VISIONAR

TOKENS

Sete ajustes pra gastar menos token no Claude

Os sete ajustes do vídeo, cada um com o comando ou o prompt pra aplicar hoje no Claude Code. A ideia é uma só: mandar menos texto pro modelo a cada mensagem e errar menos no caminho.

Pra quem usa o Claude Code e bate no limite do plano antes do fim do dia. Os ajustes 2, 3 e 7 também valem no aplicativo do Claude.

Passo a passo

1. **Instale o plugin de resposta curta**

   O plugin se chama Caveman. Ele faz o Claude responder em frases secas, sem enfeite, mantendo código e comandos exatos. Cole o comando de instalação no terminal. Pra desligar, digite /caveman off.

2. **Escolha o modelo pelo tamanho da tarefa**

   Digite /model e troque no meio da sessão. A documentação do Claude Code diz que o Sonnet dá conta da maior parte do trabalho de código e que o Opus fica pra decisão difícil. Haiku serve pra tarefa simples e pra subagente.

3. **Resuma o PDF antes de usar**

   Em vez de colar o PDF inteiro na conversa de trabalho, abra uma conversa só pra resumir, com o prompt abaixo. Depois leve só o resumo pra conversa principal.

4. **Enxugue a conversa longa**

   A cada mensagem o Claude Code reenvia a conversa inteira. Use /compact com uma instrução do que guardar. Mudou de assunto? /clear começa do zero e não gasta nada.

5. **Escreva a configuração uma vez no CLAUDE.md**

O arquivo CLAUDE.md, na raiz do projeto, é carregado no começo de toda sessão. Coloque ali o que você repete em todo chat. A documentação recomenda manter esse arquivo abaixo de 200 linhas, porque ele também ocupa contexto.

6 Mande um subagente varrer os arquivos

Leitura de pasta grande, log e documentação enche a sua conversa. Peça pra um subagente no modelo Haiku fazer a varredura e te devolver só o resumo. O subagente também consome do seu plano, só que menos.

7 Peça o plano antes do código

Aperte Shift+Tab até aparecer o modo de plano, ou use o prompt abaixo. O Claude estuda o projeto, te mostra o caminho e só constrói depois que você aprova. Corrigir um plano custa bem menos que corrigir código pronto.

Prompts para copiar

1. Instalar o Caveman (no terminal)

```
claude plugin marketplace add JuliusBrussee/caveman && claude plugin install caveman@caveman
```

3. Resumir um PDF antes de trabalhar com ele

```
Você vai preparar um documento pra eu usar em outra conversa.

Leia o PDF anexado e escreva um resumo de no máximo [NÚMERO] linhas com: os números e datas que aparecem, as decisões ou regras que o documento estabelece, os nomes de pessoas e empresas, e o que ficou em aberto.

O que eu vou fazer com esse resumo: [EX.: MONTAR UMA PROPOSTA PRA ESSE CLIENTE].

Formato: tópicos curtos, sem introdução e sem conclusão. Copie os números do jeito que estão no documento. Se algum trecho estiver ilegível, diga qual página em vez de adivinhar. Não acrescente nada que não esteja no PDF.
```

4. Enxugar a conversa guardando o que importa

```
/compact Guarde as decisões tomadas, os arquivos alterados e o que ainda falta fazer em [NOME DA TAREFA]
```

5. Modelo de CLAUDE.md pra colar e preencher

```
# Projeto
[O QUE É O PROJETO, EM DUAS LINHAS]

# Como eu trabalho
- Responda em português do Brasil, direto, sem repetir o que eu pedi.
- Antes de mudar mais de um arquivo, me mostre o plano e espere eu aprovar.
- [OUTRA REGRA QUE VOCÊ REPETE EM TODO CHAT]

# Onde ficam as coisas
- [PASTA]: [O QUE TEM NELA]
- [PASTA]: [O QUE TEM NELA]

# Comandos
- Rodar: [COMANDO]
- Testar: [COMANDO]

# Não fazer
- [EX.: NÃO APAGAR ARQUIVO SEM PERGUNTAR]
```

6. Varrer arquivos com um subagente Haiku

```
Use um subagente com o modelo Haiku pra esta tarefa, sem trazer o conteúdo dos arquivos pra esta conversa.

Tarefa do subagente: ler a pasta [CAMINHO DA PASTA] e achar [O QUE VOCÊ PROCURA, EX.: ONDE O PREÇO DOS PLANOS É DEFINIDO].

O subagente me devolve só: a lista dos arquivos que importam, com uma linha sobre cada um, e os trechos exatos onde a resposta está. Sem colar arquivo inteiro. Se não achar, diz que não achou e onde procurou.
```

7. Pedir o plano antes de construir

```
Não escreva código ainda. Quero só o plano.

O que eu preciso: [DESCREVA O QUE QUER PRONTO].

Me devolva: os passos na ordem, quais arquivos você vai criar ou mudar em
cada passo, o que pode dar errado e como você vai conferir que funcionou.
Se faltar informação pra decidir alguma coisa, me pergunte agora.

Só comece a construir quando eu responder APROVADO.
```

Links

Caveman

Repositório oficial do plugin de resposta curta, com a instalação e os testes de economia.

Claude Code: gerenciar custos

Página oficial que explica /compact, /clear, /model, modo de plano, subagentes e CLAUDE.md.

Os 40% do Caveman valem contra o Claude sem instrução nenhuma: no teste publicado no repositório, a saída caiu de 6.983 pra 4.119 tokens. Contra um simples 'responda de forma concisa', o ganho medido foi bem menor. Ele corta o que o Claude escreve, e o histórico da conversa continua pesando.

Não achei fonte oficial pro número de 80% do contexto gasto com PDF. O peso depende do tamanho do arquivo. O que a documentação confirma é que tudo que entra na conversa é reenviado a cada mensagem.

O /compact também gasta: ele lê a conversa inteira pra resumir. Se você não precisa mais do assunto anterior, /clear sai de graça.

Conferido em 14/11/2026. Ferramenta muda rápido: se algo não bater, me avisa no direct.