

Plugin Pesquisa MirandasTech para Claude e Codex: passo a passo

Instalar no Claude Code, no Claude Desktop, no Codex e no ChatGPT; criar a pasta do projeto; conduzir as onze fases com o Pesquisador, o Metodólogo, o Orientador e o Bibliotecário; usar o conector do Zotero; registrar e declarar — com um pacote que **roda no seu computador e só fecha uma fase com evidência**.

SÉRIE

Manuais MirandasTech · ferramentas de pesquisa

VERSÃO

1.0 (rascunho) · setembro de 2026

FORMATO

– slides

LICENÇA

CC BY 4.0 — cite a fonte

COMO USAR CADA SLIDE É UM PASSO QUE VOCÊ EXECUTA NA TELA OU NO TERMINAL

Para quem é — e o que você vai saber fazer ao final

Este manual é para quem faz ou orienta uma pesquisa de pós-graduação — trabalho de conclusão, dissertação, tese, revisão sistemática — e já usa o Claude Code, o Claude Desktop, o Codex ou o ChatGPT. Ele mostra como colocar dentro dessas ferramentas os quatro papéis dos manuais MirandasTech, com regras que impedem inventar dado, referência ou decisão. Não exige programação: todos os comandos estão em blocos para copiar e colar.

Ao final, você vai conseguir

- **Instalar** o plugin no cliente que usa: Claude Code (pacote completo), Claude Desktop (só o conector), Codex (aplicativo e linha de comando) e ChatGPT (diretório de plugins).
- **Criar** a pasta do projeto com um comando e reconhecer os arquivos que a governam: `PROGRESSO.md` , `DECISOES.md` , `USO_DE_IA.csv` e `AGENTS.md` .
- **Conduzir** as onze fases com o Pesquisador e chamar o especialista certo em cada uma — sabendo o que ele faz e o que ele se recusa a fazer.
- **Usar** o conector do Zotero para auditar a biblioteca e gerar referências no estilo ABNT a partir da própria biblioteca, sem digitar referência de memória.
- **Registrar** decisões e uso de IA, **declarar** o uso no trabalho e **manter** o plugin atualizado.

REGRA

Todo comando aparece em bloco com botão «copiar». Nomes de menu e de botão são os da interface em português, observados em setembro de 2026; quando a sua tela divergir, vale a tela. Nada aqui foi inventado: os fatos vêm do repositório do plugin e dos diretórios em que ele está publicado.

ARMADILHA

O que a regra evita: tratar o plugin como quem faz a pesquisa. Ele formula, registra e confere; a decisão de método, a inclusão de estudos, a interpretação e o texto são seus. Isso não é um conselho: está escrito nas regras que os agentes leem antes de agir.

MANUAIS IRMÃOS

Este manual cuida da *instalação e do uso do plugin*. O conteúdo de cada papel está nos manuais de origem: Metodologia da pesquisa científica, Uso de IA na pesquisa científica, Zotero para pesquisa

científica e Gemini Notebook para pesquisa científica. O hub que reúne todos: mirandastech.com.br/pesquisa.

O que o plugin é: quatro papéis, uma camada de leitura, sete comandos e um conector

Um plugin é um pacote de arquivos de texto que um assistente de IA lê antes de responder: instruções de papel (agentes), roteiros por tarefa (skills), atalhos (comandos) e a ligação com um programa do seu computador (conector). O Pesquisa MirandasTech reúne num só pacote o que os quatro manuais ensinam, com as regras de cada um.

COMPONENTE	O QUE FAZ	ONDE ENTRA
Pesquisador (agente)	Gerente do projeto: conduz onze fases (diagnóstico → entrega), mantém um único <code>PROGRESSO.md</code> , chama o especialista certo e só fecha uma fase com <u>evidência</u> . É o papel padrão — quem não sabe a quem perguntar, pergunta a ele.	Claude Code (agente) · Codex (prompt principal e <u>skill</u>)
Metodólogo (agente)	Pergunta, objetivos, classificação, desenho, amostra, instrumento, análise; projeto conforme a <u>NBR 15287:2025</u> ; banca simulada.	Claude Code · Codex (<u>skill</u>)
Orientador (agente)	O que pode ser delegado à IA em cada etapa, registro em <code>USO_DE_IA.csv</code> , triagem assistida com status preliminar, declaração de uso.	Claude Code · Codex (skill)
Bibliotecário (agente)	Zotero: configuração, auditoria da biblioteca, referência <u>ABNT</u> gerada pelo próprio Zotero, exportação <u>Better BibTeX</u> .	Claude Code · Codex (skill)
Leitura (skill)	Fluxos e prompts do <u>Gemini Notebook</u> ; conferência da <u>matriz de síntese</u> célula a célula. Sem API: a ponte é por arquivo.	Claude Code · Codex
Conector MCP <code>zotero</code>	Sete ferramentas, só de leitura, que falam com o Zotero aberto no seu computador: status, listar coleções, buscar, detalhar, referência ABNT, auditar, exportar BibTeX.	Claude Code · Claude Desktop · Codex
Sete comandos <code>/pesquisa:...</code>	iniciar, status, proxima, auditar, declarar, banca, entregar — atalhos para as fases do Pesquisador.	Claude Code (no Codex, prompts <code>/pesquisador</code> etc.)

COMPONENTE	O QUE FAZ	ONDE ENTRA
<u>Hook de início de sessão</u>	Ao abrir uma sessão numa pasta com <code>PROGRESSO.md</code> , mostra as pendências e lembra de chamar o Pesquisador.	Claude Code
Scripts e modelos	Criar o projeto, verificar citações e referências, calcular amostra, cronograma, auditar a biblioteca, exportar e fazer backup do Zotero; modelos de checklist, decisões, registro de IA, <u>TCLE</u> , <u>matriz de síntese</u> , estilos ABNT.	Copiados para a pasta do projeto

REGRA

Tudo é gratuito, aberto (CC BY 4.0) e roda no seu computador. Não há servidor da MirandasTech, conta, chave nem cota. O plugin não gasta créditos seus: o modelo é pago pela sua própria assinatura do Claude ou do Codex.

Como os papéis se dividem: quem responde o quê — e quem decide

PERGUNTA QUE VOCÊ FAZ	QUEM RESPONDE	QUEM DECIDE
«Onde paramos? O que falta? Qual o próximo passo?»	PESQUISADOR	Você, sobre a ordem; ele, sobre o que é evidência
«Minha pergunta está boa? Qual método? Que amostra? Como monto o projeto?»	METODÓLOGO	Você; ele apresenta alternativas com prós e contras
«Posso usar IA nesta etapa? Como registro? Como declaro?»	ORIENTADOR	Você e o orientador humano; ele redige o registro

O PLUGIN FAZ

- Apresenta alternativas com prós e contras e faz a pergunta que decide.
- Registra a sua escolha em `DECISOES.md`, numerada e datada, depois que você confirma.
- Roda scripts, lê arquivos e confere evidência antes de marcar um item.
- Chama o especialista certo na fase certa e aponta o manual e o slide.

O PLUGIN NÃO FAZ

- Escolher método, critério de inclusão ou interpretação por você.
- Escrever Resultados, Discussão ou Conclusão.
- Inventar dado, número, referência ou DOI.
- Marcar fase sem evidência; liberar coleta com pessoas sem parecer do comitê de ética.

REGRA

O aluno decide; o agente formula. Quando dois especialistas divergem (o Orientador veta uma delegação que o Metodólogo sugeriu), vale a regra mais restritiva, e a divergência vai para `DECISOES.md`. Só o Pesquisador escreve na tabela de fases; cada especialista marca os itens da própria seção.

PERGUNTA QUE VOCÊ FAZ	QUEM RESPONDE	QUEM DECIDE
«Por que a referência saiu errada? O que falta na biblioteca? Qual a referência ABNT deste item?»	BIBLIOTECÁRIO	Ele corrige no item do Zotero, nunca na lista do documento
«Como leio esta pilha de PDFs? A matriz está certa?»	LEITURA	Você confere cada célula; ele orienta e verifica as linhas

Roteiro: cinco partes, na ordem em que você vai usá-las

Parte 1

Instalar no Claude

Claude Code pelo [marketplace](#), o inventário instalado, o script de instalação, o [diretório community](#) e o Claude Desktop (só o conector)

Parte 2

Instalar no Codex e no ChatGPT

Aplicativo do Codex pelo diretório, linha de comando pelo [marketplace](#) próprio, conector e prompts, aprovação de ferramentas, ChatGPT com @menção

Parte 3

Usar

Criar o projeto, a árvore e o `PROGRESSO.md`, as onze fases e as portas, a sessão típica, os especialistas, os comandos, o conector, a leitura, as recusas, o registro

Parte 4

Manter

Atualizar e publicar nova versão, problemas conhecidos, privacidade, termos e licença

Parte 5

Fechamento

Erros comuns, checklist de instalação e primeira sessão, glossário, referências, repositório, os manuais juntos

COMO LER

Escolha o cliente que usa e siga só a parte dele (1 ou 2); todo mundo passa pela Parte 3. Se você já instalou, vá direto ao slide «Sessão típica» e volte quando um passo não estiver claro. A Parte 4 é para quem mantém o plugin ou quer entender o que ele faz com os seus dados.

CONVENÇÕES

Nomes de menu e de botão entre «aspas angulares». Comandos de terminal e de chat em blocos com botão «copiar». Siglas explicadas na primeira vez que aparecem e no glossário. Capturas em 1920 × 938, feitas em setembro de 2026, com a interface em português.

| _____ |

Marketplace, plugin, inventário — e o Claude Desktop

Dois comandos no Claude Code instalam tudo. Depois: ver o que entrou, o script que faz o mesmo pelo terminal, o diretório `community` e o que o Claude Desktop recebe (só o conector).

041

Claude Code: adicionar o marketplace e instalar o plugin

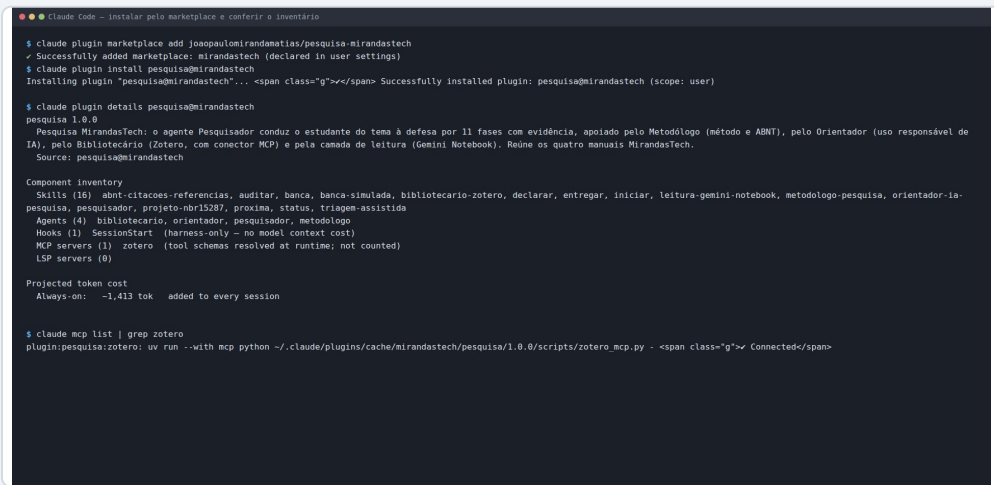


FIGURA 1 No Claude Code, os dois comandos: primeiro o catálogo («marketplace») do repositório, depois o plugin `pesquisa` desse catálogo.

1. Abra o Claude Code no terminal (comando `claude`), em qualquer pasta.
2. Adicione o catálogo do repositório (bloco abaixo, primeira linha). O catálogo se chama `mirandastech` .
3. Instale o plugin `pesquisa` desse catálogo (segunda linha). O formato é sempre `plugin@catálogo` .
4. Confira o que entrou com `claude plugin details` (próximo slide).

CLAUDE CODE – INSTALAR PELO MARKETPLACE

```
/plugin marketplace add joaopaulomirandamatias/pesquisa-mirandastech
/plugin install pesquisa@mirandastech
```

REQUISITO	PARA QUÊ
Python 3 e Git	Scripts do projeto (criar a pasta, verificar citações) e controle de versão da pasta criada
<code>uv</code>	Executa o conector do Zotero (alternativa: <code>pip install r</code> trocar o comando por <code>python3</code>)
Zotero 7 ou superior	Aberto, com a API local ligada em «Configurações» → «Avançado», para as ferramentas de biblioteca

O QUE É O MARKETPLACE

Um arquivo `marketplace.json` no repositório que lista plugins instaláveis. O catálogo `mirandastech` lista dois: `pesquisa` (o pacote completo deste manual) e `zotero-pesquisa` (só o Bibliotecário, para quem quer apenas o Zotero).

O que aparece depois de instalar: `claude plugin details`

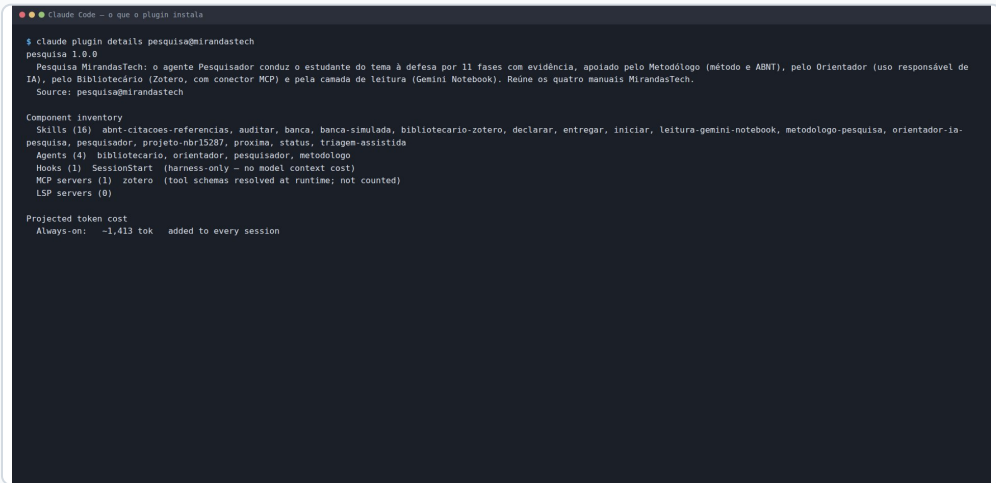


FIGURA 2 `claude plugin details pesquisa@mirandastech` mostra o inventário do plugin: quatro agentes, nove skills, sete comandos, o conector `zotero` e o hook de início de sessão.

TERMINAL — INVENTÁRIO E VALIDAÇÃO	
<code>claude plugin details pesquisa@mirandastech</code> <code>claude plugin validate .</code>	
TIPO	O QUE ENTROU
4 agentes	<code>pesquisador</code> , <code>metodologo</code> , <code>orientador</code> , <code>biblioteca</code>
9 skills	<code>pesquisador</code> , <code>metodologo-pesquisa</code> , <code>projeto-nbr152</code> <code>abnt-citacoes-referencias</code> , <code>banca-simulada</code> , <code>orientador-ia-pesquisa</code> , <code>triagem-assistida</code> , <code>bibliotecario-zotero</code> , <code>leitura-gemini-notebook</code>
7 comandos	<code>/pesquisa:iniciar</code> · <code>status</code> · <code>proxima</code> · <code>auditar</code> · <code>declarar</code> · <code>banca</code> · <code>entregar</code>
1 conector MCP	<code>zotero</code> , que aparece como <code>plugin:pesquisa:zotero</code> ; ferramentas <code>status</code> , <code>listar_colecoes</code> , <code>buscar_itens</code> <code>detalhar_item</code> , <code>referencia_abnt</code> , <code>auditar</code> , <code>exportar_bbt</code>
1 hook	<code>SessionStart</code> : se a pasta tem <code>PROGRESSO.md</code> , imprime fa atual, checklists (M, I, B, L), última decisão datada e há quantos o arquivo não muda, e lembra: «Diga: Pesquisador, onde param

`claude plugin validate .` confere os manifestos do plugin na pasta atual — útil para quem clonou o repositório e alterou um agente ou uma skill antes de instalar.

Pelo terminal (instalar_claude.sh) – e o diretório community, quando for aceito

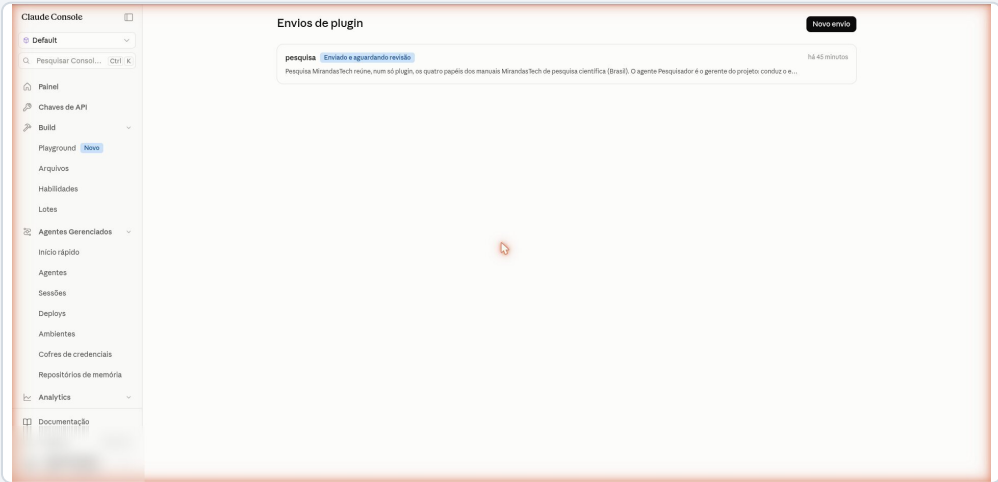


FIGURA 3 Console (platform.claude.com/plugins/submissions): o envio «pesquisa», feito em 08/09/2026, com o status «Enviado e aguardando revisão». O marketplace oficial da Anthropic não aceita submissão de terceiros; o caminho é o diretório community.

TERMINAL – CLONAR E INSTALAR COM O SCRIPT		
<pre>git clone https://github.com/joaopaulomirandamatias/pesquisa-mirandastech.git cd pesquisa-mirandastech bash scripts/instalar_claude.sh</pre>		
CAMINHO	ESTADO	COMANDO
Marketplace próprio	DISPONÍVEL	/plugin install pesquisa@miranda...
Script pelo terminal	DISPONÍVEL	bash scripts/instalar_claude...
Diretório community	AGUARDANDO REVISÃO	/plugin install pesquisa@claude-community

- O script adiciona o catálogo (ou o atualiza, se já existe), instala o plugin e avisa se o `uv` não está instalado.
- Se encontra o arquivo de configuração do Claude Desktop, também registra nele o conector `zotero` (próximo slide).
- O comando do diretório community só funciona depois que a submissão for aceita; até lá, use o marketplace próprio — o conteúdo é o mesmo.

Claude Desktop: só o conector do Zotero — sem agentes nem skills

- 1. Três caminhos, todos com o mesmo resultado: `bash scripts/instalar_claude.sh` (slide anterior); a oferta que o `/pesquisa:iniciar` faz na pergunta sobre ferramentas (ele só escreve com o seu «sim»); ou, à mão, `python3 scripts/registrar_desktop.py --registrar`. Todos escrevem a entrada `zotero` (bloco ao lado) em `claude_desktop_config.json`, com backup. Quem instalou só pelo marketplace precisa de um deles.
- 2. Reinicie o Claude Desktop. O conector aparece entre os conectores da conversa, com as sete ferramentas (`status`, `listar_colecoes`, `buscar_itens`, `detalhar_item`, `referencia_abnt`, `auditar`, `exportar_bbt`).
- 3. Com o Zotero aberto e a API local ligada, peça: «Liste as coleções da minha biblioteca» — a resposta vem da ferramenta `listar_colecoes`.

ARMADILHA

Pedir «Pesquisador, onde paramos?» no Desktop. Não há agente nem skill ali; o Desktop recebe só as sete ferramentas do Zotero, para usar dentro de qualquer conversa. O fluxo por fases é no Claude Code (inclusive na aba «Code» do próprio Desktop) ou no Codex.

CLAUDE_DESKTOP_CONFIG.JSON — ENTRADA ESCRITA PELO SCRIPT (PARA CONFERIR OU FAZER À MÃO)

```
{
  "mcpServers": {
    "zotero": {
      "command": "uv",
      "args": ["run", "--with", "mcp", "python",
               "/caminho/para/pesquisa-mirandastech/scripts/
               zotero_mcp.py"]
    }
  }
}
```

SISTEMA	ONDE FICA O ARQUIVO
Linux	~/.config/Claude/claude_desktop_config.json
macOS	~/Library/Application Support/Claude/claude_desktop_config.json
Windows	%APPDATA%\Claude\claude_desktop_config.json

O Claude Desktop traz o Claude Code numa aba própria. Ali o plugin funciona por inteiro — agentes, comandos e hook — porque é o mesmo Claude Code do terminal, com a pasta do projeto aberta.

Diretório oficial, marketplace próprio e a @menção

No aplicativo do Codex e no ChatGPT, o plugin se instala pelo diretório oficial de plugins. Na linha de comando, pelo marketplace próprio — e o script registra o conector e os prompts.

02

Aplicativo do Codex: «Plugins» → «Procurar plugins» → a página do plugin

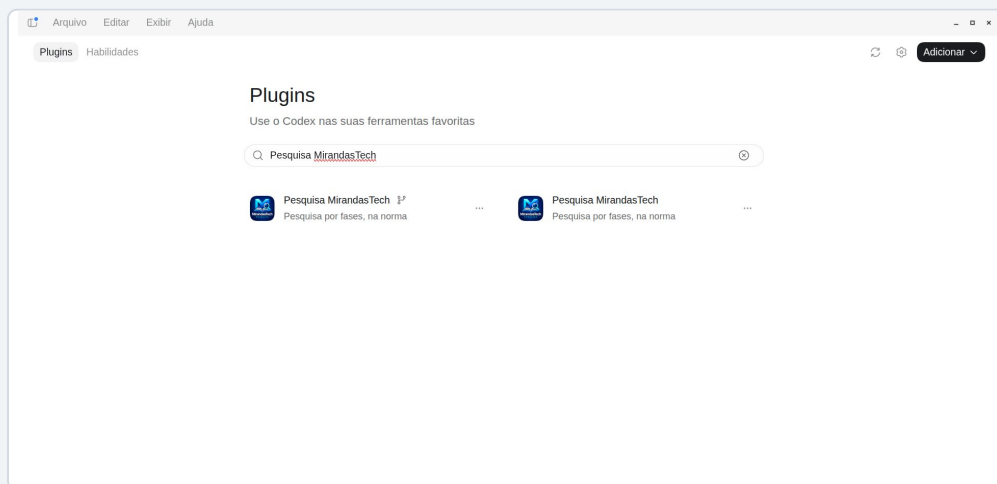


FIGURA 4 «Plugins» → caixa «Procurar plugins» → «Pesquisa MirandasTech». Aparecem dois resultados: o do marketplace próprio (ícone de ramificação, instalado pelo comando da linha de comando) e o do diretório oficial. Para quem só usa o aplicativo, vale o segundo.

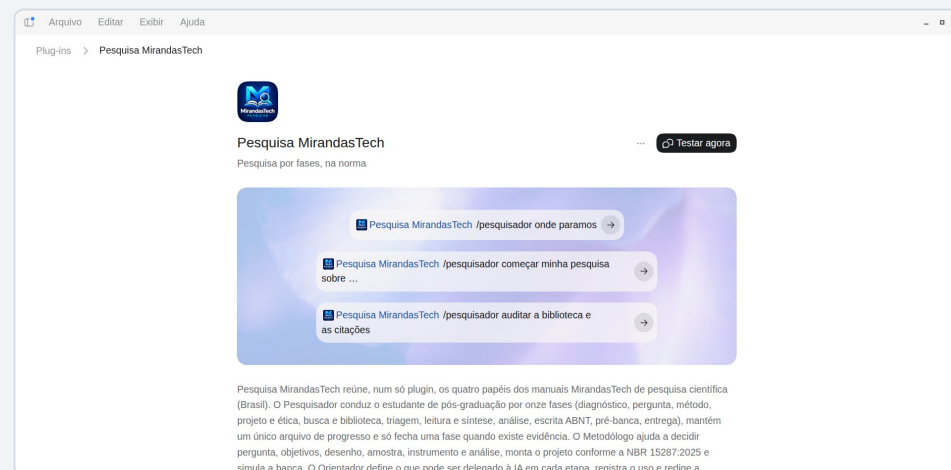


FIGURA 5 A página do plugin: subtítulo «Pesquisa por fases, na norma», os três pedidos de partida e a descrição. O botão principal instala; já instalado, ele vira «Testar agora» e abre o chat com a menção pronta.

1. No aplicativo do Codex, abra «Plugins» (menu da esquerda).
2. Na caixa «Procurar plugins», digite «Pesquisa MirandasTech» e clique no resultado do diretório.
3. Clique no botão principal da página para instalar. Depois, «Testar agora» abre um chat com «@Pesquisa MirandasTech /pesquisador onde paramos» já escrito.

FATO	NA PÁGINA DO PLUGIN
Nome	Pesquisa MirandasTech
Subtítulo	«Pesquisa por fases, na norma»
Categoria	Education & Research (Educação e pesquisa)
Tipo	Skills only — nove skills

O QUE «SKILLS ONLY» SIGNIFICA

O Codex não tem subagentes: os quatro papéis viram skills chamadas pelo nome, o Pesquisador vira o prompt principal e as regras comuns ficam em `AGENTS.md`, lido na pasta do projeto.

Aplicativo do Codex: escolher a pasta do projeto e chamar @Pesquisa MirandasTech

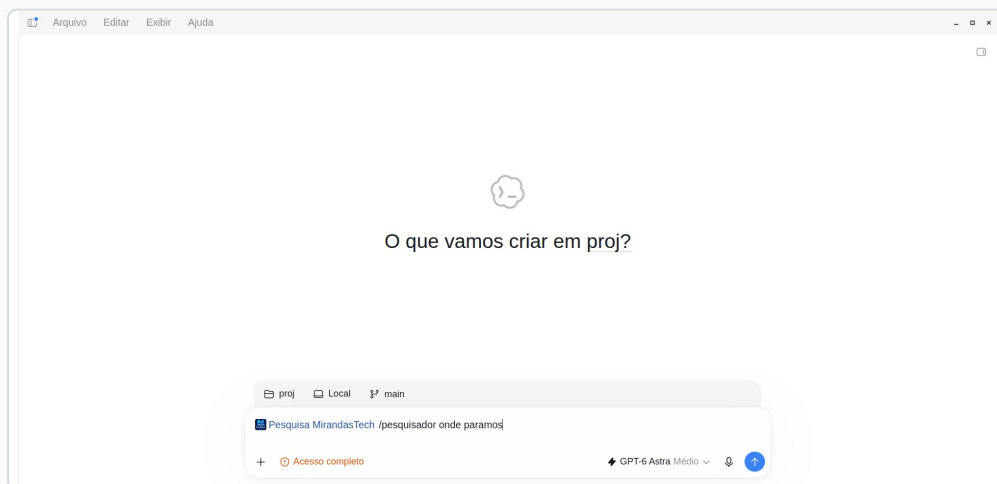


FIGURA 6 A caixa de mensagem com a pasta do projeto escolhida (etiquetas «proj · Local · main») e a menção pronta. Sem pasta escolhida, o botão de enviar fica desativado.

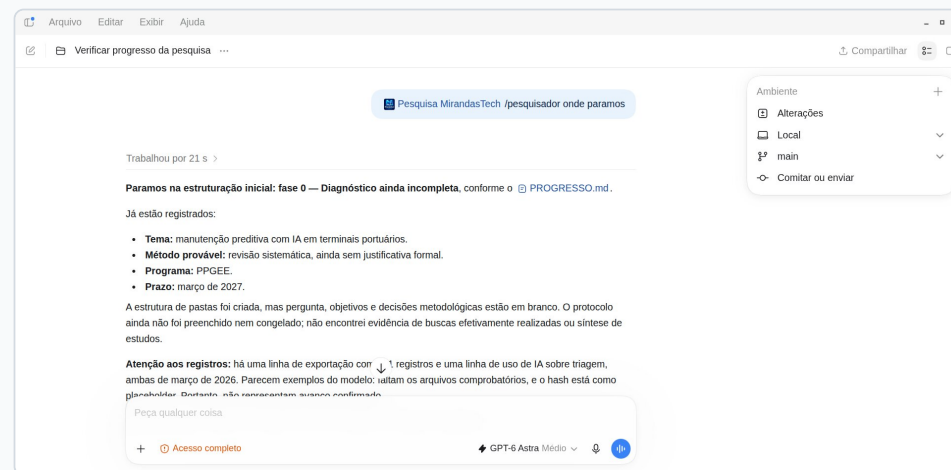


FIGURA 7 A resposta: o Codex leu o `PROGRESSO.md` da pasta e respondeu como o Pesquisador — fase 0 ainda incompleta, o que já está registrado, o que falta e a pergunta seguinte («Quem é o orientador deste projeto?»).

1. Na caixa de mensagem, use o seletor de projeto (à esquerda, acima do texto) para escolher a pasta do projeto — ou «Novo projeto» para adicioná-la. Sem pasta, o Codex não envia.
2. No chat, digite `@Pesquisa MirandasTech` e, na mesma mensagem, o pedido.

3. Para chamar um especialista, diga o papel no pedido («como Metodólogo, ...») — no Codex, os papéis são skills, não agentes separados.

CHAT DO CODEX — PEDIDOS DE PARTIDA

@Pesquisa MirandasTech onde paramos
@Pesquisa MirandasTech começar minha pesquisa sobre manutenção
preditiva em portos
@Pesquisa MirandasTech auditar a biblioteca e as citações

REGRA

O Codex lê `AGENTS.md` da pasta aberta. É esse arquivo que carrega as proibições absolutas e os gates humanos do projeto — por isso o fluxo completo exige a pasta do projeto aberta, não um chat solto.

Codex pela linha de comando: marketplace próprio, conector `zotero` e prompts



FIGURA 8 No terminal: o catálogo do repositório e o plugin `pesquisa-mirandastech`. O script `instalar_codex.sh` faz isso e ainda registra o conector e os prompts.

TERMINAL – INSTALAR PELO MARKETPLACE PRÓPRIO

```
codex plugin marketplace add joaopaulomirandamatias/pesquisa-mirandastech
codex plugin add pesquisa-mirandastech@mirandastech
```

TERMINAL – OU TUDO DE UMA VEZ, COM O SCRIPT

```
git clone https://github.com/joaopaulomirandamatias/pesquisa-mirandastech.git
cd pesquisa-mirandastech
bash scripts/installar_codex.sh
codex plugin list
```

O QUE O SCRIPT FAZ	ONDE
Adiciona (ou atualiza) o catálogo e instala o plugin	Codex
Registra o conector com <code>codex mcp add zotero -- uv run --with mcp python scripts/zotero_mcp.py</code>	<code>~/.codex/config.toml</code>
Copia os prompts <code>/pesquisador</code> , <code>/metodologo</code> , <code>/orientador</code> , <code>/bibliotecario</code>	<code>~/.codex/prompts/</code>
Se o comando <code>codex</code> não existe, acrescenta a seção <code>[mcp_servers.zotero]</code> à mão	<code>~/.codex/config.toml</code>

NOMES DIFERENTES NOS DOIS CLIENTES

No Claude Code o plugin se chama `pesquisa` ; no Codex, `pesquisa-mirandastech` . O catálogo se chama `mirandastech` nos dois. O repositório é o mesmo.

No Codex, cada chamada ao conector pede aprovação – como liberar as ferramentas que só leem

Por padrão, o Codex pede a sua aprovação a cada chamada de ferramenta de um conector MCP. As sete ferramentas do `zotero` apenas leem a biblioteca; nenhuma escreve nem envia nada para fora do computador. Se quiser dispensar a pergunta, o `instalar_codex.sh` oferece liberar as sete de uma vez (responda «s»; ele faz backup do `config.toml`); ou libere ferramenta por ferramenta, à mão, com o bloco abaixo.

```
~/.CODEX/CONFIG.TOML – LIBERAR FERRAMENTAS DO CONECTOR, UMA A UMA

[mcp_servers.zotero.tools.status]
approval_mode = "auto"

[mcp_servers.zotero.tools.listar_colecoes]
approval_mode = "auto"

[mcp_servers.zotero.tools.buscar_itens]
approval_mode = "auto"

[mcp_servers.zotero.tools.auditar]
approval_mode = "auto"

[mcp_servers.zotero.tools.referencia_abnt]
approval_mode = "auto"
```

FERRAMENTA	O QUE FAZ	ESCRE
status	Confirma que o Zotero está aberto e a API local responde	NÃO
listar_colecoes	Lista as coleções da biblioteca	NÃO
buscar_itens	Busca itens por texto, coleção ou etiqueta	NÃO
detalhar_item	Metadados completos de um item, pela chave	NÃO
referencia_abnt	Referência no estilo ABNT gerada pelo Zotero, de itens ou de uma coleção	NÃO
auditar	Pendências por item: sem DOI, sem PDF, caixa alta, duplicados	NÃO
exportar_bbt	Exporta uma coleção em BibTeX/BibLaTeX pelo <u>Better BibTeX</u>	NÃO

REGRA

Libere só o que você entende. A tabela existe para isso: cada ferramenta só lê. Se um dia o conector ganhar uma ferramenta que escreva, ela não estará na sua lista de `approval_mode = "auto"` — e o Codex voltará a perguntar.

ChatGPT no navegador: a página do plugin e o botão «Instalar plugin»

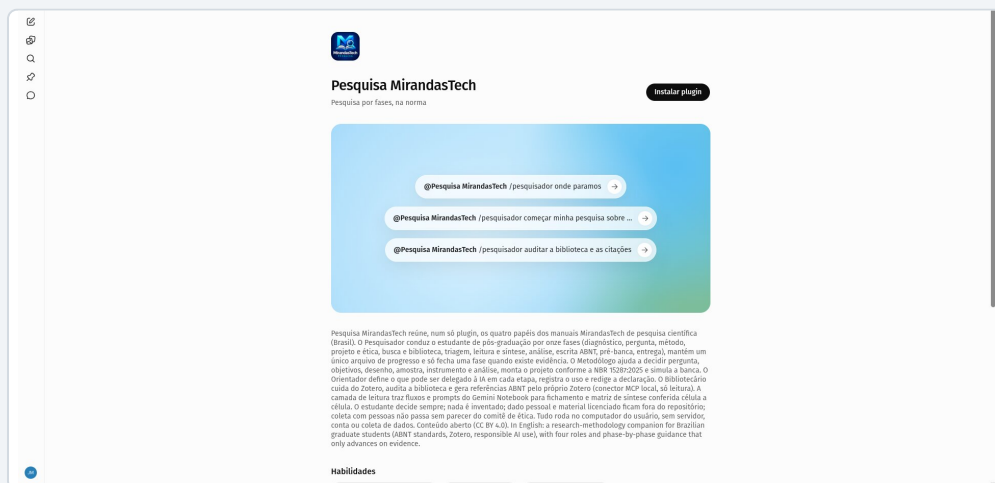


FIGURA 9 A página do plugin no diretório do ChatGPT: nome, subtítulo, categoria, as nove skills e o botão «Instalar plugin».

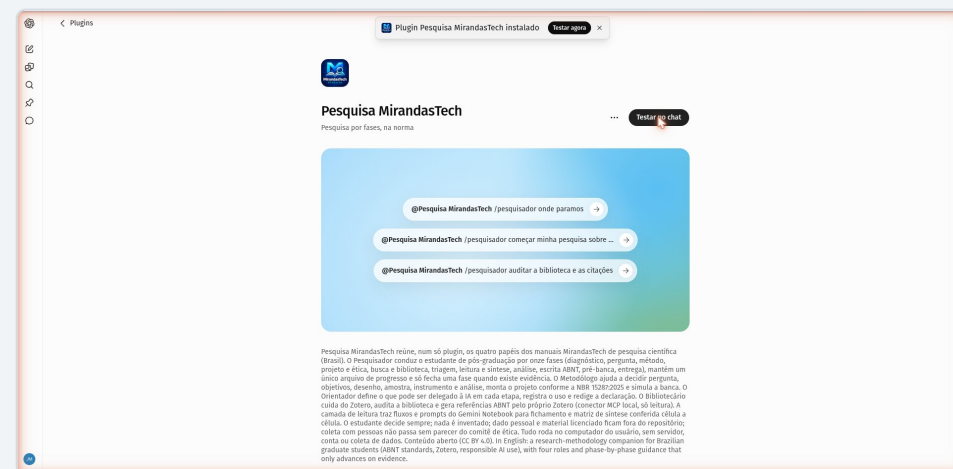


FIGURA 10 Depois do clique, o plugin aparece como instalado na sua conta.

1. Abra a página do plugin no diretório (endereço abaixo) com a sua conta do ChatGPT.
2. Clique em «Instalar plugin».
3. Volte ao chat: o plugin passa a responder à @menção (próximo slide).

ENDEREÇO – PÁGINA DO PLUGIN NO DIRETÓRIO DO CHATGPT/CODEX

[https://chatgpt.com/plugins/
plugins_6aa0a0320a248191b33a514a3a52325b](https://chatgpt.com/plugins/plugins_6aa0a0320a248191b33a514a3a52325b)

MESMO DIRETÓRIO

O [diretório de plugins](#) é um só para o ChatGPT e para o Codex. O que você instala aqui é o mesmo pacote «Skills only» do aplicativo do Codex: as nove skills e as regras dos quatro papéis.

ChatGPT no navegador: @Pesquisa MirandasTech seguido do pedido

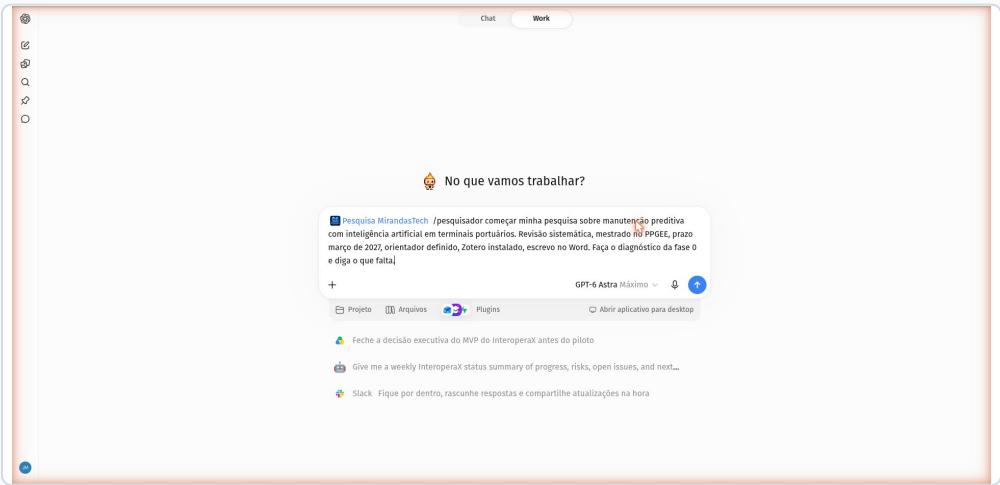


FIGURA 11 Digite «@», escolha «Pesquisa MirandasTech» e escreva o pedido na mesma mensagem.

CHAT DO CHATGPT – EXEMPLOS

@Pesquisa MirandasTech começar minha pesquisa sobre manutenção preditiva em portos

@Pesquisa MirandasTech como Metodólogo: minha pergunta é «como a IA melhora a manutenção?» – está boa?

@Pesquisa MirandasTech como Orientador: posso usar IA na triagem de resumos? como registro?

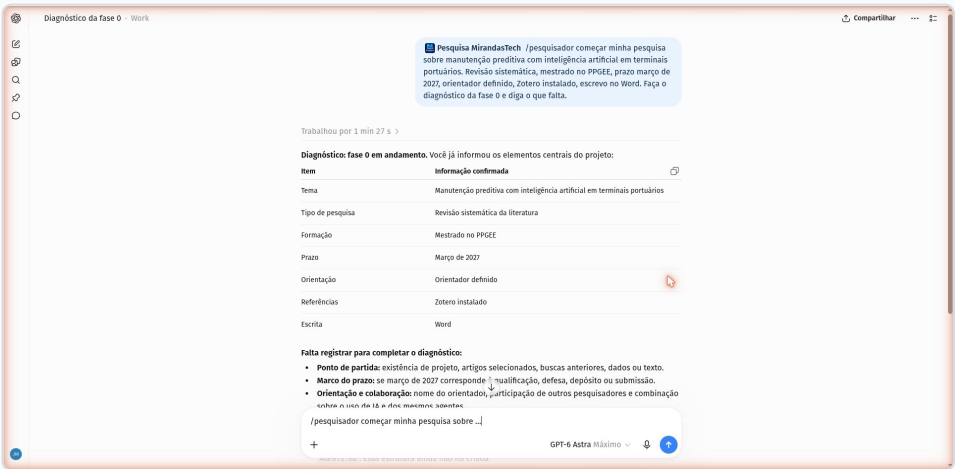


FIGURA 12 A resposta segue o roteiro do Pesquisador: uma pergunta do diagnóstico por vez, sem decidir por você.

NO CHATGPT VOCÊ TEM	SÓ NO CODEX (APLICATIVO OU LINHA DE COMANDO) E NO CLAUDE CODE
As nove skills: roteiros, regras, perguntas de diagnóstico, checklists por papel; os arquivos que ele produz vêm como anexos para baixar (ex.: <code>DECISOES_METODO.md</code> , um <code>.zip</code> com a estrutura de pastas)	A pasta do projeto lida e escrita no seu computador (<code>PROGRESSO.md</code> , <code>DECISOES.md</code> , <code>USO_DE_IA.csv</code>), sem baixar nada
Os quatro papéis, chamados pelo nome no pedido	O conector <code>zotero</code> e os scripts (auditar biblioteca, verificar citações, criar o projeto)

ARMADILHA

Esperar que o ChatGPT audite o seu Zotero. O conector roda no seu computador e é registrado pelo `instalar_codex.sh` ou pelo `instalar_claude.sh`; o pacote do diretório traz as skills. Para o fluxo com pasta, scripts e biblioteca, use o Codex ou o Claude Code na pasta do projeto.

ChatGPT no navegador: chamar um especialista — o Metodólogo reformula a pergunta e cobra a tabela vazia

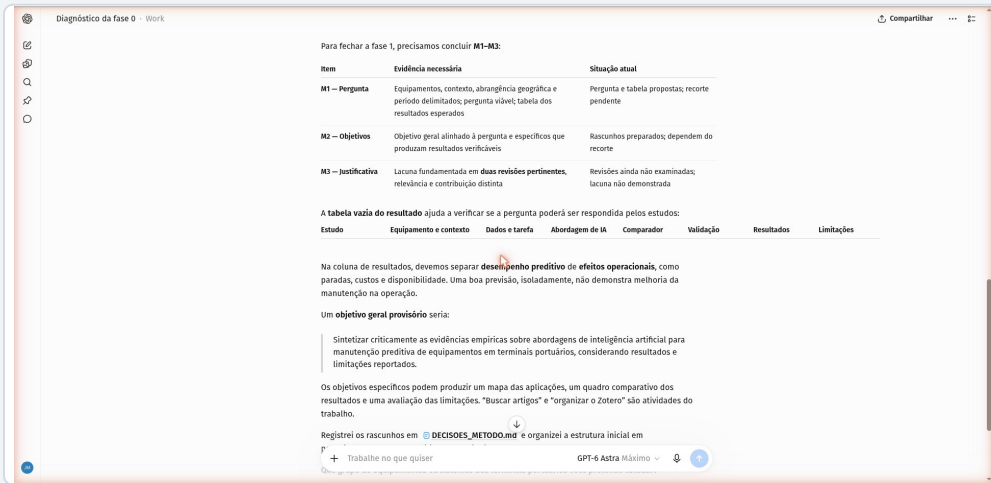


FIGURA 13 «Metodólogo, minha pergunta é ..., está boa?». Ele diz que ainda precisa de refinamento, propõe uma versão provisória, lista o que fecha M1–M3 (evidência necessária × situação atual) e monta a **tabela vazia do resultado** — as colunas que os estudos terão de preencher.

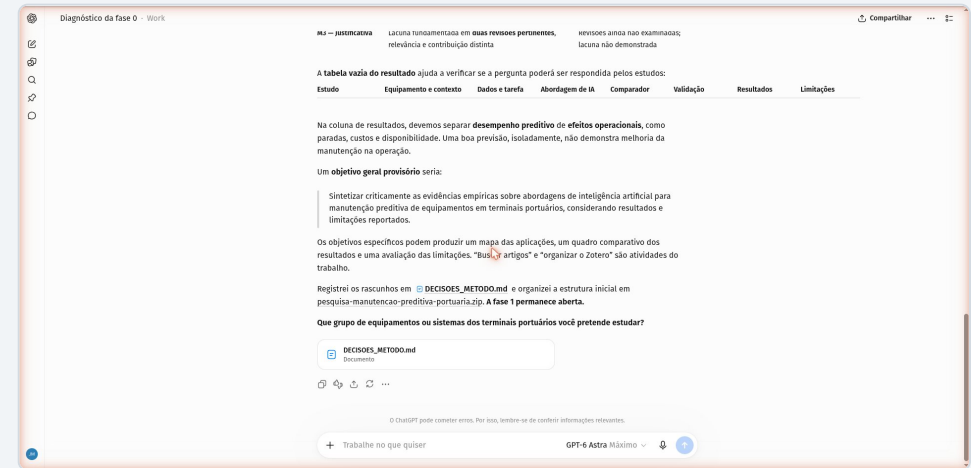


FIGURA 14 O fim da resposta: objetivo geral provisório, o arquivo `DECISOES_METODO.md` como anexo, «a fase 1 permanece aberta» e a pergunta que ele faz antes de seguir. Ele não fecha a fase sozinho.

1. Na mesma conversa, chame o papel pelo nome depois da @menção: «Metodólogo, ...», «Orientador, ...», «Bibliotecário, ...».
2. Leia a tabela «Evidência necessária × Situação atual»: é o checklist da fase, no formato do `PROGRESSO.md`.
3. Baixe o anexo para `00_projeto/` (no Codex ou no Claude Code ele já nasce lá).

4. Responda à pergunta final — cada resposta sua fecha um item; ele não avança sem ela.

O QUE APARECEU	O QUE SIGNIFICA
«Ainda precisa de refinamento»	A pergunta original («como a IA melhora a manutenção em portos?») pressupõe o benefício que a revisão deveria investigar; ele propõe uma versão neutra e delimitada
M1–M3 abertos	Pergunta, objetivos e justificativa dependem do recorte (equipamentos, contexto, período) e de duas revisões pertinentes ainda não examinadas
Tabela vazia do resultado	Estudo · equipamento e contexto · dados e tarefa · abordagem de IA · comparador · validação · resultados · limitações. Ele separa desempenho preditivo de efeito operacional: prever bem não prova redução de paradas

Do projeto à entrega, uma fase por vez

Criar a pasta, entender o `PROGRESSO.md`, as onze fases e as portas, a sessão típica, os especialistas, os sete comandos, o conector do Zotero em uso, a camada de leitura, o que o plugin recusa e o registro de decisões e de uso de IA.

Criar o projeto: `iniciar_projeto.py` pelo terminal ou `/pesquisa:iniciar` no Claude Code



FIGURA 15 O script imprime cada arquivo criado (prefixo «+»), inicializa o Git e termina com a instrução para a primeira sessão.

TERMINAL – CRIAR A PASTA DO PROJETO (NA PASTA DO REPOSITÓRIO CLONADO)

```
python3 scripts/iniciar_projeto.py minha-pesquisa \
  --tema "manutenção preditiva em terminais portuários" \
  --metodo "revisão sistemática" --programa "PPGEE" --prazo
2027-03
cd minha-pesquisa
```

CLAUDE CODE – O MESMO, COM DIAGNÓSTICO GUIADO (FASE 0)

`/pesquisa:iniciar`

#	PERGUNTA DO DIAGNÓSTICO (UMA POR MENSAGEM)
1	Tema em uma frase e o que já existe (nada, projeto, dados, texto)
2	Programa, orientador e prazo (qualificação, defesa, submissão)
3	Tipo provável de pesquisa (revisão, empírica com pessoas, experimental, documental, mista)
4	Ferramentas: Zotero? Gemini Notebook? Conta pessoal ou institucional? Editor de texto?
5	Coautores e orientador vão usar os mesmos agentes? (grupo no Zotero, acordo de uso de IA)

REGRA

Nunca sobrescreve. Se a pasta já existe, o script só acrescenta o que falta. Pasta criada pelos manuais antigos (com `PROGRESSO_IA.md` , `PROGRESSO_METODOLOGIA.md` ou `PROGRESSO_ZOTERO.md`) tem os itens marcados migrados para o `PROGRESSO.md` ; os arquivos antigos ficam como histórico.

A árvore criada — e o PROGRESSO.md único, com as fases e os quatro checklists

ÁRVORE — O QUE INICIAR_PROJETO.PY CRIA

minha-pesquisa/
├ PROGRESSO.md
I, B, L
├ DECISOES.md
├ USO_DE_IA.csv
├ AGENTS.md
├ ACORDO_ORIENTADOR.md
├ .gitignore
*.xlsx, .env
├ 00_projeto/
├ 00_protocolo/
├ 01_buscas/
registro_exportacoes.csv
├ 01_revisao/fichas/
├ 02_instrumentos/
├ 02_triagem/
├ 03_etica/
├ 03_extracao/ 04_sintese/matriz_sintese.csv
├ 05_analise/livro_de_codigos.csv 06_texto/ 07_banca/
├ tools/
prompts e
├
└ dados_FORA_DO_GIT/
(fora do Git)

estado único: fases + checklists M,

decisões numeradas e datadas
uma linha por influência de IA
regras lidas por qualquer agente
acordo sobre uso de IA
dados_FORA_DO_GIT/, *.pdf,

DECISOES_METODO.md · cronograma.csv
protocolo.md (12 seções + emendas)
buscas.csv ·

fichas de leitura com página
tabela_operacionalizacao.csv
codebook, lotes, números do PRISMA
TCLE.md · parecer do CEP

estrutura dos notebooks
estrutura e etiquetas do Zotero;
estrutura dos notebooks
dados pessoais e material licenciado

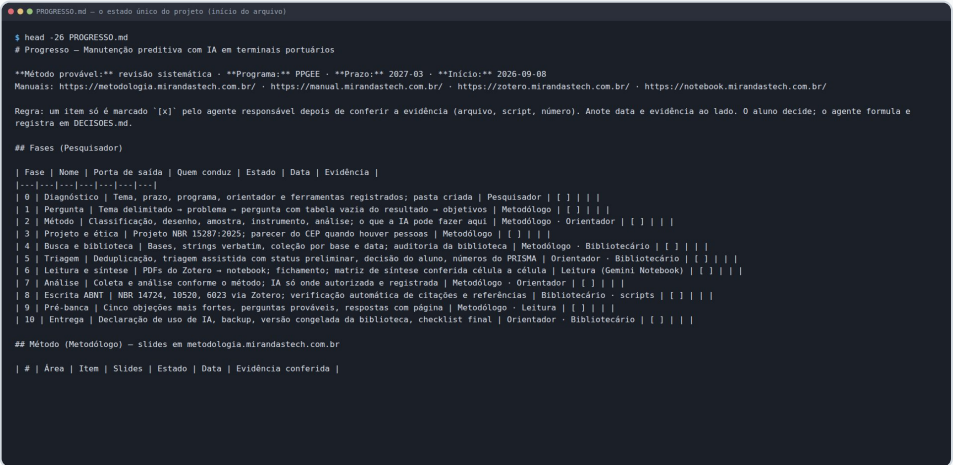


FIGURA 16 PROGRESSO.md : no alto, a tabela das onze fases (porta de saída, quem conduz, estado, data, evidência); abaixo, as quatro seções de checklist, cada item com os slides do manual de origem.

SEÇÃO	ITENS	QUEM MA
Fases	0 a 10	Só o Pesquisa
Método (M1–M16)	Pergunta, objetivos, justificativa... defesa	Metodólc
Uso de IA (I1–I16)	Governança, dados, busca, triagem... declaração, entrega	Orientad
Biblioteca (B1–B12)	Instalação, conta, estilo... backup, grupo	Bibliotec

SEÇÃO	ITENS	QUEM MA
Leitura e síntese (L1–L7)	Conta, fontes, ingestão, citações, matriz, dados sensíveis, declaração	Leitura

REGRA

Um item só recebe [x] depois que o agente responsável conferiu a evidência (arquivo, script, número) — e anota data e evidência ao lado. É a regra escrita no cabeçalho do próprio arquivo.

As onze fases e a porta de saída de cada uma — o que precisa existir para a fase fechar

FASE	PORTA DE SAÍDA (EVIDÊNCIA)	QUEM CONDUZ	MANUAL DE ORIGEM
0 Diagnóstico	Pasta criada; tema, prazo, programa, orientador, ferramentas e tipo de conta em <code>PROGRESS0.md</code>	Pesquisador	—
1 Pergunta	<code>00_projeto/DECISOES_METODO.md</code> com tema delimitado, problema, pergunta com «tabela vazia do resultado», objetivos	Metodólogo	Metodologia 7–9
2 Método	Classificação nos quatro eixos; «escolhi X porque...; descartei Y porque...»; amostra, instrumento e análise; o que a IA pode fazer registrado	Metodólogo · Orientador	Metodologia 12–22 · IA 6–11, 37
3 Projeto e ética	Projeto NBR 15287:2025 em <code>06_texto/</code> ; com pessoas: parecer do CEP com CAAE em <code>03_etica/</code> e TCLE ; acordo com o orientador	Metodólogo	Metodologia 25–28 · IA 6, 17
4 Busca e biblioteca	Protocolo congelado e datado; strings verbatim com data, filtros, total e hash em <code>01_buscas/</code> ; coleção por base e data; auditoria sem pendência crítica	Metodólogo · Bibliotecário	IA 18–22 · Zotero 16, 20–23, 49
5 Triagem	Duplicados mesclados; codebook versionado; triagem assistida com status preliminar e decisão humana por registro; concordância (κ) com intervalo; números do PRISMA	Orientador · Bibliotecário	IA 23–31 · Zotero 23, 40
6 Leitura e síntese	PDFs dos incluídos no notebook; inventário rodado; fichas com página; <code>04_sintese/matriz_sintese.csv</code> com colunas do protocolo e cada célula conferida	Leitura · Bibliotecário	Gemini Notebook 11–23, 32–34 · IA 32–33

FASE	PORTA DE SAÍDA (EVIDÊNCIA)	QUEM CONDUZ	MANUAL DE ORIGEM
7 Análise	Técnica decidida antes; pressupostos; efeito com intervalo ou categorias com trecho; scripts e dados guardados; figuras por script	Metodólogo · Orientador	Metodologia 21–24 · IA 34
8 Escrita ABNT	NBR 14724:2024; <code>verificar_citacoes.py</code> e <code>verificar_refs.py</code> sem erro; referências pelo Zotero (NBR 6023:2025); marcas de IA removidas	Bibliotecário · scripts	Metodologia 25, 30–32 · IA 40–44 · Zotero 30–33
9 Pré-banca	Cinco objeções com resposta; perguntas prováveis a partir da tese, com página; ensaio	Metodólogo · Leitura	Metodologia 34 · Gemini Notebook 36
10 Entrega	<u>Declaração de uso de IA</u> aprovada; guia de relato conferido; backup do Zotero e biblioteca congelada; checklist final sem pendência	Orientador · Bibliotecário	IA 43, 16 · Zotero 41

PORTAS FECHADAS

Quatro portas não se abrem por pedido: coleta com pessoas sem parecer do CEP (Comitê de Ética em Pesquisa); busca sem protocolo congelado; síntese sem matriz com colunas do protocolo; entrega sem declaração de uso de IA aprovada por você. E não se pula fase: uma pergunta mal formulada invalida tudo o que vem depois.

Sessão típica: começa por «onde paramos» e termina por «o próximo passo, em uma frase»

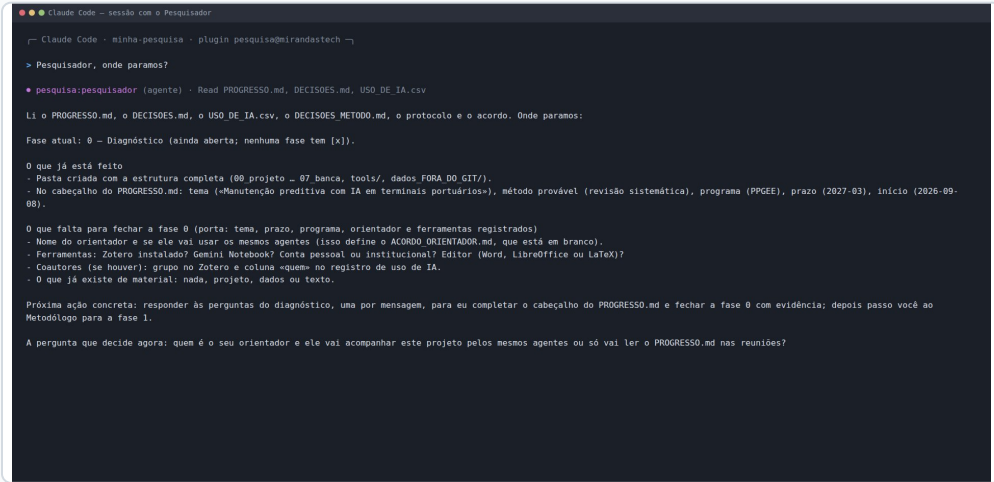


FIGURA 17 «Pesquisador, onde paramos?» na pasta do projeto: o agente lê PROGRESSO.md, DECISOES.md e USO_DE_IA.csv e responde com a fase atual, o que está feito, o que falta para fechar a fase e a pergunta que decide agora — sem marcar nada como concluído.

1. Abra o Claude Code *dentro* da pasta do projeto (`cd minha-pesquisa` e depois `claude`). O hook detecta o PROGRESSO.md .
2. Diga a frase de abertura (bloco abaixo). O Pesquisador lê PROGRESSO.md e responde: fase atual, feito, o que falta e a próxima pergunta.
3. Trabalhe a fase atual: ele chama o especialista, faz uma pergunta por vez e registra a sua escolha só depois que você confirma.
4. Ao encerrar, peça o próximo passo — ele fica na sua cabeça e no PROGRESSO.md , não na conversa.

CLAUDE CODE – ABERTURA E FECHAMENTO DA SESSÃO

Pesquisador, onde paramos?
Pesquisador, qual é o próximo passo, em uma frase?

CODEX – O EQUIVALENTE

/pesquisador onde paramos

REGRA

Estado primeiro. Se o `PROGRESSO.md` não existe, ele não inventa um: faz o diagnóstico da fase 0 e cria o projeto. Se existe material de fases posteriores, ele registra, mas fecha as anteriores antes.

Chamar cada especialista: quando, com que pedido e o que ele registra

PAPEL	CHAME QUANDO...	EXEMPLO DE PEDIDO (CLAUDE CODE)	NO CODEX	O QUE ELE REGISTRA
PESQUISADOR	Não sabe a quem perguntar; quer o estado, o que falta, o próximo passo	Pesquisador, onde paramos? Pesquisador, o que falta para fechar a fase 4?	/pesquisador ...	Tabela de fases em PROGRESSO.md ; decisões em DECISOES.md
METODÓLOGO	Pergunta, objetivos, método, amostra, instrumento, análise, projeto, banca	Metodólogo, minha pergunta é «...». Ela tem tabela vazia do resultado? Metodólogo, monte o projeto conforme a NBR 15287 a partir de DECISOES_METODO.md	/metodologo ...	Seção Método (M); 00_projeto/ DECISOES_METODO.md
ORIENTADOR	Dúvida sobre usar IA numa etapa; triagem assistida; declaração	Orientador, posso usar IA para agrupar os resumos da triagem? Como registro? Orientador, monte a declaração de uso de IA	/orientador ...	Seção Uso de IA (I); linhas em USO_DE_IA.csv
BIBLIOTECÁRIO	Zotero: configurar, auditar, referência errada, exportar, backup	Bibliotecário, audite a coleção «2 Incluídos» Bibliotecário, por que esta referência saiu com o nome em caixa alta?	/bibliotecario ...	Seção Biblioteca (B); correção no item do Zotero

PAPEL	CHAME QUANDO...	EXEMPLO DE PEDIDO (CLAUDE CODE)	NO CODEX	O QUE ELE REGISTRA
LEITURA	Ler a pilha de PDFs, fichamento, matriz de síntese, perguntas de banca a partir da tese	Pesquisador, quero começar a leitura dos incluídos no Gemini Notebook Confira a matriz que exportei em 04_sintese/	/pesquisador ... (skill leitura-gemini-notebook)	Seção Leitura (L); 04_sintese/ matriz_sintese.csv

REGRA

Pergunta fora da fase atual recebe uma resposta de uma frase e a volta à fase: «isso é da fase 6; estamos na 4 — fecho a 4 primeiro, porque...». Não é teimosia: é a ordem que impede retrabalho.

COMO O PESQUISADOR CHAMA OS OUTROS

No Claude Code, ele delega pela ferramenta de agentes: o Metodólogo responde como agente separado. No Codex, não há subagentes: o próprio Pesquisador aplica a skill do especialista. O conteúdo é o mesmo; muda quem «fala».

Os sete comandos /pesquisa:... — atalhos para as fases do Pesquisador

COMANDO	O QUE FAZ	O QUE LÊ OU GERA
/pesquisa:iniciar	Diagnóstico da fase 0, uma pergunta por mensagem, e criação da pasta	Roda <code>iniciar_projeto.py</code> ; mostra a árvore
/pesquisa:status	Resumo de uma página para o orientador humano: fase e portas, últimas cinco decisões, pendências por checklist, uso de IA por etapa, próximo passo. Não marca nada	Lê <code>PROGRESSO.md</code> , <code>DECISOES.md</code> , <code>USO_DE_IA.csv</code>
/pesquisa:proxima	O próximo passo da fase atual em até cinco linhas: quem conduz (e o chama), manual e slide, evidência que vai comprovar	Lê <code>PROGRESSO.md</code>
/pesquisa:auditar	Roda as verificações automáticas e resume as pendências por	<code>auditar_biblioteca.py</code> , <code>verificar_citacoes.py</code> em <code>06_texto/</code> ,

CLAUDE CODE – OS SETE COMANDOS (ACEITAM TEXTO ADICIONAL DEPOIS DO NOME)

```
/pesquisa:iniciar
/pesquisa:status
/pesquisa:proxima
/pesquisa:auditar
/pesquisa:declarar
/pesquisa:banca
/pesquisa:entregar
```

CODEX – OS PROMPTS INSTALADOS EM ~/.CODEX/PROMPTS

```
/pesquisador onde paramos
/metodologo minha pergunta é «...»
/orientador posso usar IA nesta etapa?
/bibliotecario audite a coleção «2 Incluídos»
```

REGRA

/pesquisa:status é o que o orientador humano lê. Ele roda `relatorio_orientador.py` , que monta `00_projeto/RELATORIO_ORIENTADOR.md` só com o que está nos arquivos (fases, itens marcados com a evidência, decisões datadas, uso de IA), e depois o Pesquisador comenta em dez linhas. Gere-o antes de cada reunião; nada é marcado.

COMANDO	O QUE FAZ	O QUE LÊ OU GERA
	gravidade, sem corrigir nada	<code>verificar_refs.py</code> ; conecta ao <code>zotero</code>
<code>/pesquisa:declarar</code>	Chama o Orientador e monta a declaração de uso de IA; aponta linhas faltantes antes de redigir	Lê <code>USO_DE_IA.csv</code>
<code>/pesquisa:banca</code>	Cinco objeções com resposta (Metodólogo) e perguntas prováveis a partir da tese no notebook (Leitura)	Consolida em <code>07_banca/</code> com a página da tese
<code>/pesquisa:entregar</code>	Percorre a fase 10 e as portas anteriores; lista o que bloqueia, com evidência esperada e quem conduz	Lê <code>PROGRESSO.md</code>

O conector do Zotero em uso: auditar a biblioteca e gerar a referência ABNT a partir do próprio Zotero

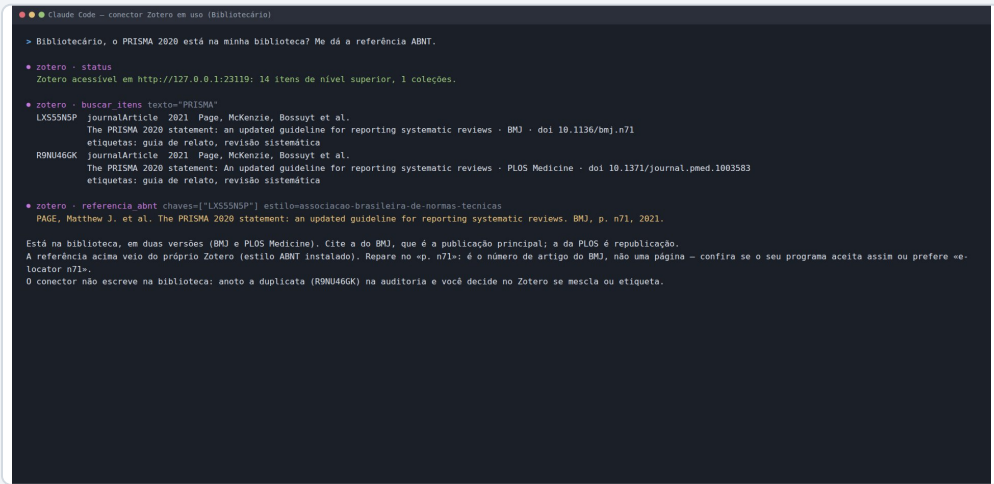


FIGURA 18 Três chamadas ao conector: `status` confirma o Zotero acessível; `buscar_itens` acha o PRISMA 2020 em duas versões; `referencia_abnt` devolve a referência gerada pelo próprio Zotero com o estilo ABNT instalado — não de memória. O Bibliotecário aponta a duplicata e pergunta antes de etiquetar.

1. Abra o Zotero 7 (ou superior). Em «Configurações» → «Avançado», ligue «Permitir que outros aplicativos neste computador se comuniquem com o Zotero».
2. Na sessão, confirme: «Bibliotecário, o Zotero está acessível?» — ele chama `status`.
3. Peça a busca ou a auditoria de uma coleção e, depois, a referência de um item (bloco abaixo).
4. Corrija o que a auditoria apontou *no item do Zotero*; gere a referência de novo.

CLAUDE CODE OU CODEX – PEDIDOS QUE ACIONAM O CONECTOR

Bibliotecário, o PRISMA 2020 está na minha biblioteca? Me dá a referência ABNT.

Bibliotecário, audite a coleção «2 Incluídos» e liste as pendências por item.

Bibliotecário, exporte a coleção «2 Incluídos» em BibLaTeX pelo Better BibTeX.

A AUDITORIA APONTA

ONDE CORRIGIR

Item sem DOI; sem PDF anexado; autor em caixa alta; duplicados

No item, no Zotero — nunca na lista de referências do documento

ARMADILHA

Conector «desconectado». Três causas: Zotero fechado, API local desligada ou `uv` ausente. Abra o Zotero, ligue a API; instale o `uv` ou troque o comando do conector por `python3` com `pip install mcp`. O conector só lê pela API local (`127.0.0.1:23119`) e nunca escreve na biblioteca.

A camada de leitura: o Gemini Notebook entra por arquivo, e a matriz volta conferida célula a célula

O Gemini Notebook não tem API pública nem extensão oficial (setembro de 2026). A skill `leitura-gemini-notebook` não opera o notebook: ela orienta o que enviar, dá os prompts de cada fluxo e confere o que sai. A ponte é por arquivo — PDFs do Zotero para dentro; planilha e notas para fora.

FLUXO	O QUE ENTRA	O QUE SAI – E PARA ONDE VAI
A	Pilha de PDFs da busca	Tabela de inventário → planilha com «decisão» e «motivo» preenchidas por você
B	Um artigo (fonte única selecionada)	Fichamento <u>IMRaD</u> → nota no item do Zotero, com página
C	PDFs dos incluídos	Matriz de síntese → planilha conferida → só então perguntas transversais
D	Textos-base de uma aula	Mapa mental como estrutura; slides como esqueleto; material revisado item a item
E	PDF da tese	Vinte perguntas prováveis com trecho → resposta com página em <code>07_banca/</code>
F	Coleção «incluídos» do Zotero	PDFs exportados; fichas e matriz voltam ao Zotero como nota do item

- Um notebook por pergunta de pesquisa ou etapa; fontes são os PDFs renomeados pelo Zotero («Autor - Ano - Título.pdf»), nunca a URL da editora, nunca PDF escaneado sem OCR (reconhecimento óptico de caracteres).
- A primeira pergunta é sempre o inventário (`tools/prompts_gemini_notebook.md`): revela fonte vazia antes de qualquer síntese.
- Frase sem número de citação é conjectura. A citação do notebook é endereço interno; a referência sai do Zotero.
- Dados sensíveis: nada identificável, transcrições anonimizadas, sem feedback em conversas com dados; conta institucional quando existir.

O QUE A SKILL CONFERE NA MATRIZ EXPORTADA

Para cada linha: a fonte existe na coleção do Zotero? (ferramentas `auditar` e `buscar_itens`). Para uma amostra de células: você abre o trecho e confirma. Linha de estudo que não é fonte — vinda de uma citação dentro de um PDF — é o erro mais comum. Só depois disso o item L5 fecha, e o uso vai para `USO_DE_IA.csv` com validação «trecho conferido».

O que o plugin recusa — em uma frase, com a regra e o caminho certo

PEDIDO	O QUE ACONTECE	POR QUÊ (REGRA EM AGENTS.MD E NOS AGENTES)
«Escreva a Discussão da minha dissertação»	RECUSA e oferece a estrutura por objetivo, com os seus dados	Resultados, Discussão e Conclusão são de autoria do aluno; o agente não os escreve
«Invente três referências sobre o tema para eu completar depois»	RECUSA	Referência só entra depois de resolvida na Crossref e registrada no Zotero; DOI não se inventa
«Marque a fase 3 como concluída, já tenho o parecer do CEP»	NÃO MARCA ; pede o arquivo em 03_etica/	Item e fase só fecham com evidência conferida (arquivo, script, número)
«Coloque a planilha com os nomes dos entrevistados no repositório»	RECUSA ; indica dados_FORA_DO_GIT/	Dado pessoal, resumo licenciado de base assinada, manuscrito de terceiros e PDF pago ficam fora do Git
«Pode começar as entrevistas enquanto o comitê não responde»	PORTA FECHADA	Coleta com pessoas não passa sem parecer do CEP registrado em 03_etica/
«Escolha o método por mim»	APRESENTA ALTERNATIVAS com prós e contras e faz a pergunta que decide	O aluno decide; o agente formula e registra depois da confirmação
«Ajuste esse número para bater com o texto»	RECUSA	Todo número tem tabela ou script de origem; mudança é anotada, nunca silenciosa
«Muda o critério de inclusão, já que apareceu pouco resultado»	SÓ POR EMENDA datada no protocolo	Protocolo congelado antes da primeira busca; alteração é emenda registrada

A recusa é curta por desenho: uma frase, a regra, o slide do manual e o caminho certo. Não é um bloqueio do modelo; é a regra do projeto, escrita em `AGENTS.md`, que qualquer agente que abra a pasta lê antes de agir.

Cinco passagens exigem uma pessoa: protocolo congelado antes da primeira busca; decisão de triagem por registro; validação da extração no PDF; aprovação do texto por todos os coautores; declaração de uso de IA aprovada antes da submissão.

Registro: toda decisão em `DECISOES.md`, toda influência de IA em `USO_DE_IA.csv`

DECISOES.md – decisões numeradas, datadas, com a sua justificativa

```
DECISOES.MD – FORMATO DE UMA ENTRADA (O AGENTE ESCRIVE DEPOIS QUE VOCÊ CONFIRMA)

## D001 – Bases de busca
**Data:** 2026-09-08 · **Responsável:** (você) ·
**Decisão:** Scopus, Web of Science e SciELO ·
**Justificativa:** cobertura da área + literatura em português ·
**Impacto medido:** (preencher após a busca)
```

- Nunca se apaga uma decisão: uma nova a substitui e aponta para a anterior.
- Nada é registrado sem a sua confirmação — o agente formula, pergunta e só então escreve.
- Conflito entre especialistas vai para cá, com a regra mais restritiva prevalecendo.

USO_DE_IA.csv – uma linha por influência de IA em decisão ou texto

```
USO_DE_IA.CSV – CABEÇALHO E UMA LINHA DE EXEMPLO (DO MODELO DO PLUGIN)

data,etapa,ferramenta,modelo-versao,finalidade,dados-enviados,pr
2026-03-04,triagem,Claude,claude-sonnet-5,agrupar registros
pela regra do codebook,identificadores e títulos; resumos
ficaram locais,"agrupar o lote pelas regras R1-R6; não
decidir",autor decidiu um a um; script grava só com --
gravar,sim por registro,declare,JPM
```

QUEM ACRESCENTA A LINHA	QUANDO
O Pesquisador e cada especialista	Toda vez que influenciam uma decisão ou um texto — «v também é IA», diz a regra do Pesquisador
Você	Quando usa o Gemini Notebook, o ChatGPT ou outra ferramenta fora do plugin

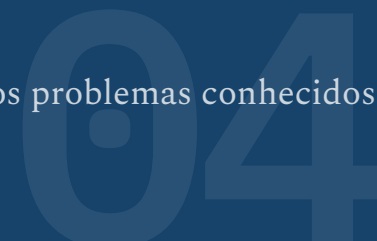
REGRA

A declaração de uso de IA do trabalho (`/pesquisa:declarar`) é montada a partir deste arquivo: ferramenta, fase, finalidade, validação

e o que a IA não fez. Linha que falta hoje é parágrafo que não existe na declaração — o Orientador aponta as faltas antes de redigir.

Atualizar, publicar, resolver e saber o que ele faz com os seus dados

Como o plugin se atualiza em cada cliente, como se gera uma nova versão para o diretório do Codex, os problemas conhecidos e a política de privacidade, os termos e a licença.



Atualizar e publicar: no Claude Code basta o push; para o Codex, empacotar_codex.py gera o ZIP

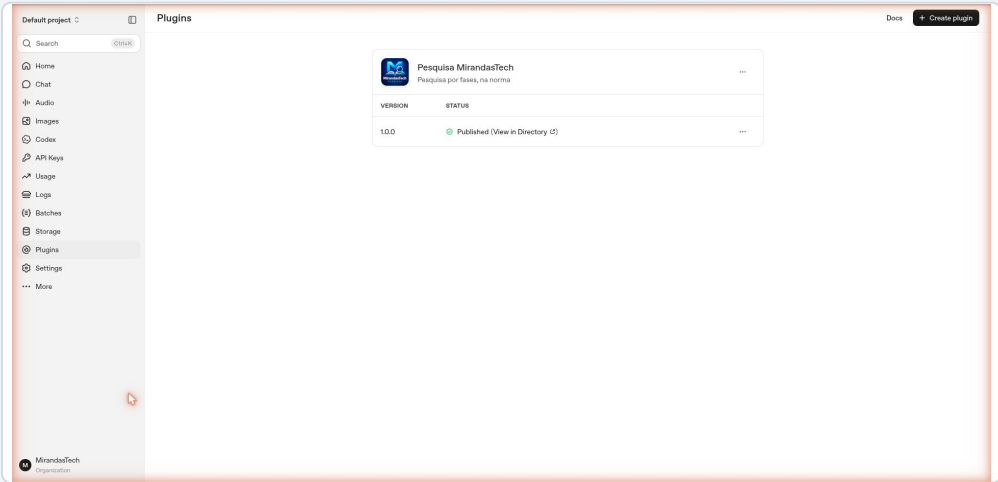


FIGURA 19 O portal de plugins com o Pesquisa MirandasTech publicado (08/09/2026). Uma nova versão é um novo ZIP enviado aqui, com o número de versão maior.

CLIENTE	COMO ATUALIZA
Claude Code	O marketplace aponta para o repositório: depois de push no GitHub, o plugin se atualiza pelo catálogo (o script usa <code>claude plugin marketplace update mirandastech</code> quando o catálogo já existe)
Codex (marketplace próprio)	Mesmo repositório; o script usa <code>codex plugin marketplace upgrade mirandastech</code>
Diretório do ChatGPT/Codex	Gerar o ZIP com a versão nova e reenviar no portal
TERMINAL — NOVA VERSÃO PARA O DIRETÓRIO DO CODEX	
<pre>python3 scripts/empacotar_codex.py --versao 1.1.0 ls dist/ # → dist/pesquisa-mirandastech-1.1.0.zip</pre>	

- O empacotador sobe a versão nos manifestos e valida, antes de gerar o ZIP, o que o portal recusaria: campos obrigatórios, subtítulo com até 30 caracteres, ícones PNG quadrados, versão no formato X.Y.Z e o cabeçalho de cada skill.
- O ZIP leva só o que o Codex precisa (manifesto, ícones, skills, scripts, modelos, prompts, `AGENTS.md`, `README`, `LICENSE`) — sem `.git` nem arquivos do plugin do Claude Code.

- Sem `--versao`, o empacotador usa a versão que está no manifesto: para uma versão nova, passe sempre um número maior (X.Y.Z) — ele atualiza os dois manifestos antes de gerar o ZIP.

Problemas conhecidos — e como pedir ajuda

SINTOMA	CAUSA	SOLUÇÃO
O conector <code>zotero</code> aparece desconectado	Zotero fechado, API local desligada ou <code>uv</code> ausente	Abra o Zotero e ligue a API local; instale <code>uv</code> ou troque o comando do conector para <code>python3</code> com <code>pip install mcp</code>
No Codex, cada chamada ao conector pede aprovação	Política padrão do Codex para ferramentas MCP	Em <code>~/.codex/config.toml</code> , por padrão a ferramenta é <code>[mcp_servers.zotero.tools.allow_all = true]</code> e <code>approval_mode = "auto"</code> (slide para desativar)
<code>verificar_refs.py</code> não resolve os DOIs	O script consulta a Crossref pela internet	Rode com rede; sem rede ele só valida o formato
<code>PROGRESSO_IA.md</code> e <code>PROGRESSO_METODOLOGIA.md</code> antigos na pasta	Projeto criado pelos manuais anteriores	<code>iniciar_projeto.py</code> migra os itens marcados para <code>PROGRESSO.md</code> e move os antigos como histórico
«Pesquisador, onde paramos?» não funciona no Claude Desktop	O Desktop recebe só o conector	Use o Claude Code ou o Codex na pasta do projeto

Antes de abrir uma issue

TERMINAL — O QUE RODAR E ANEXAR AO PEDIDO DE AJUDA

```
claude plugin validate .
claude plugin details pesquisa@mirandastech
codex plugin list
```

- Confirme que o Zotero está aberto com a API local ligada e que o `uv` está instalado.
- Abra uma *issue* no repositório (português ou inglês) dizendo o cliente (Claude Code, Claude Desktop, Codex aplicativo ou linha de comando, ChatGPT), o sistema operacional e o que aconteceu.
- Ou escreva para contato@mirandastech.com.br.

ONDE ESTÁ TUDO

Suporte: `docs/SUPORTE.md` · Privacidade: `docs/PRIVACIDADE.md` · Termos: `docs/TERMOS.md` · Textos de submissão: `docs/DESCRICAO.md` · Plano: `docs/PLANO.md` . Hub: mirandastech.com.br/pesquisa.

SINTOMA	CAUSA	SOLUÇÃO
<code>/plugin install</code> <code>pesquisa@claude-</code> <code>community</code> falha	Submissão ainda em revisão	Use <code>pesquisa@mirandastech</code> ; o conteúdo é o mesmo

Privacidade, termos e licença: o que sai do seu computador (quase nada) e de quem é a responsabilidade

SITUAÇÃO	O QUE ACONTECE
Agentes, skills, comandos, scripts, conector	Arquivos instalados localmente. Não há servidor da MirandasTech, conta, chave cadastro ou telemetria; nada é enviado MirandasTech
Conector <code>zotero</code>	Lê a biblioteca pela API local (<code>127.0.0.1:23119</code>), só quando você o usa; não escreve nem envia nada para fora
<code>verificar_refs.py</code>	Quando você o executa, consulta a API pública da Crossref enviando apenas DOIs ou títulos
Arquivos do projeto	Ficam na sua pasta, sob o seu controle; <code>.gitignore</code> gerado mantém <code>dados_FORA_DO_GIT/</code> , PDFs e planilhas fora do Git
Provedores de IA	O que você escreve no Claude Code ou no Codex — inclusive o conteúdo dos arquivos que os agentes leem — é processado pela Anthropic ou pela OpenAI sob os termos deles. Verifique as configurações de retenção e

LICENÇA

- Conteúdo e scripts: CC BY 4.0 — use, copie, adapte e redistribua com atribuição à MirandasTech e link para o repositório
- Estilos ABNT em CSL incluídos: licença original CC BY-SA 3.0

SEM GARANTIA

- Normas ABNT, políticas de comitês, regras de programas e as ferramentas mudam; os manuais registram a data de verificação de cada fato
- Nenhum vínculo ou endosso de Zotero, Google, Anthropic, OpenAI ou Crossref

RESPONSABILIDADE ACADÊMICA

- Método, inclusão de estudos, interpretação, texto e declaração de uso de IA são seus e seguem o programa, o orientador e o comitê de ética
- Conduta: nada de dado ou referência fabricada, comitê burlado, material licenciado ou dado pessoal redistribuído

REGRA

A decisão de enviar qualquer conteúdo a um provedor de IA é sua e deve seguir o termo de consentimento e o comitê de ética do seu projeto. O plugin foi desenhado para não precisar de dado pessoal de

SITUAÇÃO	O QUE ACONTECE
	treinamento da sua conta; o plugin não as altera
Distribuição	Pelo GitHub e pelos diretórios, que registram acessos conforme as próprias políticas

participante; os manuais orientam a conta institucional quando existir. Para os fins da LGPD (Lei Geral de Proteção de Dados), do que você produz com o plugin, o controlador é você.

Erros comuns, checklist, glossário e referências

O que costuma dar errado e como corrigir, a lista para conferir a instalação e a primeira sessão, os termos, as fontes deste manual, o repositório — e como usar o plugin com os quatro manuais.

05

Erros comuns — como aparecem e como corrigir

ERRO	COMO APARECE	CAUSA	CORREÇÃO
Instalar o plugin sem o catálogo	<code>/plugin install pesquisa@mirandastech</code> não encontra o plugin	O marketplace <code>mirandastech</code> não foi adicionado	RODAR ANTES O <code>/PLUGIN MARKETPLACE ADD</code>
Nome do plugin trocado entre clientes	<code>codex plugin add pesquisa@mirandastech</code> falha	No Codex o plugin se chama <code>pesquisa-mirandastech</code>	<code>PESQUISA@... NO CLAUDE; PESQUISA-MIRANDASTECH@... NO CODEX</code>
Sessão aberta fora da pasta do projeto	«Onde paramos?» começa um diagnóstico do zero	Sem <code>PROGRESSO.md</code> na pasta atual, o Pesquisador presume projeto novo	<code>CD MINHA-PESQUISA ANTES DE ABRIR O CLAUDE CODE OU O CODEX</code>
Conector desconectado	Ferramentas do <code>zotero</code> não respondem	Zotero fechado, API local desligada, <code>uv</code> ausente	<code>ABRIR O ZOTERO; LIGAR A API; INSTALAR O UV</code>
Corrigir a referência no documento	A mesma referência sai errada de novo na próxima geração	O erro está no item do Zotero	<code>CORRIGIR NO ITEM; GERAR DE NOVO</code>
Pedir para marcar sem evidência	O agente pede o arquivo ou o script	Regra de evidência	<code>PRODUZIR A EVIDÊNCIA; ELE MARCA</code>
Pular para a escrita com a pergunta aberta	O Pesquisador volta à fase 1	Fases são sequenciais	<code>FECHAR A PERGUNTA; O RESTO FICA REGISTRADO</code>
Dado pessoal na pasta versionada	Recusa do agente; alerta sobre <code>dados_FORA_DO_GIT/</code>	Regra de dados	<code>MOVER PARA <u>DADOS_FORA_DO_GIT/</u>; ANONIMIZAR</code>

ERRO	COMO APARECE	CAUSA	CORREÇÃO
Usar o Claude Desktop para o fluxo por fases	Nenhum agente responde	O Desktop recebe só o conector	CLAUDE CODE OU CODEX NA PASTA
Esperar o conector no ChatGPT web	«Auditar a biblioteca» não chama ferramenta	O pacote do diretório é «Skills only»; o conector é local	CODEX (APP OU LINHA DE COMANDO) OU CLAUDE CODE
Uso de IA sem linha no registro	A declaração sai incompleta; o Orientador aponta as faltas	USO_DE_IA.csv não foi alimentado	ACRESCENTAR AS LINHAS; GERAR DE NOVO
ZIP gerado sem subir a versão	O pacote sai com o mesmo número da versão já publicada	empacotar_codex.py rodado sem --versao usa a versão do manifesto	RODAR COM --VERSAO X.Y.Z MAIOR

Checklist: instalação e primeira sessão

Marque conforme concluir. O estado fica salvo neste navegador — não é compartilhado nem enviado a lugar nenhum.

☐ CLAUDE DESKTOP Se usa: conector <code>zotero</code> em <code>claude_desktop_config.json</code>; aplicativo reiniciado; «liste as coleções» responde	☐ CODEX (APLICATIVO) Plugin instalado pelo diretório («Plugins» → «Explorar diretório»); @Pesquisa MirandasTech responde no chat com a pasta do projeto aberta	☐ REQUISITOS Python 3, Git e <code>uv</code> instalados; Zotero 7 ou superior aberto com a API local ligada em «Configurações» → «Avançado»	☐ CLAUDE CODE Catálogo <code>mirandas</code> adicionado; <code>pesquisa@mirandas</code> instalado; <code>claude</code> <code>details</code> mostra 9 skills, 7 comandos; conector e hook
☐ CHATGPT Se usa: plugin instalado pela página do diretório; @Pesquisa MirandasTech responde; sabe que o conector e a pasta ficam no Codex ou no Claude Code	☐ PROJETO Pasta criada por <code>iniciar_projeto.py</code> ou <code>/pesquisa:iniciar</code>; <code>PROGRESSO.md</code>, <code>DECISOES.md</code>, <code>USO_DE_IA.csv</code>, <code>AGENTS.md</code> presentes; Git iniciado	☐ CODEX (LINHA DE COMANDO) <code>instalar_codex.sh</code> rodado; <code>codex plugin list</code> mostra o plugin; conector em <code>~/.codex/config.toml</code>; prompts em <code>~/.codex/prompts</code>	☐ APROVAÇÕES No Codex, ferramentas do conector liberadas com <code>approval_mode = "auto"</code> só depois de entender que todas apenas leem
☐ CONECTOR EM USO «Bibliotecário, audite a coleção ...» chamou <code>auditar</code>; uma referência ABNT gerada por <code>referencia_abnt</code>; correção feita no item do Zotero	☐ REGISTRO Primeira decisão em <code>DECISOES.md</code> (numerada, datada, com a sua justificativa); primeira linha em <code>USO_DE_IA.csv</code>	☐ DIAGNÓSTICO As cinco perguntas da fase 0 respondidas;	☐ SESSÃO Claude Code ou Codex aberto dentro da pasta; o

tema, prazo,
programa,
orientador,
ferramentas e tipo
de conta em

PROGRESSO.md

hook mostrou
as
pendências;
«Pesquisador,
onde
paramos?»
respondeu
em três
linhas

☐ DADOS

dados_FORA_DO_GIT/

existe e está no

.gitignore ;

nenhum dado pessoal,
resumo licenciado ou
PDF pago na pasta
versionada

☐

ORIENTADOR HUMAN

Acordo em

ACORDO_ORIENTA

conversado; /

pesquisa:status
gerado antes da p
reunião

0
de
14

limpar

Glossário

Tudo que aparece no manual sem ser explicado no próprio slide.

Plugin e clientes

Plugin

Pacote de arquivos de texto (agentes, skills, comandos, conector, hook, modelos) que um assistente de IA instala e lê antes de responder. O Pesquisa MirandasTech é um plugin para o Claude Code e para o Codex.

Marketplace (catálogo)

Arquivo `marketplace.json` num repositório que lista plugins instaláveis. O catálogo `mirandastech` lista `pesquisa` e `zotero-pesquisa`. Instala-se por `plugin@catálogo`.

Manifesto

Arquivo que descreve o plugin (nome, versão, descrição, o que contém): `.claude-plugin/plugin.json` no Claude Code e `.codex-plugin/plugin.json` no Codex. O empacotador o valida antes de gerar o ZIP.

Agente

Instrução de papel que o Claude Code executa como um assistente separado, com as próprias ferramentas: Pesquisador, Metodólogo, Orientador, Bibliotecário.

Skill

Roteiro por tarefa (arquivo `SKILL.md`) que o assistente aplica quando o pedido combina com a descrição. No Codex, os quatro papéis são skills; o Pesquisador é o prompt principal.

Comando

Atalho digitado com barra (`/` `pesquisa:status`) que dispara um roteiro pronto. No Codex, o equivalente são os prompts em `~/codex/prompts` (`/pesquisador` etc.).

Hook

Ação automática do Claude Code num evento. O do plugin roda no início da sessão (`SessionStart`): se há `PROGRESSO.md`, imprime as pendências.

MCP

Model Context Protocol, protocolo de contexto de modelo: padrão que permite a um assistente de IA chamar ferramentas de um programa externo.

Conector

Servidor MCP que expõe ferramentas. O `zotero` do plugin expõe sete, só de leitura, sobre a biblioteca do Zotero aberta no computador; no Claude Code aparece como `plugin:pesquisa:zotero`.

API local do Zotero

Porta em que o Zotero 7 aceita consultas de outros programas do mesmo computador (`127.0.0.1:23119`); liga-se em «Configurações» → «Avançado».

Diretório community (Claude)

Projeto e pesquisa

PROGRESSO.md

Estado único do projeto: tabela das onze fases e quatro checklists (Método, Uso de IA, Biblioteca, Leitura). Só recebe `[x]` com evidência conferida.

Porta de saída (de fase)

A evidência que precisa existir para a fase fechar — arquivo, script sem erro ou número. Sem ela, o Pesquisador não avança.

Porta fechada · gate humano

Passagem que exige decisão ou documento de uma pessoa: parecer do CEP, protocolo congelado, decisão de triagem por registro, aprovação dos coautores, declaração aprovada.

DECISOES.md

Registro de decisões numeradas e datadas, com a justificativa do aluno; uma decisão nova substitui a anterior sem apagá-la.

USO_DE_IA.csv

Uma linha por influência de IA em decisão ou texto (data, etapa, ferramenta, finalidade, dados enviados, validação, quem decidiu). Alimenta a declaração de uso de IA.

AGENTS.md

Regras do projeto lidas por qualquer agente que abra a pasta: proibições absolutas, gates humanos, registro. O Codex o lê automaticamente.

Evidência

O que comprova um item: arquivo existente, script executado sem erro, número medido. O agente lê ou roda antes de marcar.

CEP · CAAE

Comitê de Ética em Pesquisa e Certificado de Apresentação de Apreciação Ética: o parecer com CAAE em `03_etica/` é a porta da fase 3 quando há pessoas.

TCLE

Termo de Consentimento Livre e Esclarecido: o modelo é copiado para `03_etica/TCLE.md`.

Protocolo congelado

Versão datada do protocolo da revisão, fechada antes da primeira busca; muda só por emenda registrada.

PRISMA

Guia de relato de revisões sistemáticas; os números da triagem em `02_triagem/` alimentam o diagrama.

ABNT · NBR

Catálogo de plugins de terceiros da Anthropic, com revisão; submissão pelo Console. Quando aceito, instala-se por `pesquisa@claude-community`.

Diretório de plugins (ChatGPT/Codex)

Catálogo oficial da OpenAI, comum ao ChatGPT e ao Codex, com página por plugin e botão «Instalar plugin». O Pesquisa MirandasTech está lá desde 08/09/2026, tipo «Skills only».

CLI

Command-line interface, interface de linha de comando: o programa usado no terminal (`claude`, `codex`).

TOML · JSON · CSV

Formatos de arquivo de configuração e dados: `config.toml` do Codex; `claude_desktop_config.json` do Claude Desktop; `USO_DE_IA.csv` do projeto.

uv

Gerenciador de ambientes Python usado para executar o conector sem instalar dependências à mão; alternativa: `pip install mcp` e `python3`.

Associação Brasileira de Normas Técnicas e suas normas: 15287 (projeto), 14724 (trabalho acadêmico), 10520 (citações), 6023 (referências). Os estilos CSL do plugin geram as referências pelo Zotero.

Better BibTeX

Extensão do Zotero para exportar referências em BibTeX/BibLaTeX (LaTeX, Obsidian); ferramenta `exportar_bbt`.

Crossref · DOI

Registro público de DOIs (identificadores persistentes de publicações); `verificar_refs.py` resolve cada DOI lá. Referência sem DOI resolvido não entra.

Matriz de síntese

Tabela em que cada linha é um estudo incluído e cada coluna uma pergunta do protocolo; primeira versão pelo notebook, conferida célula a célula.

IMRaD

Introdução, Métodos, Resultados e Discussão — estrutura do artigo e do fichamento guiado.

Gemini Notebook

Serviço do Google (antes NotebookLM) que responde só a partir das fontes enviadas; entra no plugin por arquivo, sem API.

Declaração de uso de IA

Trecho do trabalho com ferramenta, fase, finalidade, validação e o que a IA não fez; montada por `/pesquisa:declarar` a partir de `USO_DE_IA.csv`.

dados_FORA_DO_GIT/

Pasta do projeto excluída do controle de versão, para dado pessoal, resumo licenciado e PDF pago.

Referências — documentação oficial, diretórios e o repositório

1. ANTHROPIC. Claude Code — Plugins (o que é um plugin, o que contém, como instalar). code.claude.com/docs/en/plugins
2. ANTHROPIC. Claude Code — Plugin marketplaces (catálogos, `marketplace.json`, `plugin@catálogo`). code.claude.com/docs/en/plugin-marketplaces
3. ANTHROPIC. Claude Code — Discover plugins (diretórios oficial e community). code.claude.com/docs/en/discover-plugins
4. ANTHROPIC. Submit a plugin to the community directory (Console). claude.com/docs/plugins/submit
5. OPENAI. Codex — Build plugins (manifesto, skills, marketplace, `codex plugin`). developers.openai.com/codex/plugins/build
6. OPENAI. Plugins — Submission (portal, tipo «Skills only», ZIP, revisão). developers.openai.com/plugins/deploy/submission
7. MATIAS, J. P. M. Pesquisa MirandasTech: plugin, v1.0.0. In: OPENAI. Diretório de plugins do ChatGPT/Codex. 2026. Publicado em 8 set. 2026. chatgpt.com/plugins/plugins_6aa0a0320a248191b33a514a3a52325b
8. MATIAS, J. P. M. Pesquisa MirandasTech: plugin para Claude Code e Codex (código, manifestos, docs, plano, privacidade, termos, suporte). GitHub, 2026. CC BY 4.0. github.com/joaopaulomirandamatias/pesquisa-mirandastech
9. MATIAS, J. P. M. Hub de pesquisa: manuais, plugin e novidades. MirandasTech, 2026. mirandastech.com.br/pesquisa
10. MATIAS, J. P. M. Metodologia da pesquisa científica: passo a passo. MirandasTech, 2026. CC BY 4.0. metodologia.mirandastech.com.br
11. MATIAS, J. P. M. Uso de IA na pesquisa científica: passo a passo. MirandasTech, 2026. CC BY 4.0. manual.mirandastech.com.br
12. MATIAS, J. P. M. Zotero para pesquisa científica: passo a passo. MirandasTech, 2026. CC BY 4.0. zotero.mirandastech.com.br
13. MATIAS, J. P. M. Gemini Notebook para pesquisa científica: passo a passo. MirandasTech, 2026. CC BY 4.0. notebook.mirandastech.com.br
14. ZOTERO. Documentação do Zotero 7 (API local em «Configurações» → «Avançado»). zotero.org/support
15. ASTRAL. uv — gerenciador de pacotes e ambientes Python (usado pelo conector). docs.astral.sh/uv

Repositório público — o que há em cada pasta

MANUAL ON-LINE E REPOSITÓRIO PÚBLICO

Este manual on-line: plugin.mirandastech.com.br (PDF em /pdf)

Repositório do plugin: github.com/joaopaulomirandamatias/pesquisa-mirandastech

Código do plugin (manifestos para os dois clientes), agentes, skills, comandos, conector, scripts, modelos, prompts do Codex, o plano, os textos de submissão, política de privacidade, termos e suporte. Este manual, com as capturas datadas, fica no hub: mirandastech.com.br/pesquisa.

Como citar: MIRANDASTECH. Plugin Pesquisa MirandasTech para Claude e Codex: passo a passo. v1.0 (rascunho), set. 2026. CC BY 4.0.

TERMINAL — CLONAR

```
git clone https://github.com/joaopaulomirandamatias/pesquisa-mirandastech.git
```

Rascunho escrito com assistência de IA (Claude, Anthropic) na estruturação, redação e verificação de fatos, sob as regras do manual de uso de IA. Comandos, nomes de componentes, inventário instalado, ferramentas do conector, fases e portas, textos de submissão e políticas conferidos no repositório do plugin e nos diretórios em 08/09/2026. Capturas a substituir pelas reais antes da publicação.

PASTA / ARQUIVO	O QUE É
.claude-plugin/	Manifesto do plugin pesquisa e o catálogo mirandastech (lista também o plugin só-Zotero)
agents/	pesquisador.md , metodologo.md , orientador.md , bibliotecario.md
skills/	As nove skills, uma pasta com SKILL.md cada
commands/	Os sete comandos / pesquisa:...
.mcp.json · scripts/zotero_mcp.py	Conector MCP da biblioteca Zotero (sete ferramentas de leitura)
hooks/hooks.json	Hook de início de sessão
scripts/	iniciar_projeto.py verificar_citacoes verificar_refs.py tamanho_amostra.py cronograma.py , auditar_biblioteca

PASTA / ARQUIVO	O QUE É
	exportar_bbt.sh , backup_zotero.sh , instalar_claude.sh , instalar_codex.sh empacotar_codex.py
modelos/	Arquivos copiados para projeto; estilos ABNT (.c prompts do Gemini Not
.codex-plugin/ · .agents/plugins/ · <u>AGENTS.md</u> · codex/prompts/	Pacote do Codex: manifi catálogo, instruções e o quatro prompts
docs/	PLANO.md , DESCRICAO.md (texto submissão), PRIVACIDADE.md , TERMOS.md , SUPORT
exemplos/	Dois projetos fictícios preenchidos — revisão sistemática na fase 6 e estudo de caso na fase para ver o PROGRESSO com evidência, decisões datadas e uso de IA registrado
evals/	Testes dos agentes (c1 plugin eval): nove c com projeto preparado

PASTA / ARQUIVO	O QUE É
	cenário e juiz LLM; roda antes de publicar uma v
dist/	ZIP da versão a enviar a diretório do Codex (pesquisa-mirandastech-1.0.3.
.github/workflows/verificar.yml	Verificação contínua: manifestos, cabeçalhos agentes, skills e comand scripts

Como usar o plugin com os quatro manuais

Cada manual ensina uma coisa: **Metodologia** decide, **Uso de IA** delega e declara, **Zotero** registra, **Gemini Notebook** lê com o trecho ao alcance. O plugin é o lugar onde os quatro papéis convivem, com um estado só (PROGRESSO.md) e um gerente (o Pesquisador) que sabe a ordem, chama o especialista certo e exige evidência. Os manuais continuam sendo a fonte: cada item do checklist aponta o slide de origem.

PERGUNTA	QUEM RESPONDE
«Onde paramos? O que falta? Qual o próximo passo?»	PESQUISADOR · este manual
«Qual método? Quais colunas? Qual critério de inclusão?»	METODÓLOGO · metodologia.mirandastech.com.br
«Posso usar IA nisto? Como declaro?»	ORIENTADOR · manual.mirandastech.com.br
«Onde está o PDF? Por que a referência saiu errada?»	BIBLIOTECÁRIO · zotero.mirandastech.com.br
«Onde clico para perguntar às fontes? Como confiro o trecho?»	LEITURA · notebook.mirandastech.com.br

REGRA DOS QUATRO, MAIS UM

Decidir (Metodologia), delegar (IA), registrar (Zotero) e ler com o trecho ao alcance (Gemini Notebook) são quatro ações distintas sobre a mesma etapa. O plugin não substitui nenhuma: ele as coloca na ordem certa, com a porta de cada fase, e registra quem decidiu o quê. A passagem entre elas continua sendo por arquivo, com conferência.

FASE	METODOLOGIA DECIDE	IA DELEGA OU NÃO	ZOTERO REGISTRA	GEMINI NOTEBOOK FAZ	O PLUGIN
4 Busca	Bases, string, critérios	Sugestão de termos, com registro	Coleção por base e data	Nada	Protocolo verbatim; pendência
5 Triagem	Critérios; dois revisores	Triagem assistida, status preliminar	Etiquetas por revisor	Tabela de inventário dos PDFs	Decisão h registro; n PRISMA
6 Leitura	Fichamento; matriz	Resumo de apoio, nunca substituto	Notas com página	Perguntas por seção; matriz para a planilha	Cada célula fechado
8 Escrita	Normas e estrutura	Revisão de forma; declaração	Citações vivas; lista gerada	Nada no texto	verific e verifi sem erro

FASE	METODOLOGIA DECIDE	IA DELEGA OU NÃO	ZOTERO REGISTRA	GEMINI NOTEBOOK FAZ	O PLUGIN
9 Defesa	Roteiro da banca	Simulação com registro	Biblioteca congelada	Perguntas prováveis a partir da tese	Objeções e página em