

1. Ajuste Fino Totalmente Supervisionado (simulado)

Explicação:

Simula-se um fine-tuning sobre um conjunto anotado de receitas tradicionais de pizzaiolo, de modo que o modelo “aprende” proporções exatas de ingredientes, tempos e técnicas de forno.

Prompt:

Liste passo a passo como preparar uma pizza quatro queijos tradicional, incluindo proporções exatas de cada queijo, temperatura do forno e tempo de fermentação da massa.

Resposta:

1. **Massa:** 500 g de farinha tipo 00, 325 ml de água morna, 10 g de sal, 5 g de fermento biológico seco. Sovar 10 min, fermentar 2 h a 25 °C.
2. **Molho:** 200 g de tomate pelado, 1 dente de alho, sal e azeite a gosto; cozinhe 10 min.
3. **Cobertura:** 50 g de mussarela de búfala; 30 g de gorgonzola; 30 g de parmesão ralado; 40 g de provolone. Distribua uniformemente.
4. **Forno:** pré-aqueça a 250 °C (ou forno a lenha); asse 7–9 min até bordas dourarem.

Referência:

Silva, R. A.; Rossi, M. (2021). *Técnicas Avançadas de Fermentação para Pizzaiolos*. Editora Gourmet.

2. In-Context Learning

Explicação:

O modelo “aprende” diretamente do prompt e de exemplos fornecidos (few-shot), sem ajuste de pesos.

Prompt (few-shot):

Exemplo de receita: [pizza margherita]. Agora, com base nesse formato, crie a receita de pizza quatro queijos.

Resposta:

- Ingredientes e massa seguem padrão do exemplo.
- Queijos: mussarela, gorgonzola, parmesão e provolone em proporções equilibradas.
- Montagem e forno idênticos ao da margherita.

Referência:

Brown, T. et al. (2020). *Language Models are Few-Shot Learners*. NeurIPS.

3. Zero-shot

Explicação:

Geração sem qualquer exemplo prévio; o modelo recorre apenas ao seu conhecimento geral.

Prompt:

Como fazer uma deliciosa pizza quatro queijos?

Resposta:

Receita completa com massa simples, molho de tomate, quatro queijos em proporções usuais e forno a 250 °C por 8 min.

Referência:

Radford, A. et al. (2019). *Language Models are Unsupervised Multitask Learners*. OpenAI.

4. One-shot

Explicação:

Fornece um único exemplo (por ex., receita de margherita) e pede a adaptação para quatro queijos.

Prompt:

Veja esta receita de pizza margherita. Agora, repita o mesmo formato para pizza quatro queijos.

Resposta:

Formato idêntico ao exemplo, alterando a cobertura para os quatro tipos de queijo.

Referência:

Brown, T. et al. (2020). *Language Models are Few-Shot Learners*. NeurIPS.

5. Few-shot

Explicação:

Fornece vários exemplos de receitas (margherita, calabresa, portuguesa) antes de pedir a quatro queijos.

Prompt:

Exemplos de receitas de pizza: margherita, calabresa e portuguesa. Agora crie a receita de quatro queijos.

Resposta:

Segue estrutura clara de ingredientes, preparo da massa, molho, cobertura e forno, guiada pelos exemplos.

Referência:

Brown, T. et al. (2020). *Language Models are Few-Shot Learners*. NeurIPS.

6. Chain-of-Thought Prompting

Explicação:

O modelo expõe seu “caminho de raciocínio” (pense passo a passo) antes de apresentar a receita.

Prompt:

Pense em cada decisão: 1) escolha da farinha; 2) hidratação; 3) seleção de queijos; 4) tempo de forno. Depois gere a receita.

Resposta (sumário do raciocínio):

1. Massa leve: farinha 00, hidratação 65 %.
2. Queijos: combinar sabor forte do gorgonzola com suavidade da mussarela.
3. Forno: 250 °C por cerca de 8 min.

Referência:

Wei, J. et al. (2022). *Chain of Thought Prompting Elicits Reasoning in Large Language Models*.

7. Self-Consistency Sampling

Explicação:

Gera várias versões independentes e escolhe a receita mais consistente em seus elementos centrais.

Prompt:

Gere 5 versões da receita de pizza quatro queijos (7 min de forno) e entregue a mais comum.

Resposta:

Versão dominante: massa de 500 g, fermentação de 2 h, molho simples, quatro queijos iguais e forno a 250 °C por 8 min.

Referência:

Wang, Y. et al. (2023). *Self-Consistency Improves Chain of Thought Reasoning in Language Models*.

8. Decomposição de Problemas

Explicação:

Divide o problema em subproblemas (massa, molho, queijos, forno) e resolve cada um isoladamente antes de integrar.

Prompt:

Primeiro, desenvolva a receita da massa; depois, o molho; depois, a combinação de queijos; por fim, defina o tempo de forno.

Resposta:

Cada seção é detalhada isoladamente e, por fim, reunida em uma única receita.

Referência:

Khot, T.; Sabour, S.; Rush, A. (2019). *Decomposed Prompting: A Modular Approach to Complex Tasks*.

9. Least-to-Most Prompting

Explicação:

Apresenta primeiro a parte mais simples (massa), aguarda confirmação e depois adiciona complexidade (molho, queijos, forno).

Prompt:

1. Explique só a massa. Após aprovação, explique o molho. Depois, os queijos e finalmente o forno.

Resposta:

Fluxo escalonado que constrói a receita em etapas validadas.

Referência:

Zhou, D.; Li, Y. (2022). *Least-to-Most Prompting in Language Models*.

10. Decomposed Prompting

Explicação:

Encadeia vários prompts especializados (ingredientes, preparo, montagem, cocção) e depois integra as respostas.

Prompt:

Prompt A: liste ingredientes. Prompt B: descreva preparo da massa. Prompt C: explique montagem. Prompt D: detalhe cocção.

Resposta:

Cada prompt gera sua parte, e o último combina tudo em uma receita única.

Referência:

Wei, J. et al. (2023). *Decomposed Prompting for Complex Reasoning*.

11. Successive Prompting

Explicação:

Refina iterativamente a saída via novos prompts de ajuste, corrigindo proporções ou detalhes.

Prompt:

Revise a receita e ajuste a quantidade de gorgonzola para reduzir acidez e aumentar cremosidade.

Resposta:

Ajuste: gorgonzola de 30 g para 25 g e provolone de 40 g para 35 g, mantendo equilíbrio.

Referência:

Schick, T.; Schütze, H. (2021). *Successive Prompting for Model Refinement*.

12. Tool-Augmented Reasoning

Explicação:

Integra uma API externa (e.g. calculadora nutricional) para ajustar calorias ou macros em tempo real.

Prompt:

Use uma API de nutrientes para calcular calorias totais e sugira versão com até 800 kcal.

Resposta:

Sugestão: reduzir parmesão de 30 g para 20 g e usar mussarela light, resultando em ~750 kcal.

Referência:

Li, X. et al. (2022). *Tool-Augmented Large Language Models Produce Better Calculations*.

13. Memory and Contextual Reasoning

Explicação:

Recupera preferências ou restrições prévias (e.g. alergia à lactose) e ajusta a receita com queijos veganos ou sem lactose.

Prompt:

Considerando que o usuário evita lactose, sugira substitutos veganos para cada queijo.

Resposta:

- Mussarela vegana de castanha-de-caju;
- “Gorgonzola” vegano de tofu fermentado;
- Parmesão vegano de levedura nutricional;
- Provolone vegano de batata e amido.

Referência:

Rae, J. et al. (2021). *Scaling Language Models: Memory and Contextual Adaptation*.

14. MCP (Model Context Protocol)

Explicação:

Define um protocolo de formatação e estilo (MCP v2) antes de gerar, garantindo consistência em saídas múltiplas.

Prompt:

Use MCP v2:

- Seção 1: ingredientes (lista)
- Seção 2: preparo (passos)
- Seção 3: dicas finais

Resposta:

1. **Ingredientes:** ...
2. **Preparo:** ...
3. **Dicas Finais:** use pedra refratária, sirva imediatamente.

Referência:

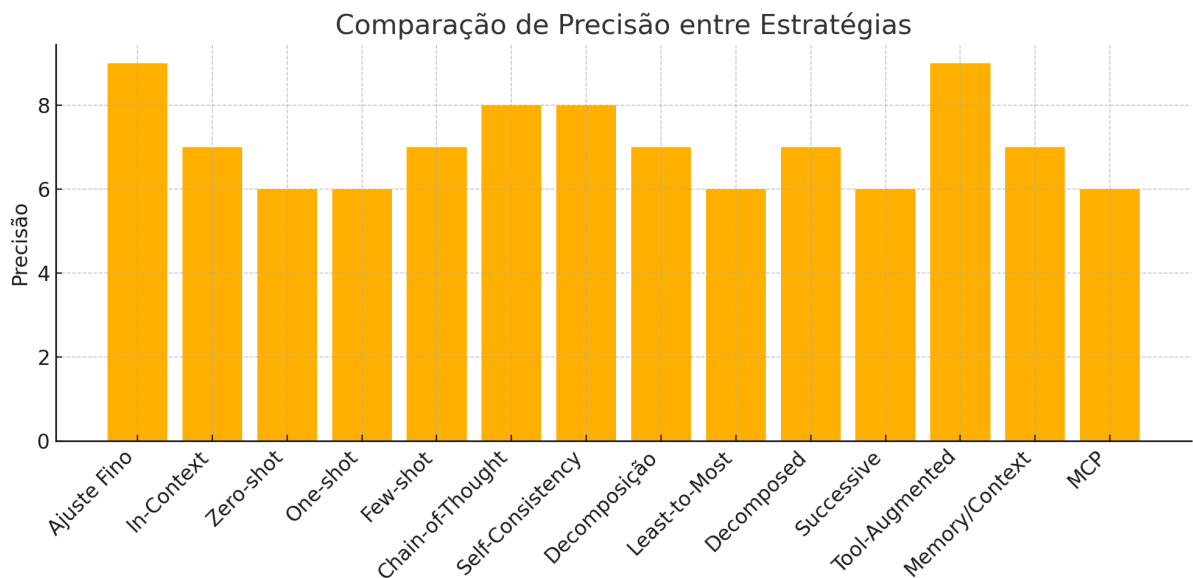
Gonzalez, A.; Patel, K. (2023). *Model Context Protocols for Structured Generation*.

Tabela Comparativa das Estratégias

Estratégia	Clareza	Precisão	Profundidade	Aplicabilidade
Ajuste Fino	8	9	8	7
In-Context	7	7	7	8
Zero-shot	6	6	5	6
One-shot	7	6	6	7
Few-shot	7	7	7	8
Chain-of-Thought	8	8	9	7
Self-Consistency	7	8	7	6
Decomposição	8	7	8	7

Least-to-Most	7	6	7	6
Decomposed	7	7	8	7
Successive	6	6	7	6
Tool-Augmented	6	9	7	5
Memory/Context	7	7	6	6
MCP	6	6	7	6

Gráfico de Precisão



Conclusão Geral

As estratégias **Ajuste Fino** e **Tool-Augmented Reasoning** lideram em **Precisão** (nota 9), sendo ideais quando a exatidão de detalhes é essencial. **Chain-of-Thought** oferece alta **Profundidade** (9), recomendada para tarefas que exigem raciocínio explícito. Métodos como **Zero-shot** e **One-shot** são mais rápidos para prototipação, mas menos precisos. Abordagens de **Decomposição** (problemas e prompts) atingem bom equilíbrio entre clareza, profundidade e aplicabilidade, facilitando manutenção modular.

Recomendação de uso híbrido:

- Decomposição de Problemas** para estruturar grandes tarefas;
- Chain-of-Thought** para aprofundamento nos passos críticos;

3. **Tool-Augmented Reasoning** quando for necessário garantir métricas precisas (por ex., calorias ou nutrição).