**11/JUL/2025**

- conseguir implementar o controller do Usuario e o Service com modelMapper (interface Mapper)

- OpenApi 3.0 para o controller implementada, tem que testar....

- Usuario Inputs e Usuario DTOs também implementados

- Construindo endpoints de usuario, mas estou com problema de erro 403

- Considerar incorporar as paginas em resource/static dentro do backend

**12/JUL/2025**

Mudei a abordagem, estou construindo o endpoint de CRIAR NOVO USUARIO primeiro, encontrei um problema na carga dos scripts javascript, vou tentar fazer mapeamento dinamico de forma a não deixar hardcoded no codigo

**Excelente descoberta!** Você identificou o problema raiz: o carregamento dinâmico de páginas via innerHTML não executa scripts que estão hardcoded no HTML carregado.

**Por que isso acontece?**

Quando você usa innerHTML para injetar HTML, o browser **não executa** as tags <script> por questões de segurança. É um comportamento padrão do DOM.

// ❌ Scripts não são executados

document.getElementById("conteudo-principal").innerHTML = html;

**Soluções Técnicas**

**Solução 1: Modificar o Core para Executar Scripts (Recomendada)### Solução 2: Remover Scripts Hardcoded e Usar Apenas Mapeamento**

**Opção mais elegante:** Remova todos os scripts hardcoded do HTML e mapeie todas as dependências:

const pageToScriptMap = {

"tela\_cadastro\_recebimento.html": [

"assets/js/fiberguardian\_utils.js",

"assets/js/tela\_cadastro\_recebimento.js"

],

"tela\_cadastro\_usuario.html": [

"assets/js/fiberguardian\_utils.js",

"assets/js/tela\_cadastro\_usuario.js"

],

};

**Solução 3: Abordagem Híbrida (Mais Robusta)**

HTML da tela de cadastro sem scripts hardcoded:

<!-- Remover estas linhas do HTML -->

<!-- <script src="assets/js/fiberguardian\_utils.js" defer></script> -->

<!-- <script src="assets/js/tela\_cadastro\_usuario.js" defer></script> -->

E manter apenas o mapeamento no Core.

**Implementado a solução 02.**