Mapper Idea®

FREEMAPPER 1.0.23 · GPL V3 OU POSTERIOR

Guia de uso do Freemapper

Tudo o que o Freemapper tem a mais que o FreeMind: mapas `.mi`, mapas de gerador, interface nova e o assistente de IA, com o caminho para cada função.

Sumário

01 Novidades da 1.0.23
02 Novidades da 1.0.22
03 Novidades da 1.0.21
04 Novidades da 1.0.20
05 Resumo dos atalhos
06 Como executar
07 Mapas .mi
08 Mapas de gerador
09 Interface
10 Assistente de IA (Claude Code)
11 Limitações conhecidas
12 Licença

O Freemapper é uma versão do FreeMind 1.0.1 adaptada para editar mapas do **Mapper Idea**: mapas de negócio e mapas de arquitetura (geradores), no formato de texto `.mi`. Ele continua abrindo e salvando mapas `.mm` como o FreeMind original.

Este guia mostra tudo o que o Freemapper tem a mais que o FreeMind, com o caminho para acessar cada função. Os menus aparecem em português, com o nome em inglês entre parênteses quando a interface está nesse idioma. Dentro do programa, este guia abre por Ajuda (Help) → Guia de uso do Freemapper (Freemapper User Guide).

01 Novidades da 1.0.23

Novidade	Onde está	Detalhes
Correção: com a interface em português, textos acrescentados pelo Freemapper apareciam com os acentos trocados ("MemÃ³ria" em vez de "Memória", "PadrÃ£o" em vez de "Padrão")	Preferências, painel do assistente e outros pontos da interface	
Correção: ao destacar o painel do assistente com a janela principal maximizada num monitor só, a janela nova abria fora da tela e não dava para recolocar o painel	Botão de destacar, no topo do painel do assistente	A janela destacada agora sempre cabe num monitor; o `Alt` + `9` a traz para a frente
Ajuste: no Windows, o `Ctrl` + `C` digitado no terminal do painel do assistente sempre chega ao Claude Code (interromper, limpar a linha, sair)	Painel do assistente	Copiar e colar no terminal
A janela **Ajuda → Licença** mostra a licença do Freemapper (GPL versão 3 ou posterior)	Menu Ajuda	Licença

02 Novidades da 1.0.22

Novidade	Onde está	Detalhes
Terminal do painel do assistente refeito com código próprio, sem a biblioteca pty4j, cuja licença não podia ir junto do Freemapper. O uso não muda	Painel do assistente (`Alt` + `9`)	No Windows, exige o Windows 10 versão 1809 (de 2018) ou mais novo. No macOS o painel ainda não funciona
Licença: o Freemapper passa a ser distribuído sob a GPL versão 3 ou posterior. O pacote traz `LICENCA.txt`, `GPL-3.0.txt` e `AVISOS-DE-TERCEIROS.txt`, e o código-fonte vai em `freemapper-1.0.22-src.zip`	Pasta do programa	Licença

03 Novidades da 1.0.21

Novidade	Onde está	Detalhes
Diagramas Mermaid e código colorido nos documentos Markdown, abertos no navegador	Botão **Abrir no navegador** no topo do documento, ou o link **Ver diagrama no navegador** embaixo de cada diagrama	Documentos Markdown
Correção: ao editar um nó com `F2` fora do zoom 100%, o cursor e a seleção não acompanhavam as letras (o cursor parecia uma letra antes do fim)	Edição direta no nó	

Novidade	Onde está	Detalhes
Correção: ao mudar o zoom, o mapa "pulava" e perdia o nó em que se estava trabalhando; agora o nó selecionado fica parado na tela	Zoom pelos botões, `Ctrl` + roda do mouse ou caixa de zoom	

04 Novidades da 1.0.20

Novidade	Onde está	Detalhes
Memória máxima nas preferências, sem editar arquivo	Preferências → Ambiente → Memória	Memória para mapas grandes
Perfis do assistente: uma conta do Claude por cliente ou projeto	Painel do assistente, tela de início de cada aba	Perfis
Leitor de Markdown, em aba ou em janela própria para outro monitor	Painel de arquivos, `Ctrl` + `P` ou Arquivo → Abrir	Documentos Markdown
Testador e validador de XPath dos geradores	`Ctrl` + `Alt` + `X`	XPath dos geradores
Texto `.mi` ao lado do mapa, sincronizado com ele	`Alt` + `8`	Texto `.mi` ao lado do mapa
Ferramentas novas para o assistente: abrir documento, mostrar o texto `.mi`, validar e testar XPath	Automático, com a integração ligada	Como o assistente trabalha com os mapas

05 Resumo dos atalhos

Atalho	Função	Detalhes
`Alt` + `Shift` + `J`	Ir para a declaração de um pattern ou mode, e voltar	Ir para a declaração e voltar
`Alt` + `Shift` + `G`	Criar os filhos esperados de um nó do Mapper Idea	Completar a estrutura
`Alt` + `Shift` + `C`	Copiar o ramo selecionado como texto `.mi`	Exportar para `.mi`
`Ctrl` + `V`	Colar texto `.mi` como nós	Colar texto `.mi`
`Ctrl` + `Shift` + `T`	Buscar uma aba pelo nome	Abas dos mapas
`Ctrl` + `Page Up` / `Ctrl` + `Page Down`	Mapa anterior / próximo (do FreeMind)	Abas dos mapas
`Ctrl` + `P`	Ir para um mapa do projeto ou um nó, pelo nome	Busca rápida

Atalho	Função	Detalhes
`Alt` + `0`	Mostrar ou esconder os arquivos do projeto	Arquivos do projeto
`Alt` + `9`	Mostrar ou esconder o assistente de IA	Painel do assistente
`Alt` + `8`	Mostrar ou esconder o texto `.mi` do mapa, ao lado dele	Texto `.mi` ao lado do mapa
`Ctrl` + `Alt` + `X`	Testar e validar o XPath dos geradores	XPath dos geradores
`Alt` + `I`, depois `/`	Buscar um ícone pelo nome	Seleção de ícones
`Alt` + `Enter`	Editor de nó longo	Editor de nó longo
`Ctrl` + `F` / `Ctrl` + `H`	Localizar e substituir, dentro do editor de nó longo	Editor de nó longo
`Ctrl` + `Z` / `Ctrl` + `Y`	Desfazer e refazer, dentro do editor de nó longo	Editor de nó longo

06 Como executar

Há dois tipos de pacote:

Pacote	Precisa de Java instalado?	Como abrir
`freemapper-<versão>-linux-x64.zip`	Não, o Java 21 vem dentro	`bash freemind.sh`
`freemapper-<versão>-windows-x64.zip`	Não, o Java 21 vem dentro	duplo clique em `Freemapper.exe`
`freemapper-<versão>.zip` (qualquer sistema)	Sim, Java 8 ou mais novo	Linux e macOS: `bash freemind.sh`; Windows: `Freemapper.exe`

Descompacte o pacote em uma pasta sua e abra pelo arquivo indicado. No Windows, se o `Freemapper.exe` não abrir, `freemind.bat` abre o programa com uma janela de console que mostra as mensagens de erro (`freemindw.bat` faz o mesmo, sem console).

Os lançadores usam o Java da pasta `jre`, ao lado deles, quando ela existe; senão, o Java instalado no sistema.

O assistente de IA precisa do Java 11 ou mais novo. Os dois pacotes com Java embutido já atendem; no pacote sem Java, vale o Java instalado no sistema.

Aviso do Windows na primeira vez: o `Freemapper.exe` ainda não tem assinatura digital, então o Windows mostra "O Windows protegeu o computador", com fornecedor desconhecido. Clique em "Mais informações" (se o botão não estiver à vista) e em "Executar assim mesmo". Depois disso o aviso não volta nessa máquina.

Atalho

- **Linux:** execute `bash instalar-atalho.sh` uma vez. O Freemapper passa a aparecer no menu de aplicativos.
- **Windows:** dê duplo clique em `criar-atalho.bat`. Ele cria o atalho "Freemapper" na área de trabalho.

O atalho aponta para a pasta em que o Freemapper está. Se mudar a pasta de lugar, crie o atalho de novo.

Memória para mapas grandes

O limite padrão de memória é 256 MB, o mesmo do FreeMind. Mapas com centenas de milhares de nós precisam de mais.

Como mudar: Extras (Tools) → Preferências (Preferences) → Ambiente (Environment) → Memória → "Memória máxima (MB)". O valor vale a partir da próxima abertura do Freemapper e fica guardado nas preferências do usuário, então continua valendo ao trocar de versão.

- No Linux e no macOS, o `freemind.sh` lê o valor das preferências. A variável `FREEMAPPER_XMX` (por exemplo `export FREEMAPPER_XMX=4096M`), se definida, tem precedência.
- No Windows, o Freemapper grava o valor no `Freemapper.l4j.ini`, ao lado do `Freemapper.exe`. Se a pasta do programa não puder ser gravada, ele avisa, e o arquivo pode ser editado à mão (linha `-Xmx4096M`). Para os `.bat`, vale a variável de ambiente `FREEMAPPER_XMX`.

07 Mapas `.mi`

O formato

Um arquivo `.mi` é texto puro. A indentação (4 espaços) define a hierarquia, e os ícones vão entre colchetes no começo da linha.

```
[p] domain
    [b] Cliente
        @
            [e] description
                [v] Pessoa que compra na loja.
                  | Pode ser pessoa física ou jurídica.
        [g] atributos
            [d] nome: Texto(128)
[e] generators
    [e] createTable
        #
            generators/sql-createTable.mi
```

Sintaxe	Significado
`[c] texto`	Nó com ícone, pelo atalho
`[x, y] texto`	Nó com vários ícones
`[Descriptor.window.editor] texto`	Ícone pelo nome completo
`| continuação`	Linha a mais do nó anterior (texto com várias linhas)
`#` e, na linha seguinte, um caminho	Link do nó pai para outro arquivo

Em nós `[v]` e `[y]`, os espaços no começo e no fim do texto são preservados. Colchetes que não correspondem a ícones existentes, como `[TODO] revisar`, ficam como texto.

Atalhos de ícone:

Atalho	Ícone	Atalho	Ícone
`b`	Descriptor.bean	`m`	Mapping.manyToOne
`c`	Descriptor.class	`x`	Method.public
`g`	Descriptor.grouping	`e`	element
`p`	Package	`v`	tag_green
`d`	Mapping.directToField	`t`	textNode
`r`	Mapping.oneToOne	`k`	bullet_key
`o`	Mapping.oneToMany	`y`	tag_yellow
`h`	Mapping.directMap		

Abrir e salvar

> **Como acessar:** Arquivo (File) → Abrir, e escolha um arquivo `.mi`. Para criar um `.mi` a partir de um mapa `.mm`, use Arquivo → Salvar como e termine o nome com `.mi`.

- O mapa abre **fechado**: só a raiz e o primeiro nível ficam visíveis. Isso mantém a abertura rápida em mapas grandes.
- `Ctrl` + `S` grava o `.mi` como texto.
- Um arquivo com mais de uma raiz ganha na tela uma raiz com o nome do arquivo, que não é gravada.
- Cada nó recebe como ID a linha em que começa no arquivo.

Estado visual por usuário

O `.mi` guarda só estrutura, texto, ícones e links. O que é visual fica fora dele, numa pasta por usuário ao lado do mapa:

```
<pasta do mapa>/.freemapper/<usuario>/<mapa>.mi.state
<pasta do mapa>/.freemapper/<usuario>/<mapa>.mi.base
```

Como usar: não há nada a acionar. Abra e feche nós, mude cores e fontes normalmente.

- **Dobrar e desdobrar** não marca o mapa como alterado. O estado é gravado sozinho ao fechar o mapa ou o programa.
- **Cores, fontes e outros estilos** marcam o mapa como alterado. Salve com `Ctrl` + `S`: só o arquivo de estado é gravado, e o `.mi` fica intocado.
- **Ao reabrir**, o mapa volta com os mesmos nós abertos, os estilos e o último nó selecionado.
- **Se o `.mi` mudar fora do Freemapper** (editor de texto, git, IA), o estado acompanha os nós na abertura seguinte. Nós novos entram fechados.

O que vai para o estado: nós abertos, cores, fontes, arestas, nuvens, lado do nó, setas entre nós, notas e atributos. Notas e atributos, por ficarem no estado do usuário, não são vistos por outras pessoas.

Nome do usuário: é o login do sistema. Para usar outro nome, feche o Freemapper e acrescente a linha `mi_state_user=nome` ao arquivo `~/.freemind/auto.properties`.

Git: cada projeto decide se versiona a pasta `.freemapper/` ou a coloca no `.gitignore`.

CLI do Mapper Idea: o `push` envia tudo o que está na pasta do projeto, inclusive a `.freemapper/`. Ela não interfere no `generate`.

Gerar código a partir do `.mi`

Os mapas `.mi` gravados pelo Freemapper são lidos direto pelo engine do Mapper Idea, sem conversão para `.mm`. Com o CLI (`mapperidea`, ou `mi`):

```
cd pasta-do-main.mi
mi init meu-projeto main.mi        # uma vez por projeto
mi push meu-projeto                # a cada alteração nos mapas
mi generate meu-projeto sql createTable className=Pedido
```

O ciclo é: editar o mapa no Freemapper, salvar (`Ctrl`+`S`), `push` e `generate`. Um erro `EMI1003 ... Content is not allowed in prolog` no `generate` indica um servidor cujo engine ainda só lê `.mm`.

Mapa alterado no disco

O Freemapper confere a cada 5 segundos se o arquivo do mapa ativo mudou no disco (editado em outro programa, por um `git pull` ou por uma ferramenta).

- **Mapa `.mi` sem alterações por salvar:** é recarregado sozinho. A aba fica no mesmo lugar, o nó selecionado continua selecionado (pela linha, ajustada ao que mudou) e os nós abertos continuam abertos. A barra de status avisa.
- **Mapa com alterações por salvar:** o Freemapper pergunta se deve recarregar. Recarregar descarta as alterações do mapa aberto; recusar mantém o mapa como está, e ao salvar ele substitui a versão do disco.

Se você está digitando em um nó, a recarga espera a edição terminar. Para recarregar na hora: Arquivo (File) → Reverter (Revert).

Texto `.mi` ao lado do mapa (`Alt`+`8`)

> **Como acessar:** `Alt`+`8`, ou Visualizar (View) → Texto .mi do mapa (.mi Text of the Map). O painel começa fechado.

Mostra, à direita do mapa, o texto `.mi` que o representa, com os números das linhas, e mantém os dois lados sincronizados:

- **Do mapa para o texto:** selecionar um nó destaca as linhas dele no texto e rola até lá. Alterar o mapa refaz o texto em seguida.
- **Do texto para o mapa:** clicar em uma linha, ou andar com as setas, seleciona o nó dela. Se o nó está dentro de um ramo fechado, é selecionado o nó fechado mais acima, sem abrir nada; duplo clique na linha abre o caminho até o nó.

O que saber:

- O painel é só para leitura. Para alterar, edite o mapa; para editar o texto, use um editor externo, e o Freemapper recarrega o arquivo sozinho (veja Mapa alterado no disco).
- O texto é o do mapa como está agora em memória, com o que ainda não foi salvo: é o que seria gravado. Com alterações por salvar, a linha do painel pode ser diferente da linha da barra de status, que é a do arquivo em disco.
- Em um mapa `.mm`, o painel mostra como o mapa ficaria em `.mi`.
- Acima de 20 mil linhas o texto aparece sem cores. A largura muda arrastando a borda esquerda do painel.

Colar texto `.mi` (`Ctrl` + `V`)

Como usar: copie um trecho de texto no formato `.mi` de qualquer lugar, selecione o nó que será o pai e pressione `Ctrl` + `V`. Os nós são criados com hierarquia, ícones, textos de várias linhas e links.

Endereços `http` e e-mails colados não viram link automaticamente.

Exportar para `.mi`

O quê	Como acessar
Mapa inteiro, para um arquivo	Arquivo (File) → Exportar (Export) → As MI...
Ramo selecionado, para um arquivo	Arquivo → Exportar → Branch as MI, ou botão direito no nó → Exportar → Branch as MI
Ramo selecionado, para a área de transferência	`Alt` + `Shift` + `C`, ou Exportar → Branch as MI to Clipboard

Dica: `Alt` + `Shift` + `C` seguido de `Ctrl` + `V` em outro nó duplica um ramo, inclusive entre mapas. A cópia leva o texto `.mi`; cores e fontes não vão junto.

08 Mapas de gerador

Ir para a declaração e voltar (`Alt` + `Shift` + `J`)

Para que serve: em um gerador, pular do lugar onde um pattern ou um template é usado para o lugar onde ele é declarado, e voltar.

> **Como acessar:** `Alt` + `Shift` + `J`, ou Extras (Tools) → "Mapper Idea: Ir para a declaração / Voltar".

Como usar:

1. Selecione uma **referência**:
   - o nome do pattern, filho de `write-pattern`; ou
   - o nome do mode, em `apply-templates` → `mode` (também vale selecionar o `mode` ou o próprio `apply-templates`).
2. Pressione `Alt` + `Shift` + `J`. O mapa vai para a **declaração**:
   - do pattern, dentro de `patterns`, ignorando agrupadores com o mesmo nome;

- do mode, dentro de `templates` → `mode`.

3. Pressione `Alt` + `Shift` + `J` de novo, na declaração ou em qualquer nó que não seja uma referência, para **voltar** ao nó de origem.

Outros comportamentos:

- Se o nó selecionado for outra referência, o atalho faz uma busca nova a partir dela, em vez de voltar.
- As origens ficam empilhadas. Dá para ir de um `apply-templates` ao template, dali a um pattern, e voltar os dois passos.
- Com vários geradores no mesmo mapa, vale a declaração do gerador em que o nó está.
- Em um nó comum, sem nada para voltar, o atalho guarda a posição atual. O próximo `Alt` + `Shift` + `J` em outro nó comum volta para ela.
- As mensagens aparecem na barra de status, na parte de baixo da janela.

Completar a estrutura do Mapper Idea (`Alt` + `Shift` + `G`)

Para que serve: criar os filhos esperados de um nó, sem digitar.

> **Como acessar:** `Alt` + `Shift` + `G`, ou Extras (Tools) → "Mapper Idea: Completar a estrutura".

Como usar: selecione o nó e pressione o atalho. Filhos que já existem não são duplicados.

Nó selecionado	Filhos criados
`config`	`mapperidea`
`mapperidea`	`generators`, `injectables`, `maps`, `fragments`
`generators`	`meuGerador` (renomeie)
um gerador (filho de `generators`)	`parameters`, `vars`, `start`, `templates`, `patterns`
`parameters`	`className` com o valor `NOT_DEFINED`
`start` ou `template`	`match`, `body`
`match`	`classes/class`
`body`, `then` ou `else`	`write-pattern`, `apply-templates`, `if`, `vars`
`apply-templates`	`select`, `mode`, `parameters`
`select`	`attributes/attribute`
`write-pattern`	`nomeDoPadrao`
`patterns`	`nomeDoPadrao`

XPath dos geradores (`Ctrl` + `Alt` + `X`)

> **Como acessar:** `Ctrl` + `Alt` + `X`, ou Extras (Tools) → "Mapper Idea: XPath dos geradores...".

As expressões de um gerador não rodam sobre o mapa: rodam sobre o XML normalizado que o servidor monta a partir do mapa de domínio. A janela tem três abas.

Testar. Avalia uma expressão contra o XML de uma classe.

1. Escolha o projeto e a classe e clique em "Buscar no servidor". O Freemapper chama `mi generate <projeto> struct xml className=<classe>` e mostra o XML, com cores. Dá para buscar várias classes, separadas por vírgula.
2. Selecione no mapa do gerador o nó da expressão (um `[v]` de `match`, `select`, `expr`, `if`, de um fragmento, ou uma linha de pattern com `{{ }}`) e clique em "Do nó selecionado". O Freemapper traz a expressão, com os `f#[fragmentos]` já substituídos; o contexto, que é o `match` do template que a executa; e as variáveis que ela usa, tiradas de `parameters` e `vars` do gerador.
3. "Avaliar" (`Ctrl` + `Enter`) mostra o resultado para cada nó de contexto e realça no XML os nós de contexto e os nós encontrados.

Sem XML carregado, "Avaliar" só diz se a expressão está correta. Em um mapa de domínio, "Do nó selecionado" monta o caminho da classe, do atributo ou do método selecionado.

Validar gerador. Confere todas as expressões do mapa ativo de uma vez, sem precisar do servidor, e lista os erros com a linha do `.mi`. Clicar em uma linha leva ao nó.

Referência. O formato do XML, caminhos comuns, o que é o contexto, e as funções `mi:` e `functx:`.

O que é preciso e o que fica de fora:

- O projeto precisa estar registrado pelo CLI (`mi init`) e registrar o gerador `struct` no mapa principal. Sem o `struct`, o servidor não devolve nada, e a janela avisa.
- O servidor responde sobre o último `push`: alterações do mapa de domínio que ainda não foram enviadas não aparecem no XML.
- O XML traz só as classes buscadas. Caminhos que leem outras partes do projeto, como `//maps`, voltam vazios.
- O motor é o mesmo do servidor (Saxon-HE 10.1), e as expressões são avaliadas dentro de uma folha XSLT, como no gerador. As funções `mi:` foram reescritas a partir da documentação: `mi:tabulate-text` e `mi:concat-change-pattern` são aproximações.
- `position()` e `last()` contam todos os nós que o `match` seleciona; no gerador, contam o que o `apply-templates` selecionou.
- O comando do CLI pode ser trocado em Preferências → Ambiente → Mapper Idea. Vazio, vale o `mi` da pasta `bin` do usuário, se existe; senão, o do `PATH`.

O módulo fica em `lib/xpath` e só é carregado quando a janela é aberta.

09 Interface

Abas dos mapas

As abas ficam em uma linha só. Os controles estão no canto direito da barra de ferramentas:

Controle	O que faz
`◂` e `▸`	Rolam as abas. Segurar o botão continua rolando

Controle	O que faz
▼	Lista todos os mapas abertos; clique em um para ir direto a ele
Roda do mouse sobre as abas	Rola as abas
Campo **Buscar aba** (Find Tab), `Ctrl` + `Shift` + `T`	Digite parte do nome do mapa e pressione `Enter`. `Esc` limpa o campo

A aba do mapa ativo fica em negrito, com uma barra colorida embaixo.

Do FreeMind original: `Ctrl` + `Page Up` e `Ctrl` + `Page Down` trocam de mapa, e Mapas mentais (Maps) → Mova Mapa para esquerda / Mova Mapa para Direita (Move Map Left / Right), com `Ctrl` + `Shift` + `Page Up` / `Ctrl` + `Shift` + `Page Down`, muda a ordem das abas.

Se a barra de ferramentas estiver oculta, os botões e o campo de busca somem com ela.

Editor de nó longo (`Alt` + `Enter`)

> **Como acessar:** selecione um nó e pressione `Alt` + `Enter`, ou Editar (Edit) → Editar nó com texto longo.

Atalho, dentro do editor	Função
`Ctrl` + `Z` / `Ctrl` + `Y`	Desfazer / refazer
`Ctrl` + `F` ou `Ctrl` + `H`	Abrir o painel de localizar e substituir
`Enter`, no campo de busca	Ir para a próxima ocorrência

O painel mostra o número de ocorrências ("3 de 15") e tem botões de anterior e próximo, a opção de diferenciar maiúsculas, substituir e substituir tudo. As substituições podem ser desfeitas.

A caixa **Enter Confirma** (Enter Confirms), embaixo do editor, decide o que o `Enter` faz no texto:

Enter Confirma	Nova linha	Confirmar a edição
Marcada	`Shift` + `Enter`	`Enter`
Desmarcada	`Enter`	`Alt` + `Enter`

Ela começa como está em Preferências → "Enter confirma por padrão".

Seleção de ícones (`Alt` + `I`)

> **Como acessar:** selecione um nó e pressione `Alt` + `I`.

Tecla, dentro do diálogo	Função
`/`	Iniciar a busca; digite parte do nome do ícone (aceita `.`, `-` e `_`)
Setas	Navegar entre os ícones encontrados

Tecla, dentro do diálogo	Função
`Enter`	Encerrar a busca; pressione de novo para aplicar o ícone
`Esc`	Cancelar a busca
`Delete`	Remover todos os ícones do nó
`Backspace`	Remover o último ícone do nó

Os ícones aparecem ordenados por nome. Os do Mapper Idea estão incluídos: `Descriptor.*`, `Mapping.*`, `Method.*`, `Package`, `Policy.*`, `Table*` e outros.

Tema claro ou escuro

> **Como acessar:** Extras (Tools) → Preferências (Preferences) → Aparência (Appearance) → Tema (Theme).

O tema muda as cores do mapa (fundo, texto, seleção, arestas, nuvens e links). As janelas e os menus seguem o visual escolhido em "Estilo" (Look and Feel), na mesma tela; com o estilo "Moderno (acompanha o tema)", o padrão, eles também ficam claros ou escuros.

Documentos Markdown em aba

Um arquivo `.md` abre em uma aba, ao lado das abas de mapa, já formatado: títulos, listas, tarefas, tabelas, citações, blocos de código, imagens e ligações. É só para leitura.

Como abrir: duplo clique no arquivo no painel de arquivos do projeto, que também lista os `.md`, busque pelo nome com `Ctrl` + `P`, ou use Arquivo (File) → Abrir (Open), que também lista os `.md`. O assistente de IA também pode abrir um documento (ferramenta `open_document`).

- A aba acompanha o arquivo: quando ele muda no disco, a página é refeita na mesma altura.
- Uma ligação para outro `.md` ou para um mapa abre em aba; um endereço da web abre no navegador.
- O botão no canto direito do topo do documento leva-o para a janela de documentos, para deixar em outro monitor. Essa janela tem uma aba por documento, e o Freemapper lembra onde ela ficou.
- Com a janela de documentos aberta, um `.md` novo abre nela, e não na janela principal.
- Na janela de documentos, o botão do topo devolve o documento à janela principal, `Ctrl` + `W` fecha a aba e o X fecha a janela com todos os documentos dela.
- Fechar (`Ctrl` + `W`) fecha a aba do documento. A tela inicial só volta quando não há mapa nem documento aberto.
- Enquanto um documento está na tela, o mapa ativo continua sendo o último mapa usado: os comandos de mapa dos menus agem sobre ele.
- Na aba não há realce de sintaxe nem diagramas. O botão **Abrir no navegador**, no topo do documento, abre o documento no navegador do sistema com os diagramas Mermaid desenhados e o código colorido (Java, SQL, bash, JSON, XML, YAML, Python, Groovy, `.mi` e outras). Funciona sem internet: as bibliotecas vêm no pacote.
- Cada bloco ```` ```mermaid ```` aparece na aba como código, seguido do link **Ver diagrama no navegador**, que abre a página já rolada até aquele diagrama.
- A página segue o tema (claro ou escuro) do Freemapper. Atualizar a página no navegador (`F5`) mostra o arquivo como está no disco. Imagens fora da pasta do documento não aparecem nela.

- A página é entregue por um endereço local (`http://127.0.0.1:...`), que só atende neste computador e muda a cada vez que o Freemapper abre. Scripts escritos dentro do Markdown não rodam nela.

Busca rápida (`Ctrl` + `P`)

> **Como acessar:** `Ctrl` + `P`, ou Editar (Edit) → Ir para mapa ou nó (Go to Map or Node).

Um campo só para ir a qualquer lugar. Digite parte do nome e use as setas e `Enter`; `Esc` fecha.

- **Mapas**: os abertos e os da pasta do projeto. Vale um pedaço do nome, da pasta, ou as letras na ordem (`sqct` acha `sql-createTable.mi`). `Enter` abre o mapa ou troca para a aba dele.
- **Nós** do mapa ativo: aparecem com o caminho até eles (`pacote › classe › atributos`). `Enter` desdobra o caminho e seleciona o nó. Nós fechados também são encontrados.

Imprimir, que no FreeMind era `Ctrl` + `P`, passou para `Ctrl` + `Shift` + `P`.

Tela inicial

Quando não há mapa aberto, o lugar das abas mostra a tela inicial: botões para abrir um mapa, abrir a pasta do projeto ou criar um mapa novo, e a lista dos mapas recentes (um clique abre). Ela some ao abrir um mapa ou um documento e volta quando não resta nenhum aberto.

Com a opção de restaurar a sessão ligada (o padrão), o Freemapper reabre os mapas da última vez, e a tela inicial só aparece se não houver nenhum.

Arquivos do projeto (`Alt` + `0`)

> **Como acessar:** `Alt` + `0`, o primeiro botão da barra de ferramentas, ou Visualizar (View) → Arquivos do projeto (Project Files).

Painel à esquerda com as pastas, os mapas (`.mi` e `.mm`) e os documentos Markdown (`.md`) do projeto. Duplo clique ou `Enter` abre o arquivo em uma aba; se ele já está aberto, só troca de aba. Os arquivos abertos aparecem em negrito e o mapa ativo fica selecionado.

A pasta do projeto é a do `main.mi`: ao abrir um mapa, o Freemapper sobe pelas pastas até achar uma com `main.mi`. Para escolher outra, use Arquivo (File) → Abrir pasta do projeto, ou o botão de pasta no topo do painel. Pastas ocultas e pastas sem mapas nem documentos não são listadas. A largura muda arrastando a borda direita do painel.

Com um mapa aberto, o painel fica abaixo da faixa de abas, ao lado do mapa, e as abas ocupam toda a largura da janela. O mesmo vale para o painel do assistente, à direita.

Barra de status

No rodapé da janela, à direita das mensagens do programa:

- **Linha**: a linha do nó selecionado no arquivo `.mi` em disco, para achar o mesmo ponto em um editor de texto. Um traço no lugar do número indica que o nó ainda não foi gravado. Só aparece em mapas `.mi`.
- **Zoom** do mapa.
- **Salvo** ou **Modificado**.

Escala da interface

> **Como acessar:** Extras (Tools) → Preferências (Preferences) → Aparência (Appearance) → Escala da interface (Interface scale).

Para telas grandes ou de alta resolução. Aumenta tudo de uma vez: menus, barras, ícones, diálogos e o zoom inicial dos mapas. Valores: 100%, 125%, 150%, 175% e 200%. Reinicie o Freemapper para aplicar.

Vale para o estilo "Moderno (acompanha o tema)", o padrão. Um mapa que já foi aberto antes volta com o zoom em que foi deixado.

10 Assistente de IA (Claude Code)

O Freemapper pode trabalhar junto com o Claude Code, o assistente de IA da Anthropic que roda em um terminal. São duas partes, que podem ser usadas juntas ou separadas:

- **Painel do assistente**: um terminal dentro do Freemapper, à direita do mapa, com o Claude Code rodando na pasta do projeto.
- **Integração com os mapas**: o assistente enxerga os mapas abertos e o nó selecionado, navega neles e, se você permitir, altera os mapas na sua frente.

O Claude Code não vem dentro do Freemapper. Você instala o programa e usa a sua própria conta.

Preparar pela primeira vez

Faça os passos abaixo uma vez por computador.

1. Tenha uma conta do Claude com acesso ao Claude Code. Servem os planos Pro, Max, Team e Enterprise, ou uma conta do Console (API). O plano gratuito do claude.ai não inclui o Claude Code.

2. Instale o Claude Code. Abra um terminal e execute o comando do seu sistema:

Sistema	Onde executar	Comando
Windows	PowerShell	`irm https://claude.ai/install.ps1 | iex`
Windows	Prompt de Comando (CMD)	`curl -fsSL https://claude.ai/install.cmd -o install.cmd && install.cmd && del install.cmd`
Linux e macOS	terminal	`curl -fsSL https://claude.ai/install.sh | bash`

No Windows não é preciso executar como administrador. Para abrir o PowerShell, pressione a tecla Windows, digite `powershell` e pressione `Enter`. Instalar o Git for Windows é opcional, mas recomendado: com ele o Claude Code consegue executar comandos de shell como no Linux.

3. Confira a instalação. Feche o terminal, abra um novo e execute:

```
claude --version
```

Deve aparecer um número de versão. Se aparecer que `claude` não foi encontrado ou não é reconhecido, a pasta de instalação ainda não está no `PATH`. No Windows, acrescente-a com este comando no PowerShell e abra um terminal novo:

```
[Environment]::SetEnvironmentVariable("Path", [Environment]::GetEnvironmentVariable("Path", "User") + ";$env:USERPROFILE\.local\bin", "User")
```

Pelas janelas do Windows, o caminho é: Configurações do sistema → Variáveis de ambiente → em "Variáveis de usuário", selecione **Path** → **Editar** → **Novo** → `%USERPROFILE%\.local\bin`. Edite a variável `Path` que já existe; uma variável nova com outro nome (por exemplo, `claude`) não funciona. Use a pasta, sem o `claude.exe` no fim. No Linux e no macOS, veja Problemas comuns.

4. Faça o login. No mesmo terminal, execute `claude`. Na primeira vez ele abre o navegador para você entrar com a sua conta. Depois que o Claude Code responder no terminal, saia com `/exit`. O login fica guardado; não é preciso repetir.

5. Ligue a integração no Freemapper. Extras (Tools) → Preferências (Preferences) → Ambiente (Environment) → Integração com IA:

Opção	O que libera
Permitir que um assistente de IA veja e navegue nos mapas abertos	O assistente lê os mapas, vê o nó selecionado, procura nós e muda a seleção
Permitir também que o assistente altere os mapas	Acrescenta criar, editar e remover nós, desfazer e salvar
Porta	Porta local usada pela integração (39909). Só troque se outro programa já a usa

As duas opções vêm desligadas. Marque a primeira e, se quiser que o assistente também altere os mapas, a segunda. **Feche e abra o Freemapper** para valer.

6. Abra o painel. Com um mapa do projeto aberto, pressione `Alt` + `9` e clique em "Iniciar o Claude Code". Na primeira vez em cada pasta, o Claude Code pergunta se você confia nela; confirme.

7. Teste. Selecione um nó no mapa e pergunte no painel:

```
Qual nó está selecionado no Freemapper?
```

O assistente deve responder com o texto do nó e o caminho até ele. Na primeira vez que usa uma ferramenta do Freemapper, o Claude Code pode pedir a sua permissão; aceite para a sessão ou para sempre.

Painel do assistente (`Alt` + `9`)

> **Como acessar:** `Alt` + `9`, o segundo botão da barra de ferramentas (o das estrelinhas), ou Visualizar (View) → Assistente de IA (AI Assistant). O `Alt` + `9` também esconde o painel quando o foco está no terminal.

O painel sempre começa fechado, e abri-lo não inicia nada: ele mostra a pasta em que vai trabalhar e o botão "Iniciar o Claude Code". Só depois do clique a sessão começa. Esconder o painel não encerra a sessão.

- **Pasta de trabalho:** a pasta do projeto do mapa ativo, a mesma do painel de arquivos do projeto.

- **Uma aba por projeto.** Cada aba é uma sessão independente do Claude Code e leva o nome da pasta do projeto. Fechar a aba encerra a sessão dela.
- **A aba do projeto do mapa ativo fica marcada com um ponto (●).** O painel não troca de aba sozinho ao mudar de mapa, para não interromper uma conversa em andamento.
- **Largura:** arraste a borda esquerda do painel.

Botões no topo do painel:

Botão	O que faz
`+`	Abre a aba do projeto do mapa ativo, ou vai para ela se já existe
Janela	Leva o assistente para uma janela própria, para usar em outro monitor, e o traz de volta. As sessões continuam as mesmas. Fechar a janela própria devolve o assistente ao painel
Recarregar	Inicia uma sessão nova na aba selecionada

Copiar e colar no terminal do painel seguem os atalhos do Claude Code, que mudam conforme o sistema:

Tecla	Linux	Windows
`Ctrl` + `C`	Interrompe o que o Claude está fazendo; com o campo vazio, um segundo `Ctrl` + `C` sai	O mesmo; com texto selecionado, copia
`Ctrl` + `V`	Cola	Não faz nada (no Windows, colar no Claude Code é `Alt` + `V`)
`Alt` + `V`	—	Cola (texto ou imagem)
`Ctrl` + `Shift` + `C` / `Ctrl` + `Shift` + `V`	Copia o texto selecionado / cola texto	Copia o texto selecionado / cola texto

O botão direito do mouse sobre o terminal também abre um menu com Copiar e Colar.

Se a integração com os mapas estiver ligada, a sessão do painel já nasce conectada a este Freemapper, sem nenhum registro manual.

Perfis (mais de uma conta do Claude)

Para quem usa mais de uma conta do Claude, por exemplo uma por cliente. Na tela de início de cada aba do painel há o seletor **Perfil** e o botão **Perfis...**.

Cadastrar: clique em "Perfis...", depois em "Adicionar", e preencha o nome e o comando. Exemplos:

Nome	Comando
Surecred	`HOME=$HOME/ai_profiles/surecred claude`
Subbase	`claude-subbase` (um alias do seu shell)

Usar: escolha o perfil no seletor e clique em "Iniciar o Claude Code".

- O comando pode definir variáveis antes do programa ou ser um alias: ele é executado pelo seu shell, com o seu perfil de shell carregado. No Windows, use a forma `set "VARIAVEL=valor" && claude`.

- O perfil escolhido fica guardado para a pasta do projeto e vale na próxima sessão iniciada nela. Outro projeto pode usar outro perfil.
- Enquanto a sessão está aberta, o nome do perfil aparece na aba, depois do nome do projeto.
- "Padrão" é o comando das preferências: Preferências → Ambiente → Integração com IA → "Comando do assistente" (de fábrica, `claude`).
- Cada perfil tem as próprias configurações do Claude Code: login, plugins e skills precisam estar feitos em cada um.
- Os perfis ficam em `ai-profiles.properties`, na pasta `.freemind` da sua pasta de usuário.

Como o assistente trabalha com os mapas

Para **escrever conteúdo** em um mapa `.mi`, o assistente edita o próprio arquivo e pede ao Freemapper para recarregar o mapa. O mapa recarrega na hora e o primeiro nó novo ou alterado aparece selecionado no centro da tela. Se o projeto tem a skill do Mapper Idea instalada no Claude Code, ela é usada para escrever o `.mi`.

Para **ajustes pequenos**, para mapas `.mm` e para quando o mapa tem alterações por salvar, o assistente usa as ferramentas de alteração. Elas acontecem no mapa aberto, na sua frente, e cada uma pode ser desfeita com `Ctrl` + `Z`. Nada vai para o disco até o assistente salvar ou você pressionar `Ctrl` + `S`.

Exemplos de pedidos:

```
Me explique o gerador que está selecionado.
Crie a classe Pedido no pacote que está selecionado, com cliente, data e itens.
Ache onde o pattern cabecalho é declarado e me mostre.
```

Ver e navegar (primeira opção das preferências):

Ferramenta	O que faz
`list_maps`	Lista os mapas abertos, o arquivo de cada um e qual está ativo
`get_map`	Devolve o mapa, ou um ramo, em formato `.mi`, com o identificador de cada nó, a linha no arquivo e se está fechado ou selecionado
`get_selection`	Devolve o nó selecionado, o caminho até ele e os filhos
`find_nodes`	Acha os nós cujo texto contém um trecho, inclusive os fechados
`screenshot`	Devolve uma imagem do mapa como está na tela
`select_node`	Seleciona um nó e o traz para o centro da tela, abrindo o caminho até ele
`set_folded`	Fecha ou abre um nó
`reload_map`	Recarrega um `.mi` que foi editado fora do Freemapper e mostra o primeiro nó alterado. Recusa se o mapa tiver alterações por salvar
`open_map`	Abre um arquivo de mapa, ou troca para um mapa já aberto
`open_document`	Mostra um arquivo Markdown formatado, em aba
`show_mi_text`	Mostra ou esconde o painel com o texto `.mi` do mapa ativo

Ferramenta	O que faz
`validate_generator`	Confere todas as expressões XPath de um mapa de gerador, sem o servidor, e devolve os erros com o nó e a linha
`get_class_xml`	Busca no servidor o XML normalizado de uma ou mais classes (pelo gerador `struct`)
`xpath_evaluate`	Avalia uma expressão contra esse XML, a partir de um nó do gerador ou de uma expressão informada

As três últimas fazem o mesmo que a janela XPath dos geradores. O assistente pode abrir essa janela já preenchida, para você acompanhar o teste na tela. Exemplos de pedidos:

```
Valide o gerador que está aberto e me mostre os erros.
Teste a expressão do nó selecionado contra a classe Produto e abra a janela para eu ver.
Abra o roteiro.md ao lado e mostre onde este nó fica no texto .mi.
```

Alterar (segunda opção das preferências):

Ferramenta	O que faz
`add_nodes`	Cria nós novos sob um nó, a partir de um texto `.mi` (como o `Ctrl` + `V`)
`set_node_text`	Troca o texto de um nó
`set_node_icons`	Troca os ícones de um nó, pelos atalhos do `.mi` ou pelo nome
`delete_node`	Remove um nó e o que está abaixo dele
`undo`	Desfaz a última alteração, como o `Ctrl` + `Z`
`save_map`	Salva o mapa, como o `Ctrl` + `S`

Usar o Claude Code em um terminal à parte

O painel é opcional. Para ligar aos mapas um Claude Code aberto em outro terminal:

1. Ligue a integração nas preferências e reinicie o Freemapper (passo 5 de Preparar pela primeira vez).
2. Extras (Tools) → "Integração com IA: copiar o comando do Claude Code".
3. Cole o comando em um terminal, uma vez. Ele registra o Freemapper no Claude Code. Numa sessão nova, as ferramentas já aparecem.

Se a porta for trocada nas preferências, copie e cole o comando de novo.

Segurança e privacidade

- As duas opções da integração vêm desligadas. Sem a segunda, as ferramentas de alteração nem existem.
- A integração só atende conexões do próprio computador, e cada pedido precisa de um código secreto gerado pelo Freemapper, guardado em `~/.freemind/auto.properties`. Páginas abertas no navegador são recusadas.

- O que o assistente lê dos mapas é enviado à Anthropic, nos termos da sua conta do Claude. Em mapas com conteúdo sigiloso, confira antes as regras da sua empresa.
- O painel não guarda senha nem login. Ele só executa o `claude` instalado no seu computador.

Problemas comuns

O que aparece	O que fazer
No terminal, `claude` não é encontrado ou não é reconhecido	Abra um terminal novo. Se continuar, a pasta de instalação não está no `PATH`: no Windows é `%USERPROFILE%\.local\bin`; no Linux e no macOS, `~/.local/bin`. Acrescente-a ao `PATH` (no Windows, veja o comando em Preparar pela primeira vez) e abra o terminal de novo
O terminal acha o `claude`, mas o painel do Freemapper não	Feche e abra o Freemapper. No Windows, se continuar, saia da sua sessão do Windows e entre de novo, para o `PATH` novo valer
"O painel do assistente precisa do Java 11 ou mais novo"	O Freemapper está rodando em Java 8. Use um pacote com Java embutido, ou instale um Java mais novo
"Esta instalação não inclui o módulo de terminal (lib/terminal)"	A pasta `lib/terminal` não está na instalação. Descompacte o pacote de novo
"A integração com IA não pôde ser ligada: a porta … já está em uso"	Outro programa, ou outro Freemapper aberto, usa a porta. Troque a porta nas preferências e reinicie
O assistente diz que não tem ferramentas do Freemapper	A integração está desligada, ou foi ligada sem reiniciar o Freemapper. Depois de reiniciar, comece uma sessão nova no painel (botão de recarregar)
O assistente não consegue alterar o mapa	A segunda opção das preferências está desmarcada
A sessão pede login toda vez	Faça o login uma vez em um terminal comum (passo 4)

Para usar outro comando no lugar de `claude` (um caminho completo, por exemplo): Preferências → Ambiente → Integração com IA → "Comando do assistente". Para mais de uma conta, veja Perfis.

11 Limitações conhecidas

- No `.mi`, um nó cujo texto é só `#`, ou que começa com `|`, é reinterpretado ao reabrir o arquivo.
- Texto rico (HTML) em um nó é gravado no `.mi` como texto puro.
- Só o mapa da aba ativa é conferido contra o disco; o de outra aba é conferido quando você volta para ela. Veja Mapa alterado no disco.
- O assistente de IA é experimental. No Windows, ainda não foram testados a mudança de largura do painel, a saída com `/exit` e o fechamento do Freemapper com sessões abertas.

- O testador de XPath só enxerga as classes buscadas no servidor: caminhos que leem outras partes do projeto, como `//maps`, voltam vazios. As funções `mi:tabulate-text` e `mi:concat-change-pattern` são aproximações.
- O leitor de Markdown só mostra realce de sintaxe e diagramas Mermaid na página do navegador, não na aba, e os documentos abertos não são restaurados ao reabrir o Freemapper.
- Com um documento Markdown na tela, os comandos de mapa dos menus agem sobre o último mapa usado.
- Com dois Freemappers abertos ao mesmo tempo, só o primeiro consegue ligar a integração na porta padrão.
- O painel do assistente não funciona no macOS, e no Windows exige o Windows 10 versão 1809 ou mais novo. A integração com os mapas funciona em todos os sistemas, com o Claude Code aberto num terminal comum.

---

12 Licença

> **Como acessar:** Ajuda (Help) → Licença (License).

O Freemapper é software livre, distribuído sob a GNU General Public License versão 3 ou posterior. Ele é baseado no FreeMind 1.0.1, cujo código continua sob a GPL versão 2 ou posterior. O código-fonte completo de cada versão acompanha os pacotes, no arquivo `freemapper-<versão>-src.zip`.

Na pasta do programa estão o `LICENCA.txt` (resumo, marcas e código-fonte), o `GPL-3.0.txt` (texto da licença) e o `AVISOS-DE-TERCEIROS.txt` (bibliotecas de terceiros e as suas licenças).

Os nomes "Mapper Idea" e "Freemapper" e os seus logotipos são marcas e não estão licenciados pela GPL. Claude e Claude Code são produtos da Anthropic; o Freemapper não é afiliado à Anthropic.

Mapper Idea®
Plataforma de Automação para Engenharia de Software.
Dúvidas: contato@mapperidea.io