


Domine o Claude.ia: Dicas Rápidas Para Automatizar e Vencer no Jogo da IA

Torne o Claude seu parceiro de produtividade com comandos prontos, workflows e automações práticas. Mesmo se você for iniciante.

Autora: Ana Evelyn

- ✓ Criou sua pasta **.claude/commands/**?
- ✓ Preencheu seu **/memory** com contexto pessoal e técnico?
- ✓ Criou seu **/init** com instruções gerais?

 Introdução	3
1.1 Seja Claro e Específico	5
1.2 Forneça Contexto e Detalhes Relevantes	5
1.3 Use Exemplos (Few-Shot Prompting)	6
1.4 Quebre Tarefas Complexas	6
1.5 Dê um "Papel" ao Claude (System Prompts)	6
1.6 Peça para o Claude "Pensar" (Chain-of-Thought)	7
Capítulo 5- Dicas Essenciais para Otimizar sua Experiência	13
5.2 Formato de Saída	14
5.3 Personalização de Marca	14
5.4 Lidar com Dados e Documentos	14
5.5 Segurança e Responsabilidade	14
Conclusão	15
O Claude.ai como seu Aliado Inteligente	15

Introdução

O Claude.ai, desenvolvido pela Anthropic, é um assistente de IA projetado para ser útil, inofensivo e honesto. Ele se destaca em uma variedade de tarefas, desde a geração de conteúdo e análise de dados até a assistência em programação e automação de fluxos de trabalho. Este ebook foi criado para guiá-lo através das melhores práticas e dicas para extrair o máximo valor do Claude.ai, seja para uso pessoal ou profissional.

Claude.ai é uma poderosa inteligência artificial com grande capacidade de raciocínio e memória. Mas, para extrair todo seu potencial, é preciso saber se comunicar com ela de maneira clara, estratégica e reutilizável. Este pequeno guia foi criado para ajudar você a padronizar suas interações, ganhar tempo e tornar a IA parte do seu fluxo de trabalho diário.

O que você vai aprender:

Transformar Claude.ai em seu parceiro de produtividade, economizar tempo e criar seu agente pessoal de IA para automatizar interações e aumentar produtividade no trabalho e estudos.

Você já usa o Claude.ai mas sente que ele poderia ser muito mais útil?

Neste guia curto e direto, você vai aprender como configurar comandos, criar memórias persistentes e iniciar conversas sempre com a IA pronta para trabalhar do seu jeito.

Com poucos arquivos e uma estrutura simples, você poderá:

- ✓ Automatizar interações
- ✓ Economizar tempo
- ✓ Criar seu "agente pessoal de IA"
- ✓ Aumentar sua produtividade no trabalho e nos estudos

Ideal para devs, empreendedores, estudantes ou qualquer pessoa que quer extrair o máximo da IA com o mínimo de esforço.

Capítulo 1 – A Arte de Criar Prompts Eficazes

A qualidade da sua interação com o Claude.ai depende diretamente da clareza e especificidade dos seus prompts. Pense no prompt como a "instrução" que você dá à IA.

1.1 Seja Claro e Específico

- **Início da Mensagem:** Comece sua mensagem declarando claramente sua tarefa ou pergunta.
 - **Ruim:** "Me ajuda a escrever."
 - **Bom:** "Preciso de um rascunho de e-mail para anunciar o lançamento de um novo produto para clientes. O produto é uma ferramenta de gerenciamento de projetos."

1.2 Forneça Contexto e Detalhes Relevantes

Quanto mais contexto você der, melhores serão os resultados.

- **Cenário:** "Estou escrevendo um artigo de blog sobre os benefícios da meditação para profissionais de tecnologia. O tom deve ser inspirador e prático."
- **Público-alvo:** "O conteúdo é para empreendedores iniciantes, portanto, evite jargões técnicos e use uma linguagem acessível."

1.3 Use Exemplos (Few-Shot Prompting)

Se você tem um formato ou estilo específico em mente, mostre ao Claude.

- **Exemplo:** "Por favor, resuma este texto no estilo de um tweet curto e impactante, como este: 'Novo estudo revela X, transformando Y! #Inovação'"

1.4 Quebre Tarefas Complexas

Para projetos grandes, divida-os em etapas menores.

- **Em vez de:** "Escreva um e-book completo sobre marketing digital."
- **Tente:**
 1. "Gere um índice para um e-book sobre marketing digital, com 5 capítulos."
 2. "Para o Capítulo 1: 'Fundamentos do Marketing Digital', escreva uma introdução de 200 palavras."
 3. "Para o Capítulo 1, liste os principais conceitos e suas definições em formato de bullet points."

1.5 Dê um "Papel" ao Claude (System Prompts)


Instrua o Claude a assumir uma persona para respostas mais direcionadas.

- "Você é um especialista em marketing digital. Elabore uma estratégia de lançamento para um novo aplicativo de fitness."
- "Atue como um editor de texto. Revise a gramática e o estilo deste parágrafo."

1.6 Peça para o Claude "Pensar" (Chain-of-Thought)

Incentive o Claude a mostrar seu raciocínio para tarefas complexas.

- "Passo a passo, explique como você chegaria à solução para este problema de matemática: [problema]."
- "Analise este relatório financeiro e, antes de me dar a conclusão, liste os três principais pontos de dados que você considerou."

 **DICA PROFISSIONAL:** Use templates com variáveis para reutilização rápida. Exemplo: [PRODUTO], [PÚBLICO], [TOM] nos seus prompts.

Capítulo 2 – Use o `/commands` para criar um padrão de projeto

Você pode transformar tarefas repetitivas em comandos prontos. Crie sua pasta `.claude/commands/` e dentro dela coloque arquivos como:

Exemplo de Arquivo:

```
# SIGA AS INSTRUÇÕES DE TAREFA
Siga as instruções de prompt passado entre <instrucoes></instrucoes>
para executar as tarefas informadas como argumentos.

## Tarefas a serem executadas: $ARGUMENTS

## INSTRUÇÕES PARA EXECUÇÃO DA TAREFA

<instrucoes>
Você é um especialista em desenvolvimento de software, arquitetura de
software e em todas as habilidades envolvidas na construção de
software, seja para projetos pequenos ou sistemas de grande escala.
Inclusive especialista em Testes de Software para resolver problemas de
BUGS e inconsistências.

Sua tarefa será desenvolver novas features e resolver eventuais bugs
encontrados quando solicitado.

Seu raciocínio deve ser minucioso, e não há problema se for muito
longo. Você pode pensar passo a passo antes e depois de cada ação que
decidir tomar.

Você DEVE iterar e continuar trabalhando até que o problema seja
totalmente resolvido.

Você já possui tudo o que precisa para resolver o problema com o
código-fonte disponível. Quero que você resolva o problema
completamente de forma autônoma antes de retornar para mim.

Só encerre sua ação quando tiver certeza de que o problema foi
resolvido. Analise o problema passo a passo e certifique-se de
```

verificar se as suas alterações estão corretas. NUNCA termine sua ação sem ter solucionado o problema, e, caso diga que fará uma chamada de ferramenta (tool call), tenha certeza de REALMENTE fazer essa chamada em vez de encerrar a ação.

Utilize a Internet para buscar documentações necessárias em caso de dúvidas de implementação, ou outros, que possam. Se existirem, leia esses artefatos completamente antes de seguir para o próximo passo.

****Fluxo de Trabalho Detalhado:****

3. ****Investigue a base de código.**** Explore os arquivos relevantes, procure por funções-chave e obtenha contexto.
 4. ****Desenvolva um plano de ação claro, passo a passo.**** Divida em formato de tarefas gerenciáveis e incrementais.
 5. ****Implemente o desenvolvimento de forma incremental.**** Faça alterações pequenas e testáveis no código.
 6. ****Em caso de erros ou falhas, faça o debug conforme necessário.**** Utilize técnicas de depuração para isolar e resolver problemas.
 7. ****Teste frequentemente.**** Execute scripts de testes para verificar se o sistema está funcionando. Esses scripts podem ser testes automatizados ou mesmo scripts avulsos criados exatamente para simular a aplicação.
 8. ****Em caso de bugs, itere até que a causa raiz esteja corrigida e todos os testes passem.****
 9. ****Em caso de interrupção pelo usuário com alguma solicitação ou sugestão:**** Entenda sua instrução e contexto, realize a ação solicitada, e entenda passo a passo como essa solicitação pode ter impactado suas tarefas e plano de ação. Atualize seu plano de ação e tarefas e continue de onde parou sem voltar a dar o controle ao usuário.
 10. ****Em caso de interrupção pelo usuário com alguma dúvida:**** Dê sempre uma explicação clara passo a passo. Após a explicação, pergunte ao usuário se você deve continuar sua tarefa de forma autônoma sem voltar o controle ao usuário.
 11. ****Utilize sempre boas práticas de desenvolvimento, como SOLID, Clean Code:**** Utilize boas práticas e código limpo
- </instrucoes>

Workflow

11. Realização de Alterações no Código

- Remover logs, instruções e mensagens descritivas que utilizou para entender o problema.

- Para testar hipóteses, adicione declarações ou funções de teste.
- Reavalie seus pressupostos caso comportamentos inesperados ocorram.
- NUNCA crie scripts e arquivos totalmente isolados no projeto apenas para executar testes, provas de conceito, incluindo arquivos `.sh`, `makefiles`, entre outros.
- NUNCA faça upgrade ou altere versões de bibliotecas e/ou frameworks utilizados no projeto, mesmo que você não esteja encontrando uma solução.
- Quando for instalar uma dependência utilize sempre a última versão. Caso ache necessário, consulte a @web para garantir que você realmente está utilizando a última versão.
- Evite ao máximo criar complexidades desnecessárias. Mantenha sempre o código simples, claro, objetivo e expressivo. Evite a criação demasiada de Interfaces, porém, não deixe de utilizá-las, principalmente em casos de alto acoplamento entre componentes.

2.1 Como utilizar?

1. Crie uma pasta `.claude/commands/`.
2. Dentro dela, crie um arquivo chamado `exec-prompt.md`.
3. Escreva ali um prompt base com o comportamento desejado para seu agente.

```
> /exe-prompt Implemente ou mude a logica de Visualização do Payload em Relatórios de Dashboard.
Ao clicar no botão "Ver Payload" no relatório do dashboard, será exibido o payload que foi enviado na automação do agente de teste que falhou. Isso permitirá que o usuário possa facilmente testar manualmente o endpoint no Postman.

▶▶ auto-accept edits on (alt+m to cycle)
```

Capítulo 3 – Use o `/memory` para Ensinar o Agente Sobre Você

O comando `/memory` permite que você defina dados persistentes que o Claude irá lembrar entre sessões. Isso é excelente para:

- Ensinar seu estilo de escrita.
- Gravar projetos que você está trabalhando.
- Explicar preferências de resposta.

```
Sou engenheira de software focada em QA. Utilizo Cypress, Playwright e Postman.  
Meus projetos atuais envolvem automação de APIs e UI para a plataforma ImportX.  
Prefiro respostas com linguagem técnica direta e exemplos práticos.
```

Exemplo Prático de Memória Avançada:

```
/memory  
  
=== PERFIL PROFISSIONAL ===  
• Função: [Sua área e senioridade]  
• Empresa: [Contexto do negócio]  
• Especialidades: [Suas expertises]  
• Projetos atuais: [Foco atual]  
  
=== ESTILO DE TRABALHO ===  
• Metodologia: [Framework preferido]  
• Ferramentas: [Stack tecnológico]  
• Preferências: [Como gosta das respostas]  
• Evitar: [O que não funciona para você]
```

Capítulo 4 – Use o `/init` Para Preparar o Ambiente

O `/init` é o local perfeito para colocar tudo que seu agente precisa saber logo ao ser iniciado.

É como um *boot inicial* com instruções permanentes para que ele sempre se comporte do mesmo jeito ao ser invocado.

Você pode utilizar de várias maneiras:

1. Ao iniciar um projeto digite na sua linha de comando CLI do Claude o seguinte: `/init`

```
> /init

/init Initialize a new CLAUDE.md file with codebase documentation
(Install with npm, Set up Claude GitHub Actions, etc.)
```

Esse comando irá fazer com que o claude.ai entenda todo o seu projeto e criar um arquivo do tipo `.md` para ele aprender seu projeto, como realizar instalações, como utilizar, como fazer deploy, Operações e Banco de Dados, Manutenção do Projeto etc..

```
# CLAUDE.md

This file provides guidance to Claude Code (claude.ai/code) when
working with code in this repository.

## Common Development Commands

### Quick Start
```bash
Install dependencies
python instalar_dependencias.py
pip install -r requirements.txt

Set up real database connection
python setup_banco_real.py
```
```

2. Utilizar para dizer o que o agente é e o que deve fazer:

Você é meu assistente de engenharia de testes. Toda vez que eu pedir um caso de teste, você deve usar a linguagem Gherkin. Quando eu pedir um fluxo, gere em estilo de mapa mental. Quando eu pedir automação, use Cypress com boas práticas.

Você é um assistente de QA. Sempre gere casos de teste em Gherkin. Use linguagem técnica.

Template Avançado para Desenvolvimento:

```
/init workflow=desenvolvimento-web

=== CONFIGURAÇÃO AUTOMÁTICA ===
✓ Modo: Desenvolvedor Full-Stack experiente
✓ Stack: React + Node.js + PostgreSQL
✓ Qualidade: Sempre incluir testes e documentação
✓ Segurança: Validar inputs, sanitizar dados
✓ Performance: Otimizar queries e renderização

=== CHECKLIST AUTOMÁTICO ===
Para cada tarefa de código:
1. Analisar requisitos e edge cases
2. Sugerir arquitetura/padrões apropriados
3. Gerar código limpo e comentado
4. Criar testes unitários
5. Documentar APIs e funções
6. Revisar segurança e performance
```

Capítulo 5- Dicas Essenciais para Otimizar sua Experiência

Para ter uma experiência ainda melhor com o Claude.ai, considere estas dicas:

5.1 Refinamento Iterativo

Se a primeira resposta não for perfeita, refine seu prompt.

- "Isso foi bom, mas você poderia expandir o segundo parágrafo e torná-lo mais persuasivo?"
- "Ainda não é exatamente o que eu procurava. Gostaria que o tom fosse mais formal e direto."

5.2 Formato de Saída

Especifique o formato desejado para a resposta.

- "Liste em formato de bullet points..."
- "Gere uma tabela com três colunas: [coluna 1], [coluna 2], [coluna 3]."
- "Gere o resultado em formato HTML para que eu possa abrir no navegador." (Útil para relatórios e documentos interativos)
- "Inclua um botão 'Salvar como PDF' na saída HTML."

5.3 Personalização de Marca

Se você estiver gerando conteúdo para sua marca, forneça informações visuais.

- "Use as cores da minha marca (ex: #RRGGBB) neste documento."
- "Inspire o design deste documento neste arquivo de design que estou anexando."
- "Inclua meu logo (link para a imagem) no cabeçalho."

5.4 Lidar com Dados e Documentos

- **Transcrição de Notas:** Tire uma foto de suas anotações manuscritas e peça ao Claude para transcrever em texto digital.

- **Análise de Dados Não Estruturados:** Peça ao Claude para extrair informações importantes de grandes volumes de dados textuais.

5.5 Segurança e Responsabilidade

- **Verificação:** Sempre revise e verifique o conteúdo gerado pela IA para garantir precisão e adequação ao contexto.
- **Limitações:** Esteja ciente de que, como qualquer IA, o Claude pode ocasionalmente gerar informações imprecisas ou "alucinações".

⚡ **DICA:** Sistemas organizados são 300% mais eficientes. Dedique tempo inicial à estruturação.

Conclusão

O Claude.ai como seu Aliado Inteligente

O Claude.ai é mais do que uma ferramenta; é um colaborador inteligente que pode expandir suas capacidades e otimizar seus processos. Ao dominar a arte da criação de prompts e explorar suas funcionalidades avançadas, você estará no caminho certo para transformar a maneira como você trabalha, cria e inova.

Continue experimentando, aprendendo e adaptando suas abordagens. A IA está em constante evolução, e a chave para o sucesso é a exploração contínua e a aplicação inteligente.

Referências e Próximos Passos

Anthropic Claude Documentation:

<https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/overview>

 **Sucesso na sua jornada de automação com IA**

 **APÓS 30 DIAS DE USO:**

- 80% redução em tarefas repetitivas
- 15-20 horas semanais economizadas
- 300% aumento na produtividade
- Qualidade superior nas entregas

Sobre a Autora:

Sou Ana Evelyn, engenheira de qualidade com mais de 5 anos de experiência em Qualidade de Software, especializada em automação com IA e produtividade digital. Minha trajetória profissional é marcada por entregas de alto impacto, incluindo a redução de 30% nos custos com AWS e pegada de carbono em projetos críticos, além da implementação de testes inteligentes de API utilizando heurísticas avançadas como VADER para empresas como a Samsung no Cesar.

Como pioneira na criação de Agentes de IA nos processos de testes, venho evangelizando a squad QA na Ferreira Costa com práticas DevOps e cultura Shift Left Testing, contribuindo ativamente para a maturidade dos processos de QA. Minha paixão por criar produtos mais seguros, eficientes e centrados no usuário, combinada com minha expertise em automação e IA, me posiciona como uma das principais referências em transformação digital e otimização de processos através da inteligência artificial.

Neste guia, compartilho dicas profissionais que utilizo no meu dia a dia para automatizar processos complexos, economizar dezenas de horas semanais e aumentar drasticamente a produtividade da Squad QA e projetos através do uso inteligente do Claude e outras ferramentas de IA.

DICAS de CLAUDE.IA - Dicas Práticas para o dia a dia que todo desenvolvedor precisa saber

© 2025 Ana Evelyn - Todos os direitos reservados