógica de processamento para habilitar um comportamento dinâmico. Os componentes podem ser aninhados, reutilizados, compartilhados entre projetos e usados em aplicativos MVC e Razor Pages.

Os componentes são implementados usando uma combinação de marcação C# e HTML com a extensão de arquivo .razor.





# Eventos de ciclo de vida

O componente Razor processa eventos de ciclo de vida em um conjunto de métodos síncronos e assíncronos. Os métodos de ciclo de vida podem ser substituídos para executar operações adicionais nos componentes durante a inicialização e a renderização do componente.

MÉTODOS	DESCRIÇÃO
OnInitialized OnInitializedAsync	Chamado na inicialização do componente
OnParametersSet OnParametersSetAsync	Chamado quando recebemos parâmetros do componente pai
OnAfterRender OnAfterRenderAsync	Chamado no término da renderização do componente
Dispose IDisposable	Chamado na desconstrução do componente





# Rotas e navegação



# Eventos de ciclo de vida

O componente Razor processa eventos de ciclo de vida do componente Razor em um conjunto de métodos de ciclo de vida síncronos e assíncronos. Os métodos de ciclo de vida podem ser substituídos para executar operações adicionais nos componentes durante a inicialização e a renderização do componente.

