

Trabalho Programação AV2

Alunos: Bruno Seixas

Luis Guilherme

Luiz Fernando

Miriã

Vinicius Mota

Introdução

Blazor: Framework para Aplicações Web Interativas

• O que é Blazor?

- Framework da Microsoft
- Cria aplicações web interativas usando C# (em vez de JavaScript)
- Baseado na plataforma .NET e linguagem C#

• Funcionalidades do Blazor

- **Criação de UIs interativas com C#**
 - Suporte ao WebAssembly (WASM)
 - Execução de código C# no cliente
 - Interação com JavaScript via JavaScript Interop
- **Compartilhamento de lógica entre servidor e cliente**
 - Mesma base de código .NET
 - Simplifica desenvolvimento e manutenção
- **Renderização de interface como HTML e CSS**
 - Compatibilidade com navegadores modernos
 - Execução em diversas plataformas
- **Modalidades de hospedagem**
 - Blazor WebAssembly (WASM): Execução no navegador
 - Blazor Server: Execução no servidor com SignalR

- **Características dos componentes no Blazor**

- **Baseado em componentes**
 - Classes .NET encapsuladas em assemblies
 - Modularização e reutilização de código
- **Definição de lógica de renderização**
 - Cada componente define sua renderização
 - Interfaces de usuário dinâmicas e flexíveis
- **Manipulação de eventos do usuário**
 - Eficiente para cliques de botões, entradas de texto, etc.
- **Aninhamento e reutilização**
 - Componentes aninhados e reutilizados
- **Compartilhamento e distribuição**
 - Bibliotecas de classes Razor ou pacotes NuGet

- **Ferramentas de desenvolvimento**

- Blazor CLI, Visual Studio, Visual Studio Code
- Live Reload para mudanças em tempo real

Diferenças entre Blazor e Razor

•Razor:

- É uma sintaxe de marcação baseada em C#.
- Utilizada principalmente para escrever marcação HTML com a adição de C#.
- Comum em Visualizações do ASP.NET MVC.
- Permite vinculação de dados e lógica de visualização na marcação.
- Suporta expressões condicionais e vinculação de modelo.

•Blazor:

- É um framework que usa componentes Razor para gerar HTML dinâmico.
- Diferencia-se do Razor por ser um framework completo, não apenas uma linguagem de marcação.
- Permite a execução de código C# no navegador.
- Combina a pilha Razor .NET com arquitetura SPA moderna.
- Promove a criação de layouts baseados em componentes.
- Espera-se que seja rápido devido ao uso de WebAssembly.

Vantagens e Desvantagens do Blazor

Vantagens do Blazor

1.Simplificação do Desenvolvimento

1. Foco em uma única linguagem: C#
2. Redução da complexidade
3. Ideal para equipes menores

2.Reutilização de Código

1. Compartilhamento entre servidor e cliente
2. Evita duplicação de esforço
3. Melhora a manutenção

3.Execução Nativa

1. Blazor WebAssembly: Quase velocidade de código nativo
2. Comparação com JavaScript interpretado

4.Eficiência

1. Tempo de execução e uso de memória otimizados
2. Aplicações mais rápidas e responsivas

5. Sem Necessidade de Plugins

1. Funciona em navegadores modernos sem plugins
2. Simplifica a experiência do usuário

6. Compatibilidade e Acessibilidade

1. Qualquer usuário com navegador moderno pode acessar
2. Base de usuários ampliada

7. Segurança Aprimorada

1. Menos dependências de plugins
2. Redução de vetores de ataque

Desvantagens do Blazor

1.Tamanho da Aplicação Inicial (Blazor WebAssembly)

1. As aplicações Blazor WebAssembly tendem a ter um tamanho inicial maior devido ao runtime do .NET que é baixado junto com a aplicação.
2. Isso pode resultar em tempos de carregamento mais longos, especialmente em conexões de internet mais lentas.

2.Suporte de Navegador e Compatibilidade

1. Embora o WebAssembly seja suportado pela maioria dos navegadores modernos, pode não ser totalmente compatível com navegadores mais antigos ou versões específicas.
2. Isso pode limitar a audiência da aplicação ou exigir soluções alternativas para garantir a compatibilidade.

3.Performance

1. O Blazor Server pode mitigar algumas questões de desempenho ao fazer o processamento no servidor, mas isso pode introduzir latência devido à comunicação constante entre o cliente e o servidor.
2. Para aplicações em tempo real ou com alta interatividade, essa latência pode ser uma limitação.
3. Por outro lado, o Blazor WebAssembly pode ter limitações de desempenho em comparação com aplicações JavaScript nativas devido ao overhead do runtime .NET e à execução em um ambiente sandbox.

4. **SEO (Search Engine Optimization)**

4. Aplicações SPAs (Single Page Applications), incluindo aquelas desenvolvidas com Blazor, podem enfrentar desafios em termos de SEO.
5. O conteúdo é frequentemente carregado dinamicamente via JavaScript, o que pode ser menos eficaz para indexação por motores de busca.
6. Embora o Blazor Server possa mitigar parte desse problema, ainda pode não ser tão eficiente quanto aplicações renderizadas no servidor.

5. **Ecossistema e Bibliotecas**

4. O ecossistema do Blazor está crescendo, mas ainda é relativamente novo em comparação com frameworks JavaScript estabelecidos, como React, Angular e Vue.
5. Isso pode resultar em menos bibliotecas, componentes de terceiros e recursos disponíveis para os desenvolvedores.

6. **Aprendizado e Adoção**

4. Para desenvolvedores com experiência principalmente em desenvolvimento web tradicional (HTML, CSS, JavaScript), aprender Blazor e C# pode ter uma curva de aprendizado.
5. Além disso, a adoção de uma nova tecnologia pode exigir investimento significativo em treinamento e adaptação.

Conclusão

O Blazor, como um framework emergente para desenvolvimento web, apresenta uma série de vantagens e desafios que os desenvolvedores devem considerar cuidadosamente ao escolher a tecnologia para seus projetos além de oferece uma alternativa promissora para o desenvolvimento web, especialmente para aqueles familiarizados com a plataforma .NET. Ao pesar as vantagens e desafios, os desenvolvedores devem avaliar cuidadosamente as necessidades específicas de seus projetos.

A escolha entre Blazor Server e Blazor WebAssembly dependerá do contexto, dos requisitos de desempenho e da equipe de desenvolvimento. Com uma abordagem estratégica, o Blazor pode ser uma ferramenta poderosa para criar aplicações web modernas e eficientes.

Link para o diretório Github: [LuisGuilhermeRampaso](#)
[\(LuisGuilhermeRampaso\) / Repositories · GitHub](#)