

☑ CSV vs JSON — Qual usar?

Critério	CSV (.csv)	JSON (.json)
Formato	Texto simples em tabela (linhas e colunas)	Estrutura organizada em objetos <code>{ chave: valor }</code>
Facilidade para exibir como tabela	Excelente (ideal para planilhas, tabelas simples)	Precisa converter ou mapear os dados para tabela
Facilidade para usar no JavaScript	Mais difícil → precisa separar por vírgulas, converter manualmente	Muito fácil → já vira objeto manipulável com <code>fetch().json()</code>
Dados complexos (listas dentro de listas, objetos dentro de objetos)	Ruim → não suporta estrutura hierárquica	Excelente → suporta aninhamento, arrays, objetos
Tamanho do arquivo (compacto ou pesado)	Menor	Um pouco maior (pois tem chaves e aspas)
API, aplicações web modernas	Pouco usado diretamente hoje	Muito usado (padrão de APIs e JavaScript)
Leitura por humanos	Boa se for tabela simples	Boa para quem entende estrutura de objeto
Uso comum em GitHub Pages	Usável, mas exige mais código para tratar	Preferido, pois integra direto com JavaScript

🔗 Qual é melhor para o seu caso?

✓ Use **.json** se:

- Vai usar com JavaScript (`fetch()`) para atualizar a página automaticamente.
- Os dados têm estrutura (ex.: lista de produtos, usuários, configurações).
- Pretende atualizar os dados e manipular eles em código.
- Quer evitar complicações com parsing (separar vírgula, linha, etc.).

✓ Use **.csv** se:

- São dados tabulares simples (tabelas parecidas com Excel).
 - Precisa exportar/importar facilmente para Excel, Google Sheets, Power BI.
 - Os dados não precisam de estruturas complexas (apenas linhas e colunas).
-

📌 Resumo rápido:

- Para sites estáticos no GitHub Pages + JavaScript → **.json** é geralmente melhor.
- Para dados estilo planilha, relatórios ou integração com Excel → **.csv** é mais prático.

✓ O que é “cross” (CORS)?

CORS (Cross-Origin Resource Sharing) é um mecanismo de segurança dos navegadores que **bloqueia requisições feitas de um site para outro domínio/diretório/origem diferente**, caso o servidor não autorize isso.

Exemplo de situações que geram erro de *cross*:

Situação	Vai funcionar?	Explicação
<code>index.html</code> e <code>dados.csv</code> estão no mesmo repositório GitHub Pages	✓ Funciona	Mesma origem (mesmo domínio, protocolo e porta)
Página tenta carregar um <code>.csv</code> de outro site ou domínio (Ex: <code>https://outro-site.com/arquivo.csv</code>)	✗ Bloqueado	O navegador exige permissão CORS do servidor remoto
Carregar arquivo <code>.csv</code> do seu próprio computador (<code>file://</code>)	✗ Bloqueado	Navegador não permite leitura local sem um servidor
Carregar <code>.csv</code> com <code>fetch()</code> normalmente dentro do GitHub Pages	✓ Sim	Desde que esteja no mesmo domínio

📁 Limitações ao usar CSV diretamente no frontend:

Limitação	O que significa?
⚠ Só funciona se o CSV estiver no mesmo domínio/origem	Se estiver em outro servidor, precisa de CORS habilitado.
⚠ Não pode ser atualizado dinamicamente sem reupload	GitHub Pages é estático; sempre precisa de novo commit no arquivo.
⚠ Não há backend para processar CSV no GitHub Pages	Toda leitura e tratamento tem que ser feita com JavaScript no navegador.
⚠ CSV é texto plano sem estrutura semântica	Precisa converter manualmente para objetos (Array, JSON) no JavaScript.
⚠ Pode ser cacheado pelo navegador	Igual ao JSON: você pode ver dados antigos se não limpar cache ou evitar com <code>?nocache</code> .

✓ Como carregar um `.csv` no GitHub Pages usando JavaScript (sem erro CORS):

Se o arquivo estiver no mesmo repositório:

```
fetch('dados.csv')
  .then(response => response.text())
  .then(texto => {
    const linhas = texto.split('\n').map(l => l.split(','));
    console.log(linhas); // Agora é um array com os dados do CSV
  })
  .catch(err => console.error('Erro ao carregar CSV:', err));
```

☑ Resumindo:

✓ Não existe problema de CORS (cross) se o arquivo .CSV estiver **no mesmo repositório/domínio do GitHub Pages**.

✗ **Problemas aparecem se estiver em outro domínio ou local (file://).**

🔗 CSV é **estático**, sem backend — só dá para atualizar com novos commits.

🔗 Você precisa tratá-lo com JS para converter para objetos utilizáveis.