Claude + Codex

Manual do cliente / Instalação, configuração e primeiros passos

Seu acesso. Seu próximo projeto.

PLANO DE ACESSO

R$160 por 30 dias.
Pagamento avulso. Ao final do período, você escolhe se deseja renovar.

Use o acesso por API em ferramentas compatíveis para desenvolver sites, revisar projetos, escrever código e resolver tarefas de programação.

Bem-vindo

Este manual foi preparado para acompanhar você da configuração à primeira tarefa. Escolha o sistema e a ferramenta que pretende usar; não é necessário instalar todas as opções.

Comece por aqui	Página
Seu plano e o que você vai receber	2
Escolha da ferramenta e dados da conexão	3
Windows: preparação e Claude Code no terminal	4-6
Windows: extensão Claude no VS Code	7-8
Windows: Codex com perfil separado	9-10
macOS e Linux	11-12
Aplicativos de janela	13
Outros editores e agentes	14-17
Primeira tarefa, erros e suporte	18-19
Perguntas frequentes e referências técnicas	20

Serviço independente de acesso por API. Claude e Codex são marcas de seus respectivos titulares. A conexão utiliza infraestrutura de terceiros. Edição de 09/11/2026.

Seu plano, sem complicação

R$160 / validade de 30 dias / renovação manual

Item	Como funciona
Valor	R$160 por período de 30 dias.
Renovação	Você decide se deseja contratar um novo período.
Acesso	Chave API para as rotas habilitadas de Claude e Codex.
Ferramentas	Uso nas ferramentas compatíveis com a conexão contratada.
Conta pessoal	A chave API é uma forma de acesso diferente do login pessoal.

Antes de começar, tenha em mãos

Sua chave de acesso, a confirmação de ativação e uma pasta de projeto. Confira com o atendimento a data de vencimento e os modelos liberados na sua chave. O PDF não contém sua chave pessoal.

O que você pode fazer

Pedir explicações de código, planejar uma implementação, criar páginas e componentes, investigar erros, revisar alterações e preparar testes. A capacidade de ler arquivos ou executar comandos depende da ferramenta instalada e das permissões concedidas por você.

Como o acesso é utilizado

O aplicativo envia sua solicitação ao endereço configurado. A chave identifica o acesso e o modelo selecionado determina a rota usada. Nas ferramentas de programação, é você quem escolhe a pasta em que o agente pode trabalhar.

INFORMAÇÕES DO SEU ACESSO

Guarde a chave como uma senha. Para saber sobre dispositivos, volume de uso, disponibilidade, prazo de entrega ou condições de suporte, considere o que foi combinado no atendimento. Este manual não acrescenta benefícios ou garantias não contratados.

Ao final dos 30 dias

Entre em contato pelo mesmo canal da compra se quiser renovar. Consulte a validade informada na ativação; não presuma que a instalação do programa renove o acesso automaticamente.

Escolha seu caminho

Comece por uma única ferramenta e valide a conexão antes de configurar outra.

Você quer...	Caminho indicado
Usar Claude no terminal	Claude Code: páginas 4-6 no Windows; 11 no Mac/Linux.
Conversar pelo painel do VS Code	Extensão oficial Claude Code com perfil API: páginas 7-8.
Usar Codex no terminal	Codex CLI com perfil API: páginas 9-10 e 12.
Usar aplicativo de janela	Veja os cuidados e os passos da página 13.
Usar outro editor	Escolha o guia das páginas 14-17.

Dados de conexão utilizados neste manual

Integração	Endereço de referência
Formato Anthropic	https://api.tn1.top
Formato OpenAI / Codex CLI	https://api.tn1.top/v1

A conexão depende também do protocolo, do modelo e da validade da chave. Se o atendimento fornecer um endereço atualizado para uma ferramenta, use essa orientação específica. Não acrescente /v1 a todos os programas automaticamente.

Nomes do catálogo

Os identificadores publicados para o gateway incluem Opus 5.5, Sonnet 5.5, Haiku, Fable-5 e gpt-6-luna. Use apenas os modelos liberados na sua chave e preserve exatamente letras, espaços e maiúsculas. O catálogo pode mudar.

CONTA PESSOAL E API EM SEPARADO

Os passos principais criam configurações próprias para o acesso API. Mantenha seu perfil habitual para sua conta pessoal. Um histórico de conversa antigo pode ter sido criado com outra conexão; para validar, abra uma conversa nova.

A configuração manual é o caminho principal deste material. Ela torna visível o que está sendo alterado e evita depender de um instalador automático para começar.

Windows: preparar o computador

Esta etapa é comum a quem pretende usar as ferramentas pelo terminal.

1. Instale Node.js

Acesse o site oficial do Node.js, escolha a versão LTS para Windows e siga o instalador. Mantenha a opção de adicionar ao PATH quando apresentada. Se já estiver instalado, confira a versão antes de reinstalar.

2. Instale Git

Acesse Git para Windows. A instalação disponibiliza Git e Git Bash. Isso é útil para projetos e ferramentas que usam comandos Bash. Não é necessário ativar WSL para seguir este caminho nativo do Windows.

3. Abra um terminal novo

Feche o terminal usado durante a instalação. No menu Iniciar, procure PowerShell e abra uma nova janela normal. Nos exemplos abaixo, não copie o texto PS C:\... que aparece antes do cursor; copie apenas o comando.

```
node --version
npm.cmd --version
git --version
```

Execute uma linha por vez. O resultado esperado é um número de versão em cada comando. Se aparecer que o comando não foi encontrado, reabra o terminal e confira se a instalação terminou.

4. Instale as ferramentas que vai usar

Para Claude no terminal, instale o pacote correspondente. Para Codex, use o segundo pacote. Você pode instalar ambos, mas isso ainda não configura a sua chave.

```
npm.cmd install -g @anthropic-ai/claude-code
npm.cmd install -g @openai/codex
```

```
claude --version
codex --version
```

SE O POWERSHELL BLOQUEAR npm.ps1

Os exemplos usam npm.cmd para chamar o executável do npm no Windows. Isso evita precisar mudar a política de execução apenas para rodar esses comandos. Não desative proteções do sistema para continuar.

Windows: perfil do Claude

Passo a passo para criar uma configuração separada do seu login habitual.

1. Crie a pasta do acesso

No PowerShell, execute as duas primeiras linhas. O Bloco de Notas pode perguntar se deseja criar o arquivo; confirme. A pasta fica no seu usuário, fora do projeto.

```
$perfilClaudeApi = Join-Path $env:USERPROFILE ".claude-acesso-api"
New-Item -ItemType Directory -Force -Path $perfilClaudeApi
notepad (Join-Path $perfilClaudeApi "settings.json")
```

2. Cole a configuração

No arquivo novo, cole o bloco abaixo. Substitua somente COLE_SUA_CHAVE_AQUI pela chave recebida, mantendo as aspas. O exemplo começa com Sonnet 5.5; ajuste se sua ativação indicar outro modelo.

```
{
  "env": {
    "ANTHROPIC_BASE_URL": "https://api.tn1.top",
    "ANTHROPIC_AUTH_TOKEN": "COLE_SUA_CHAVE_AQUI",
    "ANTHROPIC_MODEL": "Sonnet 5.5",
    "ANTHROPIC_SMALL_FAST_MODEL": "Haiku",
    "ANTHROPIC_DEFAULT_OPUS_MODEL": "Opus 5.5",
    "ANTHROPIC_DEFAULT_SONNET_MODEL": "Sonnet 5.5",
    "ANTHROPIC_DEFAULT_HAIKU_MODEL": "Haiku"
  }
}
```

3. Salve com o nome correto

Pressione Ctrl+S e feche o Bloco de Notas. O arquivo deve se chamar settings.json, sem .txt no final. Como ele foi aberto pelo caminho no comando, o nome já estará definido.

ONDE A CHAVE FICA

Neste método, a chave fica no arquivo do perfil, em texto legível. Não envie esse arquivo, não publique no Git e não use uma pasta sincronizada ou compartilhada. Ele é separado do settings.json da sua conta habitual.

Se o arquivo já existir, faça uma cópia antes de editar e preserve as demais opções. Não cole o conteúdo do seu arquivo pessoal por cima deste perfil.

Windows: abrir e testar o Claude

Use esta sequência no terminal em que deseja trabalhar com a API.

1. Selecione o perfil desta sessão

No PowerShell, defina a pasta alternativa. Esse comando vale para o processo atual e para os programas iniciados por ele; não grava uma variável permanente no Windows.

```
$env:CLAUDE_CONFIG_DIR = Join-Path `
  $env:USERPROFILE ".claude-acesso-api"
```

2. Entre na pasta do projeto

Troque o caminho do exemplo pela sua pasta. Mantenha aspas se o nome tiver espaços. Você pode usar uma pasta de teste vazia antes de abrir um projeto importante.

```
Set-Location "C:\caminho\do\seu-projeto"
claude
```

3. Confirme a conexão

Se aparecer uma pergunta sobre confiança na pasta, confirme apenas se for um diretório seu. Dentro do Claude, consulte o status e confira a configuração carregada e o modelo. Use o modelo habilitado no seu acesso.

```
/status
```

4. Faça um teste pequeno

Abra uma conversa nova e envie a frase abaixo. O resultado esperado é uma resposta curta, sem alteração de arquivos. A resposta confirma que uma chamada funcionou, não a identidade independente do modelo.

MENSAGEM DE TESTE

Responda apenas OK, sem usar ferramentas e sem alterar arquivos.

5. Volte à sua conta pessoal

Encerre a sessão e feche esta janela do PowerShell. Abra outra janela normal e use seu caminho habitual. Como não usamos setx, a seleção de perfil acima não foi tornada permanente.

SE APARECER CONFLITO DE AUTENTICAÇÃO

Não faça logout nem apague credenciais da sua conta pessoal como primeira medida. Confira o perfil utilizado e peça ajuda para identificar variáveis antigas. Compartilhe somente a mensagem do erro, com a chave ocultada.

VS Code: perfil próprio para API

Preparação do painel gráfico do Claude sem misturar preferências da conta pessoal.

1. Abra o VS Code

Se necessário, instale pelo site oficial do VS Code. Não precisa abrir o programa como administrador.

2. Crie um perfil vazio

Na engrenagem de gerenciamento, abra Profiles / Perfis e procure New Profile / Novo Perfil. Dê o nome Acesso API. Prefira um perfil vazio, com configurações próprias, em vez de reutilizar todas as configurações do perfil padrão.

3. Instale a extensão nesse perfil

Pressione Ctrl+Shift+X, procure Claude Code e confira o publicador Anthropic. Instale a extensão oficial no perfil Acesso API. Se já estiver em outro perfil, confirme que também está disponível neste.

4. Abra a pasta de trabalho

Use Arquivo > Abrir Pasta. Escolha um projeto seu. Confie na pasta somente se conhecer seu conteúdo. O perfil ativo pode ficar associado ao projeto; observe o nome do perfil ao voltar a ele.

5. Abra as configurações do usuário do perfil

Pressione Ctrl+Shift+P, digite Preferences: Open User Settings (JSON) e abra essa opção. Use as configurações do usuário do perfil Acesso API, não Workspace Settings / configurações do projeto.

ARQUIVO VAZIO

Se o arquivo estiver vazio, você pode começar com o JSON completo da próxima página. Se ele já tiver opções, acrescente as novas propriedades dentro do objeto existente, separadas por vírgula. Não crie dois objetos JSON independentes.

O JSON desta etapa é do VS Code. Ele é diferente do settings.json do Claude criado na página 5. A chave ficará nas configurações locais deste perfil para que a extensão receba os dados ao iniciar.

O painel da extensão inclui sua própria cópia da ferramenta. Instalar a extensão não disponibiliza automaticamente o comando claude em todos os terminais; para isso, siga também a instalação da página 4.

VS Code: conectar o painel Claude

Cole os valores apenas no perfil Acesso API criado na página anterior.

1. Descubra seu caminho de usuário

No PowerShell, execute $env:USERPROFILE. Use o resultado para montar o caminho completo da pasta .claude-acesso-api. No JSON, as barras invertidas do caminho precisam ser duplicadas.

2. Preencha o JSON

Troque SEU_USUARIO pelo nome da sua pasta de usuário e COLE_SUA_CHAVE_AQUI pela chave. O caminho precisa ser absoluto; a extensão não expande ~ neste campo.

```
{
  "claudeCode.disableLoginPrompt": true,
  "claudeCode.environmentVariables": [
    {
      "name": "CLAUDE_CONFIG_DIR",
      "value": "C:\\Users\\SEU_USUARIO\\.claude-acesso-api"
    },
    {
      "name": "ANTHROPIC_BASE_URL",
      "value": "https://api.tn1.top"
    },
    {
      "name": "ANTHROPIC_AUTH_TOKEN",
      "value": "COLE_SUA_CHAVE_AQUI"
    }
  ]
}
```

3. Salve e recarregue

Pressione Ctrl+S. Na paleta Ctrl+Shift+P, execute Developer: Reload Window. Abra o painel Claude Code e uma conversa nova. Selecione um modelo habilitado ou use o padrão do arquivo da página 5.

4. Teste e alterne perfis

Envie o teste de resposta OK da página 6. Para sua conta pessoal, volte ao perfil habitual do VS Code. As variáveis da extensão não configuram automaticamente todos os terminais integrados: para o terminal, selecione o perfil como na página 6.

Não exporte nem compartilhe este perfil com a chave preenchida. Se houver erro de JSON, procure a vírgula, as aspas e as barras do caminho antes de trocar a chave.

Windows: configurar o Codex

Instalação e arquivo de perfil separado, sem substituir o config.toml habitual.

1. Confirme a instalação

No PowerShell, consulte a versão. Se não estiver instalado, use npm.cmd install -g @openai/codex. O exemplo abaixo segue o formato de perfis da versão 0.134.0 ou posterior.

```
codex --version
```

2. Crie o arquivo do perfil

Execute os comandos. Se já houver arquivo com esse nome, preserve uma cópia antes de editá-lo. Não altere o auth.json nem apague suas credenciais.

```
$pastaCodexApi = Join-Path $env:USERPROFILE ".codex"
New-Item -ItemType Directory -Force -Path $pastaCodexApi
notepad (Join-Path $pastaCodexApi "acesso-api.config.toml")
```

3. Cole o bloco abaixo e salve

O nome acesso_api é um rótulo local do provedor. A chave não entra nesse arquivo: ele guarda o nome de uma variável que será definida na próxima página.

```
model = "gpt-6-luna"
model_provider = "acesso_api"

[model_providers.acesso_api]
name = "Acesso API"
base_url = "https://api.tn1.top/v1"
wire_api = "responses"
env_key = "SERVICO_API_KEY"
```

PERFIS EM VERSÕES ATUAIS

O perfil fica em acesso-api.config.toml, ao lado do config.toml. Ele não usa um bloco [profiles.acesso-api] dentro do arquivo principal. Se sua versão for anterior à indicada, atualize ou confirme o formato com o atendimento.

O endereço e o modelo acima seguem o guia dedicado do gateway. Se sua ativação informar outros valores, confirme antes de mudar. Criar este arquivo ainda não seleciona o perfil nem valida o acesso.

Windows: chave e teste do Codex

A chave abaixo será usada apenas no processo atual do PowerShell.

1. Digite a chave sem deixá-la no comando

Execute a primeira linha, cole sua chave no campo protegido e pressione Enter. Depois execute as outras duas linhas. O valor não é impresso no terminal.

```
$chaveTemporariaApi = Read-Host "Cole sua chave" -AsSecureString
$credencialApi = [System.Net.NetworkCredential]::new(
  "", $chaveTemporariaApi)
$env:SERVICO_API_KEY = $credencialApi.Password
```

2. Abra o projeto e selecione o perfil

Troque o caminho pela sua pasta. O argumento --profile é o que seleciona a conexão API, sem mudar o padrão do arquivo principal.

```
Set-Location "C:\caminho\do\seu-projeto"
codex --profile acesso-api
```

3. Confirme o provedor e teste

No Codex, confira o cabeçalho e /status quando disponível. O provedor esperado é acesso_api / Acesso API e o modelo é o configurado. Em uma conversa nova, peça apenas OK, sem ferramentas e sem alterar arquivos.

4. Use dentro do VS Code se preferir

Abra Terminal > Novo Terminal no editor e repita a definição de chave nesta sessão e o comando com --profile. Isso usa a CLI dentro do editor. A extensão gráfica e o aplicativo são superfícies diferentes; não presuma que herdem esse perfil.

5. Encerre o uso da API

Saia do Codex e feche esse terminal. Ao abrir um novo terminal, a variável de sessão não estará definida por este procedimento. Para voltar à sua conta, abra o Codex pelo seu modo habitual, sem selecionar acesso-api.

SE O TESTE NÃO RESPONDER

Confira validade da chave, /v1 no endereço, protocolo responses e grafia do modelo. O endpoint precisa suportar a versão do Codex instalada. Um arquivo TOML válido, sozinho, não garante compatibilidade do gateway.

macOS e Linux: Claude Code

Mesmo acesso, com comandos próprios do terminal desses sistemas.

1. Instale a ferramenta

Abra Terminal no macOS ou o terminal da sua distribuição Linux. O comando abaixo é o instalador nativo publicado pela Anthropic. Aguarde o término e abra um terminal novo se o comando claude ainda não for encontrado.

```
curl -fsSL https://claude.ai/install.sh | bash
claude --version
```

2. Crie o perfil separado

Execute as linhas abaixo. O editor nano abre o arquivo para edição. Se a pasta ou o arquivo já existir, faça backup antes de mudar.

```
mkdir -p "$HOME/.claude-acesso-api"
nano "$HOME/.claude-acesso-api/settings.json"
```

3. Preencha a configuração

Cole o JSON completo da página 5 e substitua a chave. O JSON é igual nos três sistemas. No nano, use Ctrl+O para gravar, Enter para confirmar o nome e Ctrl+X para sair.

4. Proteja o arquivo

O arquivo contém sua chave. Restrinja a leitura ao seu usuário com o comando abaixo. Não coloque essa pasta em repositórios ou pastas compartilhadas.

```
chmod 600 "$HOME/.claude-acesso-api/settings.json"
```

5. Abra no projeto usando o perfil

Substitua o caminho de exemplo por uma pasta sua. A variável na mesma linha do comando seleciona o perfil para essa execução.

```
cd "/caminho/do/seu-projeto"
CLAUDE_CONFIG_DIR="$HOME/.claude-acesso-api" claude
```

6. Valide antes de trabalhar

Consulte /status, abra uma conversa nova e peça apenas OK. Para usar sua conta pessoal, encerre a sessão e inicie do modo habitual, sem essa seleção de perfil. Se existir uma variável global antiga, peça ajuda para identificá-la.

Para o painel do VS Code no Mac/Linux, siga as páginas 7-8, usando Cmd no lugar de Ctrl nos atalhos correspondentes e um caminho absoluto do sistema no campo CLAUDE_CONFIG_DIR, como /Users/seu-usuario/.claude-acesso-api.

macOS e Linux: Codex

Perfil separado e chave digitada no terminal, sem gravá-la no arquivo TOML.

1. Instale Node e Codex

Instale uma versão LTS atual do Node.js adequada ao seu sistema. Abra um terminal novo e confira node --version e npm --version. Instale a CLI e confira sua versão:

```
npm install -g @openai/codex
codex --version
```

2. Crie o perfil

Use a versão 0.134.0 ou posterior para o formato de perfil apresentado. Abra o arquivo abaixo e cole o bloco TOML da página 9. Salve sem substituir seu config.toml habitual.

```
mkdir -p "$HOME/.codex"
nano "$HOME/.codex/acesso-api.config.toml"
```

3. Informe a chave com entrada protegida

Os comandos abaixo são para Bash ou Zsh. A chave digitada no read não aparece na tela. Pressione Enter após colar e execute as linhas seguintes. Não substitua o nome da variável pela chave no arquivo TOML.

```
printf "Cole sua chave: "
read -r -s SERVICO_API_KEY
printf "\n"
export SERVICO_API_KEY
```

4. Abra o projeto e inicie

Substitua a pasta do exemplo e selecione o perfil. Confira o provedor e envie um pedido curto, sem alterações de arquivo.

```
cd "/caminho/do/seu-projeto"
codex --profile acesso-api
```

5. Finalize a sessão

Ao sair do Codex, remova a variável do shell ou feche o terminal. Não acrescente a chave ao .zshrc ou .bashrc se a intenção for usá-la somente nessa sessão.

```
unset SERVICO_API_KEY
```

SE HOUVER ERRO DE INSTALAÇÃO

Não resolva erro de permissão do npm executando tudo como administrador ou com sudo sem avaliar o ambiente. Envie ao atendimento a mensagem exata, seu sistema e a versão do Node. O problema pode ser apenas o local em que o npm instala pacotes.

Aplicativos de janela

Caminhos opcionais: a configuração do terminal não configura todos os aplicativos.

Claude Desktop: integração por gateway

Este caminho depende de a versão instalada oferecer inferência de terceiros. A interface pode variar. Use o aplicativo obtido em claude.ai/download.

1. Abra Help > Troubleshooting e procure Enable Developer Mode.
2. Confirme a ativação e aguarde o reinício.
3. Abra Developer > Configure Third-Party Inference.
4. Escolha Gateway.
5. Informe a raiz https://api.tn1.top e sua chave.
6. Use o esquema x-api-key quando solicitado.
7. Preencha a lista com os modelos liberados na sua ativação.
8. Feche o app por completo e reabra.
9. Inicie uma conversa nova e envie o teste de resposta OK.

Se o menu não existir, não edite arquivos internos tentando forçá-lo. Informe a versão ao atendimento. Esse fluxo é para desktop; não configura o app Claude de iPhone.

Codex em aplicativo ou extensão gráfica

O perfil --profile deste manual é selecionado na CLI. Para o aplicativo ou painel gráfico, confirme primeiro como a versão instalada escolhe a configuração. Alterar o provedor no config.toml principal pode fazer outras sessões usarem a API também.

Antes de qualquer mudança no arquivo principal, salve uma cópia. O bloco técnico da página 9 descreve o provedor, mas não deve ser colado sobre suas preferências existentes. Após uma configuração assistida, reinicie completamente o aplicativo e confira a conexão em uma conversa nova.

QUER USAR SEM ALTERAR O PADRÃO?

Use a CLI do Codex dentro do terminal do VS Code, seguindo a página 10. Isso permite escolher o perfil explicitamente a cada sessão. Para configurar aplicativo ou extensão, confirme com o atendimento a versão e a forma de selecionar o provedor.

Um instalador automático pode alterar padrões globais. Se optar por esse caminho com orientação do atendimento, revise as ferramentas selecionadas, confirme o backup e não suponha que as opções de app e CLI sejam equivalentes.

Cursor, Cline, Kilo e Roo

Configuração pelo painel da ferramenta. Faça primeiro um teste curto de conexão.

Cursor

1. Abra Settings > Models.
2. Informe sua chave no campo de API.
3. Ative a opção de endereço OpenAI personalizado.
4. Use https://api.tn1.top/v1.
5. Adicione um identificador habilitado, como gpt-6-luna.
6. Verifique a conexão, selecione o modelo e teste no chat.

A conexão personalizada não inclui automaticamente todos os recursos do Cursor. Autocomplete, edição inline e recursos de agente dependem da versão e da infraestrutura do editor. Se outros modelos pararem de funcionar após ativar o endereço alternativo, reveja essa substituição.

Cline

Abra a engrenagem da extensão, escolha OpenAI Compatible e preencha chave, modelo e endereço. Configure Plan e Act separadamente. Salve e reabra o painel. A documentação do gateway diverge entre raiz e /v1 para essa ferramenta: confirme o endereço aplicável antes de fechar a configuração.

Kilo Code

Na extensão, selecione OpenAI Compatible, base https://api.tn1.top/v1, sua chave e o modelo habilitado. Salve, reabra o painel e envie o teste OK. A configuração do terminal do Kilo é independente da configuração da extensão.

Roo Code

Na interface da extensão, selecione o formato compatível OpenAI e informe https://api.tn1.top/v1, chave e modelo. Salve, reabra o painel e teste numa pasta de exemplo. Se importar settings de outra máquina, remova credenciais antes de compartilhar o arquivo.

AO VALIDAR UMA EXTENSÃO

Conexão, leitura de arquivos e execução de ferramentas são verificações diferentes. Teste cada recurso que pretende usar. Não presuma que uma resposta de texto confirme a execução completa do agente.

Continue e OpenCode

Exemplos de configuração com referência à variável SERVICO_API_KEY.

Continue

Abra o arquivo .continue/config.yaml do seu usuário. Preserve o conteúdo existente e acrescente o modelo conforme o formato aceito pela versão instalada. Este trecho mostra os campos de conexão:

```
models:
  - name: Acesso API
    provider: openai
    model: gpt-6-luna
    apiBase: https://api.tn1.top/v1
    apiKey: ${{ secrets.SERVICO_API_KEY }}
    roles:
      - chat
      - edit
      - apply
```

Configure SERVICO_API_KEY no mecanismo de secrets/ambiente aceito pela sua instalação. Recarregue o editor e selecione o modelo. O processo do editor precisa receber a credencial; definir uma variável em outro terminal não garante isso. Não substitua o arquivo todo por esse trecho se ele exigir outros campos de cabeçalho.

OpenCode

No arquivo .config/opencode/opencode.json do seu usuário, declare um provedor e o modelo. Exemplo mínimo a combinar com a configuração existente:

```
{
  "provider": {
    "acesso_api": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "Acesso API",
      "options": {
        "baseURL": "https://api.tn1.top/v1",
        "apiKey": "{env:SERVICO_API_KEY}"
      },
      "models": {"gpt-6-luna": {"name": "gpt-6-luna"}}
    }
  }
}
```

Defina a chave no terminal conforme a página 10 ou 12 e inicie opencode. Selecione acesso_api/gpt-6-luna. O prefixo identifica o provedor local; não é parte do nome do modelo no gateway.

Crush e Qwen Code

Use um projeto de exemplo para validar antes de trabalhar em arquivos importantes.

Crush

Crie ou edite .config/crush/crush.json do seu usuário. O exemplo usa uma variável para evitar escrever a chave diretamente no JSON:

```
{
  "providers": {
    "acesso_api": {
      "type": "openai-compat",
      "base_url": "https://api.tn1.top/v1",
      "api_key": "$SERVICO_API_KEY",
      "models": [
        {"id": "gpt-6-luna", "name": "Acesso API"}
      ]
    }
  }
}
```

Defina SERVICO_API_KEY no terminal em que vai abrir o programa. Execute crush na pasta do projeto e selecione o modelo. Confira se existe um crush.json no projeto, pois ele pode sobrepor a configuração global.

Qwen Code

O exemplo do gateway usa uma lista de provedores no arquivo .qwen/settings.json do usuário. A extensão exata do esquema depende da versão; preserve as demais opções do arquivo:

```
{
  "modelProviders": [
    {
      "name": "acesso_api",
      "authType": "openai",
      "baseUrl": "https://api.tn1.top/v1",
      "envKey": "SERVICO_API_KEY",
      "model": "gpt-6-luna"
    }
  ]
}
```

Defina a variável no mesmo terminal e inicie qwen. envKey recebe o nome da variável, não o valor da chave. Selecione o modelo e teste. Usar o programa Qwen Code com uma API personalizada não demonstra que o modelo atendendo a solicitação seja Qwen.

Essas integrações são descritas pelo gateway, mas não foram testadas neste manual. Se a interface ou o esquema divergir, informe a versão antes de alterar os campos.

Aider, Goose e Trae Agent

Atenção aos nomes das variáveis: cada ferramenta pode esperar um nome diferente.

Aider

No arquivo .aider.conf.yml, o guia do gateway orienta o endereço raiz e o prefixo openai/ no identificador:

```
openai-api-base: https://api.tn1.top
model: openai/gpt-6-luna
```

Defina a chave na variável OPENAI_API_KEY do processo que executa aider. O programa pode editar arquivos e criar commits; comece numa branch ou projeto de teste. Se houver erro de rota, confirme o comportamento da versão antes de acrescentar /v1.

Goose

No arquivo .config/goose/config.yaml, utilize os campos abaixo. A chave é lida de OPENAI_API_KEY, não de SERVICO_API_KEY, a menos que você configure uma adaptação explícita.

```
GOOSE_PROVIDER: openai
GOOSE_MODEL: gpt-6-luna
OPENAI_HOST: https://api.tn1.top
```

Defina a credencial no mesmo terminal, entre na pasta e execute goose session. O tipo openai seleciona o formato de conexão; não autentica uma conta pessoal OpenAI.

Trae Agent

Este caminho é para trae-cli, não para a IDE Trae. No trae_config.yaml do projeto, declare o provedor com base https://api.tn1.top/v1 e tipo openrouter, conforme o guia do gateway. A chave pode ser lida de OPENROUTER_API_KEY.

O guia exige também a estrutura completa de agente/modelo e campos top_p, top_k, max_retries e parallel_tool_calls. Não substitua um arquivo funcional por um trecho incompleto. Use trae-cli show-config para conferir endereço e presença da chave; depois valide uma tarefa pequena.

QUANDO USAR ESTES CAMINHOS

São opções para quem já utiliza essas ferramentas. Para começar sem experiência, prefira o Claude no VS Code ou uma das CLIs detalhadas nas páginas anteriores. Instalação, permissões e recursos de cada agente podem variar.

Sua primeira tarefa

Um pedido bem definido ajuda o agente a trabalhar com menos tentativas.

1. Comece com leitura

Escolha uma pasta de teste ou um projeto com backup. Peça um resumo da estrutura antes de autorizar mudanças. Confira se a ferramenta está realmente vendo a pasta esperada.

PRIMEIRO PEDIDO

Leia a estrutura deste projeto e explique os arquivos principais. Não altere arquivos e não execute comandos de instalação. Se faltar alguma informação, diga o que precisa saber.

2. Defina um objetivo pequeno

Escolha uma alteração que você consiga conferir, como um texto de botão, um componente simples ou uma correção isolada. Informe onde a alteração deve acontecer e como o resultado será verificado.

EXEMPLO DE PEDIDO

Na página de contato, altere apenas o texto do botão para Enviar mensagem. Preserve o layout. Antes de editar, diga qual arquivo será modificado. Depois, mostre a alteração e explique como conferir.

3. Revise as permissões

Leia pedidos para executar comandos, instalar pacotes, acessar outras pastas ou enviar dados. Autorize somente o necessário para a tarefa. Não forneça senhas de outros serviços em prompts para resolver erros comuns de instalação.

4. Confira a entrega

Veja os arquivos alterados e teste o comportamento. Uma mensagem de conclusão não substitui abrir a página ou executar a verificação adequada. Se o agente disser que não conseguiu testar, peça o motivo e o próximo passo.

5. Mantenha o contexto organizado

Abra nova conversa ao trocar de projeto ou de conexão. Informe linguagem, ferramenta, resultado esperado e mensagem completa do erro. Ao relatar um problema, evite enviar o histórico inteiro sem necessidade.

Esses exemplos orientam o uso. Eles não garantem qualidade de uma resposta específica nem substituem revisão dos resultados produzidos pela IA.

Erros comuns e suporte

Resolva a configuração antes de trocar de chave ou reinstalar tudo.

Sintoma	Primeira verificação
Comando não encontrado	Feche e abra o terminal. Confira a instalação e o PATH. Instalar a extensão não instala necessariamente a CLI global.
Bloqueio de npm.ps1	No Windows, tente npm.cmd, como nos exemplos. Não altere a política do sistema só para contornar o aviso.
401 / acesso inválido	Confira validade e chave inteira. Veja se chegou ao endereço certo e se a variável foi definida no processo atual.
404 / rota inexistente	Confira a combinação entre base, /v1 e protocolo da ferramenta. Evite duplicar /v1 ou o caminho completo.
Modelo não encontrado	Use o identificador habilitado, com letras e espaços exatos. Em OpenCode/Aider, confira o prefixo local.
JSON inválido	Revise aspas, vírgulas e barras do caminho. Não mantenha dois objetos completos no mesmo arquivo.
TOML não carrega	Confira nome do arquivo, seções e versão. O perfil atual fica em acesso-api.config.toml.
Thinking block / erro 400	Abra conversa nova no perfil correto e repita um pedido simples. Se persistir, envie o erro. Não apague seu histórico como primeira medida.
429 / lentidão	Interrompa repetições automáticas e confira disponibilidade com o atendimento. Não interprete como autorização para aumentar concorrência.
API afeta conta pessoal	Confira perfil ativo e variáveis globais antigas. Não faça logout geral ou apague credenciais sem identificar a origem.

Ao pedir ajuda, envie

Sistema operacional, ferramenta e versão, modelo selecionado, endereço configurado, etapa em que falhou e mensagem do erro. Use o mesmo canal em que contratou o acesso. Oculte a chave e qualquer dado privado nas capturas.

SEM EXPOR A CREDENCIAL

Diga apenas se a chave foi reconhecida ou se está ativa. Não envie settings.json, exportação de perfil, auth.json ou logs completos sem antes remover segredos.

Perguntas frequentes

Guarde este manual para consultar a instalação e a renovação.

Qual o valor e por quanto tempo funciona?

R$160 por 30 dias de acesso. Ao final, você decide se deseja renovar. Confira a data de validade informada no atendimento.

Preciso instalar tudo?

Não. Escolha a ferramenta que vai usar. Para o terminal, siga a instalação da CLI. Para o painel do VS Code, instale e configure a extensão no perfil Acesso API.

Isso altera minha assinatura pessoal?

Os passos principais selecionam um perfil de acesso API. O plano pessoal continua dependendo da sua conta e das suas condições próprias. Evite copiar a conexão API para configurações globais ou para o perfil pessoal.

Vai funcionar no iPhone ou Android?

Depende de um cliente que aceite o protocolo e o endereço personalizados. A compatibilidade móvel precisa ser validada; o aplicativo oficial de chat não é configurado automaticamente por este manual.

Todos os recursos e modelos estão incluídos?

Use o catálogo liberado na sua chave. Recursos do editor, recursos de conta, agentes e aplicativos podem ter requisitos próprios. Para dispositivos, limites, disponibilidade e condições adicionais, consulte o que foi combinado na contratação.

Referências técnicas

As instruções de perfil foram adaptadas para este manual. Os exemplos não foram executados com sua chave. As interfaces e os esquemas podem mudar conforme a versão instalada. Referências para suporte e manutenção:

Instalação do Claude Code

Configurações e diretório de perfil do Claude

Extensão Claude Code no VS Code

Perfis do VS Code

Perfis e provedores do Codex

Referência de configuração do Codex

Compatibilidade publicada pelo gateway

Conexão Codex documentada pelo gateway