Engenharia de Prompt – Resumo Estruturado

# 1. Fundamentos de Modelos de Linguagem (LLM)

• Aprendem padrões da linguagem humana para prever a próxima palavra em uma sequência.

• Utilizam embeddings numéricos para representar relações semânticas entre palavras.

## 1.1 Word Embeddings

Representam palavras como vetores em um espaço de alta dimensão, permitindo medir similaridade (ex.: "rei" – "homem" + "mulher" ≈ "rainha").

## 1.2 Tokens

Unidades básicas de entrada de um LLM. Podem ser palavras completas ou subpalavras (ex.: "infeliz" → "in", "feliz").

## 1.3 Temperatura

Controla a aleatoriedade da geração. Temperatura baixa → respostas mais determinísticas; alta → respostas mais criativas.

# 2. Princípios de Engenharia de Prompt

• Seja claro e específico nas instruções.

• Divida tarefas complexas em subtarefas.

• Peça que o modelo descreva seus passos ou justificativas.

• Gere múltiplas respostas e selecione a melhor.

• Ajuste parâmetros (temperatura, top‑p) conforme o objetivo.

# 3. Técnicas Avançadas

## 3.1 Chain‑of‑Thought (CoT)

Solicita ao modelo que "pense em voz alta" antes de responder para problemas que exigem raciocínio lógico.   
Exemplo de prompt: "Mostre seu raciocínio passo a passo antes da resposta final."

## 3.2 Least‑to‑Most (LtM) Prompting

Faz o modelo decompor um problema em partes menores, resolvendo do mais simples ao mais complexo.

## 3.3 Chain‑of‑Verification

Após gerar uma resposta, peça ao modelo que verifique cada afirmação com citações ou fontes, reduzindo alucinações.

## 3.4 Self‑Consistency

Execute o mesmo prompt várias vezes (com temperatura >0) e escolha a resposta mais frequente ou coerente.

# 4. Boas Práticas

• Defina formato de saída (bullet points, JSON, tabela).

• Especifique persona ou tom de voz desejado.

• Limite escopo para evitar respostas vagas.

• Combine técnicas: CoT + LtM + verificação para maior robustez.

# 5. Conversão de Imagens

As figuras originais foram substituídas por descrições textuais sintéticas para manter o documento auto‑contido. Nenhuma informação essencial foi perdida.

# 6. Conclusão

Uma boa engenharia de prompt é iterativa: formule, teste, avalie e refine até que o resultado atenda ao objetivo com o menor custo possível.