

# Large Language Models: Da Teoria à Prática

Fundamentos, Inferência, Fine-tuning e Aplicações

Bruno Leonardo Santos Menezes  
Fabio André Machado Porto  
Daniel Rocha de Senna

LNCC – Laboratório Nacional de Computação Científica

2026



### **Créditos**

Este texto foi desenvolvido a partir do minicurso  
“*Large Language Models: Da Teoria à Prática*”,  
apresentado na Jornada de Ciência de Dados do LNCC (fevereiro de 2026).

© 2026 — Bruno L. S. Menezes, Fabio A. M. Porto, Daniel R. de Senna



# Sumário

<b>Prefácio</b>	<b>xiii</b>
<b>1 Introdução aos Large Language Models</b>	<b>1</b>
1.1 Panorama 2023–2025: contexto, escala e mudança de regime . . . . .	1
1.2 As cinco gerações da modelagem de linguagem . . . . .	1
1.2.1 Primeira geração: modelos estatísticos . . . . .	1
1.2.2 Segunda geração: modelos neurais . . . . .	1
1.2.3 Terceira geração: pré-treinamento e adaptação . . . . .	2
1.2.4 Quarta geração: LLMs e in-context learning . . . . .	2
1.2.5 Quinta geração: modelos multimodais . . . . .	2
1.3 Consolidação e ecossistema open-source (2023–2025) . . . . .	3
1.4 Perspectiva: de modelos isolados a sistemas orquestrados . . . . .	3
Pontos para lembrar . . . . .	4
O que vem a seguir . . . . .	5
<b>2 Núcleo do Transformer</b>	<b>7</b>
2.1 Visão geral: do texto ao espaço vetorial . . . . .	7
2.2 Tokenização: do texto aos tensores . . . . .	7
2.2.1 Implementação no BERT . . . . .	7
2.2.2 Inspeção do objeto retornado . . . . .	8
2.2.3 Tokens observados . . . . .	8
2.2.4 O tensor <code>input_ids</code> . . . . .	9
2.2.5 A máscara de atenção <code>attention_mask</code> . . . . .	9
2.2.6 Entrada formal do modelo . . . . .	10
2.3 De índices a vetores: embeddings . . . . .	10
2.3.1 Matriz de embeddings . . . . .	11
2.3.2 Do ID ao vetor . . . . .	11
2.3.3 O que embeddings ainda não são . . . . .	11
2.3.4 Intuição: o que ainda não é contexto (e quando passa a ser) . . . . .	11
2.4 Embeddings de entrada no BERT: do código ao tensor . . . . .	12
2.4.1 Comprimento da sequência . . . . .	12
2.4.2 IDs posicionais . . . . .	12
2.4.3 IDs de segmento ( <code>token_type_ids</code> ) . . . . .	12
2.4.4 Três componentes: token, posição e segmento . . . . .	13
2.4.5 Composição final da entrada . . . . .	14

2.4.6	Interpretação dos tensores impressos . . . . .	14
2.4.7	Leitura guiada da projeção PCA (tokens 0 a 10) . . . . .	14
2.5	Self-attention observada no BERT: do código ao fenômeno . . . . .	15
2.5.1	O que o modelo retorna quando pedimos as atenções . . . . .	15
2.5.2	Leitura conceitual do mapa de atenção . . . . .	15
2.5.3	Como interpretar o heatmap: guia de leitura . . . . .	16
2.5.4	Exemplo concreto: para quem o token <code>teddy</code> olha . . . . .	17
2.5.5	O que aprendemos antes das equações . . . . .	17
2.6	Self-Attention: definição formal . . . . .	18
2.6.1	Ponto de partida: o tensor de entrada . . . . .	18
2.6.2	Três projeções para três papéis: <b>Q</b> , <b>K</b> e <b>V</b> . . . . .	19
2.6.3	Compatibilidade entre tokens: produto interno $\mathbf{QK}^\top$ . . . . .	19
2.6.4	Normalização por $\sqrt{d_k}$ . . . . .	19
2.6.5	De escores a probabilidades: softmax . . . . .	19
2.6.6	Mistura de informação: aplicando a atenção aos valores . . . . .	20
2.7	Conexões Residuais e Multi-Head Attention . . . . .	20
2.7.1	Conexões residuais: somar em vez de reescrever . . . . .	20
2.7.2	A equação da atenção, agora no lugar certo . . . . .	21
2.7.3	Multi-Head Attention: projeções paralelas e recombinação . . . . .	21
2.8	Mini-corpus didático: cálculo completo (conta a conta) . . . . .	22
2.8.1	Vocabulário e mapeamento (token $\rightarrow$ índice) . . . . .	22
2.8.2	Embeddings: por que existem e por que aqui são bidimensionais . . . . .	22
2.8.3	Dimensão latente e matriz de embeddings . . . . .	23
2.8.4	Entrada “teddy reads”: de tokens a tensor . . . . .	23
2.8.5	Definindo $Q$ , $K$ e $V$ : escolha didática . . . . .	24
2.8.6	Escore: calculando $QK^\top$ elemento a elemento . . . . .	24
2.8.7	Normalização: dividindo por $\sqrt{d_k}$ . . . . .	24
2.8.8	Softmax linha a linha: de escores a probabilidades . . . . .	25
2.8.9	Mistura final: $Z = AV$ . . . . .	25
2.9	Feed-Forward Network — não linearidade por token . . . . .	26
2.9.1	Definição matemática . . . . .	26
2.9.2	Conexão direta com o código do BERT . . . . .	26
2.10	Normalização — por que os vetores não podem crescer sem controle . . . . .	26
2.10.1	O que pode dar errado sem normalização . . . . .	27
2.10.2	Layer Normalization: aplicada dentro do token . . . . .	27
2.10.3	Parâmetros aprendidos: $\gamma$ e $\beta$ . . . . .	28
2.10.4	Equação compacta . . . . .	28
2.10.5	Pós-norm no BERT e pré-norm em modelos modernos . . . . .	28
2.10.6	O que a normalização não faz . . . . .	28
2.10.7	Mensagem-chave . . . . .	28
	Pontos para lembrar . . . . .	29
	O que vem a seguir . . . . .	29

<b>3</b>	<b>Variações Modernas do Transformer</b>	<b>31</b>
3.1	O que realmente mudou (e o que permaneceu invariável)	31
3.2	Mapa mental: três refinamentos, um mesmo princípio	31
3.3	Um experimento-base para visualizar números	32
3.3.1	Mini-corpus e embeddings explícitos	32
3.3.2	Leitura do código: o que está sendo controlado	33
3.4	Rotary Embeddings (RoPE)	34
3.4.1	Por que embeddings posicionais aditivos são limitantes	34
3.4.2	Ideia central: posição como operador geométrico	34
3.4.3	O bloco fundamental: rotação 2D	34
3.4.4	RoPE em dimensão real: rotações em pares	35
3.4.5	Implementação: RoPE aplicado a <b>Q</b> e <b>K</b>	35
3.5	O efeito de RoPE nos logits	35
3.5.1	Sem RoPE: produtos escalares normalizados	36
3.5.2	Com RoPE: compatibilidade passa a incorporar posição	36
3.5.3	Por que RoPE produz efeito relativo	37
3.6	Como interpretar o heatmap dos logits	37
3.6.1	Leitura estrutural (linhas, colunas e diagonal)	37
3.7	Attention Scaling e Temperatura	38
3.7.1	O problema numérico da atenção em altas dimensões	38
3.7.2	A correção clássica: normalização por $\sqrt{d_k}$	38
3.7.3	Escala, <i>softmax</i> e entropia da atenção	39
3.7.4	Temperatura como generalização contínua do regime	39
3.8	Mixture of Experts (MoE)	40
3.8.1	Motivação: capacidade sem custo linear por token	40
3.8.2	Definição: especialistas e roteamento top- <i>k</i>	40
3.8.3	O roteador como operador (e a analogia com atenção)	40
3.8.4	Carga (load) e a patologia do desbalanceamento	41
3.8.5	Ganho de capacidade: total versus ativo	41
3.8.6	Balanceamento em modelos reais	41
3.9	Síntese do capítulo: geometria, escala e capacidade	42
	Pontos para lembrar	43
	O que vem a seguir	44
<b>4</b>	<b>Inferência em Modelos Causais</b>	<b>45</b>
4.1	Abertura: o que será controlado na inferência	45
4.2	Transição conceitual: de BERT para GPT	45
4.2.1	Diferença estrutural essencial: atenção bidirecional vs causal	46
4.2.2	Por que começar com GPT-2	46
4.2.3	O que mudou nos GPTs modernos (sem mudar o princípio)	46
4.3	Comparação visual: Transformer genérico, BERT e GPT-2	47
4.3.1	Como ler os diagramas: convenções e legenda	47
4.3.2	Transformer genérico: estrutura base comum	47
4.3.3	BERT: encoder bidirecional	47
4.3.4	GPT-2: decoder causal	48

4.4	Do diagrama ao PyTorch: conectando visual e implementação . . . . .	48
4.5	GPT-2: decoder causal na prática . . . . .	48
4.5.1	Entrada (cinza): <code>input_ids</code> . . . . .	48
4.5.2	Embeddings (azul): <code>wte</code> e <code>wpe</code> . . . . .	48
4.5.3	Profundidade (cluster): <code>Decoder x12</code> como <code>ModuleList</code> . . . . .	48
4.5.4	Atenção causal (verde): <code>GPT2Attention</code> . . . . .	49
4.5.5	MLP/FFN (verde): <code>GPT2MLP</code> . . . . .	49
4.5.6	Saída (amarelo): <code>lm_head</code> e predição do próximo token . . . . .	49
4.6	BERT: encoder bidirecional na prática . . . . .	49
4.7	Por que o diagrama funciona . . . . .	50
4.8	O problema matemático da geração . . . . .	50
4.9	Temperatura . . . . .	51
4.9.1	Definição formal . . . . .	51
4.9.2	Prova: temperatura baixa concentra; temperatura alta espalha . . . . .	51
4.9.3	O que observar nos outputs . . . . .	51
4.10	Top- $p$ (Nucleus Sampling) . . . . .	51
4.10.1	Definição formal . . . . .	51
4.11	KV cache . . . . .	52
4.11.1	O que é recalculado sem cache . . . . .	52
4.11.2	Ideia do cache . . . . .	52
4.11.3	Leitura do experimento . . . . .	52
4.12	Código: geração controlada e interpretação . . . . .	52
4.12.1	Carregando o GPT-2 . . . . .	53
4.12.2	Função de geração com temperatura, top- $p$ e cache . . . . .	53
4.12.3	Teste base . . . . .	53
4.12.4	Experimento: temperatura . . . . .	54
4.12.5	Experimento: top- $p$ . . . . .	54
4.12.6	Experimento: KV cache (tempo) . . . . .	54
4.13	Demonstração guiada: prompting e variabilidade . . . . .	55
4.13.1	Direto vs. passo a passo . . . . .	55
4.14	Regras de bolso antes da prática . . . . .	55
4.15	Atividade prática: inferência interativa em modelos de linguagem . . . . .	56
4.15.1	Contexto da atividade . . . . .	56
4.15.2	Objetivos de aprendizagem . . . . .	56
4.15.3	Instruções gerais . . . . .	57
4.15.4	Parte A: efeito da temperatura . . . . .	57
4.15.5	Parte B: efeito do top- $p$ (nucleus sampling) . . . . .	57
4.15.6	Parte C: chain-of-thought — estilo vs. correção . . . . .	58
4.15.7	Parte D: escolha de parâmetros por objetivo . . . . .	58
4.15.8	Critérios de avaliação . . . . .	58
4.15.9	Leitura guiada da arquitetura e interpretação das saídas experimentais . . . . .	59
4.15.10	Contexto desta seção . . . . .	59
4.15.11	Confirmação do ambiente e do modelo carregado . . . . .	59
4.15.12	Leitura estrutural do GPT-2 impresso . . . . .	59
	Pontos para lembrar . . . . .	60



O que vem a seguir . . . . .	60
<b>5 Prompt Engineering e In-Context Learning</b>	<b>61</b>
5.1 Contexto da atividade . . . . .	61
5.2 Objetivos de aprendizagem . . . . .	62
5.3 Modelo utilizado . . . . .	62
5.3.1 Carregamento do modelo . . . . .	62
5.3.2 Saída observada . . . . .	62
5.4 Funções de geração . . . . .	63
5.4.1 Geração determinística (reprodutível) . . . . .	63
5.4.2 Geração com amostragem (para variabilidade) . . . . .	63
5.5 Few-shot e In-Context Learning . . . . .	63
5.5.1 Execução zero-shot . . . . .	63
5.5.2 Saída observada . . . . .	64
5.5.3 Execução few-shot . . . . .	64
5.5.4 Saída observada . . . . .	64
5.6 O papel do template . . . . .	65
5.6.1 Template pouco restritivo . . . . .	65
5.6.2 Saída observada . . . . .	65
5.6.3 Template restritivo . . . . .	65
5.6.4 Saída observada . . . . .	65
5.7 Self-consistency . . . . .	66
5.7.1 Execução (amostragem habilitada) . . . . .	66
5.7.2 Saída observada . . . . .	66
5.8 Limites de contexto . . . . .	66
5.8.1 Carregamento mínimo do GPT-2 para inspeção de janela . . . . .	67
5.8.2 Saída observada . . . . .	67
5.9 Na prática: conduzindo a atividade e analisando resultados . . . . .	67
Síntese . . . . .	68
Pontos para lembrar . . . . .	68
O que vem a seguir . . . . .	69
<b>6 Pré-treinamento e Otimização Computacional de LLMs</b>	<b>71</b>
6.1 Contexto e objetivo da atividade . . . . .	71
6.2 Configuração do ambiente computacional . . . . .	71
6.3 Pré-treinamento: formulação estatística . . . . .	72
6.4 Modelo base em precisão total (FP32) . . . . .	73
6.5 Quantização INT4 e eficiência . . . . .	74
6.6 Comparação de memória . . . . .	74
6.7 Conexão direta com QLoRA . . . . .	75
Síntese . . . . .	75
Pontos para lembrar . . . . .	75
O que vem a seguir . . . . .	76

<b>7</b>	<b>Treinamento e Otimização de LLMs</b>	<b>77</b>
7.1	Abertura: o problema real de engenharia . . . . .	77
7.2	Treinamento, alinhamento e avaliação como uma decisão única . . . . .	77
7.2.1	Treinamento: onde o gradiente passa . . . . .	77
7.2.2	Alinhamento: qual comportamento é reforçado . . . . .	78
7.2.3	Avaliação: o que você decide medir . . . . .	78
7.2.4	Integração: a decisão real . . . . .	78
7.3	O que significa treinar um LLM: formulação matemática . . . . .	79
7.3.1	Modelo causal e função objetivo . . . . .	79
7.3.2	SFT clássico: otimização no espaço completo . . . . .	79
7.3.3	LoRA: restrição do espaço de atualização . . . . .	80
7.3.4	QLoRA: separação entre armazenamento e otimização . . . . .	80
7.3.5	Comparação unificada . . . . .	80
7.4	Prova numérica: “treinar é escolher espaço, direção e precisão” . . . . .	81
7.4.1	Setup: vocabulário, entrada e pesos . . . . .	81
7.4.2	Forward: logits $z = Wx$ . . . . .	81
7.4.3	Softmax: probabilidades $p = \text{softmax}(z)$ . . . . .	81
7.4.4	Perda: cross-entropy para $y = 1$ . . . . .	82
7.4.5	Gradiente: $\nabla_W \mathcal{L}$ (conta a conta) . . . . .	82
7.4.6	Um passo de treino SFT: $W \leftarrow W - \eta \nabla_W \mathcal{L}$ . . . . .	82
7.4.7	Verificação numérica: a perda cai? . . . . .	83
7.4.8	Onde entra a “escolha do espaço”? (LoRA) . . . . .	83
7.4.9	Onde entra a “escolha da precisão”? (QLoRA) . . . . .	84
7.5	Visão geral do pipeline moderno . . . . .	84
7.6	Preparação do ambiente . . . . .	84
7.7	Modelo base e referência comportamental . . . . .	85
7.8	Construção do dataset supervisionado . . . . .	85
7.9	Estágio 1: SFT clássico e seus limites . . . . .	86
7.10	Estágio 2: LoRA como adaptação controlada . . . . .	87
7.11	Estágio 3: QLoRA com base quantizada . . . . .	87
	Síntese . . . . .	88
7.12	Na prática: como conduzir o pipeline com transparência . . . . .	88
	Pontos para lembrar . . . . .	89
	O que vem a seguir . . . . .	89
<b>8</b>	<b>Alinhamento de Preferências e Avaliação Moderna em LLMs</b>	<b>91</b>
8.1	Contexto e motivação . . . . .	91
8.2	Alinhamento por preferências diretas: DPO . . . . .	91
8.2.1	Ideia central . . . . .	91
8.2.2	Configuração do experimento . . . . .	92
8.2.3	Comportamento antes do DPO . . . . .	92
8.2.4	O que o treinamento DPO otimiza . . . . .	92
8.2.5	Comportamento após o DPO . . . . .	92
8.3	RLHF e GRPO em miniatura . . . . .	93
8.3.1	Intuição do método . . . . .	93

8.3.2	Experimento didático . . . . .	93
8.3.3	Interpretação das saídas . . . . .	93
8.4	Avaliação moderna e LLM-as-a-judge . . . . .	94
8.4.1	Por que a avaliação ficou difícil . . . . .	94
8.4.2	Falha 1: fluência versus factualidade . . . . .	94
8.4.3	Falha 2: contaminação (leakage) . . . . .	94
8.4.4	Limitações estruturais . . . . .	94
8.5	Leitura do hands-on: código e outputs observados . . . . .	94
8.5.1	Inicialização do ambiente e carregamento dos modelos . . . . .	95
8.5.2	Policy antes do alinhamento (pré-DPO) . . . . .	95
8.5.3	Treinamento com DPO: o que o log indica . . . . .	95
8.5.4	Policy após o DPO: o que muda (e o que não muda) . . . . .	95
8.5.5	Mini-RL: leitura do loop REINFORCE/GRPO . . . . .	96
8.5.6	LLM-as-a-judge: leitura dos casos do hands-on . . . . .	96
8.6	Na prática: checklist mínimo para experimentos de alinhamento e avaliação . . . . .	96
	Pontos para lembrar . . . . .	97
	O que vem a seguir . . . . .	97
<b>9</b>	<b>Tendências e desafios em LLMs abertos</b>	<b>99</b>
9.1	Interoperabilidade . . . . .	99
9.1.1	Padrões de interface . . . . .	99
9.1.2	Interoperabilidade de componentes . . . . .	99
9.1.3	Interoperabilidade computacional . . . . .	99
9.2	Governança . . . . .	100
9.2.1	Documentação e transparência . . . . .	100
9.2.2	Avaliação como instrumento de governança . . . . .	100
9.2.3	Alinhamento como política técnica . . . . .	100
9.3	Impacto científico . . . . .	100
9.4	Um desafio aberto . . . . .	101
	Pontos para lembrar . . . . .	101
	O que vem a seguir . . . . .	101
	<b>Referências Bibliográficas</b>	<b>103</b>