# Technology Radar

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

Volume 31
Outubro 2024

/thoughtworks

Estratégia. Design. Engenharia.

# 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.

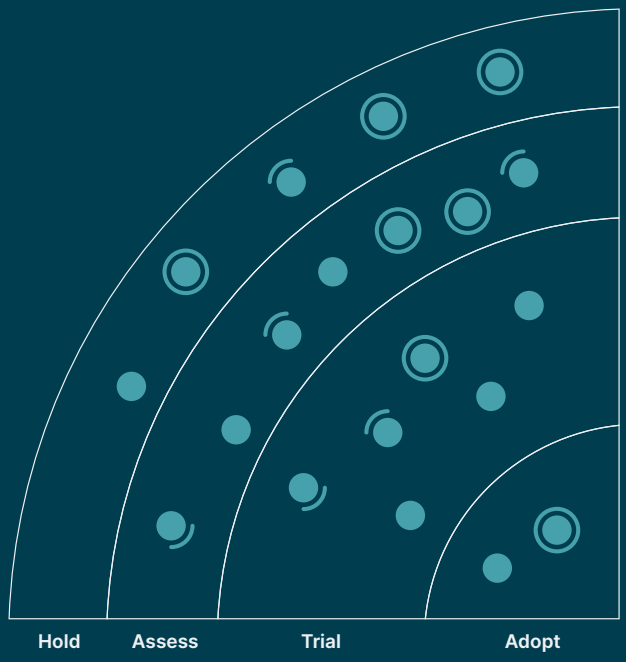
Nós encorajamos você a explorar essas tecnologias. O Radar é gráfico por natureza, agrupando os itens em técnicas, ferramentas, plataformas, linguagens e frameworks. No caso de itens que podem ser classificados em mais de um quadrante, escolhemos aquele que parece mais adequado. Além disso, agrupamos 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 20 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 virtual do TAB realizada em Setembro de 2024.

Rachel Laycock (CTO)

Martin Fowler (Chief Scientist)

Rebecca Parsons (CTO Emerita)

Bharani Subramaniam

Birgitta Böckeler

Camilla Falconi Crispim

Erik Dörnenburg

James Lewis

Ken Mugrage

Maya Ormaza

Mike Mason

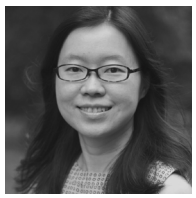
Neal Ford

Pawan Shah

Scott Shaw

Selvakumar Natesan

Shangqi Liu



Sofia Tania

Thomas Squeo

Vanya Seth

Will Amaral

## Tradutoras:

Aloysio Chagas, Arthur Santos, Elisa Sattyam, Gabriela Alves, Giovanni Watanabe, Gisele Hammerschmitt, Guilherme Silveira, Guilherme Vandresen, Francisco Silva, Isaías Barroso, Marcelo Viana, Marcelo Vidal, Nina da Hora, Pietra Freitas, Patrick Prado, Pedro Souza, Rafael Auday, Renan Martins, Thiago Gregorio, Taluna Mendes e Vanessa Cordeiro.

# Temas

## Antipadrões em assistência de codificação

Para a surpresa de ninguém, IA generativa e LLMs dominaram nossas conversas nesta edição do Radar, incluindo padrões emergentes em torno do seu uso por desenvolvedoras. Padrões inevitavelmente levam a antipadrões — situações contextualizadas que as pessoas desenvolvedoras devem evitar. Nós identificamos alguns antipadrões começando a aparecer no espaço hiperativo da IA, incluindo a noção equivocada de que humanos podem substituir totalmente a programação em pares pela IA, a dependência excessiva em sugestões de assistência de codificação, problemas de qualidade com códigos gerados e taxas de crescimento aceleradas nas bases de código. A IA tende a resolver problemas por meio de força bruta em vez de usar abstrações, como utilizar dezenas de condicionais empilhadas em vez do padrão Strategy. Particularmente, os problemas de qualidade de código destacam uma área de esforço contínuo por parte de desenvolvedoras e arquitetas para que não se afoguem em um código terrível mas funcional. Assim, integrantes das equipes devem redobrar as boas práticas de engenharia, como testes unitários, funções de aptidão arquitetural e outras técnicas de governança e validação comprovadas, para garantir que a IA esteja ajudando em seus esforços em vez de complicar suas bases de código com complexidade.

## Rust é tudo, menos ultrapassada

Rust se tornou gradualmente a linguagem de programação de sistemas preferida. Em cada sessão do Radar, ela aparece repetidamente nas entrelinhas de nossas conversas; um grande número das ferramentas que discutimos é escrito em Rust. É a linguagem escolhida ao substituir utilitários de sistema mais antigos e também na categoria de reescrita de parte de um ecossistema para melhorar o desempenho — o epíteto mais comum para ferramentas baseadas em Rust parece ser “incrivelmente rápido”. Por exemplo, notamos várias ferramentas no ecossistema Python que têm alternativas baseadas em Rust, oferecendo desempenho visivelmente melhor. As pessoas que criaram a linguagem e a comunidade conseguiram gerar um ecossistema bem recebido de SDKs, bibliotecas e ferramentas de desenvolvimento, ao mesmo tempo em que proporcionam uma velocidade de execução excepcional com menos armadilhas do que muitos de seus predecessores. Muitas pessoas dos nossos times são fãs do Rust, e parece que a maioria das pessoas desenvolvedoras que a utilizam a tem em alta consideração.

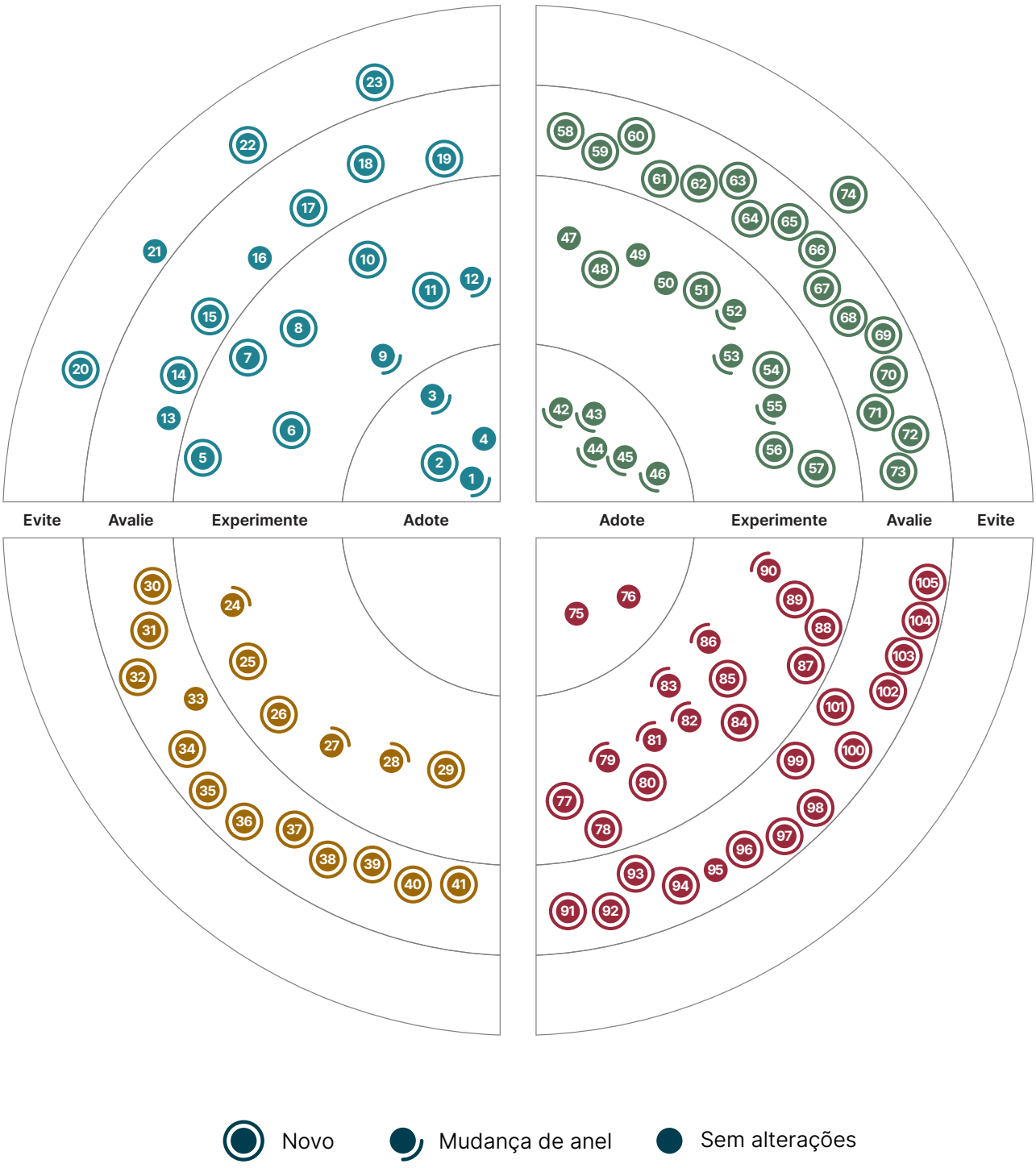
## A ascensão gradual do WASM

WASM (WebAssembly) é um formato de instrução binária para uma máquina virtual baseada em pilha, que soa esotérico e de nível muito baixo para a maioria dos interesses da desenvolvedora até que as pessoas vejam as implicações: a capacidade de executar aplicativos complexos dentro de um sandbox de navegador. WASM pode ser executado dentro de máquinas virtuais JavaScript existentes, permitindo que aplicações, que as desenvolvedoras anteriormente só poderiam implementar em frameworks e extensões nativas, sejam incorporadas em navegadores. Os quatro principais navegadores agora suportam WASM 1.0 (Chrome, Firefox, Safari e Edge), abrindo possibilidades interessantes para o desenvolvimento portátil e a multiplataforma sofisticada. Temos observado esse padrão nos últimos anos com grande interesse, e estamos felizes em vê-lo começar a flexionar suas capacidades como um alvo de implantação legítimo.

## A explosão cambriana de ferramentas de IA generativa

Seguindo a trajetória estabelecida nos últimos volumes do Radar, esperávamos que a IA generativa tivesse um lugar de destaque em nossas discussões. E, ainda assim, ficamos surpresas com a explosão do ecossistema de tecnologia de suporte a modelos de linguagem: guardrails, evals, ferramentas para criar agentes, frameworks para trabalhar com resultados estruturados, bancos de dados vetoriais, serviços de nuvem e ferramentas de observabilidade. De muitas maneiras, esse crescimento rápido e variado faz todo sentido: a experiência inicial, a simplicidade de um prompt de texto simples para um modelo de linguagem, deu lugar à engenharia de produtos de software. Esses produtos podem não estar à altura dos sonhos e das afirmações extravagantes que foram feitas depois que as pessoas enviaram seus primeiros prompts para o ChatGPT, mas acompanhamos o uso sensato e produtivo da IA generativa em muitas de nossas clientes, e todas essas ferramentas, plataformas e frameworks desempenham um papel importante na colocação em produção de soluções baseadas em LLM. Assim como aconteceu com a explosão do ecossistema JavaScript por volta de 2015, esperamos que esse crescimento caótico continue por algum tempo.

# O Radar



# O Radar

## Técnicas

**Adote**

1. 1% canário
2. Testes de componentes
3. Implantação contínua
4. Geração Aumentada por Recuperação (RAG)

**Experimente**

5. Narrativa de domínio
6. Fine-tuning em modelos de embedding
7. Chamada de função com LLMs
8. LLM como juíz
9. Passkeys
10. Modelos de linguagem de pequeno porte
11. Dados sintéticos para teste e treinamento de modelos
12. Usando GenAI para compreender bases de código legadas

**Avalie**

13. Assistentes de equipe baseados em IA
14. Prompt dinâmico de poucos disparos
15. GraphQL para produtos de dados
16. Agentes autônomos impulsionados por LLM
17. Observabilidade 2.0
18. Inferência de LLM em dispositivos
19. Saída estruturada de LLMs

**Evite**

20. Complacência com código gerado por IA
21. Ambientes de testes de integração em toda empresa
22. Proibições de LLM
23. Substituição da programação em pares por IA

## Plataformas

**Adote**

—

**Experimente**

24. Databricks Unity Catalog
25. FastChat
26. GCP Vertex AI Agent Builder
27. Langfuse
28. Qdrant
29. Vespa

**Avalie**

30. Pesquisa de IA do Azure
31. Databricks Delta Live Tables
32. Elastisys Compliant Kubernetes
33. FoundationDB
34. Golem
35. Iggy
36. Iroh
37. Plataformas de modelos de visão ampla (LVM)
38. OpenBCI Galea
39. PGLite
40. SpinKube
41. Unblocked

**Evite**

—

## Ferramentas

**Adote**

42. Bruno
43. K9s
44. SOPS
45. Ferramentas de teste de regressão visual
46. Wiz

**Experimente**

47. AWS Control Tower
48. CCMenu
49. ClickHouse
50. Devbox
51. Difftastic
52. LinearB
53. pgvector
54. Ferramenta de construção Snapcraft
55. Spinnaker
56. TypeScript OpenAPI
57. Unleash

**Avalie**

58. Astronomer Cosmos
59. ColPali
60. Cursor
61. Data Mesh Manager
62. GitButler
63. Assistente de IA JetBrains
64. Mise
65. Mockoon
66. Raycast
67. ReadySet
68. Rspack
69. Semantic Router
70. Agentes de engenharia de software
71. uv
72. Warp
73. Zed

**Evite**

74. CocoaPods

## Linguagens e Frameworks

**Adote**

75. dbt
76. Testcontainers

**Experimente**

77. CAP
78. CARLA
79. Pacotes de ativos do Databricks
80. Instructor
81. Kedro
82. LiteLLM
83. LlamaIndex
84. LLM Guardrails
85. Medusa
86. Pkl
87. ROS 2
88. seL4
89. SetFit
90. vLLM

**Avalie**

91. Apache XTable™
92. dbldatagen
93. DeepEval
94. DSPy
95. Flutter para Web
96. kotaemon
97. Lenis
98. LLMLingua
99. Microsoft Autogen
100. Pingora
101. Ragas
102. Score
103. shadcn
104. Slint
105. SST

**Evite**

—

# Técnicas

**Adote**

1. 1% canário
2. Teste de componentes
3. Implantação contínua
4. Geração Aumentada por Recuperação (RAG)

**Experimente**

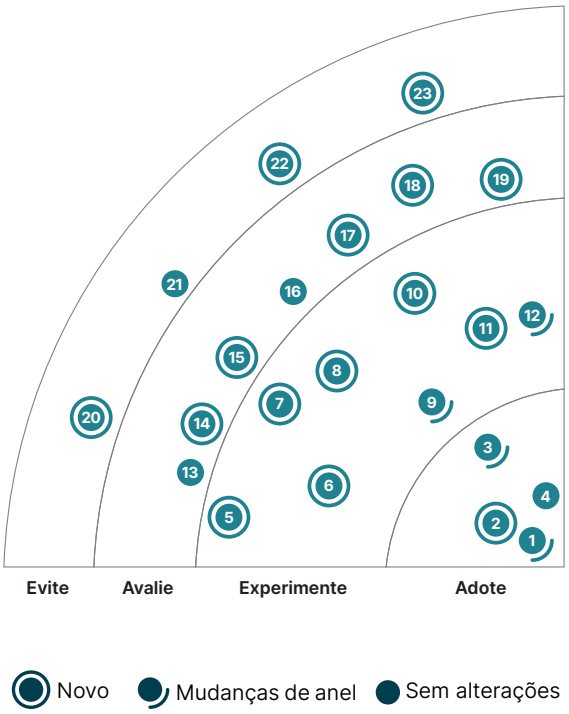
5. Narrativa de domínio
6. Fine-tuning em modelos de embedding
7. Chamada de função com LLMs
8. LLM como juíz
9. Passkeys
10. Modelos de linguagem de pequeno porte
11. Dados sintéticos para teste e treinamento de modelos
12. Usando GenAI para compreender bases de código legadas

**Avalie**

13. Assistentes de equipe baseados em IA
14. Prompt dinâmico de poucos disparos
15. GraphQL para produtos de dados
16. Agentes autônomos impulsionados por LLM
17. Observabilidade 2.0
18. Inferência de LLM em dispositivos
19. Saída estruturada de LLMs

**Evite**

20. Complacência com código gerado por IA
21. Ambientes de testes de integração em toda empresa
22. Proibições de LLM
23. Substituição da programação em pares por IA



## 1. 1% canário

*Adote*

Por muitos anos, temos usado a abordagem de implantação canário para incentivar o feedback inicial sobre novas versões de software, ao mesmo tempo em que reduzimos o risco por meio da implementação incremental para usuárias selecionadas. O 1% canary é uma técnica útil em que lançamos novos recursos para um segmento muito pequeno (digamos 1%) de usuárias cuidadosamente escolhidas dentre várias categorias. Isso permite que as equipes capturem feedback rápido da usuária e observem o impacto dos novos lançamentos em categorias como desempenho e estabilidade, aprendendo e respondendo conforme necessário. Essa técnica se torna especialmente crucial quando as equipes estão lançando atualizações de software para aplicativos móveis ou uma frota de dispositivos como os de computação de ponta ou veículos definidos por software. Com observabilidade adequada e feedback precoce, ela oferece a oportunidade de conter o raio de impacto em caso de cenários inesperados em produção. Embora a implantação canário possa ser útilas implantações canário possam ser úteis para obter feedback mais rápido de usuárias, acreditamos que começar com uma pequena porcentagem de usuárias também é obrigatório para reduzir e conter o risco de lançamentos de funcionalidades em grande escala.

## 2. Testes de componentes

*Adote*

O teste automatizado continua sendo a base do desenvolvimento de software eficaz. Para testes de frontend, podemos discutir se a distribuição de diferentes tipos de teste deve ser a pirâmide de teste clássica ou se deve ter o formato de um troféu. Em ambos os casos, porém, as equipes devem se concentrar em teste de componentes, pois os conjuntos de testes devem ser estáveis e executados rapidamente. Em vez disso, o que estamos notando é que as equipes abrem mão de dominar os testes de componentes, em favor de testes ponta a ponta baseados em navegador, bem como testes unitários definidos de forma muito restrita. Os testes unitários tendem a forçar os componentes a expor o que deveria ser uma funcionalidade puramente interna, enquanto os testes baseados em navegador são lentos, mais instáveis e mais difíceis de depurar. Nossa recomendação é ter uma quantidade significativa de testes de componentes e usar uma biblioteca como jsdom para executar os testes de componentes na memória. Ferramentas de navegador como Playwright ainda têm um lugar em testes ponta a ponta, mas não devem ser usadas para testes de componentes.

## 3. Implantação contínua

*Adote*

Acreditamos que as organizações devem adotar práticas de implantação contínua sempre que possível. Implantação contínua é a prática de implantar automaticamente em produção todas as mudanças que passam nos testes automatizados. Essa prática é um facilitador chave para ciclos de feedback rápidos e permite que as organizações entreguem valor às clientes de forma mais rápida e eficiente. A implantação contínua difere da entrega contínua no sentido de que apenas requer que o código possa ser implantado a qualquer momento, já que não exige que cada mudança realmente seja implantada em produção. Hesitamos em mover a implantação contínua para o anel de Adoção no passado, pois é uma prática que requer um alto nível de maturidade em outras áreas de entrega de software e, portanto, não é apropriada para todas as equipes. No entanto, o recente livro da Thoughtworker Valentina Servile, *Continuous Deployment*, fornece um guia abrangente para implementar a prática em uma organização. Ele oferece um roteiro para as organizações seguirem a fim de alcançar o nível de maturidade necessário para adotar práticas de implantação contínua.

## 4. Geração Aumentada por Recuperação (RAG)

*Adote*

Geração aumentada por recuperação (RAG) é o padrão preferido por nossas equipes para melhorar a qualidade das respostas geradas por um modelo de linguagem de grande porte (LLM). Nós a usamos com sucesso em muitos projetos, incluindo a plataforma Jugalbandi AI. Com RAG, informações sobre documentos relevantes e confiáveis são armazenadas em um banco de dados. Para um determinado prompt, o banco de dados é consultado, documentos relevantes são recuperados, e o prompt é aumentado com o conteúdo dos documentos, fornecendo assim um contexto mais rico ao LLM. Isso resulta em uma saída de maior qualidade e alucinações drasticamente reduzidas. A janela de contexto — que determina o tamanho máximo da entrada do LLM — cresceu significativamente com os modelos mais recentes, mas selecionar os documentos mais relevantes ainda é uma etapa crucial. Nossa experiência indica que um contexto menor cuidadosamente construído pode produzir melhores resultados do que um contexto amplo e grande. Usar um contexto grande também é mais lento e mais caro. Costumávamos confiar apenas em embeddings armazenados em um banco de dados vetorial para identificar contexto adicional. Agora, estamos vendo reclassificação e busca híbrida: ferramentas de busca como Elasticsearch Relevance Engine, bem como abordagens como GraphRAG que utilizam grafos de conhecimento criados com a ajuda de um LLM. Uma abordagem baseada em grafos funcionou particularmente bem em nosso trabalho de compreensão de bases de código legadas com GenAI.

## 5. Narrativa de domínio

*Experimente*

O design orientado a domínio (DDD) se tornou uma abordagem fundamental na forma como desenvolvemos software. Nós o utilizamos para modelar eventos, orientar designs de software, estabelecer limites de contexto em torno de microsserviços e elaborar requisitos de negócios detalhados. O DDD estabelece uma linguagem ubíqua que tanto as stakeholders não técnicas quanto as pessoas desenvolvedoras de software podem usar para se comunicar de forma eficaz sobre o negócio. Uma vez estabelecidos, os modelos de domínio evoluem, mas muitas equipes acham difícil começar com o DDD. Não há uma abordagem única para construir um modelo de domínio inicial. Uma técnica promissora que encontramos recentemente é a narrativa de domínio. Ela é uma técnica de facilitação na qual especialistas do negócio são incentivados a descrever atividades do negócio. À medida que as especialistas são guiadas em sua narrativa, uma facilitadora utiliza uma linguagem pictográfica para capturar as relações e ações entre entidades e atores. O processo de tornar essas histórias visíveis ajuda a esclarecer e desenvolver um entendimento compartilhado entre as participantes. Como não há uma única abordagem ideal para desenvolver um modelo de domínio, a narrativa de domínio oferece uma alternativa notável ou, para uma abordagem mais abrangente ao DDD, pode ser um complemento ao Event Storming, outra técnica que frequentemente usamos para começar com o DDD.

## 6. Fine-tuning em modelos de embedding

*Experimente*

Ao construir aplicações de LLM baseadas em geração aumentada por recuperação (RAG), a qualidade dos embeddings impacta diretamente tanto na recuperação dos documentos relevantes quanto na qualidade da resposta. O fine-tuning em modelos de embedding pode aumentar a precisão e a relevância dos embeddings para tarefas ou domínios específicos. Nossas equipes realizaram ajustes finos dos embeddings ao desenvolver aplicações de LLM específicas para domínios onde a extração de informações precisas era crucial. No entanto, considere os contrapontos dessa abordagem antes de se apressar para ajustar seu modelo de embedding.

## 7. Chamada de função por LLMs

*Experimente*

A chamada de função com LLMs refere-se à capacidade de integrar LLMs com funções externas, APIs ou ferramentas, determinando e invocando a função apropriada com base em uma determinada consulta e na documentação associada. Isso amplia a utilidade dos LLMs para além da geração de texto, permitindo que eles executem tarefas específicas, como recuperação de informações, execução de código e interação com APIs. Ao acionar funções externas ou APIs, os LLMs podem executar ações que antes estavam fora de seus recursos autônomos. Essa técnica permite que os LLMs atuem em seus resultados, preenchendo efetivamente a lacuna entre o pensamento e a ação, de forma muito semelhante à maneira como as pessoas usam ferramentas para realizar várias tarefas. Ao introduzir a chamada de função, os LLMs acrescentam determinismo e factualidade ao processo de geração, atingindo um equilíbrio entre criatividade e lógica. Esse método permite que os LLMs se conectem a sistemas e bancos de dados internos ou até mesmo realizem pesquisas na Internet por meio de navegadores conectados. Modelos como a série GPT da OpenAI suportam chamadas de função e modelos refinados, como o Gorilla, são projetados especificamente para aprimorar a precisão e a consistência da geração de chamadas de API executáveis a partir de instruções de linguagem natural. Como técnica, a chamada de função se encaixa na geração aumentada por recuperação (RAG) e nas arquiteturas de agentes. Ela deve ser vista como um padrão abstrato de uso, enfatizando seu potencial como uma ferramenta fundamental em diversas implementações, em vez de uma solução específica.

## 8. LLM como juíz

*Experimente*

Muitos sistemas que construímos possuem duas características principais: serem capazes de prover uma resposta baseada em questões sobre um grande conjunto de dados e quase impossíveis de acompanhar como chegaram a essa resposta. Apesar desta opacidade, nós ainda queremos avaliar e melhorar a qualidade das respostas. Com o padrão de LLM como juíz, usamos uma LLM para avaliar as respostas de outros sistemas, que por sua vez pode ser baseado em um LLM. Notamos esse padrão ser utilizado para avaliar a relevância dos resultados de pesquisa em um catálogo de produtos e para avaliar quando um chatbot baseado em LLM guiou suas usuárias em uma direção sensata. Naturalmente, o sistema avaliador deve ser configurado e calibrado cuidadosamente. Isto pode gerar ganhos significativos, o que, por sua vez, se traduz em custos menores. Esta é uma área de pesquisa em andamento, tendo seu estado atual resumido neste artigo.

## 9. Passkeys

*Experimente*

Orientadas pela aliança FIDO e apoiadas pela Apple, Google e Microsoft, as passkeys estão se aproximando da usabilidade convencional. A configuração de um novo login com passkeys gera um par de chaves: o site recebe a chave pública e a usuária fica com a chave privada. O processamento do login usa criptografia assimétrica. A usuária prova que está de posse da chave privada, que é armazenada no dispositivo da usuária e nunca é enviada ao site. O acesso às senhas é protegido por meio de biometria ou de um PIN. As chaves de acesso podem ser armazenadas e sincronizadas dentro dos grandes ecossistemas de tecnologia, usando o iCloud Keychain da Apple, o Password Manager do Google ou o Windows Hello. Para usuárias de várias plataformas, o Client to Authenticator Protocol (CTAP) possibilita que as senhas sejam mantidas em um dispositivo diferente daquele que cria a chave ou que precisa dela para o login. A objeção mais comum ao uso de chaves de acesso

alega que elas são um desafio para as usuárias menos experientes em tecnologia, o que acreditamos ser contraditório. Em geral, essas são as mesmas usuárias que têm pouca disciplina com senhas e que, portanto, se beneficiam mais com métodos alternativos. Na prática, os sistemas que usam passkeys podem recorrer a métodos de autenticação mais tradicionais, se necessário.

## 10. Modelos de linguagem de pequeno porte

***Experimente***

Modelos de linguagem de grande porte (LLMs) têm se provado muito úteis em várias áreas de aplicação, mas o fato de serem grandes pode ser uma fonte de problemas: responder a um prompt requer grandes recursos computacionais, tornando as consultas lentas e caras; os modelos são proprietários e tão grandes que eles devem ser hospedados em uma nuvem por um terceiro, o que pode ser problemático quanto a dados sensíveis; e treinar um modelo é proibitivamente caro na maioria dos casos. Esse último problema pode ser resolvido com o padrão RAG, que contorna a necessidade de treinar e otimizar modelos fundamentais, mas preocupações quanto ao custo e a privacidade geralmente persistem. Em resposta, temos identificado um crescente interesse em modelos de linguagem de pequeno porte (SLMs). Em comparação ao seu irmão mais popular, eles têm menos peso e menor precisão, geralmente entre 3,5 e 10B de parâmetros. Pesquisas recentes sugerem que, no contexto correto, quando configurados corretamente, SLMs podem performar ou até mesmo superar os LLMs. E seu tamanho torna possível rodá-los em dispositivos de borda. Nós mencionamos anteriormente o Gemini Nano da Google, mas o cenário está evoluindo rapidamente, com a Microsoft introduzindo a série Phi-3, por exemplo.

## 11. Dados sintéticos para teste e treinamento de modelos

***Experimente***

A criação de conjuntos de dados sintéticos envolve a geração de dados artificiais que podem imitar cenários do mundo real sem depender de fontes de dados sensíveis ou de acesso limitado. Embora os dados sintéticos para conjuntos de dados estruturados tenham sido explorados extensivamente (por exemplo, para testes de desempenho ou ambientes seguros em termos de privacidade), estamos notando um uso renovado de dados sintéticos para dados não estruturados. As empresas frequentemente enfrentam dificuldades com a falta de dados rotulados específicos do domínio, especialmente para uso no treinamento ou ajuste fino de LLMs. Ferramentas como Bonito e AgentInstruct da Microsoft podem gerar dados sintéticos de ajuste de instrução a partir de fontes brutas, como documentos de texto e arquivos de código. Isso ajuda a acelerar o treinamento do modelo, reduzindo custos e a dependência da curadoria manual de dados. Outro caso de uso importante é a geração de dados sintéticos para abordar dados desbalanceados ou esparsos, o que é comum em tarefas como detecção de fraudes ou segmentação de clientes. Técnicas como SMOTE ajudam a equilibrar conjuntos de dados criando artificialmente instâncias de classes minoritárias. Da mesma forma, em indústrias como a financeira, redes adversárias generativas (GANs) são usadas para simular transações raras, permitindo que os modelos sejam robustos na detecção de casos de borda e melhorando o desempenho geral.

## 12. Usando GenIA para compreender bases de código legadas

***Experimente***

A IA generativa (GenIA) e os modelos de linguagem de grande porte (LLMs) podem auxiliar pessoas desenvolvedoras a escrever e entender código. O auxílio na compreensão de código é especialmente útil em caso de bases de código legadas com documentação escassa, desatualizada e errônea. Desde a última vez que escrevemos sobre isso, as técnicas e produtos para usar GenIA para entender

bases de código legadas evoluiram ainda mais, e nós temos utilizado com sucesso algumas delas na prática, principalmente para auxiliar nos esforços de engenharia reversa para modernização de mainframes. Uma técnica particularmente promissora que temos utilizado é a geração aumentada por recuperação (RAG), uma abordagem onde a recuperação da informação é feita em um grafo de conhecimento da base de código. O grafo de conhecimento consegue preservar as informações estruturais sobre a base de código que vão além do que uma LLM pode extrair apenas do código textual. Isso é especialmente útil em bases de código legadas que são menos autodescritivas e coesas. Uma oportunidade adicional para melhorar a compreensão do código é que o grafo pode ser enriquecido com a documentação existente e gerada por IA, dependências externas, conhecimento do domínio do negócio ou qualquer outro recurso disponível que possa facilitar o trabalho da IA.

## 13. Assistentes de equipe baseados em IA

***Avalie***

As ferramentas de assistência à codificação baseadas em IA são geralmente discutidas no contexto de apoiar e aprimorar o trabalho individual. No entanto, a entrega de software é e continuará sendo um trabalho em equipe. Por isso, você deve procurar maneiras de criar assistentes de equipe baseados em IA que ajudem a criar a equipe 10x, o oposto de pessoas engenheiras 10x isoladas e habilitadas por IA. Felizmente, desenvolvimentos recentes no mercado de ferramentas estão nos aproximando de tornar isso uma realidade. Unblocked é uma plataforma que reúne todas as fontes de conhecimento de uma equipe e as integra de forma inteligente às ferramentas dos membros da equipe. E o Rovo da Atlassian traz a IA para a plataforma de colaboração em equipe mais utilizada, oferecendo às equipes novos tipos de pesquisa e acesso à sua documentação, além de desbloquear novas formas de automação e suporte a práticas de software com agentes Rovo. Enquanto esperamos que o mercado evolua ainda mais nesse espaço, temos explorado o potencial da IA para amplificação de conhecimento e suporte a práticas de equipe por conta própria: disponibilizamos como código aberto, o Haiven, nosso assistente à equipe baseado em IA e começamos a reunir aprendizados com assistência de IA para tarefas não relacionadas à codificação, como a análise de requisitos.

## 14. Prompt dinâmico de poucos disparos

***Avalie***

Prompt dinâmico de poucos disparos aprimora o conceito de few-shot prompting ao incluir dinamicamente exemplos específicos no prompt para guiar as respostas do modelo. Ajustar o número e a relevância desses exemplos otimiza o comprimento e a pertinência do contexto, melhorando assim a eficiência e o desempenho do modelo. Bibliotecas como scikit-llm implementam essa técnica utilizando a busca pelos casos mais similares para encontrar os exemplos mais relevantes alinhados com a consulta da usuária. Essa técnica permite uma melhor utilização da janela de contexto limitada do modelo e reduz o consumo de tokens. O gerador de código aberto do SQL, vanna, utiliza o dynamic few-shot prompting para aprimorar a precisão das respostas.

## 15. GraphQL para produtos de dados

***Avalie***

GraphQL para produtos de dados é a técnica de usar GraphQL como uma porta de saída para produtos de dados buscando o consumo do produto pela cliente. Nós citamos GraphQL como um protocolo de API e como ele permite com que desenvolvedoras criem uma camada de API unificada que abstrai a complexidade dos dados subjacentes, concedendo uma interface mais

coesa e gerenciável para clientes. GraphQL para produtos de dados torna fácil para consumidoras descobrirem o formato dos dados e suas relações com o esquema do GraphQL, além de fazer uso de ferramentas que são conhecidas por esse público. Os nossos times estão explorando essa técnica em usos específicos como o de conversas sobre dados para analisar e descobrir insights de big data com a ajuda de grandes modelos de linguagem, onde as consultas GraphQL são construídas por LLMs com base no prompt da usuária e o esquema do GraphQL é usado nos prompts LLM para referência.

## 16. Agentes autônomos impulsionados por LLM

***Avalie***

Agentes autônomos com tecnologia LLM estão evoluindo além de agentes únicos e sistemas multi-agentes estáticos com o surgimento de frameworks como o Autogen e o CrewAI. Essa técnica permite que desenvolvedoras dividam uma atividade complexa em várias tarefas menores realizadas por agentes, onde cada agente é designado a uma função específica. Desenvolvedoras podem utilizar ferramentas pré-configuradas para realizar a tarefa, e os agentes conversam entre si e orquestram o fluxo. A técnica ainda está em estágio inicial de desenvolvimento. Em nossos experimentos até agora, nossos times têm encontrado problemas como agentes entrando em loops contínuos e comportamento descontrolado. Bibliotecas como LangGraph oferecem um maior controle das interações do agente com a habilidade de definir o fluxo como um gráfico. Se você utiliza esta técnica, nós sugerimos implementar mecanismos de segurança contra falhas, incluindo timeouts e supervisão humana.

## 17. Observabilidade 2.0

***Avalie***

Observabilidade 2.0 representa uma mudança de ferramentas de monitoramento tradicionais e díspares para uma abordagem unificada que promove dados de eventos estruturados e de alta cardinalidade em um único armazenamento de dados. Este modelo captura eventos ricos em sua forma bruta, com metadados detalhados, para oferecer uma única fonte de verdade para uma análise abrangente. Ao armazenar eventos em sua forma bruta, ele simplifica a correlação e oferece suporte a análises em tempo real e forense, além de permitir insights mais profundos em sistemas complexos e distribuídos. A abordagem permite monitoramento de alta resolução e capacidades de investigação dinâmica. Observabilidade 2.0 prioriza a captura de dados de alta cardinalidade e alta dimensionalidade, permitindo um exame detalhado sem gargalos de desempenho. O armazenamento de dados unificado reduz a complexidade, oferecendo uma visão coerente do comportamento do sistema e alinhando as práticas de observabilidade mais de perto com o ciclo de vida do desenvolvimento de software.

## 18. Inferência de LLM em dispositivos

***Avalie***

Modelos de linguagem de grande porte (LLMs) agora podem ser executados em navegadores da internet e em dispositivos móveis como smartphones e laptops, permitindo a construção de aplicações de IA em dispositivos. Isso proporciona uma manipulação segura de dados sensíveis, eliminando a necessidade de transferi-los para a nuvem, a baixa latência para tarefas computacionais e processamento em tempo real de imagens e vídeo nos dispositivos, e a redução de custos pela execução computacional local e funcionamento das aplicações mesmo quando a conectividade à internet é instável ou indisponível. Esse é um campo de pesquisa ativo e em crescimento. No

passado, nós destacamos MLX, uma ferramenta de código aberto para aprendizado de máquina eficiente em processadores Apple. Outras ferramentas emergentes incluem Transformers.js e Chatty. Transformers.js possibilita a execução de Transformers em navegadores usando ONNX Runtime, dando suporte a modelos convertidos de PyTorch, TensorFlow e JAX. Chatty, por sua vez, roda LLMs de forma nativa e privada nos navegadores através da WebGPU, enriquecendo a experiência de IA com uma gama completa de funcionalidades.

## 19. Saída estruturada de LLMs

***Avalie***

Saída estruturada de LLMs se refere a prática de restringir a resposta de um modelo de linguagem a um esquema definido. Isso pode ser alcançado tanto através de um modelo generalizado, para que ele responda em um formato específico, quanto ajustando um modelo para que ele nativamente produza um JSON, por exemplo. A OpenAI agora suporta saídas estruturadas, permitindo que as desenvolvedoras forneçam um esquema JSON, pydantic ou um objeto Zod para restringir as respostas do modelo. Essa capacidade é especialmente valiosa para permitir chamadas de função, integração com APIs e realizando integrações externas, onde a precisão e conformidade com o formato são críticas. A saída estruturada não apenas aprimora a maneira como as LLMs podem interagir com o código, mas também suporta casos de uso mais amplos, como a geração de marcação para renderizar gráficos. Além disso, a saída estruturada demonstrou reduzir a probabilidade de erros nas respostas do modelo.

## 20. Complacência com código gerado por IA

***Evite***

Assistentes de codificação de IA como GitHub Copilot e Tabnine tornaram-se muito populares. De acordo com a pesquisa de desenvolvedoras da StackOverflow em 2024, 72% de todas as pessoas entrevistadas são favoráveis ou muito favoráveis às ferramentas de IA para desenvolvimento. Embora também vejamos seus benefícios, estamos cautelosas sobre o impacto de médio a longo prazo que isso terá na qualidade do código e alertamos as desenvolvedoras sobre a complacência com o código gerado por IA. É muito tentador sermos menos vigilantes ao revisar sugestões de IA após algumas experiências positivas com um assistente. Estudos como este da GitClear mostram uma tendência de bases de código crescendo mais rapidamente, o que suspeitamos coincidir com pull requests maiores. E este estudo do GitHub nos faz questionar se o aumento mencionado de 15% na taxa de merge de pull requests é realmente uma coisa boa ou se as pessoas estão mesclando pull requests maiores mais rapidamente porque confiam demais nos resultados da IA. Ainda estamos usando o conselho que compartilhamos há mais de um ano de como iniciar o uso, que é ter cuidado com o viés de automação, a falácia do custo irrecuperável, o viés de ancoragem e a fadiga de revisão. Também recomendamos que as programadoras desenvolvam uma boa estrutura mental sobre onde e quando não usar e confiar na IA.

## 21. Ambientes de testes de integração em toda empresa

***Evite***

A criação de ambientes de testes de integração em toda empresa é uma prática comum e ineficiente que deixa tudo mais lento. Esses ambientes invariavelmente se tornam um recurso precioso, difícil de replicar e um gargalo no desenvolvimento. Eles também dão uma falsa sensação de segurança devido a inevitável discrepância nos dados e na sobrecarga de configuração entre os ambientes.

Ironicamente, uma objeção comum às alternativas — como ambientes efêmeros ou múltiplos ambientes de testes locais — é o custo. No entanto, isso não leva em conta o custo de atraso causado pela utilização de ambientes de testes de integração em toda companhia, como times de desenvolvimento esperando por outros times terminarem ou por novas versões de sistemas dependentes serem implementados. Em vez disso, as equipes devem utilizar ambientes efêmeros e, preferencialmente, uma suíte de testes mantida pela equipe de desenvolvimento, que pode ser criada e descartada de forma barata, usando stubs falsos para seus sistemas em vez de réplicas reais. Para outras técnicas que suportam esta alternativa veja sobre testes de contrato, desacoplamento da implantação do lançamento, foco no tempo médio de recuperação e testes em produção.

## 22. Proibições de LLM

***Evite***

Em vez de instituir proibições gerais de LLM no local de trabalho, as organizações devem se concentrar em fornecer acesso a um conjunto aprovado de ferramentas de IA. Uma proibição apenas leva as funcionárias a encontrarem soluções alternativas não aprovadas e potencialmente inseguras, criando riscos desnecessários. Assim como nos primeiros dias da computação pessoal, as pessoas usarão quaisquer ferramentas que considerem eficazes para realizar seu trabalho, independentemente das barreiras existentes. Ao não fornecer uma alternativa segura e endossada, as empresas correm o risco de que as funcionárias usem LLMs não aprovados, o que traz riscos de propriedade intelectual, vazamento de dados e responsabilidade legal. Em vez disso, oferecer LLMs ou ferramentas de IA seguras e aprovadas pela empresa garante tanto a segurança quanto a produtividade. Uma abordagem bem governada permite que as organizações gerenciem preocupações com privacidade de dados, segurança, conformidade e custos, ao mesmo tempo em que empoderam as funcionárias com as capacidades que LLMs oferecem. No melhor cenário, o acesso bem gerenciado às ferramentas de IA pode acelerar o aprendizado organizacional sobre as melhores maneiras de usar a IA no local de trabalho.

## 23. Substituição da programação em pares por IA

***Evite***

Quando as pessoas falam sobre assistentes de programação, o tópico programação em pares inevitavelmente aparece. Nossa profissão tem uma relação de amor e ódio com isso: algumas pessoas acreditam veemente, outras não suportam. Assistentes de programação agora imploram pela pergunta: pode um ser humano parear com uma IA, ao invés de outro ser humano, e obter os mesmos resultados para o time? GitHub Copilot chama a si mesmo deseu programador em par de IA. Enquanto acreditamos que um assistente de programação pode trazer alguns dos benefícios da programação em pares, nós aconselhamos totalmente contra substituir programação em pares por IA. Visualizar assistentes de código para o pareamento ignora um dos benefícios chave da programação em pares: fazer o time, não apenas os indivíduos que contribuem, melhor. Assistentes de código podem oferecer benefícios para se desbloquear, aprender sobre uma nova tecnologia, integrar ou tornar o trabalho tático mais rápido para que possamos focar em design estratégico. Mas eles não ajudam com nenhum dos benefícios de colaboração de time, como manter em baixo nível o trabalho em progresso, reduzir handoffs e reaprendizados, tornando a integração contínua possível ou melhorando a propriedade coletiva de código.

# Plataformas

**Adote**

—

**Experimente**

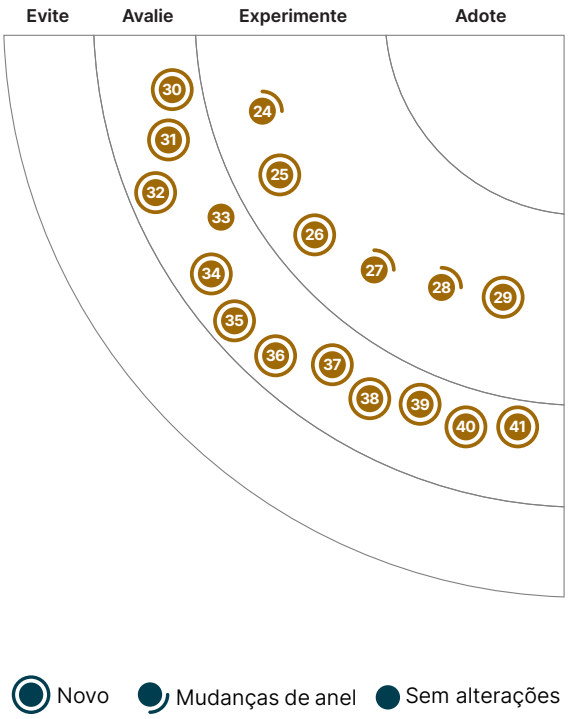
24. Databricks Unity Catalog
25. FastChat
26. GCP Vertex AI Agent Builder
27. Langfuse
28. Qdrant
29. Vespa

**Avalie**

30. Pesquisa de IA do Azure
31. Databricks Delta Live Tables
32. Elastisys Compliant Kubernetes
33. FoundationDB
34. Golem
35. Iggy
36. Iroh
37. Plataformas de modelos de visão ampla (LVM)
38. OpenBCI Galea
39. PGLite
40. SpinKube
41. Unblocked

**Evite**

—



## 24. Databricks Unity Catalog

***Experimente***

Databricks Unity Catalog é uma solução de governança de dados para ativos como arquivos, tabelas ou modelos de aprendizado de máquina em um lakehouse. É uma versão gerenciada do Unity Catalog de código aberto que pode ser usada para governar e consultar dados mantidos em armazenamentos externos ou sob gerenciamento do Databricks. No passado, nossas equipes trabalharam com uma variedade de soluções de gerenciamento de dados, como Hive metastore ou Microsoft Purview. No entanto, o suporte combinado do Unity Catalog para governança, gerenciamento de metastore e descoberta de dados o torna atraente porque reduz a necessidade de gerenciar múltiplas ferramentas. Uma complicação que nossa equipe descobriu é a falta de recuperação automática de desastres no Unity Catalog gerenciado pelo Databricks. Conseguiram configurar sua própria funcionalidade de backup e restauração, mas uma solução fornecida pelo Databricks teria sido mais conveniente. Observe que, embora essas plataformas de governança geralmente implementam uma solução centralizada para garantir consistência entre espaços e cargas de trabalho, a responsabilidade de governar ainda pode ser federada, permitindo que equipes individuais governem seus próprios ativos.

## 25. FastChat

***Experimente***

FastChat é uma plataforma aberta para treinamento, disponibilização e avaliação de grandes modelos de linguagem (LLMs). Nossas equipes usam seus recursos de fornecimento de modelos para hospedar vários modelos — Llama 3.1 (8B e 70B), Mistral 7B e Llama-SQL — para diferentes finalidades, tudo em um formato consistente de API OpenAI. O FastChat opera em uma arquitetura de controlador-agente, permitindo que vários agentes hospedem modelos diferentes. Ele oferece suporte a tipos de agentes, como vLLM, LiteLLM e MLX. Usamos vLLM model workers por seus recursos de alta taxa de transferência. Dependendo do caso de uso (latência ou taxa de transferência), diferentes tipos de agentes modelo do FastChat podem ser criados e dimensionados. Por exemplo, o modelo usado para sugestões de código em IDEs de desenvolvedoras exige baixa latência e pode ser dimensionado com vários agentes FastChat para lidar com solicitações simultâneas de forma eficiente. Por outro lado, o modelo usado para Text-to-SQL não precisa de vários agentes devido à menor demanda ou a diferentes requisitos de desempenho. Nossas equipes aproveitam os recursos de dimensionamento do FastChat para testes A/B. Configuramos os agentes do FastChat com o mesmo modelo, mas com valores de hiperparâmetro diferentes, e fazemos perguntas idênticas a cada um deles, identificando os valores ideais de hiperparâmetro. Ao fazer a transição de modelos em tempo real, realizamos testes A/B para garantir uma migração perfeita. Por exemplo, recentemente migramos do CodeLlama 70B para o Llama 3.1 70B para sugestões de código. Executando os dois modelos simultaneamente e comparando os resultados, verificamos que o novo modelo atendia ou excedia o desempenho do modelo anterior sem interromper a experiência da pessoa desenvolvedora.

## 26. GCP Vertex AI Agent Builder

***Experimente***

GCP Vertex AI Agent Builder oferece uma plataforma flexível para a criação de agentes de IA utilizando linguagem natural ou uma abordagem baseada primariamente em código. A ferramenta se integra de forma transparente com dados empresariais por meio de conectores de terceiros e possui todas as ferramentas necessárias para construir, prototipar e implantar agentes de IA. À medida que a demanda por agentes de IA cresce, muitas equipes enfrentam dificuldades em compreender seus

benefícios e sua implementação. O GCP Vertex AI Agent Builder facilita para as desenvolvedoras a prototipagem rápida de agentes e o gerenciamento de tarefas complexas com dados, exigindo apenas um simples setup. Nossas desenvolvedoras consideraram a ferramenta particularmente útil para a construção de sistemas baseados em agentes, como bases de conhecimento ou sistemas automatizados de suporte, que gerenciam de forma eficiente tanto dados estruturados quanto não estruturados. Isso a torna uma ferramenta valiosa para o desenvolvimento de soluções impulsionadas por IA.

## 27. Langfuse

***Experimente***

LLMs funcionam como caixas pretas, tornando difícil determinar seu comportamento. A observabilidade é crucial para abrir essa caixa preta e entender como os aplicativos LLM operam em produção. Nossas equipes tiveram experiências positivas usando Langfuse para observar, monitorar e avaliar aplicativos baseados em LLM. Suas capacidades de rastreamento, análise e avaliação nos permitem não apenas analisar o desempenho e a precisão da conclusão, mas também gerenciar custos e latência. Além disso, também permite entender padrões de uso da produção, facilitando melhorias contínuas e orientadas a dados. Os dados de instrumentação fornecem rastreabilidade completa do fluxo de solicitação-resposta e das etapas intermediárias, que podem ser usados como dados de teste para validar o aplicativo antes de implementar novas alterações. Utilizamos o Langfuse com RAG (geração aumentada por recuperação), entre outras arquiteturas LLM, e agentes autônomos impulsionados por LLM. Em um aplicativo baseado em RAG, por exemplo, a análise de rastreamentos de conversas com baixa pontuação ajuda a identificar quais partes da arquitetura (pré-recuperação, recuperação ou geração) precisam de refinamento. Outra opção que vale a pena considerar neste espaço é Langsmith.

## 28. Qdrant

***Experimente***

Qdrant é um mecanismo de busca por similaridade vetorial e banco de dados de código aberto escrito em Rust. Ele suporta uma ampla gama de texto e um multimodal denso de modelos de vetores embedding. Nossas equipes têm usado embeddings de código aberto como MiniLM-v6 e BGE para várias bases de conhecimento de produtos. Usamos o Qdrant como um armazenamento de vetores corporativo com multi-tenancy para organizar vetores embeddings como coleções separadas, isolando a base de conhecimento de cada produto no armazenamento. As políticas de acesso da usuária são gerenciadas na camada do aplicativo.

## 29. Vespa

***Experimente***

Vespa é um mecanismo de busca de código aberto e uma plataforma de processamento de big data. É particularmente recomendado para aplicativos que exigem baixa latência e alto rendimento. Nossos times gostam da capacidade do Vespa de implementar pesquisa híbrida usando várias técnicas de recuperação, filtrar e classificar com eficiência muitos tipos de metadados, implementar classificação em várias fases, indexar vários vetores (por exemplo, para cada pedaço) por documento sem duplicar todos os metadados em documentos indexados separadamente e recuperar dados de vários campos indexados de uma só vez.

## 30. Pesquisa de IA do Azure

***Avalie***

A Pesquisa de IA do Azure, anteriormente conhecida como Pesquisa Cognitiva do Azure, é considera um serviço de busca baseado em nuvem projetado para lidar com dados estruturados e não estruturados para aplicações como bases de conhecimento, particularmente em geração aumentada por recuperação (RAG). Ela suporta vários tipos de busca, incluindo busca por palavras-chave, vetorial e híbrida, que acreditamos que se tornarão cada vez mais importantes. O serviço ingere automaticamente formatos comuns de dados não estruturados, incluindo PDF, DOC e PPT, simplificando o processo de criação de conteúdo pesquisável. Além disso, ele se integra com outros serviços do Azure, como o Azure OpenAI, permitindo que as usuárias construam aplicações com o mínimo de esforço de integração manual. Com base em nossa experiência, a Pesquisa de IA do Azure tem um desempenho confiável e é bem adequado para projetos hospedados no ambiente Azure. Através de suas habilidades personalizadas, as usuárias também podem definir etapas específicas de processamento de dados. No geral, se você está trabalhando dentro do ecossistema Azure e precisa de uma solução de busca robusta para uma aplicação RAG, vale a pena considerar a Pesquisa de IA do Azure.

## 31. Databricks Delta Live Tables

***Avalie***

O Databricks Delta Live Tables é um framework declarativo projetado para a criação de pipelines de processamento de dados confiáveis, sustentáveis e testáveis. Ele permite que as pessoas engenheiras de dados definam as transformações de dados usando uma abordagem declarativa e gerencia automaticamente a infraestrutura subjacente e o fluxo de dados. Um dos recursos de destaque do Delta Live Tables é a sua robusta capacidade de monitoramento. Ele fornece um DAG (Gráfico acíclico dirigido) de todo o seu pipeline de dados, representando visualmente a movimentação de dados da fonte para as tabelas finais. Essa visibilidade é crucial para pipelines complexos, ajudando engenheiras e cientistas de dados a rastrear a linhagem e as dependências dos dados. O Delta Live Tables está profundamente integrado ao ecossistema Databricks, o que também traz alguns desafios para a customização de interfaces. Recomendamos que as equipes avaliem cuidadosamente a compatibilidade das interfaces de entrada e saída antes de usar o Delta Live Tables.

## 32. Elastisys Compliant Kubernetes

***Avalie***

Elastisys Compliant Kubernetes é uma distribuição especializada do Kubernetes projetada para atender a rigorosos requisitos regulatórios e de conformidade, particularmente para organizações que operam em indústrias altamente regulamentadas, como saúde, finanças e governo. Ele possui processos de segurança automatizados, fornece suporte para múltiplas nuvens e ambientes locais, e é construído sobre uma arquitetura de segurança de confiança zero. A ênfase na conformidade integrada com leis como GDPR e HIPAA e controles como ISO27001 o torna uma opção atraente para empresas que precisam de um ambiente Kubernetes seguro, compatível e confiável.

## 33. FoundationDB

*Avalie*

FoundationDB é um banco de dados multi-modelo, adquirido pela Apple em 2015 e disponibilizado como código aberto em abril de 2018. O núcleo do FoundationDB é um armazenamento chave-valor distribuído, que oferece transações com serialização rigorosa. Desde que o mencionamos pela primeira vez no Radar, ele passou por melhorias significativas, incluindo distribuições inteligentes de dados para evitar escrita em pontos críticos, um novo mecanismo de armazenamento, otimizações de desempenho e suporte à replicação multi-região. Estamos utilizando o FoundationDB em um de nossos projetos em andamento e estamos muito impressionadas pela sua arquitetura desacoplada. Essa arquitetura nos permite escalar diferentes partes do cluster de forma independente. Por exemplo, podemos ajustar o número de logs de transação, servidores de armazenamento e proxies com base em nossa carga de trabalho e hardware específicos. Apesar de seus recursos extensivos, o FoundationDB continua sendo extremamente fácil de executar e operar em grandes clusters.

## 34. Golem

*Avalie*

A computação durável, um movimento recente na computação distribuída, utiliza um estilo de arquitetura de máquina de estado explícito para persistir a memória de servidores serverless para uma melhor tolerância a falhas e a recuperação. O Golem é um dos promotores desse movimento. Esse conceito pode ser aplicado em alguns cenários como em sagas de microsserviços de longa duração ou fluxos de trabalho prolongados na orquestração de agentes de IA. Já avaliamos o Temporal em outros volumes do Radar para propósitos semelhantes, e o Golem é outra opção. Com o Golem, é possível escrever componentes em WebAssembly em qualquer linguagem suportada, além do Golem ser determinístico e oferecer tempos de inicialização rápidos. Acreditamos que o Golem é uma plataforma empolgante que vale a pena ser avaliada.

## 35. Iggy

*Avalie*

Iggy, uma plataforma de streaming de mensagens persistentes escrita em Rust, é um projeto relativamente novo, mas com recursos impressionantes. Ela já oferece suporte a múltiplos streams, tópicos e partições, entrega at-most-once, expiração de mensagens e suporte a TLS sobre os protocolos QUIC, TCP e HTTP. Executando como um único servidor, a Iggy atualmente alcança alta taxa de transferência para operações de leitura e escrita. Com o suporte a clustering e io_uring em desenvolvimento, Iggy pode se tornar uma alternativa potencial ao Kafka.

## 36. Iroh

*Avalie*

Iroh é um sistema de armazenamento de arquivos distribuídos e entrega de conteúdo relativamente novo que foi projetado como a evolução dos sistemas descentralizados existentes como IPFS (InterPlanetary File System). Tanto o Iroh quanto o IPFS podem ser usados para criar redes descentralizadas de armazenamento, compartilhamento e acesso de conteúdo endereçado utilizando identificadores de conteúdo opacos. No entanto, Iroh removeu algumas limitações da implementação do IPFS, como não ter um tamanho máximo de bloco e fornecer um mecanismo de sincronização de dados via reconciliação de conjunto baseada em intervalo sobre documentos. O roadmap do projeto

inclui trazer a tecnologia para o browser via WASM, o que levanta algumas possibilidades intrigantes para construir a descentralização em aplicativos web. Se você não quiser hospedar no Iroh, você pode utilizar seu serviço de nuvem, iroh.network. Já existem vários SDKs disponíveis em uma variedade de idiomas, e um dos objetivos é ser mais amigável e fácil de usar que sistemas IPFS alternativos. Embora o Iroh ainda esteja em seus primeiros dias, vale a pena acompanhá-lo, pois ele pode se tornar um player significativo no meio de armazenamento descentralizado.

## 37. Plataformas de modelos de visão ampla (LVM)

***Avalie***

Os modelos de linguagem de grande porte (LLMs) chamam tanto a nossa atenção atualmente que tendemos a negligenciar os desenvolvimentos em curso nos modelos de visão ampla (LVMs). Esses modelos podem ser usados para segmentar, sintetizar, reconstruir e analisar fluxos de vídeo e imagens, às vezes em combinação com modelos de difusão ou redes neurais convolucionais padrão. Apesar do potencial dos LVMs para revolucionar a forma como trabalhamos com dados visuais, ainda enfrentamos desafios significativos na adaptação e aplicação deles em ambientes de produção. Os dados de vídeo, por exemplo, apresentam desafios de engenharia únicos para coletar dados de treinamento, segmentar e rotular objetos, ajustar modelos e, em seguida, implantar os modelos resultantes e monitorá-los em produção. Assim, enquanto os LLMs se prestam a interfaces de chat simples ou APIs de texto simples, uma engenheira de visão computacional ou cientista de dados deve gerenciar, versionar, anotar e analisar grandes quantidades de dados de vídeo em streaming; este trabalho requer uma interface visual. Plataformas LVM são uma nova categoria de ferramentas e serviços — incluindo V7, Nvidia Deepstream SDK e Roboflow — que estão surgindo para enfrentar esses desafios. Deepstream e Roboflow são particularmente interessantes para nós porque combinam um ambiente de desenvolvimento GUI integrado para gerenciar e anotar fluxos de vídeo com um conjunto de APIs Python, C++ ou REST para invocar os modelos a partir do código da aplicação.

## 38. OpenBCI Galea

***Avalie***

Há um crescente interesse no uso de interfaces cérebro-computador (ICCs) e em seu potencial de aplicação em tecnologias assistivas. Tecnologias não invasivas, como a eletroencefalografia (EEG) e outros sinais eletrofísicos, oferecem uma alternativa de menor risco em relação a implantes cerebrais para aquelas que estão se recuperando de lesões. Estão surgindo plataformas nas quais pesquisadoras e empreendedoras podem criar aplicações inovadoras sem se preocupar com os desafios de processamento de sinais de baixo nível e integração. Alguns exemplos dessas plataformas são Emotive e OpenBCI, que oferecem hardware e software de código aberto para o desenvolvimento de aplicações ICC. O produto mais recente da OpenBCI, o OpenBCI Galea, combina ICC com as capacidades de um headset de realidade virtual (VR). Ele oferece às pessoas desenvolvedoras acesso a uma variedade de fluxos de dados fisiológicos sincronizados no tempo, juntamente com sensores de posicionamento espacial e rastreamento ocular. Essa ampla gama de dados de sensores pode ser usada para controlar uma variedade de dispositivos físicos e digitais. O SDK suporta várias linguagens de programação e disponibiliza os dados dos sensores no Unity ou Unreal. Estamos entusiasmadas em notar essa capacidade oferecida em uma plataforma de código aberto, permitindo que pesquisadoras tenham acesso às ferramentas e aos dados necessários para inovar nesse campo.

## 39. PGLite

*Avalie*

PGLite é uma construção WASM do banco de dados PostgreSQL. Diferentemente de tentativas anteriores que dependiam de uma máquina virtual Linux, o PGLite constrói diretamente o PostgreSQL para WASM, permitindo que você o execute inteiramente no navegador web. Você pode criar um banco de dados temporário na memória ou persisti-lo no disco via indexedDB. Desde a última vez que mencionamos aplicações local-first no Radar, as ferramentas evoluíram consideravelmente. Com Electric e PGLite, você agora pode construir aplicações reativas local-first no PostgreSQL.

## 40. SpinKube

*Avalie*

SpinKube é um runtime serverless de código aberto para WebAssembly no Kubernetes. Embora o Kubernetes ofereça capacidades robustas de autoescalonamento, o tempo de inicialização dos containers ainda pode exigir pré-provisionamento para cargas máximas. Acreditamos que o tempo de inicialização em milissegundos do WebAssembly proporciona uma solução serverless mais dinâmica e flexível para workloads sob demanda. Desde nossa discussão anterior sobre Spin, o ecossistema do WebAssembly fez avanços significativos. Estamos empolgadas em destacar o SpinKube como uma plataforma que simplifica o desenvolvimento e a implantação de workloads baseados em WebAssembly no Kubernetes.

## 41. Unblocked

*Avalie*

Unblocked oferece a descoberta de ativos e artefatos no ciclo de vida de desenvolvimento de software (SDLC). Ele se integra com ferramentas comuns de gerenciamento de ciclo de vida de aplicativos (ALM) e de colaboração para ajudar os times a entenderem bases de códigos e recursos relacionados. Ele melhora a compreensão do código ao fornecer contexto imediato e relevante, facilitando a navegação e o entendimento de sistemas complexos. Os times de engenharia podem acessar com segurança e em conformidade as discussões, os ativos e os documentos relacionados ao seu trabalho. O Unblocked também captura e compartilha o conhecimento local que muitas vezes está com pessoas mais experientes do time, tornando esses insights valiosos acessíveis a todas, independentemente do nível de experiência.

# Ferramentas

**Adote**

42. Bruno
43. K9s
44. SOPS
45. Ferramentas de teste de regressão visual
46. Wiz

**Experimente**

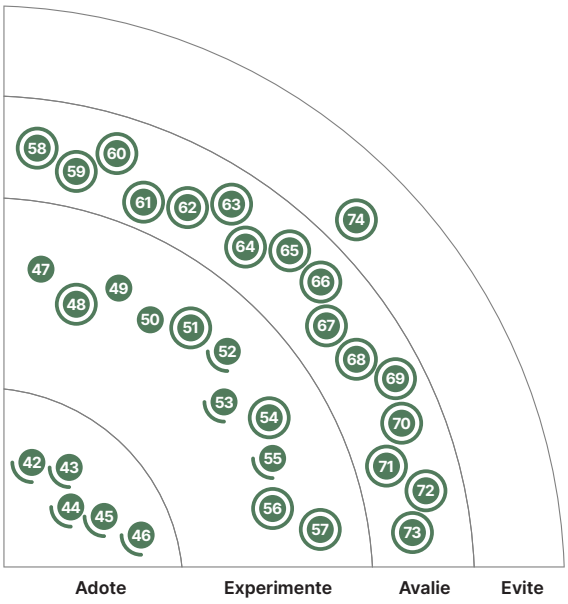
47. AWS Control Tower
48. CCMenu
49. ClickHouse
50. Devbox
51. Difftastic
52. LinearB
53. pgvector
54. Ferramenta de construção Snapcraft
55. Spinnaker
56. TypeScript OpenAPI
57. Unleash

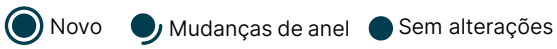
**Avalie**

58. Astronomer Cosmos
59. ColPali
60. Cursor
61. Data Mesh Manager
62. GitButler
63. Assistente de IA JetBrains
64. Mise
65. Mockoon
66. Raycast
67. ReadySet
68. Rspack
69. Semantic Router
70. Agentes de engenharia de software
71. uv
72. Warp
73. Zed

**Evite**

74. CocoaPods



## 42. Bruno

*Adote*

Bruno é uma alternativa de código aberto para desktop do Postman e Insomnia para testes, desenvolvimento e depuração de APIs. Seu objetivo é oferecer colaboração, privacidade e segurança superiores com seu design simples, exclusivo para uso offline. As coleções são armazenadas diretamente no seu sistema de arquivos — escritas em uma linguagem de marcação de texto personalizada, chamada Bru Lang — e podem ser compartilhadas com Git ou uma ferramenta de controle de versão de sua escolha para colaboração. Bruno está disponível tanto como uma aplicação desktop e uma ferramenta de CLI. Ele também oferece uma extensão oficial para o VS Code, com planos para suporte adicional a IDEs. O Bruno se tornou a escolha padrão para várias equipes da Thoughtworks, mas também aconselhamos as equipes a ficarem atentas ao trabalhar em ambientes com VPN e proxy, já que as requisições feitas nessas condições têm apresentado falhas inesperadas.

## 43. K9s

*Adote*

K9s melhorou suas capacidades de visualização integrando gráficos e visualizações mais detalhados. Agora ele oferece uma melhor representação de logs e métricas e é mais flexível em como exibe recursos personalizados (CRDs). As operações nos pods foram expandidas e incluem uma melhor integração com ferramentas de depuração, como kubectl debug, e suporte aprimorado a ambientes muti-cluster. O suporte a CRDs melhorou significativamente e agora fornece uma melhor navegação e gerenciamento desses recursos bem como uma interação mais suave com recursos personalizados. O painel de atalhos também foi melhorado para deixá-lo mais acessível para as desenvolvedoras que são menos experientes com o kubectl. Isso é uma melhora significativa, já que o K9s inicialmente era focado prioritariamente em equipes de DevOps.

## 44. SOPS

*Adote*

SOPS é um editor de arquivos criptografados que suporta vários formatos de arquivo criptografados com KMS. Nosso conselho em relação ao gerenciamento de segredos sempre foi desacoplá-lo do código-fonte. No entanto, quando confrontadas com a escolha entre automação completa (no princípio de infraestrutura como código) e algumas etapas manuais (usando ferramentas como vaults) para gerenciar, gerar e rotacionar segredos iniciais, as equipes geralmente enfrentam um dilema. Por exemplo, nossas equipes utilizam o SOPS para gerenciar as credenciais iniciais de segurança necessárias para a inicialização da infraestrutura. Em algumas situações, no entanto, é impossível remover segredos de repositórios de código legados. Nessas situações, usamos o SOPS para criptografar segredos em arquivos de texto. O SOPS integra-se com serviços de gerenciamento de chaves na nuvem, tais como AWS e GCP Key Management Service (KMS) ou Azure Key Vault, que fornecem as chaves de criptografia. Ele também funciona em várias plataformas e suporta chaves PGP. Várias de nossas equipes usam o SOPS por padrão quando precisam gerenciar segredos no repositório de código.

## 45. Ferramentas de teste de regressão visual

*Adote*

Já destacamos ferramentas de teste de regressão visual previamente e observamos seus algoritmos evoluírem da comparação primitiva em nível de pixel para sofisticados métodos de correspondência de padrões e reconhecimento óptico de caracteres (OCR). As primeiras ferramentas de regressão visual geravam muitos falsos positivos e só foram úteis em estágios finais do desenvolvimento, quando a interface já estava estável. BackstopJS evita esse problema configurando seletores e viewports para apontar testes visuais para elementos específicos na página. O aprendizado de máquina tornou mais fácil detectar e comparar elementos visuais com mais precisão, mesmo que eles tenham se movido ou contenham conteúdo dinâmico. Essas ferramentas se tornaram cada vez mais úteis e estão bem posicionadas para aproveitar os últimos desenvolvimentos em IA e aprendizado de máquina. Várias ferramentas comerciais, como Applitools e Percy, agora afirmam usar IA em seus testes de regressão visual. Uma de nossas equipes tem usado o Applitools Eyes extensivamente com resultados satisfatórios. Embora os testes de regressão visual não substituam os bem formulados testes funcionais de ponta a ponta, eles são uma adição valiosa à caixa de ferramentas de teste. Estamos incentivando sua adoção porque eles se tornaram uma opção padrão segura como um elemento de uma estratégia abrangente de teste de IU.

## 46. Wiz

*Adote*

Wiz se destacou como a plataforma de segurança em nuvem preferida em muitos dos nossos projetos. Nossos times apreciam o fato de que ela permite detectar riscos e ameaças mais rápido do que ferramentas semelhantes, pois realiza varreduras contínuas para identificar mudanças. A Wiz pode detectar e alertar sobre configurações incorretas, vulnerabilidades e segredos vazados, tanto em artefatos que ainda não foram implantados em ambientes de produção — imagens de contêiner e código de infraestrutura — quanto em cargas de trabalho em execução — contêineres, VMs e serviços em nuvem. Também valorizamos a capacidade de relatórios poderosos, tanto para times de desenvolvimento quanto para a liderança. Essa análise nos ajuda a entender como uma vulnerabilidade pode afetar um determinado serviço, permitindo que possamos resolver os problemas dentro desse contexto.

## 47. AWS Control Tower

*Experimente*

AWS Control Tower continua sendo nossa principal escolha para gerenciar contas AWS em um ambiente com vários times. Ele oferece um mecanismo conveniente para pré-configurar controles de segurança e conformidade que serão automaticamente aplicados a novas landing zones. Isso é um exemplo de conformidade no ponto de mudança, pois os controles são aplicados e verificados sempre que uma nova infraestrutura é criada, eliminando a necessidade de verificações manuais de conformidade posteriormente. O AWS Control Tower Account Factory for Terraform (AFT) continuou a evoluir desde nossa última análise e agora está disponível em mais regiões da AWS. O AFT permite que as contas do Control Tower sejam provisionadas por meio de um pipeline de infraestrutura como código. Gostamos do fato de o AFT poder ser customizado para enviar webhooks ou realizar ações específicas para integrar de forma segura com ferramentas externas, como o GitHub Actions. Nossos times relataram ótimos resultados usando o AWS Control Tower para gerenciar contas, mas gostaríamos que a AWS aceitasse contribuições da comunidade para o projeto quando oportunidades de melhorias são identificadas.

## 48. CCMenu

*Experimente*

Para equipes que praticam integração contínua, é importante estar ciente do estado da build central no sistema de integração contínua (CI). Antes da pandemia, dashboards em grandes telas de TV nas salas de equipe forneciam essas informações de relance. Como o trabalho remoto chegou para ficar, é necessária uma solução que funcione nas estações de trabalho individuais das desenvolvedoras. Para o Mac, esse nicho é coberto pelo CCMenu, um pequeno aplicativo escrito por uma Thoughtworker. Originalmente parte do CruiseControl, ele funciona com todos os servidores que podem fornecer informações no formato cctray, incluindo Jenkins e TeamCity. Uma reescrita recente adicionou suporte para GitHub Actions e abriu caminho para uma integração mais profunda com mais servidores de CI e estilos de autenticação

## 49. ClickHouse

*Experimente*

ClickHouse é um banco de dados de processamento analítico on-line em colunas (OLAP) de código aberto para análises em tempo real. Começou como um projeto experimental em 2009 e desde então amadureceu como um banco de dados analítico de alto desempenho e linearmente escalável. Seu motor de processamento de consultas eficiente, juntamente com a compactação de dados, o torna ideal para consultas interativas sem pré-agregação. ClickHouse é também uma ótima escolha de armazenamento para dados do OpenTelemetry. Sua integração com o Jaeger permite que você armazene grandes volumes de traces e os analise de forma eficiente.

## 50. Devbox

*Experimente*

Apesar dos avanços nas ferramentas de desenvolvimento, manter ambientes de desenvolvimento locais consistentes continua a ser um desafio para muitas equipes. O onboarding de pessoas engenheiras frequentemente envolve a execução de comandos ou scripts customizados que imprevisivelmente podem falhar em diferentes máquinas e resultar em inconsistências. Para vencer esse desafio, nossas equipes têm se apoiado cada vez mais no Devbox. Dexbox é uma ferramenta de linha de comando que fornece uma interface acessível para criar ambientes de desenvolvimento reproduzíveis por projeto, aproveitando o gerenciador de pacotes Nix sem usar máquinas virtuais ou contêineres. Ela tem simplificado notavelmente o onboarding de nossas equipes porque, uma vez configurado para uma codebase, é necessário um comando CLI (devbox shell) para reproduzir o ambiente definido em um novo dispositivo. Devbox suporta shell hooks, scripts customizados e geração de devcontainer.json para integração com VSCode.

## 51. Difftastic

*Experimente*

Difftastic é uma ferramenta para destacar diferenças entre arquivos de código fonte de uma maneira que reconhece a sintaxe. Isso é bem diferente de ferramentas de comparação textual, como o venerável comando Unix diff. Por exemplo, o Difftastic ignora novas linhas inseridas para quebrar instruções longas em linguagens como Java ou TypeScript que são delimitadas por ponto e vírgula. A ferramenta destaca apenas as alterações que impactam a sintaxe do programa. Ela faz isso primeiro

analisando os arquivos em árvores de sintaxe abstratas e, em seguida, calculando a distância entre eles usando o algoritmo de Dijkstra. Descobrimos que Difftastic é particularmente útil para entender as alterações ao revisar grandes quantidades de código fonte. Difftastic pode ser usado em qualquer linguagem de programação para a qual um analisador esteja disponível e, nativamente, suporta mais de 50 linguagens de programação, bem como formatos de texto estruturados como CSS e HTML. Essa não é uma ferramenta nova, mas achamos que valeria a pena chamar a atenção na era dos assistentes de codificação LLM, onde revisões humanas de código fonte cada vez maiores são gradualmente mais críticas.

## 52. LinearB

*Experimente*

LinearB é uma plataforma de inteligência em engenharia de software que tem empoderado nossas lideranças de engenharia com insights orientados por dados para apoiar a melhoria contínua. A plataforma alinha áreas chave como benchmarking, automação de fluxo de trabalho e investimentos direcionados para aprimorar a experiência e a produtividade das desenvolvedoras. Em nossa experiência com o LinearB, destacamos sua capacidade de promover uma cultura de melhoria e eficiência dentro das equipes de engenharia. Nossas equipes têm utilizado a plataforma para acompanhar métricas chave de engenharia, identificar áreas para aprimoramento e implementar ações baseadas em evidências. Essas capacidades estão bem alinhadas com a proposta de valor central do LinearB: benchmarking, automação da coleta de métricas e habilitação de melhorias orientadas por dados. O LinearB integra-se com ferramentas de código-fonte, ciclo de vida de aplicativos, CI/CD e comunicação e utiliza métricas de engenharia pré-configuradas e personalizadas para fornecer insights quantitativos abrangentes sobre a experiência da desenvolvedora, produtividade e desempenho da equipe. Como defensoras do DORA, valorizamos a forte ênfase do LinearB nessas métricas específicas e sua capacidade de avaliar aspectos cruciais de desempenho da entrega de software, essenciais para aprimorar a eficiência. Historicamente, as equipes enfrentaram desafios para coletar métricas específicas do DORA, muitas vezes dependendo de dashboards personalizados e complexos ou processos manuais. O LinearB continua a oferecer uma solução convincente que automatiza o rastreamento dessas métricas e entrega dados em tempo real que suportam a tomada de decisão proativa em torno da experiência da desenvolvedora, da produtividade e previsibilidade.

## 53. pgvector

*Experimente*

pgvector é uma extensão de código aberto para busca de similaridade para o PostgreSQL, permitindo o armazenamento de vetores juntamente com dados estruturados em uma única e bem estabelecida base de dados. Embora careça de alguns recursos avançados de bancos de dados vetoriais especializados, ele se beneficia da conformidade com ACID, recuperação de pontos no tempo e outras funcionalidades robustas do PostgreSQL. Com o aumento de aplicações impulsionadas por IA generativa, identificamos um padrão crescente de armazenamento e busca eficiente de vetores de embeddings por similaridade, algo que o pgvector aborda de forma eficaz. Com o uso crescente do pgvector em ambientes de produção, especialmente onde equipes já utilizam um provedor de nuvem que oferece PostgreSQL gerenciado, e sua comprovada capacidade de atender às necessidades comuns de busca vetorial sem exigir um banco de dados vetorial separado, estamos confiantes de seu potencial. Nossos times o consideram valioso em projetos que comparam dados estruturados e não estruturados, demonstrando seu potencial para uma adoção mais ampla, e por isso, estamos movendo-o para o anel Experimente.

## 54. Ferramenta de construção Snapcraft

*Experimente*

Snapcraft é uma ferramenta de linha de comando de código aberto para construir e empacotar aplicativos independentes chamados snaps no Ubuntu, outras distribuições Linux e macOS. Os snaps são fáceis de implementar e manter em plataformas de hardware, incluindo máquinas Linux, ambientes virtuais e sistemas de computador de bordo de veículos. Enquanto o Snapcraft oferece uma loja de aplicativos pública para publicação de snaps, nossas equipes usam a ferramenta de construção para empacotar o sistema de direção autônoma como um snap sem publicá-lo na loja de aplicativos pública. Isso nos permite construir, testar e depurar o sistema de software embarcado localmente enquanto o publicamos em um repositório de artefatos internos.

## 55. Spinnaker

*Experimente*

Spinnaker é uma plataforma open-source de entrega contínua criada pela Netflix. Ela implementa gerenciamento de clusters e imagens preparadas para a nuvem como funcionalidade principal. Gostamos da abordagemopinionada da Spinnaker para a implantação de microsserviços. Nas edições anteriores, notamos a falta da capacidade em configurar pipelines como código, mas isso foi solucionado com a adição do spin CLI. Embora não recomendemos a Spinnaker para cenários simples de CD, ela se tornou uma ferramenta de escolha para situações complexas com pipelines de implantação igualmente complexas.

## 56. TypeScript OpenAPI

*Experimente*

TypeScript OpenAPI (ou tsoa) é uma alternativa ao Swagger para gerar especificações OpenAPI a partir do seu código. Ela adota uma abordagem code-first, com controladores e modelos TypeScript como a única fonte de verdade, utilizando anotações ou decoradores TypeScript em vez de exigir arquivos e configurações mais complexos, como os que são necessários ao usar ferramentas OpenAPI para TypeScript. Ela gera especificações de API nas versões 2.0 e 3.0, e as rotas podem ser geradas para Express, Hapi e Koa. Se você está escrevendo APIs em TypeScript, esse projeto vale a pena ser explorado.

## 57. Unleash

*Experimente*

Embora o uso de feature toggle mais simples possível continue sendo nossa abordagem recomendada, o crescimento das equipes e o desenvolvimento mais ágil tornam a gestão de toggles feitos manualmente mais complexa. Unleash é uma opção amplamente utilizada por nossas equipes para lidar com essa complexidade e viabilizar o CI/CD. A ferramenta pode ser usada tanto como serviço quanto de forma auto-hospedada. Oferece SDKs em várias linguagens com uma boa experiência para desenvolvedoras e uma interface amigável de fácil administração. Embora ainda não haja suporte oficial para a especificação OpenFeature, você pode encontrar providers mantidos pela comunidade para Go e Java. A Unleash pode ser utilizada tanto para feature toggles simples quanto para segmentação e lançamentos graduais, tornando-se uma opção adequada para o gerenciamento de features em larga escala.

## 58. Astronomer Cosmos

*Avalie*

Astronomer Cosmos é um plugin Airflow desenvolvido para prover mais suporte nativo para workflows dbt no Airflow. Com o plugin instalado, quando DbtDag é usado para empacotar um workflow dbt, ele transforma nodos dbt em grupos de tarefas/tarefa Airflow, permitindo que pessoas engenheiras visualizem grafos de dependência dbt e os seus progressos de execução diretamente na interface Airflow. Ele também suporta a utilização de conexões Airflow ao invés de perfis dbt, potencialmente reduzindo configuration sprawl. Estamos experimentando o potencial da ferramenta de tornar o trabalho com dbt no Airflow mais fácil.

## 59. ColPali

*Avalie*

ColPali é uma ferramenta emergente para recuperação de documentos PDF utilizando modelos de linguagem visual (VLMs), abordando os desafios de construir uma aplicação robusta de geração aumentada por recuperação (RAG) que pode extrair dados de documentos multimídia contendo imagens, diagramas e tabelas. Diferentemente dos métodos tradicionais que dependem de embeddings baseados em texto ou técnicas de reconhecimento óptico de caracteres (OCR), o ColPali processa páginas inteiras de PDFs, utilizando um transformador visual para criar embeddings que consideram tanto o conteúdo textual quanto o visual. Essa abordagem holística permite uma recuperação mais eficaz, além de fornecer justificativas para a seleção de certos documentos, aprimorando significativamente o desempenho do RAG em relação a PDFs ricos em dados. Nós testamos o ColPali com várias clientes e ele demonstrou resultados promissores, mas a tecnologia ainda está nos estágios iniciais. Vale a pena avaliar, particularmente para organizações com dados em documentos visuais complexos.

## 60. Cursor

*Avalie*

A corrida por ferramentas de programação assistidas por IA continua, e a mais chamativa é Cursor. O Cursor é um editor de código com foco em IA projetado para aumentar a produtividade do desenvolvedor, integrando profundamente a IA ao fluxo de trabalho de codificação. Temos prestado atenção a ele em edições anteriores do radar, mas é claro que a recente melhoria contínua do Cursor inaugurou uma mudança qualitativa. Em nosso uso, o Cursor demonstrou fortes capacidades de raciocínio contextual com base na base de código existente. Enquanto outras ferramentas de código de IA como GitHub Copilot tendem a gerar e colaborar em torno de trechos de código, as operações de edição multi-linha e multi-arquivo do Cursor o destacam. O Cursor é um fork do VSCode e desenvolvido com base nele, fornecendo um método de interação rápido e intuitivo que está em conformidade com a intuição do desenvolvedor. Operações poderosas podem ser concluídas com `ctrl/cmd+K` e `ctrl/cmd+L`. O Cursor está liderando uma nova rodada de competição em ferramentas de programação de IA, especialmente em relação à interação da desenvolvedora e à compreensão das bases de código

## 61. Data Mesh Manager

*Avalie*

Data Mesh Manager fornece camada de metadados de uma típica plataforma de data mesh. Particularmente, ela foca na definição de produtos de dados e na especificação de contratos de dados fazendo uso da iniciativa OpenContract e pode ser integrada à construção de pipelines usando o DataContract CLI associado. A aplicação também disponibiliza um catálogo de dados para a descoberta e análise de produtos de dados e seus metadados, além de permitir a governança federada, incluindo a definição de métricas de qualidade dos dados e o gerenciamento de suas regras. É uma das primeiras ferramentas nativas nessa categoria, o que significa que não está apenas tentando retroadaptar plataformas existentes ao paradigma do data mesh.

## 62. GitButler

*Avalie*

Apesar de sua potência e utilidade, a interface de linha de comando do Git é notoriamente complexa quando se trata de gerenciar múltiplos branches e preparar commits dentro deles. O GitButler é um cliente Git que oferece uma interface gráfica com o objetivo de simplificar esse processo. Ele faz isso rastreando mudanças em arquivos não commitados de forma independente do Git e, em seguida, preparando essas mudanças em branches virtuais. Pode-se argumentar que isso é uma solução para um problema que não deveria existir; se você fizer pequenas alterações e enviar para a branch principal com frequência, não há necessidade de múltiplos branches. No entanto, quando seu fluxo de trabalho envolve pull requests, a estrutura de branches pode se tornar complexa, especialmente se houver um longo ciclo de revisão antes de um PR ser integrado. Para lidar com isso, o GitButler também se integra ao GitHub, permitindo que você agrupe seletivamente as alterações em pull requests e os envie diretamente da ferramenta. O GitButler é mais uma entrada na crescente categoria de ferramentas voltadas para gerenciar a complexidade inerente ao processo de PR.

## 63. JetBrains AI Assistant

*Avalie*

JetBrains AI Assistant é uma assistente de codificação projetada para se integrar perfeitamente com todas as IDEs da JetBrains para oferecer suporte com a autocomplementação do código, geração de testes e adesão a guias de estilo. Baseada em modelos como o OpenAI e o Google Gemini, destaca-se por sua capacidade de garantir consistência ao lembrar estilos de codificação para sessões futuras. Nossas pessoas desenvolvedoras acharam suas capacidades de geração de testes particularmente úteis e notaram sua habilidade de lidar com resultados mais longos sem problemas de estabilidade. No entanto, ao contrário de algumas concorrentes, a JetBrains não hospeda seus próprios modelos, o que pode não ser ideal para clientes preocupadas com o manuseio de dados por terceiros. Ainda assim, a integração da ferramenta com as IDEs da JetBrains faz dela uma escolha promissora para times que buscam assistentes de codificação impulsionadas por IA.

## 64. Mise

***Avalie***

Desenvolvedoras que trabalham em um ambiente poliglota frequentemente precisam gerenciar várias versões de diferentes linguagens e ferramentas. Mise mira resolver esse problema fornecendo uma única ferramenta para substituir o nvm, pyenv, rbenv, rustup, entre outros, e é um substituto direto para o asdf. Mise é escrito em Rust para garantir maior velocidade na interação com o shell e, ao contrário do asdf, que usashims baseados em shell, o Mise modifica a variável de ambiente PATH antecipadamente, de modo que as ferramentas sejam chamadas diretamente. É em parte por isso que o Mise é mais rápido que o asdf. Para aquelas desenvolvedoras que já estão familiarizadas com o asdf, ele oferece a mesma funcionalidade, mas com algumas diferenças importantes. Por ser escrito em Rust, ele é mais rápido e tem alguns recursos que o asdf não tem, como a capacidade de instalar várias versões da mesma ferramenta ao mesmo tempo e comandos mais tolerantes, incluindo correspondência difusa. Ele também fornece um executor de tarefas integrado, útil para executar linters, testes, builders, servidores e outras tarefas específicas de um projeto.
Se você está um pouco cansada de usar várias ferramentas para gerenciar seu ambiente de desenvolvimento, ou da sintaxe às vezes complicada de outras ferramentas, o Mise definitivamente vale a pena ser considerado.

## 65. Mockoon

***Avalie***

Mockoon é uma ferramenta open-source para simulação de APIs. Possui uma interface intuitiva, rotas personalizáveis e respostas dinâmicas, bem como a capacidade de automatizar a criação de conjuntos de dados simulados. Mockoon é compatível com OpenAPI e permite gerar diferentes cenários que podem ser testados localmente e integrados a uma pipeline de desenvolvimento. Você também pode criar “mocks parciais” nterceptando as solicitações e simulando apenas as chamadas definidas no Mockoon. Os partial mocks ajudam a simular rotas ou endpoints específicos da API e encaminhar outras solicitações para servidores reais. Embora os partial mocks possam ser úteis em certos cenários, existe o risco de uso excessivo, o que pode resultar em complexidade desnecessária. Além disso, o Mockoon continua sendo uma ferramenta valiosa para configurar rapidamente APIs simuladas e melhorar e automatizar os fluxos de trabalho de desenvolvimento.

## 66. Raycast

***Avalie***

Raycast é um inicializador de modelo freemium (combinação das palavras “gratuito” e “premium”) para macOS que permite iniciar rapidamente aplicativos, executar comandos, buscar arquivos e automatizar tarefas diretamente do teclado. Nossas equipes valorizam seus recursos prontos para desenvolvedoras e sua fácil extensibilidade, que permite interação com aplicativos e serviços de terceiros, como VSCode, Slack, Jira e Google, entre muitos outros. O Raycast é voltado para a produtividade e minimiza a troca de contexto, tornando-se uma ferramenta útil para quem busca otimizar suas tarefas diárias. Usuárias da versão Pro têm acesso ao Raycast AI, um assistente de busca especializado emcom tecnologia de IA.

## 67. ReadySet

*Avalie*

ReadySet é um cache de consultas para MySQL e PostgreSQL. Ao contrário das soluções de cache tradicionais que dependem de invalidação manual, o ReadySet utiliza fluxos de replicação do banco de dados para atualizar seu cache de forma incremental. Através da materialização parcial de views, o ReadySet alcança latências finais mais baixas do que uma réplica de leitura tradicional. O ReadySet é compatível com os protocolos de MySQL e PostgreSQL, permitindo que você o implante entre sua aplicação e seu banco de dados para escalar horizontalmente as cargas de leitura sem exigir mudanças na aplicação.

## 68. Rspack

*Avalie*

Muitas de nossas equipes que trabalham em frontends baseados na web mudaram de ferramentas de empacotamento mais antigas, como a Webpack, para Vite. Uma nova participante nesse campo é a Rspack, que, após um ano e meio de desenvolvimento, acaba de lançar sua versão 1.0. Projetada como um substituto direto para a Webpack, é compatível com plugins e loaders do ecossistema Webpack. Isso pode ser uma vantagem sobre a Vite ao migrar configurações complexas do Webpack. A principal razão pela qual nossas equipes estão migrando para ferramentas mais recentes como Vite e Rspack é a experiência da desenvolvedora e, em particular, a velocidade. Nada interrompe mais o fluxo de desenvolvimento do que ter que esperar um ou dois minutos para obter feedback sobre as últimas alterações no código. Escrita em Rust, Rspack oferece um desempenho significativamente mais rápido do que a Webpack e, em muitos casos, é até mais rápido que a Vite.

## 69. Semantic Router

*Avalie*

Ao desenvolver aplicações baseadas em LLM, é fundamental determinar a intenção da usuária antes de direcionar uma solicitação para um agente especializado ou iniciar um fluxo específico. O Semantic Router atua como uma camada de tomada de decisão ultrarrápida para LLMs e agentes, permitindo o roteamento eficiente e confiável de solicitações com base no significado semântico. Usando vector embeddings para inferir a intenção, o Semantic Router reduz chamadas desnecessárias aos LLMs, proporcionando uma abordagem mais enxuta e econômica para entender a intenção da usuária. Seu potencial vai além da inferência de intenção, atuando como um bloco de construção versátil para diversas tarefas semânticas. A velocidade e flexibilidade que ele oferece o posicionam como um forte concorrente em ambientes que exigem tomada de decisão rápida e em tempo real, sem o custo adicional dos LLMs.

## 70. Agentes de engenharia de software

*Avalie*

Um dos tópicos mais comentados, atualmente, no espaço de IA generativa, é o conceito de agentes de engenharia de software. Essas ferramentas de assistência de codificação fazem mais do que apenas ajudar a pessoa engenheira com trechos de código aqui e ali; elas ampliam o tamanho do problema que podem resolver, idealmente de forma autônoma e com o mínimo de interferência de uma pessoa. A ideia é que essas ferramentas possam pegar um problema do GitHub ou um tíquete

do Jira e propor um plano e alterações de código para implementá-lo, ou até mesmo criar uma pull request para uma pessoa revisar. Embora este seja o próximo passo lógico para aumentar o impacto da assistência de codificação com IA, a meta frequentemente anunciada de agentes genéricos que podem cobrir uma ampla gama de tarefas de codificação é muito ambiciosa, e o estado atual das ferramentas ainda não está mostrando isso de forma convincente. No entanto, podemos acompanhar isso funcionando mais cedo ou mais tarde para um escopo mais limitado de tarefas diretas, liberando tempo da pessoa desenvolvedora para trabalhar em problemas mais complexos. As ferramentas que estão lançando e comercializando versões beta de agentes incluem GitHub Copilot Workspace, qodo flow, agents for JIRA Tabnine e Amazon Q Developer. O comparador SWE Bench lista ferramentas neste contexto, mas alertamos para levar as comparações no campo da IA com cautela.

## 71. uv

***Avalie***

Rust é uma boa indicação para escrever ferramentas de linha de comando devido a sua rápida performance de inicialização, e acompanhamos pessoas reescrevendo algumas toolchains com a linguagem. Nós mencionamos o Ruff, um linter Python escrito em Rust, no Radar anterior. Para esta edição, nós avaliamos uv, uma ferramenta de gerenciamento de pacotes Python escrita em Rust. A proposta de valor da uv é serextremamente rápida e ela supera outras ferramentas de gerenciamento de pacotes Python por uma grande margem em seus benchmarks. No entanto, durante nossa avaliação para o Radar, nós discutimos se otimizar em segundos para ferramentas de construção é uma melhoria significativa. Comparado ao desempenho, o mais importante para um sistema de gerenciamento de pacotes é o ecossistema, uma comunidade madura e suporte a longo prazo. Dito isso, o feedback da equipe do projeto nos mostrou que essa melhoria pequena de velocidade pode ser um grande benefício para melhorar os ciclos de feedback e a experiência geral da pessoa desenvolvedora, uma vez que temos a tendência de manualmente tornar o cache de CI/CD muito complexo para conseguir este pequeno ganho de performance. uv simplifica nosso ambiente de gerenciamento Python. Considerando que ainda há muito espaço para melhorias no gerenciamento de pacotes e ambientes para desenvolvimento Python, nós achamos que a uv é uma opção que vale a pena avaliar.

## 72. Warp

***Avalie***

Warp é um terminal para macOS e Linux. Ele divide as saídas de comandos em blocos para melhorar a legibilidade. Warp oferece recursos orientados por IA como sugestão inteligente de comandos e processamento de linguagem natural. Também inclui funcionalidades de anotação que permitem às usuarias organizar comandos e saídas, além de adicionar anotações e documentação. Você pode aproveitar esses recursos para criar arquivos LEIAME ou materiais de onboarding, oferecendo uma maneira estruturada e interativa de apresentar e gerenciar fluxos de trabalho de terminal. Warp se integra facilmente com Starship, um prompt flexível com suporte a múltiplos shells, permitindo personalizar a experiência no terminal e recuperar informações sobre processos em execução, a versão específica da ferramenta que você está usando, detalhes do Git ou da usuária atual do Git, entre outras informações.

## 73. Zed

*Avalie*

Depois da descontinuação do projeto do editor de textos Atom, as pessoas responsáveis pela sua criação desenvolveram um novo editor chamado Zed. Escrito em Rust e otimizado para aproveitar os hardwares atuais, o Zed parece rápido. Ele possui todas as funcionalidades que esperamos em um editor moderno: suporte a muitas linguagens de programação, um terminal integrado e edição multi-buffer. Codificação assistida por IA está disponível através da integração com diversos provedores de LLM. Como entusiastas da programação em pares, estamos interessadas pela funcionalidade de colaboração remota integrada ao Zed. As pessoas desenvolvedoras podem se encontrar através de seus IDs do Github e podem colaborar no mesmo workspace em tempo real. É muito cedo para dizer se as equipes de desenvolvimento podem e querem sair da atração do ecossistema Visual Studio Code, mas o Zed é uma alternativa a ser explorada.

## 74. CocoaPods

*Evite*

CocoaPods tem sido uma ferramenta popular de gerenciamento de dependências para projetos Cocoa em Swift e Objective-C. No entanto, a equipe do CocoaPods anunciou que o projeto está em modo de manutenção após mais de uma década sendo uma ferramenta essencial para desenvolvedoras em iOS e macOS. Embora a ferramenta e seus recursos permaneçam disponíveis, o desenvolvimento ativo será interrompido. As desenvolvedoras são encorajadas a transicionar para o Swift Package Manager, que oferece integração nativa com o Xcode e um melhor suporte a longo prazo da Apple.

# Linguagens e Frameworks

**Adote**

75. dbt
76. Testcontainers

**Experimente**

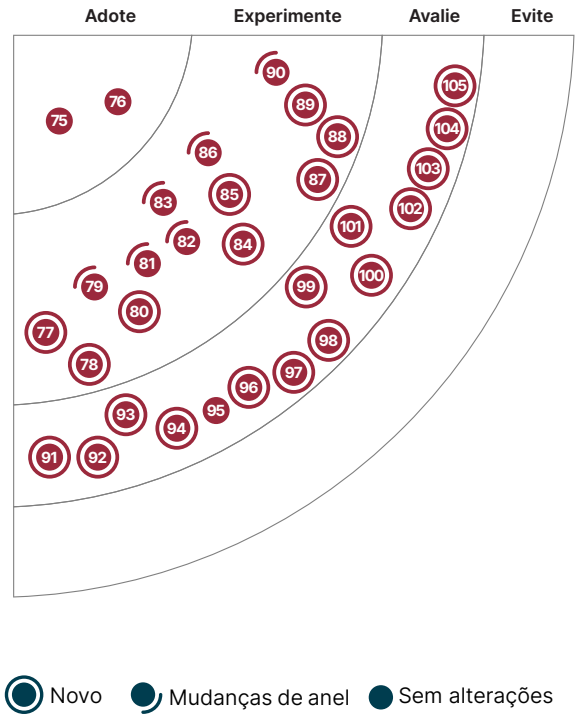
77. CAP
78. CARLA
79. Pacotes de ativos do Databricks
80. Instructor
81. Kedro
82. LiteLLM
83. LlamaIndex
84. LLM Guardrails
85. Medusa
86. Pkl
87. ROS 2
88. seL4
89. SetFit
90. vLLM

**Avalie**

91. Apache XTable™
92. dbldatagen
93. DeepEval
94. DSPy
95. Flutter para Web
96. kotaemon
97. Lenis
98. LLMLingua
99. Microsoft Autogen
100. Pingora
101. Ragas
102. Score
103. shadcn
104. Slint
105. SST

**Evite**

—




Novo  Mudanças de anel  Sem alterações

## 75. dbt

*Adote*

Continuamos a considerar o dbt como uma opção sólida e sensata para implementar transformações de dados em pipelines de ELT. Gostamos do fato de que ele preza pelo rigor da engenharia e permite práticas como modularidade, capacidade de teste e reutilização de transformações baseadas em SQL. O dbt integra-se bem com muitos data warehouses, lakehouses e bancos de dados em nuvem, incluindo Snowflake, BigQuery, Redshift, Databricks e Postgres, e tem um ecossistema saudável de pacotes comunitários em torno dele. O suporte nativo introduzido recentemente (no dbt core 1.8+ e na recém-introduzida experiência "sem versão" do dbt Cloud) para testes unitários fortalece ainda mais sua posição em nossa caixa de ferramentas. Nossas equipes valorizam o fato de a nova funcionalidade de teste unitário permitir que elas definam facilmente os dados de teste estáticos, configurem as expectativas de saída e testem os modos incremental e de atualização total de seus pipelines. Em muitos casos, isso permitiu que elas descontinuassem os scripts desenvolvidos internamente, mantendo o mesmo nível de qualidade.

## 76. Testcontainers

*Adote*

Em nossa experiência, Testcontainers são uma opção padrão útil para criar um ambiente confiável para a execução de testes. Trata-se de uma biblioteca, adaptada para várias linguagens, que Dockeriza dependências de teste comuns – incluindo vários tipos de bancos de dados, tecnologias de fila, serviços de nuvem e dependências de teste de interface de usabilidade, como navegadores da web – com a capacidade de executar Dockerfiles personalizados quando necessário. Recentemente, foi lançada uma versão para desktop que permite o gerenciamento visual de sessões de teste e a capacidade de gerenciar cenários mais complexos, o que nossas equipes consideraram muito útil.

## 77. CAP

*Experimente*

CAP é uma biblioteca .NET que implementa o Outbox pattern. Ao trabalhar com sistemas de mensageria distribuída como RabbitMQ ou Kafka, frequentemente enfrentamos o desafio de garantir que as atualizações no banco de dados e as publicações de eventos sejam realizadas de forma atômica. A biblioteca CAP resolve esse desafio registrando a intenção de publicar o evento na mesma transação de banco de dados que o gerou. Consideramos a CAP bastante útil, pois oferece suporte a vários bancos de dados e plataformas de mensageria, garantindo a entrega pelo menos uma vez.

## 78. CARLA

*Experimente*

CARLA é um simulador para pesquisas de direção autônoma de código aberto utilizado para testar sistemas de condução autônoma antes de seu lançamento em produção. Ele oferece flexibilidade na criação e reutilização de modelos tridimensionais de veículos, terrenos, pessoas, animais e mais, tornando possível simular cenários, como os de um pedestre atravessando a rua ou o encontro com um veículo vindo em uma velocidade específica. O sistema de condução autônoma em testes deve reconhecer estes atores dinâmicos e tomar a ação apropriada, como frear. Nossos times usam CARLA para o desenvolvimento contínuo e testes de sistemas de direção autônoma.

## 79. Pacotes de ativos do Databricks

*Experimente*

Os pacotes de ativos do Databricks (DABs), que atingiram disponibilidade geral em abril de 2024, estão se tornando a ferramenta preferida para empacotamento e implantação de ativos no Databricks, facilitando a adoção de práticas de engenharia de software em nossos times de dados. Os DABs permitem englobar a configuração de fluxos de trabalho e tarefas, além do código a ser executado nessas tarefas, em um pacote que pode ser implantado em diversos ambientes por meio de pipelines de CI/CD. A ferramenta oferece templates para tipos comuns de ativos e também suporta templates personalizados, o que possibilita a criação de modelos de serviço personalizados para projetos de engenharia de dados e aprendizado de máquina. Nossos times têm adotado cada vez mais essa ferramenta como parte essencial de seus fluxos de trabalho de engenharia. Embora os DABs incluam templates para notebooks e suportem sua implantação em produção, não recomendamos a produção de notebooks. Em vez disso, incentivamos a escrita intencional de código de produção utilizando práticas de engenharia que garantam a manutenção, resiliência e escalabilidade dessas cargas de trabalho.

## 80. Instructor

*Experimente*

Quando usamos chatbots baseados em modelos de linguagem de grande porte (LLM) como usuários finais, eles geralmente nos retornam uma resposta em linguagem natural não estruturada. Ao construir aplicações de GenAI que vão além de chatbots, pode ser útil solicitar ao LLM uma resposta estruturada em JSON, YAML ou outros formatos, para então analisar e usar essa resposta na aplicação. No entanto, como os LLMs não são determinísticos, eles nem sempre fazem exatamente o que pedimos. A biblioteca Instructor pode ser usada para nos ajudar a solicitar saídas estruturadas de LLMs. Você pode definir a estrutura de saída desejada e configurar tentativas adicionais se o LLM não retornar a estrutura que você solicitou. Como a melhor experiência da usuária ao trabalhar com LLMs muitas vezes envolve o streaming dos resultados em vez de esperar pela resposta completa, o Instructor também cuida de analisar estruturas parciais de um fluxo..

## 81. Kedro

*Experimente*

Kedro melhorou significativamente como uma ferramenta para MLOps e manteve seu foco em modularidade e práticas de engenharia, pontos que gostamos desde o início. Um exemplo que destaca sua modularidade é a introdução do package independente kedro-datasets, que separa o código dos dados. O Kedro adicionou melhorias na CLI, nos templates de projetos iniciais e nas capacidades de telemetria. Além disso, o lançamento recente de uma extensão para VS Code é um ótimo reforço para a experiência das pessoas desenvolvedoras.

## 82. LiteLLM

*Experimente*

LiteLLM é uma biblioteca para perfeita integração com várias APIs de provedores de modelos de linguagem de grande porte (LLM) que padroniza as interações através de um formato da OpenAI API. Ela suporta uma extensa variedade de modelos e provedores, e oferece uma interface unificada para preenchimento, incorporação e geração de imagens. O LiteLLM simplifica a integração traduzindo as entradas para corresponder aos requisitos específicos de endpoint de cada provedor. Também

fornece uma estrutura necessária para implementar muitos dos recursos operacionais necessários em uma aplicação de produção, como cache, registro de logs, limitação de taxa de request e balanceamento de carga. Isso garante uma operação uniforme em diferentes LLMs. Nossas equipes estão usando o LiteLLM para facilitar a troca de vários modelos; um recurso necessário no cenário atual, onde os modelos estão evoluindo rapidamente. É crucial reconhecer que as respostas do modelo a prompts idênticos variam, indicando que um método de invocação consistente por si só pode não otimizar totalmente o desempenho da geração de respostas completas. Além disso, cada modelo implementa recursos adicionais de forma única e uma única interface pode não ser suficiente para todos. Por exemplo, uma de nossas equipes teve dificuldade em identificar vantagens na chamada de função em um modelo AWS Bedrock ao fazer proxy através do LiteLLM.

## 83. LlamaIndex

*Experimente*

LLamaIndex traz mecanismos que lhe permite definir aplicações específicas para um domínio conectadas a LLMs, além de dar suporte a tarefas como ingestão de dados, indexação vetorial e busca de respostas a perguntas em linguagem natural. Nossos times usaram LlamaIndex para construir uma pipeline de geração aumentada por recuperação (RAG) que automatiza a ingestão de documentos, indexando seus embeddings e permitindo sua busca com base nas informações fornecidas pela usuária. Usando LlamaHub, você pode estender e customizar módulos do LLamaIndex para melhor atender suas necessidades e construir, por exemplo, aplicações LLM com seus LLMs preferidos, embeddings e provedores de armazenamento de vetores.

## 84. LLM Guardrails

*Experimente*

LLM Guardrails é um conjunto de diretrizes, políticas ou filtros projetados para evitar que modelos de linguagem de grande porte (LLMs) gerem conteúdo prejudicial, enganoso ou irrelevante. Os guardrails também podem ser usados para proteger aplicações de LLMs contra usuárias mal-intencionadas que tentem manipular o sistema com técnicas como a manipulação de entrada. Elas atuam como uma rede de segurança, estabelecendo limites para o modelo ao processar e gerar conteúdo. Existem alguns frameworks emergentes nesse espaço, como o NeMo Guardrails, Guardrails AI e Aporia Guardrails, que nossas equipes têm achado úteis. Recomendamos que toda aplicação que utilize LLMs tenha guardrails implementados e que suas regras e políticas sejam continuamente aprimoradas. Eles são cruciais para construir aplicações de chat com LLMs que sejam responsáveis e confiáveis.

## 85. Medusa

*Experimente*

Em nossa experiência, a maioria das soluções de e-commerce para construção de sites de compras geralmente caem na armadilha 80/20: podemos facilmente construir 80% do que queremos, mas não podemos fazer nada sobre os 20% restantes. Medusa oferece um bom equilíbrio. É uma plataforma de e-commerce de código aberto altamente customizável que permite às pessoas desenvolvedoras criarem experiências de compras únicas e personalizadas que podem ser auto-hospedadas ou executadas na plataforma da Medusa. Construído em Next.js e PostgreSQL, Medusa acelera o processo de desenvolvimento com sua gama abrangente de módulos — desde o carrinho de compras básico e gerenciamento de pedidos até recursos avançados como módulos de vale-presente e cálculo de impostos para diferentes regiões. Descobrimos que a Medusa é uma estrutura valiosa e a aplicamos a alguns projetos.

## 86. Pkl

*Experimente*

Pkl é uma ferramenta de código aberto para configuração de linguagem criada inicialmente para uso interno na Apple. Sua funcionalidade principal é o sistema de validação e tipos, que permite que os erros de configuração sejam detectados antes da implementação. O Pkl permitiu que nossas equipes reduzissem a duplicação de código (para casos como sobreposições de ambiente) e realizassem a validação antes que as alterações de configuração fossem aplicadas a ambientes de produção. Ele gera arquivos JSON, PLIST, YAML e .properties e tem ampla integração de IDE e linguagem, incluindo geração de código.

## 87. ROS 2

*Experimente*

ROS 2 é um framework de código aberto projetado para o desenvolvimento de sistemas robóticos. Ele oferece um conjunto de bibliotecas e ferramentas que possibilitam a implementação modular de aplicações, abrangendo funções como comunicação entre processos, execução multi-thread e qualidade de serviço. O ROS 2 se baseia em seu predecessor ao oferecer capacidades aprimoradas em tempo real, melhor modularidade, maior suporte a diversas plataformas e padrões sensatos. O ROS 2 está ganhando força na indústria automotiva; sua arquitetura baseada em nós e o modelo de comunicação baseado em tópicos são especialmente atraentes para fabricantes com aplicações veiculares complexas e em constante evolução, como funcionalidades de condução autônoma.

## 88. seL4

*Experimente*

Em veículos definidos por software (SDV) ou outros cenários críticos para a segurança, a estabilidade em tempo real do sistema operacional é crucial. Algumas empresas monopolizam esse campo devido às altas barreiras de entrada, por isso soluções de código aberto, como seL4, são preciosas. O seL4 é um micronúcleo de sistema operacional de alto desempenho e alta confiabilidade. Ele utiliza métodos de verificação formal para garantir, de forma matemática, que o comportamento do sistema operacional esteja em conformidade com a especificação. Sua arquitetura de micronúcleo também minimiza as responsabilidades principais para garantir a estabilidade do sistema. Temos acompanhado empresas de veículos elétricos como a NIO se envolverem com o ecossistema seL4, e pode haver mais desenvolvimentos nessa área no futuro.

## 89. SetFit

*Experimente*

A maioria das ferramentas atuais baseadas em IA são generativas — elas geram textos e imagens utilizando transformadores pré-treinados generativos (GPTs) para esse fim. Para casos de uso que exigem trabalhar com textos existentes, como classificar trechos de texto ou determinar a intenção, os transformadores de sentença são a ferramenta de escolha. Nesse campo, SetFit é uma estrutura para o ajuste fino de transformadores de sentença. Gostamos do SetFit porque ele utiliza aprendizado contrastivo para separar diferentes classes de intenção umas das outras, frequentemente alcançando uma separação nítida com um número muito pequeno de exemplos, até mesmo 25 ou menos. Transformadores de sentença também podem desempenhar um papel em um sistema de IA generativa. Utilizamos com sucesso o SetFit para detecção de intenção em um sistema de chatbot voltado para a cliente que utiliza um LLM, e mesmo estando cientes da API de moderação da OpenAI, optamos por um classificador baseado no SetFit para realizar um ajuste fino adicional visando alcançar um filtro mais rigoroso.

## 90. vLLM

*Experimente*

vLLM é um motor de inferência de alto rendimento e eficiente em termos de memória para LLMs que pode rodar na nuvem ou localmente. Suporta perfeitamente múltiplas arquiteturas de modelos e modelos populares de código aberto. Nossos times implementam agentes vLLM em contêineres Docker em plataformas GPU como NVIDIA DGX e Intel HPC, hospedando modelos como Llama 3.1(8B e 70B), Mistral 7B e Llama-SQL para assistência de desenvolvimento de código, busca de conhecimento e interação com banco de dados de linguagem natural. O vLLM é compatível com o padrão SDK da OpenAI, facilitando uma consistente entrega de modelos. O catálogo de modelos de IA da Azure usa um contêiner de inferência personalizado para aprimorar a performance na entrega de modelos, com o vLLM como motor de inferência padrão devido a sua alta taxa de transferência e eficiente gerenciamento de memória. O framework vLLM está se consolidando como padrão para implantações de modelos em larga escala.

## 91. Apache XTable™

*Avalie*

Entre os formatos de tabela abertos disponíveis que possibilitam lakehouses — como Apache Iceberg, Delta e Hudi — ainda não surgiu um vencedor. Em vez disso, identificamos ferramentas que possibilitam a interoperabilidade entre esses formatos. Delta UniForm, por exemplo, permite interoperabilidade unidirecional, possibilitando que clientes Hudi e Iceberg leiam tabelas Delta. Outro novo participante nesse espaço é o Apache XTable™, um projeto incubado pela Apache que facilita a interoperabilidade omnidirecional entre Hudi, Delta e Iceberg. Assim como o UniForm, ele converte metadados entre esses formatos sem criar uma cópia dos dados subjacentes. O XTable pode ser útil para equipes que estão experimentando com múltiplos formatos de tabela. No entanto, para uso a longo prazo, dadas as diferenças nas funcionalidades desses formatos, depender fortemente da interoperabilidade omnidirecional pode fazer com que as equipes acabem utilizando apenas o denominador comum de funcionalidades.

## 92. dbldatagen

*Avalie*

Preparar dados de teste para engenharia de dados é um desafio significativo. Transferir dados do ambiente de produção para ambientes de teste pode ser arriscado, então as equipes muitas vezes dependem de dados falsos ou sintéticos. Neste Radar, exploramos abordagens inovadoras como dados sintéticos para teste e treinamento de modelos. Mas, na maioria das vezes, a geração procedural de baixo custo é suficiente. O dbldatagen (Databricks Labs Data Generator) é uma dessas ferramentas; é uma biblioteca Python para gerar dados sintéticos no ambiente Databricks para testes, benchmarks, demonstrações e outros usos. dbldatagen pode gerar dados sintéticos em grande escala, até bilhões de linhas em minutos, suportando vários cenários como múltiplas tabelas, captura de dados alterados e operações de mesclagem/junção. Ele lida bem com os tipos primitivos do Spark SQL, gera intervalos e valores discretos, além de aplicar distribuições específicas. Ao criar dados sintéticos usando o ecossistema Databricks, o dbldatagen é uma opção que vale a pena avaliar.

## 93. DeepEval

*Avalie*

DeepEval é um framework de código aberto, baseado em Python, de avaliação do desempenho de LLMs. Você pode utilizar para avaliar a geração aumentada por recuperação (RAG) e outros tipos de aplicativos feitos com frameworks populares como LlamaIndex ou LangChain, bem como para estabelecer uma linha de base e benchmark quando você está comparando diferentes modelos para as suas necessidades. DeepEval oferece um conjunto abrangente de métricas e recursos para avaliar o desempenho de LLMs, incluindo detecção de alucinação, relevância de respostas e otimização de hiperparâmetros. Ele oferece integração com pytest e, além dessas asserções, você pode facilmente integrar a suíte de testes em uma pipeline de integração contínua. Se você está trabalhando com LLMs, considere experimentar o DeepEval para melhorar seu processo de testes e garantir a confiabilidade de suas aplicações.

## 94. DSPy

*Avalie*

A maioria das aplicações baseadas em modelos de linguagem hoje depende de modelos de prompt ajustados manualmente para tarefas específicas. DSPy, um framework para desenvolver tais aplicações, adota uma abordagem diferente que dispensa a engenharia direta de prompts. Em vez disso, ele introduz abstrações de alto nível orientadas ao fluxo do programa através de módulos que podem ser sobrepostos, métricas para otimizar e dados para treinar/testar. Em seguida, otimiza os prompts e/ou pesos do modelo de linguagem subjacente com base nessas métricas definidas. O código resultante se assemelha muito ao treinamento de redes neurais com PyTorch. Achamos a abordagem que eles adotam inovadora por sua perspectiva diferente e acreditamos que vale a pena experimentar.

## 95. Flutter para Web

*Avalie*

O Flutter é conhecido por seu suporte multiplataforma para aplicativos iOS e Android. Agora, ele se expandiu para mais plataformas. Avaliamos anteriormente o Flutter para Web — que nos permite construir aplicativos para iOS, Android e navegadores a partir da mesma base de código. Nem toda aplicação web faz sentido em Flutter, mas acreditamos que ele é particularmente adequado para casos como aplicativos web progressivos, aplicativos de página única e para a conversão de aplicativos móveis Flutter já existentes para a web. O Flutter já oferecia suporte ao WebAssembly (WASM) como um alvo de compilação em seu canal experimental, o que significava que estava em desenvolvimento ativo com possíveis bugs e problemas de performance. As versões mais recentes o tornaram estável. O desempenho das aplicações web em Flutter compiladas para o alvo WASM é muito superior ao de sua compilação para JavaScript. O desempenho quase nativo em diferentes plataformas é também uma das razões pelas quais muitas desenvolvedoras escolhem o Flutter inicialmente.

## 96. kotaemon

*Avalie*

kotaemon é uma ferramenta e framework de código aberto baseada em RAG para desenvolver aplicativos de perguntas e respostas para documentos de base de conhecimento. Ele pode entender vários tipos de documentos, incluindo formatos PDF e DOC, e oferece uma interface web, baseada no Gradio, que permite às usuárias organizar e interagir com uma base de conhecimento por meio de chat. Ele possui pipelines RAG integradas com armazenamento de vetores e pode ser estendido com SDKs. O kotaemon também referencia os documentos fonte em suas respostas, além de fornecer pré-visualização na web e um pontuação de relevância. Para quem deseja criar uma aplicação de perguntas e respostas baseado em RAG, o framework personalizável é um ótimo ponto de partida.

## 97. Lenis

*Avalie*

Lenis é uma biblioteca de rolagem suave, leve e poderosa, projetada para navegadores modernos. Ela permite experiências de rolagem suave, como sincronização de rolagem WebGL e efeitos de paralaxe, tornando-a ideal para equipes que criam páginas com interações de rolagem fluidas e contínuas. Nossas desenvolvedoras acharam Lenis simples de usar, oferecendo uma abordagem simplificada para criar rolagens suaves. No entanto, a biblioteca pode ter problemas de acessibilidade, principalmente com interações de rolagem vertical e horizontal, podendo confundir usuárias com deficiências. Embora visualmente atrativa, ela precisa de uma implementação cuidadosa para manter a acessibilidade.

## 98. LLMLingua

*Avalie*

LLMLingua melhora a eficiência de LLMs ao comprimir prompts usando um pequeno modelo de linguagem para remover tokens não essenciais com perda mínima de desempenho. Essa abordagem permite que LLMs mantenham a capacidade de raciocínio e aprendizado contextual, enquanto processam prompts mais longos de forma eficiente, abordando desafios como eficiência de custos, latência de inferência e manejo de contexto. Compatível com vários LLMs sem necessidade de treinamento adicional e suportando frameworks como LLamaIndex, o LLMLingua é ideal para otimizar o desempenho de inferência de LLMs.

## 99. Microsoft Autogen

*Avalie*

Microsoft Autogen é um framework de código aberto que simplifica a criação e orquestração de agentes de IA, permitindo a colaboração multi-agentes para resolver tarefas complexas. Ele suporta tanto fluxos autônomos quanto fluxos com interação humana, ao mesmo tempo que oferece compatibilidade com uma variedade de modelos de linguagem de grande escala (LLMs) e ferramentas para interação de agentes. Um de nossos times utilizou Autogen para uma cliente para construir uma plataforma com IA onde cada agente representava uma habilidade específica, como geração de código, revisão de código ou resumo de documentação. O framework permitiu que a equipe criasse novos agentes de forma contínua e consistente, definindo o modelo e o fluxo de trabalho corretos. Eles utilizaram o LlamaIndex para orquestrar o fluxo de trabalho. Embora o Autogen tenha se mostrado promissor, especialmente em ambientes de produção, ainda há preocupações em relação à escalabilidade e ao gerenciamento da complexidade à medida que mais agentes são adicionados. Uma avaliação mais aprofundada é necessária para avaliar sua viabilidade a longo prazo no escalonamento de sistemas baseados em agentes.

## 100. Pingora

*Avalie*

Pingora é um framework Rust para criar serviços de rede rápidos, confiáveis e programáveis. Originalmente desenvolvido pela Cloudflare para resolver as deficiências do Nginx, o Pingora vem demonstrando um grande potencial, pois proxies mais novos, como o River, estão sendo criados com base nele. Embora a maioria de nós não enfrente um nível de escala como a do Cloudflare, encontramos cenários em que o roteamento flexível da camada de aplicativos é essencial para nossos serviços de rede. A arquitetura do Pingora nos permite aproveitar todo o poder do Rust nessas situações sem abandonar segurança ou desempenho.

## 101. Ragas

*Avalie*

Ragas é um framework projetado para avaliar o desempenho de pipelines de geração aumentada por recuperação (RAG), que endereça o desafio de avaliar tanto os componentes de recuperação quanto de geração nesses sistemas. Ele fornece métricas estruturadas como fidelidade, relevância das respostas e utilização de contexto, que ajudam a avaliar a eficácia dos sistemas baseados em RAG. Nossas desenvolvedoras acharam o framework útil para realizar avaliações periódicas, com o objetivo de ajustar parâmetros como recuperações top-k e modelos de embedding. Algumas equipes integraram o Ragas em pipelines que são executados diariamente, sempre que o template do prompt ou o modelo muda. Embora suas métricas ofereçam insights sólidos, temos receio de que o framework possa não capturar todas as nuances e interações complexas dos pipelines RAG, e recomendamos considerar frameworks adicionais de avaliação. No entanto, Ragas se destaca por sua capacidade de simplificar a avaliação de RAG em ambientes de produção, oferecendo melhorias valiosas baseadas em dados.

## 102. Score

*Avalie*

Muitas organizações que implementam suas próprias plataformas de desenvolvimento internas tendem a criar seus próprios sistemas de orquestração de plataforma para impor padrões organizacionais entre desenvolvedoras e suas equipes de hospedagem de plataforma. No entanto, as características básicas de uma plataforma de implantação de caminho pavimentado para hospedar cargas de trabalho em contêineres de maneira segura, consistente e compatível são semelhantes de uma organização para outra. Não seria bom se tivéssemos uma linguagem compartilhada para especificar esses requisitos? Score está demonstrando a promessa de se tornar um padrão neste espaço. É uma linguagem declarativa na forma de YAML que descreve como uma carga de trabalho em contêiner deve ser implantada e quais serviços e parâmetros específicos ela precisará para funcionar. Score foi originalmente desenvolvido pela Humanitec como a linguagem de configuração para seu produto Platform Orchestrator, mas agora está sob custódia da Cloud Native Computing Foundation (CNCF) como um projeto de código aberto. Com o apoio da CNCF, Score tem o potencial de ser mais amplamente utilizado além do produto Humanitec. Foi lançado com duas implementações de referência: Kubernetes e Docker Compose. A extensibilidade do Score esperançosamente levará a contribuições da comunidade para outras plataformas. Score certamente tem uma semelhança com a especificação modelo de aplicação aberto (OAM) para Kubevela, mas está mais focado na implantação de cargas de trabalho em contêineres do que no aplicativo inteiro. Também há uma sobreposição com SST, mas o SSI está mais preocupado com a implantação diretamente em uma infraestrutura de nuvem do que em uma plataforma de engenharia interna. Estamos observando o Score, com interesse, à medida que ele evolui.

## 103. shadcn

*Avalie*

shadcn desafia o conceito tradicional de bibliotecas de componentes ao oferecer componentes reutilizáveis no estilo “copia-e-cola” que se tornam parte da sua base de códigos. Essa abordagem concede às equipes total propriedade e controle, permitindo uma customização e extensão mais fácil — áreas onde bibliotecas convencionais mais populares como MUI e Chakra UI frequentemente ficam aquém. Construído com Radix UI e Tailwind CSS, o shadcn se integra de forma simples em qualquer aplicação baseada em React, tornando-o uma boa opção para projetos que priorizam controle e extensibilidade. Ele inclui uma ferramenta de linha de comando para ajudar no processo de copiar e colar os componentes em seu projeto. Seus benefícios também incluem a redução de dependências ocultas e a prevenção de implementações fortemente acopladas, razões pelas quais o shadcn está ganhando força como uma alternativa atraente para equipes que buscam uma abordagem mais prática e adaptável para o desenvolvimento frontend.

## 104. Slint

*Avalie*

Slint é um framework declarativo com interface gráfica para construir interfaces de usuário nativas para aplicativos Rust, C++ ou JavaScript. Embora seja um framework de interface multiplataforma com recursos importantes, como pré-visualização em tempo real, design responsivo de UI, integração com VS Code e uma experiência de usuária nativa, queremos destacar especialmente sua utilidade para sistemas embarcados. Equipes que desenvolvem aplicações embarcadas tradicionalmente enfrentam um número limitado de opções para desenvolvimento de interfaces, cada uma com suas próprias limitações. O Slint oferece o equilíbrio perfeito entre a experiência da pessoa desenvolvedora e o desempenho, utilizando uma linguagem de marcação fácil de usar, semelhante ao HTML, e compilando diretamente para código de máquina. Em tempo de execução, ele também possui um baixo consumo de recursos, o que é fundamental para sistemas embarcados. Em resumo, gostamos do Slint porque ele traz práticas comprovadas do desenvolvimento web e móvel para o ecossistema de sistemas embarcados.

## 105. SST

*Avalie*

SST é um framework para implantação de aplicações em ambientes de nuvem juntamente com o provisionamento de todos os serviços que a aplicação precisa para ser executada. SST não é só uma ferramenta de IaC; é um framework com com uma API TypeScript que permite que você defina oambiente de sua aplicação, um serviço que implanta sua aplicação quando acionado através de push do GIT, bem como uma interface gráfica de console para gerenciar a aplicação resultante e invocar os recursos de gerenciamento do SST. Embora o SST tenha sido originalmente baseado no AWS Cloud Formation e no CDK, sua versão mais recente foi implementada sobre o Terraform e o Pulumi. Então, na teoria, é agnóstico em relação à nuvem. O SST tem suporte nativo para a implantação de diversas estruturas de aplicativos da web padrão, incluindo Next.js e Remix, mas também oferece suporte a aplicações de API sem interface. O SST parece estar em uma categoria própria. Embora tenha alguma semelhança com ferramentas de orquestração de plataforma como o Kubevela, ele também fornece conveniências para pessoas desenvolvedoras, como um modo ao vivo que faz proxy de invocações do AWS Lambda de volta para uma função em execução na máquina local da pessoa desenvolvedora. No momento, o SST continua sendo uma curiosidade, mas é um projeto e uma categoria de ferramentas que vale a pena observar à medida que evolui.

## 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

A Thoughtworks é uma consultoria global de tecnologia que integra estratégia, design e engenharia de software para alavancar a inovação digital. Somos mais de 10,5 mil pessoas distribuídas entre 48 escritórios e em 19 países. Há mais de 30 anos, trabalhamos junto a nossas clientes para criar impacto extraordinário, usando a tecnologia como diferenciador para ajudá-las a resolver problemas de negócio complexos.

/thoughtworks

Estratégia. Design. Engenharia.