/thoughtworks

Design. Engenharia. IA.

# Technology Radar

## Um guia de opinião sobre o universo de tecnologia atual

Volume 33
Nov. 2025

# Sobre o Radar

Thoughtworkers são pessoas apaixonadas por tecnologia. Nós desenvolvemos, pesquisamos, testamos, contribuímos com código livre, escrevemos sobre e visamos a sua constante melhoria — para todas as pessoas. Nossa missão é liderar e promover a excelência de software e revolucionar a indústria de TI. Nós criamos e compartilhamos o Technology Radar da Thoughtworks para apoiar essa missão. O Conselho Consultivo de Tecnologia (Technology Advisory Board, ou TAB), um grupo de lideranças experientes em tecnologia da Thoughtworks, é responsável por criar o Radar. O grupo se reúne regularmente para discutir a estratégia global de tecnologia da empresa e as tendências tecnológicas que impactam significativamente a nossa indústria.

O Radar capta o resultado das discussões do TAB em um formato que procura oferecer valor a uma ampla gama de pessoas interessadas, de pessoas que desenvolvem software a CTOs. O conteúdo é concebido para ser um resumo conciso.

Nós encorajamos você a explorar essas tecnologias. O Radar é gráfico por natureza, agrupando os itens em técnicas, ferramentas, plataformas, linguagens e frameworks. No caso de itens que podem ser classificados em mais de um quadrante, escolhemos aquele que parece mais adequado.

Além disso, agrupamos esses itens em quatro anéis para refletir nossas opiniões atuais em relação a cada um. Para mais informações sobre o Radar, acesse: thoughtworks.com/pt/radar/faq.

# Radar em um relance

A ideia por trás do Radar é rastrear tópicos interessantes, que chamamos de blips. Organizamos blips no Radar usando duas categorias: quadrantes e anéis. Os quadrantes representam as diferentes naturezas dos blips. Os anéis indicam nossa recomendação para utilizar a tecnologia.

Um blip é uma tecnologia ou técnica que desempenha um papel no desenvolvimento de software. Os blips estão "em movimento" — suas posições no Radar estão constantemente mudando — geralmente indicando que nossa confiança em recomendá-los tem crescido à medida que se movimentam entre os anéis.


**Adote:** Acreditamos firmemente que a indústria deveria adotar esses itens. Quando apropriado, nós os usamos em nossos projetos.

**Experimente:** Vale a pena ir atrás. É importante entender como desenvolver essa capacidade. As empresas devem testar a tecnologia em um projeto que possa lidar com o risco.

**Avalie:** Vale explorar com o objetivo de compreender como afetará sua empresa.

**Evite:** Prossiga com cautela.

**Novo**

**Mudança de anel**

**Sem alterações**

Nosso Radar é um olhar para o futuro. Para abrir espaço para novos itens, apagamos itens que não foram modificados recentemente, o que não é um reflexo de seus valores, mas uma solução para nossa limitação de espaço.

# Contribuidoras

O Conselho Consultivo de Tecnologia (TAB) é um grupo formado por 22 tecnologistas experientes da Thoughtworks. O TAB se reúne duas vezes por ano pessoalmente e quinzenalmente por vídeochamada. Sua principal atribuição é ser um grupo consultivo para nossa CTO Rachel Laycock.

O TAB atua examinando tópicos que afetam soluções de tecnologia e tecnologistas da Thoughtworks. Esta edição do Thoughtworks Technology Radar é baseada em uma reunião do TAB realizada em Bucareste em Setembro de 2025.

Rachel Laycock (CTO)

Martin Fowler (Chief Scientist)

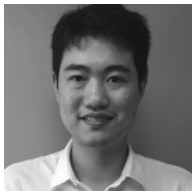
Alessio Ferri

Bharani Subramaniam

Birgitta Böckeler

Bryan Oliver

Camilla Falconi Crispim



Chris Chakrit Riddhagni

Effy Elden

James Lewis

Kief Morris

Ken Mugrage

Maya Ormaza

Nati Rivera

Neal Ford

Ni Wang

Nimisha Asthagiri

Pawan Shah

Selvakumar Natesan

Shangqi Liu

Vanya Seth

Will Amaral

# Créditos de Produção

## Equipe Editorial

- **Willian Amaral**
  *Product Owner*
- **Nati Rivera**
  *Product Owner*
- **Preeti Mishra**
  *Project and Campaign Manager*
- **Richard Gall**
  *Content Editor*
- **Michael Koch**
  *Copy Editor*
- **Gareth Morgan**
  *Head of Content and Thought Leadership*

## Design e Multimídia

- **Leticia Nunes**
  *Lead Designer*
- **Sruba Deb**
  *Visual Designer*
- **Ryan Cambage**
  *Multimedia Specialist*
- **Anish Thomas**
  *Multimedia Designer*

## Experiência Digital

- **Rashmi Naganur**
  *Business Analyst*
- **Brigitte Britten-Kelly**
  *Digital Content Strategist*
- **Vandita Kamboj**
  *UX Designer*
- **Lohith Amruthappa**
  *Analytics Specialist*
- **Neeti Thakur**
  *Marketing Automation Specialist*

## Comunicação

- **Shalini Jagadish**
  *Internal Communications Specialist*
- **Hiral Shah**
  *Social Media Specialist*
- **Abhishek Kasegaonkar**
  *Social Media Specialist*
- **Michelle Surendran**
  *Public Relations Specialist*
- **Anushree Tapuriah**
  *Campaigns and Advertisement Specialist*
- **Prakhar Nigam**
  *Campaigns and Advertisement Specialist*

## Tradução — Português

- Ayrton Araujo
- Bruno Barreto
- Elis Meza
- Gabriela Tiago
- Gustavo de Paula Andrade
- Isabella Velleda
- Paulo Calixto
- Pedro Cavalcante
- Victor Luz
- Vivianne Guimarães
- Yuri Condori
- Willian Amaral

# Temas

## A orquestração de infraestrutura chega para a IA

Os workloads de IA estão levando as organizações a orquestrarem grandes frotas de GPUs tanto para treinamento quanto para inferência. Os times trabalham cada vez mais com modelos de tamanhos que excedem a capacidade de um único acelerador (mesmo com 80 GB de HBM), empurrando-os para o treinamento distribuído e a inferência multi-GPU. Como resultado, os times de plataforma estão construindo pipelines complexos, de múltiplos estágios, e fazendo o tuning contínuo para taxa de transferência e latência. As discussões neste espaço incluíram o Nvidia DCGM Exporter para telemetria da frota e o agendamento consciente de topologia para alocar jobs onde a largura de banda de interconexão é maior.

Antes desse aumento na demanda por GPUs, o Kubernetes já havia se consolidado como o orquestrador de contêineres de fato — e ele continua sendo um substrato forte para gerenciar workloads de IA em escala, mesmo enquanto também exploramos alternativas como micro e Uncloud. Estamos acompanhando padrões emergentes de agendamento cientes de GPU — como enfileiramento e cota via Kueue, acoplados com posicionamento consciente de topologia e gang scheduling — para co-localizar jobs multi-GPU em links GPU-para-GPU rápidos (por exemplo, NVLink/NVSwitch) e dentro de "ilhas" de data center contíguas (por exemplo, racks ou pods com RDMA). Melhorias recentes nas APIs cientes de multi-GPU e NUMA no Kubernetes fortalecem ainda mais essas capacidades, melhorando a largura de banda entre dispositivos, reduzindo a latência de cauda e aumentando a utilização efetiva.

Esperamos uma inovação rápida na infraestrutura de IA à medida que os times de plataforma correm para suportar a crescente demanda por workflows de programação com IA e o surgimento de agentes elevados pelo MCP. Em nossa visão, a orquestração ciente de GPU está se tornando o básico — a topologia agora é uma preocupação de primeira classe no agendamento.

## A ascensão de agentes elevados por MCP

A ascensão dupla do MCP e dos agentes — e do ecossistema em expansão de protocolos e ferramentas construídos em torno deles — domina esta edição do Radar. Praticamente todos os principais fornecedores estão adicionando a consciência de MCP às suas ferramentas, o que faz sentido: de muitas maneiras, o MCP se tornou o protocolo de integração definitivo para potencializar os agentes e permitir que eles trabalhem de forma eficiente e semi-autônoma. Essas capacidades são centrais para tornar produtivos os fluxos de trabalho com agentes.

Observamos a inovação contínua nos fluxos de trabalho com agentes, onde a engenharia de contexto provou ser crítica para otimizar tanto o comportamento quanto o consumo de recursos. Novos protocolos como A2A e AG-UI estão reduzindo o código repetitivo necessário para construir e escalar

aplicações multiagente voltadas para a pessoa usuária. No espaço de desenvolvimento de software, comparamos diferentes maneiras de fornecer contexto aos agentes de programação — desde arquivos AGENTS.md até padrões como ancorar agentes de programação a uma aplicação de referência. Como esperado no ecossistema de IA, cada Radar traz uma nova onda de inovação — na edição passada foi o RAG; desta vez, são os fluxos de trabalho com agentes e a crescente constelação de ferramentas, técnicas e plataformas que os suportam, juntamente com alguns antipadrões de IA emergentes que merecem atenção.

## Workflows de programação com IA

É inegável que a IA está transformando a forma como construímos e mantemos software — e continua sendo um tema central em nossas discussões recentes. À medida que a IA se integra estrategicamente em toda a cadeia de valor do desenvolvimento — desde o uso de IA para compreender bases de código legadas até a GenAI para engenharia direta — estamos aprendendo a fornecer conhecimento de forma mais eficaz aos agentes de programação. As equipes estão experimentando novas práticas, como a definição de instruções personalizadas por meio de arquivos AGENTS.md e a integração com servidores MCP, como o Context7, para acessar documentação atualizada de dependências.

Há também uma compreensão crescente de que a IA deve potencializar o time inteiro, não apenas contribuidoras individuais. Técnicas como bibliotecas de instruções compartilhadas e curadas e comandos personalizados estão surgindo para promover uma disseminação mais equitativa do conhecimento. O cenário de ferramentas é vibrante: designers exploram o UX Pilot e o AI Design Reviewer, enquanto pessoas desenvolvedoras prototipam rapidamente com o v0 e o Bolt para prototipagem de UI de autoatendimento.

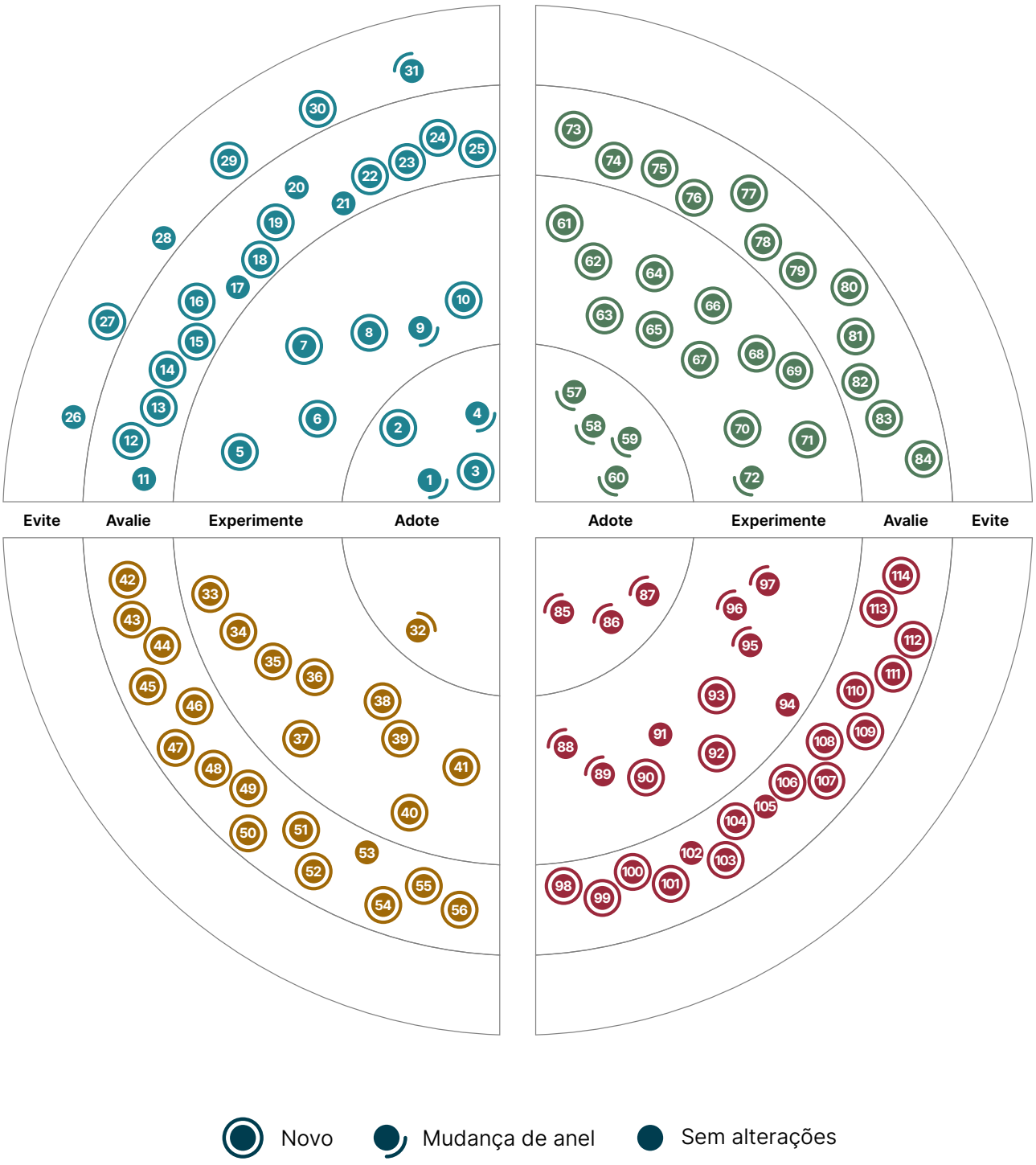
Também seguimos debatendo o desenvolvimento orientado por especificação — seu escopo, granularidade e potencial para atuar como uma única fonte da verdade na entrega incremental. No entanto, em meio ao entusiasmo, a complacência com o código gerado por IA permanece uma preocupação compartilhada, lembrando-nos de que, embora a IA possa acelerar a engenharia, o discernimento humano continua sendo indispensável.

## Antipadrões emergentes de IA

A adoção acelerada da IA em todos os setores tem revelado tanto práticas eficazes quanto anti-padrões emergentes. Embora vejamos uma utilidade clara em conceitos como a prototipagem de UI de autoatendimento e descartável com GenAI, também reconhecemos seu potencial para levar as organizações ao antipadrão de shadow IT acelerado por IA. Da mesma forma, à medida que o Model Context Protocol (MCP) ganha tração, muitos times estão sucumbindo ao antipadrão de conversão ingênua de API para MCP.

Também descobrimos que a eficácia das soluções de Text to SQL não atendeu às expectativas iniciais, e a complacência com o código gerado por IA continua a ser uma preocupação relevante. Mesmo dentro de práticas emergentes como o desenvolvimento guiado por especificação, notamos o risco de reverter a anti-padrões tradicionais da engenharia de software — mais notavelmente, um viés em direção à especificação pesada inicial e a entregas big-bang. Como a GenAI está avançando em um ritmo e escala sem precedentes, esperamos que novos antipadrões surjam rapidamente. Os times devem permanecer vigilantes para padrões que parecem eficazes no início, mas que, com o tempo, degradam e retardam o feedback, minam a adaptabilidade ou desfocam a responsabilidade.

# O Radar



# O Radar

## Técnicas

**Adote**

1. Conformidade contínua
2. Instruções compartilhadas e curadas para times de software
3. Hooks de pré-commit
4. Uso de IA generativa para entender bases de código legadas

**Experimente**

5. AGENTS.md
6. IA para migrações de código
7. Liquid Clustering do Delta Lake
8. Prototipagem de UI de autoatendimento com GenAI
9. Saída estruturada de LLMs
10. TCR (Test && Commit || revert)

**Avalie**

11. Testes de UI impulsionados por IA
12. Ancorando agentes de programação a uma aplicação de referência
13. Engenharia de contexto
14. IA generativa para engenharia direta
15. GraphQL como padrão de acesso a dados para LLMs
16. Fluxos de conhecimento em vez de estoques de conhecimento
17. LLM como juiz
18. Recuperação de informação no dispositivo
19. SAIF
20. Service mesh sem sidecar
21. Modelos de linguagem pequenos
22. Desenvolvimento orientado por especificação
23. Time de agentes de programação
24. Agendamento consciente de topologia
25. Análise de fluxo tóxico para IA

**Evite**

26. TI invisível acelerada por IA
27. Desenvolvimento guiado por capacidade
28. Complacência com código gerado por IA
29. Conversão ingênua de API para MCP
30. Times de engenharia de dados independentes
31. Text to SQL

## Plataformas

**Adote**

32. Arm na nuvem

**Experimente**

33. Apache Paimon
34. DataDog LLM Observability
35. Delta Sharing
36. Dovetail
37. Langdock
38. LangSmith
39. Model Context Protocol (MCP)
40. n8n
41. OpenThread

**Avalie**

42. Protocolo AG-UI
43. Protocolo agente-para-agente (A2A)
44. Amazon S3 Vectors
45. Ardoq
46. CloudNativePg
47. Coder
48. Graft
49. groundcover
50. Karmada
51. OpenFeature
52. Oxide
53. Restate
54. SkyPilot
55. StarRocks
56. Uncloud

**Evite**

—

# O Radar

## Ferramentas

**Adote**

57. ClickHouse
58. NeMo Guardrails
59. pnpm
60. Pydantic

**Experimente**

61. Revisor de design com IA
62. Barman
63. Claude Code
64. Cleanlab
65. Context7
66. Data Contract CLI
67. Databricks Assistant
68. Hoppscotch
69. NVIDIA DCGM explorer
70. Relational AI
71. UXPilot
72. v0

**Avalie**

73. Augment Code
74. Azure AI Document Intelligence
75. Docling
76. E2B
77. Editor Helix
78. Kueue
79. MCP-Scan
80. oRPC
81. Extensão Power user for dbt para VS Code
82. Serena
83. Sweetpad
84. Tape/Z Tools for Assembly Program Exploration for Z/OS

**Evite**

—

## Linguagens e Frameworks

**Adote**

85. Fastify
86. LangGraph
87. vLLM

**Experimente**

88. Crossplane
89. DeepEval
90. fastMCP
91. LiteLLM
92. MLForecast
93. Nuxt
94. Phoenix
95. Presidio
96. PydanticAI
97. Tauri

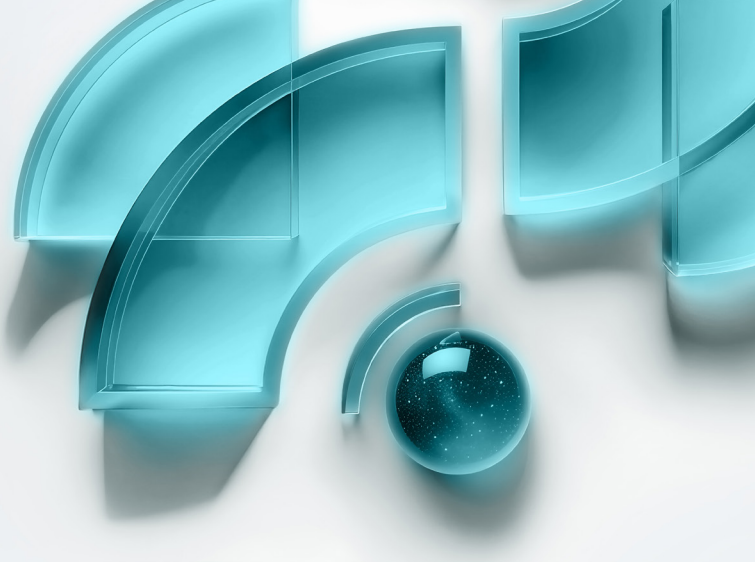
**Avalie**

98. Agent Development Kit (ADK)
99. Agno
100. assistant-ui
101. AutoRound
102. Browser Use
103. DeepSpeed
104. Drizzle
105. Criptografia pós-quântica em Java
106. kagent
107. LangExtract
108. Langflow
109. LMCache
110. Mem0
111. Linguagem de avaliação de controles de segurança abertos (OSCAL)
112. OpenInference
113. Valibot
114. Vercel AI SDK

**Evite**

—

# Técnicas



**Adote**

1. Conformidade contínua
2. Instruções compartilhadas e curadas para times de software
3. Hooks de pré-commit
4. Uso de IA generativa para entender bases de código legadas

**Experimente**

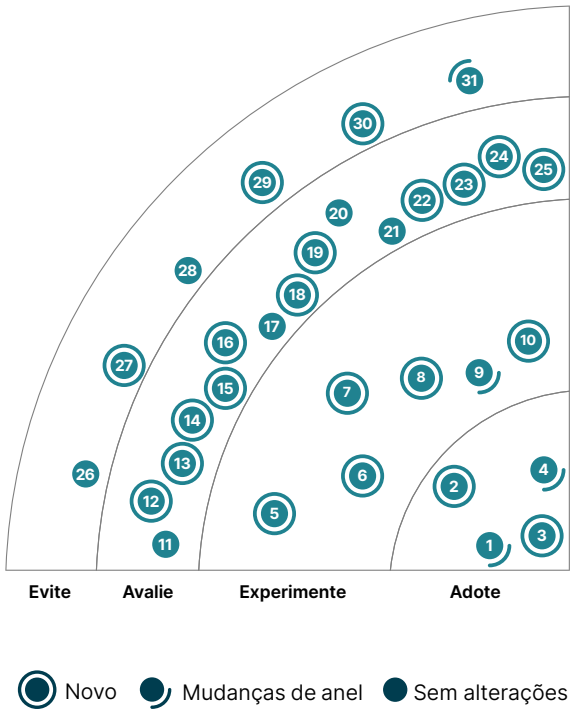
5. AGENTS.md
6. IA para migrações de código
7. Liquid Clustering do Delta Lake
8. Prototipagem de UI de autoatendimento com GenAI
9. Saída estruturada de LLMs
10. TCR (Test && Commit || revert)

**Avalie**

11. Testes de UI impulsionados por IA
12. Ancorando agentes de programação a uma aplicação de referência
13. Engenharia de contexto
14. IA generativa para engenharia direta
15. GraphQL como padrão de acesso a dados para LLMs
16. Fluxos de conhecimento em vez de estoques de conhecimento
17. LLM como juiz
18. Recuperação de informação no dispositivo
19. SAIF
20. Service mesh sem sidecar
21. Modelos de linguagem pequenos
22. Desenvolvimento orientado por especificação
23. Time de agentes de programação
24. Agendamento consciente de topologia
25. Análise de fluxo tóxico para IA

**Evite**

26. TI invisível acelerada por IA
27. Desenvolvimento guiado por capacidade
28. Complacência com código gerado por IA
29. Conversão ingênua de API para MCP
30. Times de engenharia de dados independentes
31. Text to SQL



## 1. Conformidade contínua

*Adote*

Conformidade contínua é a prática de garantir que os processos e tecnologias de desenvolvimento de software atendam aos padrões regulatórios e de segurança de forma contínua por meio da automação. As verificações manuais de conformidade podem retardar o desenvolvimento e introduzir erros humanos, enquanto as verificações e auditorias automatizadas fornecem feedback mais rápido, evidências mais claras e relatórios simplificados. Ao integrar ferramentas de política como código, como o Open Policy Agent, e ao gerar SBOMs dentro dos pipelines de CD — alinhados às diretrizes do SLSA — os times podem detectar e resolver problemas de conformidade antecipadamente. Programar regras e melhores práticas impõe padrões de forma consistente entre os times, sem criar gargalos. O OSCAL também se mostra promissor como um framework para automatizar a conformidade em escala. As práticas e ferramentas para conformidade contínua estão agora maduras o suficiente para que sejam tratadas como um padrão sensato, e é por isso que movemos nossa recomendação para Adote. O uso crescente de IA na programação — e o risco associado de complacência com o código gerado por IA — torna a incorporação da conformidade no processo de desenvolvimento mais crítica do que nunca.

## 2. Instruções compartilhadas e curadas para times de software

*Adote*

Para times que usam ativamente IA na entrega de software, o próximo passo é ir além do prompting individual em direção a instruções curadas para times de software. Essa prática ajuda a aplicar a IA de forma eficaz em todas as tarefas de entrega — não apenas na programação — compartilhando instruções comprovadas e de alta qualidade. A maneira mais direta de implementar isso é comitando arquivos de instrução, como um AGENTS.md, diretamente no repositório do seu projeto. A maioria das ferramentas de programação com IA — incluindo Cursor, Windsurf e Claude Code — suporta o compartilhamento de instruções por meio de comandos de barra ou workflows customizados. Para tarefas que não envolvem programação, você pode configurar bibliotecas de prompts em toda a organização, prontas para uso. Essa abordagem sistemática permite a melhoria contínua: assim que um prompt é refinado, todo o time se beneficia, garantindo acesso consistente às melhores instruções de IA.

## 3. Hooks de pré-commit

*Adote*

Os Git hooks existem há muito tempo, mas sentimos que ainda são subutilizados. O surgimento da programação assistida por IA e agentes aumentou o risco de comitar segredos ou código problemático acidentalmente. Embora existam muitos mecanismos para a validação de código, como a integração contínua, os hooks de pré-commit são uma salvaguarda simples e eficaz que mais times deveriam adotar. No entanto, sobrecarregar os hooks com verificações lentas pode desencorajar as pessoas desenvolvedoras de usá-los, então é melhor mantê-los mínimos e focados nos riscos que são mais eficazmente capturados nesta fase do workflow, como a varredura de segredos.

## 4. Uso de IA generativa para entender bases de código legadas

*Adote*

Nos últimos meses, vimos evidências claras de que o uso de IA generativa para entender bases de código legadas pode acelerar significativamente a compreensão de sistemas grandes e complexos. Ferramentas como Cursor, Claude Code, Copilot, Windsurf, Aider, Cody, Swimm, Unblocked e

PocketFlow-Tutorial-Codebase-Knowledge ajudam pessoas desenvolvedoras a revelarem regras de negócio, resumirem lógicas e identificarem dependências. Usadas em conjunto com frameworks abertos e prompting direto em LLMs, elas reduzem drasticamente o tempo necessário para entender bases de código legadas. Nossa experiência em múltiplos clientes mostra que a compreensão de sistemas legados assistida por IA generativa é agora um padrão prático, em vez de um experimento. O esforço de configuração varia, particularmente para abordagens avançadas como o GraphRAG, e tende a escalar com o tamanho e a complexidade da base de código sendo analisada. Apesar disso, o impacto na produtividade é consistente e substancial. A GenAI tornou-se uma parte essencial de como exploramos e entendemos sistemas legados.

## 5. AGENTS.md

***Experimente***

AGENTS.md é um formato comum para fornecer instruções a agentes de programação de IA que trabalham em um projeto. Essencialmente um arquivo README para agentes, ele não tem campos ou formatação obrigatórios além do Markdown, confiando na capacidade dos agentes de programação baseados em LLM de interpretar orientações escritas e legíveis por humanos. Usos típicos incluem dicas sobre o uso de ferramentas no ambiente de programação, instruções de teste e práticas preferenciais para o gerenciamento de commits. Embora as ferramentas de IA suportem vários métodos para a engenharia de contexto, o valor do AGENTS.md reside na criação de uma convenção simples para um arquivo que atua como ponto de partida.

## 6. IA para migrações de código

***Experimente***

Migrações de código assumem muitas formas — desde reescritas de linguagem até atualizações de dependência ou framework — e raramente são simples, muitas vezes exigindo meses de esforço manual. Um de nossos times, ao atualizar a versão do seu framework .NET, experimentou o uso de IA para encurtar o processo. No passado, publicamos sobre o OpenRewrite, uma ferramenta de refatoração determinística e baseada em regras. O uso isolado de IA para tais atualizações muitas vezes se mostrou caro e propenso a interações sinuosas. Em vez disso, a equipe combinou pipelines tradicionais de atualização com assistentes agênticos de programação para gerenciar transições complexas. Em vez de delegar uma atualização completa, eles dividiram o processo em passos menores e verificáveis: analisar erros de compilação, gerar diffs de migração e validar testes iterativamente. Essa abordagem híbrida posiciona os agentes de codificação com IA como colaboradores pragmáticos na manutenção de software. Exemplos da indústria, como a migração em grande escala de int32 para int64 do Google, refletem uma tendência semelhante. Embora nossos resultados sejam mistos em termos de economia de tempo mensurável, o potencial para reduzir o trabalho repetitivo da pessoa desenvolvedora é claro e vale a exploração contínua.

## 7. Liquid Clustering do Delta Lake

***Experimente***

Liquid Clustering é uma técnica para tabelas do Delta Lake que funciona como alternativa ao particionamento e ao Z-ordering. Historicamente, otimizar tabelas Delta para desempenho de leitura exigia a definição de chaves de partição e Z-order no momento da criação da tabela, com base em padrões de consulta esperados. Modificar essas chaves posteriormente exigia a reescrita completa dos dados. Em contraste, o Liquid Clustering utiliza um algoritmo baseado em árvore para agrupar os dados segundo chaves designadas, que podem ser alteradas incrementalmente sem precisar reescrever todos os dados. Isso proporciona maior flexibilidade para suportar diversos padrões

de consulta, reduzindo os custos de computação e melhorando o desempenho de leitura. Além disso, o Databricks Runtime para Delta Lake oferece suporte ao Automatic Liquid Clustering, que analisa workloads de consulta históricos, identifica colunas ideais e agrupa os dados de acordo. Tanto usuários do Delta Lake standalone quanto do Databricks Runtime podem aproveitar o Liquid Clustering para otimizar o desempenho de leitura.

## 8. Prototipagem de UI de autoatendimento com GenAI

***Experimente***

Usamos a frase prototipagem de UI de autoatendimento com GenAI para descrever uma técnica emergente onde ferramentas como o Claude Code, Figma Make, Miro AI e v0 permitem que gerentes de produto gerem protótipos interativos e testáveis por pessoas usuárias diretamente a partir de prompts de texto. Em vez de criar wireframes manualmente, os times podem gerar artefatos funcionais de HTML, CSS e JS em minutos — oferecendo a velocidade de um esboço, mas com interatividade real e maior fidelidade. Esses protótipos "descartáveis" trocam o polimento pelo aprendizado rápido, tornando-os ideais para validação inicial durante as design sprints. No entanto, uma maior fidelidade pode levar a um foco equivocado em pequenos detalhes ou a expectativas irreais sobre o esforço de produção — tornando o enquadramento claro e o gerenciamento de expectativas essenciais. Usada em conjunto com a pesquisa com pessoas usuárias, essa técnica acelera a descoberta, transformando ideias abstratas em experiências tangíveis para que as pessoas usuárias possam reagir. No entanto, os times devem ter o cuidado de não deixar que essas ferramentas substituam o processo de pesquisa em si. Quando bem-feita, a prototipagem de autoatendimento encurta os ciclos de feedback, diminui as barreiras para quem não é designer e ajuda os times a iterarem rapidamente, mantendo um equilíbrio saudável entre velocidade e qualidade.

## 9. Saída estruturada de LLMs

***Experimente***

Saída estruturada de LLMs é a prática de restringir um modelo de linguagem de grande porte (LLM) para gerar respostas em um formato predefinido — como JSON ou uma classe de programação específica. Essa técnica é essencial para construir aplicações confiáveis e de nível de produção, transformando o texto tipicamente imprevisível do LLM em um contrato de dados determinístico e legível por máquina. Com base no uso bem-sucedido em produção, estamos promovendo esta técnica da fase de Avaliação (Avalie) para a fase de Experimento (Experimente). As abordagens variam desde a formatação simples baseada em prompt e saídas estruturadas nativas do modelo até métodos de decodificação restrita mais robustos, usando ferramentas como Outlines e Instructor, que aplicam máquinas de estados finitos para garantir uma saída válida. Temos usado com sucesso essa técnica para extrair dados complexos e não estruturados de diversos tipos de documentos e convertê-los em JSON estruturado para a lógica de negócio downstream.

## 10. TCR (Test && Commit || revert)

***Experimente***

Test && Comitar || Revert (TCR) é um workflow de programação derivado do desenvolvimento orientado a testes que incentiva passos muito pequenos e contínuos por meio de uma regra simples: após cada alteração, se os testes passarem, as alterações são comitadas; se falharem, as alterações são revertidas. Implementar o TCR é simples: você só precisa definir um script que automatize esse ciclo em sua base de código. Originalmente introduzido em um artigo canônico de Kent Beck, descobrimos que o TCR reforça práticas de programação positivas como YAGNI e KISS. Vale a pena avaliá-lo à medida que experimentamos novos workflows para construir software usando GenAI.

## 11. Testes de UI impulsionados por IA

*Avalie*

No Radar anterior, os testes de UI impulsionados por IA focavam, principalmente, em testes exploratórios, onde notamos que o não-determinismo dos LLMs poderia introduzir instabilidade. Com o surgimento do MCP, agora estamos vendo os principais frameworks de teste de UI como Playwright e Selenium introduzirem seus próprios servidores MCP (playwright-mcp, mcp-selenium). Eles fornecem uma confiável automação de navegador por meio de suas tecnologias nativas, permitindo que os assistentes de programação gerem testes de UI confiáveis em Playwright ou Selenium. Embora os testes de UI impulsionados por IA permaneçam um espaço em rápida evolução, — a versão mais recente do Playwright, por exemplo, introduziu os Playwright Agents — estamos entusiasmadas com esses desenvolvimentos e ansiosas para ver mais orientações práticas e experiência de campo surgirem.

## 12. Ancorando agentes de programação a uma aplicação de referência

*Avalie*

No passado, publicamos sobre a técnica de modelos de serviço personalizados, que ajudou organizações na adoção de microsserviços, fornecendo padrões sensatos para o bootstrap de novos serviços e integrando-os de forma transparente com a infraestrutura existente. Com o tempo, no entanto, o desvio de código (code drift) entre esses templates e os serviços existentes tende a crescer à medida que novas dependências, frameworks e padrões arquitetônicos surgem. Para manter boas práticas e consistência arquitetônica — especialmente na era dos agentes de programação — temos experimentado ancorar agentes de programação a uma aplicação de referência. Esse padrão orienta os agentes de código generativo, fornecendo uma aplicação de referência viva e compilável em vez de exemplos de prompt estáticos. Um servidor de Model Context Protocol (MCP) expõe tanto o código do template de referência quanto os diffs de commits, permitindo que os agentes detectem desvios e proponham correções. Essa abordagem transforma templates estáticos em projetos vivos e adaptáveis que a IA pode referenciar de forma inteligente — mantendo a consistência, reduzindo a divergência e melhorando o controle sobre a geração de arquivos pela IA à medida que os sistemas evoluem.

## 13. Engenharia de contexto

*Avalie*

Engenharia de contexto é o design sistemático e a otimização da informação fornecida a um modelo de linguagem de grande porte durante a inferência para produzir de forma confiável a saída desejada. Ela envolve a estruturação, seleção e sequenciamento de elementos contextuais — como prompts, dados recuperados, memória, instruções e sinais ambientais — para que as camadas internas do modelo operem em um estado ótimo. Diferente da engenharia de prompt, que foca apenas na formulação dos prompts, a engenharia de contexto considera toda a configuração do contexto: como o conhecimento relevante, as instruções e o contexto prévio são organizados e entregues para alcançar os resultados mais eficazes. Hoje, as pessoas de engenharia usam uma gama de técnicas discretas que podem ser agrupadas em três áreas: a *configuração de contexto* abrange táticas de curadoria, como o uso de prompts de sistema mínimos, exemplos few-shot canônicos e ferramentas eficientes em tokens para ações decisivas. O gerenciamento de contexto para tarefas de longo prazo lida com janelas de contexto finitas por meio da sumarização de contexto, anotações estruturadas para persistir memórias externas e arquiteturas de sub-agentes para isolar e resumir subtarefas complexas. A recuperação dinâmica de informação depende da recuperação de contexto just-in-time (JIT), onde os agentes carregam autonomamente dados externos apenas quando são imediatamente relevantes, maximizando a eficiência e a precisão.

## 14. IA generativa para engenharia direta

*Avalie*

IA generativa para engenharia direta é uma técnica emergente para modernizar sistemas legados por meio de descrições de bases de código legadas geradas por IA. Ela introduz um passo explícito focado no *que* o código legado faz (sua especificação), enquanto oculta deliberadamente *como* ele está implementado atualmente. Isso se relaciona com o desenvolvimento guiado por especificação, mas é aplicado especificamente à modernização de sistemas legados. Ao gerar e iterar sobre descrições funcionais antes de reescrever o código, os times podem usar a IA generativa para revelar lógicas ocultas, dependências e casos excepcionais que, de outra forma, poderiam ser ignorados. Enfatizar o espaço do problema em vez do sistema existente também permite que os modelos de IA generativa explorem soluções mais criativas e escaláveis. O workflow segue um ciclo de engenharia reversa → design/solução → engenharia direta, o que permite que tanto humanos quanto agentes de IA raciocinem em um nível mais alto antes de se comprometerem com uma implementação. Na Thoughtworks, estamos vendo múltiplos times aplicarem essa abordagem com sucesso para acelerar a reescrita de sistemas legados. O objetivo não é ocultar completamente os detalhes da implementação, mas introduzir uma abstração temporária que ajuda os times e os agentes a explorarem alternativas sem serem limitados pela estrutura atual. Essa técnica está mostrando resultados promissores na produção de código mais limpo, de fácil manutenção e pronto para o futuro, ao mesmo tempo que reduz o tempo gasto para entender as implementações existentes.

## 15. GraphQL como padrão de acesso a dados para LLMs

*Avalie*

O GraphQL como padrão de acesso a dados para LLMs é uma abordagem emergente para criar uma camada de acesso a dados uniforme e amigável para modelos, que aprimora a engenharia de contexto. Ele permite que os times exponham dados estruturados e consultáveis sem conceder aos modelos acesso direto ao banco de dados. Diferente das APIs REST, que tendem a buscar dados em excesso ou a exigir novos endpoints ou filtros para cada caso de uso, o GraphQL permite que o modelo recupere apenas os dados de que precisa — reduzindo o ruído, melhorando a relevância do contexto e cortando o uso de tokens. Um schema GraphQL bem definido também fornece metadados que os LLMs podem usar para raciocinar sobre as entidades e os relacionamentos disponíveis, permitindo consultas dinâmicas e conscientes do schema para casos de uso com agentes. Esse padrão oferece um meio-termo seguro entre REST e SQL, equilibrando os controles de governança com o acesso flexível. No entanto, a abordagem depende de schemas bem estruturados e nomes de campo significativos. Interpretar a semântica do schema e navegar por estruturas complexas continuam sendo desafios — e o que é difícil para as pessoas raciocinarem, muitas vezes é igualmente difícil para os LLMs. Também é importante estar ciente dos vetores aumentados para ataques de DoS, bem como dos desafios típicos do GraphQL, como cache e versionamento.

## 16. Fluxos de conhecimento em vez de estoques de conhecimento

*Avalie*

Uma pergunta que nos fazem muito é “como podemos melhorar a forma como compartilhamos informações entre nossos times?” As técnicas para gerenciar o conhecimento organizacional continuam a evoluir, e uma perspectiva que consideramos valiosa vem do pensamento sistêmico: os conceitos de fluxos de conhecimento e estoques de conhecimento. Originalmente da economia, essa perspectiva incentiva os times a verem seu conhecimento organizacional como um sistema — estoques representando o conhecimento acumulado e fluxos representando como o conhecimento se move e evolui através da organização. Aumentar o fluxo de conhecimento externo para uma

organização tende a impulsionar a inovação. Uma maneira comprovada para melhorar o fluxo é estabelecer comunidades de prática, que consistentemente mostram benefícios mensuráveis. Outra é buscar deliberadamente fontes diversas e externas de insight. À medida que as ferramentas de GenAI tornam os estoques de conhecimento existentes mais acessíveis, vale a pena lembrar que nutrir novas ideias e perspectivas externas é tão crucial quanto adotar novas tecnologias.

## 17. LLM como juiz

***Avalie***

Usar um LLM como juiz — para avaliar a saída de outro sistema, geralmente um gerador baseado em LLM — tem chamado a atenção por seu potencial de oferecer avaliação automatizada e escalável em IA generativa. No entanto, estamos movendo este blip de Experimente para Avalie para refletir as complexidades e os riscos recém-reconhecidos. Embora essa técnica ofereça velocidade e escala, ela muitas vezes falha como um substituto confiável para o julgamento humano. As avaliações são propensas a viés de posição, viés de verbosidade e baixa robustez. Um problema mais sério é a contaminação de escala: quando o LLM como juiz é usado em pipelines de treinamento para modelagem de recompensa, ele pode introduzir um viés de auto-reforço — onde uma família de modelos favorece suas próprias saídas — e o vazamento de preferências, borrando a fronteira entre treinamento e teste. Essas falhas levaram a resultados super-ajustados que inflam as métricas de performance sem validade no mundo real. Há estudos que conduzem investigações mais rigorosas sobre esse padrão. Para combater essas falhas, estamos explorando técnicas aprimoradas, como o uso de LLMs como um júri (empregando múltiplos modelos para consenso) ou o raciocínio em cadeia de pensamento durante a avaliação. Embora esses métodos visam aumentar a confiabilidade, eles também aumentam o custo e a complexidade. Aconselhamos os times a tratar essa técnica com cautela — garantindo verificação humana, transparência e supervisão ética antes de incorporar juízes LLM em workflows críticos. A abordagem permanece poderosa, mas menos madura do que se acreditava anteriormente.

## 18. Recuperação de informação no dispositivo

***Avalie***

Recuperação de informação no dispositivo é uma técnica que permite que busca, consciência de contexto e geração aumentada por recuperação (RAG) rodem inteiramente nos dispositivos da pessoa usuária — mobile, desktop ou dispositivos de edge — priorizando a privacidade e a eficiência computacional. Ela combina um banco de dados local leve com um modelo otimizado para inferência no dispositivo. Uma implementação promissora une o sqlite-vec, uma extensão do SQLite que suporta busca vetorial dentro do banco de dados embarcado, com o EmbeddingGemma, um modelo de embedding de 300 milhões de parâmetros construído sobre a arquitetura Gemma 3. Otimizada para eficiência e ambientes com recursos restritos, essa combinação mantém os dados próximos ao edge, reduzindo a dependência de APIs na nuvem e melhorando a latência e a privacidade. Recomendamos que os times avaliem essa técnica para aplicações local-first e outros casos de uso onde a soberania de dados, a baixa latência e a privacidade são críticas.

## 19. SAIF

***Avalie***

SAIF (Secure AI Framework) é um framework desenvolvido pelo Google para fornecer um guia prático para o gerenciamento de riscos de segurança em IA. Ele aborda sistematicamente ameaças comuns, como envenenamento de dados (data poisoning) e injeção de prompt (prompt injection), por meio de um mapa de riscos claro e análise de componentes. O SAIF também fornece estratégias práticas

de mitigação para cada ameaça. Consideramos seu foco na evolução nos riscos da construção de sistemas com agentes especialmente oportuno e valioso. O SAIF oferece um playbook conciso e acionável que os times podem usar para fortalecer as práticas de segurança para o uso de LLMs e aplicações impulsionadas por IA.

## 20. Service mesh sem sidecar

***Avalie***

Como o custo e a complexidade operacional das malhas de serviço (service meshes) baseadas em sidecar persistem, estamos entusiasmadas em ver outra opção de malha de serviço sem sidecar surgir: o modo ambiente do Istio. O modo ambiente introduz uma arquitetura em camadas que separa as responsabilidades entre dois componentes principais: o proxy L4 por nó (ztunnel) e o proxy L7 por namespace (proxy Waypoint). O ztunnel garante que o tráfego L3 e L4 seja transportado de forma eficiente e segura. Ele potencializa o plano de dados ambiente buscando certificados para todas as identidades do nó e lida com o redirecionamento de tráfego de e para as cargas de trabalho habilitadas para o modo ambiente. O proxy Waypoint, um componente opcional do modo ambiente, permite recursos avançados do Istio, como gerenciamento de tráfego, segurança e observabilidade. Tivemos uma boa experiência com o modo ambiente em clusters de pequena escala e esperamos obter mais insights e melhores práticas em grande escala à medida que a adoção cresce.

## 21. Modelos de linguagem pequenos

***Avalie***

Observamos um progresso constante no desenvolvimento de modelos de linguagem pequenos (SLMs) ao longo de vários volumes do Technology Radar. Com o crescente interesse na construção de soluções agênticas, estamos vendo cada vez mais evidências de que os SLMs podem potencializar agentes de AI de forma eficiente. A maioria dos workflows agênticos atuais está focada em tarefas repetitivas e de escopo restrito que não exigem raciocínio avançado, tornando-os uma boa combinação para os SLMs. Os avanços contínuos em SLMs como Phi-3, SmolLM2 e DeepSeek sugerem que eles oferecem capacidade suficiente para essas tarefas — com os benefícios adicionais de menor custo, latência reduzida e menor consumo de recursos em comparação com os LLMs. Vale a pena considerar os SLMs como a escolha padrão para workflows agênticos, utilizando os LLMs maiores e mais intensivos em recursos apenas quando necessário.

## 22. Desenvolvimento orientado por especificação

***Avalie***

Desenvolvimento orientado por especificação é uma abordagem emergente para workflows de programação assistidos por IA. Embora a definição do termo ainda esteja evoluindo, ele geralmente se refere a fluxos de trabalho que começam com uma especificação funcional estruturada e, em seguida, avançam por várias etapas para dividi-la em partes menores, soluções e tarefas. A especificação pode assumir diferentes formas — um único documento, um conjunto de documentos ou artefatos estruturados que capturem diversos aspectos funcionais. Temos observado muitas pessoas desenvolvedoras adotarem esse estilo (inclusive criamos uma versão interna na Thoughtworks). Três ferramentas, em particular, vêm explorando interpretações distintas do desenvolvimento orientado por especificação. O Kiro, da Amazon, orienta as pessoas usuárias por três estágios do fluxo — requisitos, design e criação de tarefas. O spec-kit, do GitHub, segue um processo semelhante em três etapas, mas adiciona uma orquestração mais rica, prompts configuráveis e uma “constituição” que define princípios imutáveis que devem ser sempre seguidos. Já o Tessl Framework (ainda em beta privado em setembro de 2025) adota uma abordagem mais radical, na qual a própria especificação se torna o artefato principal, em vez do código. Consideramos esse espaço fascinante, embora os workflows

ainda sejam complexos e opinativos. Essas ferramentas se comportam de maneiras muito diferentes dependendo do tamanho e do tipo da tarefa; algumas geram arquivos de especificação longos e difíceis de revisar e, quando produzem PRDs ou histórias de usuário, nem sempre fica claro quem é o usuário final. Podemos estar reaprendendo uma lição amarga: que a elaboração manual de regras detalhadas para a IA, no fim das contas, não é escalável.

## 23. Time de agentes de programação

***Avalie***

Time de agentes de programação refere-se a uma técnica na qual uma pessoa desenvolvedora orquestra múltiplos agentes de programação de IA, cada um com um papel distinto — por exemplo, arquiteta ou arquiteto, especialista de back-end, tester — para colaborar em uma tarefa de desenvolvimento. Essa prática é suportada por ferramentas como Claude Code, Roo Code e Kilo Code, que permitem o uso de subagentes e múltiplos modos de operação. Com base no princípio comprovado de que atribuir papéis e personas específicos a LLMs melhora a qualidade da saída, a ideia é alcançar melhores resultados coordenando múltiplos agentes especializados para cada papel, em vez de depender de um único agente de propósito geral. O nível ideal de granularidade dos agentes ainda está sendo explorado, mas pode se estender além de simples mapeamentos um-para-um com os papéis tradicionais de um time. Essa abordagem representa uma mudança em direção a pipelines de desenvolvimento assistidos por IA, orquestrados e compostos por múltiplos passos.

## 24. Agendamento consciente de topologia

***Avalie***

GPUs e LPUs não são mais “apenas um monte de dispositivos”. Elas são redes de chips fortemente acoplados cujo desempenho depende de onde o trabalho é alocado. Por exemplo, sistemas em escala de rack como o NVL72 da Nvidia apresentam domínios NVLink de 72 GPUs que se comportam como um único acelerador massivo (pense em mais de 13 TB de VRAM compartilhada) apenas se o posicionamento mantiver os jobs dentro da mesma ilha de switch. Saltos entre ilhas transformam as operações coletivas no gargalo. O Groq, como outro exemplo, usa uma rede agendada por software em tempo de compilação, que assume a movimentação de dados determinística e cronometrada pelo compilador entre as placas. Quando os workloads são mal alocados ou agendados aleatoriamente, quebramos essas premissas e a previsibilidade. Além disso, as GPUs variam muito em seu desempenho, mesmo no mesmo datacenter e rack. Há uma demanda crescente para que os agendadores estejam cientes dessa variabilidade e posicionem os jobs em uma fatia da topologia que também leve essa variabilidade e o tipo de job em consideração ao agendar.

Agendadores ingênuos que ignoram a topologia de NVLink/PCIe/NIC e a variabilidade das GPUs espalharão aleatoriamente os jobs multi-GPU e destruirão o tempo de passo e a eficiência; o posicionamento e o agendamento conscientes de topologia corrigem isso e, geralmente, você terá dois tipos de workloads. Treinamento: (síncrono, limitado por largura de banda): Favorece ilhas contíguas com caminhos uniformes e de alta largura de banda para estágios de all-reduce e pipeline; co-agenda a largura de banda do fabric; evita saltos entre switches; trata os limites de link/switch/nó como domínios de falha. E favorece blocos de alta variabilidade de desempenho. Inferência: (limitada por latência/SLO, elástica): Geralmente escolhe entre replicação (espalhar réplicas entre blocos de disponibilidade para alta disponibilidade) e sharding (manter os shards/experts de MoE e a localidade do cache KV nos caminhos mais curtos). Otimiza o posicionamento de prefill vs. decode, micro-batching, isolamento de tenants, etc. Em suma, acreditamos que a dependência dos aceleradores na topologia da rede e do datacenter continuará a aumentar, e com isso, também o agendamento consciente de topologia. Estamos avaliando projetos como o Kueue e outros neste espaço para alcançar um agendamento de topologia de maior desempenho em nossos clientes.

## 25. Análise de fluxo tóxico para IA

*Avalie*

A piada agora familiar de que o S em MCP significa "segurança" esconde um problema muito real. Quando os agentes se comunicam uns com os outros — por meio da invocação de ferramentas ou chamadas de API — eles podem rapidamente encontrar o que ficou conhecido como a combinação letal (lethal trifecta): acesso a dados privados, exposição a conteúdo não confiável e a capacidade de se comunicar externamente. Agentes com os três atributos são altamente vulneráveis. Como os LLMs tendem a seguir as instruções em sua entrada, um conteúdo que inclua uma diretiva para exfiltrar dados para uma fonte não confiável pode facilmente levar a vazamentos de dados. Uma técnica emergente para mitigar esse risco é a análise de fluxo tóxico, que examina o grafo de fluxo de um sistema com agentes para identificar caminhos de dados potencialmente inseguros para investigação posterior. Embora ainda em seus estágios iniciais, a análise de fluxo tóxico representa uma das várias abordagens promissoras para detectar os novos vetores de ataque aos quais os sistemas com agentes e os servidores MCP estão cada vez mais expostos.

## 26. TI invisível acelerada por IA

*Evite*

A IA está diminuindo as barreiras para que pessoas que não programam construam e integrem software por conta própria, em vez de esperar que o departamento de TI atenda às suas necessidades. Embora estejamos entusiasmadas com o potencial que isso desbloqueia, também estamos atentas aos primeiros sinais da TI invisível acelerada por IA. Plataformas de automação de fluxo de trabalho sem código agora oferecem suporte à integração de API de IA (por exemplo, OpenAI ou Anthropic), tornando tentador usar a IA como tapa-buraco — unindo integrações que antes não eram possíveis, como transformar mensagens de chat em chamadas de API de ERP via IA. Ao mesmo tempo, assistentes de programação de IA estão se tornando mais "agênticos", ou autônomos, permitindo que pessoas com treinamento básico construam aplicativos de utilidade interna.

Isso tem todas as características da próxima evolução das planilhas que ainda alimentam processos críticos em algumas empresas — mas com uma pegada muito maior. Se não for controlada, essa nova TI invisível pode levar a uma proliferação de aplicativos não regulamentados e potencialmente inseguros, espalhando dados por cada vez mais sistemas. As organizações devem estar cientes desses riscos e avaliar cuidadosamente as compensações entre a resolução rápida de problemas e estabilidade a longo prazo.

## 27. Desenvolvimento guiado por capacidade

*Evite*

A chave para o sucesso das práticas modernas de desenvolvimento de software é manter o foco no fluxo de trabalho. Times alinhados, ao se concentrarem em um único e valioso fluxo — como uma jornada de pessoa usuária ou produto — conseguem entregar valor de ponta a ponta com eficiência. No entanto, estamos observando uma tendência preocupante de desenvolvimento guiado por capacidade, em que times com essa organização assumem features de outros produtos ou fluxos quando têm capacidade ociosa. Embora isso possa parecer eficiente no curto prazo, é uma otimização local mais adequada para lidar com picos repentinos de demanda. Quando normalizada, essa prática aumenta a carga cognitiva e a dívida técnica e, no pior dos casos, pode levar a um colapso por congestionamento, à medida que o custo da troca de contexto entre produtos aumenta. É mais vantajoso para times com capacidade ociosa focar em melhorar a saúde do sistema. Para

gerenciar a capacidade de forma eficaz, use limites de WIP para controlar o trabalho em fluxos adjacentes, considere cross-training para equilibrar os períodos de alta demanda e aplique técnicas de reformulação dinâmica de times quando necessário.

## 28. Complacência com código gerado por IA

***Evite***

À medida que os assistentes e agentes de programação de IA ganham tração, também aumenta o corpo de dados e pesquisas que destacam preocupações sobre a complacência com o código gerado por IA. Embora haja ampla evidência de que essas ferramentas podem acelerar o desenvolvimento — especialmente para prototipagem e projetos greenfield — estudos mostram que a qualidade do código pode diminuir com o tempo. A pesquisa de 2024 da GitClear constatou que o código duplicado e o code churn aumentaram mais do que o esperado, enquanto a atividade de refatoração nos históricos de commit diminuiu. Refletindo uma tendência semelhante, a pesquisa da Microsoft sobre trabalhadores do conhecimento mostra que a confiança impulsionada pela IA muitas vezes ocorre em detrimento do pensamento crítico — um padrão que observamos à medida que a complacência se instala com o uso prolongado de assistentes de programação. O surgimento de agentes de programação amplifica ainda mais esses riscos, já que a IA agora gera conjuntos de alterações maiores que são mais difíceis de revisar. Como em qualquer sistema, acelerar uma parte do workflow aumenta a pressão sobre as outras. Nossos times estão descobrindo que usar a IA de forma eficaz em produção requer um foco renovado na qualidade do código. Recomendamos reforçar práticas estabelecidas como TDD e análise estática, e incorporá-las diretamente nos workflows de programação, por exemplo, por meio de instruções compartilhadas e curadas para times de software.

## 29. Conversão ingênua de API para MCP

***Evite***

Organizações estão ansiosas para permitir que agentes de IA interajam com sistemas existentes, muitas vezes tentando uma conversão direta e transparente de APIs internas para o Model Context Protocol (MCP). Um número crescente de ferramentas, como o MCP link e o FastAPI-MCP, visa apoiar essa conversão. Desaconselhamos essa conversão ingênua de API para MCP. As APIs são tipicamente projetadas para pessoas desenvolvedoras humanas e muitas vezes consistem em ações granulares e atômicas que, quando encadeadas por uma IA, podem levar ao uso excessivo de tokens, poluição de contexto e baixo desempenho do agente. Além disso, essas APIs — especialmente as internas — frequentemente expõem dados sensíveis ou permitem operações destrutivas. Para pessoas desenvolvedoras humanas, tais riscos são mitigados por meio de padrões de arquitetura e revisões de código, mas quando as APIs são expostas ingenuamente a agentes via MCP, não há uma maneira confiável e determinística de impedir que um agente de IA autônomo use indevidamente tais endpoints. Recomendamos arquitetar um servidor MCP dedicado e seguro, especificamente adaptado para workflows com agentes, construído sobre suas APIs existentes.

## 30. Times de engenharia de dados independentes

***Evite***

Organizar times de engenharia de dados independentes para desenvolver e serem donos de pipelines e produtos de dados — separados dos domínios de negócio alinhados ao fluxo *(stream-aligned)* que atendem — é um antipadrão que leva a ineficiências e a resultados de negócio fracos. Essa estrutura repete erros do passado de isolar funções de DevOps, testes ou deployment, criando silos de conhecimento, gargalos e esforço desperdiçado. Sem uma colaboração próxima,

as pessoas de engenharia de dados frequentemente não têm o contexto de negócio e de domínio necessário para projetar produtos de dados significativos, limitando tanto a adoção quanto o valor. Em contrapartida, os times de plataforma de dados devem se concentrar na manutenção da infraestrutura compartilhada, enquanto os times de negócio multifuncionais constroem e são donos de seus produtos de dados, seguindo os princípios de data mesh. Colocamos essa prática em Evite para desencorajar fortemente os padrões organizacionais em silos — especialmente à medida que a necessidade de dados ricos em domínio e prontos para IA continua a crescer.

## 31. Text to SQL

***Evite***

O Text to SQL utiliza LLMs para converter linguagem natural em comandos SQL executáveis. No entanto, sua confiabilidade frequentemente fica aquém das expectativas. Movemos este *blip* para Evite, a fim de desencorajar seu uso em *workflows* não supervisionados — por exemplo, na conversão dinâmica de consultas geradas por pessoas usuárias quando a saída é oculta ou automatizada. Nesses casos, os LLMs costumam alucinar devido à compreensão limitada do schema ou do domínio, o que pode resultar na recuperação incorreta ou na modificação não intencional de dados. Além disso, a natureza não determinística das saídas dos LLMs torna a depuração e a auditoria de erros ainda mais desafiadoras.

Recomendamos tratar o uso de Text to SQL com cautela e sempre exigir revisão humana das consultas geradas. Para cenários de *business intelligence* agêntico, evite o acesso direto ao banco de dados e prefira uma camada semântica de abstração de dados governada — como a camada semântica do Cube ou do dbt — ou uma camada de acesso semanticamente rica, como o GraphQL ou o MCP.

# Plataformas

**Adote**

32. Arm na nuvem

**Experimente**

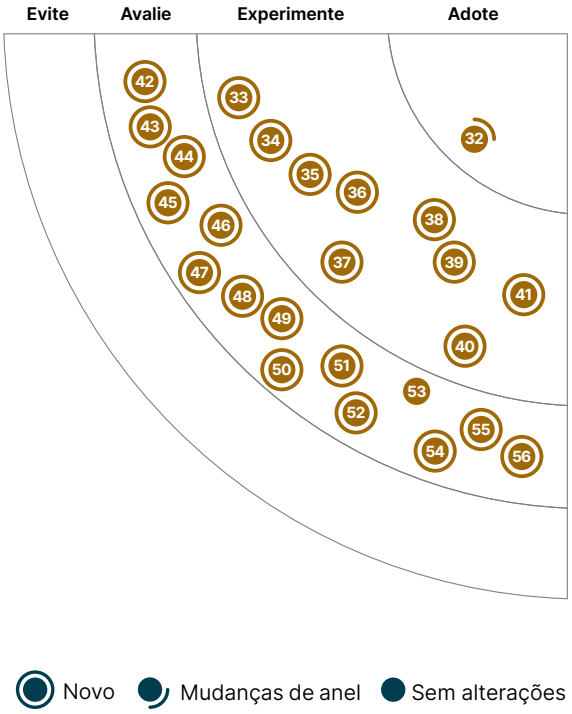
33. Apache Paimon
34. DataDog LLM Observability
35. Delta Sharing
36. Dovetail
37. Langdock
38. LangSmith
39. Model Context Protocol (MCP)
40. n8n
41. OpenThread

**Avalie**

42. Protocolo AG-UI
43. Protocolo agente-para-agente (A2A)
44. Amazon S3 Vectors
45. Ardoq
46. CloudNativePg
47. Coder
48. Graft
49. groundcover
50. Karmada
51. OpenFeature
52. Oxide
53. Restate
54. SkyPilot
55. StarRocks
56. Uncloud

**Evite**

—



## 32. Arm na nuvem

*Adote*

As instâncias de computação Arm na nuvem tornaram-se cada vez mais populares nos últimos anos devido ao seu menor custo e maior eficiência energética em comparação às instâncias tradicionais baseadas em x86. Os principais provedores de nuvem — incluindo AWS, Azure e GCP — agora oferecem opções robustas baseadas em Arm. Essas instâncias são especialmente atraentes para workloads de grande escala ou sensíveis a custos. Muitos de nossos times migraram com sucesso workloads como microsserviços, bancos de dados de código aberto e até mesmo aplicações de computação de alta performance para Arm, exigindo apenas alterações mínimas de código e pequenos ajustes em scripts de build. Novas aplicações e sistemas nativos da nuvem estão cada vez mais adotando Arm na nuvem como padrão. Com base em nossa experiência, recomendamos instâncias de computação Arm para a maioria dos workloads, a menos que existam dependências específicas de arquitetura. Ferramentas modernas, como as imagens Docker multi-arquitetura, tornam ainda mais simples o processo de build e deploy em ambientes Arm e x86.

## 33. Apache Paimon

*Experimente*

Apache Paimon é um formato de data lake de código aberto projetado para viabilizar a arquitetura lakehouse. Ele se integra perfeitamente com mecanismos de processamento como Flink e Spark, suportando tanto operações de streaming quanto em lote (batch). Uma vantagem fundamental da arquitetura do Paimon está na fusão de um formato de data lake padrão com uma estrutura LSM (log-structured merge-tree). Essa combinação resolve os desafios tradicionais de atualizações de alta performance e leituras de baixa latência em data lakes. O Paimon suporta tabelas de chave primária para atualizações em tempo real e de alta vazão, e inclui um motor de merge customizável para deduplicação, atualizações parciais e agregações. Esse design permite a ingestão eficiente de dados de streaming e o gerenciamento de estado mutável diretamente no lake. O Paimon também oferece recursos maduros de data lake, como metadados escaláveis, transações ACID, time travel, evolução de schema e layouts de dados otimizados por meio de compressão e Z-ordering. Recomendamos avaliar o Paimon para projetos que precisem de uma camada de armazenamento unificada, capaz de lidar eficientemente com dados append-only em grande escala e atualizações complexas de streaming em tempo real.

## 34. DataDog LLM Observability

*Experimente*

Datadog LLM Observability oferece rastreamento de ponta a ponta, monitoramento e diagnósticos para modelos de linguagem de grande porte e workflows de aplicações agênticas. Ele mapeia cada comando, chamada de ferramenta e passo intermediário em segmentos e traços; rastreia latência, uso de tokens, erros e métricas de qualidade; e se integra com a suíte mais ampla de APM e observabilidade do Datadog. Organizações que já utilizam o Datadog — e estão familiarizadas com sua estrutura de custos — podem achar a ferramenta de observabilidade de LLM uma maneira direta de ganhar visibilidade sobre workloads de IA, presumindo que esses workloads possam ser instrumentados. No entanto, configurar e usar a instrumentação de LLM exige cuidado e um entendimento sólido tanto dos workloads quanto de sua implementação. Recomendamos que as pessoas de engenharia de dados e de operações colaborem de perto ao implantá-lo. Veja também nosso conselho sobre evitar times de engenharia de dados separados.

## 35. Delta Sharing

*Experimente*

Delta Sharing é um padrão e protocolo aberto para o compartilhamento seguro e multiplataforma de dados, desenvolvido pela Databricks e pela Linux Foundation. Ele é agnóstico à nuvem, permitindo que as organizações compartilhem dados em tempo real entre provedores de nuvem e locais on-premise sem copiar ou replicar os dados — preservando a atualidade dos dados e eliminando custos de duplicação. Vimos uma empresa de e-commerce usar o Delta Sharing com sucesso para substituir um sistema fragmentado de compartilhamento de dados com parceiros por uma plataforma centralizada, segura e que funciona em tempo real, melhorando significativamente a colaboração. O protocolo usa uma API REST simples para emitir URLs pré-assinadas de curta duração, permitindo que os destinatários recuperem grandes conjuntos de dados usando ferramentas como pandas, Spark ou Power BI. Ele suporta o compartilhamento de tabelas de dados, views, modelos de IA e notebooks. Embora forneça uma governança e auditoria centralizadas fortes, as pessoas usuárias devem permanecer atentas aos custos de saída de dados da nuvem, que podem se tornar um risco operacional significativo se não forem gerenciados.

## 36. Dovetail

*Experimente*

A plataforma aborda o desafio persistente de gerenciar dados fragmentados de pesquisas qualitativas. Ela fornece um repositório centralizado para entrevistas com pessoas usuárias, transcrições e insights, transformando dados brutos em um ativo estruturado e analisável. Nós a consideramos extremamente valiosa em fluxos de trabalho de descoberta de produto, especialmente para criar uma trilha de evidências que conecta citações de clientes e temas sintetizados diretamente às hipóteses de produto e ao ROI estimado. Ao fazer isso, o Dovetail fortalece o papel dos dados qualitativos na tomada de decisão de produto.

## 37. Langdock

*Experimente*

Langdock é uma plataforma para que organizações desenvolvam e executem agentes e workflows de IA generativa para operações internas. Ela fornece um ambiente unificado com assistentes de chat internos, uma camada de API para conexão com múltiplos LLMs e ferramentas para construir workflows com agentes que se integram a sistemas como Slack, Confluence e Google Drive. A plataforma destaca a soberania de dados, oferecendo opções on-premise e hospedadas na União Europeia com padrões de conformidade empresariais. As organizações que implantam o Langdock ainda devem prestar muita atenção à governança de dados e usar técnicas como a análise de fluxo tóxico para evitar a combinação letal (lethal trifecta). Quem o adota também deve considerar a maturidade da plataforma, avaliar as integrações específicas requeridas por eles e planejar qualquer desenvolvimento customizado que possa ser necessário.

## 38. LangSmith

*Experimente*

LangSmith é uma plataforma hospedada do time do LangChain que oferece observabilidade, tracing e avaliação para aplicações de LLM. Ela captura rastreamentos detalhados de chains, ferramentas e prompts, permitindo que os times façam o debug e meçam o comportamento do modelo, monitorem regressões de performance e gerenciem conjuntos de dados de avaliação. O LangSmith é um serviço

SaaS proprietário com suporte limitado para workflows que não são do LangChain, o que o torna mais atraente para times que já investiram nesse ecossistema. Seu suporte integrado para avaliação e experimentação de prompts é notavelmente mais polido do que o de alternativas de código aberto como o Langfuse.

## 39. Model Context Protocol (MCP)

***Experimente***

O Model Context Protocol (MCP) é um padrão aberto que define como as aplicações e agentes de LLM se integram com fontes de dados e ferramentas externas, melhorando significativamente a qualidade dos resultados gerados por IA. O MCP foca no acesso a contexto e ferramentas, o que o torna diferente do protocolo Agent2Agent (A2A), que governa a comunicação entre agentes. Ele especifica servidores (para dados e ferramentas como bancos de dados, wikis e serviços) e clientes (agentes, aplicações e assistentes de programação). Desde nosso último blip, a adoção do MCP aumentou, com grandes empresas como JetBrains (IntelliJ) e Apple se juntando ao ecossistema, ao lado de frameworks emergentes como o FastMCP. Um padrão de preview do Registro MCP agora suporta a descoberta de ferramentas públicas e proprietárias. No entanto, a rápida evolução do MCP também revelou algumas lacunas arquiteturais, atraindo críticas por ignorar as melhores práticas estabelecidas de RPC (Chamada de Procedimento Remoto). Para aplicações em produção, os times devem olhar além do hype e aplicar um escrutínio adicional, mitigando fluxos tóxicos usando ferramentas como o MCP-Scan e monitorando de perto o rascunho do módulo de autorização para segurança.

## 40. n8n

***Experimente***

n8n é uma plataforma de automação de workflow com licença fair-code, semelhante ao Zapier ou Make (anteriormente Integromat), mas construída para pessoas desenvolvedoras que desejam uma opção auto-hospedada, extensível e controlável por código. Ela oferece low-code e uma abordagem mais visual para a criação de workflows em comparação com o Apache Airflow, enquanto ainda suporta código customizado em JavaScript ou Python. Seu principal caso de uso é a integração de múltiplos serviços em workflows automatizados, mas também pode conectar LLMs com fontes de dados, memória e ferramentas configuráveis. Muitos de nossos times usam o n8n para prototipar rapidamente workflows agênticos acionados por aplicações de chat ou webhooks, muitas vezes aproveitando suas capacidades de importação e exportação para gerar workflows com a assistência de IA. Como sempre, aconselhamos cautela ao usar plataformas de baixo código (low-code) em produção. No entanto, os workflows auto-hospedados e definidos por código do n8n podem mitigar alguns desses riscos.

## 41. OpenThread

***Experimente***

OpenThread é uma implementação de código aberto do protocolo de rede Thread desenvolvida pelo Google. Ele suporta todas as features principais da especificação Thread — incluindo camadas de rede como IPv6, 6LoWPAN e LR-WPAN — bem como capacidades de rede mesh que permitem que um dispositivo funcione tanto como um nó quanto como um roteador de borda. O OpenThread roda em uma ampla gama de plataformas de hardware, aproveitando uma camada de abstração flexível e hooks de integração que permitem aos fornecedores incorporar suas próprias capacidades de rádio e criptografia. Este protocolo maduro é amplamente utilizado em produtos comerciais e, em nossa experiência, provou ser confiável para a construção de diversas soluções de IoT — desde dispositivos de baixa potência operados por bateria até redes de sensores mesh em grande escala.

## 42. Protocolo AG-UI

***Avalie***

AG-UI é um protocolo aberto e uma biblioteca projetados para padronizar a comunicação entre interfaces de usuário ricas e agentes. Focado em agentes que interagem diretamente com a pessoa usuária, ele utiliza middleware e integrações de cliente para se generalizar entre qualquer frontend e backend. O protocolo define uma maneira consistente para que agentes de backend se comuniquem com aplicações de frontend, permitindo uma colaboração stateful e em tempo real entre a IA e pessoas usuárias humanas. Ele suporta múltiplos protocolos de transporte, incluindo SSE e WebSockets, e fornece tipos de eventos padronizados para representar diferentes estados de execução do agente. Há suporte nativo para frameworks de agentes populares, como LangGraph e Pydantic AI, com integrações da comunidade para outros.

## 43. Protocolo agente-para-agente (A2A)

***Avalie***

Agent2Agent (A2A) é um protocolo que define um padrão para comunicação e interação entre agentes em workflows complexos e multiagente. Ele utiliza *Agent Card*s para descrever os elementos-chave da comunicação entre agentes, incluindo a descoberta de habilidades e a definição dos esquemas de transporte e segurança. O A2A complementa o Model Context Protocol (MCP), concentrando-se na comunicação entre agentes sem expor detalhes internos, como estado, memória ou funcionamento interno de cada agente. O protocolo incentiva boas práticas, como uma abordagem *asynchronous-first* para tarefas de longa duração, respostas via streaming para atualizações incrementais e transporte seguro com HTTPS, autenticação e autorização. SDKs estão disponíveis em Python, JavaScript, Java e C#, facilitando a adoção rápida. Embora ainda recente, o A2A possibilita que equipes construam agentes de domínio específico capazes de colaborar para formar workflows complexos — tornando-o uma opção promissora para esse tipo de cenário.

## 44. Amazon S3 Vectors

***Avalie***

O Amazon S3 Vectors estende o armazenamento de objetos do S3 com capacidades vetoriais nativas, oferecendo armazenamento e busca por similaridade de vetores integrados. Ele se integra de forma transparente com o ecossistema da AWS, incluindo o Amazon Bedrock e o OpenSearch, e fornece recursos adicionais como filtragem de metadados e governança via IAM. Embora ainda esteja em prévia e sujeito a restrições e limitações, consideramos sua proposta de valor convincente. Essa abordagem acessível e de custo-benefício para o armazenamento de vetores poderia viabilizar uma gama de aplicações que envolvem grandes volumes de dados e onde a baixa latência não é a principal preocupação.

## 45. Ardoq

***Avalie***

Ardoq é uma plataforma de arquitetura corporativa (EA) que permite às organizações construir, gerenciar e escalar suas bases de conhecimento de arquitetura para que possam planejar o futuro com mais eficácia. Diferente da documentação estática tradicional, que é propensa a se desatualizar e a ficar isolada, a abordagem orientada a dados do Ardoq extrai informações de sistemas existentes para criar um grafo de conhecimento dinâmico que se mantém atualizado à medida que o cenário

evolui. Uma feature que achamos particularmente útil é o Ardoq Scenarios, que permite modelar e definir visualmente estados futuros hipotéticos *what-if* usando uma abordagem de branching e merging semelhante à do Git. Organizações que buscam uma transformação arquitetônica devem avaliar plataformas de EA dedicadas como o Ardoq pelo seu potencial de otimizar e acelerar esse processo.

## 46. CloudNativePg

***Avalie***

CloudNativePG é um Kubernetes Operator que simplifica a hospedagem e o gerenciamento de clusters PostgreSQL de alta disponibilidade no Kubernetes. Rodar um serviço stateful como o PostgreSQL no Kubernetes pode ser complexo, exigindo conhecimento profundo tanto do Kubernetes quanto da replicação do PostgreSQL. O CloudNativePG abstrai grande parte dessa complexidade ao tratar todo o cluster PostgreSQL como um único recurso declarativo e configurável. Ele fornece uma arquitetura primário/standby transparente, usando replicação via streaming nativa, e inclui features de alta disponibilidade prontas para uso, incluindo capacidades de autorrecuperação, failover automatizado que promove a réplica mais alinhada e recriação automática de réplicas que falharam. Se você está procurando hospedar o PostgreSQL no Kubernetes, o CloudNativePG é um ponto de partida sólido.

## 47. Coder

***Avalie***

Coder É uma plataforma para provisionar rapidamente ambientes de programação padronizados, seguindo a prática de ambientes de desenvolvimento na nuvem que descrevemos anteriormente. Em comparação com ferramentas similares como o Gitpod (agora rebatizado como Ona) e o GitHub Codespaces, o Coder oferece maior controle sobre a customização da estação de trabalho por meio do Terraform. Ele hospeda as estações de trabalho em sua própria infraestrutura, seja na nuvem ou em um data center, em vez de nos servidores de um fornecedor. Essa abordagem proporciona mais flexibilidade, incluindo a capacidade de executar agentes de programação de IA e acessar sistemas organizacionais internos. No entanto, essa flexibilidade vem com contrapartidas: mais esforço para configurar e manter os templates das estações de trabalho e maior responsabilidade no gerenciamento dos riscos de segurança de dados em workflows com agentes.

## 48. Graft

***Avalie***

Graft é um motor de armazenamento transacional que permite a sincronização de dados fortemente consistente e eficiente em ambientes de edge e distribuídos. Ele consegue isso usando replicação preguiçosa para sincronizar dados apenas sob demanda, replicação parcial para minimizar o consumo de largura de banda e isolamento de snapshot serializável para garantir a integridade dos dados. Mencionamos o Electric no Radar para um caso de uso semelhante, mas vemos o Graft como único ao transformar o armazenamento de objetos em um sistema transacional que suporta atualizações consistentes em nível de página sem impor um formato de dados. Isso o torna bem adequado para potencializar aplicações móveis local-first, gerenciar sincronizações complexas multiplataforma e servir como a espinha dorsal para réplicas stateless em sistemas serverless ou embarcados.

## 49. groundcover

*Avalie*

groundcover é uma plataforma de observabilidade cloud-native que unifica logs, traces, métricas e eventos do Kubernetes em um painel unificado. Ele utiliza o eBPF para capturar dados granulares de observabilidade sem instrumentação de código — ou seja, sem inserir agentes ou SDKs no código da aplicação. O sensor eBPF do groundcover roda em um nó dedicado em cada cluster monitorado, operando de forma independente das aplicações que observa. As features principais incluem visibilidade profunda em nível de kernel, uma arquitetura traga-sua-própria-nuvem (BYOC) para privacidade de dados e um modelo de precificação agnóstico ao volume de dados que mantém os custos previsíveis.

## 50. Karmada

*Avalie*

Karmada ("Kubernetes Armada") é uma plataforma para orquestrar workloads em múltiplos clusters Kubernetes, nuvens e data centers. Atualmente, muitos times fazem deploy em vários clusters usando ferramentas de GitOps como Flux ou ArgoCD combinadas com scripts customizados, portanto, uma solução de propósito específico é bem-vinda. O Karmada utiliza APIs nativas do Kubernetes, sem exigir alterações em aplicações já desenvolvidas para ambientes cloud-native. Ele oferece capacidades avançadas de agendamento para gerenciamento multi-cloud, alta disponibilidade, recuperação de falhas e roteamento de tráfego. O Karmada ainda é relativamente novo, então é importante avaliar a maturidade dos recursos dos quais seu time depende. No entanto, como um projeto da CNCF, ele tem um forte impulso, e vários de nossos times já o utilizam com sucesso. Vale notar que certas áreas — como rede, gerenciamento de estado e armazenamento entre clusters — estão fora do escopo do Karmada. A maioria dos times ainda precisará de uma service mesh como Istio ou Linkerd para lidar com tráfego e deve planejar como gerenciar workloads stateful e dados distribuídos.

## 51. OpenFeature

*Avalie*

À medida que as empresas escalam, o gerenciamento de feature flags muitas vezes se torna cada vez mais complexo; os times precisam de uma camada de abstração que vá além do feature toggle mais simples possível. OpenFeature fornece essa camada por meio de uma especificação de API independente de fornecedor e orientada pela comunidade, que padroniza como as feature flags são definidas e consumidas, desacoplando o código da aplicação da solução de gerenciamento. Essa flexibilidade permite que os times troquem de provedores facilmente — desde configurações básicas usando variáveis de ambiente ou configurações em memória, até plataformas maduras como ConfigCat ou LaunchDarkly. No entanto, uma ressalva crítica permanece: os times devem gerenciar diferentes categorias de flags separadamente e com disciplina para evitar a proliferação de flags, a complexidade da aplicação e a sobrecarga excessiva de testes.

## 52. Oxide

*Avalie*

Construir e operar infraestrutura privada é complexo. Essa é uma das principais razões pelas quais a nuvem pública é o padrão para a maioria das organizações. No entanto, para aquelas que precisam, a Oxide oferece uma alternativa à montagem e integração de hardware e software do zero. Ela fornece racks pré-construídos com computação, rede e armazenamento, rodando um software de

sistema totalmente integrado. Os times podem gerenciar os recursos por meio das APIs IaaS da Oxide usando Terraform e outras ferramentas de automação — o que a Oxide chama de infraestrutura elástica on-premises. A VxRail da Dell e VMware, a Nutanix e a HPE SimpliVity também fornecem soluções de infraestrutura hiperconvergente (HCI), mas o que distingue a Oxide é sua abordagem de propósito específico. Ela projeta toda a stack — desde as placas de circuito e as fontes de alimentação até o firmware — em vez de montar componentes de diferentes fornecedores. A Oxide também desenvolveu e tornou de código aberto o Hubris, um kernel leve, com memória protegida e de passagem de mensagens, escrito em Rust para sistemas embarcados, juntamente com outros projetos de infraestrutura baseados em Rust. Também apreciamos que a Oxide venda seus equipamentos e software sem taxas de licença.

## 53. Restate

***Avalie***

Restate é uma plataforma de execução durável projetada para lidar com desafios complexos de sistemas distribuídos na construção de aplicações stateful e tolerantes a falhas. Ela registra cada passo por meio de journaling de execução, garantindo tolerância a falhas, recuperação confiável e comunicação exactly-once entre os serviços. A principal vantagem arquitetônica da plataforma reside na separação da lógica da aplicação em três tipos de serviços duráveis: Serviços Básicos para funções stateless; Objetos Virtuais para modelar entidades concorrentes e stateful; e Workflows para orquestrar processos complexos e de múltiplos passos. Temos avaliado cuidadosamente o Restate em um grande sistema de seguros e estamos bastante satisfeitas com sua performance até agora.

## 54. SkyPilot

***Avalie***

SkyPilot É uma plataforma de código aberto para executar e escalar workloads de IA on-premises ou na nuvem. Desenvolvido pelo Sky Computing Lab na UC Berkeley, o SkyPilot atua como um intermediário inteligente que encontra e provisiona automaticamente as GPUs mais baratas e disponíveis nas principais nuvens e clusters Kubernetes, muitas vezes cortando custos de computação. Para os times de infraestrutura, ele simplifica a execução de IA no Kubernetes, oferecendo uma facilidade de uso semelhante ao Slurm, robustez nativa da nuvem, acesso SSH direto aos pods e features como gang scheduling e suporte multi-cluster para escalar de forma transparente os workloads de treinamento ou inferência.

## 55. StarRocks

***Avalie***

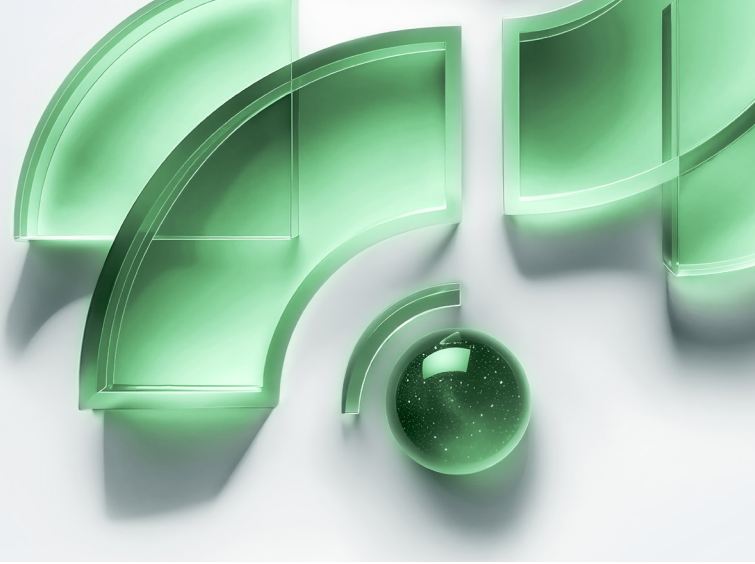
StarRocks é um banco de dados analítico que redefine o business intelligence em tempo real. Ele combina a velocidade dos sistemas OLAP tradicionais com a flexibilidade de uma data lakehouse moderna. Ele alcança latência de consulta abaixo de um segundo em escala massiva por meio de um motor de execução otimizado para SIMD, armazenamento colunar e um otimizador sofisticado baseado em custo. Essa arquitetura de alta performance permite que as pessoas usuárias executem análises complexas diretamente em formatos de dados abertos como o Apache Iceberg, sem pré-computação ou cópia de dados. Embora existam muitas plataformas nesta área, vemos o StarRocks como um forte candidato para soluções de custo-benefício que exigem tanto concorrência extrema quanto dados consistentemente atualizados ao segundo.

## 56. Uncloud

*Avalie*

Uncloud é uma ferramenta leve de orquestração de contêineres e clustering que permite às pessoas desenvolvedoras levarem aplicações do Docker Compose para a produção, oferecendo uma experiência simples, semelhante à da nuvem, sem o overhead operacional do Kubernetes. Ele alcança o escalonamento entre máquinas e deploys sem tempo de inatividade ao configurar automaticamente uma rede mesh segura com WireGuard para comunicação e ao usar o proxy reverso Caddy para fornecer HTTPS automático e balanceamento de carga. A principal vantagem arquitetônica do Uncloud é seu design totalmente descentralizado, que elimina a necessidade de um plano de controle central e garante que as operações do cluster permaneçam funcionais mesmo que máquinas individuais fiquem offline. Com o Uncloud, você pode misturar e combinar livremente VMs da nuvem e servidores bare-metal em um ambiente de computação unificado e de custo-benefício.

# Ferramentas



**Adote**

57. ClickHouse
58. NeMo Guardrails
59. pnpm
60. Pydantic

**Experimente**

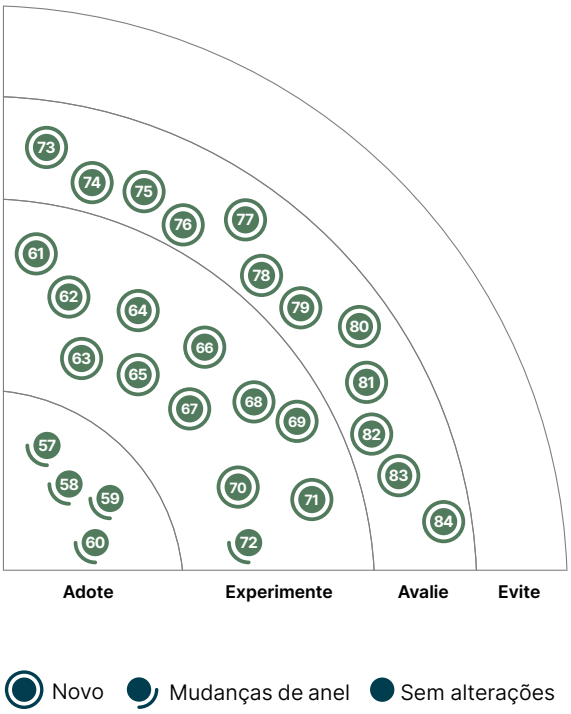
61. Revisor de design com IA
62. Barman
63. Claude Code
64. Cleanlab
65. Context7
66. Data Contract CLI
67. Databricks Assistant
68. Hoppscotch
69. NVIDIA DCGM explorer
70. Relational AI
71. UXPilot
72. v0

**Avalie**

73. Augment Code
74. Azure AI Document Intelligence
75. Docling
76. E2B
77. Editor Helix
78. Kueue
79. MCP-Scan
80. oRPC
81. Extensão Power user for dbt para VS Code
82. Serena
83. Sweetpad
84. Tape/Z Tools for Assembly Program Exploration for Z/OS

**Evite**

—




Novo  Mudanças de anel  Sem alterações

## 57. ClickHouse

*Adote*

ClickHouse é um banco de dados de processamento analítico online (OLAP) colunar, distribuído e de código aberto, voltado para análises em tempo real. Ele amadureceu e se tornou um mecanismo altamente performático e escalável, capaz de lidar com a análise de dados em grande escala. Sua visão materializada incremental, motor de consulta eficiente e forte compressão de dados o tornam ideal para consultas interativas. O suporte nativo para funções de agregação aproximadas permite equilibrar precisão e performance, o que é especialmente útil para análises de alta cardinalidade. A adição do motor de armazenamento S3 e do MergeTree permite a separação de armazenamento e computação, usando armazenamento compatível com S3 para as tabelas do ClickHouse. Também descobrimos que o ClickHouse é uma excelente base para dados do OpenTelemetry e ferramentas de análise de falhas como o Sentry. Para times que buscam um motor de analytics rápido e de código aberto, o ClickHouse é uma excelente escolha.

## 58. NeMo Guardrails

*Adote*

NeMo Guardrails é um toolkit de código aberto da NVIDIA que facilita a adição de mecanismos programáveis de segurança e controle a aplicações conversacionais baseadas em LLM. Ele garante que as saídas permaneçam seguras, dentro do tópico e em conformidade, definindo e aplicando regras de comportamento. Pessoas desenvolvedoras usam Colang, uma linguagem de domínio específico, para criar fluxos de diálogo flexíveis e gerenciar conversas, impondo caminhos e procedimentos operacionais predefinidos. O NeMo Guardrails também fornece uma API asynchronous-first para performance e suporta mecanismos para segurança de conteúdo, proteção e moderação de entradas e saídas. Estamos vendo uma adoção constante em times que constroem aplicações que vão desde chatbots simples até workflows agênticos complexos. Com seu conjunto de features em expansão e a cobertura cada vez mais madura de vulnerabilidades comuns de LLMs, estamos movendo o NeMo Guardrails para Adote.

## 59. pnpm

*Adote*

Desde o último Radar, continuamos a receber feedback positivo sobre o pnpm dos times. O pnpm é um gestor de pacotes para Node.js que oferece melhorias significativas de performance em relação às alternativas, tanto em velocidade quanto em eficiência de espaço em disco. Ele cria hard-links de pacotes duplicados das pastas `node_modules` de múltiplos projetos para um único local no disco e suporta otimizações incrementais em nível de arquivo que aumentam ainda mais o desempenho. Como o pnpm oferece um ciclo de feedback muito mais rápido com problemas mínimos de compatibilidade, ele se tornou nossa escolha padrão para a gestão de pacotes Node.js.

## 60. Pydantic

*Adote*

Pydantic é uma biblioteca Python que usa as dicas de tipo (type hints) padrão para definir modelos de dados e impor esquemas de dados em tempo de execução. Originalmente, as anotações de tipo (type annotations) foram adicionadas ao Python para análise estática, mas sua crescente versatilidade levou a usos mais amplos, incluindo validação em tempo de execução. Construído sobre um núcleo rápido em Rust, ele fornece validação, parsing e serialização de dados de forma eficiente. Embora seja mais conhecido pelo desenvolvimento de APIs web, o Pydantic também se tornou essencial em aplicações

de LLM. Nós tipicamente usamos a técnica de saída estruturada de LLMs para gerenciar a natureza imprevisível dos LLMs. Ao definir um esquema de dados rígido, ele atua como uma rede de segurança para a natureza imprevisível da saída do modelo — convertendo respostas de texto de formato livre em objetos Python determinísticos e com tipagem segura (type-safe), como JSON, por exemplo. Essa abordagem, muitas vezes implementada por meio do Pydantic AI ou LangChain, transforma interações com LLMs, potencialmente frágeis, em contratos de dados confiáveis e legíveis por máquina. Nossos times têm usado o Pydantic com sucesso em produção para extrair representações estruturadas de documentos não estruturados, garantindo que a saída esteja em conformidade com uma estrutura válida. Dada sua maturidade, performance e confiabilidade, o Pydantic é agora nossa escolha padrão para aplicações de IA em Python em nível de produção.

## 61. Revisor de design com IA

***Experimente***

AI Design Reviewer é um plugin para o Figma para conduzir auditorias de design ou avaliações heurísticas e coletar feedback acionável sobre designs existentes ou novos. Suas auditorias cobrem críticas de UX, inconsistências de UI, lacunas de acessibilidade, qualidade de conteúdo e cenários de casos de borda. Além de identificar problemas, ele fornece recomendações conscientes do domínio que ajudam os times a construir um vocabulário de design compartilhado e a lógica por trás das escolhas de design. Nossos times usaram o AI Design Reviewer para analisar designs legados — identificando experiências positivas a serem mantidas e negativas a serem abordadas — o que informou os objetivos de UX para os redesenhos. Ele também serviu como um substituto para a revisão por pares, oferecendo feedback inicial sobre novos designs antes do handoff para o desenvolvimento.

## 62. Barman

***Experimente***

Barman (backup and recovery manager) é uma ferramenta de código aberto para gerenciar backups e recuperação de desastres de servidores PostgreSQL. Ele oferece suporte a todo o processo de recuperação de desastres, simplificando a criação de backups físicos por meio de uma variedade de métodos, organizando-os em um catálogo abrangente e restaurando backups para um servidor ativo com capacidades de recuperação pontual no tempo. Descobrimos que o Barman é poderoso e fácil de usar, e ficamos impressionados com a velocidade das operações de recuperação pontual no tempo durante as atividades de migração. Também o consideramos uma solução robusta para backups agendados, com a habilidade de lidar com configurações complexas e mistas de agendamento e retenção.

## 63. Claude Code

***Experimente***

O Claude Code da Anthropic é uma ferramenta de codificação com agente de IA, que fornece uma interface de linguagem natural e um modelo de execução agêntico para planejar e implementar fluxos de trabalho complexos e de múltiplos passos. Lançado há menos de um ano, ele já foi amplamente adotado por pessoas desenvolvedoras dentro e fora da Thoughtworks, o que nos leva a colocá-lo em Experimente. Agentes de programação baseados em console, como o Codex CLI da OpenAI, o Gemini CLI do Google e o OpenCode de código aberto, foram lançados, enquanto assistentes baseados em IDE, como Cursor, Windsurf e GitHub Copilot, agora incluem modos de agente. Mesmo assim, o Claude Code continua sendo um favorito. Vemos times usando-o não apenas para escrever e modificar código, mas também como um agente de IA de propósito geral para gerenciar especificações, histórias, configuração, infraestrutura e documentação. A programação com agentes

desloca o foco da pessoa desenvolvedora da escrita de código para a especificação da intenção e a delegação da implementação. Embora isso possa acelerar os ciclos de desenvolvimento, também pode levar ao comodismo com o código gerado por IA, o que, por sua vez, pode resultar em um código mais difícil de manter e evoluir — tanto para humanos quanto para agentes de IA. Portanto, é essencial que os times gerenciem rigorosamente como o Claude Code funciona, usando técnicas como engenharia de contexto, instruções compartilhadas e curadas e, potencialmente, times de agentes de programação.

## 64. Cleanlab

*Experimente*

No paradigma de IA centrada em dados, melhorar a qualidade do conjunto de dados muitas vezes proporciona maiores ganhos de performance do que fazer o ajuste (tuning) do próprio modelo. Cleanlab é uma biblioteca Python de código aberto projetada para lidar com esse desafio, identificando automaticamente problemas comuns de dados — como rotulagem incorreta, outliers e duplicatas — em conjuntos de dados de texto, imagem, tabulares e de áudio. Construída com base no princípio confident learning, o Cleanlab utiliza as probabilidades previstas pelo modelo para estimar o ruído nos rótulos e quantificar a qualidade dos dados. Essa abordagem independente de modelo permite que pessoas desenvolvedoras diagnostiquem e corrijam erros no conjunto de dados e, em seguida, treinem novamente os modelos para melhorar a robustez e a acurácia. Nossos times têm usado o Cleanlab com sucesso em produção, confirmando sua eficácia em cenários do mundo real. Nós o recomendamos como uma ferramenta valiosa para promover a padronização de dados e melhorar a qualidade de conjuntos de dados em projetos de engenharia de IA.

## 65. Context7

*Experimente*

Context7 é um servidor MCP que lida com imprecisões no código gerado por IA. Enquanto os LLMs se baseiam em dados de treinamento desatualizados, o Context7 garante que eles gerem código preciso, atualizado e específico da versão para as bibliotecas e frameworks usados em um projeto. Ele faz isso buscando a documentação mais recente e exemplos de código funcionais diretamente dos repositórios de origem do framework e injetando-os na janela de contexto do LLM no momento do prompting. Em nossa experiência, o Context7 reduziu bastante as alucinações de código e a dependência de dados de treinamento obsoletos. Você pode configurá-lo com editores de código de IA como o Claude Code, Cursor ou VS Code para gerar, refatorar ou depurar código dependente de framework.

## 66. Data Contract CLI

*Experimente*

Data Contract CLI é uma ferramenta de linha de comando de código aberto projetada para trabalhar com a especificação de Data Contract. Ela ajuda a criar e editar contratos de dados e, crucialmente, permite validar dados em relação ao seu contrato, o que é essencial para garantir a integridade e a qualidade de seus produtos de dados. A CLI oferece amplo suporte para múltiplas definições de schema (Avro, SQL DDL, Open Data Contract Standard, etc.) e pode comparar diferentes versões de contrato para detectar imediatamente alterações interruptivas (breaking changes). Nós a consideramos especialmente útil no espaço de data mesh para operacionalizar a governança de contratos entre produtos de dados via integração com CI/CD. Essa abordagem reduz erros manuais e garante a qualidade, integridade e compatibilidade dos dados nas trocas entre serviços.

## 67. Databricks Assistant

*Experimente*

Databricks Assistant é uma ferramenta de IA-conversacional, integrada diretamente na plataforma Databricks, atuando como um "par programador" contextual para profissionais de dados. Diferente dos assistentes de programação de uso geral, ele se beneficia de uma compreensão nativa do ambiente Databricks e do contexto dos dados, incluindo metadados do Unity Catalog. O Databricks Assistant vai além de gerar trechos de código; ele pode criar consultas complexas e com múltiplas etapas em SQL e Python, diagnosticar erros e fornecer explicações detalhadas e específicas do workspace. Para organizações que já utilizam o ecossistema Databricks, ele pode acelerar a produtividade e reduzir a barreira de entrada para tarefas complexas de dados.

## 68. Hoppscotch

*Experimente*

Hoppscotch é uma ferramenta leve de código aberto para desenvolvimento, depuração, teste e compartilhamento de APIs. Ela suporta múltiplos protocolos — incluindo HTTP, GraphQL e WebSocket — e oferece clientes multiplataforma para ambientes web, desktop e CLI. Embora o espaço de ferramentas de API esteja cheio de alternativas como Postman, Insomnia e Bruno, o Hoppscotch se destaca por sua pegada leve e design amigável à privacidade. Ele omite analytics, usa armazenamento local-first e suporta auto-hospedagem. É uma escolha forte para organizações que buscam uma maneira intuitiva de compartilhar scripts de API, mantendo uma forte privacidade de dados.

## 69. NVIDIA DCGM explorer

*Experimente*

O NVIDIA DCGM Exporter é uma ferramenta de código aberto que ajuda os times a monitorarem o treinamento distribuído de GPUs em escala. Ele converte a telemetria proprietária do NVIDIA Data Center GPU Manager (DCGM) em formatos abertos compatíveis com sistemas de monitoramento padrão. O Exporter expõe métricas críticas em tempo real — incluindo utilização da GPU, temperatura, energia e contagem de erros ECC — tanto da GPU quanto dos servidores hospedeiros. Essa visibilidade é essencial para organizações que fazem o ajuste fino (fine-tuning) de LLMs customizados ou executam trabalhos de treinamento de longa duração e intensivos em GPU. O efeito do retardatário — onde um worker lento se torna o gargalo de todo o processo — pode reduzir a taxa de transferência em mais de 10% e desperdiçar até 45% das horas de GPU alocadas. Projetado para ambientes cloud-native de larga escala, o DCGM Exporter se integra de forma transparente com o Prometheus e o Grafana, ajudando a garantir que cada GPU opere dentro dos limites ideais de performance.

## 70. Relational AI

*Experimente*

Quando grandes volumes de dados diversos são trazidos para o Snowflake, os relacionamentos inerentes e as regras implícitas dentro desses dados podem se tornar obscuros. Construído como uma aplicação nativa do Snowflake, o RelationalAI permite que os times construam modelos sofisticados que capturam conceitos significativos, definem entidades de negócio centrais e incorporam lógica complexa diretamente nas tabelas do Snowflake. Seu poderoso Graph Reasoner permite que as pessoas usuárias criem, analisem e visualizem grafos de conhecimento relacionais com base nesses modelos. Algoritmos integrados ajudam a explorar as estruturas do grafo e a revelar

padrões ocultos. Para organizações que gerenciam conjuntos de dados massivos e que mudam rapidamente, a construção de um grafo de conhecimento pode ser essencial para o monitoramento proativo e para a geração de insights mais ricos e acionáveis.

## 71. UXPilot

*Experimente*

UX Pilot é uma ferramenta de IA que suporta múltiplos estágios do processo de design de UX — desde o protótipo básico até o design visual de alta fidelidade e a revisão. Ela aceita entradas de texto ou imagem e pode gerar telas, fluxos e layouts automaticamente. Sua feature Autoflow cria transições de fluxo de pessoa usuária, enquanto o Deep Design produz resultados mais ricos e detalhados. O UX Pilot também inclui um plugin para o Figma que exporta os designs gerados para refinamento em ferramentas de design padrão. Nossos times têm usado o UX Pilot para ideação e inspiração, gerando múltiplas opções durante exercícios de Crazy 8's e traduzindo listas de histórias de projetos em quadros de visão de produto e conceitos de design em nível de épico. Ferramentas como o UX Pilot também permitem que pessoas que não são designers, como gerentes de produto, criem protótipos rápidos e coletem feedback inicial das partes interessadas — uma tendência crescente nos workflows de design assistidos por IA.

## 72. v0

*Experimente*

O v0 evoluiu desde a última vez que o apresentamos no Radar. Ele agora inclui um modo de design que diminui ainda mais a barreira para que gerentes de produto criem e ajustem protótipos de UI de autoatendimento. A versão mais recente introduz um modelo próprio com grandes janelas de contexto e capacidades multimodais, permitindo que o v0 gere e melhore UIs a partir de entradas de texto e visuais. Outra adição notável é seu modo de agente, que permite ao sistema dividir trabalhos mais complexos e selecionar o modelo apropriado para cada um. No entanto, essa feature ainda é nova, e o feedback inicial tem sido misto.

## 73. Augment Code

*Avalie*

Augment Code é um assistente de programação de IA que oferece suporte profundo e consciente de contexto em grandes bases de código. Ele se destaca por meio de uma engenharia de contexto avançada que permite atualizações rápidas do índice de código e recuperação rápida, mesmo com o código mudando frequentemente. O Augment suporta modelos como Claude Sonnet 4 e 4.5 e GPT-5, integra-se com GitHub, Jira e Confluence, e suporta o Model Context Protocol (MCP) para interoperabilidade com ferramentas externas. Ele fornece orientação passo a passo para alterações complexas na base de código — desde refactors e atualizações de dependências até atualizações de schema — juntamente com completudes em linha personalizadas que refletem as dependências específicas do projeto. O Augment também promove a colaboração, permitindo que os times consultem e compartilhem insights de código diretamente no Slack.

## 74. Azure AI Document Intelligence

*Avalie*

O Azure AI Document Intelligence (anteriormente Form Recognizer) extrai texto, tabelas e pares chave-valor de documentos não estruturados e os transforma em dados estruturados. Ele usa modelos de deep learning pré-treinados para interpretar layouts e semântica, e modelos customizados podem ser treinados por meio de uma interface no-code para formatos especializados.

Em alguns casos, no entanto, usuários avançados podem precisar, em vez disso, de uma interface de ajuste fino (fine-tuning) customizada. Um de nossos times relatou que o ADI reduziu significativamente a entrada manual de dados, melhorou a acurácia dos dados e acelerou a geração de relatórios, levando a decisões orientadas a dados mais rápidas. Assim como o Amazon Textract e o Google Document AI, ele fornece processamento de documentos de nível empresarial com uma forte compreensão de layout. Uma alternativa de código aberto emergente é o Docling da IBM, que oferece uma abordagem mais flexível e centrada em código para a extração de dados estruturados. Em comparação com as ferramentas de OCR tradicionais, o ADI captura não apenas o texto, mas também a estrutura e os relacionamentos, facilitando a integração em pipelines de dados downstream. Dito isso, observamos uma latência ocasional ao incorporá-lo em workflows de usuário síncronos, por isso, recomendamos usá-lo principalmente para processamento assíncrono.

## 75. Docling

***Avalie***

Docling é uma biblioteca de código aberto em Python e TypeScript para o processamento avançado de documentos com dados não estruturados. Ele aborda o problema frequentemente negligenciado da "última milha" de converter documentos do mundo real — como PDFs e PowerPoints — em formatos limpos e legíveis por máquina. Diferente dos extratores tradicionais, o Docling usa uma abordagem baseada em visão computacional para interpretar o layout do documento e a estrutura semântica, o que torna sua saída particularmente valiosa para pipelines de geração aumentada por recuperação (RAG). Ele converte documentos complexos em formatos estruturados como JSON ou Markdown, suportando técnicas como a saída estruturada de LLMs. Isso contrasta com o ColPali, que alimenta imagens de página diretamente a um modelo de linguagem e visão para recuperação. A natureza de código aberto do Docling e seu núcleo em Python, construído sobre um modelo de dados customizado baseado em Pydantic, fornecem uma alternativa flexível e auto-hospedada a ferramentas de nuvem proprietárias como o Azure Document Intelligence, o Amazon Textract e o Google Document AI. Apoiado pela IBM Research, o rápido desenvolvimento do projeto e a arquitetura plug-and-play para integração com outros frameworks como o LangGraph fazem com que valha a pena avaliá-lo para times que constroem pipelines de dados prontos para IA em nível de produção.

## 76. E2B

***Avalie***

E2B é uma ferramenta de código aberto para executar códigos gerados por IA em sandboxes seguras e isoladas na nuvem. Os agentes podem usar essas sandboxes, construídas sobre as microVMs Firecracker, para executar código com segurança, analisar dados, realizar pesquisas ou operar uma máquina virtual. Isso permite construir e implantar agentes de IA de nível empresarial com controle total e segurança sobre o ambiente de execução.

## 77. Editor Helix

***Avalie***

Houve um certo ressurgimento de editores de texto simples com o objetivo de substituir o favorito da linha de comando, o Vim. O Helix é um desses concorrentes no espaço concorrido, ao lado do Neovim e, mais recentemente, do Kakoune. Descrevendo-se — de forma um tanto divertida — como um editor de texto pós-moderno, o Helix apresenta múltiplos cursores, suporte ao Tree-sitter e suporte integrado ao Language Server Protocol (LSP), que foi o que primeiro chamou nossa atenção. O Helix está em desenvolvimento ativo, com um sistema de plugins a caminho. No geral, é um editor modal leve que parece familiar para quem usa o Vim, mas adiciona algumas conveniências modernas.

## 78. Kueue

*Avalie*

Kueue é um controller nativo para Kubernetes para enfileiramento de jobs que gerencia cotas e consumo de recursos. Ele fornece APIs para lidar com cargas de trabalho do Kubernetes com prioridades e requisitos de recursos variados, funcionando como um gerenciador em nível de job que determina quando admitir ou remover jobs. Projetado para gerenciamento eficiente de recursos, priorização de jobs e agendamento avançado, o Kueue ajuda a otimizar a execução de cargas de trabalho em ambientes Kubernetes — particularmente para cargas de trabalho de ML que usam ferramentas como Kubeflow. Ele funciona em conjunto com o cluster-autoscaler e o kube-scheduler em vez de substituí-los, focando na admissão de jobs com base na ordem, cota, prioridade e consciência de topologia. Como parte do ecossistema do Kubernetes Special Interest Group (SIG), o Kueue adere aos seus padrões de desenvolvimento.

## 79. MCP-Scan

*Avalie*

MCP-Scan é um scanner de segurança para servidores de Model Context Protocol (MCP) que opera em dois modos: scan e proxy. No modo de scan, ele analisa configurações e descrições de ferramentas para detectar vulnerabilidades conhecidas, como injeções de prompt (prompt injections), envenenamento de ferramenta (tool poisoning) e fluxos tóxicos. No modo proxy, o MCP-Scan atua como uma ponte entre o sistema de agente e o servidor MCP, monitorando continuamente o tráfego em tempo de execução. Esse modo também impõe regras de segurança e guardrails customizados, incluindo validação de chamadas de ferramentas, detecção de PII e restrições de fluxo de dados. A ferramenta fornece uma camada de segurança proativa para os agentes, garantindo que, mesmo que um prompt malicioso seja aceito, o agente não possa executar ações prejudiciais. O MCP-Scan é uma solução de segurança de propósito específico para o campo emergente de sistemas com agentes.

## 80. oRPC

*Avalie*

O oRPC (OpenAPI Remote Procedure Call) fornece APIs com segurança de tipos de ponta a ponta em TypeScript, ao mesmo tempo que adere totalmente à especificação OpenAPI. Ele pode gerar automaticamente uma especificação OpenAPI completa, simplificando a integração e a documentação. Consideramos o oRPC particularmente forte para integrações. Enquanto alternativas como o tRPC e o ElysiaJS muitas vezes exigem a adoção de um novo framework para obter segurança de tipos, o oRPC se integra de forma transparente com frameworks Node.js existentes, incluindo Express, Fastify, Hono e Next.js. Essa flexibilidade o torna uma excelente escolha para times que buscam adotar segurança de tipos de ponta a ponta em APIs existentes sem uma refatoração disruptiva.

## 81. Extensão Power user for dbt para VS Code

*Avalie*

A Power user for dbt é uma extensão para o Visual Studio Code que se integra diretamente com ambientes dbt e dbt Cloud. Como o dbt continua sendo uma de nossas ferramentas favoritas, qualquer coisa que melhore sua usabilidade é uma adição bem-vinda ao ecossistema. Anteriormente, as pessoas desenvolvedoras dependiam de múltiplas ferramentas para validar o código SQL ou inspecionar a linhagem do modelo fora da IDE. Com esta extensão, essas capacidades agora estão

integradas ao VS Code, oferecendo autocompletar de código, resultados de query em tempo real e linhagem visual de modelos e colunas. Esta última feature facilita a navegação entre os modelos. Nossos times relatam que o plugin reduz os erros de pipeline e melhora a experiência geral de desenvolvimento. Se você usa o dbt, recomendamos fortemente que dê uma olhada nesta ferramenta.

## 82. Serena

***Avalie***

Serena é um poderoso toolkit de desenvolvimento que equipa agentes de programação, como o Claude Code, com recursos semelhantes aos de uma IDE para busca e edição semântica de código. Ao operar no nível de símbolo e entender a estrutura relacional do código, o Serena melhora muito a eficiência de tokens. Em vez de ler arquivos inteiros ou depender de substituições de string rudimentares, os agentes de programação podem usar ferramentas precisas do Serena, como find_symbol, find_referencing_symbols e insert_after_symbol, para localizar e editar o código. Embora o impacto seja mínimo em projetos pequenos, essa eficiência é extremamente valiosa à medida que a base de código cresce.

## 83. Sweetpad

***Avalie***

A extensão SweetPad permite que pessoas desenvolvedoras usem o VS Code ou o Cursor para todo o ciclo de desenvolvimento de aplicações Swift em plataformas da Apple. Ela elimina a necessidade de alternar constantemente para o Xcode, integrando ferramentas essenciais como xcodebuild, xcode-build-server e swift-format. Pessoas desenvolvedoras podem construir, rodar e depurar aplicações Swift para iOS, macOS e watchOS diretamente de suas IDEs, ao mesmo tempo que gerenciam simuladores e fazem deploy para dispositivos sem abrir o Xcode.

## 84. Tape/Z Tools for Assembly Program Exploration for Z/OS

***Avalie***

Tape/Z (Tools for Assembly Program Exploration for Z/OS), ou, em português, “Ferramentas Tape/Z para exploração de programas assembly para Z/OS”, é um toolkit em evolução para analisar código HLASM (High Level Assembler) de mainframe. Desenvolvido por uma Thoughtworker, ele fornece capacidades como parsing, construção de grafo de fluxo de controle, rastreamento de dependências e visualização de fluxogramas. Há muito tempo notamos a escassez de ferramentas abertas e orientadas pela comunidade no espaço de mainframe, onde a maioria das opções permanece proprietária ou vinculada a ecossistemas de fornecedores. O Tape/Z ajuda a preencher essa lacuna, oferecendo capacidades de análise acessíveis e programáveis. Juntamente com o COBOL REKT — um toolkit complementar para COBOL que também usamos várias vezes com clientes — ele representa um progresso encorajador em direção a ferramentas modernas e amigáveis para pessoas desenvolvedoras de sistemas mainframe.

# Linguagens e Frameworks

**Adote**

85. Fastify
86. LangGraph
87. vLLM

**Experimente**

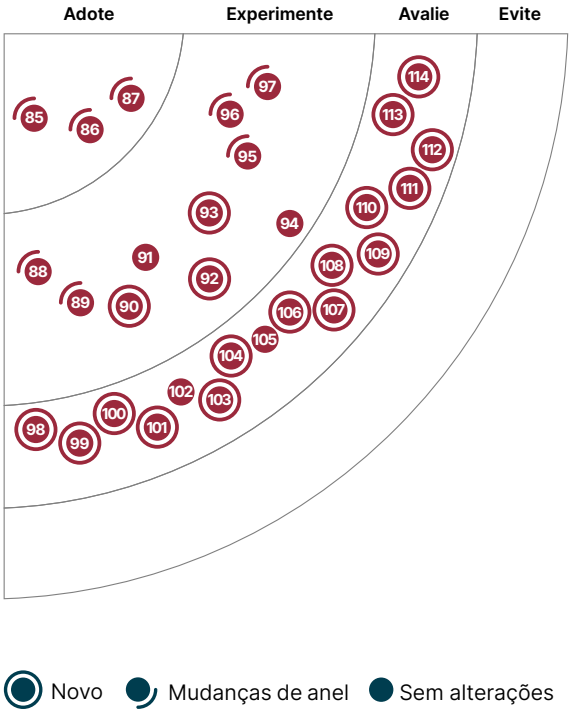
88. Crossplane
89. DeepEval
90. fastMCP
91. LiteLLM
92. MLForecast
93. Nuxt
94. Phoenix
95. Presidio
96. PydanticAI
97. Tauri

**Avalie**

98. Agent Development Kit (ADK)
99. Agno
100. assistant-ui
101. AutoRound
102. Browser Use
103. DeepSpeed
104. Drizzle
105. Criptografia pós-quântica em Java
106. kagent
107. LangExtract
108. Langflow
109. LMCache
110. Mem0
111. Linguagem de avaliação de controles de segurança abertos (OSCAL)
112. OpenInference
113. Valibot
114. Vercel AI SDK

**Evite**

—



## 85. Fastify

*Adote*

Continuamos a ter uma experiência positiva com o Fastify — um framework web rápido, não opinativo e de baixo overhead para Node.js. Ele fornece todas as capacidades essenciais de um framework web mínimo — incluindo parsing, validação e serialização — juntamente com um sistema de plugins robusto e forte apoio da comunidade. Nossos times não viram desvantagens significativas no uso do Fastify em relação a alternativas como o Express.js, ao mesmo tempo que obtiveram melhorias de performance mensuráveis, tornando-o uma escolha convincente para o desenvolvimento web mínimo em Node.js.

## 86. LangGraph

*Adote*

LangGraph é um framework de orquestração para construir aplicações stateful multi-agente usando LLMs. Ele fornece primitivas de baixo nível, como nós e arestas, juntamente com features integradas que dão às pessoas desenvolvedoras controle granular sobre os workflows dos agentes, gerenciamento de memória e persistência de estado. Isso significa que as pessoas desenvolvedoras podem começar com um grafo pré-construído simples e escalar para arquiteturas de agentes complexas e em evolução. Com suporte para streaming, gerenciamento avançado de contexto e padrões de resiliência como fallbacks de modelo e tratamento de erros de ferramentas, o LangGraph permite construir aplicações agênticas robustas e de nível de produção. Sua abordagem baseada em grafos garante workflows previsíveis e customizáveis, e simplifica a encontrabilidade de erros e a escalabilidade. Nossos times tiveram ótimos resultados usando o LangGraph para construir sistemas multi-agente, graças ao seu design leve e modular.

## 87. vLLM

*Adote*

vLLM é um motor de inferência para LLMs de alta taxa de transferência (high-throughput) e eficiente em memória, que pode rodar na nuvem ou no local. Ele suporta múltiplas arquiteturas de modelo e modelos populares de código aberto. Nossos times implantam workers vLLM dockerizados em plataformas de GPU como NVIDIA DGX e Intel HPC, hospedando modelos que incluem Llama 3.1 (8B e 70B), Mistral 7B e Llama-SQL para assistência à programação para pessoas desenvolvedoras, busca de conhecimento e interações com bancos de dados em linguagem natural. O vLLM é compatível com o padrão do SDK da OpenAI, permitindo um model serving consistente. O AI Model Catalog do Azure usa um contêiner de inferência customizado construído sobre o vLLM para aumentar a performance de serviço, com o vLLM como o motor de inferência padrão devido à sua alta taxa de transferência e gerenciamento de memória eficiente. O framework vLLM tornou-se referência para implantações de modelos em grande escala.

## 88. Crossplane

*Experimente*

Desde sua última aparição no Radar, a adoção do Crossplane continuou a crescer, particularmente para estender clusters Kubernetes. Em nosso trabalho, descobrimos que o Crossplane se destaca em casos de uso específicos, em vez de como uma ferramenta de infraestrutura como código (IaC) de propósito geral. Nossas observações anteriores ainda se mantêm: o Crossplane funciona melhor como um complemento para workloads implantados no Kubernetes, não como um substituto completo

para ferramentas como o Terraform. Os times que apostaram "todas as fichas" no Crossplane como sua principal solução de IaC muitas vezes enfrentaram dificuldades, enquanto aqueles que o usaram de forma pragmática — para casos de uso direcionados e customizados — viram ótimos resultados. Descarregar o gerenciamento do ciclo de vida dos recursos para o Crossplane enquanto se usam as APIs XRD para customização leve provou ser especialmente eficaz. O Crossplane é particularmente valioso ao gerenciar recursos cujos ciclos de vida são simples, mas não nativos do Kubernetes. Embora ele agora possa criar clusters Kubernetes — uma capacidade que faltava anteriormente — aconselhamos cautela na adoção do Crossplane como um substituto completo do Terraform. Em nossa experiência, ele funciona melhor quando a IaC serve como a camada fundamental, com o Crossplane adicionado por cima para requisitos especializados.

## 89. DeepEval

***Experimente***

DeepEval é um framework de avaliação de código aberto, baseado em Python, para aferir a performance de LLMs. Ele pode ser usado para avaliar a geração aumentada por recuperação (RAG) e outras aplicações construídas com frameworks como LlamaIndex ou LangChain, bem como para fazer o baseline e o benchmark de modelos. O DeepEval vai além das pontuações baseadas em correspondência de palavras, avaliando acurácia, relevância e consistência para fornecer uma avaliação mais confiável em cenários do mundo real. Ele inclui métricas como detecção de alucinações, relevância da resposta e otimização de hiperparâmetros, e suporta o GEval para a criação de métricas customizadas e específicas para o caso de uso. Nossos times estão usando o DeepEval para fazer o ajuste fino (fine-tuning) das saídas de agentes usando a técnica de LLM como juiz. Ele se integra com o pytest e pipelines de CI/CD, tornando-o fácil de adotar e valioso para a avaliação contínua. Para times que desenvolvem aplicações baseadas em LLM em ambientes regulados, o Inspect AI, desenvolvido pelo Instituto de Segurança de IA do Reino Unido, oferece uma alternativa com um foco mais forte em auditoria e conformidade.

## 90. fastMCP

***Experimente***

O Model Context Protocol (MCP) está rapidamente se tornando um padrão para fornecer contexto e ferramentas para aplicações de LLM. No entanto, implementar um servidor MCP normalmente envolve bastante código repetitivo para configuração, manipulação de protocolo e gerenciamento de erros. FastMCP é um framework Python que simplifica esse processo, abstraindo a complexidade do protocolo e permitindo que pessoas desenvolvedoras definam recursos e ferramentas MCP por meio de decorators Python intuitivos. Essa abstração permite que os times se concentrem na lógica de negócio, resultando em implementações MCP mais limpas e de fácil manutenção. Embora o FastMCP 1.0 já esteja incorporado no SDK oficial, o padrão MCP continua a evoluir rapidamente. Recomendamos monitorar o release 2.0 e garantir que os times permaneçam alinhados com as mudanças na especificação oficial.

## 91. LiteLLM

***Experimente***

LiteLLM é um SDK que fornece integração transparente com múltiplos provedores de LLM por meio de um formato padronizado da API da OpenAI. Ele suporta uma ampla gama de provedores e modelos, oferecendo uma interface unificada para geração de texto, *embeddings* e geração de imagens. Ao abstrair as diferenças de API específicas de cada provedor, o LiteLLM simplifica a integração e roteia

automaticamente as requisições para o endpoint do modelo correto. Ele também inclui features de nível de produção, como guardrails, caching, logging, limitação de taxa e balanceamento de carga por meio de seu framework de proxy. À medida que as organizações incorporam aplicações impulsionadas por IA mais profundamente em seus fluxos de trabalho, governança e observabilidade se tornam essenciais. Nossos times têm usado o LiteLLM como um gateway de IA para padronizar, proteger e ganhar visibilidade sobre o uso de IA em toda a empresa.

## 92. MLForecast

*Experimente*

MLForecast é um framework e biblioteca Python para previsão de séries temporais que aplica modelos de machine learning a conjuntos de dados em grande escala. Ele simplifica o processo tipicamente complexo de engenharia de features automatizada — incluindo defasagens, estatísticas móveis e features baseadas em data — e é uma das poucas bibliotecas com suporte nativo para frameworks de computação distribuída como Spark e Dask, garantindo escalabilidade. Ele também suporta previsão probabilística usando métodos como previsão conforme (conformal prediction), fornecendo medidas quantitativas da incerteza da previsão. Em nossa avaliação, o MLForecast escalou eficientemente para milhões de pontos de dados e consistentemente superou ferramentas comparáveis. Para times que buscam operacionalizar rapidamente a previsão de séries temporais em dados de alto volume, o MLForecast é uma escolha convincente.

## 93. Nuxt

*Experimente*

Nuxt é um meta-framework opinativo construído sobre o Vue.js para a criação de aplicações web full-stack, frequentemente conhecido como o "Next.js para Vue.js". Semelhante à sua contraparte em React, o Nuxt oferece recursos amigáveis para SEO, como pré-renderização, renderização no lado do servidor (SSR, server-side rendering) e gerenciamento de metadados, tornando-o uma excelente opção para desenvolver sites orientados à performance e otimizados para SEO na stack do Vue.js. O Nuxt é mantido pela Vercel, a mesma empresa por trás do Next.js, e apoiado por uma forte comunidade e um ecossistema de módulos oficiais e de terceiros. Esses módulos simplificam a integração de funcionalidades como processamento de imagem, sitemaps e Tailwind CSS. O Nuxt é uma boa escolha para times que procuram um framework abrangente e opinativo para construir aplicações amigáveis para SEO com Vue.js.

## 94. Phoenix

*Experimente*

Continuamos a ter experiências positivas com o Phoenix, um framework web MVC server-side escrito em Elixir. O Phoenix se baseia nos aprendizados de desenvolvimento rápido de aplicações e de experiência da pessoa desenvolvedora do Ruby on Rails, ao mesmo tempo que avança para paradigmas de programação funcional. Neste volume, estamos destacando o lançamento do Phoenix LiveView 1.0. O LiveView é uma solução HTML-over-the-wire — semelhante ao HTMX ou Hotwire — que permite que pessoas desenvolvedoras construam experiências de usuário ricas e em tempo real inteiramente com HTML renderizado no servidor. Enquanto tecnologias similares geralmente lidam com atualizações parciais por meio de troca de HTML, o LiveView fornece uma arquitetura completa baseada em componentes por meio dos LiveComponents, oferecendo composição,

passagem de props, gerenciamento de estado e hooks de ciclo de vida semelhantes aos do React ou Vue.js. O LiveView combina a produtividade e a escalabilidade do Phoenix com um gerenciamento de complexidade robusto, tornando-o bem adequado para a construção de frontends altamente interativos sem a necessidade de um framework JavaScript completo.

## 95. Presidio

*Experimente*

Presidio é um SDK de proteção de dados para identificar e anonimizar dados sensíveis em texto estruturado e não estruturado. Ele detecta informações de identificação pessoal (PII), como números de cartão de crédito, nomes e locais, usando reconhecimento de entidade nomeada, expressões regulares e lógica baseada em regras. O Presidio suporta reconhecedores de entidade e pipelines de desidentificação customizados, permitindo que as organizações o adaptem às suas necessidades de privacidade e conformidade. Nossos times têm usado o Presidio em ambientes empresariais com controles rigorosos de compartilhamento de dados ao integrar com LLMs. Embora ele automatize a detecção de informações sensíveis, não é infalível e pode omitir ou identificar entidades incorretamente. Os times devem ter cautela ao confiar em seus resultados.

## 96. PydanticAI

*Experimente*

O Pydantic AI continua se consolidando como um framework de código aberto estável e com bom suporte para a construção de agentes de GenAI em produção. Construído sobre a base confiável do Pydantic, ele oferece forte segurança de tipos, observabilidade de primeira classe por meio do OpenTelemetry e ferramentas de avaliação integradas. O lançamento da versão 1.0 em 4 de setembro de 2025 marcou um marco significativo em sua maturidade. Desde então, a consideramos confiável e amplamente adotado por sua simplicidade e manutenibilidade, juntando-se a outros frameworks de agentes populares como o LangChain e o LangGraph. Atualizações recentes facilitaram a implementação de servidores e clientes do Model Context Protocol (MCP), com suporte adicional para padrões emergentes como AG-UI e A2A. Com sua API limpa e ecossistema crescente, o Pydantic AI tornou-se uma escolha atraente para nossos times que constroem aplicações de GenAI prontas para produção em Python.

## 97. Tauri

*Experimente*

Tauri é um framework para construir aplicações de desktop de alta performance usando uma única base de código de interface web. Diferente de empacotadores web tradicionais, como o Electron, o Tauri é construído em Rust e utiliza a webview nativa do sistema operacional, resultando em binários menores e maior segurança. Nós avaliamos o Tauri pela primeira vez há alguns anos; desde então, ele se expandiu para além do desktop para oferecer suporte a iOS e Android. A versão mais recente introduz um modelo de permissão e escopo mais flexível, substituindo a lista de permissões mais antiga, e apresenta uma camada de comunicação entre processos (IPC, inter-process communication) reforçada, que suporta a transferência de dados brutos e melhora o desempenho. Essas atualizações são respaldadas por uma auditoria de segurança externa. Juntamente com as diretrizes oficiais de distribuição para as principais lojas de aplicativos, essas melhorias fortalecem ainda mais a posição do Tauri no espaço de desenvolvimento multiplataforma.

## 98. Agent Development Kit (ADK)

*Avalie*

O Agent Development Kit (ADK) é um *framework* para o desenvolvimento e implantação de agentes de IA que aplica a disciplina da engenharia de software moderna, em vez de depender apenas de *prompting*. Ele introduz abstrações familiares, como classes, métodos, padrões de *workflow* e suporte a CLI. Em comparação com *frameworks* como LangGraph ou CrewAI, o ponto forte do ADK está em sua profunda integração com a infraestrutura de IA do Google — fornecendo fundamentação, acesso a dados e monitoramento prontos para o ambiente corporativo. O ADK também foi projetado com foco em interoperabilidade, oferecendo suporte a *wrappers* de ferramentas e ao protocolo A2A para comunicação entre agentes. Para organizações que já investem no GCP, o ADK representa uma base promissora para construir uma arquitetura agêntica escalável, segura e gerenciável. Embora ainda esteja em fase inicial de evolução, o ADK sinaliza a direção do Google rumo a um ambiente de desenvolvimento de agentes nativo e *full-stack*. Recomendamos acompanhar de perto sua maturidade e o crescimento do ecossistema.

## 99. Agno

*Avalie*

Agno é um framework para construir, executar e gerenciar sistemas multi-agente. Ele oferece a flexibilidade para criar agentes totalmente autônomos ou workflows controlados e baseados em etapas, com suporte nativo para human-in-the-loop, gestão de sessão, memória e conhecimento. Apreciamos seu foco na eficiência, com tempos de inicialização de agente impressionantes e baixo consumo de memória. O Agno também vem com seu próprio runtime, o AgentOS, uma aplicação FastAPI com um plano de controle integrado para testes, monitoramento e gerenciamento otimizados de sistemas agênticos.

## 100. assistant-ui

*Avalie*

assistant-ui é uma biblioteca de código aberto em TypeScript e React para interfaces de chat com IA. Ela lida com as partes complexas da implementação da UI de chat — como streaming, gerenciamento de estado para edição de mensagens e troca de branches, e features comuns de UX — enquanto permite que pessoas desenvolvedoras projetem seus próprios componentes usando primitivas Radix. Ela suporta integração com runtimes populares, incluindo o Vercel AI SDK e o LangGraph, e oferece soluções de runtime customizáveis para casos de uso complexos. Construímos com sucesso uma interface de chat simples com o assistant-ui e ficamos satisfeitos com os resultados.

## 101. AutoRound

*Avalie*

O AutoRound da Intel é um algoritmo de quantização avançado para compressão de grandes modelos de IA, como LLMs e modelos de linguagem e visão (VLMs), com perda mínima de acurácia. Ele reduz o tamanho do modelo para larguras de bit ultrabaixas (2–4 bits) usando otimização por gradiente descendente de sinal e aplica larguras de bit mistas entre as camadas para uma eficiência ótima. Esse processo de quantização também é notavelmente rápido: você pode quantizar um modelo de 7 bilhões de parâmetros em apenas alguns minutos em uma única GPU. Como o AutoRound se integra com motores de inferência populares, como o vLLM e o Transformers, ele é uma opção atraente para a quantização de modelos.

## 102. Browser Use

*Avalie*

Browser Use é uma biblioteca Python de código aberto que permite que agentes baseados em LLM operem navegadores web e interajam com aplicações web. Ela pode navegar, inserir dados, extrair texto e gerenciar múltiplas abas para coordenar ações entre aplicações. A biblioteca é particularmente útil quando agentes de IA precisam acessar, manipular ou recuperar informações de conteúdo web. Ela suporta uma variedade de LLMs e utiliza o Playwright para combinar o entendimento visual com a extração da estrutura HTML para interações web mais ricas. Nossos times integraram o Browser Use com o framework Pytest e relatórios do Allure para explorar testes automatizados com LLMs. Os passos de teste foram escritos em linguagem natural para o agente executar, capturando screenshots em asserções ou falhas. O objetivo era permitir QA fora do horário de expediente, buscando automaticamente os casos de teste do Confluence para verificação pós-desenvolvimento. Os resultados iniciais são promissores, embora as respostas do agente pós-tarefa muitas vezes não tenham descrições detalhadas de falhas, exigindo relatórios de erro customizados.

## 103. DeepSpeed

*Avalie*

DeepSpeed é uma biblioteca Python que otimiza o deep learning distribuído tanto para treinamento quanto para inferência. Para o treinamento, ela integra tecnologias como o Zero Redundancy Optimizer (ZeRO) e o paralelismo 3D para escalar modelos de forma eficiente em milhares de GPUs. Para a inferência, ela combina paralelismo de tensor, de pipeline, de expert e ZeRO com kernels customizados e otimizações de comunicação para minimizar a latência. O DeepSpeed potencializou alguns dos maiores modelos de linguagem do mundo, incluindo o Megatron-Turing NLG (530B) e o BLOOM (176B). Ele suporta tanto modelos densos quanto esparsos, entrega alta taxa de transferência do sistema e permite o treinamento ou a inferência em múltiplas GPUs com recursos restritos. A biblioteca se integra de forma transparente com as populares Hugging Face Transformers, PyTorch Lightning e Accelerate, tornando-se uma opção altamente eficaz para workloads de deep learning de grande escala ou com recursos limitados.

## 104. Drizzle

*Avalie*

Drizzle é um ORM TypeScript leve. Diferente do Prisma ORM, ele oferece às pessoas desenvolvedoras tanto uma simples API estilo SQL quanto uma interface de consulta mais tradicional no estilo ORM. Ele também suporta a extração de schemas a partir de bancos de dados existentes, permitindo abordagens tanto database-first quanto code-first. O Drizzle foi projetado com ambientes serverless em mente: ele tem um tamanho de bundle pequeno e suporta prepared statements, permitindo que as consultas SQL sejam pré-compiladas para que o driver do banco de dados execute SQL binário diretamente, em vez de fazer o parse das consultas a cada vez. Sua simplicidade e suporte a serverless tornam o Drizzle uma escolha atraente para o uso de ORM no ecossistema TypeScript.

## 105. Criptografia pós-quântica em Java

*Avalie*

A computação quântica continua a avançar rapidamente, com ofertas SaaS como o AWS Braket agora fornecendo acesso a algoritmos quânticos em múltiplas arquiteturas. Desde março, o Java 24 introduziu a criptografia pós-quântica em Java, adicionando suporte para algoritmos criptográficos

pós-quânticos como o ML-KEM e o ML-DSA, e o .Net 10 também expandiu seu suporte. Nosso conselho é simples: se você está construindo software nessas linguagens, comece a adotar algoritmos seguros contra a computação quântica agora para preparar seus sistemas para o futuro.

## 106. kagent

*Avalie*

Kagent é um framework de código aberto para executar IA com agentes dentro de clusters Kubernetes. Ele permite que agentes baseados em LLM planejem e executem tarefas operacionais, como diagnosticar problemas, remediar configurações ou interagir com ferramentas de observabilidade por meio de APIs nativas do Kubernetes e integrações com o Model Context Protocol (MCP). Seu objetivo é trazer o "AgentOps" para a infraestrutura cloud-native, combinando gerenciamento declarativo com raciocínio autônomo. Como um projeto Sandbox da CNCF, o Kagent deve ser introduzido com cuidado, especialmente dados os riscos de conceder aos LLMs capacidades de gerenciamento operacional. Técnicas como a análise de fluxo tóxico podem ser especialmente valiosas ao avaliar e mitigar esses riscos.

## 107. LangExtract

*Avalie*

LangExtract é uma biblioteca Python que usa LLMs para extrair informações estruturadas de texto não estruturado com base em instruções definidas pela pessoa usuária. Ela processa materiais de domínio específico — como anotações e relatórios clínicos — identificando e organizando detalhes importantes, ao mesmo tempo em que mantém cada ponto de dado extraído rastreável até sua fonte. As entidades extraídas podem ser exportadas como um arquivo `.jsonl`, um formato padrão para dados de modelo de linguagem, e visualizadas por meio de uma interface HTML interativa para revisão contextual. Nossos times avaliaram o LangExtract para extrair entidades para preencher um grafo de conhecimento de domínio e o consideraram eficaz para transformar documentos complexos em representações estruturadas e legíveis por máquina.

## 108. Langflow

*Avalie*

LangFlow é uma plataforma de código aberto e baixo código (low-code) para construir e visualizar workflows de LLM. Construída sobre o LangChain, ela permite que pessoas desenvolvedoras possam encadear comandos, ferramentas, bancos de dados vetoriais e componentes de memória por meio de uma interface de arrastar e soltar (drag-and-drop), enquanto ainda oferece suporte a código Python customizado para lógicas avançadas. É particularmente útil para prototipar aplicações agênticas sem escrever o código de backend completo. No entanto, o LangFlow ainda é relativamente novo e tem algumas imperfeições para o uso em produção. Nossa cautela usual em relação a plataformas low-code se aplica aqui. Dito isso, gostamos que os workflows do LangFlow podem ser definidos e versionados como código, o que pode mitigar algumas das desvantagens das plataformas low-code.

## 109. LMCache

*Avalie*

LMCache É uma solução de cache chave-valor (KV) que acelera a infraestrutura de serviço de LLMs. Ele atua como uma camada de cache especializada em um conjunto de motores de inferência de LLM, armazenando entradas de cache KV pré-computadas para textos que provavelmente serão processados várias vezes, como históricos de chat ou coleções de documentos. Ao persistir esses

valores em disco, as computações de pré-preenchimento podem ser descarregadas da GPU, reduzindo o tempo para o primeiro token (TTFT) e cortando custos de inferência em workloads exigentes, como pipelines de RAG, aplicações de chat de múltiplos turnos e sistemas agênticos. Você pode integrar o LMCache com os principais servidores de inferência, como vLLM ou NVIDIA Dynamo, e achamos que vale a pena avaliar seu impacto na sua configuração.

## 110. Mem0

*Avalie*

Mem0 é uma camada de memória projetada para agentes de IA. Abordagens ingênuas frequentemente armazenam históricos de chat inteiros em um banco de dados e os reutilizam em conversas futuras, o que leva ao uso excessivo de tokens. O Mem0 substitui isso por uma arquitetura mais sofisticada que separa a memória em uma de curto prazo e uma camada inteligente de longo prazo que extrai e armazena apenas fatos e relações importantes. Sua arquitetura combina um armazenamento vetorial para similaridade semântica com um grafo de conhecimento para entender dados temporais e relacionais. Esse design reduz significativamente o uso de tokens de contexto, ao mesmo tempo que permite que os agentes mantenham uma consciência de longo prazo, o que é extremamente útil para personalização e muitos outros casos de uso.

## 111. Linguagem de avaliação de controles de segurança abertos (OSCAL)

*Avalie*

A linguagem de avaliação de controles de segurança abertos (OSCAL) é um formato de intercâmbio de informações aberto e legível por máquina, projetado para aumentar a automação na conformidade e no gerenciamento de riscos, e ajudar os times a se afastarem de abordagens manuais baseadas em texto. Liderada pelo Instituto Nacional de Padrões e Tecnologia (NIST), a OSCAL fornece representações padrão em XML, JSON e YAML para expressar controles de segurança associados a frameworks da indústria, como SOC 2 e PCI, bem como frameworks governamentais como o FedRAMP nos Estados Unidos, o Catálogo de Controles de Cibersegurança de Singapura e o Manual de Segurança da Informação da Austrália. Embora a OSCAL ainda não tenha sido amplamente adotada fora do setor público e seu ecossistema ainda esteja amadurecendo, estamos entusiasmadas com seu potencial para otimizar as avaliações de segurança, reduzir a dependência de planilhas e exercícios de marcar caixas, e até mesmo permitir a conformidade automatizada quando incorporada a plataformas de conformidade como código e de conformidade contínua.

## 112. OpenInference

*Avalie*

OpenInference é um conjunto de convenções e plugins; é complementar ao OpenTelemetry e projetado para observar aplicações de IA. Ele fornece instrumentação padronizada para frameworks e bibliotecas de machine learning, o que ajuda pessoas desenvolvedoras a rastrearem invocações de LLM juntamente com o contexto ao redor, como recuperações de vector store ou chamadas de ferramentas externas a APIs e motores de busca. Os spans podem ser exportados para qualquer coletor compatível com OTEL, garantindo o alinhamento com os pipelines de telemetria existentes. Anteriormente, publicamos sobre o Langfuse, uma plataforma de observabilidade de LLM comumente usada — o SDK do OpenInference pode registrar traces no Langfuse e em outras plataformas de observabilidade compatíveis com OpenTelemetry.

## 113. Valibot

*Avalie*

Valibot é uma biblioteca de validação de esquemas em TypeScript. Como outras bibliotecas de validação populares em TypeScript, como Zod e Ajv, ela fornece inferência de tipos, mas seu design modular é o que a diferencia. Essa arquitetura permite que os bundlers realizem tree shaking e code splitting eficazes, incluindo apenas as funções de validação realmente utilizadas. O Valibot pode reduzir o tamanho do bundle em até 95% em comparação com o Zod em cenários ideais. É uma escolha atraente para a validação de esquemas em ambientes onde o tamanho do bundle é crítico, como validação no lado do cliente ou funções sem servidor.

## 114. Vercel AI SDK

*Avalie*

O Vercel AI SDK é um conjunto de ferramentas full-stack de código aberto para a construção de aplicações e agentes impulsionados por IA no ecossistema TypeScript. Ele consiste em dois componentes principais: o AI SDK Core padroniza as chamadas de LLM agnósticas de modelo, suportando geração de texto, geração de objetos estruturados e chamada de ferramentas; o AI SDK UI simplifica o desenvolvimento de frontend com streaming, gerenciamento de estado e atualizações de UI em tempo real em React, Vue, Next.js e Svelte, de forma semelhante ao assistant-ui. Para times que já trabalham no ecossistema TypeScript e Next.js, o Vercel AI SDK fornece uma maneira rápida e transparente de construir aplicações de IA com experiências ricas no lado do cliente.

## Mantenha-se atualizada com os artigos e informações relacionados ao Radar

**Assine o Technology Radar para receber e-mails mensais com insights de tecnologia da Thoughtworks e divulgações dos futuros volumes.**

Assine