MIRANDASTECH · MANUAL PROFISSIONAL

TRILHA 03 · DADOS E FERRAMENTAS · MANUAL 3.4

# 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 | VERSÃO | FORMATO | LICENÇA |
|---|---|---|---|
| Manuais MirandasTech · ferramentas de pesquisa | 1.0 (rascunho) · setembro de 2026 | – slides | 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 Laboratório reúne todos: lab.mirandastech.com.br.

INTRODUÇÃO 1 DE 3 · INVENTÁRIO CONFERIDO NO REPOSITÓRIO EM 08/09/2026

# 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 `PROGRESSO.md`, chama o especialista certo e só fecha uma fase com evidência. É o papel padrão — quem não sabe a quem perguntar, pergunta a ele. | Claude Code (agente) · Codex (prompt principal e skill) |
| **Metodólogo** (agente) | Pergunta, objetivos, classificação, desenho, amostra, instrumento, análise; projeto conforme a NBR 15287:2025; banca simulada. | Claude Code · Codex (skill) |
| **Orientador** (agente) | O que pode ser delegado à IA em cada etapa, registro em `USO_DE_IA.csv`, triagem assistida com status preliminar, declaração de uso. | Claude Code · Codex (skill) |
| **Bibliotecário** (agente) | Zotero: configuração, auditoria da biblioteca, referência ABNT gerada pelo próprio Zotero, exportação Better BibTeX. | Claude Code · Codex (skill) |
| **Leitura** (skill) | Fluxos e prompts do Gemini Notebook; conferência da matriz de síntese célula a célula. Sem API: a ponte é por arquivo. | Claude Code · Codex |
| **Conector MCP** `zotero` | 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 |

| COMPONENTE | O QUE FAZ | ONDE ENTRA |
|---|---|---|
| **Sete comandos** `/pesquisa:…` | iniciar, status, proxima, auditar, declarar, banca, entregar — atalhos para as fases do Pesquisador. | Claude Code (no Codex, prompts `/pesquisador` etc.) |
| **Hook de início de sessão** | Ao abrir uma sessão numa pasta com `PROGRESSO.md`, 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, TCLE, matriz de síntese, 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? | 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

| PERGUNTA QUE VOCÊ FAZ | QUEM RESPONDE | QUEM DECIDE |
| --- | --- | --- |
| Como declaro?» | | |
| «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 |

> `DECISOES.md`. Só o Pesquisador escreve na tabela de fases; cada especialista marca os itens da própria seção.

# Roteiro: seis 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

### Demonstração

Uma pesquisa começando do zero no ChatGPT, em quatro passos gravados: diagnóstico, fechamento da fase 0, pergunta com o Metodólogo, retomada sem pasta

Parte 5

### Manter

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

Parte 6

### Fechamento

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

COMO LER

CONVENÇÕES

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 mostra uma sessão real, do início ao fim, em animações. A Parte 5 é para quem mantém o plugin ou quer entender o que ele faz com os seus dados.

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 e animações em 1200 × 585 (Parte 4), feitas em setembro de 2026, com a interface em português.

PARTE 1 INSTALAR NO CLAUDE CODE

# 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).

01

# Claude Code: adicionar o marketplace e instalar o plugin

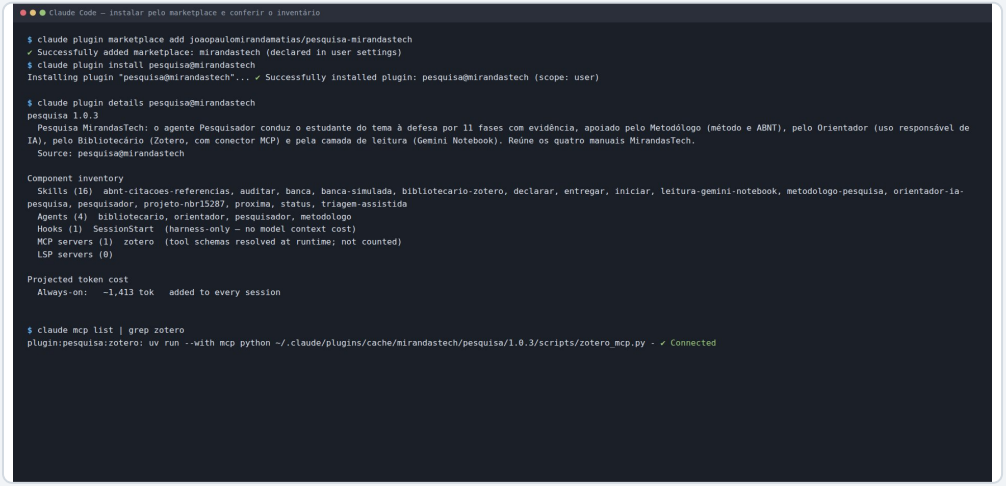



FIGURA 1 No Claude Code, os dois comandos: primeiro o catálogo («marketplace») do repositório, depois o plugin `pesquisa` desse catálogo.

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 |
| `uv` | Executa o conector do Zotero (alternativa: `pip install mcp` e trocar o comando por `python3`) |
| **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).

CLAUDE CODE 2 DE 4

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

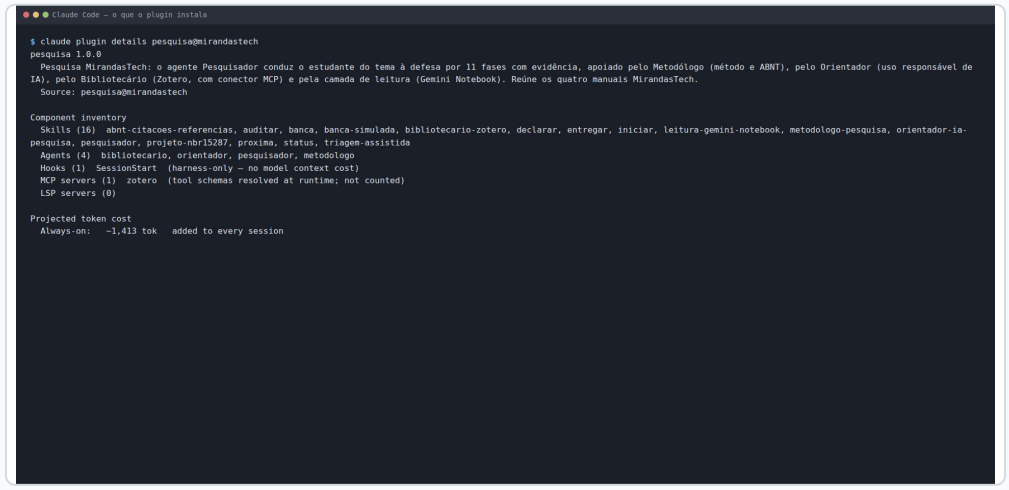



FIGURA 2 `claude plugin details pesquisa@mirandastech` mostra o inventário do plugin: quatro agentes, nove skills, sete comandos, o conector `zotero` e o hook de início de sessão.

TERMINAL — INVENTÁRIO E VALIDAÇÃO

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

| TIPO | O QUE ENTROU |
|---|---|
| **4 agentes** | `pesquisador`, `metodologo`, `orientador`, `bibliotecario` |
| **9 skills** | `pesquisador`, `metodologo-pesquisa`, `projeto-nbr15287`, `abnt-citacoes-referencias`, `banca-simulada`, `orientador-ia-pesquisa`, `triagem-assistida`, `bibliotecario-zotero`, `leitura-gemini-notebook` |
| **7 comandos** | `/pesquisa:iniciar` · `status` · `proxima` · `auditar` · `declarar` · `banca` · `entregar` |
| **1 conector MCP** | `zotero`, que aparece como `plugin:pesquisa:zotero`; ferramentas `status`, `listar_colecoes`, `buscar_itens`, `detalhar_item`, `referencia_abnt`, `auditar`, `exportar_bbt` |
| **1 hook** | `SessionStart`: se a pasta tem `PROGRESSO.md`, imprime fase atual, checklists (M, I, B, L), última decisão datada e há quantos dias o arquivo não |

| TIPO | O QUE ENTROU |
| --- | --- |
| | muda, e lembra: «Diga: Pesquisador, onde paramos?» |

> VALIDAR UM CLONE
>
> `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

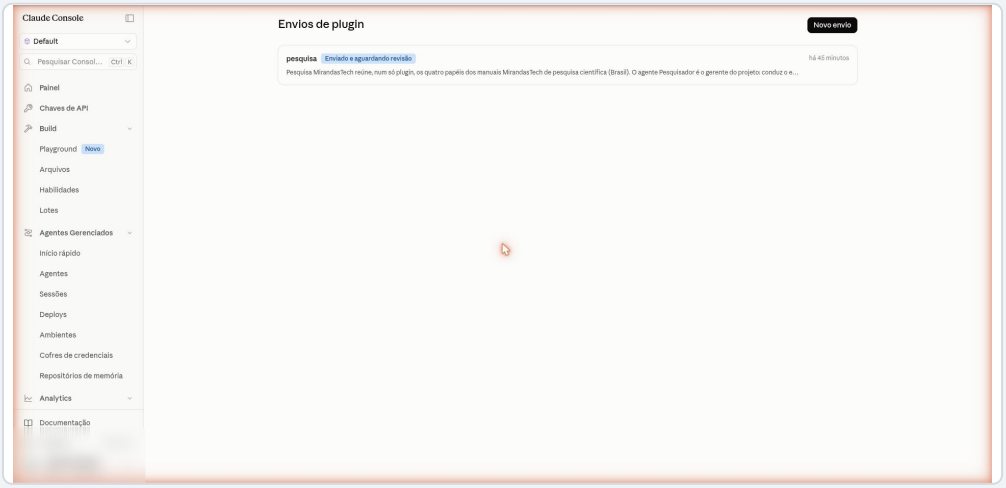



FIGURA 3 Console (platform.claude.com/plugins/submissions): o envio «pesquisa», feito em 08/09/2026, com o status «Enviado e aguardando revisão». O marketplace oficial da Anthropic não aceita submissão de terceiros; o caminho é o diretório community.

TERMINAL — CLONAR E INSTALAR COM O SCRIPT

```
git clone https://github.com/joaopaulomirandamatias/pesquisa-mirandastech.git
cd pesquisa-mirandastech
bash scripts/instalar_claude.sh
```

| CAMINHO | ESTADO | COMANDO |
|---|---|---|
| **Marketplace próprio** | DISPONÍVEL | `/plugin install pesquisa@mirandastech` |
| **Script pelo terminal** | DISPONÍVEL | `bash scripts/instalar_claude.sh` |
| **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 CODE 4 DE 4 · CLAUDE DESKTOP

# 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

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` |

usar dentro de qualquer conversa. O fluxo por fases é no Claude Code (inclusive na aba «Code» do próprio Desktop) ou no Codex.

ABA «CODE» DO DESKTOP

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.

PARTE 2 INSTALAR NO CODEX E NO CHATGPT

# Diretório oficial, marketplace próprio e a @menção

02

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.

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

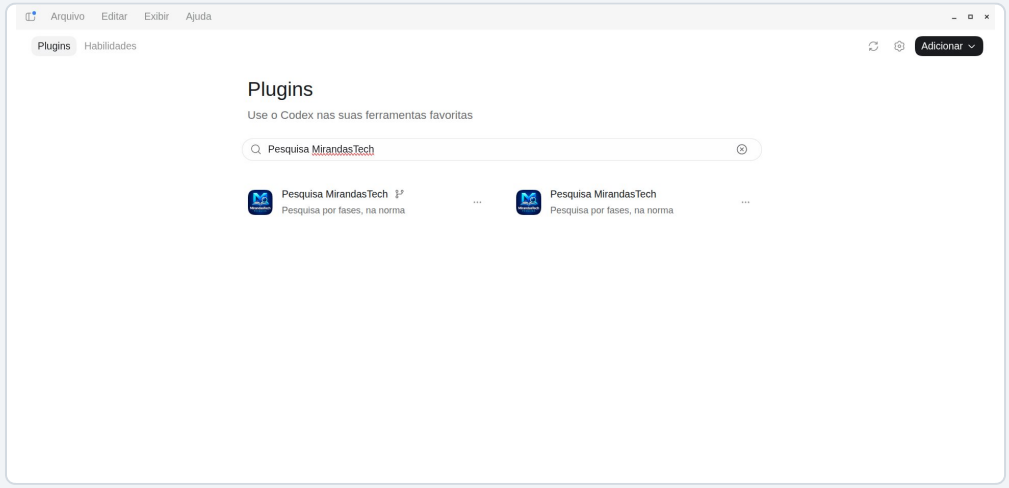



FIGURA 4 «Plugins» → caixa «Procurar plugins» → «Pesquisa MirandasTech». Aparecem dois resultados: o do marketplace próprio (ícone de ramificação, instalado pelo comando da linha de comando) e o do diretório oficial. Para quem só usa o aplicativo, vale o segundo.

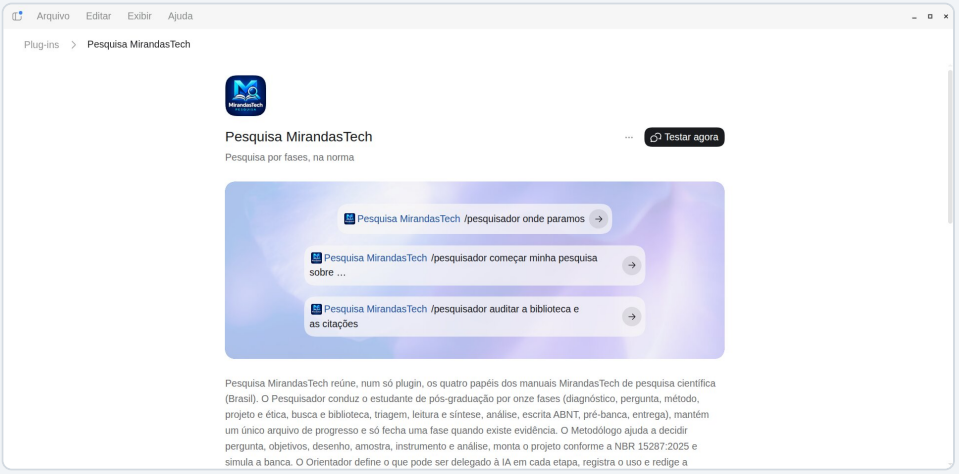



FIGURA 5 A página do plugin: subtítulo «Pesquisa por fases, na norma», os três pedidos de partida e a descrição. O botão principal instala; já instalado, ele vira «Testar agora» e abre o chat com a menção pronta.

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

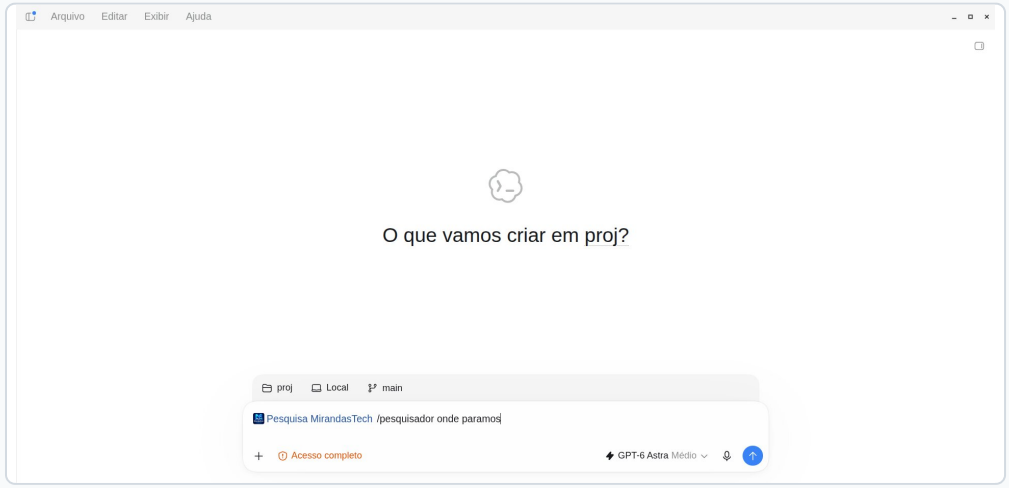



FIGURA 6 A caixa de mensagem com a pasta do projeto escolhida (etiquetas «proj · Local · main») e a menção pronta. Sem pasta escolhida, o botão de enviar fica desativado.

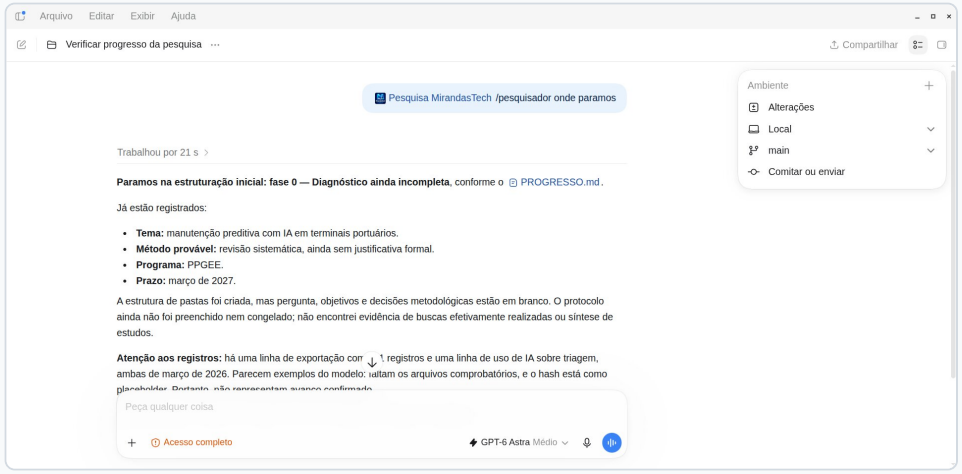



FIGURA 7 A resposta: o Codex leu o `PROGRESSO.md` da pasta e respondeu como o Pesquisador — fase 0 ainda incompleta, o que já está registrado, o que falta e a pergunta seguinte («Quem é o orientador deste projeto?»).

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

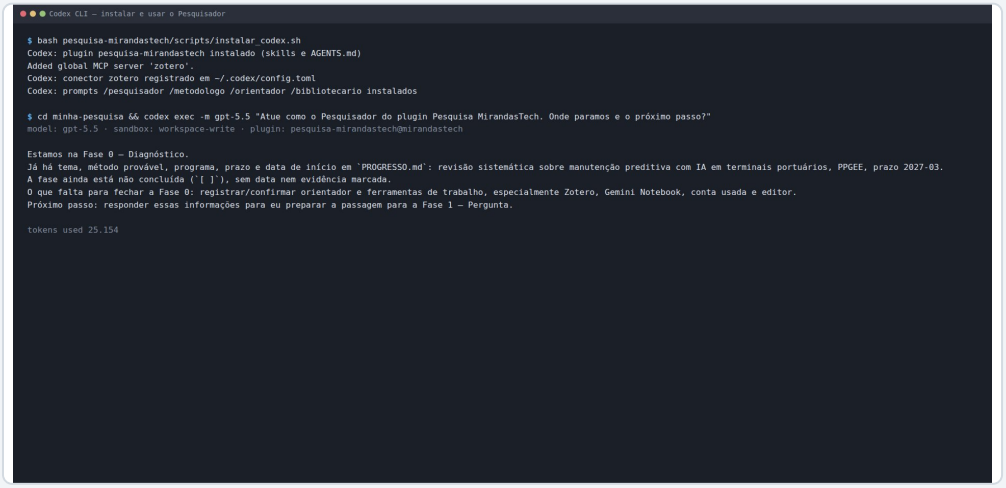



FIGURA 8 No terminal: o catálogo do repositório e o plugin `pesquisa-mirandastech`. O script `instalar_codex.sh` faz isso e ainda registra o conector e os prompts.

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/instalar_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 `codex mcp add zotero -- uv run --with mcp python scripts/zotero_mcp.py` | `~/.codex/config.toml` |
| Copia os prompts `/pesquisador`, `/metodologo`, `/orientador`, `/bibliotecario` | `~/.codex/prompts/` |
| Se o comando `codex` não existe, acrescenta a seção `[mcp_servers.zotero]` à mão | `~/.codex/config.toml` |

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 | ESCREVE? |
|---|---|---|
| `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 Better BibTeX | 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»

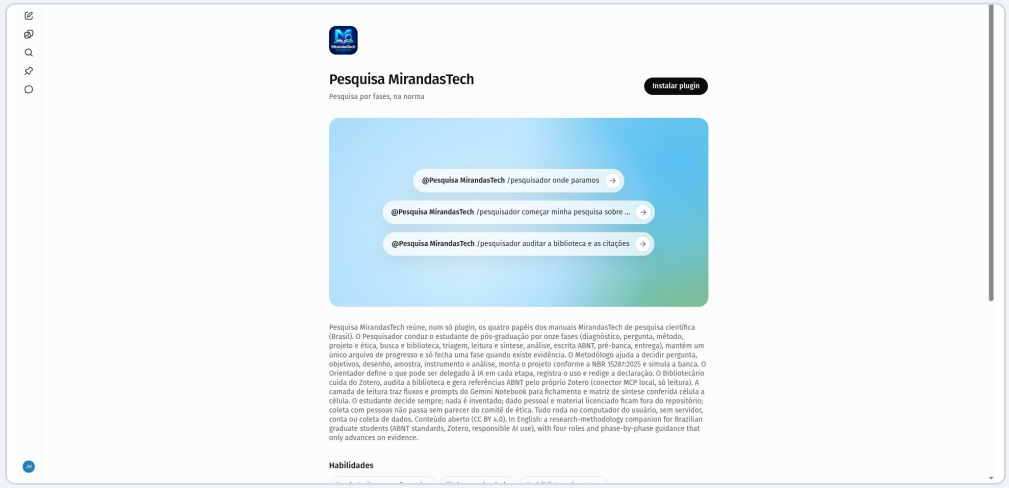



FIGURA 9 A página do plugin no diretório do ChatGPT: nome, subtítulo, categoria, as nove skills e o botão «Instalar plugin».

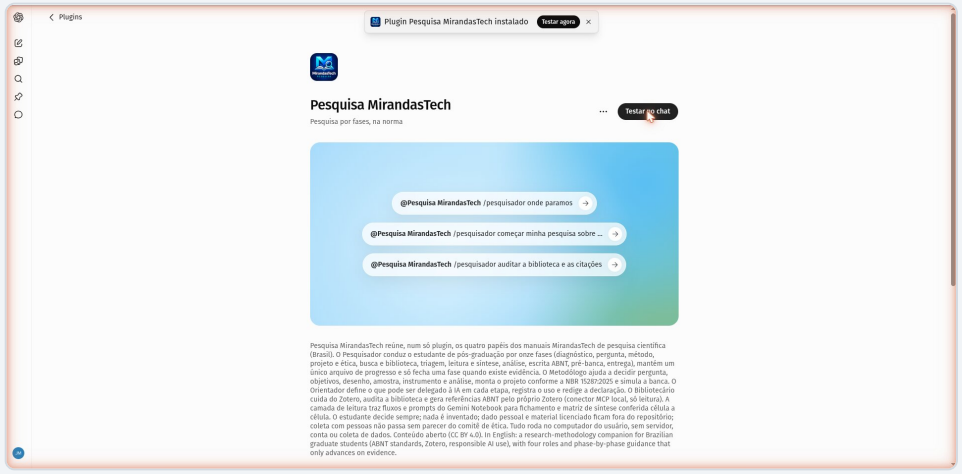



FIGURA 10 Depois do clique, o plugin aparece como instalado na sua conta.

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
```

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

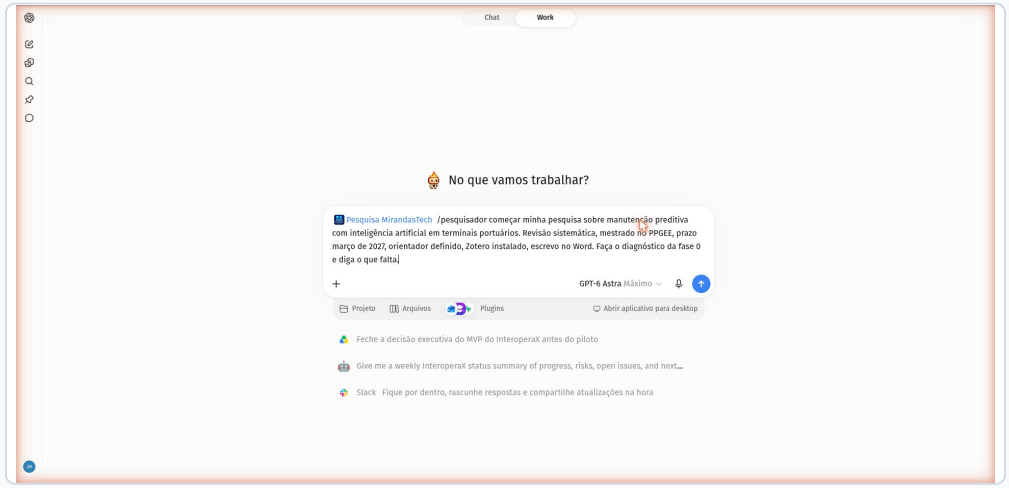



FIGURA 11 Digite «@», escolha «Pesquisa MirandasTech» e escreva o pedido na mesma mensagem.

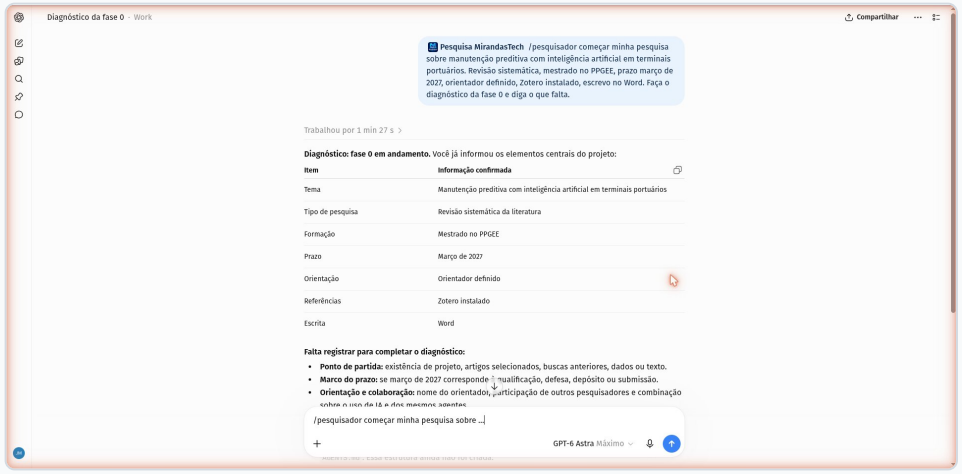



FIGURA 12 A resposta segue o roteiro do Pesquisador: uma pergunta do diagnóstico por vez, sem decidir por você.

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?
```

| 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.: `DECISOES_METODO.md`, um `.zip` com a estrutura de pastas) | A pasta do projeto lida e escrita no seu computador (`PROGRESSO.md`, `DECISOES.md`, `USO_DE_IA.csv`), sem baixar nada |
| Os quatro papéis, chamados pelo nome no pedido | O conector `zotero` 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.

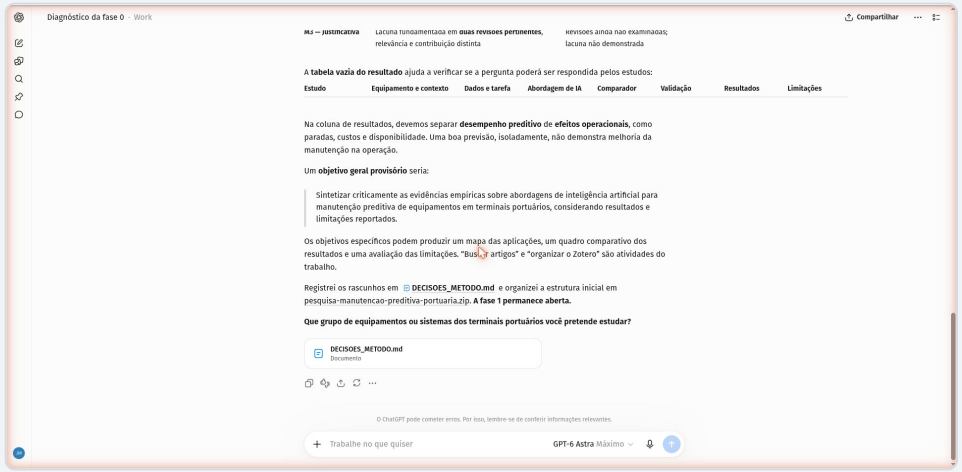



FIGURA 14 O fim da resposta: objetivo geral provisório, o arquivo `DECISOES_METODO.md` como anexo, «a fase 1 permanece aberta» e a pergunta que ele faz antes de seguir. Ele não fecha a fase sozinho.

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 |

PARTE 3 USAR

# 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.

03

USAR 1 DE 10

# 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            estado único: fases + checklists M,
I, B, L
├── DECISOES.md             decisões numeradas e datadas
├── USO_DE_IA.csv           uma linha por influência de IA
├── AGENTS.md               regras lidas por qualquer agente
├── ACORDO_ORIENTADOR.md    acordo sobre uso de IA
├── .gitignore              dados_FORA_DO_GIT/, *.pdf,
*.xlsx, .env
├── 00_projeto/             DECISOES_METODO.md · cronograma.csv
├── 00_protocolo/           protocolo.md (12 seções + emendas)
├── 01_buscas/              buscas.csv ·
registro_exportacoes.csv
├── 01_revisao/fichas/      fichas de leitura com página
├── 02_instrumentos/        tabela_operacionalizacao.csv
├── 02_triagem/             codebook, lotes, números do PRISMA
├── 03_etica/               TCLE.md · parecer do CEP
├── 03_extracao/  04_sintese/matriz_sintese.csv
├── 05_analise/livro_de_codigos.csv  06_texto/  07_banca/
├── tools/                  estrutura e etiquetas do Zotero;
prompts e
│                           estrutura dos notebooks
└── dados_FORA_DO_GIT/      dado pessoal e material licenciado
(fora do Git)
```

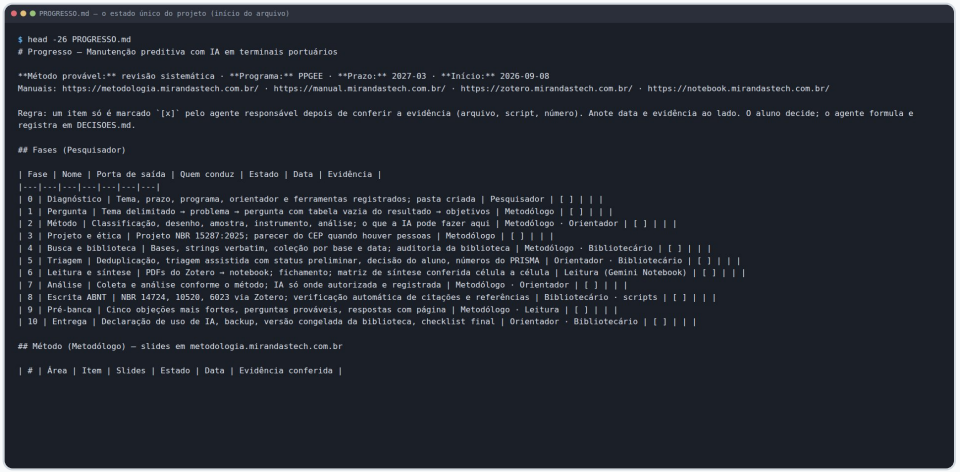



FIGURA 16 `PROGRESSO.md`: no alto, a tabela das onze fases (porta de saída, quem conduz, estado, data, evidência); abaixo, as quatro seções de checklist, cada item com os slides do manual de origem.

| SEÇÃO | ITENS | QUEM MARCA |
|---|---|---|
| **Fases** | 0 a 10 | Só o Pesquisador |
| **Método** (M1–M16) | Pergunta, objetivos, justificativa… defesa | Metodólogo |
| **Uso de IA** (I1–I16) | Governança, dados, busca, triagem… declaração, entrega | Orientador |

| SEÇÃO | ITENS | QUEM MARCA |
|---|---|---|
| **Biblioteca** (B1–B12) | Instalação, conta, estilo… backup, grupo | Bibliotecário |
| **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.

USAR 3 DE 10

# 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 `PROGRESSO.md` | Pesquisador | — |
| **1 Pergunta** | `00_projeto/DECISOES_METODO.md` 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 `06_texto/`; com pessoas: parecer do CEP com CAAE em `03_etica/` 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 `01_buscas/`; 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 |

| FASE | PORTA DE SAÍDA (EVIDÊNCIA) | QUEM CONDUZ | MANUAL DE ORIGEM |
|---|---|---|---|
| **6 Leitura e síntese** | PDFs dos incluídos no notebook; inventário rodado; fichas com página; `04_sintese/matriz_sintese.csv` com colunas do protocolo e cada célula conferida | Leitura · Bibliotecário | Gemini Notebook 11–23, 32–34 · IA 32–33 |
| **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; `verificar_citacoes.py` e `verificar_refs.py` 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** | Declaração de uso de IA 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»

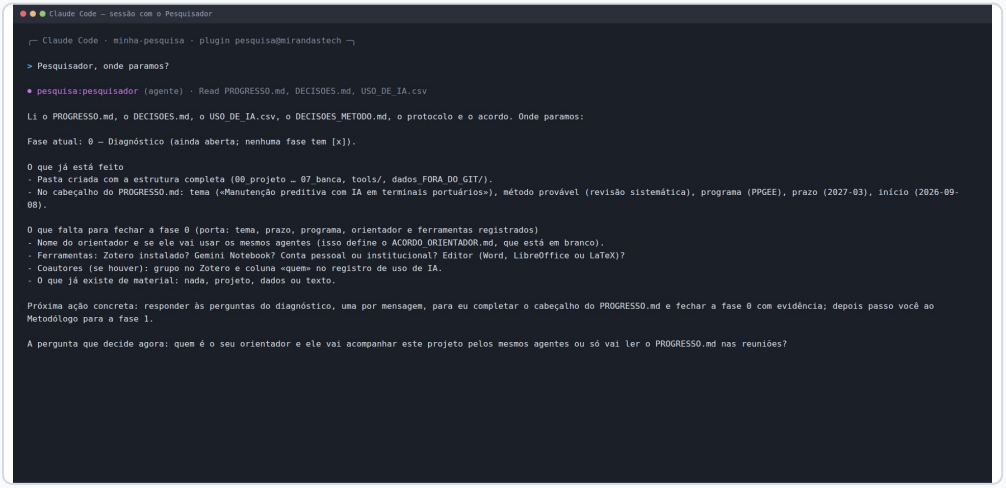



FIGURA 17 «Pesquisador, onde paramos?» na pasta do projeto: o agente lê `PROGRESSO.md`, `DECISOES.md` e `USO_DE_IA.csv` e responde com a fase atual, o que está feito, o que falta para fechar a fase e a pergunta que decide agora — sem marcar nada como concluído.

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.

USAR 5 DE 10

# 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 | `Bibliotecário, audite a coleção «2 Incluídos»` | `/bibliotecario …` | Seção Biblioteca (B); correção no item do Zotero |

| PAPEL | CHAME QUANDO… | EXEMPLO DE PEDIDO (CLAUDE CODE) | NO CODEX | O QUE ELE REGISTRA |
|---|---|---|---|---|
| | errada, exportar, backup | `Bibliotecário, por que esta referência saiu com o nome em caixa alta?` | | |
| 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`<br>`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 `iniciar_projeto.py`; 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 `PROGRESSO.md`, `DECISOES.md`, `USO_DE_IA.csv` |
| `/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 `PROGRESSO.md` |

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

| COMANDO | O QUE FAZ | O QUE LÊ OU GERA |
|---|---|---|
| `/pesquisa:auditar` | Roda as verificações automáticas e resume as pendências por gravidade, sem corrigir nada | `auditar_biblioteca.py`, `verificar_citacoes.py` em `06_texto/`, `verificar_refs.py`; conector `zotero` |
| `/pesquisa:declarar` | Chama o Orientador e monta a declaração de uso de IA; aponta linhas faltantes antes de redigir | Lê `USO_DE_IA.csv` |
| `/pesquisa:banca` | Cinco objeções com resposta (Metodólogo) e perguntas prováveis a partir da tese no notebook (Leitura) | Consolida em `07_banca/` com a página da tese |
| `/pesquisa:entregar` | Percorre a fase 10 e as portas anteriores; lista o que bloqueia, com evidência esperada e quem conduz | Lê `PROGRESSO.md` |

> depois o Pesquisador comenta em dez linhas. Gere-o antes de cada reunião; nada é marcado.

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

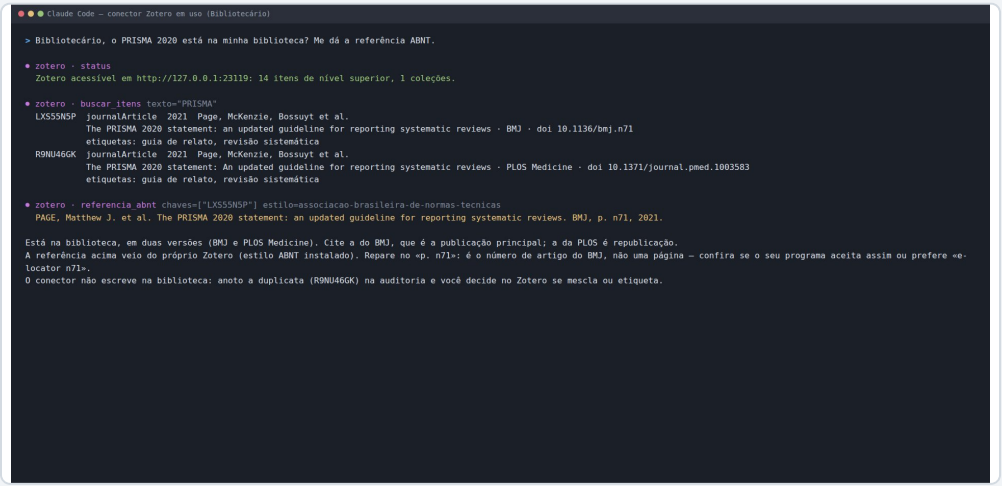



FIGURA 18 Três chamadas ao conector: `status` confirma o Zotero acessível; `buscar_itens` acha o PRISMA 2020 em duas versões; `referencia_abnt` devolve a referência gerada pelo próprio Zotero com o estilo ABNT instalado — não de memória. O Bibliotecário aponta a duplicata e pergunta antes de etiquetar.

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 IMRaD → 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 → respostas com página em `07_banca/` |
| 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 |

REGRA

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.

GATES HUMANOS

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.

USAR 10 DE 10

# 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 ESCREVE 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 — «você 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.

PARTE 4 DEMONSTRAÇÃO

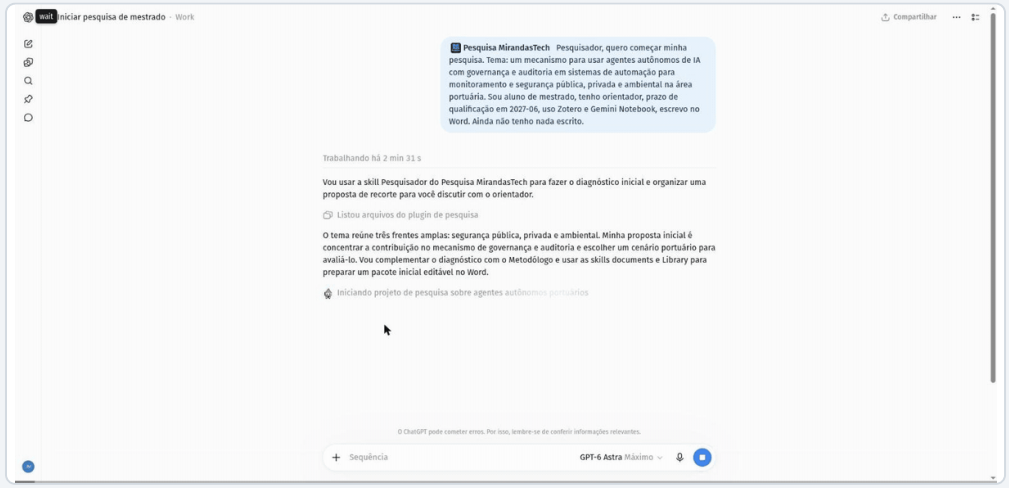
# Uma pesquisa começando do zero, no ChatGPT

Sessão real de 9 de setembro de 2026, no ChatGPT do navegador com o plugin na versão 1.0.3: tema de automação portuária, do diagnóstico à pergunta de pesquisa. As animações mostram a tela como ela é; o que o plugin gerou está descrito ao lado de cada uma.

04

DEMONSTRAÇÃO 1 DE 4 · FASE 0

# Passo 1 — chamar o Pesquisador com o contexto: tema, nível, prazo e ferramentas




ANIMAÇÃO 1 Digite «@», escolha «Pesquisa MirandasTech» e escreva o pedido na mesma mensagem. O Pesquisador detecta a fase 0, propõe um recorte, lista o que falta e entrega o pacote inicial como anexos.

MENSAGEM ENVIADA

```
@Pesquisa MirandasTech Pesquisador, quero começar minha
pesquisa. Tema: um mecanismo para usar agentes autônomos de
IA com governança e auditoria em sistemas de automação para
monitoramento e segurança pública, privada e ambiental na
área portuária. Sou aluno de mestrado, tenho orientador,
prazo de qualificação em 2027-06, uso Zotero e Gemini
Notebook, escrevo no Word. Ainda não tenho nada escrito.
```

| O QUE ELE FEZ | COMO APARECEU |
|---|---|
| Fase | «iniciei sua pesquisa na fase 0 — diagnóstico»; nada marcado como feito |
| Recorte | Concentrar a contribuição no mecanismo de governança e auditoria; as três frentes (pública, privada, ambiental) como contexto; título provisório |
| Desenho candidato | Protótipo comparado com e sem o mecanismo em ambiente controlado: violações de política, reconstrução das execuções, latência, conclusão das tarefas |
| O que falta | |

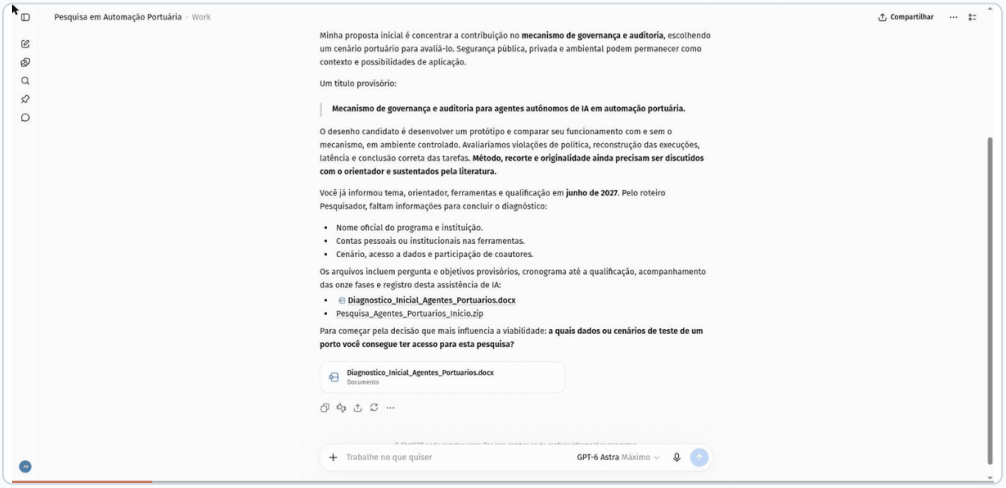
| O QUE ELE FEZ | COMO APARECEU |
| --- | --- |
| | Programa e instituição; contas pessoais ou institucionais; cenário, acesso a dados, coautores |
| Anexos | Um DOCX de diagnóstico e um ZIP com a pasta do projeto ( `PROGRESSO.md` , `DECISOES.md` , `USO_DE_IA.csv` , `00_projeto/` ) |
| Fecho | Uma única pergunta: a quais dados ou cenários de teste de um porto você tem acesso |

ARMADILHA

**Tomar o DOCX como projeto pronto.** A pergunta e os objetivos que saem na fase 0 são provisórios, e o próprio texto diz isso; eles só viram decisão na fase 1, com o Metodólogo. A resposta levou onze minutos porque o ChatGPT montou o DOCX e o ZIP; no Claude Code ou no Codex, a pasta nasce em segundos.

DEMONSTRAÇÃO 2 DE 4 · FASE 0 FECHADA

# Passo 2 — responder às pendências e pedir para fechar a fase 0




ANIMAÇÃO 2 Com as pendências respondidas, o Pesquisador fecha a fase 0 e devolve o `PROGRESSO.md` inteiro para substituir o seu.

MENSAGEM ENVIADA

```
Tenho acesso a um simulador de terminal de contêineres (gêmeo
digital com dados sintéticos) do laboratório do meu grupo;
não tenho dados reais de porto. Programa: mestrado em
engenharia elétrica de uma universidade pública. Contas:
Gemini Notebook institucional, Zotero pessoal. Sem coautores;
o orientador vai ler o PROGRESSO.md nas reuniões. Pode fechar
a fase 0?
```

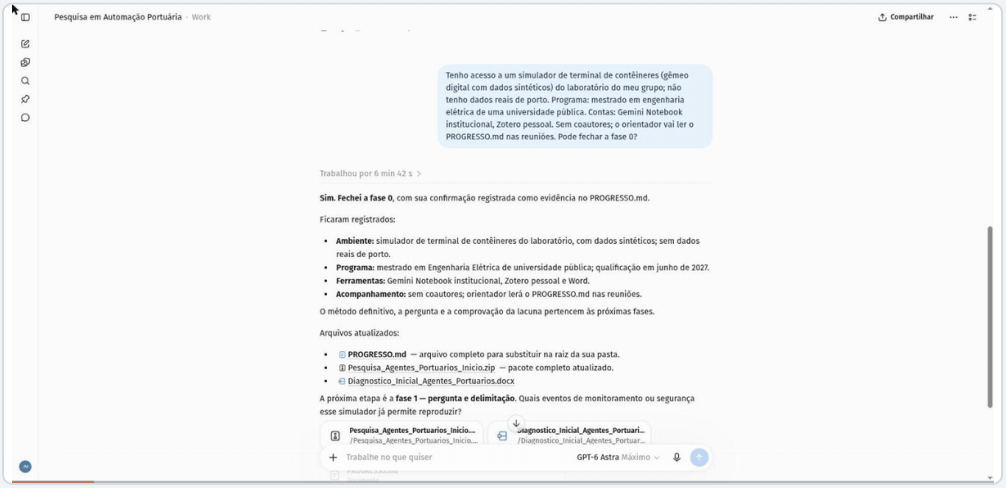
- «Fechei a fase 0, com sua confirmação registrada como evidência no PROGRESSO.md»: a porta da fase é tema, prazo, programa, orientador e ferramentas registrados.
- Registrou ambiente (simulador com dados sintéticos), programa, ferramentas e acompanhamento; avisou que a validade dos resultados para operações reais fica para o desenho do estudo.
- Devolveu três anexos: `PROGRESSO.md` completo «para substituir na raiz da sua pasta», o ZIP atualizado e o DOCX.
- Terminou pela pergunta da fase seguinte: quais eventos de monitoramento ou segurança o simulador reproduz.

REGRA

> **Sem pasta, o estado vive no arquivo devolvido.** No ChatGPT o plugin não lê o seu computador: guarde o `PROGRESSO.md` que ele devolveu (nesta conta, ele também ficou na biblioteca do ChatGPT, e a retomada do passo 4 usou isso). Se a conversa seguinte não o encontrar, anexe ou cole o arquivo; sem ele, o Pesquisador não deve inventar onde você parou.

DEMONSTRAÇÃO 3 DE 4 · FASE 1

# Passo 3 — chamar o Metodólogo: a pergunta «boa» que ainda não está pronta




ANIMAÇÃO 3 Na mesma conversa, o papel é chamado pelo nome. O Metodólogo critica a pergunta, propõe uma versão verificável, monta a tabela vazia do resultado e lista o que fecha M1–M3.

MENSAGEM ENVIADA

```
O simulador reproduz intrusão em área restrita do pátio,
colisão iminente entre veículos e guindastes, e vazamento em
tanque de combustível (alarme ambiental). Metodólogo, minha
pergunta de pesquisa é: «como agentes autônomos melhoram a
segurança do porto?». Está boa? O que falta para fechar a
fase 1?
```

| O QUE ELE APONTOU | POR QUÊ |
|---|---|
| «Melhoram» pressupõe benefício | A pesquisa tem de admitir melhora, ausência de efeito ou piora |
| «Segurança» é ampla | Conformidade das ações, auditoria e resposta ao incidente pedem medidas distintas |
| «Porto» não diz o ambiente | É um terminal de contêineres simulado, com dados sintéticos |
| A contribuição sumiu | O mecanismo de governança e auditoria não estava na pergunta |
| Pergunta candidata | «Em um gêmeo digital de terminal de contêineres, qual é o efeito de um mecanismo de governança e auditoria sobre as ações não |

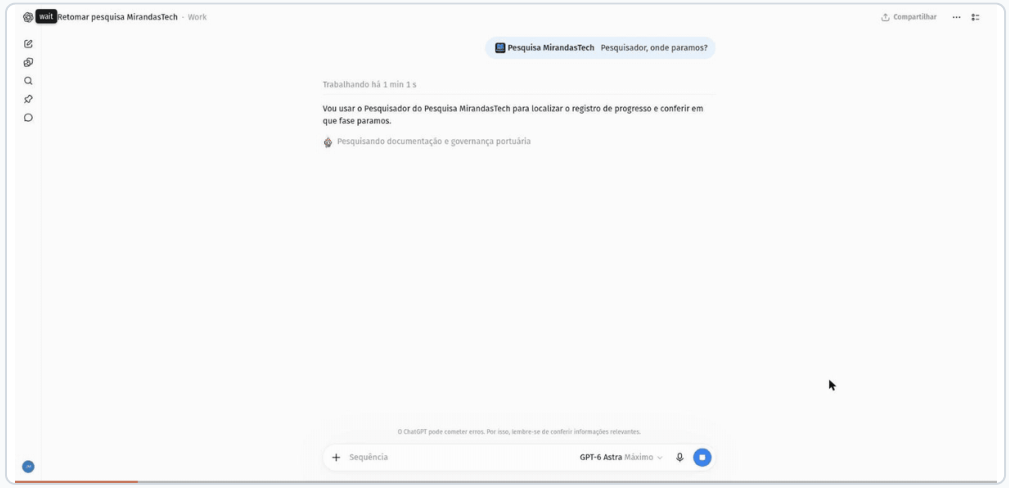
| O QUE ELE APONTOU | POR QUÊ |
| --- | --- |
| | autorizadas e a reconstrução das execuções dos agentes…» |
| Tabela vazia | Condição (sem e com o mecanismo) × episódios, ações não autorizadas, execuções reconstruídas, respostas corretas no prazo |

REGRA

**O aluno decide o recorte.** Ele sugeriu a intrusão em área restrita como cenário principal e deixou claro: «essa escolha ainda é sua». M1–M3 seguem pendentes (0/16 itens de método) até a pergunta, os objetivos e a lacuna com duas revisões estarem registrados. «Bloquear tudo e registrar tudo não basta para demonstrar melhora da segurança.»

DEMONSTRAÇÃO 4 DE 4 · RETOMADA

# Passo 4 — numa conversa nova: «Pesquisador, onde paramos?»




ANIMAÇÃO 4 Conversa nova, sem anexar nada. O Pesquisador localizou o `PROGRESSO.md` na biblioteca da conta e respondeu com fase atual, o que está registrado, a última pergunta em aberto e o que falta — cada afirmação com a fonte marcada.

MENSAGEM ENVIADA

```
@Pesquisa MirandasTech Pesquisador, onde paramos?
```

- «Paramos na fase 1 — pergunta e delimitação»; «a fase 0 foi concluída em 09/09/2026». Nada foi inventado: cada frase traz o marcador PROGRESSO da fonte lida.
- Repetiu o que está registrado (qualificação, orientador, ferramentas, simulador com três eventos) e a última pergunta em aberto, «ainda não aprovada por você».
- Listou M1–M3 como pendentes e fechou com a próxima decisão: escolher o cenário principal do simulador.
- No ChatGPT, os arquivos ficaram na biblioteca da conta e por isso a retomada funcionou sem anexo. Se a sua conta não guarda arquivos entre conversas, anexe ou cole o `PROGRESSO.md` na primeira mensagem; sem ele, a resposta correta é «não há estado a retomar».

O MESMO NO CLAUDE CODE E NO CODEX

> Lá não há biblioteca nem anexo: abra a sessão dentro da pasta do projeto e diga a mesma frase. O hook mostra o resumo do estado antes mesmo da pergunta (Parte 3).

PARTE 5 MANUTENÇÃO

# 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.

05

MANUTENÇÃO 1 DE 3 · VERSÃO ATUAL 1.0.3

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

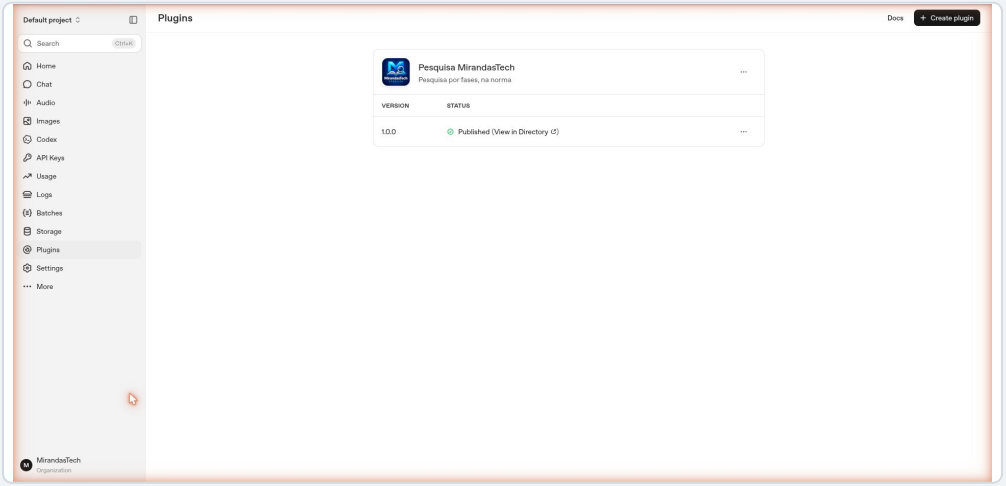



FIGURA 19 O portal de plugins com o Pesquisa MirandasTech publicado (08/09/2026). Uma nova versão é um novo ZIP enviado aqui, com o número de versão maior.

| CLIENTE | COMO ATUALIZA |
|---|---|
| **Claude Code** | O marketplace aponta para o repositório: depois de um push no GitHub, o plugin se atualiza pelo catálogo (o script usa `claude plugin marketplace update mirandastech` quando o catálogo já existe) |
| **Codex (marketplace próprio)** | Mesmo repositório; o script usa `codex plugin marketplace upgrade mirandastech` |
| **Diretório do ChatGPT/Codex** | Gerar o ZIP com a versão nova e reenviar no portal |

TERMINAL — NOVA VERSÃO PARA O DIRETÓRIO DO CODEX

```
python3 scripts/empacotar_codex.py --versao 1.1.0
ls dist/
# → dist/pesquisa-mirandastech-1.1.0.zip
```

- 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 `zotero` aparece desconectado | Zotero fechado, API local desligada ou `uv` ausente | Abra o Zotero e ligue a API loca o `uv` ou troque o comando do conector por `python3` com `p install mcp` |
| No Codex, cada chamada ao conector pede aprovação | Política padrão do Codex para ferramentas MCP | Em `~/.codex/config.toml` , p ferramenta: `[mcp_servers.zotero.tools.a` e `approval_mode = "auto"` (s |
| `verificar_refs.py` não resolve os DOIs | O script consulta a Crossref pela internet | Rode com rede; sem rede ele s o formato |
| `PROGRESSO_IA.md` e `PROGRESSO_METODOLOGIA.md` antigos na pasta | Projeto criado pelos manuais anteriores | `iniciar_projeto.py` migra os marcados para `PROGRESSO.md` mantém os antigos como histó |
| «Pesquisador, onde paramos?» não funciona no Claude Desktop | O Desktop recebe só o conector | Use o Claude Code ou o Codex 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` . Laboratório: lab.mirandastech.com.br.

| SINTOMA | CAUSA | SOLUÇÃO |
| --- | --- | --- |
| `/plugin install pesquisa@claude-community` falha | Submissão ainda em revisão | Use `pesquisa@mirandastech`; 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 à MirandasTech |
| **Conector** `zotero` | Lê a biblioteca pela API local ( `127.0.0.1:23119` ), só quando você o usa; não escreve nem envia nada para fora |
| `verificar_refs.py` | 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; o `.gitignore` gerado mantém `dados_FORA_DO_GIT/` , PDFs e planilhas fora do Git |

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

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

> seu projeto. O plugin foi desenhado para não precisar de dado pessoal de 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.

PARTE 6 FECHAMENTO

# 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.

06

FECHAMENTO 1 DE 6

# Erros comuns — como aparecem e como corrigir

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

| 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 |

FECHAMENTO 2 DE 6

# 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 `zotero` em `claude_desktop_config.json`; aplicativo reiniciado; «liste as coleções» responde

- [ ] 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

- [ ] CONECTOR EM USO
  «Bibliotecário, audite a coleção …» chamou `auditar`; uma referência ABNT gerada por `referencia_abnt`; correção feita no item do Zotero

- [ ] CODEX (APLICATIVO)
  Plugin instalado pelo diretório («Plugins» → «Explorar diretório»); @Pesquisa MirandasTech responde no chat com a pasta do projeto aberta

- [ ] PROJETO
  Pasta criada por `iniciar_projeto.py` ou `/pesquisa:iniciar`; `PROGRESSO.md`, `DECISOES.md`, `USO_DE_IA.csv`, `AGENTS.md` presentes; Git iniciado

- [ ] REGISTRO
  Primeira decisão em `DECISOES.md` (numerada, datada, com a sua justificativa); primeira linha em `USO_DE_IA.csv`

- [ ] REQUISITOS
  Python 3, Git e `uv` instalados; Zotero 7 ou superior aberto com a API local ligada em «Configurações» → «Avançado»

- [ ] CODEX (LINHA DE COMANDO)
  `instalar_codex.sh` rodado; `codex plugin list` mostra o plugin; conector em `~/.codex/config.toml`; prompts em `~/.codex/prompts`

- [ ] DIAGNÓSTICO
  As cinco perguntas da fase 0 respondidas;

- [ ] CLAUDE CODE
  Catálogo `miran` adicionado; `pesquisa@mira` instalado; `claude plugin` mostra 4 agente skills, 7 comand conector e hook

- [ ] APROVAÇÕES
  No Codex, ferramentas do conector liberadas com `approval_mode = "auto"` só depois de entender que todas apenas leem

- [ ] 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:statu` gerado antes da primeira reuniã

0 de 14

limpar

FECHAMENTO 3 DE 6

# 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

## 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

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)

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.

FECHAMENTO 4 DE 6 · ENDEREÇOS CONFERIDOS EM 08/09/2026

# 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. lab.mirandastech.com.br
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

FECHAMENTO 5 DE 6

# 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: lab.mirandastech.com.br.

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, só leitura) |
| `hooks/hooks.json` | Hook de início de sessão |
| `scripts/` | `iniciar_projeto.py`, `verificar_citacoes.py`, `verificar_refs.py`, `tamanho_amostra.py`, `cronograma.py`, |

| PASTA / ARQUIVO | O QUE É |
| --- | --- |
| | `auditar_biblioteca.py`, `exportar_bbt.sh`, `backup_zotero.sh`, `instalar_claude.sh`, `instalar_codex.sh`, `empacotar_codex.py` |
| `modelos/` | Arquivos copiados para o projeto; estilos ABNT (.csl); prompts do Gemini Notebook |
| `.codex-plugin/` · `.agents/plugins/` · `AGENTS.md` · `codex/prompts/` | Pacote do Codex: manifesto, catálogo, instruções e os quatro prompts |
| `docs/` | `PLANO.md`, `DESCRICAO.md` (textos de submissão), `PRIVACIDADE.md`, `TERMOS.md`, `SUPORTE.md` |
| `exemplos/` | Dois projetos fictícios preenchidos — revisão sistemática na fase 6 e estudo de caso na fase 8 — para ver o `PROGRESSO.md` com evidência, decisões datadas e uso de IA registrado |

| PASTA / ARQUIVO | O QUE É |
| --- | --- |
| evals/ | Testes dos agentes ( `claude plugin eval` ): nove casos com projeto preparado por cenário e juiz LLM; rodar antes de publicar uma versão |
| dist/ | ZIP da versão a enviar ao diretório do Codex ( `pesquisa-mirandastech-1.0.3.zip` ) |
| .github/workflows/verificar.yml | Verificação contínua: manifestos, cabeçalhos de agentes, skills e comandos, scripts |

FECHAMENTO · ÚLTIMO SLIDE MÉTODO + USO DE IA + REFERÊNCIAS + LEITURA, DENTRO DA MESMA FERRAMENTA

# 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 PL |
|---|---|---|---|---|---|
| **4 Busca** | Bases, string, critérios | Sugestão de termos, com registro | Coleção por base e data | Nada | Prot strin audi |
| **5 Triagem** | Critérios; dois revisores | Triagem assistida, status preliminar | Etiquetas por revisor | Tabela de inventário dos PDFs | Deci regis PRIS |
| **6 Leitura** | Fichamento; matriz | Resumo de apoio, nunca substituto | Notas com página | Perguntas por seção; matriz para a planilha | Cada fech |
| **8 Escrita** | Normas e estrutura | Revisão de forma; declaração | Citações vivas; lista gerada | Nada no texto | `ver` e `ve` sem |

| FASE | METODOLOGIA DECIDE | IA DELEGA OU NÃO | ZOTERO REGISTRA | GEMINI NOTEBOOK FAZ | O PL |
|---|---|---|---|---|---|
| **9 Defesa** | Roteiro da banca | Simulação com registro | Biblioteca congelada | Perguntas prováveis a partir da tese | Obje com `07_` |