

A REFERÊNCIA COMPLETA DE MARKDOWN

Markdown Cheat Sheet

Toda a Sintaxe com Exemplos • 2026

Cada recurso do CommonMark e do GitHub Flavored Markdown, com um exemplo para copiar e colar, de cabeçalhos e tabelas a código, matemática e diagramas.

[Baixar o Markdific](#)

O cheat sheet de Markdown mais completo da internet. Cada recurso do CommonMark e do GitHub Flavored Markdown, com um exemplo funcional para cada um.

Adicione esta página aos favoritos. Ou baixe a versão gratuita do [cheat sheet de Markdown em PDF](#) para tê-la sempre à mão, mesmo offline.

Sintaxe do Markdown num relance

Elemento	Sintaxe
Cabeçalho	# H1 até ##### H6
Negrito	**negrito**
Itálico	<i>*itálico*</i>
Tachado	~~riscado~~
Link	[texto](https://example.com)
Imagem	![alt](image.jpg)
Código inline	`código`
Bloco de código	três crases, seu código, depois três crases
Citação	> texto citado
Lista não ordenada	- item
Lista ordenada	1. item
Lista de tarefas	- [] a fazer e - [x] feito
Tabela	A B com --- --- abaixo
Linha horizontal	---

O que é Markdown?

Markdown é uma sintaxe de formatação em texto puro criada por John Gruber e Aaron Swartz em 2004. Ela usa símbolos simples para marcar o texto de modo que ele possa ser convertido em HTML e outros formatos. O objetivo era a legibilidade. Um arquivo markdown deve parecer limpo como texto puro e ficar mais bonito apenas quando renderizado.

Hoje o markdown está em todo lugar. READMEs do GitHub, posts do Reddit, mensagens do Discord, páginas do Notion, notas do Obsidian, respostas de chatbots de IA e a maioria dos sites de documentação usam markdown. Se você escreve qualquer coisa em um computador em 2026, quase certamente está produzindo markdown, quer perceba isso ou não.

Para saber mais sobre o formato em si, leia [o que é um arquivo .md, como abrir e ler um arquivo .md](#), ou [Markdown vs HTML](#) para entender como os dois formatos diferem. Para uma referência completa, elemento por elemento, veja a [documentação do Markdown](#).

Cabeçalhos

Use símbolos `#` no início de uma linha para criar cabeçalhos. O número de cerquilhas define o nível do cabeçalho, de H1 (maior) a H6 (menor).

```
# Cabeçalho 1
## Cabeçalho 2
### Cabeçalho 3
#### Cabeçalho 4
##### Cabeçalho 5
##### Cabeçalho 6
```

Sintaxe alternativa de cabeçalho (Setext)

Apenas para H1 e H2, você pode usar cabeçalhos no estilo sublinhado:

```
Cabeçalho 1
=====

Cabeçalho 2
-----
```

A maioria dos autores prefere a sintaxe `#` porque ela funciona em todos os níveis e é mais fácil de escanear.

Dicas:

- Sempre coloque um espaço entre o `#` e o texto do cabeçalho
- Use apenas um H1 por documento (ele é o título)
- Deixe uma linha em branco antes e depois dos cabeçalhos para uma renderização mais segura

Guia completo: [Cabeçalhos no Markdown](#) →

Parágrafos e quebras de linha

Parágrafos são separados por uma linha em branco. Basta apertar Enter duas vezes.

```
Este é o primeiro parágrafo.

Este é o segundo parágrafo.
```

Quebras de linha dentro de um parágrafo

Para quebrar uma linha sem começar um novo parágrafo, termine a linha com **dois espaços** seguidos de Enter, ou use uma barra invertida `\`:

```
Primeira linha.  
Segunda linha em uma nova linha, mesmo parágrafo.  
  
Primeira linha.\nSegunda linha usando barra invertida.
```

Ambos produzem uma tag `<br>` em HTML. O método dos espaços no final é mais tradicional, mas fica invisível no seu editor, o que causa confusão. O método da barra invertida é mais claro e tem suporte no GitHub Flavored Markdown.

Guia completo: [Parágrafos](#) → · [Quebras de linha](#) →

Negrito, itálico e outras formatações de texto

Estes são os fundamentos de formatação que você vai usar o tempo todo.

```
**texto em negrito** ou __texto em negrito__  
*texto em itálico* ou _texto em itálico_  
***negrito e itálico*** ou ___negrito e itálico___  
~~texto tachado~~
```

Negrito e itálico fazem parte do núcleo do CommonMark e funcionam em todo lugar. O tachado é uma extensão do GFM abordada em mais detalhes abaixo.

Quando usar asteriscos vs sublinhados

Tanto `**` quanto `__` produzem negrito. Tanto `*` quanto `_` produzem itálico. Eles são intercambiáveis, então escolha um e mantenha-o por consistência.

A diferença prática aparece dentro de uma palavra. O CommonMark e o GFM ignoram sublinhados entre letras, então os sublinhados permanecem literais, enquanto os asteriscos sempre acionam ênfase:

```
Permanece literal:  esta_palavra_tem_sublinhados  
Vira itálico:       esta*palavra*tem*asteriscos
```

Use sublinhados em textos que contêm sublinhados (como `file_name`) para evitar itálico accidental, e use asteriscos quando você realmente quiser ênfase dentro de uma palavra:

```
in**acredit**ável
```

Guia completo: [Ênfase: negrito e itálico](#) →

Listas

As listas vêm em dois tipos: não ordenada (marcadores) e ordenada (numerada).

Listas não ordenadas

Use `-`, `*` ou `+` seguido de um espaço. Os três funcionam de forma idêntica. A maioria dos guias de estilo prefere `-`.

```
- Primeiro item
- Segundo item
- Terceiro item
  - Item aninhado (reco de 2 ou 4 espaços)
  - Outro item aninhado
    - Item aninhado profundo
- Quarto item
```

Listas ordenadas

Use números seguidos de um ponto. Os números reais não importam, o markdown renumera automaticamente:

```
1. Primeiro item
2. Segundo item
3. Terceiro item
  1. Item aninhado
  2. Outro item aninhado
4. Quarto item
```

Você também pode escrever assim e obter o mesmo resultado:

```
1. Primeiro item
1. Segundo item
1. Terceiro item
```

Isso é um recurso, não um bug. Significa que você pode reordenar os itens da lista sem renumerar manualmente.

Listas mistas

Você pode aninhar listas não ordenadas dentro de listas ordenadas e vice-versa:

```
1. Passo um
2. Passo dois
  - Subponto A
  - Subponto B
3. Passo três
```

Guia completo: [Listas no Markdown](#) →

Links

Os links usam colchetes para o texto visível e parênteses para a URL.

Links inline básicos

```
[Markdific](https://markdific.com)
[Visite nosso blog](https://markdific.com/blog "Título opcional ao passar o mouse")
```

Links automáticos

Envolva uma URL em sinais de menor e maior para transformá-la em link automaticamente:

```
<https://markdific.com>
<hello@example.com>
```

Links no estilo de referência

Úteis quando o mesmo link aparece várias vezes ou quando você quer manter o texto limpo:

```
Leia nosso [cheat sheet][1] ou o [blog][2].
```

```
Mais adiante no documento ou no final:
```

```
[1]: https://markdific.com/resources/markdown-cheat-sheet
[2]: https://markdific.com/blog/
```

Você também pode usar referências nomeadas:

```
Confira o [Markdific][md-home].

[md-home]: https://markdific.com
```

Link para seções (âncoras)

O GitHub e a maioria dos renderizadores geram automaticamente IDs de âncora a partir do texto do cabeçalho (em minúsculas, espaços substituídos por hifens):

```
[Ir para Tabelas](#tables)
[Ir para Alertas do GitHub](#github-alerts)
```

Guia completo: [Links](#) → · [Links automáticos](#) → · [IDs de cabeçalho](#) →

Wikilinks e embeds

O Obsidian, e algumas outras ferramentas de notas, adicionam wikilinks e embeds com colchetes duplos além do Markdown padrão:

```
[[Título da nota]]          link para outra nota
[[Título da nota|Apelido]] link com texto personalizado
![[Título da nota]]         incorpora o conteúdo de outra nota
![[image.png]]             incorpora uma imagem
```

Estes são uma extensão do Obsidian, não Markdown padrão, então não são resolvidos no GitHub nem na maioria dos sites estáticos. Veja [Markdown no Obsidian](#) para o conjunto completo.

Imagens

A sintaxe de imagem é quase idêntica à de links, com um ponto de exclamação na frente.

```
![Texto alternativo descrevendo a imagem](path/to/image.jpg)
![Texto alternativo](path/to/image.jpg "Título opcional ao passar o mouse")
```

Dimensionamento de imagem

O markdown puro não suporta dimensionamento de imagem. Você tem duas opções:

Opção 1: usar HTML diretamente

```

```

Opção 2: usar extensões específicas de plataforma

O Obsidian, por exemplo, suporta:

```
![[image.jpg|500]]
```

Imagens no estilo de referência

```
![Logo do Markdific][logo]
```

```
[logo]: /images/logo.png "Markdific"
```

Transformar uma imagem em link

Para tornar uma imagem clicável, envolva a sintaxe de imagem em um link:

```
[![Texto alternativo](image.jpg)](https://destination-url.com)
```

Para casos que o markdown puro não cobre, arquivos locais, dimensionamento preciso ou imagens embutidas em Base64, use a tag HTML `<img>` mostrada acima.

Guia completo: [Imagens no Markdown](#) →

Código e blocos de código

A formatação de código é um dos recursos mais úteis do markdown.

Código inline

Envolva o código em crases simples:

Use a função ``print()`` em Python.

Se o seu código contém uma crase, envolva-o em crases duplas:

O caractere ``` ` ` ``` é chamado de crase.

Blocos de código (cercados)

Use crases triplas para blocos de código de várias linhas. Adicione um identificador de linguagem após as crases de abertura para realce de sintaxe:

```
```python
def hello_world():
 print("Hello, world!")
```
```

```
```javascript
function helloWorld() {
 console.log("Hello, world!");
}
```
```

```
```bash
npm install markdown-it
```
```

Identificadores de linguagem comuns

- `python` , `py`
- `javascript` , `js`
- `typescript` , `ts`
- `bash` , `sh` , `shell`
- `html` , `xml`
- `css` , `scss`
- `json` , `yaml`
- `sql` , `graphql`
- `markdown` , `md`
- `diff` (para diffs do git)
- `plaintext` , `text` (sem realce)

Blocos de código com recuo (legado)

Recue qualquer linha em quatro espaços ou um tab para transformá-la em um bloco de código. Isso funciona, mas é menos legível do que os blocos cercados. Evite.

Guia completo: [Código](#) → · [Blocos de código cercados](#) → · [Realce de sintaxe](#) →

Citações

Use `>` no início de uma linha para criar uma citação.

```
> Esta é uma citação.  
> Ela pode se estender por várias linhas.  
  
> Você também pode ter uma citação de uma única linha.
```

Citações aninhadas

Empilhe caracteres `>`:

```
> Citação externa.  
>  
> > Citação aninhada.  
> >  
> > > Citação profundamente aninhada.
```

Citações com outros elementos de Markdown

As citações podem conter cabeçalhos, listas, código e outras formatações:

```
> ### Um cabeçalho dentro de uma citação  
>  
> - Um item de lista  
> - Outro item  
>  
> Um pouco de `código inline` e texto em negrito.
```

Guia completo: [Citações](#) →

Linhas horizontais

Três ou mais hifens, asteriscos ou sublinhados em sua própria linha:

```
---  
***  
---
```

Os três renderizam como `<hr>`. Use-os com moderação. A maioria dos documentos modernos usa hierarquia de cabeçalhos em vez disso.

Guia completo: [Linhas horizontais](#) →

Tabelas

As tabelas são uma extensão do GitHub Flavored Markdown, não do CommonMark padrão, mas hoje têm suporte em quase todo lugar.

Tabela básica

```
Cabeçalho 1	Cabeçalho 2	Cabeçalho 3
Célula 1	Célula 2	Célula 3
Célula 4	Célula 5	Célula 6
```

Alinhamento de colunas

Adicione dois-pontos à linha separadora:

```
Alinhado à esquerda	Centralizado	Alinhado à direita
Texto	Texto	Texto
Texto mais longo	Texto mais longo	Texto mais longo
```

- `:---` alinha à esquerda (padrão)
- `:---:` centraliza
- `---:` alinha à direita

Dicas para tabelas mais limpas

- As barras verticais externas são opcionais, mas melhoram a legibilidade
- As larguras das colunas no código-fonte não afetam a saída (os renderizadores ignoram espaços extras)
- Caracteres de barra vertical dentro das células devem ser escapados como `\|`
- As tabelas GFM padrão não podem ter células de várias linhas nem células mescladas; use HTML para isso

Para um guia completo, incluindo como lidar com tabelas complexas, além do nosso [gerador de tabelas markdown](#) gratuito, leia nosso [guia completo de tabelas em markdown](#).

Guia completo: [Tabelas no Markdown](#) →

Listas de tarefas

As listas de tarefas (também chamadas de listas de afazeres) são uma extensão do GFM com suporte no GitHub, GitLab e na maioria dos editores modernos.

- [x] Tarefa concluída
- [] Tarefa em aberto
- [] Outra tarefa em aberto
 - [x] Subtarefa aninhada concluída
 - [] Subtarefa aninhada em aberto

O `[x]` diferencia maiúsculas de minúsculas em alguns renderizadores. Use `x` minúsculo para maior segurança.

Guia completo: [Listas de tarefas](#) →

Notas de rodapé

As notas de rodapé são uma extensão do GitHub, e também têm suporte no MultiMarkdown e no Pandoc. São perfeitas para citações e observações.

Esta é uma frase com uma nota de rodapé.^[1]

Você pode ter várias notas de rodapé.^[note]

^[1]: Este é o conteúdo da primeira nota de rodapé.

^[note]: Esta é uma nota de rodapé nomeada com conteúdo mais longo que pode se estender por várias linhas se estiver recuado.

As notas de rodapé se numeram automaticamente e aparecem no final do documento renderizado com links de retorno.

Guia completo: [Notas de rodapé](#) →

Listas de definição

As listas de definição têm suporte no Pandoc, no MultiMarkdown e em algumas outras variações, mas não no GFM básico.

Termo

: Definição do termo.

Markdown

: Uma linguagem de marcação leve para criar texto formatado.

: Criada por John Gruber em 2004.

Verifique seu renderizador de destino antes de depender delas. Elas são inconsistentes entre plataformas.

Guia completo: [Listas de definição](#) →

Abreviações

O MultiMarkdown e o PHP Markdown Extra permitem definir uma abreviação uma vez e fazê-la expandir onde quer que o termo apareça:

A especificação HTML é mantida pelo W3C.

*[HTML]: HyperText Markup Language

*[W3C]: World Wide Web Consortium

Isso não faz parte do CommonMark nem do GFM, então a maioria dos renderizadores (incluindo o GitHub) mostra as linhas de definição como texto literal. Use apenas onde você sabe que o destino oferece suporte.

Tachado

Envolva o texto em tís duplos:

~~Este texto está tachado~~

Renderizado: Este texto está tachado

Extensão do GFM. Suporte universal em renderizadores modernos.

Guia completo: [Tachado](#) →

Subscrito, sobrescrito e destaque

Estas três marcas inline são recursos estendidos. Nenhuma faz parte do núcleo do CommonMark ou do GitHub Flavored Markdown, então o suporte varia, mas cada uma tem uma alternativa confiável em HTML que funciona em qualquer lugar.

Subscrito e sobrescrito

O Pandoc, o MultiMarkdown e algumas outras variações suportam a sintaxe de til e circunflexo:

A água é H~2~0.

A área é 10 m^2^.

Onde essa sintaxe não tem suporte (GitHub e a maioria dos renderizadores CommonMark), use as tags HTML, que renderizam em todo lugar:

A água é H₂0.

A área é 10 m².

Destaque

O Obsidian e algumas outras ferramentas destacam o texto envolvido em sinais de igual duplos:

0 Markdown faz ==esta parte== se destacar.

Para renderizadores que não suportam ==, use a tag HTML `<mark>`: `<mark>destacado</mark>`.

Guia completo: [Subscrito e sobrescrito](#) → · [Destaque](#) →

Emoji

Três maneiras de usar emoji no markdown:

1. Colar diretamente

Basta digitar ou colar o caractere de emoji real: 🎉 ✅ 🚀

Isso funciona em todo lugar porque emojis são apenas caracteres Unicode.

2. Shortcodes (GFM)

O GitHub e o GitLab suportam a sintaxe de shortcode que converte códigos de texto em emoji:

```
:tada: :white_check_mark: :rocket:
```

Estes renderizam como 🎉 ✅ 🚀 em plataformas compatíveis. O Discord e o Slack usam uma sintaxe `:code:` semelhante, mas com suas próprias bibliotecas de emoji, então os códigos disponíveis diferem. Veja [Markdown no Discord](#) e [Markdown no Slack](#) para saber como cada app de chat lida com a formatação.

3. Entidades HTML

Para máxima compatibilidade, você também pode usar códigos de entidade HTML, embora os shortcodes sejam muito mais legíveis.

Guia completo: [Emojis no Markdown](#) →

HTML no Markdown

Você pode incorporar HTML puro dentro do markdown para casos que a sintaxe não cobre:

```
<details>
<summary>Clique para expandir</summary>

Este conteúdo fica oculto por padrão e é exibido quando o usuário clica.

</details>
```

Usos comuns de HTML puro:

- `<details>` e `<summary>` para seções recolhíveis
- `<sub>` e `<sup>` para subscrito e sobrescrito
- `<kbd>` para teclas do teclado (renderiza com o estilo Ctrl + C)

- `<mark>` para destaque
- `<img>` com atributos de largura para imagens dimensionadas
- Tabelas com células mescladas, células de várias linhas ou layouts complexos

Exemplo de teclas do teclado

Pressione `<kbd>Ctrl</kbd> + <kbd>C</kbd> para copiar.`

Ressalvas

Algumas plataformas removem ou sanitizam o HTML por segurança (o Reddit, por exemplo). O GitHub permite a maior parte do HTML, mas bloqueia `<script>` e algumas outras tags. Sempre teste na sua plataforma de destino.

Guia completo: [HTML inline no Markdown](#) →

Comentários

O Markdown não tem sintaxe oficial de comentário, mas há duas maneiras confiáveis de deixar notas em um arquivo que nunca aparecem na saída renderizada.

Comentários HTML

Qualquer renderizador que permita HTML também honra comentários HTML:

```
<!-- Esta nota é invisível na página renderizada. -->
```

O conteúdo visível continua aqui.

O comentário é ignorado quando o arquivo é renderizado. Ele ainda existe no código-fonte HTML, porém, então trate-o como oculto para os leitores, e não como verdadeiramente privado.

O truque do link de referência

Para renderizadores que removem HTML (alguns apps de chat e visualizadores em sandbox), use uma definição de link de referência vazia. Ela não produz saída porque nada nunca aponta para ela:

```
[//]: # (Este é um comentário que não renderiza nada.)
```

```
[comment]: # (Outra maneira de escrever a mesma coisa.)
```

Isso funciona em quase todos os parsers porque uma definição de link que nunca é referenciada é simplesmente descartada. É a maneira mais portátil de esconder uma nota.

Caracteres de escape

Para exibir um caractere de markdown literalmente, prefixe-o com uma barra invertida:

```
\* Isto não é itálico \*
\# Isto não é um cabeçalho
\[Isto não é um link\](not-a-url)
```

Caracteres que você pode escapar:

```
\ ` * _ { } [ ] ( ) # + - . ! | `
```

Guia completo: [Escapando caracteres](#) →

Equações matemáticas (LaTeX)

A matemática tem suporte no GitHub, GitLab, Obsidian, Notion e na maioria das ferramentas de documentação modernas como um recurso do renderizador. Use a sintaxe LaTeX dentro de cifrões.

Matemática inline

Cifrões simples para matemática inline:

O teorema de Pitágoras é $a^2 + b^2 = c^2$.

O GitHub também suporta um delimitador inline alternativo que envolve a matemática em um cifrão e uma crase de cada lado, para casos em que a matemática contém caracteres que conflitam com o Markdown:

Aqui $`a + b = c`$ usa os delimitadores alternativos.

Matemática em bloco

Cifrões duplos para equações em nível de bloco:

```
$$
\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}
$$
```

Padrões comuns de LaTeX

Frações:	$\frac{a}{b}$
Raiz quadrada:	$\sqrt{x}$
Potências:	x^2
Subscritos:	x_i
Letras gregas:	$\alpha, \beta, \gamma, \pi, \theta$
Somatório:	$\sum_{i=1}^n i$
Integral:	$\int_0^1 x \, dx$
Matriz:	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Os renderizadores variam na sintaxe. O GitHub usa MathJax. Algumas plataformas exigem KaTeX, que tem convenções ligeiramente diferentes.

Guia completo: [Matemática LaTeX](#) →

Diagramas Mermaid

O Mermaid é uma biblioteca JavaScript que transforma texto em diagramas. GitHub, GitLab, Notion, Obsidian e a maioria dos renderizadores de markdown modernos o suportam nativamente. É um dos recursos mais úteis do markdown moderno.

Fluxograma

```
```mermaid
flowchart TD
 A[Início] --> B{Está funcionando?}
 B -->|Sim| C[Publicar]
 B -->|Não| D[Depurar]
 D --> B
```
```

Diagrama de sequência

```
```mermaid
sequenceDiagram
 Usuário->>Navegador: Abrir arquivo .md
 Navegador->>Markdific: Renderizar markdown
 Markdific-->>Navegador: Retornar HTML
 Navegador-->>Usuário: Mostrar página formatada
```
```

Outros tipos de diagrama Mermaid

- `flowchart` (fluxos de processo)
- `sequenceDiagram` (interações ao longo do tempo)
- `classDiagram` (classes UML)
- `stateDiagram-v2` (máquinas de estado)
- `erDiagram` (esquemas de banco de dados)
- `gantt` (cronogramas de projeto)
- `pie` (gráficos de pizza)
- `mindmap` (mapas mentais)
- `journey` (jornadas do usuário)
- `gitGraph` (visualizações de branches do git)

O Mermaid suporta muito mais tipos de diagrama do que os mostrados aqui. Quais deles renderizam depende da versão do Mermaid da sua ferramenta, o GitHub e o GitLab mantêm as suas atualizadas, então a maioria dos tipos de diagrama funciona sem configuração.

Guia completo: [Diagramas Mermaid](#) →

Alertas do GitHub

Os Alertas do GitHub (também chamados de "callouts" ou "admonitions") são um recurso do GitHub lançado no final de 2023. Eles renderizam citações especiais com ícones e cores, e agora têm amplo suporte em outros renderizadores de markdown.

```
> [!NOTE]
> Informação útil que os usuários devem saber.

> [!TIP]
> Conselho útil para fazer as coisas melhor.

> [!IMPORTANT]
> Informação essencial que os usuários precisam saber.

> [!WARNING]
> Informação urgente que precisa de atenção imediata.

> [!CAUTION]
> Alerta sobre riscos ou resultados negativos.
```

Os cinco tipos suportados são `NOTE`, `TIP`, `IMPORTANT`, `WARNING` e `CAUTION`. Outras ferramentas de markdown (Obsidian, MkDocs, Docusaurus) usam uma sintaxe semelhante, mas ligeiramente diferente para callouts. Veja [Markdown no GitHub](#) para saber como os alertas e outros recursos do GitHub renderizam.

YAML Frontmatter

O frontmatter é um bloco de metadados YAML no topo de um arquivo markdown, usado por geradores de sites estáticos (Hugo, Jekyll, Astro, Next.js), plataformas CMS e apps de anotações.

```
---
title: "O Título do Meu Artigo"
date: 2026-01-15
author: Jane Doe
tags:
  - markdown
  - tutorial
  - referência
draft: false
description: Um breve resumo para SEO.
---

# O conteúdo do artigo começa aqui
```

Três hifens abrem e fecham o bloco. Dentro, use a sintaxe YAML padrão. Campos comuns:

- `title`
- `date`
- `author` ou `authors`
- `tags` ou `categories`
- `description` (para meta tags)
- `draft` (booleano)
- `slug` ou `permalink`
- `image` (caminho da imagem em destaque)

O frontmatter é invisível na saída renderizada. É puramente metadado para o sistema que processa o arquivo.

Se o seu frontmatter aparecer como texto literal na saída, o parser provavelmente não o suporta, verifique se o arquivo começa na linha 1 com `---` e não tem linha em branco acima.

Guia completo: [Front matter](#) →

Comparação das variações (flavors) de Markdown

O Markdown tem várias "variações" (flavors) com recursos diferentes. Saber qual delas sua plataforma de destino usa evita renderização quebrada.

| Variação | Onde é usada | Principais recursos |
|--------------------------------|--|--|
| CommonMark | Reddit, Stack Overflow | O núcleo padronizado. Previsível, mas limitado. |
| GitHub Flavored Markdown (GFM) | GitHub, GitLab, a maioria das ferramentas modernas | CommonMark + tabelas, listas de tarefas, tachado, autolinks, notas de rodapé, alertas |
| MultiMarkdown | Escrita acadêmica e técnica | Adiciona tabelas, notas de rodapé, matemática, citações, metadados |
| Pandoc Markdown | Conversor Pandoc | O mais poderoso. Adiciona listas de definição, matemática, tabelas, citações, HTML puro |
| Obsidian Markdown | App de anotações Obsidian | GFM + wikilinks <code>[[note]]</code> , callouts, embeds, tags |
| R Markdown | R, RStudio, Quarto | Adiciona blocos de código executáveis para análise de dados |
| Discord / Slack | Apps de chat | Subconjuntos personalizados com sintaxe específica de plataforma (spoilers, menções, códigos de emoji) |

Na dúvida, escreva em GFM. É a coisa mais próxima de um padrão universal de markdown em 2026. Para o comportamento específico de cada plataforma, veja [Markdown no Obsidian](#) e [Markdown no Notion](#).

Guia completo: [Comparação das variações de Markdown](#) →

Erros comuns

Estes confundem os autores o tempo todo. Evite-os.

1. Esquecer linhas em branco

O markdown muitas vezes precisa de uma linha em branco entre os elementos:

```
Este é um parágrafo.  
- Esta lista pode não renderizar  
- dependendo do renderizador.
```

Corrija com uma linha em branco:

```
Este é um parágrafo.  
  
- Esta lista renderiza corretamente.  
- Todos os renderizadores concordam.
```

2. Misturar recuos

Ao aninhar listas ou adicionar parágrafos a itens de lista, use recuo consistente (2 ou 4 espaços). Misturar tabs e espaços quebra a renderização.

3. Espaços no final para quebras de linha

O truque dos "dois espaços no final da linha" para quebras de linha suaves é invisível e quebra com facilidade. Use uma barra invertida em vez disso, ou simplesmente aceite que parágrafos são normalmente o que você quer.

4. Não escapar caracteres especiais

Escrever `func_name_with_underscores` em texto é seguro no CommonMark e no GFM, que ignoram sublinhados entre letras, embora alguns parsers mais antigos ainda os leiam como ênfase. Os asteriscos são mais arriscados aqui, já que `a*b*c` fica em itálico. Para ser seguro em todo lugar, envolva termos técnicos em crases para código inline. Outras colisões comuns: `<` e `>` sendo lidos como tags HTML, e `|` dentro de células de tabela quebrando o layout da tabela.

5. Depender de recursos não padrão

Se você usar wikilinks do Obsidian `[[note]]` e depois publicar em um site estático que não os suporta, seus links quebram. Conheça seu renderizador de destino.

6. Tabelas largas demais

As tabelas GFM não podem rolar horizontalmente. Se uma tabela é larga demais para a página, ela transborda. Divida tabelas largas em várias mais estreitas ou use HTML.

7. Cabeçalhos vazios ou apenas com formatação

```
## **Cabeçalho em negrito**
```

Isso funciona, mas alguns renderizadores removem a formatação dos cabeçalhos ou geram IDs de âncora estranhos. Mantenha os cabeçalhos simples.

Baixe o cheat sheet em PDF

Quer este cheat sheet offline? [Baixe a versão em PDF para imprimir](#) para deixar na sua mesa ou compartilhar com sua equipe.

Perguntas frequentes

O que é um cheat sheet de markdown?

Um cheat sheet de markdown é uma referência rápida que lista cada elemento da sintaxe markdown ao lado de um exemplo de como escrevê-lo, cabeçalhos, texto em negrito e itálico, listas, links, imagens, blocos de código, tabelas e muito mais. Esta página cobre todo o CommonMark, além das extensões do GitHub Flavored Markdown.

Como deixo o texto em negrito no markdown?

Envolva o texto em dois asteriscos: `**texto em negrito**`. Dois sublinhados (`__texto em negrito__`) também funcionam. Para itálico, use um único asterisco ou sublinhado; para negrito e itálico juntos, use três: `***texto***`.

Como crio uma tabela no markdown?

Separe as colunas com barras verticais e adicione uma linha de traços abaixo da linha de cabeçalho para marcá-la como tabela:

```
Nome	Função
Ada	Líder
```

Ou pule a sintaxe por completo e monte uma visualmente com nosso [gerador de tabelas markdown](#) gratuito.

Qual é a diferença entre Markdown e GitHub Flavored Markdown?

O CommonMark é a sintaxe central padronizada. O GitHub Flavored Markdown (GFM) o estende com tabelas, listas de tarefas, tachado, autolinks, notas de rodapé e alertas. O GFM é a variação com suporte mais amplo, então é o padrão mais seguro em 2026.

Markdown é o mesmo que HTML?

Não. O Markdown é uma sintaxe leve em texto puro que se converte em HTML. O Markdown é para escrever e editar; o HTML é para exibir. Nosso guia [Markdown vs HTML](#) explica quando usar cada um.

Como adiciono um link no markdown?

Coloque o texto visível entre colchetes seguido da URL entre parênteses: `[Markdific](https://markdific.com)`. Para transformar uma imagem em link, envolva a sintaxe de imagem nos mesmos colchetes.

Como adiciono um comentário no markdown?

O Markdown não tem sintaxe oficial de comentário. O método mais portátil é um link de referência vazio, `[//]: # (sua nota)`, que não renderiza nada. Em qualquer renderizador que permita HTML, um comentário HTML `<!-- sua nota -->` também funciona.

Como crio uma checklist no markdown?

Use uma lista de tarefas: comece cada item com `- [ ]` para uma tarefa em aberto ou `- [x]` para uma concluída. As listas de tarefas são um recurso do GitHub Flavored Markdown com suporte no GitHub, GitLab e na maioria dos editores modernos.

Como adiciono uma imagem no markdown?

Use um ponto de exclamação, o texto alternativo entre colchetes e o caminho entre parênteses: `![Texto alternativo](path/to/image.jpg)`. Para dimensionar uma imagem, use uma tag HTML `<img>` com um atributo `width`.

Como escrevo um bloco de código no markdown?

Envolva o código em três crases em suas próprias linhas. Adicione o nome de uma linguagem após as crases de abertura para realce de sintaxe. Para um trecho curto dentro de uma frase, use crases simples ao redor do `código`.

Como faço um índice (sumário) no markdown?

O Markdown não tem um índice embutido, mas você pode montar um a partir de links âncora como `[Título da seção](#titulo-da-secao)`, usando o cabeçalho em minúsculas com espaços substituídos por hifens. GitHub, GitLab e muitos editores também geram um automaticamente.

Como converto Markdown para Word ou PDF?

Use um conversor. O Markdific tem ferramentas online gratuitas para converter Markdown para [Word](#), [PDF](#), [HTML](#) e mais, todas rodando no seu navegador sem cadastro.

Guias e ferramentas relacionados

Markdown por plataforma. Veja exatamente como o Markdown se comporta nas ferramentas que você usa todos os dias:

- [Markdown no GitHub](#)
- [Markdown no Obsidian](#)
- [Markdown no Notion](#)
- [Markdown no VS Code](#)
- [Markdown no Discord](#)
- [Markdown no Slack](#)
- [Markdown no Jupyter](#)
- [Markdown no Confluence](#)
- [Como escrever um README](#)
- [Arquivos Markdown para agentes de IA](#)

Converta seu Markdown. Conversores online gratuitos que rodam no seu navegador, sem cadastro:

- [Markdown para Word](#)
 - [Markdown para PDF](#)
 - [Markdown para HTML](#)
 - [Markdown para Texto](#)
 - [Markdown para Imagem](#)
 - [Markdown para Google Docs](#)
-

Conclusão

Essa é toda a sintaxe de markdown que você vai encontrar de fato em 2026. Os pontos principais:

- **CommonMark + GFM** cobre 95% do uso real de markdown
- **Tabelas, listas de tarefas, notas de rodapé, matemática, mermaid e alertas** são as extensões que vale a pena memorizar
- **Alternativas em HTML** resolvem os casos extremos que o markdown não cobre
- **Frontmatter** é essencial para qualquer site estático ou sistema de conteúdo
- **Diagramas Mermaid** são um dos recursos mais subutilizados do markdown

O formato vai continuar evoluindo à medida que o conteúdo gerado por IA inunda a web com arquivos .md, mas a sintaxe central está estável desde 2004 e não vai mudar. Aprenda uma vez e ela compensa por décadas.

Quer ver seu markdown renderizado com clareza? O [Markdific](#) é um visualizador e editor de markdown rápido e dedicado para Mac e Windows. Para saber mais, leia [Markdown vs HTML](#), navegue pelos nossos [recursos de markdown](#), ou leia a [documentação completa do Markdown](#).

Markdific

O Markdific transforma markdown puro em documentos limpos e legíveis em qualquer dispositivo. Abra qualquer arquivo `.md` e veja instantaneamente cabeçalhos, tabelas, código, matemática e imagens adequados, depois edite com salvamento automático, troque de tema e exporte para PDF, Word ou HTML.



O Markdific é um visualizador e editor de Markdown leve que abre qualquer arquivo .md e o renderiza instantaneamente, transformando Markdown puro em cabeçalhos, tabelas, código, matemática e imagens limpos e formatados. Edite com salvamento automático, troque de tema e exporte para PDF, Word ou HTML.

Baixar o Markdific