Extração estruturada de documentos em condições reais de produção

Colocamos o pipeline da Brick frente a frente com os principais modelos multimodais do mercado, Claude Fable 5, Claude Opus 4.8, Gemini 3.1 Pro e GPT-5.5, e com produtos comerciais de extração de alto nível, como Reducto, Extend e AWS Textract. Em duas frentes: dados reais da operação de seguros no Brasil e o benchmark público RealDocBench.

Brick Research · brick.so Publicado: **julho de 2026** Experimentos: **jun/2026** Contato: **contato@brick.so**

Resumo executivo

Os benchmarks que o mercado divulga medem extração de dados em condições de laboratório: PDFs digitais nativos e capturas impecáveis. A operação de uma seguradora vive do oposto, a foto torta do celular, o scan de baixa qualidade, o carimbo sobreposto, o reflexo na CNH e as particularidades dos formatos brasileiros (pt-BR). Este whitepaper apresenta duas avaliações independentes e reproduzíveis, conduzidas pela Brick em junho de 2026, que medem o que acontece com a automação quando o dado é o da vida real.

Estudo 1, dados reais de produção: sobre 326 documentos reais e 2.769 campos validados por especialistas[1], o pipeline da Brick alcançou **96,1%** de acurácia, 7,4 pontos percentuais acima do melhor modelo de fronteira (Claude Fable 5, 88,7%) e 12,1 acima do melhor produto comercial (Extend Parse 2.0, 84,0%)[4]. Nos campos **numéricos**, o coração de qualquer esteira de seguros, as soluções externas ficam entre **60% e 67%** de acerto; a Brick sustenta **97%**. E quando o documento deixa de ser um PDF nativo e vira scan ou foto, a Brick perde cerca de 4 p.p.; os concorrentes, entre 12 e 23 p.p.[5]

Estudo 2, benchmark público: no RealDocBench (1.356 questões sobre 573 PDFs reais de negócios), o pipeline de produção da Brick, sem nenhum ajuste para o benchmark, registrou **86,4%** de acurácia por questão: a 2ª posição na tabela de referência, à frente de LlamaParse Agentic (84,3%), Reducto (83,1%), Gemini 3.5 Flash (81,6%), Azure DI (78,9%) e AWS Textract (53,6%)[6].

Acurácia geral: dados reais de produção	Campos numéricos (NUMBER)	Degradação PDF nativo → scan/foto	RealDocBench: acurácia por questão
96,1%	**97%**	**−4,2 p.p.**	**86,4%**
+7,4 p.p. vs. melhor modelo de fronteira	vs. **60–67%** de todas as soluções externas	vs. **−12 a −23 p.p.** das externas[5]	2º lugar na tabela de referência[6]

Conteúdo

1 Contexto e motivação

A escolha de tecnologia de extração define a eficiência de toda a esteira de uma seguradora, do aviso de sinistro ao reembolso saúde e à subscrição. E os números divulgados pelo mercado raramente refletem as condições em que essa esteira opera.

Benchmarks públicos de extração e parsing são construídos, em sua maioria, sobre PDFs digitais nativos, papers acadêmicos e layouts limpos. Nesse regime, como os dados a seguir mostram, praticamente todas as soluções, inclusive os modelos genéricos, entregam entre 94% e 100%. O problema é que a produção não vive nesse regime: o dado real que chega a uma seguradora brasileira é a foto torta do celular, a CNH com reflexo, o recibo médico com carimbo sobreposto, o comprovante de residência amassado, somados às particularidades de formato do pt-BR (datas `dd/mm/aaaa`, separadores de milhar, códigos de selo, números de CRM).

Este whitepaper documenta o que acontece com a acurácia de extração quando o teste usa a **distribuição real da produção**. E valida, em um benchmark público independente, que os resultados não são um artefato do nosso próprio domínio. Duas perguntas orientaram os experimentos:

P1. Um modelo multimodal de fronteira, ou um produto comercial de extração, recebendo exatamente o mesmo arquivo bruto, consegue igualar ou superar a estratégia de extração da Brick em campos estruturados?

P2. Fora do nosso domínio, em um benchmark público de documentos reais de negócios, como o pipeline de produção da Brick se posiciona frente à infraestrutura global de document AI?

2 Desenho experimental

Conduzimos dois estudos independentes, cada um com dataset, harness e critérios de sucesso próprios, ambos com artefatos imutáveis e configuração integralmente versionada, para que qualquer resultado possa ser auditado e reproduzido.

Visão geral dos dois estudos

	Estudo 1: Dados reais de produção	Estudo 2: RealDocBench (público)
Dataset	326 documentos de produção · 2.769 campos validados por especialistas[1]	1.356 questões · 573 PDFs reais de negócios (finance, mortgage, supply-chain, saúde)
Sistemas avaliados	Pipeline Brick · Claude Fable 5 · Claude Opus 4.8 · Gemini 3.1 Pro · GPT-5.5 · Reducto · Extend Parse 2.0† · AWS Textract†	Pipeline Brick (config de produção) vs. leaderboard publicado (Extend, LlamaParse, Reducto, Gemini, Azure DI, AWS Textract)[6]
Métrica	Acurácia por campo, rubrica type-aware[2]	Acurácia por questão (juiz LLM binário)[7]
Entrada	Arquivo bruto (imagem/PDF), multimodal, sem markdown intermediário	PDF bruto; o servidor monta o schema por questão e roda o pipeline completo
Reprodutibilidade	seed 42 · temperatura 0.0 · config versionada	dataset pinado `v20260615` · config de produção montada pelo servidor

† Braços "parser-markdown": o documento é convertido em markdown pelo parser e um extrator fixo (Gemini 3 Flash) responde ao mesmo schema, isola a qualidade do parser, mas não é diretamente comparável aos braços de arquivo bruto[4].

Estudo 1: Dados reais de produção

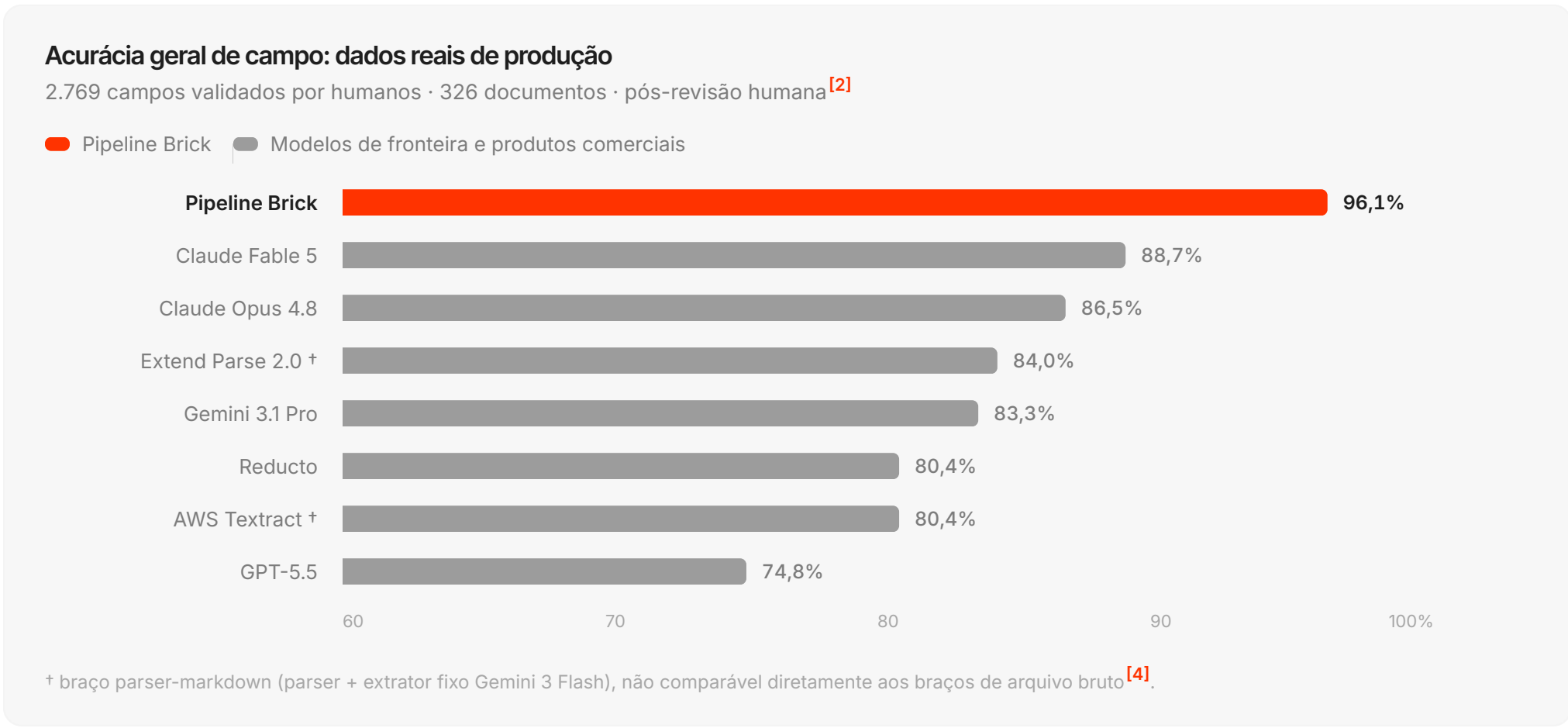
3.1 Metodologia

Cada sistema recebeu o **arquivo bruto** do documento (imagens como imagem, PDFs como arquivo; TIFF renderizado para PNG) e o mesmo schema de campos. A correção segue uma rubrica sensível ao tipo do campo: **BOOLEAN** e **DATE** exigem igualdade exata após normalização (datas interpretadas no padrão brasileiro, dia primeiro); **NUMBER** é comparado numericamente com tolerância de ±1%; **STRING** é julgado por LLM[2]. Todas as 1.205 disputas de julgamento passaram por **revisão manual**, que corrigiu 320 vereditos a favor dos modelos externos, os números abaixo já refletem essa revisão. O braço da Brick **não** participou dela e reporta o número conservador, sem correções[3].

3.2 Resultado geral

O pipeline da Brick lidera com **96,1%**, 7,4 pontos percentuais acima do melhor modelo de fronteira e 21,3 acima do mais fraco. Nenhum sistema externo, fosse modelo de fronteira ou produto comercial, superou o pipeline da Brick em nenhum recorte do dataset: nem por classe de documento, nem por tipo de captura, nem por faixa de qualidade.

Acurácia geral de campo: dados reais de produção
2.769 campos validados por humanos · 326 documentos · pós-revisão humana[2]



† braço parser-markdown (parser + extrator fixo Gemini 3 Flash), não comparável diretamente aos braços de arquivo bruto[4].

Tabela 1: Acurácia por tipo de campo, além do agregado (pós-revisão)

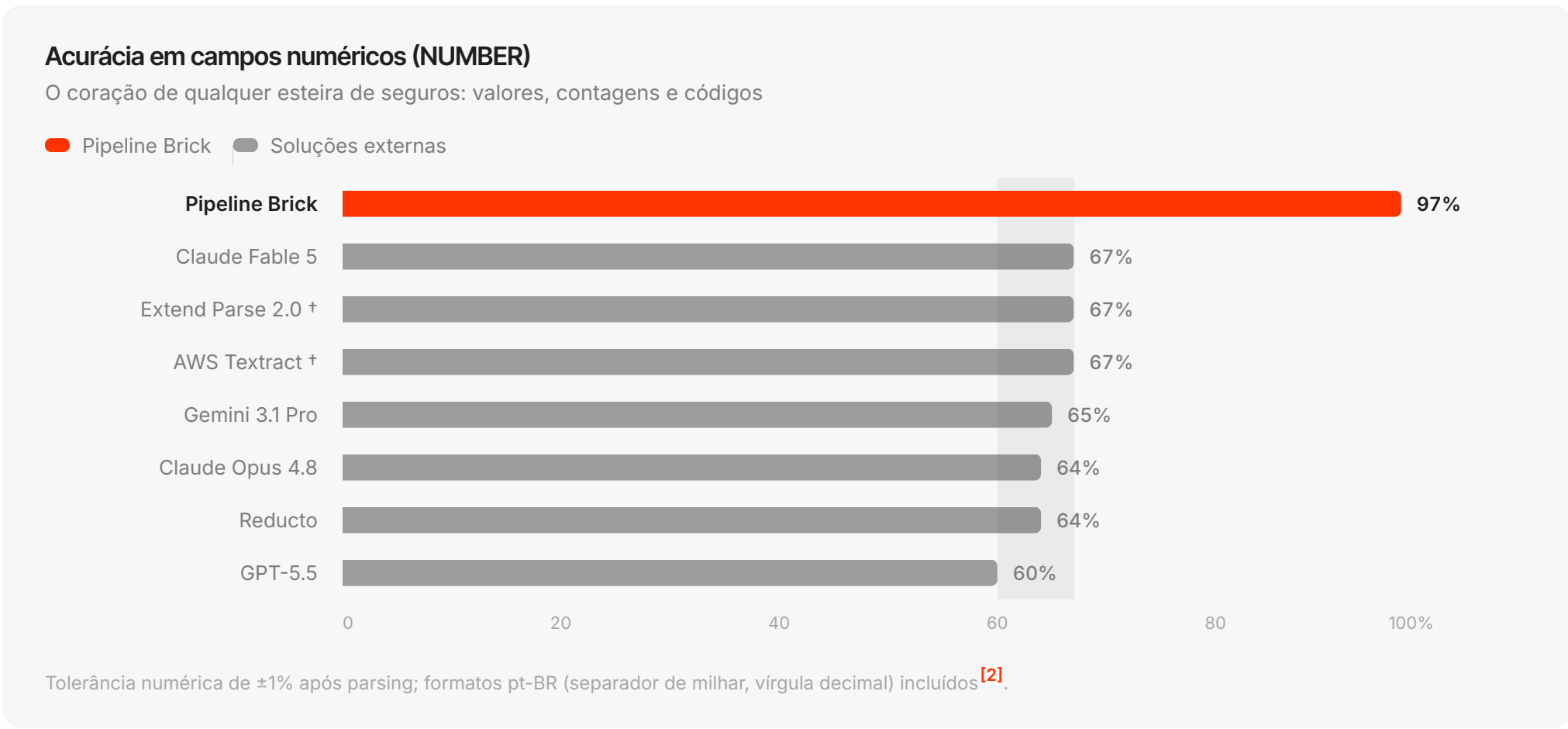
Sistema	STRING	BOOLEAN	DATE
Pipeline Brick (interno)	**96%**	**98%**	**95%**
Claude Fable 5	89%	91%	87%
Claude Opus 4.8	86%	93%	81%
Extend Parse 2.0 †	82%	93%	88%
Gemini 3.1 Pro	83%	92%	78%
Reducto	81%	84%	75%
AWS Textract †	78%	90%	87%
GPT-5.5	74%	91%	**58%**

† braço parser-markdown[4].

3.3 O abismo operacional: campos numéricos e datas

A diferença mais importante do estudo não aparece na média, aparece nos **tipos estruturados**. Em campos **NUMBER** (valores monetários, contagens, códigos de selo, números de CRM), todas as soluções externas desabam para a faixa de **60–67%** de acerto, enquanto o pipeline da Brick sustenta **97%**. **DATE** é a segunda maior lacuna (o GPT-5.5 chega a 58%). Em STRING e BOOLEAN, os modelos de fronteira são competitivos. Ou seja: é exatamente onde a normalização de domínio importa, números e datas em formato brasileiro, que os sistemas genéricos falham.

Acurácia em campos numéricos (NUMBER)
O coração de qualquer esteira de seguros: valores, contagens e códigos

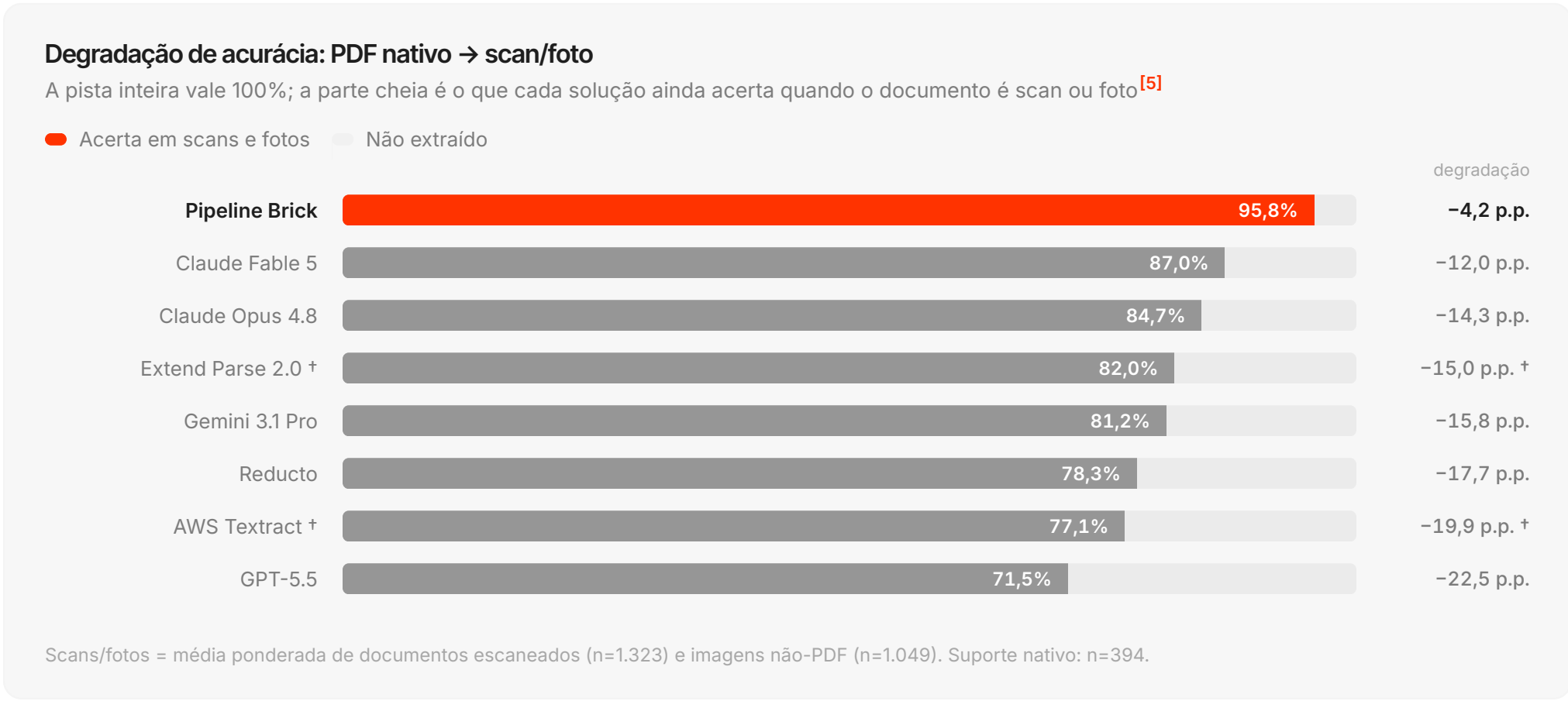


Tolerância numérica de ±1% após parsing; formatos pt-BR (separador de milhar, vírgula decimal) incluídos[2].

3.4 Resiliência à captura: nativo vs. scan/foto

Em **PDFs digitais nativos, todos os sistemas empatam (94–100%)**: o texto é limpo e não há o que errar. Esse é o regime que os benchmarks de laboratório medem, e é por isso que os números de marketing parecem todos iguais. A separação acontece nos **scans e fotos**, que dominam a produção real: a Brick cede apenas ~4 p.p. (100% → 95,8%), enquanto as soluções externas perdem de 12 a 23 p.p.[5] A diferença não está em quem acerta o documento perfeito, está em quem não colapsa quando o documento deixa de ser perfeito.

Degradação de acurácia: PDF nativo → scan/foto
A pista inteira vale 100%; a parte cheia é o que cada solução ainda acerta quando o documento é scan ou foto[5]



Scans/fotos = média ponderada de documentos escaneados (n=1.323) e imagens não-PDF (n=1.049). Suporte nativo: n=394.

Tabela 2: Acurácia por tipo de captura, separando scan de foto[5]

Sistema	Scan	Foto/imagem
Pipeline Brick	**94%**	**98%**
Claude Fable 5	91%	82%
Claude Opus 4.8	86%	83%
Extend Parse 2.0 †	86%	77%
Gemini 3.1 Pro	83%	79%
Reducto	84%	71%
AWS Textract †	82%	71%
GPT-5.5	68%	76%

O mesmo padrão se repete nas **faixas de qualidade de imagem**: a acurácia da Brick permanece praticamente estável (~95–98%) da pior faixa à melhor, enquanto as soluções externas afundam nas capturas degradadas, com a maior dispersão justamente na cauda de baixa qualidade que os benchmarks públicos sub-representam[8].

3.5 Por classe de documento

O pipeline da Brick lidera em **todas** as classes. As soluções externas vão melhor em documentos de identidade (RGs, CNHs), que têm layout limpo e padronizado, e pior nas classes heterogêneas e de formato livre (outros, comprovantes de residência), exatamente as que dominam as filas de sinistro e subscrição.

Tabela 3: Acurácia por classe de documento (suporte n entre parênteses)[8]

Sistema	Certidões (746)	Outros (611)	Compr. resid. (522)	Financeiros (307)	CNHs (295)	RGs (256)	Legais* (32)
Pipeline Brick	**96%**	**98%**	**90%**	**98%**	**100%**	**99%**	**97%**
Claude Fable 5	93%	83%	87%	83%	92%	97%	91%
Claude Opus 4.8	89%	81%	81%	83%	93%	97%	84%
Extend Parse 2.0 †	88%	77%	78%	81%	91%	96%	84%
Gemini 3.1 Pro	86%	79%	79%	79%	90%	95%	72%
Reducto	89%	72%	73%	74%	92%	86%	81%
AWS Textract †	88%	73%	70%	77%	91%	91%	66%
GPT-5.5	73%	75%	68%	78%	83%	82%	75%

* Fatia fina (n=32), leitura direcional apenas[8].

Estudo 2: Validação em benchmark público (RealDocBench)

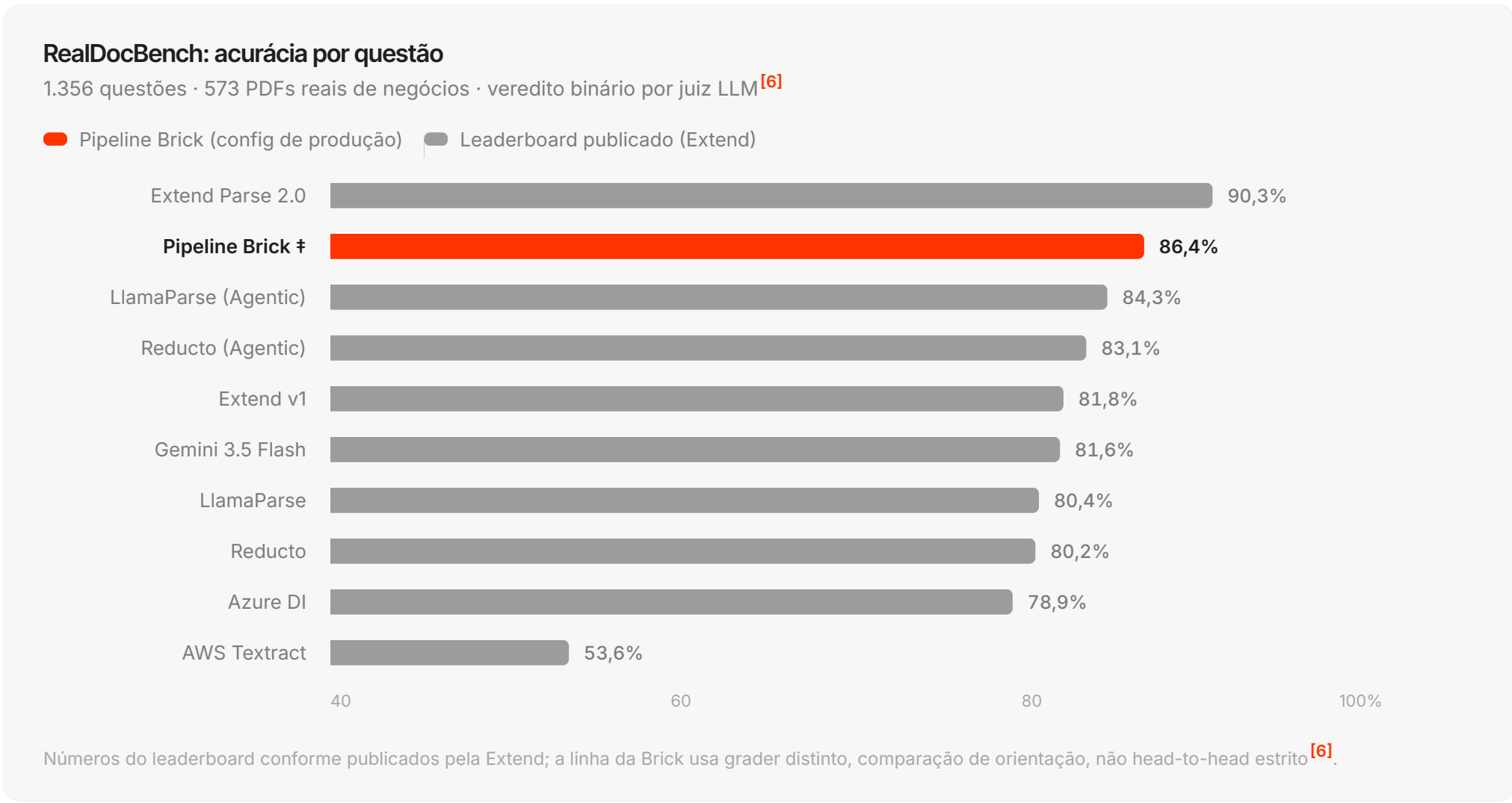
4.1 Metodologia

Para garantir que a vantagem observada não fosse um viés do nosso próprio domínio, medimos o pipeline de **produção** da Brick, a mesma configuração que atende clientes hoje, montada pelo próprio servidor, sem nenhum ajuste para o benchmark, no RealDocBench: 1.356 questões de resposta tipada sobre 573 PDFs reais de negócios (finanças, crédito imobiliário, supply chain e saúde). A métrica principal é a do próprio benchmark: **acurácia por questão**, com veredito binário de juiz LLM, certo ou errado, sem crédito parcial[7].

4.2 Resultado

O pipeline da Brick registrou **86,4%** de acurácia por questão sobre 1.337 casos respondidos (85,2% contabilizando 19 falhas de infraestrutura como erro[7]), a **2ª posição** na tabela de referência do benchmark, atrás apenas do Extend Parse 2.0 (90,3%) e à frente de LlamaParse Agentic, Reducto, Gemini 3.5 Flash, Azure DI e AWS Textract[6]. Um pipeline especializado, desenhado para o documento brasileiro imperfeito, competindo de igual para igual com a infraestrutura global de document AI, em um benchmark de outro país, outro idioma e outros domínios.

RealDocBench: acurácia por questão
1.356 questões · 573 PDFs reais de negócios · veredito binário por juiz LLM[6]



Números do leaderboard conforme publicados pela Extend; a linha da Brick usa grader distinto, comparação de orientação, não head-to-head estrito[6].

Tabela 4: Leaderboard de referência, acurácia por campo[6]

#	Sistema	Por campo
1	Extend Parse 2.0	95,7%
2	**Pipeline Brick (este estudo)**	**86,5% ‡**
3	LlamaParse (Agentic)	92,1%
4	Reducto (Agentic)	91,1%
5	Extend v1	90,4%
6	Gemini 3.5 Flash	89,0%
7	LlamaParse	89,0%
8	Reducto	88,5%
9	Azure DI	88,8%
10	AWS Textract	70,5%

Tabela 5: Brick por domínio do benchmark

Domínio	n	Por questão	Por campo ‡
Mortgage	468	88,9%	89,7%
Supply chain	316	86,1%	87,4%
Finance	373	84,7%	81,6%
Saúde	180	83,9%	79,1%
Geral	**1.337**	**86,4%**	**86,5%**

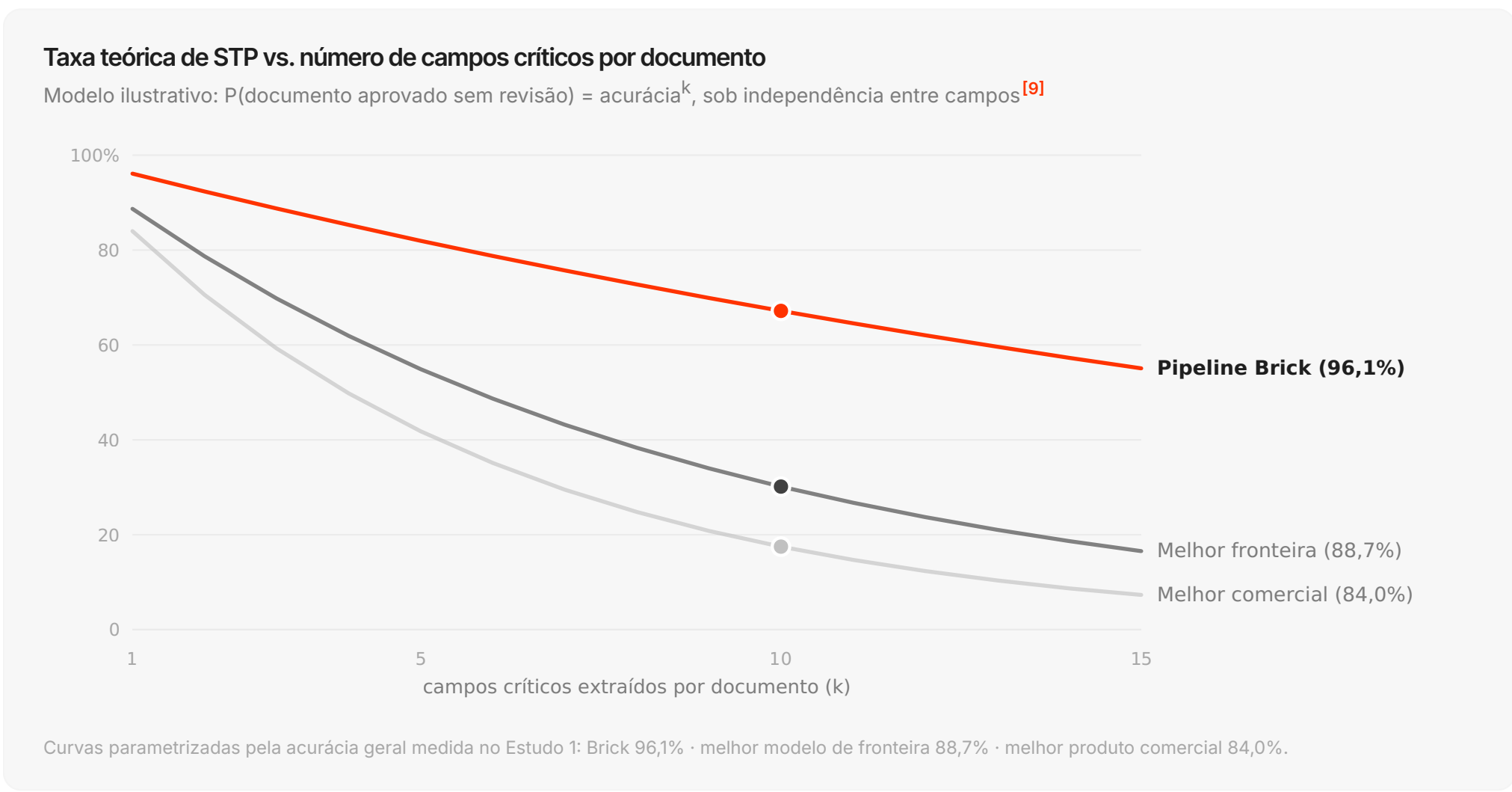
‡ Números da Brick apurados com grader distinto do leaderboard (juiz LLM da plataforma, não o `deep_equal` tipado da Extend); o por-campo da Brick é uma aproximação tudo-ou-nada por questão, que estruturalmente o subestima[6].

5 Implicações operacionais: o custo do retrabalho silencioso

Por que uma degradação de 12 a 15 pontos percentuais, somada a uma taxa de 60–67% em números e datas, é devastadora para a esteira de uma seguradora? A resposta está no Straight-Through Processing (STP).

Automação só gera retorno quando o documento cruza a esteira **sem intervenção humana**. Um documento típico de regulação de sinistro ou de reembolso exige acertar vários campos críticos ao mesmo tempo, e basta **um único valor numérico errado** para o documento inteiro cair na fila de revisão manual. A acurácia por campo se compõe de forma multiplicativa: com 10 campos críticos por documento, a taxa teórica de aprovação direta despenca muito mais rápido do que a acurácia média sugere[9].

Taxa teórica de STP vs. número de campos críticos por documento
Modelo ilustrativo: P(documento aprovado sem revisão) = acurácia^k, sob independência entre campos[9]



Curvas parametrizadas pela acurácia geral medida no Estudo 1: Brick 96,1% · melhor modelo de fronteira 88,7% · melhor produto comercial 84,0%.

Com 10 campos críticos, a taxa teórica de STP fica em torno de **67%** para o pipeline da Brick, **30%** para o melhor modelo de fronteira e **18%** para o melhor produto comercial[9]. É assim que métricas altas de laboratório viram **"falsas automações"**: o trabalho não desaparece, ele é transferido para um backoffice oculto de revisão manual, que corrói SLAs e infla o custo operacional enquanto o dashboard de acurácia segue verde.

A lição central dos dois estudos: em PDF nativo, todo mundo empata. A decisão de tecnologia precisa ser tomada sobre a distribuição real da operação, scans, fotos, formatos pt-BR e campos estruturados, porque é ali que a esteira vive, e é ali que os sistemas genéricos falham.

6 Limitações e notas metodológicas

Publicamos estes resultados com as mesmas ressalvas registradas nos relatórios internos de pesquisa, acreditamos que um benchmark só tem valor se puder ser escrutinado. As notas abaixo são parte integral das conclusões.

1. **Dataset interno.** 326 documentos de produção e 2.769 campos com gabarito validado por especialistas (`document-field-extraction@v20260611`). A distribuição sobre-representa deliberadamente capturas degradadas (scans n=1.323, fotos/imagens n=1.049, nativos n=394) em relação a benchmarks públicos, é o cenário de produção, não um cenário neutro. ↩
2. **Rubrica de pontuação.** Regra de nulo (gabarito vazio → correto somente se a predição também estiver vazia); BOOLEAN e DATE exatos após normalização (datas com `dayfirst` pt-BR); NUMBER com tolerância de ±1% após parsing; STRING julgado por LLM (Gemini 3.1 Pro). Todas as 1.205 disputas do juiz foram revisadas manualmente; 320 vereditos excessivamente rígidos foram corrigidos a favor dos braços externos. Todos os números do Estudo 1 são pós-revisão. Seed 42, temperatura 0.0, prompt único para todos os LLMs.
3. **O braço da Brick não passou pela revisão humana de disputas.** Seu 96,1% é o número conservador, apurado apenas pelo juiz automático; as 320 correções beneficiaram exclusivamente os braços externos. Uma revisão simétrica só poderia ampliar a vantagem do pipeline da Brick, a comparação não é enviesada a nosso favor.
4. **Braços parser-markdown (Extend Parse 2.0 e AWS Textract).** Nesses braços o documento é convertido em markdown pelo parser e um extrator fixo (Gemini 3 Flash, temperatura 0) responde ao schema a partir do markdown, com a mesma rubrica e o mesmo juiz. Isso isola a qualidade do parser, mas introduz um intermediário com perdas e um extrator de classe Flash, os valores absolutos não são diretamente comparáveis aos braços de arquivo bruto. Esses braços não passaram pela revisão humana, o que os torna conservadores: a revisão corrigiu ~27% das disputas STRING a favor dos avaliados e só poderia elevá-los.
5. **"Scans/fotos (combinado)" e degradação.** Coluna derivada: média ponderada pelo suporte de scanned (n=1.323) e not_pdf/imagem (n=1.049) da Tabela 2; degradação = acurácia em PDF nativo menos essa média. Os valores por célula constam do relatório-fonte; o arredondamento é para 1 casa decimal.
6. **Comparabilidade com o leaderboard RealDocBench.** O leaderboard da Extend pontua parsers: o markdown de cada parser alimenta um extrator fixo (Gemini 3 Flash) e é avaliado com `deep_equal` tipado. A linha da Brick mede o pipeline completo de ponta a ponta, avaliado pelo juiz LLM binário da própria plataforma. As duas medições vivem em espaços métricos diferentes, tratamos a posição na tabela como orientação de ordem de grandeza, não como head-to-head estrito. O por-campo da Brick (86,5%) é derivado do veredito por questão (tudo-ou-nada, ponderado pelo número de campos), o que o subestima estruturalmente frente ao scoring por campo independente da Extend. Um head-to-head verdadeiro (Brick-como-parser no estágio de score da Extend) está no roadmap de pesquisa.
7. **Falhas de infraestrutura no RealDocBench.** 19 de 1.356 casos falharam por razões de infraestrutura (14 por uma corrida de máquina de estados no servidor; 5 por timeout de extração >600 s nos documentos mais pesados), não por erro do modelo. Por isso reportamos 86,4% sobre os 1.337 respondidos e 85,2% contando falhas como erro. As falhas concentram-se nos documentos mais difíceis.
8. **Fatias finas são direcionais.** A classe legais (n=32), o tipo hybrid (n=3) e as faixas de pior qualidade (n=81–98) têm suporte pequeno; a leitura ali é de tendência, não de ponto. A tendência agregada, externos degradam com a qualidade da captura; a Brick permanece estável, é robusta.
9. **Modelo de STP é ilustrativo.** A curva P = acurácia^k assume independência entre erros de campos e acurácia homogênea entre campos, uma simplificação para dimensionar o mecanismo de composição, não uma medição de STP em produção. Erros correlacionados por documento tornam as taxas reais diferentes das teóricas; o ordenamento entre sistemas, porém, é preservado pela monotonicidade.
10. **Dependência de juiz LLM.** No Estudo 1, campos STRING usam o Gemini 3.1 Pro como juiz, auto-preferência ao julgar as próprias respostas do Gemini é possível, e o passe de revisão humana mitiga isso no conjunto disputado. No Estudo 2 o juiz é um modelo único; autoconsistência de juiz é uma ameaça de validade conhecida e declarada.

7 Reprodutibilidade

Ambos os estudos foram executados com configuração integralmente versionada e artefatos imutáveis (parquets originais e revisados, sumários, vereditos por caso e planilhas de revisão) preservados em armazenamento interno de pesquisa. Estudo 1: seed 42, temperatura 0.0, concorrência 16, prompts e tolerâncias da rubrica em `config.yaml` versionado. Estudo 2: dataset pinado (`realdocbench@v20260615-191922`, canonificação 1:1 do dataset público Extend-AI/RealDoc-Bench), com a configuração de produção montada pelo servidor e registrada junto ao run, o harness nunca constrói a configuração à mão. Pesquisadores e clientes interessados nos artefatos ou em replicação podem escrever para contato@brick.so.

8 Como citar

```
Brick Research (2026). "Extração estruturada de documentos em condições reais de
produção: uma avaliação comparativa entre pipeline especializado, modelos de
fronteira e produtos comerciais." Whitepaper técnico, Brick.
https://brick.so/research, julho de 2026.
```

Referências

[A] RealDocBench, benchmark, leaderboard e metodologia de scoring. Extend. github.com/extend-hq/realdoc-bench · extend.ai/resources/realdocbench
[B] Extend-AI/RealDoc-Bench, dataset público. huggingface.co/datasets/Extend-AI/RealDoc-Bench
[C] Reducto Extract API, docs.reducto.ai/extract/overview
[D] Brick Research, relatórios internos: `frontier-vs-internal-extraction` (2026-06-12) e `platform-reading-realdocbench` (2026-06-16), projeto `document-field-extraction`.