Jev versus LLMs generativos na anotação de sentenças judiciais

Qualidade, tempo e custo de outputs estruturados em 120 sentenças do TJSP

Julio Trecenti · LabDados FGV Direito SP

2026-09-16

Índice

1 Resumo

- **Qualidade:** o Gemini 3.8 Flash teve a maior acurácia (**98,8%**). Jev (**96,6%**) e GPT-5.6 Luna (**96,8%**) ficaram empatados, cerca de 2 p.p. abaixo. A diferença para o Gemini é estatisticamente significativa.
- **Tempo:** o Jev foi cerca de **10× mais rápido**, com 0,32 s por sentença (mediana), contra 2,8 s no Gemini e 3,5 s no GPT.
- **Custo estimado:** US$ 0,25 por mil sentenças no Jev, contra US$ 1,64 no GPT-5.6 Luna e US$ 6,56 no Gemini.
- **Confiança:** aceitando só as respostas do Jev com confiança ≥ 0,8, a cobertura é de 91% das respostas e a acurácia sobe para 99,3%. O restante iria para revisão humana.

2 Introdução

O que é o Jev. O Jev é o primeiro modelo "System One" da TypeSafe. O nome vem da distinção de Kahneman entre pensamento rápido (sistema 1) e deliberado (sistema 2). Diferentemente de um LLM, o Jev **não gera texto**. Ele recebe um documento (o *state*) e um conjunto de perguntas tipadas, e devolve respostas restritas às opções definidas pelo usuário, sempre com probabilidades. Há três tipos de pergunta: **Choice** (escolher uma opção), **Score** (posicionar numa escala) e **Noul** (probabilidade de "sim"). Segundo a empresa, o modelo é treinado por *reinforcement learning for calibrated decisions*: em vez de otimizar respostas que agradam a humanos, otimiza decisões com probabilidades calibradas. A promessa é oferecer saídas estruturadas por construção (sem *parsing* nem respostas fora do esquema), muito rápidas e baratas, com uma medida de confiança utilizável.

Por que comparar. Em pesquisa empírica em Direito, uma tarefa central é **codificar decisões judiciais**: transformar milhares de sentenças em variáveis (resultado, tipo de réu, valor da condenação etc.). Hoje isso é feito com LLMs generativos e saída estruturada (JSON Schema), como no pacote dataframeit. Boa parte dessas variáveis é fechada, exatamente o tipo de pergunta a que o Jev responde. Se o Jev tiver qualidade comparável, ganhos de tempo e custo de uma ordem de grandeza mudam o que é viável coletar. A confiança calibrada também permitiria direcionar a revisão humana para os casos incertos. Este experimento testa essa hipótese contra dois LLMs "flash" de mercado: **Gemini 3.8 Flash** e **GPT-5.6 Luna**.

3 Metodologia

Dados. Coletamos com o juscraper as sentenças da Consulta de Julgados de 1º Grau (cjpg) do TJSP com o termo `"dano moral"`, julgadas entre janeiro e junho de 2026 (4 páginas por mês, 240 decisões). Mantivemos apenas procedimento comum cível e juizado especial cível, com um documento por processo e ao menos 1.500 caracteres, e sorteamos **120**

sentenças (semente 20260916). O tamanho mediano foi de 12.337 caracteres; nenhuma passou do limite de 60.000 caracteres aplicado a todos os modelos.

Esquema de anotação. Como o Jev só aceita perguntas fechadas, definimos 12 variáveis categóricas (Choice) ou binárias (Noul), num único arquivo (`schema.py`). Os três modelos receberam as mesmas instruções e opções, com uma requisição por sentença contendo as 12 perguntas. No Jev, cada variável vira uma pergunta `choice`/`noul`. No Gemini e no GPT, vira um modelo Pydantic (`Literal`/`bool`) passado como esquema de saída estruturada.

Tabela 1: Variáveis anotadas.

Variável	Tipo	Respostas
Resultado	choice	6 opções
Pedido de dano moral	noul	sim / não
Decisão sobre dano moral	choice	4 opções
Faixa do valor do dano moral	choice	5 opções
Tipo de réu	choice	11 opções
Justiça gratuita	choice	3 opções
Tutela de urgência	choice	3 opções
Aplica o CDC	noul	sim / não
Inversão do ônus da prova	noul	sim / não
Litigância de má-fé	noul	sim / não
Revelia	noul	sim / não
Honorários	choice	5 opções

Execução. Todos os modelos rodaram em 16/09/2026, um de cada vez, na mesma máquina, com 4 requisições simultâneas e as configurações padrão de cada provedor. As retentativas automáticas dos SDKs foram desligadas, para que nenhum atraso ficasse oculto. Também incluímos a variante `jev-preview`.

Padrão-ouro. Não havia anotação humana prévia. O gabarito foi construído em três etapas:

1. **Consenso de referências:** dois modelos fortes de famílias diferentes, **GPT-5.6 Sol** (raciocínio alto) e **Gemini 3.1 Pro**, anotaram todas as sentenças. Quando concordaram, esse valor virou o gabarito.
2. **Adjudicação das divergências** pelo **Claude Opus 5** (`claude-opus-5`), em três agentes independentes. Os agentes leram a sentença inteira **às cegas**: viam apenas a pergunta, as opções e o texto, sem a resposta de nenhum modelo.
3. **Auditoria:** 5% das respostas de consenso, sorteadas, passaram pela mesma adjudicação cega.

As referências concordaram em **97,6%** das 1440 respostas (120 × 12). Foram adjudicadas 34 divergências. Na auditoria, o Opus 5 confirmou 100% dos 70 consensos sorteados.

Métricas. Usamos acurácia por resposta (sentença × variável), com IC 95% por *bootstrap* reamostrando sentenças; *bootstrap* pareado para as diferenças entre modelos; e kappa de Cohen por variável. Para o tempo, medimos a latência de cada requisição e o tempo total. O custo foi estimado com os tokens informados pelas APIs e os preços de lista de setembro de 2026.

4 Resultados

4.1 Qualidade

Tabela 2: Acurácia contra o padrão-ouro. Diferença em relação ao Jev (latest) com IC 95% por bootstrap pareado.

Modelo	Acurácia	IC 95%	Dif. vs. Jev (p.p.)	Sentenças sem erro
Gemini 3.8 Flash	98,8%	98,1% a 99,3%	+2,2 (+1,3 a +3,0)	86%
GPT-5.6 Luna	96,8%	96,0% a 97,6%	+0,2 (-0,7 a +1,0)	66%
Jev (preview)	96,7%	95,9% a 97,4%	+0,1 (-0,1 a +0,3)	62%
Jev (latest)	96,6%	95,8% a 97,4%	—	61%

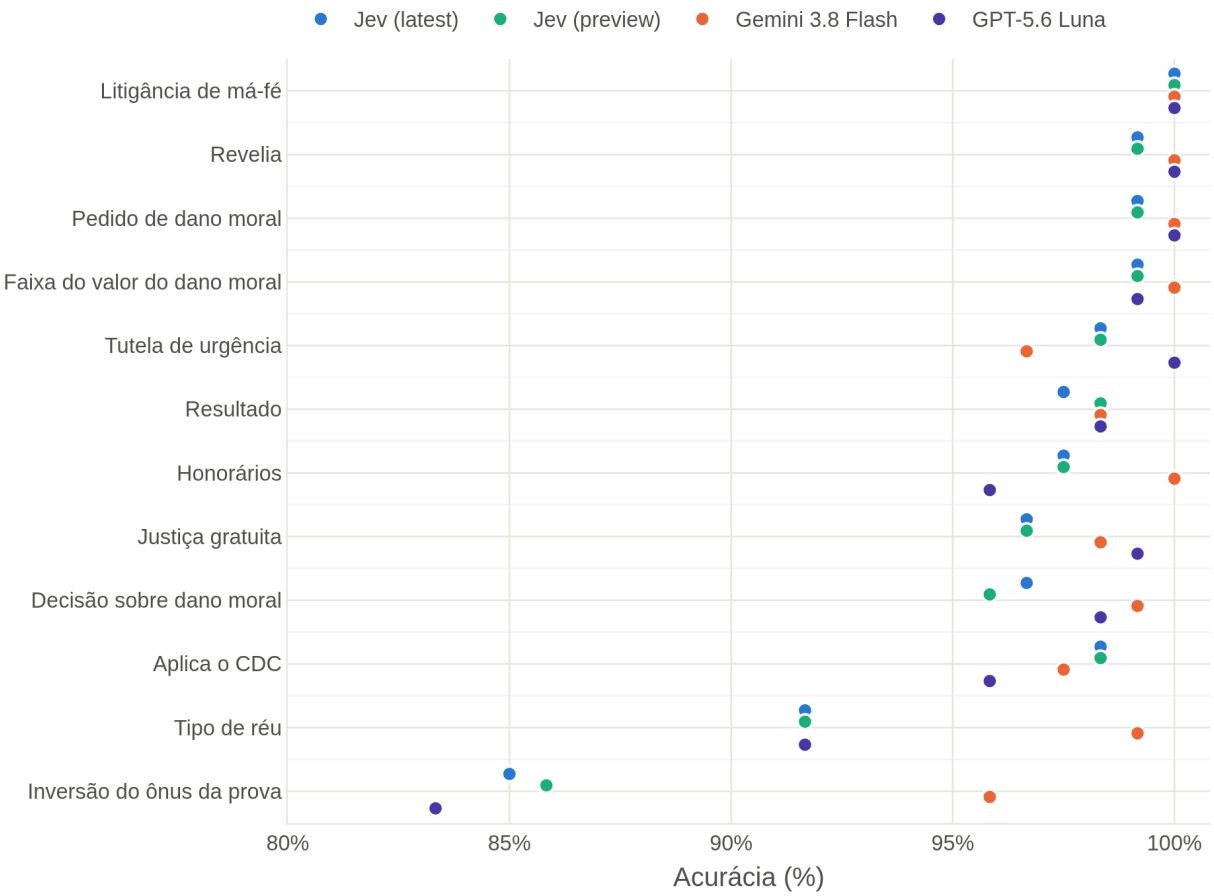

Figura 1: Acurácia por variável. As variáveis estão ordenadas da mais difícil (base) para a mais fácil (topo).

Os erros do Jev se concentram em **inversão do ônus da prova** e **tipo de réu** (57% dos erros). Nas duas é preciso interpretar a fundamentação: se a inversão do ônus foi de fato aplicada ou só citada, e qual é a atividade econômica do réu. Nas variáveis que se leem direto no dispositivo (resultado, valor, litigância de má-fé), os quatro modelos ficam praticamente empatados. O resultado se mantém ao comparar com cada referência isoladamente e ao olhar só as 34 respostas difíceis, adjudicadas às cegas: nelas o Gemini acertou 68%, o GPT 53% e o Jev 44%.

4.2 Tempo e custo

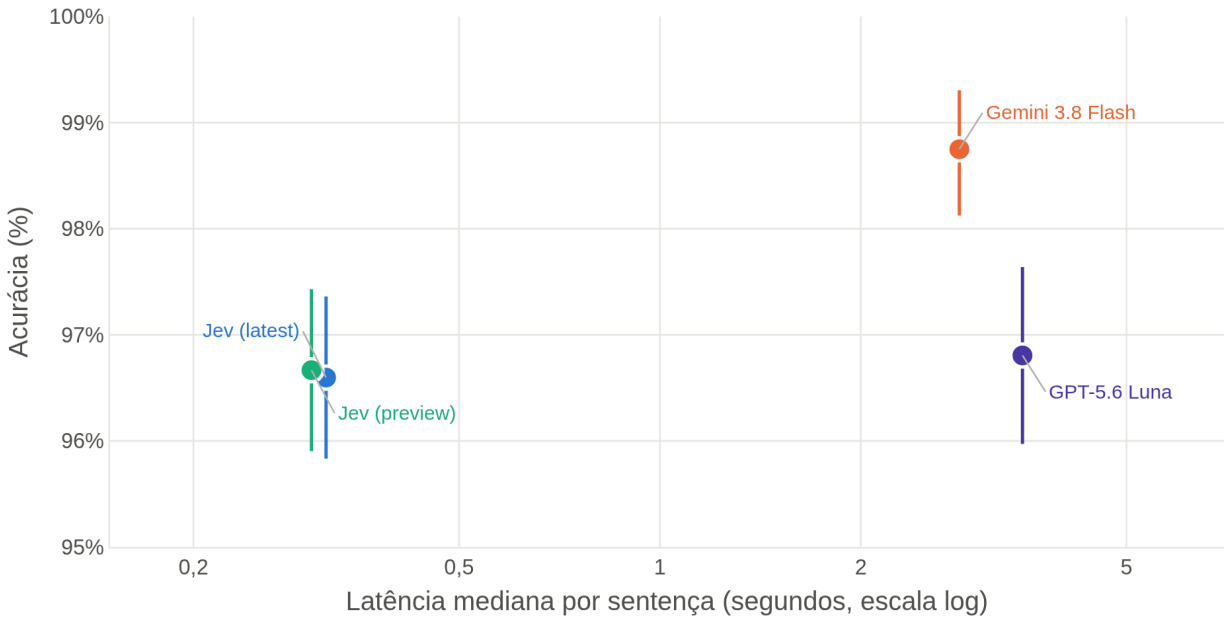

Figura 2: Qualidade versus tempo. Latência mediana por sentença em escala logarítmica; as barras verticais indicam o IC 95% da acurácia.

Tabela 3: Latência por sentença (12 perguntas por requisição, 4 requisições simultâneas), tempo total das 120 sentenças, tokens médios por sentença e custo estimado por mil sentenças.

Modelo	Mediana (s)	p90 (s)	Máx. (s)	Total (s)	Tokens entrada	Tokens saída	US$/mil
Jev (latest)	0,32	0,40	1,2	11	5.954	663	0,25
Jev (preview)	0,30	0,36	1,0	10	5.954	663	0,25
Gemini 3.8 Flash	2,81	4,41	9,2	94	5.081	733	6,56
GPT-5.6 Luna	3,49	5,37	14,4	110	6.791	232	1,64

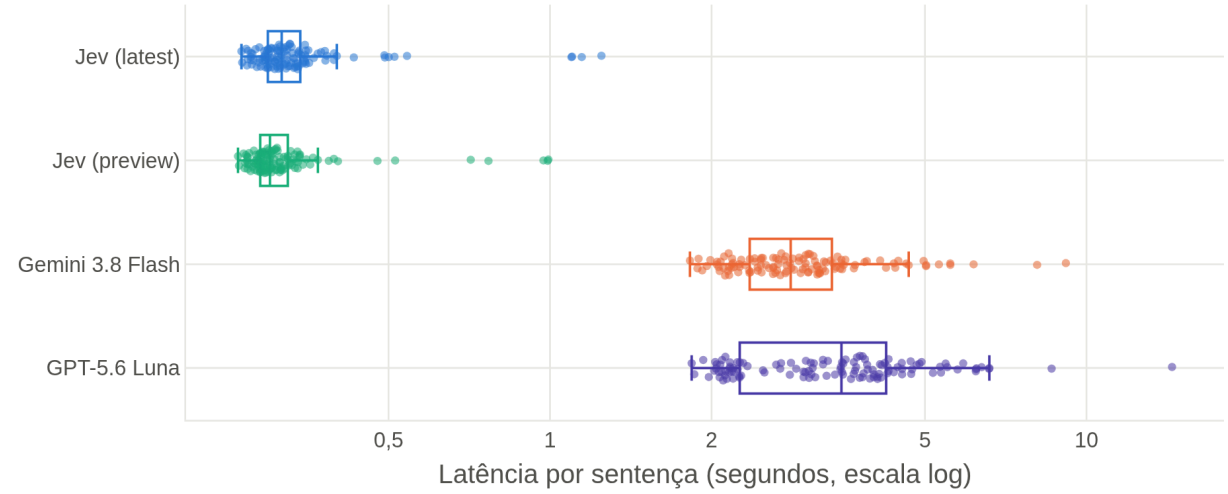

Figura 3: Distribuição da latência por sentença (escala logarítmica). Cada ponto é uma sentença.

Os preços por milhão de tokens (entrada/saída) são: Gemini 3.8 Flash US$ 0,75/3,75; GPT-5.6 Luna US$ 0,20/1,20; Jev US$ 0,042/0. O preço do Jev foi tirado dos exemplos da documentação da TypeSafe e ainda precisa ser confirmado na conta. A saída do Gemini inclui os tokens de raciocínio.

4.3 Confiança do Jev

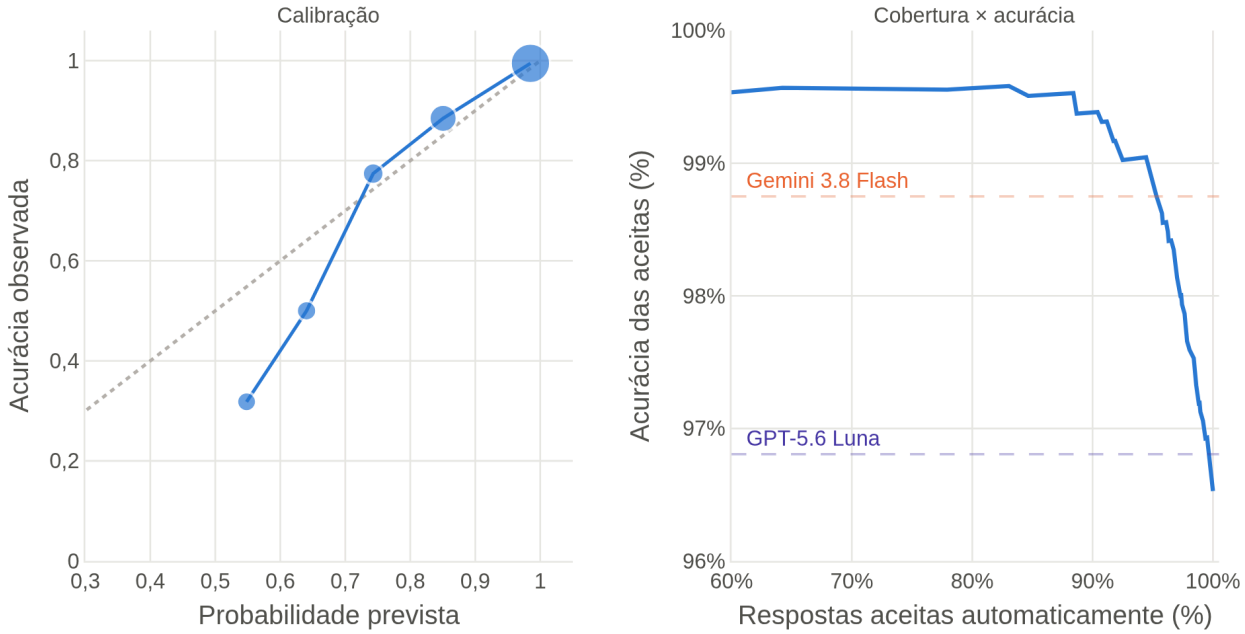

Figura 4: Esquerda: calibração, isto é, a probabilidade que o Jev atribui à resposta escolhida contra a acurácia observada; o tamanho do ponto é proporcional ao número de respostas. Direita: acurácia ao aceitar automaticamente apenas as respostas acima de um limiar de confiança, em função da fração aceita; as linhas tracejadas marcam a acurácia do Gemini e do GPT, que respondem tudo.

A confiança do Jev separa bem acertos de erros: a mediana é 0,98 nos acertos e 0,51 nos erros (ECE = 0,017; Brier das perguntas sim/não = 0,023). Com limiar de 0,8, o Jev decide sozinho 91% das respostas com 99,3% de acurácia, acima de qualquer modelo que responde tudo. Na faixa de probabilidade baixa, porém, ele é **sobreconfiante**: esses valores servem para ordenar e triar, não como probabilidade literal de acerto.

5 Conclusões

1. **O Jev cumpre a promessa de estrutura, velocidade e custo.** Não houve respostas fora do esquema nem falhas. Foi cerca de 10× mais rápido que os LLMs flash e custou de 7 a 26× menos.
2. **Na qualidade bruta, perde para o Gemini 3.8 Flash e empata com o GPT-5.6 Luna.** A diferença para o Gemini (~2 p.p. por resposta) é significativa e se concentra nas variáveis interpretativas.
3. **A confiança calibrada é o diferencial.** Com revisão humana só dos casos incertos (~9%), o Jev superou a qualidade de todos os modelos testados.
4. **Recomendação:** use o Gemini 3.8 Flash quando o volume for moderado e a qualidade máxima sem revisão for prioridade. Use o Jev em alto volume, tempo real ou orçamento apertado, idealmente com triagem pela confiança e com perguntas interpretativas quebradas em perguntas mais simples (por exemplo, "a sentença *cita* a inversão?" e "a sentença *decide* com base nela?").

6 Limitações

- **Gabarito sem juristas:** as referências e a adjudicação são modelos. As referências são das famílias de dois modelos avaliados, embora os resultados se mantenham nas análises de sensibilidade.
- **Efeito teto:** com todos os modelos acima de 96%, 120 sentenças não detectam diferenças menores que ~1 p.p.
- **Só perguntas fechadas:** o Jev não extrai texto livre nem valores exatos, e o esquema foi desenhado dentro dessa restrição.
- **Configurações:** os LLMs rodaram com o raciocínio padrão; latência e preços são de um único dia e de uma única conta.

7 Reprodutibilidade

Código, respostas dos modelos e padrão-ouro estão em github.com/lab-dados/jev-anotacao-sentencas. O texto integral das sentenças não é redistribuído (LGPD, art. 6º, III), mas a coleta é reproduzível com `scripts/01_coletar.py`.

Construir o padrão-ouro com as referências custou cerca de US$ {num(custo.loc[REFERENCIAS, 'usd_total'].sum(), 2)} em API.