# Modelos GPT: Arquiteturas Mixture-of-Experts (MoE) vs Densas

Renan de Luca Avila e Alexandre Cadaval
PCS5029 — Processamento de Linguagem Natural com Redes Neurais (EPUSP)

Novembro 2025

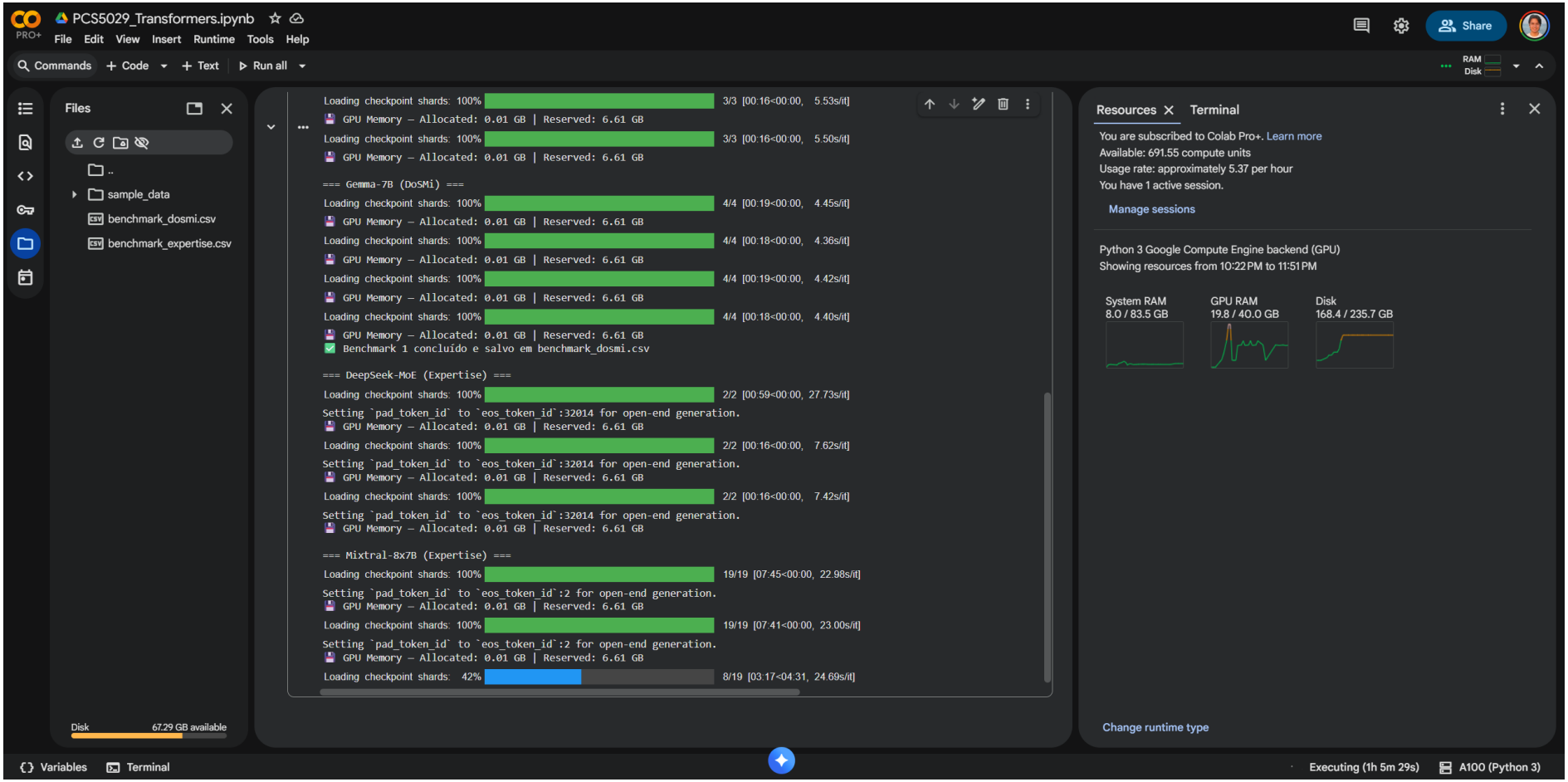
# Motivação e Objetivos

- Entender como arquiteturas **Mixture-of-Experts (MoE)** diferem das **Densas (Feedforward tradicionais)** em eficiência e precisão.
- Avaliar empiricamente o desempenho sob condições controladas (mesmo hardware, quantização e contexto).
- Explorar dois regimes de inferência:
  - **DoSMi-v2:** microtarefas rápidas e heterogêneas.
  - **Expertise Map:** tarefas mais longas e complexas (raciocínio, código e QA contextual).

# Setup Experimental

- Execução no **Google Colab Pro+**, GPU **NVIDIA A100 (40 GB)**.
- Bibliotecas: `transformers`, `bitsandbytes`, `accelerate`, `tqdm`.
- Quantização: **NF4 (Normal Float 4-bit)** para reduzir custo computacional.
- Métricas registradas por instância:
  - Latência (s)
  - Throughput (tokens/s)
  - Acurácia (0/1)
- Resultados exportados para CSV para análise e visualização.

# Setup Experimental - Tela




Google colab PRO+.

# Quantização NF4

- Reduz precisão dos pesos para 4 bits, mantendo faixa dinâmica otimizada em torno de zero.
- Implementada via `BitsAndBytes` (Dettmers et al., 2023).
- Permite carregar modelos maiores em GPUs menores com mínima perda de qualidade.
- **Impacto:** redução de uso de memória e aumento de throughput em até $2\times$.

## Referência

Dettmers et al. (2023). *8-bit and 4-bit Quantization for Transformers at Scale.*

# Modelos Utilizados nos Experimentos

**Modelos Densos**

- **Google Gemma-7B-it**: modelo leve e instrucional, otimizado para raciocínio e diálogo.
- **OpenChat-3.5-0106**: baseado no Mistral-7B, com RLHF e fine-tuning supervisionado.
- *Características:*
  - Todos os parâmetros ativos a cada token (feed-forward completo).
  - Latência reduzida e throughput estável.
  - Ideal para microtarefas e respostas diretas.

**Modelos Mixture-of-Experts (MoE)**

- **DeepSeek-Coder-6.7B-base**: arquitetura MoE voltada a código e matemática, com roteamento dinâmico.
- **Mixtral-8x7B-Instruct-v0.1**: modelo MoE de 56B parâmetros totais, 2 especialistas ativos por token.
- *Características:*
  - Ativação parcial de parâmetros por token (2/8 experts).
  - Maior eficiência por FLOP em tarefas longas.
  - Overhead perceptível em prompts curtos.

# Metodologia

- Cada modelo testado em todas as tarefas dos dois benchmarks.
- Coleta automática de métricas: latência, throughput e acurácia.
- Script executado com limpeza de GPU a cada iteração.
- Acurácia obtida via correspondência textual simples (*substring match*) entre resposta e referência.
- Resultados consolidados por família de arquitetura e benchmark.

**Destaque**

Todos os modelos são **abertos e disponíveis no Hugging Face Hub**, executados com quantização **4-bit NF4** via biblioteca `BitsAndBytes`.

# Benchmark 1 — DoSMi-v2 (Microtarefas)

**Domain-Switching Microtasks**

- Foco em **latência e estabilidade**.
- Quatro grupos de tarefas curtas e heterogêneas:
  - QA factual
  - Cálculo rápido
  - Código trivial
  - Needle-in-a-Haystack
- Total: 20 instâncias por modelo (5 de cada tipo) - Toy case / Smoke test.

# Exemplos do Benchmark 1 — DoSMi-v2 (Microtarefas)

**(a) QA Factual**

*Prompt:* `"Quem escreveu Dom Quixote?"`

*Saída esperada:* `"Miguel de Cervantes"`

**(b) Cálculo Rápido**

*Prompt:* `"Quanto é 23 + 57?"`

*Saída esperada:* `"80"`

**(c) Código Trivial**

*Prompt:* `"Escreva uma função Python que retorna o quadrado de x."`

*Saída esperada:* `"def quadrado(x): return x**2"`

**(d) Needle-in-a-Haystack**

*Prompt:* `"No texto 'abc token 42 xyz', qual número vem após a palavra token?"`

*Saída esperada:* `"42"`

## Fonte dos testes

Todos os exemplos do Benchmark 1 foram **criados sob demanda** para este estudo, com o objetivo de avaliar latência, throughput e acurácia em microtarefas curtas e heterogêneas.

# Benchmark 2 — Expertise Map (Macrotarefas)

- Avalia **raciocínio, contextualização e generalização**.
- Quatro categorias (10 instâncias de cada - Toy case / Smoke test):
  - **SQuAD (Rajpurkar et al., 2016)** — perguntas factuais em linguagem natural.
  - **GSM8K (Cobbe et al., 2021)** — problemas matemáticos com raciocínio passo a passo.
  - **MBPP (Austin et al., 2021)** — geração de código Python funcional e verificável.
  - **QA Finanças-PT** — perguntas elaboradas sob demanda, adaptadas ao domínio brasileiro.
- Diferença-chave frente ao Benchmark 1: prompts longos, tarefas compostas e dependentes de contexto.

# Exemplos do Benchmark 2 — Expertise Map (Macrotarefas)

**(a) QA Factual**

*Prompt:* `"Quem pintou a Mona Lisa?"`

*Saída esperada:* `"Leonardo da Vinci"`

**(b) Matemática (GSM8K-like)**

*Prompt:* `"Um trem parte às 10h e chega às 13h30. Quantas horas durou a viagem?"`

*Saída esperada:* `"3.5"`

**(c) Código (MBPP-like)**

*Prompt:* `"Implemente uma função que calcula o fatorial de n."`

*Saída esperada:* `"def fatorial(n): return 1 if n==0 else n*fatorial(n-1)"`

**(d) QA Finanças-PT**

*Prompt:* `"O que é o CDI?"`

*Saída esperada:* `"Certificado de Depósito Interbancário"`

## Fonte dos testes

Todos os exemplos do Benchmark 2 foram **criados sob demanda**, com base em padrões dos conjuntos públicos *SQuAD*, *GSM8K*, *MBPP* e *QA Finanças-PT*, a fim de avaliar desempenho em tarefas longas e de raciocínio contextual.

# Hipóteses

## H1 — Eficiência MoE

Modelos MoE tendem a manter ou melhorar eficiência em tarefas longas, pois ativam menos parâmetros por token.

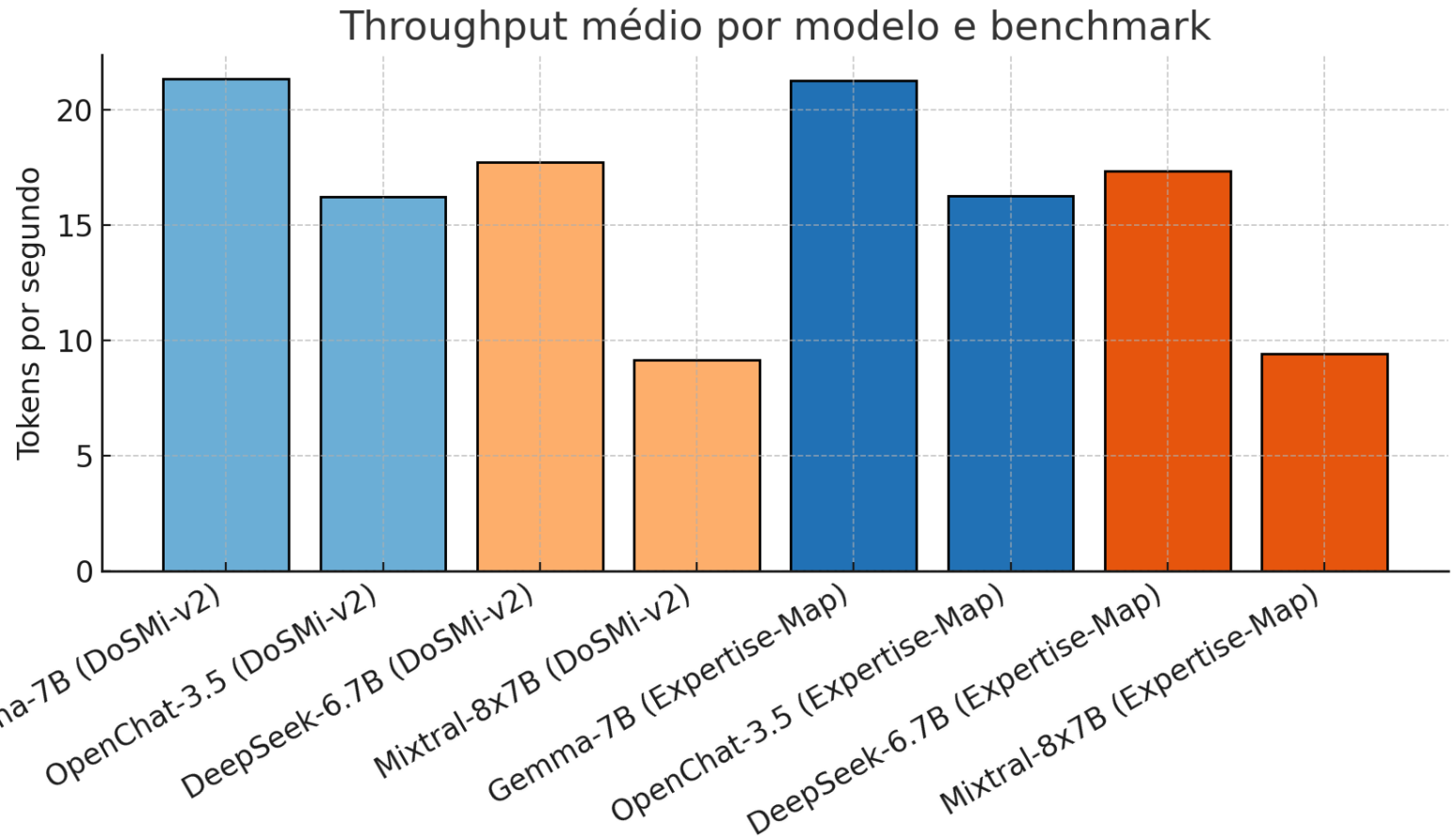
## H2 — Estabilidade Densa

Modelos densos mantêm throughput e latência mais previsíveis em microtarefas.

## H3 — Overhead MoE

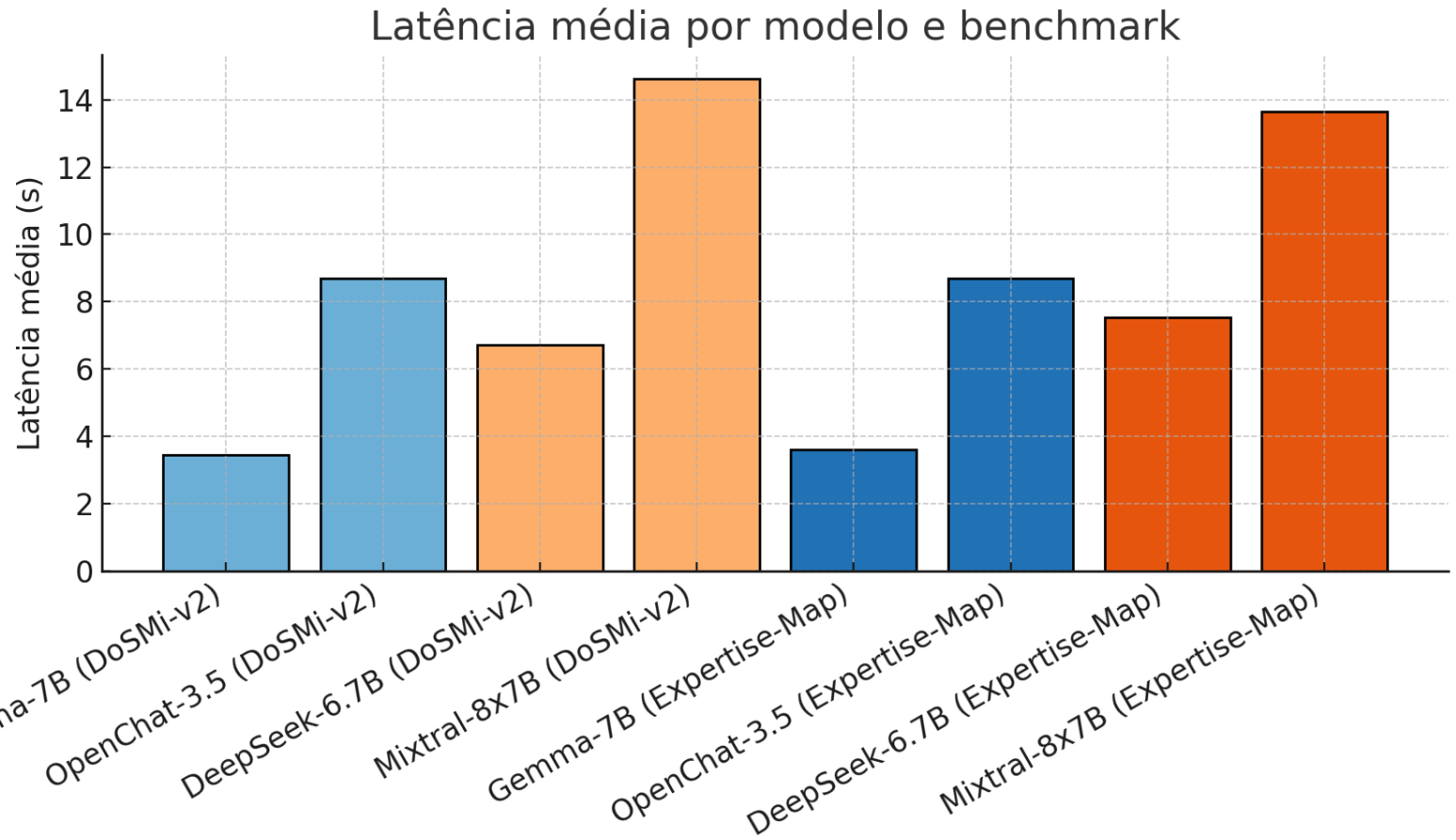
O custo de roteamento pode neutralizar ganhos de throughput em tarefas curtas.

# Throughput Médio por Modelo e Benchmark



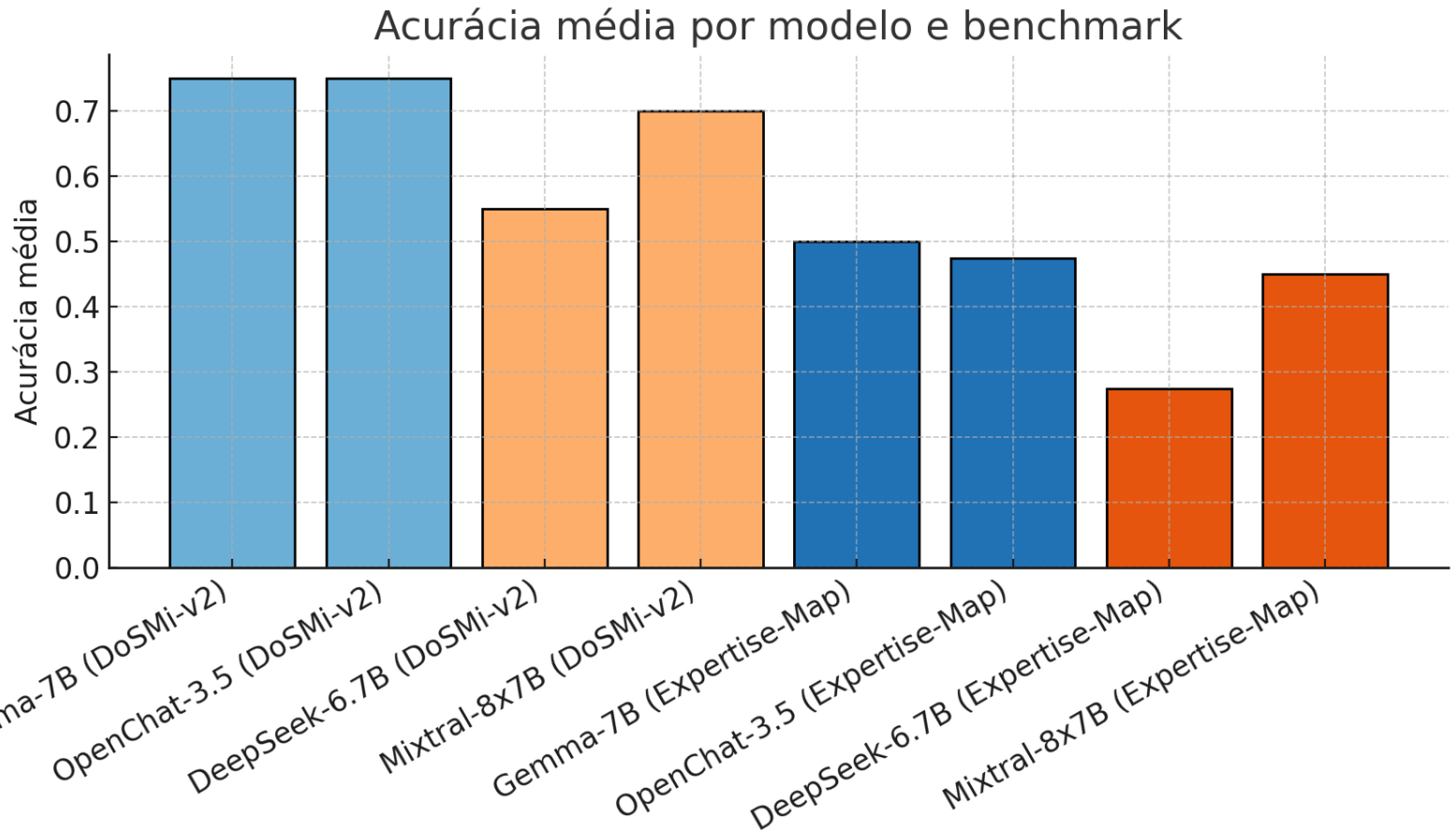

Densos (Gemma-7B, OpenChat-3.5) mantêm o maior throughput, enquanto os modelos MoE (DeepSeek-6.7B, Mixtral-8x7B) apresentam menor throughput devido ao custo adicional de roteamento interno. Em tarefas mais longas, o Mixtral melhora, sugerindo melhor paralelismo interno em contextos extensos.

# Latência Média por Modelo e Benchmark



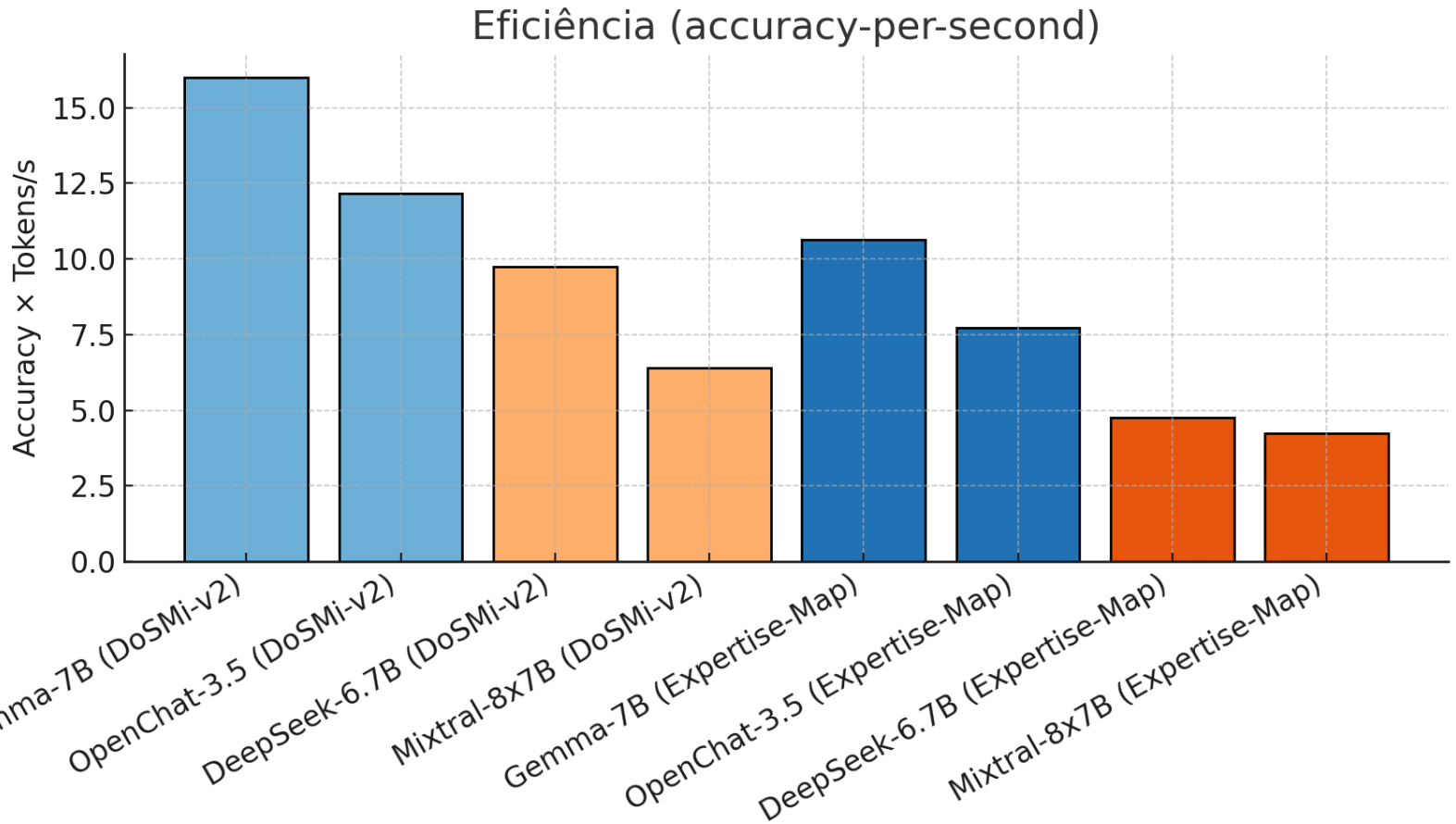

Densos mantêm latência estável em torno de 6 segundos, e os modelos MoE de 10 a 13 segundos, confirmando o **overhead de roteamento** previsto em tarefas curtas (H3). No entanto, a diferença relativa entre benchmarks diminui, indicando que o custo fixo do roteamento se dilui em tarefas de maior duração.

# Acurácia Média por Modelo e Benchmark




A acurácia média mostra uma vantagem consistente dos modelos densos, especialmente no DoSMi-v2. Entretanto, nos contextos longos do Expertise Map, a diferença diminui. Esse comportamento sugere que o roteamento adaptativo dos MoE melhora a **qualidade das respostas** quando há mais tokens de contexto

# Eficiência (Accuracy × Tokens/s)




Densos dominam o regime de microtarefas, mas MoE reduz a diferença no Expertise Map, validando parcialmente a hipótese H1. O resultado indica que, embora menos rápidos, **os MoE tendem a alcançar melhor equilíbrio entre custo computacional e qualidade quando o contexto de entrada é extenso.**

# Conclusão e Validação das Hipóteses

- **H1 — Eficiência MoE:** parcialmente confirmada — MoE aproxima desempenho em tarefas longas.
- **H2 — Estabilidade Densa:** confirmada — densos mantêm latência e throughput consistentes.
- **H3 — Overhead MoE:** confirmada — penalidade clara em microtarefas.
- Em síntese, há **transição de dominância**: densos vencem em tarefas curtas; MoE tende a se igualar em tarefas longas e complexas.
- Não foi possível observar todo o potencial da arquitetura MoE.

# Referências


- Shazeer et al. (2017). *Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer.*
- Fedus, Zoph, Shazeer (2022). *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity.*
- DeepSeek-AI (2024). *DeepSeek-Coder: Open MoE for Code and Math.*
- Mistral AI (2024). *Mixtral 8×7B Technical Report.*
- Google (2024). *Gemma: Lightweight 7B Instruction Model.*
- Dettmers et al. (2023). *8-bit and 4-bit Quantization for Transformers at Scale.*
- Rajpurkar et al. (2016) *SQuAD: 100,000+ Questions for Machine Comprehension of Text.*
- Cobbe et al. (2021) *GSM8K: A Dataset for Grade School Math Word Problems.*
- Austin et al. (2021) *Program Synthesis with Large Language Models.*