**❓ "Eu preciso mesmo recarregar fiberguardian\_utils.js em cada página HTML?"**

✅ **Sim. Em aplicações MPA (Multi-Page Applications), o navegador faz reload completo da página a cada navegação.**  
Isso significa que:

* O **DOM é recriado**
* O **JS é recarregado**
* Tudo que estava em memória **é perdido**

Logo, **todo script necessário para o funcionamento da nova página precisa ser carregado de novo** — inclusive os utilitários.

**🔒 "Não tem como carregar só uma vez e manter vivo?"**

Não com HTML estático puro (MPA). Isso **não é possível em MPA sem aplicar alguma técnica de SPA (Single Page Application)**. Em HTML padrão, **cada .html é isolado** do ponto de vista de execução JavaScript.

Para “reaproveitar um script ao longo do sistema”, só há **três caminhos possíveis**, e todos exigem mudanças importantes:

**🔁 Alternativas reais (com trade-offs)**

**🔹 1. Migrar para SPA parcial com roteamento JS (Ex: usando window.history.pushState)**

* Você carrega uma **única página HTML**, e **navega entre “telas” sem recarregar a página inteira**.
* Isso permite **carregar fiberguardian\_utils.js uma única vez**.
* Porém, exige:
  + Controle total do DOM via JavaScript
  + Manipulação de histórico e roteamento manual
  + Reimplementação de "telas" como templates dinâmicos ou innerHTML

🧠 **Não recomendado no seu caso** se a proposta é MPA com páginas reais (index.html, relatorio.html, etc). Complica mais do que ajuda.

**🔹 2. Carregar dinamicamente via import() com módulos ES6 + CDN ou bundler**

Se você configurar seus scripts com módulos ES6 (type="module"), pode **modularizar e reaproveitar código de forma declarativa**:

html

CopiarEditar

<script type="module">

import { obterTokenCsrf } from './assets/js/fiberguardian\_utils.js';

</script>

**Mas ainda será carregado por página**, embora com *caching* pelo navegador (mais leve). A vantagem aqui é a **organização e reaproveitamento via import/export**, não o carregamento único em runtime.

Requer:

* Navegadores modernos
* Ajustes no servidor para servir type=module
* Ajustar estrutura para usar export function obterTokenCsrf() etc.

**🔹 3. Manter como está: carregar fiberguardian\_utils.js em cada página**

🔁 Simples, robusto, compatível, seguro.  
💡 Aproveite o **cache do navegador**: se o servidor estiver configurado com cabeçalhos de cache adequados (Cache-Control, ETag, etc), o navegador não baixa o arquivo de novo — apenas reutiliza o conteúdo da última vez.

💬 **Na prática:**  
Você só precisa fazer isso:

html

CopiarEditar

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

em **cada .html do sistema**, e o browser vai se encarregar do caching.

**✅ Recomendação final para seu caso MPA:**

**Continue incluindo fiberguardian\_utils.js em cada página HTML.**

➡️ Para garantir performance:

1. Habilite cache no servidor (se usar Caddy, posso te mostrar como).
2. Organize esse script com funções puras e isoladas.
3. Evite dependências externas desnecessárias.