

# Data-Driven Decision Making com IA



Rocketseat © 2025  
Todos os direitos reservados

[rocketseat.com.br](https://rocketseat.com.br)

Por Mayra Melo



# Por que Prompt Engineering importa?

**Dados sem contexto =  
decisões ruins.**

**Analogia: Prompt = query para  
IA (como SQL para banco).**

**IA só entrega qualidade se o  
prompt for claro.**

**Impacto direto em arquitetura,  
documentação, automação,  
comunicação técnica.**

# Anatomia de um Prompt

Contexto do problema

1

Papel da IA

2

Tarefa específica

3

Formato da saída

4

Tom/estilo

5

Restrições

6

# Exemplo Estruturado

- Você é um *arquiteto de software*.
- **Contexto:** precisamos avaliar migração de monólito para microsserviços.
- **Tarefa:** liste 4 pontos positivos e 4 negativos.
- **Formato:** tabela comparativa.
- **Tom:** técnico, mas sem jargões avançados.

# Técnicas Avançadas de Prompt Engineering

## Systematic Role Framing Papéis compostos para profundidade

- **Ideia:** não basta só dizer "aja como um desenvolvedor".  
Você pode empilhar papéis para ganhar respostas mais ricas.
- **Exemplo:** “Você é um arquiteto de soluções com experiência em compliance bancário e otimização de custos em cloud”.
- **Por que usar:** isso orienta a AI a responder de forma multidimensional, refletindo mais a realidade de um projeto de software.

# Técnicas Avançadas de Prompt Engineering

## Context Enrichment

Adicionar stack, budget, restrições de negócio

- **Ideia:** Enriquecer o prompt com variáveis de contexto real (*tecnologia, custo, prazo, SLA*).
- **Exemplo:** “*Considere que o time usa Python + Spark, tem budget de 20k/mês em cloud, precisa rodar em infra própria e atender LGPD.*”
- **Por que usar:** sem esse detalhe, a AI propõe soluções irreais.  
Com o contexto, você força respostas aplicáveis ao cenário de decisão.

# Técnicas Avançadas de Prompt Engineering

## Constraint-Driven Prompts Impor limites

**Ideia:** Restringir o formato ou as opções que a AI pode usar.

Exemplo:

- “*Liste 3 alternativas sem usar Kubernetes.*”
- “*Explique em máximo 4 linhas.*”
- “*Retorne apenas em JSON válido.*”

**Por que usar:** Evita respostas vagas ou muito extensas, foca no que importa.

# Técnicas Avançadas de Prompt Engineering

## Multi-Lens Prompting Múltiplas perspectivas

- **Ideia:** Pedir que a AI analise o problema sob papéis diferentes (dev, arquiteto, CTO, cybersecurity).
- **Exemplo:** “Explique como esse design impacta do ponto de vista de um dev, de um arquiteto e de um profissional de cybersecurity.”
- **Por que usar:** Simula uma reunião técnica multidisciplinar sem precisar reunir todas as pessoas.

# Técnicas Avançadas de Prompt Engineering

Output as Artifact  
Pedir entregáveis concretos

**Ideia:** Pedir que a AI já devolva em formato útil.

Exemplos:

- Checklist de deploy.
- Tabela de comparação de serviços.
- JSON com parâmetros de configuração.
- Diagrama em Mermaid ou PlantUML.

**Por que usar:** Economiza tempo, você já recebe algo pronto para colar no Jira, Confluence ou código.

# Técnicas Avançadas de Prompt Engineering

**Progressive Deepening**  
**Visão macro: Detalhamento refinado**

**Ideia:** não pedir tudo de uma vez.

Primeiro visão geral, depois ir refinando com follow-ups.

**Exemplo:**

1. “Explique em alto nível a arquitetura.”
2. “Agora detalhe a parte de dados.”
3. “Agora gere código de exemplo para a ingestão.”

**Por que usar:** Evita respostas caóticas e cria um processo de exploração iterativa.

# Exemplo ChatGPT - Técnicas Avançadas

## Prompt original:

*“Compare Kafka e Python em termos de velocidade, facilidade de uso e casos comuns.  
Responda em tabela.”*

## Prompt avançado:

*“Você é um especialista em tecnologia com 10 anos de experiência. Compare Kafka e Python em termos de velocidade, facilidade de uso e casos comuns. Responda em uma tabela clara com três colunas: ‘Critério’, ‘Kafka’ e ‘Python’. Use linguagem simples, adequada para um público técnico iniciante, e inclua uma breve conclusão no final.”*

## Técnicas usadas:

- **Systematic Role Framing:** define o papel (“especialista em tecnologia”).
- **Context Enrichment:** adiciona experiência e público alvo.
- **Constraint-Driven Prompt:** especifica formato (tabela) e tom (linguagem simples).

# Exemplo Manus AI - Técnicas Avançadas

## Prompt original:

*“Transforme esse relatório de análise de dados em um resumo estratégico para reunião com stakeholders.”*

## Prompt avançado:

*“Você é um consultor executivo com experiência em apresentações corporativas. Reescreva este relatório técnico de análise de dados em um resumo estratégico de no máximo 300 palavras, com linguagem clara e objetiva, focada em insights acionáveis. Estruture o texto em bullet points e inclua um título chamativo no início.”*

## Técnicas usadas:

- **Systematic Role Framing:** define papel (“consultor executivo”).
- **Context Enrichment:** especifica formato e público (“bullet points”, “stakeholders”).
- **Constraint-Driven Prompt:** limite de palavras e estrutura.

# Exemplo Lovable - Técnicas Avançadas

## Prompt original:

*“Crie um script em Python que leia um arquivo JSON com registros de vendas, filtre entradas acima de \$1.000 e gere um gráfico em PNG.”*

## Prompt avançado:

*“Você é um engenheiro Python focado em automação de dados. Crie um script em Python bem documentado que:*

1. *Leia um arquivo JSON chamado ‘vendas.json’.*
2. *Filtre registros com valor acima de \$1.000.*
3. *Gere um gráfico em PNG mostrando as vendas filtradas.*

*Use comentários explicativos em cada etapa e escreva o código de forma clara e eficiente.”*

## Técnicas usadas:

- **Systematic Role Framing:** define papel (“engenheiro Python”).
- **Context Enrichment:** adiciona detalhes técnicos e nome de arquivo.
- **Constraint-Driven Prompt:** inclui requisitos de documentação e clareza no código.

# Template Final de Prompt

[Papel da IA]

[qual a atuação dela]

Contexto

[cenário + objetivo]

Tarefa específica

[o que deve entregar]

Formato da saída

[lista, tabela, JSON, texto curto ...]

Tom/estilo

[técnico, executivo, simples...]

Restrições

[o que evitar]