Relatório de Viabilidade e Integração

Otimização de Extração Documental

Contexto e Objetivo

Otimização de API de Extração Documental

1. **Problema Atual:** Pipeline em duas inferências gera alta latência e custo.
2. **Desafio:** Unificar extração e formatação em uma única etapa, mantendo a acurácia e diminuindo latência e custo.
3. **Objetivo:** Avaliar modelos open-source para self-hosting com melhor throughput e custo-benefício.

Dupla Inferência (OCR + JSON) = Gargalo de Custo/Tempo → Arquitetura Unificada

O Problema

Gargalo de Custo e Latência

A Ineficiência da Dupla Inferência

Etapa	Função	Custo
1ª Inferência	Interpretação e Extração da Informação Bruta (OCR/ Reconhecimento)	Custo de Token 1 + Latência T1
2ª Inferência	Estruturação e Formatação do dado no schema JSON	Custo de Token 2 + Latência T2

Problema Chave: Latência e Custo são duplicados

"O presente estudo investiga alternativas para otimizar a arquitetura... Embora funcional, essa abordagem em múltiplas etapas resulta em maior latência, aumento dos custos operacionais e menor eficiência."

Limitações Funcionais

Layouts Complexos e Multimodalidade

1. **Parsing de Layouts Complexos:** Dificuldade com formulários densos e blocos de texto distribuídos.
2. **Tabelas Aninhadas e Gráficos:** Desempenho restrito na interpretação de elementos visuais complexos.
3. **Falta de Coesão:** Incapacidade de processar o contexto visual e textual em uma única etapa coerente.

"Também se observa um desempenho restrito na interpretação de elementos visuais, como gráficos e imagens, o que reforça a necessidade de soluções mais integradas e multimodais..."

Metodologia de Viabilidade

Foco em Throughput, Acurácia e Custo

Pilar 1
Acurácia e Benchmarking

OCRBench v2 Parsing & Extraction

Pilar 2
Critérios Arquiteturais

MoE e Mamba (Otimização e Throughput)

Pilar 3
Viabilidade Técnica/ Financeira

VRAM requerida e Throughput (Self-Hosting)

Inovação Arquitetural

A lógica da Eficiência

Com base nas limitações identificadas, foram exploradas arquiteturas recentes voltadas à eficiência: MoE e Mamba.

- **MoE (Mixture-of-Experts):** Aumenta a capacidade do modelo (ex: 30B) mantendo apenas uma fração dos parâmetros ativos por token. Garante: Qualidade de modelo grande com custo de inferência de um modelo de ~3.3B.

- **Mamba-Transformer:** Otimiza o processamento de longas sequências, substituindo a complexidade quadrática $O(n^2)$ por uma eficiência quase linear. Garante: Desempenho 6x maior em documentos extensos e alto throughput (Nemotron-Nano).

Shortlist

Modelos escolhidos

Modelo	Descrição
Qwen3-Omni-30B-A3B-Instruct	Modelo da série Qwen3, desenvolvido pela Alibaba Cloud. Trata-se de um modelo MoE (Mixture-of-Experts), que apresenta maior acurácia, mas com custo e latência mais elevados.
Nemotron-Nano-12B-V2-VL	Modelo da NVIDIA, baseado na arquitetura Nemotron VL. Destaca-se pela viabilidade de execução self-hosted, custo moderado e latência reduzida.
Llama-3.1-Nemotron-Nano-VL-8B-V1	Derivado da arquitetura Meta Llama 3.1 e aprimorado pela NVIDIA para capacidades de visão-linguagem (VL). Equilibra acurácia e eficiência, oferecendo um desempenho intermediário em relação aos demais modelos.

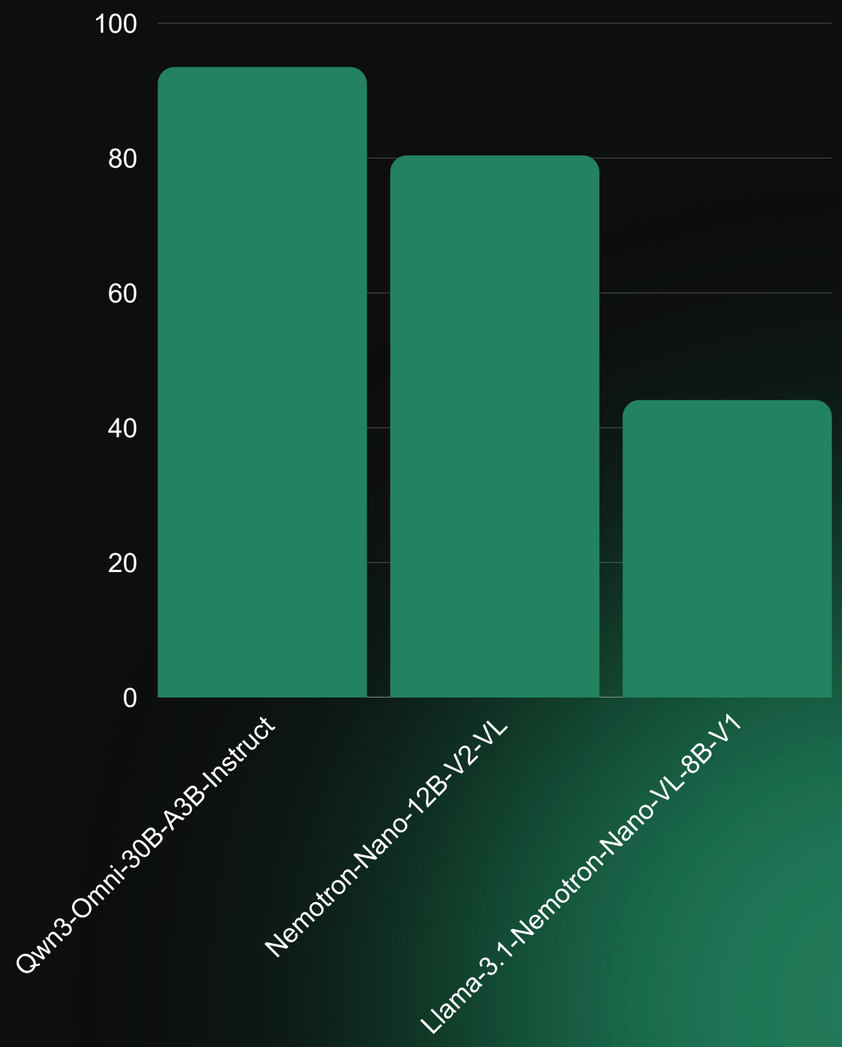
Fonte: OCRBench v2.

Analise 1

Parsing Score e Descarte do Llama

Modelo	Parsing Score
Qwen3-Omni-30B-A3B-Instruct	93,5% (Máximo)
Nemotron-Nano-12B-V2-VL	80,4% (Aceitável)
Llama-3.1-Nemotron-Nano-VL-8B-V1	44,1% (Descartado)

Fonte: OCRBench v2.



Analise 2

Acurácia vs Custo Operacional

Modelo	Parsing Score	Custo Operacional / Throughput	Requisito de VRAM
Qwen3-Omni-30B-A3B-Instruct	93,5% (Máximo)	Latência de um modelo de 3.3B	17 GB (INT4)
Nemotron-Nano-12B-V2-VL	80,4% (Aceitável)	Ganho Máximo (6x Throughput).	22 GB (BF16)

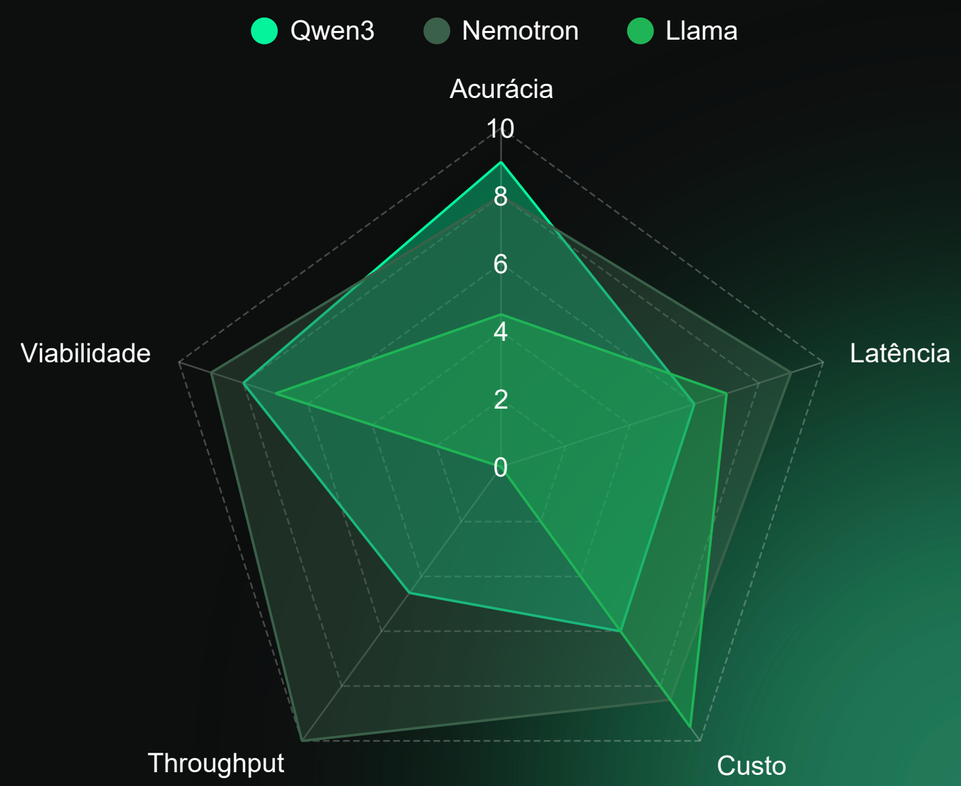
Fonte: OCRBench v2, Model Cards dos modelos no Hugging Face e documentação TensorRT-LLM, benchmarks oficiais da NVIDIA para Mamba-Transformer em execução BF16 com backend TensorRT-LLM.

Conclusão e Recomendação

O Throughput Vence a Qualidade Quantizada

- **Prioridade do Negócio:** Latência e Custo Operacional
- **Ganho de Throughput:** Throughput 6x maior (~83,3% menos tempo de espera)
- **Qualidade de dados:** Preferência pela estabilidade BF16 do Nemotron

Recomendação: Implantação do Nemotron-Nano-12B-V2-VL



**quanto maior, melhor desempenho relativo

Mitigando o Risco de Parsing

De 80,4% para 90%+

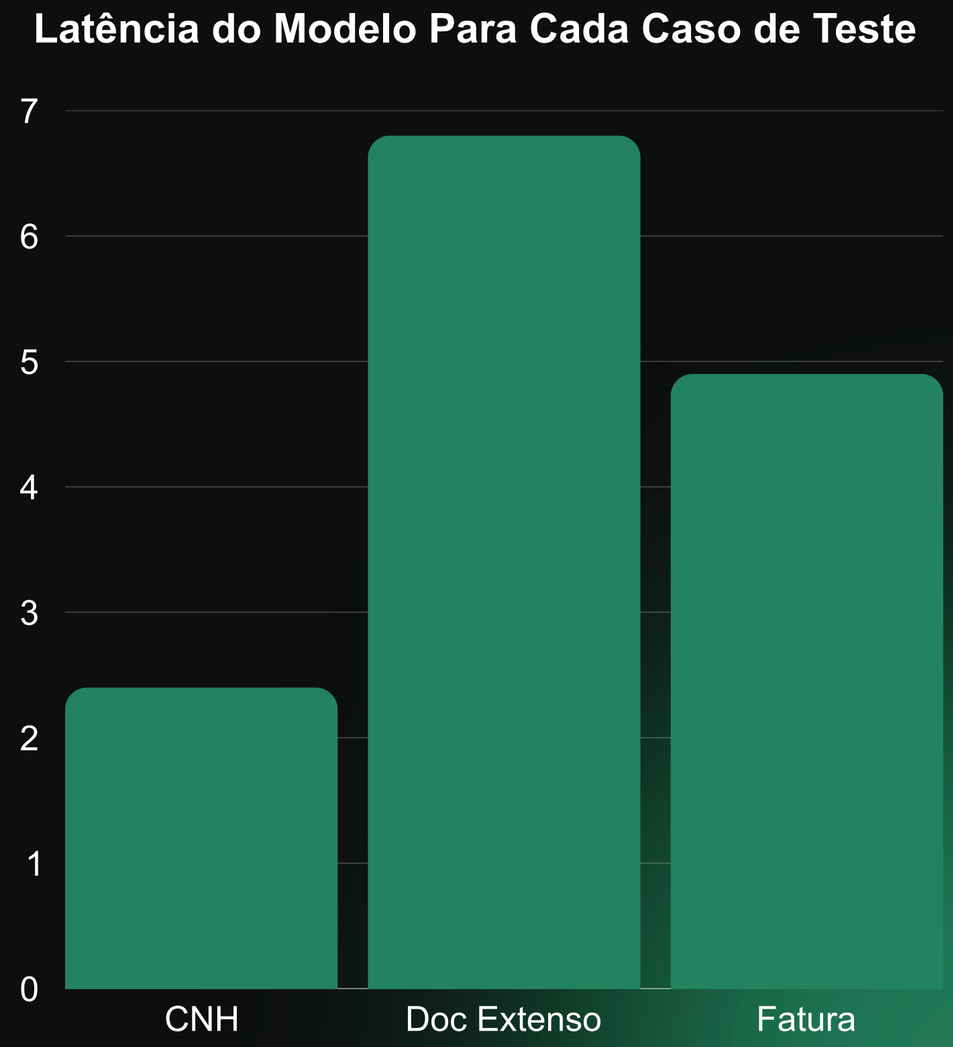
- **Validação da Saída Estruturada: (Pós Inferência):** Garantia de Confiabilidade
- **Engenharia de Prompt Avançada:** Uso do JSON Mode e coordenadas de bounding box (bbox_2d)

"Para mitigar o risco de erro em Parsing (80.4%), é incorporada uma etapa final de validação de saída estruturada, garantindo a confiabilidade dos dados extraídos sem comprometer o ganho de desempenho."

POC

Validação da Arquitetura Unificada

- **Input Unificado:** Texto (Prompt + JSON Schema) e image_url na mesma requisição
- **Output:** JSON puro na primeira tentativa, com fallback de reparo de String JSON
- **Testes:** CNH (formato complexo), PDF Extenso (Robustez) e Fatura de Energia (layout denso)



Fonte: POC

Infraestrutura Recomendada

Nemotron-Nano-12B-V2-VL

Requisitos de Hardware

- **GPU única (24 GB VRAM):** RTX 4090 / NVIDIA A10G
- **Uso total:** ~22 GB (BF16)
- **Vantagem:** dispensa cluster → **Reduz custo e complexidade**

Otimização e Latência

- **Runtimes:** vLLM ou TensorRT-LLM
- **Benefício:** throughput alto e latência 6× menor
- **Pipeline unificado:** apenas 1 inferência
- **Etapa extra:** validação estruturada → **Parsing > 90 %**

Alto throughput + menor latência + self-hosting viável = Infraestrutura otimizada e custo previsível

Obrigada!