Technology Radar

Um guia de opinião sobre o universo de tecnologia atual

Volume 32
Abril 2025

/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 21 tecnologistas experientes da Thoughtworks. O TAB se reúne duas vezes por ano pessoalmente e quinzenalmente por vídeochamada. Sua principal atribuição é ser um grupo consultivo para nossa CTO Rachel Laycock.

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

Rachel Laycock (CTO)

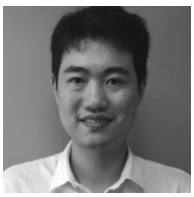
Martin Fowler (Chief Scientist)

Bharani Subramaniam

Birgitta Böckeler

Bryan Oliver

Camilla Falconi Crispim



Chris Chakrit Riddhagni

Effy Elden

Erik Dörnenburg

James Lewis

Ken Mugrage

Maya Ormaza

Mike Mason

Neal Ford

Ni Wang

Nimisha Asthagiri

Pawan Shah

Selvakumar Natesan

Shangqi Liu

Vanya Seth

Will Amaral

Créditos de Produção

Equipe Editorial

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

Design e Multimídia

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

Experiência Digital

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

Comunicação

- **Shalini Jagadish** — *Internal Communications Specialist*
- **Hiral Shah** — *Social Media Specialist*
- **Abhishek Kasegaonkar** — *Social Media Specialist*
- **Linda Horiuchi** — *Public Relations Specialist*
- **Kathryn Jansing** — *Public Relations Specialist*
- **Soumyajit Dey** — *Campaigns and Advertisement Specialist*
- **Anushree Tapuriah** — *Campaigns and Advertisement Specialist*

Tradução — Português

- Aloysio Chagas
- Ayrton Araujo
- Cássio Machado
- Cecilia Valim
- Daniele Motta
- Danielle Madrid
- Francisco Silva
- Gabriela Alves
- Guilherme Vandresen
- Gustavo Andrade
- Isabella Velleda
- Isaías Barroso
- Leonardo Berlatto
- Nina da Hora
- Patrick Prado
- Pietra Freitas
- Rafael Auday
- Renan Vizza
- Rodrigo Pasquale
- Taluna Mendes
- Thiago Albertins
- Vivianne Guimarães

Temas

Agentes supervisionados em assistentes de programação

Dois dos nossos temas destacam a rápida inovação em IA generativa, e um deles trata da crescente evolução dos assistentes de programação. Cada vez mais, essas ferramentas permitem que desenvolvedoras conduzam a implementação diretamente a partir de um chat com IA dentro do seu IDE — um modo também chamado de "agentic", "prompt-to-code" ou "chat-oriented programming (CHOP)". Nesta abordagem, os assistentes de IA não se limitam a responder a perguntas ou a gerar pequenos fragmentos de código; eles navegam e modificam o código, atualizam os testes, executam comandos e, em alguns casos, corrigem proativamente erros de linting e compilação. Enquanto permanecemos céticas em relação a agentes de programação que prometem desenvolvimento de tarefas grandes de maneira totalmente autônoma, vemos resultados promissores com essa abordagem supervisionada, em que as desenvolvedoras guiam e supervisionam as ações do agente. Cursor, Cline e Windsurf estão liderando essa tendência no espaço das ferramentas integradas ao IDE, com o GitHub Copilot também evoluindo nesse espaço. Assistentes agentes, como aider, goose e o Claude Code são alternativas baseadas em terminais. Apesar desses avanços, continuamos cautelosas sobre como isso aumenta a complacência com código gerado por IA, pois, apesar de alguns resultados muito bons, ainda vemos uma necessidade significativa de orientação e vigilância na revisão do código.

Observabilidade em evolução

Notamos um movimento significativo no espaço da observabilidade, impulsionado pela crescente complexidade das arquiteturas distribuídas. Ainda que a observabilidade sempre tenha sido crucial, ela continua a evoluir em paralelo ao resto do ecossistema de desenvolvimento de software. Um foco emergente é a observabilidade de modelos de linguagem de grande porte (LLMs), uma peça crítica na operacionalização de IA. Observamos também um aumento em ferramentas para monitoramento e avaliação de desempenho de LLM, incluindo Weights & Biases Weave, Arize Phoenix, Helicone e HumanLoop. Outra tendência é a integração de observabilidade assistida por IA, onde ferramentas utilizam IA para melhorar análises e insights. Além disso, a adoção crescente do OpenTelemetry está promovendo um ambiente de observabilidade mais padronizado, permitindo que as equipes permaneçam independentes de fornecedores (vendor-agnostic) e tenham mais flexibilidade na escolha das ferramentas. Várias ferramentas líderes de observabilidade — como Alloy, Tempo e Loki — agora suportam OpenTelemetry. A rápida inovação nas ferramentas de observabilidade demonstra uma crescente conscientização sobre a importância da observabilidade, criando um ciclo de práticas e tecnologias que se reforçam mutuamente.

R em RAG

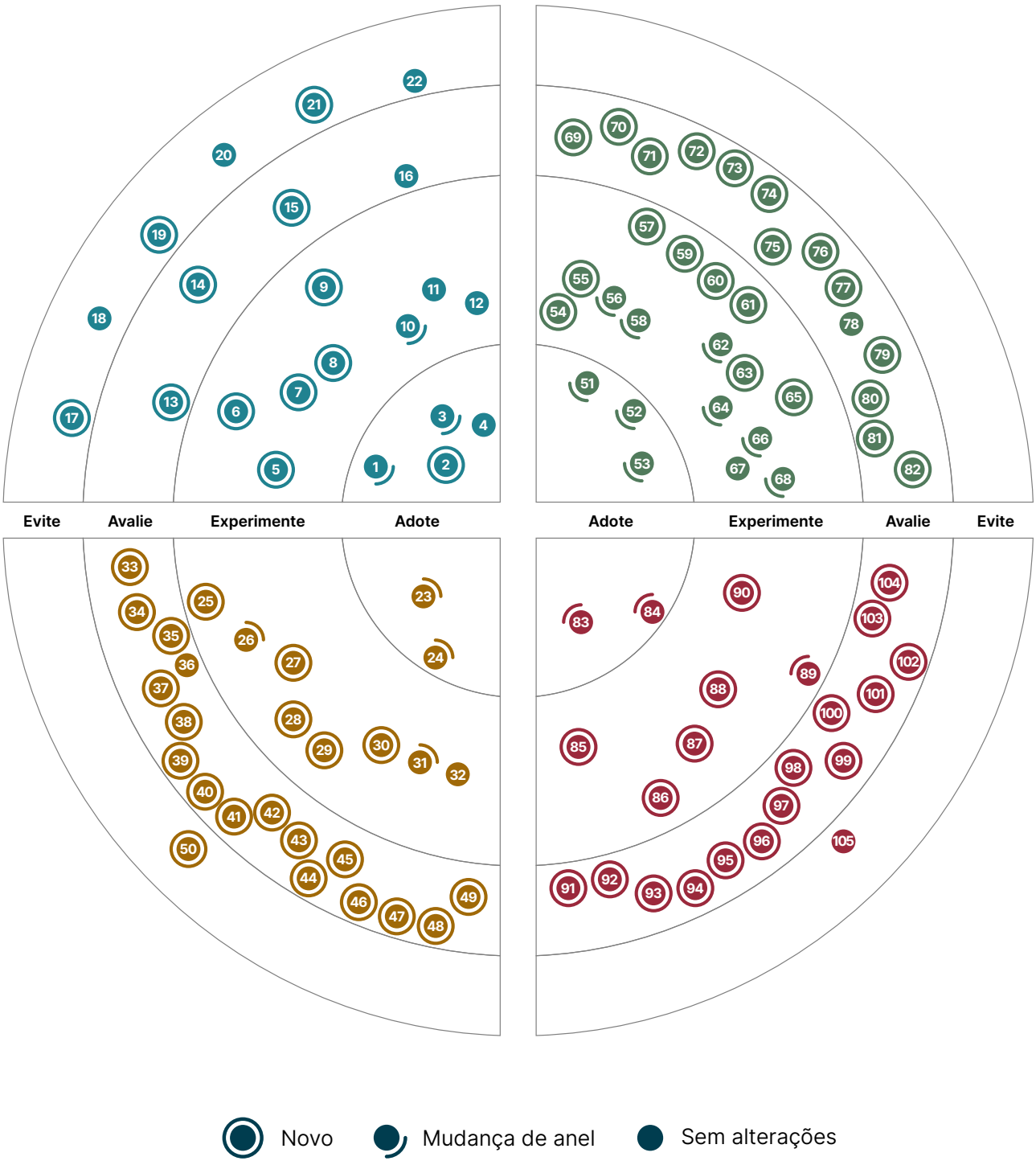
Esperamos que diferentes aspectos do ecossistema de IA generativa evoluam em ritmos variados, e nesta edição do Radar, vemos isso acontecendo com o "R" em RAG (geração aumentada por recuperação). Uma das interações principais com a caixa preta dos modelos de linguagem de grande porte (LLMs) é personalizar as entradas do prompt para gerar respostas relevantes e úteis. A crescente necessidade de recuperação eficaz em RAG levou ao surgimento de novas ferramentas e técnicas apresentadas nesta edição. Por exemplo, discutimos o RAG corretivo, que ajusta dinamicamente as respostas com base em feedback ou heurísticas; o Fusion-RAG, que combina múltiplas fontes e estratégias de recuperação para respostas mais abrangentes e robustas; e o Self-RAG, que evita completamente a etapa de recuperação, buscando dados sob demanda. Também destacamos o FastGraphRAG, que auxilia na compreensão ao criar gráficos navegáveis por humanos. Com base nas discussões e indicações da nossa equipe, o "R" em RAG é um tema quente e está evoluindo rapidamente.

Domando a fronteira dos dados

Big data tem sido uma preocupação fundamental para a indústria há muito tempo, mas as conversas em torno desta edição do Radar não se concentraram tanto no tamanho, e sim em como lidar com dados ricos e complexos. Com a crescente presença — e importância — de dados não estruturados nas empresas, garantir que os dados sejam gerenciados e organizados de maneira eficaz para que possam ser aproveitados com sucesso, desde aplicações de IA até análises de clientes, tornou-se essencial para os negócios.

Isso se reflete em muitos blips: desde ferramentas de bancos de dados vetoriais até produtos de análise como o Metabase. É surpreendente como grande parte do ecossistema de software está sendo impulsionada pelo que queremos e precisamos fazer com os dados. No entanto, não se trata apenas de ferramentas — nesta edição, destacamos a abordagem da mentalidade de produto para dados, um framework que incentiva as equipes a aplicarem os princípios da mentalidade de produto às partes analíticas de seu ecossistema. O surgimento dessa mentalidade é, em certa medida, um reflexo do desafio contínuo de utilizar os dados de forma adequada (algo discutido há anos, muito antes da ascensão da IA). O fato de esse conceito ter ganhado destaque — e estar presente em nossas conversas no Radar — demonstra que a necessidade de disciplina na gestão de dados é tão grande quanto sempre foi. Sem isso, as organizações podem ter dificuldades para inovar e acabar em desvantagem competitiva a médio e longo prazo.

O Radar



O Radar

Técnicas

Adote

1. Mentalidade de produto para dados
2. Teste de fuzzing
3. Lista de materiais de software
4. Modelagem de ameaças

Experimente

5. Coleção de requisições de API como artefato do produto
6. Processo de aconselhamento arquitetural
7. GraphRAG
8. Gerenciamento de acesso privilegiado sob demanda
9. Destilação de modelos
10. Engenharia de prompt
11. Modelos de linguagem de pequeno porte
12. Usando GenIA para entender bases de código legadas

Avalie

13. Design de código compatível com IA
14. Teste de UI baseado em IA
15. Envoltório de competência como um modelo para entender as falhas de um sistema
16. Saída estruturada de LLMs

Evite

17. TI invisível acelerada por IA
18. Complacência com código gerado por IA
19. Assistentes de programação locais
20. Substituição da programação em pares por IA
21. ETL reverso
22. SAFe™

Plataformas

Adote

23. GitLab CI/CD
24. Trino

Experimente

25. ABsmartly
26. Dapr
27. Grafana Alloy
28. Grafana Loki
29. Grafana Tempo
30. Railway
31. Unblocked
32. Pesos & Vieses

Avalie

33. Arize Phoenix
34. Chainloop
35. Deepseek R1
36. Deno
37. Graphiti
38. Helicone
39. Humanloop
40. Model Context Protocol (MCP)
41. Open WebUI
42. pg_mooncake
43. Modelos de raciocínio
44. Restate
45. Supabase
46. Synthesized
47. Tonic.ai
48. turbopuffer
49. VectorChord

Evite

50. Tyk hybrid API management

Ferramentas

Adote

51. Renovate
52. uv
53. Vite

Experimente

54. Claude Sonnet
55. Cline
56. Cursor
57. D2
58. Databricks Delta Live Tables
59. JSON Crack
60. MailSlurp
61. Metabase
62. NeMo Guardrails
63. Nyx
64. OpenRewrite
65. Plerion
66. Agentes de engenharia de software
67. Tuple
68. Turborepo

Avalie

69. AnythingLLM
70. Gemma Scope
71. Hurl
72. Jujutsu
73. kubenetmon
74. Mergiraf
75. ModernBERT
76. OpenRouter
77. Redactive
78. System Initiative
79. TabPFN
80. v0
81. Windsurf
82. YOLO

Evite

—

Linguagens e Frameworks

Adote

83. OpenTelemetry
84. React Hook Form

Experimente

85. Effect
86. Hasura GraphQL engine
87. LangGraph
88. MarkItDown
89. Federação de módulos
90. Prisma ORM

Avalie

91. .NET Aspire
92. Android XR SDK
93. Browser Use
94. Crew AI
95. ElysiaJs
96. FastGraphRAG
97. Gleam
98. GoFr
99. Criptografia pós-quântica em Java
100. Presidio
101. PydanticAI
102. Swift para aplicações com restrições de recursos
103. Tamagui
104. torchtune

Evite

105. Uso excessivo de Node

Técnicas

Adote

1. Mentalidade de produto para dados
2. Teste de fuzzing
3. Lista de materiais de software
4. Modelagem de ameaças

Experimente

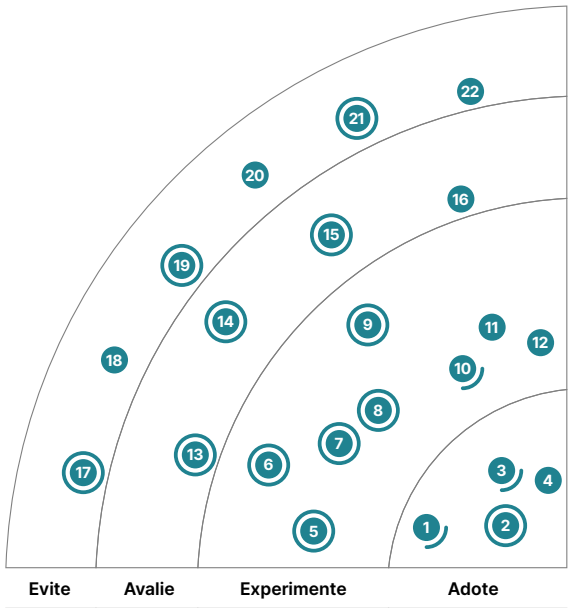
5. Coleção de requisições de API como artefato do produto
6. Processo de aconselhamento arquitetural
7. GraphRAG
8. Gerenciamento de acesso privilegiado sob demanda
9. Destilação de modelos
10. Engenharia de prompt
11. Modelos de linguagem de pequeno porte
12. Usando GenIA para entender bases de código legadas

Avalie

13. Design de código compatível com IA
14. Teste de UI baseado em IA
15. Envoltório de competência como um modelo para entender as falhas de um sistema
16. Saída estruturada de LLMs

Evite

17. TI invisível acelerada por IA
18. Complacência com código gerado por IA
19. Assistentes de programação locais
20. Substituição da programação em pares por IA
21. ETL reverso
22. SAFe™



Novo Mudanças de anel Sem alterações

1. Mentalidade de produto para dados

Adote

As organizações adotam ativamente a mentalidade de produto para dados como uma prática padrão para gerenciar os ativos de dados. Essa abordagem trata os dados como um produto com seu próprio ciclo de vida, padrões de qualidade e foco em atender às necessidades da pessoa consumidora. Agora a recomendamos como padrão para o gerenciamento de dados, independentemente de as organizações escolherem arquiteturas como malha de dados (data mesh) ou lakehouse.

Enfatizamos a importância do foco na pessoa consumidora na mentalidade de produto para dados, com o fim de promover maior adoção e realização de valor. Isso implica projetar produtos de dados antes dos casos de uso. Também damos foco em capturar e gerenciar metadados relevantes para os negócios e metadados técnicos usando catálogos de dados modernos como DataHub, Collibra, Atlan e Informatica. Essas práticas melhoram a capacidade de descoberta e a usabilidade dos dados. Além disso, adotamos a mentalidade de produto para dados para escalar iniciativas de IA e criar dados prontos para IA. Essa abordagem inclui o gerenciamento abrangente do ciclo de vida, garantindo que os dados não sejam apenas bem administrados e de alta qualidade, mas também descartados em conformidade com os requisitos legais e regulamentares quando não forem mais necessários.

2. Teste de fuzzing

Adote

Teste de fuzzing, ou simplesmente fuzzing, é uma técnica de teste que existe há muito tempo, mas ainda é uma das menos conhecidas. O seu objetivo é fornecer a um sistema de software todos os tipos de entradas inválidas e observar o seu comportamento. Para um endpoint HTTP, por exemplo, requisições inválidas deveriam resultar em erros 4*xx*, mas o teste de fuzzing provoca erros 5*xx* ou piores. Bem documentado e suportado por ferramentas, o teste de fuzzing se tornou mais relevante do que nunca, especialmente com o aumento do código gerado por IA e sua complacência. Isto significa que é uma boa hora para adotar o teste de fuzzing para garantir que o código continue robusto e seguro.

3. Lista de materiais de software

Adote

Desde o nosso blip inicial em 2021, a geração de lista de materiais de software (SBOM) passou de uma prática emergente para um um padrão sensato em nossos projetos. O ecossistema amadureceu significativamente, oferecendo um conjunto de ferramentas robusto e integração perfeita com CI/CD. Ferramentas como Syft, Trivy e Snyk fornecem uma geração de SBOM, desde código fonte até imagens de contêineres e análises de vulnerabilidades. Plataformas como a FOSSA e a Chainloop aprimoram a gestão de riscos de segurança ao se integrarem com o fluxo de desenvolvimento e aplicarem políticas de segurança. Embora um padrão universal de SBOM ainda esteja evoluindo, o amplo suporte ao SPDX e ao CycloneDX minimizou os desafios de adoção. Sistemas de IA também exigem SBOMs, como demonstrado pelo Código de Prática de Cibersegurança em IA do governo do Reino Unido e pelo Playbook de Colaboração em Cibersegurança em IA da CISA. Continuaremos monitorando os avanços nesse espaço.

4. Modelagem de ameaças

Adote

No cenário em constante evolução de desenvolvimento de software impulsionado por IA, a modelagem de ameaças é mais essencial do que nunca para construir software seguro, mantendo a agilidade e evitando o "sanduíche de segurança". A modelagem de ameaças — um conjunto de

técnicas para identificar e classificar ameaças potenciais — aplica-se em vários contextos, incluindo aplicações de IA generativa, que introduzem riscos de segurança únicos. Para ser eficaz, ela deve ser realizada regularmente ao longo do ciclo de vida do software e funciona melhor em conjunto com outras práticas de segurança. Essas incluem definir requisitos de segurança interfuncionais para abordar riscos comuns nas tecnologias do projeto e utilizar scanners de segurança automatizados para monitoramento contínuo.

5. Coleção de requisições de API como artefato do produto

Experimente

Tratar APIs como produto significa priorizar a experiência da pessoa desenvolvedora, não apenas inserindo um design sensato e padronizado nas próprias APIs, mas também fornecendo documentações abrangentes e boas experiências de onboarding. Embora as especificações de OpenAPI (Swagger) possam documentar as interfaces de APIs efetivamente, o onboarding permanece um desafio. As desenvolvedoras precisam de acesso rápido a exemplos funcionais, com autenticação pré-configurada e dados de testes realistas. Com o amadurecimento de ferramentas de clientes de API (como Postman, Bruno e Insomnia), recomendamos tratar a coleção de requisições de API como um artefato do produto. Essas coleções devem ser cuidadosamente projetadas para guiar efetivamente as desenvolvedoras através dos principais fluxos de trabalho, auxiliando na compreensão da funcionalidade e linguagem de domínio da API com o mínimo de esforço. Para manter as coleções atualizadas, recomendamos armazená-las em um repositório e integrar ao pipeline de entrega das APIs.

6. Processo de aconselhamento arquitetural

Experimente

Um dos desafios persistentes em grandes equipes de software é determinar quem toma as decisões arquiteturais que moldam a evolução dos sistemas. O Relatório sobre o Estado do DevOps revela que a abordagem tradicional dos Conselhos de Revisão de Arquitetura é contraproducente, muitas vezes prejudicando o fluxo de trabalho e correlacionando-se com baixo desempenho organizacional. Uma alternativa convincente é um processo de aconselhamento arquitetural — uma abordagem descentralizada onde qualquer pessoa pode tomar qualquer decisão arquitetural, desde que busque aconselhamento daqueles afetados e daqueles com experiência relevante. Este método permite que as equipes otimizem o fluxo sem comprometer a qualidade arquitetural, tanto em pequenas quanto em grandes escalas. À primeira vista, essa abordagem pode parecer controversa, mas práticas como Registros de Decisão de Arquitetura e fóruns consultivos garantem que as decisões permaneçam informadas, enquanto capacitam aqueles mais próximos do trabalho a tomar decisões. Temos visto este modelo ter sucesso em escala em um número crescente de organizações, incluindo aquelas em indústrias altamente regulamentadas.

7. GraphRAG

Experimente

Em nossa última atualização do RAG, introduzimos o GraphRAG, originalmente descrito no artigo da Microsoft como uma abordagem de duas etapas: (1) divisão de documentos em segmentos e uso de uma análise baseada em modelos de linguagem de grande porte (LLMs) dos segmentos para criar um gráfico de conhecimento; (2) recuperação de segmentos relevantes no momento da consulta por meio de embeddings, enquanto seguimos as bordas no gráfico de conhecimento para descobrir segmentos relacionados adicionais, que são então adicionados ao prompt aumentado. Em muitos casos, essa abordagem aprimora as respostas geradas pelo LLM. Observamos benefícios semelhantes ao usar o GenAI para entender bases de código legadas, em que usamos informações estruturais — como

árvores de sintaxe abstratas e dependências — para criar o gráfico de conhecimento. O padrão GraphRAG ganhou força com o surgimento de ferramentas e estruturas como o pacote GraphRAG Python da Neo4j para oferecer suporte a ele. Também consideramos que o Graphiti se encaixa em uma interpretação mais ampla do GraphRAG como um padrão.

8. Gerenciamento de acesso privilegiado sob demanda

Experimente

O princípio de menor privilégio garante que sistemas e usuárias tenham apenas o acesso mínimo necessário para realizar suas tarefas. O abuso de credenciais privilegiadas é um fator significativo por trás das violações de segurança, sendo a escalada de privilégios um vetor de ataque comum. Frequentemente, as invasoras começam com acesso de baixo nível e exploram vulnerabilidades de software ou configurações incorretas para obter privilégios de administrador, especialmente quando as contas têm direitos excessivos ou desnecessários. Outro risco negligenciado são os privilégios permanentes — acessos privilegiados disponíveis continuamente, que ampliam a superfície de ataque. O gerenciamento de acesso privilegiado sob demanda mitiga esse risco, concedendo o acesso privilegiado apenas quando necessário e revogando-o imediatamente após o uso, minimizando a exposição. Um modelo de segurança que aplica rigorosamente o princípio de menor privilégio garante que usuárias, aplicações e sistemas tenham apenas os direitos essenciais pelo menor tempo possível – um requisito crítico para conformidade e segurança regulatória. Nossas equipes implementaram essa abordagem por meio de um fluxo de trabalho automatizado, que aciona um processo leve de aprovação, atribui funções temporárias com acesso restrito e impõe um tempo máximo de validade (TTL) para cada função, garantindo que os privilégios expirem automaticamente assim que a tarefa for concluída.

9. Destilação de modelos

Experimente

As leis de escala têm sido um fator essencial no avanço da IA — o princípio de que modelos maiores, conjuntos de dados mais extensos e maior poder computacional resultam em sistemas de IA mais poderosos. No entanto, hardwares de consumo e dispositivos de borda frequentemente não possuem capacidade suficiente para suportar modelos em larga escala, criando a necessidade da destilação de modelos.

A destilação de modelos transfere conhecimento de um modelo maior e mais potente (professor) para um modelo menor e mais eficiente (aluno). O processo normalmente envolve a geração de um conjunto de dados amostral a partir do modelo professor e o ajuste fino do modelo aluno para capturar suas propriedades estatísticas. Diferente da poda ou da quantização, que reduzem modelos removendo parâmetros, a destilação busca preservar o conhecimento específico do domínio, minimizando a perda de precisão. Além disso, ela pode ser combinada com quantização para otimização adicional.

Originalmente proposta por Geoffrey Hinton et al., a destilação de modelos tem sido amplamente adotada. Um exemplo notável é a versão destilada do Qwen/Llama do DeepSeek R1, que mantém fortes capacidades de raciocínio em modelos menores. Com sua crescente maturidade, a técnica não está mais restrita a laboratórios de pesquisa; agora é aplicada em projetos industriais e pessoais. Provedores como OpenAI e Amazon Bedrock oferecem guias para ajudar desenvolvedoras a destilar seus próprios modelos de linguagem de pequeno porte (SLMs). Acreditamos que a adoção da destilação de modelos pode ajudar as organizações a gerenciar os custos de implantação de modelos de linguagem de grande porte (LLMs), ao mesmo tempo em que desbloqueia o potencial da inferência de LLM em dispositivos.

10. Engenharia de prompt

Experimente

Engenharia de prompt se refere ao processo de projetar e refinar prompts de modelos de IA generativa para produzir respostas de alta qualidade e contextualizadas. Isso envolve a elaboração de prompts claros, específicos e relevantes para tarefas sob medida ou aplicações, para otimizar a saída do modelo. À medida que as capacidades dos modelos de linguagem de grande porte (LLMs) evoluem, particularmente com o surgimento dos modelos de raciocínio, as práticas da engenharia de prompt devem também se adaptar. Baseado em nossa experiência em geração de código com IA, nós observamos que o few-shot prompting pode ter uma performance inferior ao zero-shot prompting quando trabalhado com modelos de raciocínio. Adicionalmente, o amplamente utilizado chain-of-thought (CoT) pode degradar a performance de modelos de raciocínio — provavelmente porque a aprendizagem por reforço já tem ajuste fino embutido no macanismo CoT.

Nossa experiência prática está alinhada com as pesquisas acadêmicas, que sugerem que "modelos avançados podem eliminar a necessidade de engenharia de prompt na engenharia de software." No entanto, técnicas de engenharia de prompt tradicionais ainda desempenham um papel crucial na redução de alucinações e na melhoria da qualidade das respostas, especialmente considerando as diferenças em tempo de resposta e custos de token entre modelos de raciocínio e LLMs genéricos. Ao construir aplicações de agente, nós recomendamos a escolha estratégica dos modelos com base nas suas necessidades, enquanto continua a refinar seus modelos de prompt e técnicas correspondentes. Encontrar o equilíbrio certo entre desempenho, tempo de resposta e custo de token continua sendo essencial para maximizar a eficácia dos LLMs.

11. Modelos de linguagem de pequeno porte

Experimente

O anúncio recente do DeepSeek R1 é um ótimo exemplo de por que modelos de linguagem de pequeno porte (SLMs) permanecem interessantes. O R1 em tamanho real tem 671 bilhões de parâmetros e requer cerca de 1.342 GB de VRAM para funcionar, o que só é possível usando um mini cluster de oito GPUs NVIDIA de última geração. Mas o DeepSeek também está disponível destilado em Qwen e Llama — modelos menores e de peso aberto — transferindo efetivamente suas habilidades e permitindo que seja executado em hardware muito mais modesto. Embora o modelo perca algum desempenho nesses tamanhos menores, ele ainda permite um grande salto de desempenho em relação aos modelos de linguagem de pequeno porte anteriores. O espaço dos SLMs continua a inovar em outros lugares também. Desde o último Radar, a Meta introduziu o Llama 3.2 nos tamanhos 1B e 3B, a Microsoft lançou o Phi-4, oferecendo resultados de alta qualidade com um modelo 14B, e o Google lançou o PaliGemma 2, um modelo de linguagem de visão nos tamanhos 3B, 10B e 28B. Esses são apenas alguns dos modelos menores que estão sendo lançados, consolidando uma tendência importante a ser acompanhada.

12. Usando GenIA para entender bases de código legadas

Experimente

Nos últimos meses, o uso de GenIA para entender bases de código legadas tem avançado significativamente. Ferramentas populares, como o GitHub Copilot, estão sendo destacadas como capazes de ajudar na modernização de bases de código legadas. Ferramentas como o Cody, da Sourcegraph, estão facilitando a navegação e compreensão de bases de código inteiras. Essas ferramentas utilizam diversas técnicas de GenIA para fornecer ajuda contextual, simplificando o trabalho com sistemas legados complexos. Além disso, frameworks especializados como o S3LLM estão demonstrando como modelos de linguagem de grande porte (LLMs) podem lidar com softwares científicos em larga escala — como aqueles escritos em Fortran ou Pascal — trazendo uma

compreensão aprimorada por GenIA para bases de código fora do tradicional ambiente corporativo de TI. Acreditamos que essa abordagem continuará ganhando força, dado o enorme volume de software legado existente no mundo.

13. Design de código compatível com IA

Avalie

Agentes de engenharia de software supervisionados estão cada vez mais capazes de identificar atualizações necessárias e fazer alterações maiores em uma base de código. Ao mesmo tempo, vemos uma crescente complacência com código gerado por IA e desenvolvedoras demonstrando resistência em revisar grandes conjuntos de mudanças feitas por IA. Uma justificativa comum para isso é a ideia de que a legibilidade do código voltada para humanos importa menos, já que a IA pode lidar com modificações futuras. No entanto, assistentes de código baseados em IA também têm um desempenho melhor em bases de código bem estruturadas, tornando o design de código compatível com IA essencial para a manutenção.

Felizmente, boas práticas de design de software para humanos também beneficiam a IA. Nomes bem definidos fornecem contexto de domínio e funcionalidade; modularidade e abstrações mantêm o contexto da IA gerenciável ao limitar as mudanças necessárias; e o princípio DRY (“don’t repeat yourself”, ou “não se repita”) reduz a duplicação de código, facilitando para a IA manter a consistência do comportamento. Até agora, os melhores padrões compatíveis com IA estão alinhados às boas práticas já estabelecidas. À medida que a IA evolui, mais padrões específicos devem surgir, por isso, considerar o design de código com isso em mente será extremamente útil.

14. Teste de UI baseado em IA

Avalie

Técnicas novas de assistência baseada em IA para equipes de software estão surgindo além da simples geração de código. Uma área que está ganhando destaque é o teste de UI baseado em IA , aproveitando a capacidade dos modelos de linguagem de grande porte (LLMs) de interpretar interfaces gráficas de usuário. Existem várias abordagens para isso. Uma categoria de ferramentas utiliza LLMs multimodais ajustados para o processamento de snapshots, permitindo que scripts de teste sejam escritos em linguagem natural para navegar em uma aplicação. Exemplos nessa área incluem QA.tech ou LambdaTests’ KaneAI. Outra abordagem, vista no Browser Use, combina modelos fundacionais multimodais com Playwright, oferecendo insights sobre a estrutura de uma página em vez de depender de modelos ajustados.

Ao integrar testes de UI baseados em IA em uma estratégia de testes, é crucial considerar onde eles agregam mais valor. Esses métodos podem complementar os testes exploratórios manuais e, embora ao não determinismo dos LLMs possa introduzir instabilidades, sua flexibilidade pode ser uma vantagem. Isso pode ser útil para testar aplicações legadas com seletores ausentes ou aplicações que frequentemente mudam rótulos e caminhos de cliques.

15. Envoltório de competência como um modelo para entender as falhas de um sistema

Avalie

A teoria da extensibilidade graciosa define as regras básicas que regem os sistemas adaptativos, incluindo os sistemas sociotécnicos envolvidos na criação e operação de software. Um conceito fundamental nessa teoria é o envoltório de competência - o limite dentro do qual um sistema pode funcionar *robustamente* diante de uma falha. Quando um sistema é levado além de seu envoltório de competência, ele se torna frágil e tem maior probabilidade de falhar. Esse modelo fornece uma lente

valiosa para entender a falha do sistema, como visto nas falhas complexas que levaram à interrupção do Canva em 2024. A Teoria da Residualidade, um desenvolvimento recente no pensamento da arquitetura de software, oferece uma maneira de testar o envoltório de competência de um sistema introduzindo deliberadamente fatores de estresse e analisando como o sistema se adaptou aos fatores de estresse históricos ao longo do tempo. As abordagens se alinham com os conceitos de antifragilidade, resiliência e robustez em sistemas sociotécnicos, e estamos ansiosas para ver aplicações práticas dessas ideias no setor.

16. Saída estruturada de LLMs

Avalie

Saída estruturada de LLMs se refere à 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 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.

17. TI invisível acelerada por IA

Evite

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

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

18. Complacência com código gerado por IA

Evite

À medida que os assistentes de programação de IA continuam a ganhar força, o mesmo acontece com o crescente conjunto de dados e pesquisas que destacam as preocupações sobre a complacência com código gerado por IA. A mais recente pesquisa de qualidade de código da GitClear mostra que, em 2024, códigos duplicados e a rejeição de códigos aumentaram ainda mais do que o previsto, enquanto a atividade de refatoração nos históricos de commit diminuiu. Também refletindo

isso, uma pesquisa da Microsoft sobre "trabalhadoras do conhecimento" descobriu que a confiança impulsionada pela IA geralmente ocorre às custas do pensamento crítico (um padrão que observamos à medida que a complacência se instala com o uso prolongado de assistentes de programação). O surgimento de agentes de engenharia de software supervisionados amplia ainda mais os riscos, pois quando a IA gera conjuntos de alterações cada vez maiores, as desenvolvedoras enfrentam desafios maiores na revisão dos resultados. O surgimento da "vibe coding", em que desenvolvedoras permitem que a IA gere código com revisão mínima, ilustra a confiança crescente nos resultados gerados pela IA. Embora essa abordagem possa ser apropriada para coisas como protótipos ou outros tipos de código descartável, advertimos fortemente contra seu uso para código de produção.

19. Assistentes de programação locais

Evite

As organizações continuam cautelosas em relação aos assistentes de programação com IA de terceiros, especialmente devido a preocupações com a confidencialidade do código. Como resultado, muitas desenvolvedoras estão considerando o uso de assistentes de programação locais — IAs que rodam inteiramente em suas máquinas — eliminando a necessidade de enviar código para servidores externos. No entanto, os assistentes locais ainda ficam atrás das versões baseadas na nuvem, que utilizam modelos maiores e mais avançados. Mesmo em máquinas de alto desempenho, os modelos menores continuam apresentando limitações. Observamos que eles têm dificuldades com prompts complexos, não possuem uma janela de contexto suficientemente grande para problemas mais amplos e, muitas vezes, não conseguem ativar integrações de ferramentas ou chamadas de funções. Essas capacidades são especialmente cruciais para os workflows baseados em agentes, que representam o estado da arte na assistência à programação atualmente.

Portanto, embora seja importante ter expectativas moderadas, algumas funcionalidades ainda são viáveis localmente. Algumas IDEs populares agora incorporam modelos menores em seus recursos principais, como a conclusão preditiva de código do Xcode e a completação de código de linha inteira da JetBrains. Além disso, LLMs que podem ser executados localmente, como o Qwen Coder, representam um avanço para sugestões *inline* locais e o manuseio de consultas simples de programação. Você pode testar essas capacidades com o Continue, que suporta a integração de modelos locais por meio de *runtimes* como o Ollama.

20. 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? O GitHub Copilot chama a si mesmo de seu programador em par de IA. Enquanto acreditamos que um assistente de programação pode trazer alguns dos benefícios da programação em pares, nós desaconselhamos totalmente substituir programação em pares por IA. Visualizar assistentes de código para o pareamento ignora um dos benefícios chave da programação em pares: tornar o time, não apenas os indivíduos que contribuem, melhor. Assistentes de programação 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. No entanto, eles não contribuem para os benefícios da colaboração em equipe, como manter o trabalho em andamento em um nível baixo, reduzir handoffs e reaprendizados, possibilitar a integração contínua ou melhorar a propriedade coletiva do código.

21. ETL reverso

Evite

Estamos observando uma proliferação preocupante do chamado ETL reverso. Os processos tradicionais de ETL têm seu lugar nas arquiteturas de dados convencionais, onde transferem dados de sistemas de processamento de transações para um sistema centralizado de análise, como um data warehouse ou um data lake. Embora essa arquitetura tenha limitações bem documentadas — muitas das quais são abordadas por uma abordagem de malha de dados —, ela ainda é comum em empresas. Numa arquitetura desse tipo, mover dados de volta de um sistema central de análise para um sistema de transações faz sentido em certos casos — por exemplo, quando o sistema central pode agregar dados de várias fontes ou como parte de uma arquitetura transicional ao migrar para uma malha de dados. No entanto, estamos observando uma tendência crescente em que fornecedoras de produtos usam ETL reverso como uma desculpa para mover cada vez mais lógica de negócios para uma plataforma centralizada — o produto delas. Essa abordagem agrava muitos dos problemas causados pelas arquiteturas de dados centralizadas, e sugerimos extrema cautela ao introduzir fluxos de dados de uma plataforma centralizada e abrangente para sistemas de processamento de transações.

22.SAFe™

Evite

Vemos uma adoção contínua do SAFe™ (Scaled Agile Framework®). Também continuamos a observar que os processos de SAFe, padronizados em excesso e com etapas faseadas, criam fricção, podem promover silos e que seu controle de cima para baixo gera desperdício no fluxo de valor e desestimula a criatividade dos talentos de engenharia, ao mesmo tempo em que limita a autonomia e a experimentação das equipes. Um dos principais motivos para a adoção do SAFe é a complexidade de tornar uma organização ágil, com empresas esperando que um framework como esse ofereça um atalho simples, baseado em processos, para alcançar a agilidade. Dada a adoção generalizada do SAFe — inclusive entre nossas clientes — treinamos mais de 100 consultoras da Thoughtworks para oferecer melhor suporte a elas. Apesar desse conhecimento aprofundado e de muita tentativa, chegamos à conclusão de que, às vezes, simplesmente não existe uma solução simples para um problema complexo e continuamos a recomendar abordagens e governança mais enxutas e orientadas por valor que funcionem em conjunto com um programa abrangente de mudança.

Scaled Agile Framework® e SAFe™ são marcas registradas da Scaled Agile, Inc.

Plataformas

Adote

23. GitLab CI/CD
24. Trino

Experimente

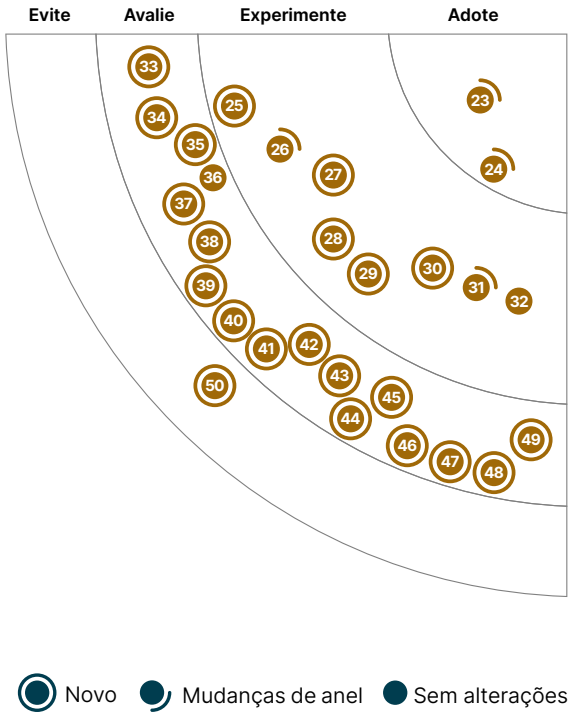
25. ABsmartly
26. Dapr
27. Grafana Alloy
28. Grafana Loki
29. Grafana Tempo
30. Railway
31. Unblocked
32. Pesos & Vieses

Avalie

33. Arize Phoenix
34. Chainloop
35. Deepseek R1
36. Deno
37. Graphiti
38. Helicone
39. Humanloop
40. Model Context Protocol (MCP)
41. Open WebUI
42. pg_mooncake
43. Modelos de raciocínio
44. Restate
45. Supabase
46. Synthesized
47. Tonic.ai
48. turbopuffer
49. VectorChord

Evite

50. Tyk hybrid API management



23. GitLab CI/CD

Adote

O GitLab CI/CD evoluiu para um sistema totalmente integrado no GitLab, abrangendo tudo, desde a integração e o teste de código até a implantação e o monitoramento. Ele oferece suporte a fluxos de trabalho complexos com recursos como pipelines de vários estágios, armazenamento em cache, execução paralela e executores de dimensionamento automático e é adequado para projetos de grande escala e necessidades complexas de pipeline. Queremos destacar suas ferramentas integradas de segurança e conformidade (como a análise SAST e DAST), que o tornam adequado para casos de uso com altos requisitos de conformidade. Ele também se integra perfeitamente ao Kubernetes, dando suporte a fluxos de trabalho nativos da nuvem, e oferece registro em tempo real, relatórios de teste e rastreabilidade para melhorar a observabilidade.

24. Trino

Adote

Trino é um mecanismo de consulta SQL distribuído e de código aberto projetado para consultas analíticas interativas sobre big data. Ele é otimizado para executar ambientes locais e na nuvem e oferece suporte à consulta de dados onde eles residem, incluindo bancos de dados relacionais e vários armazenamentos de dados proprietários por meio de conectores. Trino também pode consultar dados armazenados em formato de arquivos como Parquet e formato de tabela aberta como Apache Iceberg. Os recursos de federação de consultas integrados permitem que dados de diversas fontes sejam consultados como uma única tabela lógica, o que o torna uma ótima opção para cargas de trabalho analíticas que exigem agregação de dados de diversas fontes. Trino é uma parte fundamental de stacks populares como AWS Athena, Starburst e outras plataformas de dados proprietárias. Nossas equipes o utilizaram com sucesso em vários casos de uso, e quando se trata de consultar um conjunto de dados em diversas fontes para análise, Trino tem sido uma escolha confiável.

25. ABsmartly

Experimente

A ABsmartly é uma plataforma avançada de teste e experimentação A/B projetada para uma tomada de decisão rápida e confiável. Seu principal diferencial é o mecanismo de Group Sequential Testing (GST), que acelera os resultados dos testes em até 80% em comparação com as ferramentas tradicionais de teste A/B. A plataforma oferece relatórios em tempo real, segmentação aprofundada dos dados e integração completa através de uma abordagem baseada em APIs, permitindo experimentos em aplicações web, mobile, microsserviços e modelos de machine learning.

A ABsmartly enfrenta os principais desafios da experimentação escalável orientada por dados, possibilitando iterações mais rápidas e um desenvolvimento de produtos mais ágil. Sua execução sem latência, capacidade avançada de segmentação e suporte a experimentos multiplataforma tornam a ferramenta especialmente valiosa para organizações que desejam escalar uma cultura de experimentação e priorizar a inovação baseada em dados. Ao reduzir significativamente os ciclos dos testes e automatizar a análise dos resultados, a ABsmartly nos ajudou a otimizar funcionalidades e experiências do usuário de forma mais eficiente do que com as plataformas tradicionais de testes A/B.

26. Dapr

Experimente

O Dapr evoluiu consideravelmente desde que nós o apresentamos no Radar. Suas muitas novas funções incluem agendamento de jobs, atores virtuais, bem como políticas de retry mais sofisticadas e componentes de observabilidade. Sua lista de blocos de construção continua a crescer, incluindo

jobs, criptografia e muito mais. Nossas equipes também destacam o foco crescente em padrões seguros, com suporte para mTLS e imagens sem estado (ou seja, sem um sistema operacional completo, incluindo apenas binários e dependências necessárias). No geral, estamos satisfeitas com o Dapr e ansiosas para futuros desenvolvimentos.

27. Grafana Alloy

Experimente

Anteriormente conhecido como Grafana Agent, Grafana Alloy é um coletor do OpenTelemetry de código aberto. Alloy é projetado para ser um coletor de telemetria tudo-em-um para todos os dados de telemetria, incluindo logs, métricas e rastreamentos. Ele suporta a coleta de formatos de dados de telemetria comumente usados, como OpenTelemetry, Prometheus e Datadog. Com a recente descontinuação do Promtail, o Alloy está emergindo como uma escolha preferencial para a coleta de dados de telemetria — especialmente para logs — se você está usando o conjunto de ferramentas de observabilidade Grafana.

28. Grafana Loki

Experimente

Grafana Loki é um sistema de agregação de logs multi-tenant com alta disponibilidade e escalabilidade horizontal, inspirado no Prometheus. O Loki indexa apenas metadados sobre seus logs como um conjunto de rótulos para cada fluxo de log. Os dados de log são armazenados em uma solução de armazenamento em blocos, como S3, GCS ou Azure Blob Storage. A vantagem é que o Loki promete reduzir a complexidade operacional e os custos de armazenamento em comparação com outros concorrentes. Como era de se esperar, ele se integra perfeitamente ao Grafana e ao Grafana Alloy, embora outros mecanismos de coleta também possam ser usados.

Loki 3.0 introduziu suporte nativo a OpenTelemetry, tornando a ingestão e integração com sistemas baseados em OpenTelemetry tão simples quanto configurar um endpoint. Ele também oferece recursos avançados de multi-tenancy, como isolamento de inquilinos via “shuffle-sharding”, o que evita que inquilinos com mau comportamento (por exemplo, consultas pesadas ou indisponibilidades) afetem outros em um cluster. Se você não vem acompanhando as novidades no ecossistema Grafana, agora é um ótimo momento para dar uma olhada, pois está evoluindo rapidamente.

29. Grafana Tempo

Experimente

Grafana Tempo é um backend de tracing distribuído em larga escala que suporta padrões abertos como OpenTelemetry. Projetado para ser eficiente em custo, ele utiliza armazenamento de objetos para retenção de traces a longo prazo e permite busca de traces, geração de métricas baseada em spans e correlação com logs e métricas. Por padrão, o Grafana Tempo usa um formato de blocos em colunas baseado no Apache Parquet, melhorando a performance das consultas e permitindo que outras ferramentas downstream acessem os dados de trace. As consultas são feitas via TraceQL e Tempo CLI. O Grafana Alloy também pode ser configurado para coletar e encaminhar traces para o Grafana Tempo. Nossas equipes usaram o Grafana Tempo no GKE, utilizando MinIO para armazenamento de objetos, coletores OpenTelemetry e Grafana para visualização de traces.

30. Railway

Experimente

Heroku costumava ser uma excelente escolha para muitas desenvolvedoras que desejavam lançar e implantar seus aplicativos rapidamente. Nos últimos anos, também vimos o surgimento de plataformas

de implantação como Vercel, que são mais modernas, leves e fáceis de usar, mas projetadas para aplicações frontend. Uma alternativa full-stack neste espaço é o Railway, uma plataforma de nuvem PaaS (em português, "plataforma como serviço") que simplifica tudo, desde a implantação de GitHub/Docker até a observabilidade de produção.

Railway oferece suporte à maioria dos frameworks de programação convencionais, bancos de dados, bem como implantação em contêineres. Como uma plataforma hospedada de longo prazo para um aplicativo, você pode precisar comparar cuidadosamente os custos de diferentes plataformas. Atualmente, nossa equipe tem uma boa experiência com a implantação e observabilidade do Railway. A operação é suave e pode ser bem integrada com as práticas de implantação contínua que defendemos.

31. Unblocked

Experimente

Unblocked é um assistente de equipe baseado em IA pronto para uso. Uma vez integrado com repositórios de base de código, plataformas de documentação corporativa, ferramentas de gerenciamento de projetos e ferramentas de comunicação, o Unblocked ajuda a responder perguntas sobre conceitos técnicos e de negócios complexos, design arquitetural e implementação, bem como processos operacionais. Isso é particularmente útil para navegar em sistemas grandes ou legados. Ao usá-lo, observamos que as equipes valorizam mais o acesso rápido a informações contextuais do que a geração de código e histórias de usuário; para tais cenários, especialmente assistentes de programação, agentes de engenharia de software são mais adequados.

32. Pesos & Vieses

Experimente

Pesos & Vieses continuou a evoluir, adicionando mais recursos focados em modelos de linguagem de grande porte desde que foi apresentado pela última vez no Radar. Eles estão expandindo Traces e introduzindo Weave, uma plataforma completa que vai além do rastreamento de sistemas de agentes baseados em modelos de linguagem de grande porte (LLMs). O Weave permite que você crie avaliações de sistema, defina métricas personalizadas, use modelos de linguagem de grande porte como juízes para tarefas como sumarização e salve conjuntos de dados que capturam comportamentos diferentes para análise. Isso ajuda a otimizar os componentes do modelo de linguagem de grande porte e rastrear o desempenho em níveis local e global. A plataforma também facilita o desenvolvimento iterativo e a depuração eficaz de sistemas de agentes, onde os erros podem ser difíceis de detectar. Além disso, permite a coleta de feedback humano valioso, que pode ser usado posteriormente para ajustar modelos.

33. Arize Phoenix

Avalie

Com a popularidade dos modelos de linguagem de grande porte (LLMs) e dos agentes autônomos, a observabilidade de LLMs está se tornando cada vez mais importante. Anteriormente, recomendamos plataformas como o Langfuse e Weights & Biases (W&B). A Arize Phoenix é outra plataforma emergente nesse espaço, e nossa equipe teve uma experiência positiva ao utilizá-la. Ela oferece recursos padrões, como rastreamento de LLMs, avaliação e gerenciamento de prompts, com integração fluída com os principais provedores e frameworks de LLMs. Isso facilita a obtenção de insights sobre as respostas dos modelos, latência e uso de tokens com configuração mínima. Até agora, nossa experiência se limita à ferramenta de código aberto, mas a plataforma Arize mais ampla oferece recursos ainda mais abrangentes. Estamos ansiosas para explorá-la no futuro.

34. Chainloop

Avalie

Chainloop é uma plataforma de segurança de cadeia de suprimentos de código aberto que ajuda equipes de segurança a impor conformidade, enquanto permite que equipes de desenvolvimento integrem a conformidade de segurança de forma tranquila nos pipelines CI/CD. Ela consiste em um plano de controle, que atua como fonte única de verdade para políticas de segurança, e uma CLI, que executa atestados nos fluxos de trabalho de CI/CD para garantir a conformidade. Os times de segurança definem contratos de fluxo de trabalho especificando quais artefatos — como os SBOMs e relatórios de vulnerabilidade — devem ser coletados, onde guardá-los e como avaliar a conformidade. A Chainloop usa as linguagens de política Rego, OPA's para validar atestados — por exemplo, garantir que um CycloneDX SBOM atenda aos requerimentos de versão. Durante a execução do fluxo de trabalho, artefatos de segurança como SBOMs são anexados a um atestado e enviados ao plano de controle para aplicação e auditoria. Essa abordagem garante que a conformidade possa ser aplicada de forma consistente e em escala, minimizando o atrito nos fluxos de trabalho de desenvolvimento. Isso resulta em uma fonte única de verdade em conformidade com o nível três do SLSA para metadados, artefatos e atestados.

35. Deepseek R1

Avalie

DeepSeek-R1 é a primeira geração de modelos de raciocíno do DeepSeek. Através de uma progressão de modelos não baseados em raciocínio, as engenheiras da DeepSeek projetaram e utilizaram métodos para maximizar a utilização do hardware. Isso inclui Multi-Head Latent Attention (MLA), Mixture of Experts (MoE) gating, treinamento de pontos flutuantes de 8 bits (FP8) e programação PTX de baixo nível. Sua abordagem de co-design de computação de alto desempenho permite que o DeepSeek-R1 rivalize com modelos de última geração a um custo significativamente reduzido para treinamento e inferência.

DeepSeek-R1-Zero é notável por outra inovação: as engenheiras conseguiram extrair capacidades de raciocínio de um modelo não baseado em raciocínio utilizando simples aprendizado por reforço, sem a necessidade de ajuste fino supervisionado. Todos os modelos DeepSeek são open-weight, o que significa que estão disponíveis gratuitamente, embora o código de treinamento e os dados permaneçam proprietários. O repositório inclui seis modelos densos destilados do DeepSeek-R1, baseados no Llama e Qwen, sendo que o DeepSeek-R1-Distill-Qwen-32B supera o OpenAI-o1-mini em vários benchmarks.

36. Deno

Avalie

Criado por Ryan Dahl, o inventor do Node.js, o Deno foi desenvolvido para corrigir o que ele via como erros no Node.js. Ele possui um sistema de sandbox mais rigoroso, gerenciamento embutido de dependências e suporte nativo ao TypeScript — um grande atrativo para sua base de usuárias. Muitos de nós preferimos a utilização do Deno em projetos TypeScript por sentirmos que ele parece com um ambiente de execução e ferramenta voltada especificamente ao TypeScript, ao invés de um complemento do Node.js.

Desde sua inclusão no Radar em 2019, o Deno fez avanços significativos. O lançamento do Deno 2 trouxe compatibilidade retroativa com o Node.js, suporte a bibliotecas npm, versões com suporte de longo prazo (LTS) e outras melhorias. Anteriormente, um dos maiores obstáculos à adoção era a necessidade de reescrever aplicações originalmente feitas em Node.js. Essas atualizações reduzem

o impacto da migração e ampliam as opções de dependências para ferramentas e sistemas de apoio. Dado o enorme ecossistema do Node.js e npm, essas mudanças devem impulsionar ainda mais a adoção do Deno.

Além disso, a biblioteca padrão do Deno tornou-se estável, ajudando a combater a proliferação de pacotes npm de pouco valor no ecossistema. Sua biblioteca padrão e ferramentas tornam o uso do TypeScript ou JavaScript mais atraente para desenvolvimento no lado do servidor. Porém, recomendamos cautela ao optar por uma única plataforma somente para evitar a programação poliglota.

37. Graphiti

Avalie

Graphiti cria grafos de conhecimento dinâmicos e temporais, capturando fatos e relacionamentos em constante evolução. Nossas equipes utilizam GraphRAG para descobrir conexões entre dados, o que melhora a precisão na recuperação e nas respostas. Como os conjuntos de dados estão sempre mudando, o Graphiti mantém metadados temporais nas arestas do grafo para registrar o ciclo de vida dos relacionamentos. Ele processa dados estruturados e não estruturados como episódios distintos e suporta consultas combinando algoritmos baseados em tempo, busca textual, semântica e grafos. Para aplicações baseadas em modelos de linguagem de grande porte (LLMs) — seja RAG ou agentes autônomos — o Graphiti permite retenção de longo prazo e raciocínio baseado em estado.

38. Helicone

Avalie

Semelhante ao Langfuse, Weights & Biases e ao Arize Phoenix, Helicone é uma plataforma gerenciada de LLMOps projetada para atender à crescente demanda empresarial por gerenciamento de custos de modelos de linguagem de grande porte (LLM), avaliação de ROI e mitigação de riscos. Com foco nas desenvolvedoras e de código aberto, o Helicone suporta aplicações de IA prontas para produção, oferecendo experimentação com prompts, monitoramento, depuração e otimização ao longo de todo o ciclo de vida do LLM. Ele permite a análise em tempo real de custos, utilização, desempenho e rastreamentos de pilha agentes em diversos provedores de LLM. Embora simplifique o gerenciamento de operações de LLM, a plataforma ainda está em desenvolvimento e pode exigir algum conhecimento especializado para aproveitar completamente seus recursos avançados. Nossa equipe tem utilizado a plataforma com boas experiências até agora.

39. Humanloop

Avalie

Humanloop é uma plataforma emergente focada em tornar sistemas de IA mais confiáveis, adaptáveis e alinhados com as necessidades das usuárias, integrando feedback humano nos principais pontos de decisão. Ela oferece ferramentas para rotulagem humana, aprendizado ativo e ajuste fino com a estratégia “human in the loop”, bem como avaliação de um modelo de linguagem de grande porte (LLM) em relação aos requisitos de negócio. Além disso, ela ajuda a gerenciar o ciclo de vida financeiro das soluções de GenIA com maior controle e eficiência. A Humanloop suporta colaboração através de um espaço de trabalho compartilhado, gerenciamento de prompts com controle de versão e integração com CI/CD para prevenir regressões. Ela também disponibiliza funções de observabilidade como rastreamento, logging, alertas e proteções para monitorar e otimizar o desempenho da IA. Essas capacidades a tornam relevante para organizações que estão implementando IA em domínios regulados ou de alto risco em que a supervisão humana é essencial. Com seu foco em práticas de IA responsáveis, vale a pena avaliar a Humanloop para equipes que buscam construir sistemas de IA escaláveis e éticos.

40. Model Context Protocol (MCP)

Avalie

Um dos maiores desafios em fazer prompts é garantir que a ferramenta de IA tenha acesso a todo o contexto relevante para a tarefa. Muitas vezes, esse contexto já existe nos sistemas que usamos todos os dias: wikis, rastreadores de problemas, bancos de dados ou sistemas de observabilidade. A integração perfeita entre as ferramentas de IA e essas fontes de informação pode melhorar significativamente a qualidade dos resultados gerados pela IA.

O Model Context Protocol (MCP), um padrão aberto lançado pela Anthropic, fornece um framework padronizado para a integração de aplicações LLM com fontes de dados e ferramentas externas. Ele define servidores e clientes MCP, em que os servidores acessam as fontes de dados e os clientes integram e usam esses dados para aprimorar os prompts. Muitos assistentes de código já implementaram integração com o MCP, o que lhes permite atuar como clientes do MCP.

Os servidores MCP podem ser executados de duas maneiras: localmente, como um processo Python ou Node em execução no computador da usuárias, ou remotamente, como um servidor ao qual a cliente MCP se conecta via SSE (embora não tenhamos visto ainda nenhum uso da variante de servidor remoto). Atualmente, o MCP é usado principalmente da primeira forma, com desenvolvedoras clonando implementações de servidores MCP de código aberto. Embora os servidores executados localmente ofereçam uma maneira simples de evitar dependências de terceiros, eles permanecem menos acessíveis a usuárias não técnicas e apresentam desafios como governança e gerenciamento de atualizações. Dito isso, é fácil imaginar como esse padrão poderia evoluir para um ecossistema mais maduro e amigável no futuro.

41. Open WebUI

Avalie

Open WebUI é uma plataforma de IA de código aberto e auto-hospedada, com um conjunto versátil de recursos. Ele suporta APIs compatíveis com OpenAI e integra-se a provedores como OpenRouter e GroqCloud, entre outros. Pode rodar totalmente offline ao se conectar a modelos locais ou auto-hospedados via Ollama. Open WebUI inclui suporte nativo para RAG, permitindo interações com documentos locais e da web por meio de uma experiência conversacional. Ele oferece controles RBAC granulares, possibilitando definir quais modelos e recursos da plataforma estarão disponíveis para diferentes grupos de usuários. A plataforma é extensível por meio do Functions — blocos de construção em Python que personalizam e expandem suas funcionalidades. Outro recurso importante é a avaliação de modelos, que inclui um ambiente de comparação lado a lado para testar diferentes modelos de linguagem de grande porte (LLMs) em tarefas específicas. Open WebUI pode ser implantado em diferentes escalas — desde um assistente de IA pessoal até um assistente de colaboração para equipes ou uma plataforma de IA em nível empresarial.

42. pg_mooncake

Avalie

pg_mooncake é uma extensão do PostgreSQL que adiciona armazenamento em colunas e execução vetorizada. Tabelas de armazenamento de colunas são armazenadas como tabelas Iceberg ou Delta Lake no sistema de arquivos local ou armazenamento em nuvem compatível com S3. pg_mooncake suporta carregamento de dados a partir de formatos de arquivo como Parquet, CSV e até mesmo os conjuntos de dados do Hugging Face. Pode ser uma boa opção para análises pesadas de dados que normalmente exigem armazenamento em colunas, pois elimina a necessidade de adicionar tecnologias dedicadas de armazenamento de colunas à sua stack.

43. Modelos de raciocínio

Avalie

Um dos avanços mais significativos da IA desde o último Radar foi o surgimento e proliferação dos modelos de raciocínio. Também comercializados como modelos de pensamento, esses modelos alcançaram desempenho de alto nível humano em benchmarks como matemática avançada e programação.

Modelos de raciocínio são geralmente treinados por meio de aprendizado por reforço ou ajuste fino supervisionado, aprimorando capacidades como pensamento passo a passo (CoT), exploração de alternativas (ToT) e autocorreção. Exemplos incluem o1/o3 da OpenAI, DeepSeek R1 e Gemini 2.0 Flash Thinking. No entanto, esses modelos devem ser vistos como uma categoria distinta de modelos de linguagem de grande porte (LLMs), em vez de simplesmente versões mais avançadas.

Essa capacidade aumentada tem um custo. Modelos de raciocínio exigem maior tempo de resposta e maior consumo de tokens, levando-nos a chamá-los jocosamente de IA mais lenta (como se a IA atual já não fosse lenta o suficiente). Nem todas as tarefas justificam essa troca. Para tarefas mais simples, como sumarização de texto, geração de conteúdo ou chatbots de resposta rápida, LLMs de uso geral continuam sendo a melhor escolha. Aconselhamos o uso de modelos de raciocínio em áreas STEM (ciência, tecnologia, engenharia e matemática), resolução de problemas complexos e tomada de decisões — por exemplo, ao usar LLMs como juízes ou melhorar a explicabilidade por meio de saídas CoT explícitas. No momento em que escrevemos este texto, o Claude 3.7 Sonnet, um modelo de raciocínio híbrido, havia acabado de ser lançado, sugerindo uma possível fusão entre LLMs tradicionais e modelos de raciocínio.

44. Restate

Avalie

Restate é uma plataforma de execução durável, similar ao Temporal, desenvolvida pelas criadoras originais do Apache Flink. Em termos de funcionalidades, oferece workflows como código, processamento de eventos com estado, o padrão saga e máquinas de estado duráveis. Escrito em Rust e implantado como um único binário, ele usa um log distribuído para registrar eventos, implementado com um algoritmo de consenso virtual baseado no Flexible Paxos; isso garante durabilidade em caso de falha de nó. Há kits de desenvolvimento de software (SDKs) disponíveis para as linguagens mais comuns: Java, Go, Rust e TypeScript. Ainda recomendamos evitar transações distribuídas em sistemas distribuídos, devido à complexidade adicional e ao inevitável aumento da sobrecarga operacional. No entanto, se você não puder evitá-las no seu ambiente, vale a pena avaliar esta plataforma.

45. Supabase

Avalie

O Supabase é uma alternativa de código aberto ao Firebase para a criação de backends escaláveis e seguros. Ele oferece um conjunto de serviços integrados, incluindo um banco de dados PostgreSQL, autenticação, APIs instantâneas, Edge Functions, assinaturas em tempo real, armazenamento e embeddings vetoriais. O objetivo do Supabase é simplificar o desenvolvimento de backend, permitindo que as desenvolvedoras se concentrem na criação de experiências de frontend, aproveitando o poder e a flexibilidade das tecnologias de código aberto. Ao contrário do Firebase, o Supabase foi desenvolvido com base no PostgreSQL. Se estiver trabalhando em protótipos ou em um MVP, vale a pena considerá-lo, pois será mais fácil migrar para outra solução SQL após o estágio de prototipagem.

46. Synthesized

Avalie

Um desafio comum no desenvolvimento de software é gerar dados de teste para ambientes de desenvolvimento e de teste. Idealmente, dados de teste devem se assemelhar o máximo possível aos dados de produção, garantindo, ao mesmo tempo, que nenhuma informação pessoalmente identificável ou sensível seja exposta. Embora isso possa parecer simples, a geração de dados de teste está longe de ser trivial. É por isso que estamos interessadas no Synthesized — uma plataforma capaz de mascarar e criar subconjuntos de dados de produção existentes ou gerar dados sintéticos estatisticamente relevantes. Ele se integra diretamente aos pipelines de build e oferece mascaramento de privacidade, fornecendo anonimização por atributo por meio de técnicas irreversíveis de ofuscação de dados, como hashing, randomização e binning. Além disso, o Synthesized pode gerar grandes volumes de dados sintéticos para testes de performance. Apesar de incluir os recursos obrigatórios de GenAI, sua funcionalidade principal aborda um desafio real e persistente para as equipes de desenvolvimento, o que faz da plataforma algo que vale a pena explorar.

47. Tonic.ai

Avalie

Tonic.ai faz parte de uma tendência crescente de plataformas projetadas para gerar dados sintéticos realistas e desidentificados para ambientes de desenvolvimento, testes e QA. Semelhante ao Synthesized, o Tonic.ai é uma plataforma com uma ampla suíte de ferramentas que endereça diversas necessidades de síntese de dados, em contraste com a abordagem mais focada em bibliotecas do Syntethic Data Vault. O Tonic.ai gera dados estruturados e não estruturados, mantendo as propriedades estatísticas de produção de dados enquanto garante privacidade e conformidade por meio de técnicas de privacidade diferenciadas. Seus principais recursos incluem detecção, classificação e remoção automática de informações sensíveis em dados não estruturados, além do provisionamento sob demanda de banco de dados através do Tonic Ephemeral. Ela também oferece o Tonic Textual, um data lakehouse seguro que auxilia desenvolvedoras de IA a aproveitar os dados não estruturados para sistemas de geração aumentada por recuperação (RAG) e ajuste fino de modelos de linguagem de grande porte (LLMs). Equipes que querem acelerar velocidade da engenharia enquanto geram dados realistas e escaláveis — cumprindo requisitos rigorosos de privacidade de dados — deveriam avaliar o Tonic.ai.

48. turbopuffer

Avalie

O turbopuffer é um mecanismo de busca multi-tenant e sem servidor que integra perfeitamente buscas de vetor e de texto em armazenamento de objetos. Nós gostamos de sua arquitetura e escolhas de design, especialmente por seu foco em durabilidade, escalabilidade e eficiência de custo. Ao usar o armazenamento de objetos como um log de escrita antecipada (write-ahead log) enquanto mantém seus nós de consulta sem estado, ele é adequado para cargas de trabalho de busca em larga escala.

Projetado para performance e precisão, o turbopuffer entrega alto recall nativamente, até mesmo para consultas complexas baseadas em filtros. Ele armazena em cache os resultados de consultas frias em SSDs NVMe e mantém os namespaces frequentemente acessados em memória, permitindo buscas de baixa latência em bilhões de documentos. Isso o torna ideal para recuperação de documentos em grande escala, busca vetorial e geração aumentada por recuperação (RAG) em aplicações de IA. Entretanto, sua dependência por armazenamento de objetos introduz compensações na latência das consultas, fazendo com que ele seja mais eficaz para cargas de trabalho que se beneficiam de computação distribuída sem estado.

O turbopuffer alimenta sistemas de produção de grande escala como o Cursor mas atualmente está disponível apenas por indicação ou convite.

49. VectorChord

Avalie

VectorChord é uma extensão do PostgreSQL para pesquisa de similaridade vetorial, desenvolvida pelas criadoras do pgvecto.rs como sua sucessora. É de código aberto, compatível com tipos de dados do pgvector e projetada para pesquisa vetorial de alto desempenho e com eficiência de disco. Ela emprega Inverted File Index (IVF) junto com quantização RaBitQ para permitir uma busca vetorial rápida, escalável e precisa, ao mesmo tempo que reduz significativamente as demandas de computação. Como outras extensões do PostgresSQL neste espaço, ela aproveita o ecossistema do PostgreSQL, permitindo a busca de vetores junto com operações transacionais padrão. Embora ainda esteja em seus estágios iniciais, vale a pena avaliar o VectorChord para cargas de trabalho de busca de vetores.

50. Tyk hybrid API management

Evite

Observamos que várias equipes têm enfrentado problemas com a solução Tyk hybrid API management. Embora o conceito de um plano de controle gerenciado e planos de dados autogerenciados ofereça flexibilidade para infraestruturas complexas (como multi-cloud e híbrida), equipes relataram incidentes no plano de controle que foram descobertos internamente, e não pela Tyk, evidenciando possíveis lacunas de observabilidade no ambiente Tyk hospedado na AWS. Além disso, o suporte para incidentes parece ser lento; a comunicação via tickets e e-mails não é ideal nessas situações. As equipes também apontaram questões relacionadas à maturidade da documentação da Tyk, frequentemente considerada insuficiente para cenários e problemas mais complexos. Adicionalmente, outros produtos do ecossistema Tyk aparentam imaturidade — por exemplo, foi relatado que o portal para desenvolvedoras na versão empresarial não é retrocompatível e possui capacidades limitadas de customização. Especialmente para a configuração híbrida da Tyk, recomendamos cautela e continuaremos acompanhando sua evolução.

Ferramentas

Adote

51. Renovate
52. uv
53. Vite

Experimente

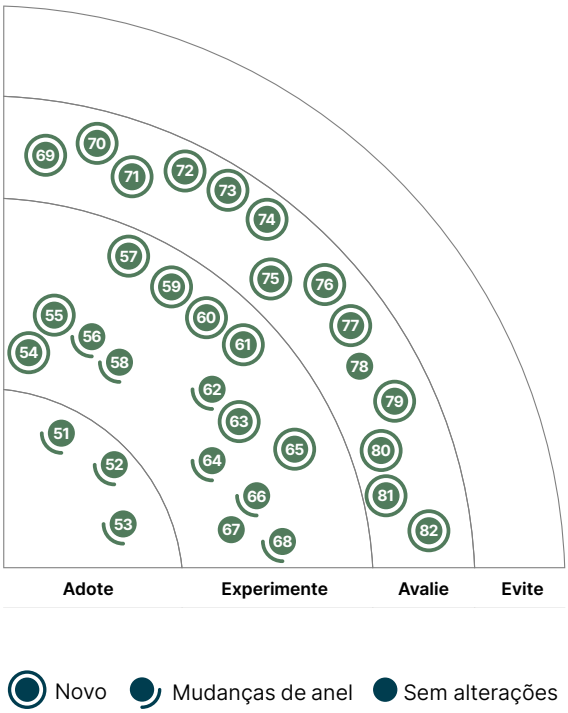
54. Claude Sonnet
55. Cline
56. Cursor
57. D2
58. Databricks Delta Live Tables
59. JSON Crack
60. MailSlurp
61. Metabase
62. NeMo Guardrails
63. Nyx
64. OpenRewrite
65. Plerion
66. Agentes de engenharia de software
67. Tuple
68. Turborepo

Avalie

69. AnythingLLM
70. Gemma Scope
71. Hurl
72. Jujutsu
73. kubenetmon
74. Mergiraf
75. ModernBERT
76. OpenRouter
77. Redactive
78. System Initiative
79. TabPFN
80. v0
81. Windsurf
82. YOLO

Evite

—



51. Renovate

Adote

Renovate tem se tornado a ferramenta preferida por muitos de nossos times que buscam uma abordagem proativa para gerenciar versões de dependências. Embora o Dependabot continue sendo uma escolha segura e padrão para repositórios hospedados no GitHub, recomendamos avaliar o Renovate como uma solução mais abrangente e personalizável. Para aproveitar ao máximo os benefícios do Renovate, configure-o para monitorar e atualizar todas as dependências, incluindo ferramentas, infraestrutura e dependências privadas ou hospedadas internamente. Para reduzir a carga operacional das desenvolvedoras, considere adotar o merge automático de PRs com atualizações de dependências.

52. uv

Adote

Desde o último Radar, ganhamos mais experiência com uv, e o feedback dos nossos times tem sido extremamente positivo. O uv é uma ferramenta de gerenciamento de pacotes e projetos Python de última geração, escrita em Rust, com uma proposta de valor principal: é extremamente rápido. Ele supera outros gerenciadores de pacotes Python por uma grande margem em benchmarks, acelerando os ciclos de compilação e teste e melhorando significativamente a experiência das desenvolvedoras. Além do desempenho, o uv oferece um conjunto de ferramentas unificado, substituindo efetivamente ferramentas como Poetry, pyenv e pipx. No entanto, nossas preocupações com ferramentas de gerenciamento de pacotes permanecem: um ecossistema forte, uma comunidade madura e suporte de longo prazo são fundamentais. Como o uv ainda é relativamente novo, movê-lo para o anel de Adoção é uma decisão ousada. No entanto, muitas equipes de dados estão ansiosas para deixar de usar o sistema legado de gerenciamento de pacotes do Python, e nossas desenvolvedoras de linha de frente recomendam consistentemente o uv como a melhor ferramenta disponível hoje.

53. Vite

Adote

Desde que o Vite foi mencionado no Radar, ele vem ganhando ainda mais atenção. Ele é uma ferramenta de alta performance de frontend com carregamento rápido (hot-reloading). Tem sido adotada e recomendada como a escolha padrão em muitos frameworks frontend, incluindo Vue, SvelteKit e React, que recentemente descontinuou o create-react-app. O Vite também recebeu recentemente um investimento significativo, o que levou à fundação da VoidZero, uma organização dedicada ao desenvolvimento do Vite. Esse investimento deve acelerar o desenvolvimento e melhorar a sustentabilidade do projeto a longo prazo.

54. Claude Sonnet

Experimente

Claude Sonnet é um modelo de linguagem avançada que se destaca em codificação, escrita, análise e processamento visual. Está disponível para navegador, terminal, na maior parte das principais IDEs e pode ser integrado ao GitHub Copilot. Até o momento, os benchmarks mostram que ele supera os modelos anteriores, como as versões 3.5 e 3.7, incluindo os modelos anteriores do Claude. Também é capaz de interpretar gráficos e extrair texto de imagens, e oferece uma experiência voltada para desenvolvedoras, como a funcionalidade “Artefatos” na interface do navegador, para gerar e interagir com conteúdo dinâmico, como trechos de código e designs em HTML.

Usamos a versão 3.5 do Claude Sonnet extensivamente no desenvolvimento de software e constatamos que ela aumenta significativamente a produtividade em diversos projetos. Ela se destaca em projetos do tipo “greenfield”, especialmente em design colaborativo de software e discussões arquitetônicas. Embora seja cedo demais para chamar qualquer modelo de IA de estável para assistência em programação, o Claude Sonnet está entre os modelos mais confiáveis com os quais trabalhamos. Durante a escrita, a versão Claude 3.7 também foi lançada e é promissora, embora ainda não tenhamos testado totalmente em produção.

55. Cline

Experimente

Cline é uma extensão de código aberto para o VSCode e atualmente um dos principais nomes na categoria de agentes para engenharia de software supervisionados. Ele permite que desenvolvedoras conduzam sua implementação diretamente pelo chat do Cline, integrando-se perfeitamente com a IDE que já utilizam. Recursos importantes como o modo Plan & Act, transparência no uso de tokens e integração com MCP ajudam as desenvolvedoras a interagir eficientemente com modelos de linguagem de grande porte (LLMs). O Cline demonstrou capacidades avançadas na execução de tarefas complexas de desenvolvimento, especialmente utilizando o modelo Claude 3.5 Sonnet. Ele oferece suporte a grandes bases de código, automatiza testes com navegadores sem interface gráfica (headless) e corrige bugs proativamente.

Ao contrário de soluções baseadas em nuvem, o Cline reforça a privacidade ao armazenar dados localmente. Sua natureza de código aberto não apenas garante maior transparência como também possibilita melhorias contínuas pela comunidade. Contudo, desenvolvedoras devem estar atentas ao custo relacionado ao uso de tokens, já que a orquestração de contexto de código do Cline, embora muito eficaz, pode consumir muitos recursos. Outro possível ponto de atenção é o limite de requisições, que pode desacelerar os fluxos de trabalho. Até que essa questão seja resolvida, recomenda-se utilizar provedores de API como o OpenRouter, que oferecem limites mais favoráveis.

56. Cursor

Experimente

Continuamos impressionados com o editor de código baseado em IA Cursor, que permanece líder no competitivo espaço de assistência à programação com IA. Sua orquestração de contexto de código é muito eficaz e oferece suporte a uma ampla variedade de modelos, incluindo a opção de usar uma chave de API personalizada. A equipe do Cursor frequentemente apresenta recursos inovadores de experiência de usuário antes de outras fornecedoras, e eles incluem uma extensa lista de provedores de contexto em seu chat, como a referência de git diffs, conversas anteriores de IA, pesquisa na web, biblioteca de documentação e integração MCP. Juntamente com ferramentas como Cline e Windsurf, o Cursor também se destaca por seu forte modo de codificação agêntico. Este modo permite que as desenvolvedoras guiem sua implementação diretamente de uma interface de chat com IA, com a ferramenta lendo e modificando arquivos de forma autônoma, bem como executando comandos. Além disso, apreciamos a capacidade do Cursor de detectar erros de compilação no código gerado e corrigi-los proativamente.

57. D2

Experimente

D2 é uma ferramenta de código aberto de diagramas como código que ajuda a criar e customizar diagramas a partir de textos. Ela introduz a linguagem de script para diagramas D2, que prioriza a legibilidade em vez da compactação, com uma sintaxe simples e declarativa. D2 vem com um tema

padrão e aproveita o mesmo layout do Mermaid. Nossas equipes apreciam sua sintaxe leve, que foi especificamente projetada para documentação de software e diagramas de arquitetura.

58. Databricks Delta Live Tables

Experimente

Delta Live Tables (DLT) continuam a demonstrar seu valor na simplificação e otimização da gestão de pipelines de dados, oferecendo suporte tanto para processamento em tempo real quanto para processamento em lote por meio de uma abordagem declarativa. Ao automatizar tarefas complexas de engenharia de dados, como o gerenciamento manual de checkpoints, a DLT reduz a sobrecarga operacional, garantindo um sistema robusto de ponta a ponta. Sua capacidade de orquestrar pipelines simples com mínima intervenção manual aumenta a confiabilidade e a flexibilidade, enquanto recursos como visualizações materializadas fornecem atualizações incrementais e otimização de desempenho para casos de uso específicos.

No entanto, as equipes devem compreender as nuances da DLT para aproveitar totalmente seus benefícios e evitar possíveis armadilhas. Sendo uma abstração opinativa, a DLT gerencia suas próprias tabelas e restringe a inserção de dados a um único pipeline por vez. As tabelas de streaming aceitam apenas inserções (append-only), exigindo considerações cuidadosas de design. Além disso, excluir um pipeline DLT também exclui automaticamente a tabela e os dados subjacentes, potencialmente gerando problemas na operação.

59. JSON Crack

Experimente

JSON Crack é uma extensão do Visual Studio Code que cria gráficos interativos a partir de dados em texto. Apesar do nome, ela suporta vários formatos, como YAML, TOML e XML. Diferente do Mermaid e do D2, onde o texto é usado para criar um gráfico visual específico, o JSON Crack serve para explorar dados que estão em formato de texto. O algoritmo de layout funciona bem e a ferramenta ainda permite esconder ramificações e nós seletivamente, o que é ótimo para explorar grandes conjuntos de dados. Também existe uma versão online da ferramenta, mas vale a pena lembrar das nossas preocupações com o uso de serviços online para formatação ou análise de código. O JSON Crack tem um limite de nós e, caso precise lidar com arquivos maiores, ele recomenda uma ferramenta comercial para isso.

60. MailSlurp

Experimente

Testar fluxos de trabalho que envolvem e-mails pode ser um processo complexo e demorado. As equipes de desenvolvimento precisam criar clientes de API de e-mail personalizados para automação, além de configurar as caixas de entrada temporárias para cenários de teste manual, como testes de usuário ou treinamentos internos do produto antes de grandes lançamentos. Esses desafios se tornam ainda mais evidentes ao desenvolver produtos de onboarding de clientes. Tivemos uma experiência positiva com o MailSlurp, um serviço de servidor de e-mail e API de SMS. Ele oferece APIs REST para criar caixas de entrada e números de telefone, além de validar e-mails e mensagens diretamente no código. Seu painel sem necessidade de programação também é útil para a preparação de testes manuais. Recursos adicionais, como domínios personalizados, webhooks, respostas automáticas e encaminhamento de e-mails, são ótimos para cenários mais avançados.

61. Metabase

Experimente

Metabase é uma ferramenta de código aberto para análise e inteligência de negócios que permite visualizar e analisar dados de diversas fontes, incluindo bancos de dados relacionais e NoSQL. A ferramenta ajuda usuárias a criarem visualizações e relatórios, organizá-los em painéis e compartilhar insights facilmente. Ela também oferece um kit de desenvolvimento de software (SDK) para incorporar painéis interativos em aplicações web, adaptando-se ao tema e estilo da aplicação, o que a torna amigável para desenvolvedoras. Com conectores de dados oficialmente suportados e mantidos pela comunidade, a Metabase se mostra versátil dentre os ambientes de dados. Como uma ferramenta de BI leve, nossas equipes a consideram útil para gerenciar painéis interativos e relatórios em suas aplicações.

62. NeMo Guardrails

Experimente

NeMo Guardrails é um kit de ferramentas de código aberto da NVIDIA, considerado fácil de usar, que capacita desenvolvedoras a implementar restrições para modelos de linguagem de grande porte (LLMs) usados em aplicações conversacionais. Desde a última vez que o mencionamos no Radar, o NeMo tem sido amplamente adotado por nossas equipes e continua a evoluir. Muitas das melhorias mais recentes do NeMo Guardrails focam na expansão das integrações e no fortalecimento da segurança, dados e controle, alinhando-se ao objetivo central do projeto.

Uma grande atualização na documentação do NeMo melhorou sua usabilidade, e novas integrações foram adicionadas, incluindo AutoAlign e Patronus Lynx, juntamente com suporte ao Colang 2.0. As principais atualizações incluem melhorias na segurança e proteção de conteúdo, bem como um lançamento recente que permite o streaming de conteúdo de LLMs através de trilhos de saída (output rails) para melhor desempenho. Também vimos suporte adicional para Segurança de Prompt. Além disso, a Nvidia lançou três novos microsserviços: microsserviço NIM de segurança de conteúdo, microsserviço NIM de controle de tópicos e detecção de jailbreak, todos integrados ao NeMo Guardrails.

Com base no conjunto crescente de funcionalidades e no aumento do uso em produção, estamos movendo o NeMo Guardrails para a fase de Teste. Recomendamos revisar as últimas notas de lançamento para obter uma visão completa das mudanças desde o nosso último blip.

63. Nyx

Experimente

Nyx é uma ferramenta versátil de release semântico que suporta uma ampla variedade de projetos de engenharia de software. Ela é agnóstica à linguagem e funciona com todas as principais plataformas de CI e SCM, tornando-se altamente adaptável. Embora muitas equipes usem versionamento semântico no desenvolvimento baseado em tronco, Nyx também suporta fluxos de trabalho como o Gitflow, OneFlow e GitHub Flow. Uma das principais vantagens do Nyx em produção é a geração automática de registros de alterações, com suporte a commits convencionais. Como observado nas edições anteriores do Radar, alertamos contra padrões de desenvolvimento que dependem de branches de longa duração (por exemplo, Gitflow, GitOps), pois introduzem desafios que nem mesmo ferramentas poderosas como o Nyx conseguem mitigar. Recomendamos fortemente testar o Nyx em fluxos de CI/CD, especialmente para desenvolvimento baseado em tronco, onde temos observado muitos sucessos.

64. OpenRewrite

Experimente

O OpenRewrite continua sendo uma ferramenta eficaz para refatorações em larga escala que seguem um conjunto de regras específicas, tais como migrações para uma nova versão da API de uma biblioteca amplamente utilizada ou aplicação de atualizações em múltiplos serviços criados a partir do mesmo template. Recentemente, a ferramenta passou a oferecer suporte a outras linguagens além do Java, especialmente ao JavaScript. Com ciclos curtos de versões incluindo suporte de longo prazo (LTS) em frameworks como Angular, manter os projetos atualizados com as versões mais recentes tornou-se cada vez mais importante. O OpenRewrite auxilia esse processo de maneira eficaz. O uso de assistentes de programação com IA pode ser uma alternativa, porém, para alterações baseadas em regras, geralmente é mais lento, mais caro e menos confiável. Gostamos que o OpenRewrite já venha acompanhado de um catálogo de recipes (regras), que descrevem exatamente as mudanças a serem feitas. O motor de refatoração, as regras pré-definidas e os plugins para ferramentas de build são todos de código aberto, tornando mais fácil para as equipes adotarem o OpenRewrite quando necessário.

65. Plerion

Experimente

Plerion é uma plataforma de segurança em nuvem focada em AWS, que se integra às provedoras de hospedagem para identificar riscos, configurações incorretas e vulnerabilidades em sua infraestrutura de nuvem, servidores e aplicativos. Semelhante ao Wiz, o Plerion utiliza priorização baseada em riscos para os problemas detectados, permitindo que você foque nos 1% que importam. Nossas equipes relataram experiências positivas com o Plerion, destacando que ele forneceu a clientes insights significativos e reforçou a importância da monitoração proativa de segurança para suas organizações.

66. Agentes de engenharia de software

Experimente

Desde a última vez que escrevemos sobre agentes de engenharia de software , há seis meses, a indústria ainda não chegou a uma definição consensual do termo “agente”. Porém, um avanço significativo surgiu — não em agentes de programação autônomos (que continuam pouco convincentes), mas em agentes de modo supervisionado presentes na IDE. Esses modos permitem que desenvolvedoras conduzam implementações via chat, com ferramentas que não apenas modificam códigos em múltiplas linhas e arquivos, mas também executam comandos, testes e respondem aos feedbacks da IDE, como erros de linting ou de compilação.

Esta abordagem, às vezes chamada de “chat-oriented programming” (CHOP) ou “prompt-to-code”, mantém as desenvolvedoras no controle enquanto transfere mais responsabilidade para a IA do que assistentes de programação tradicionais, como sugestões automáticas. As ferramentas que lideram esse espaço incluem Cursor, Cline e Windsurf, com GitHub Copilot levemente atrasado, porém conquistando espaço rapidamente. A utilidade desses agentes dependem tanto do modelo usado (com a série Sonnet do Claude sendo o estado da arte atual) quanto a da qualidade de integração com a IDE para proporcionar uma boa experiência à desenvolvedora.

Nós achamos esses fluxos de trabalho interessantes e promissores, pois trazem um aumento notável na velocidade de codificação. Porém, manter um escopo pequeno de problemas ajuda as desenvolvedoras a revisarem melhor as mudanças feitas por IA. Esses fluxos funcionam melhor com prompts de baixa abstração e bases de códigos compatíveis com IA que sejam bem estruturadas

e devidamente testadas. À medida em que esses métodos melhoram, também aumenta o risco de complacência com códigos gerados por IA. Para mitigar esse problema, aplique programação em par (pair programming) e outras práticas efetivas de revisão, especialmente para códigos em produção.

67. Tuple

Experimente

Tuple, uma ferramenta otimizada para programação em pares remota, foi originalmente projetada para preencher a lacuna deixada pelo Screenhero do Slack. Desde a última vez que a mencionamos no Radar, a ferramenta teve uma adoção mais ampla, corrigiu limitações anteriores e agora oferece suporte ao Windows.

Uma melhoria importante é o compartilhamento de tela aprimorado, com um recurso de privacidade integrado que permite ocultar janelas de aplicativos privados (como mensagens de texto) enquanto compartilha ferramentas como o navegador. Anteriormente, limitações da interface faziam o Tuple parecer mais uma ferramenta de programação em pares do que uma de colaboração geral. Com essas atualizações, as usuárias agora podem colaborar em conteúdos além do ambiente de desenvolvimento integrado (IDE).

No entanto, é importante observar que o par remoto tem acesso completo à área de trabalho. Se não for configurado corretamente, isso pode representar um risco de segurança, especialmente se o parceiro não for confiável. Recomendamos fortemente que as equipes sejam instruídas sobre as configurações de privacidade, melhores práticas e etiqueta do Tuple antes do uso.

Incentivamos as equipes a experimentarem a versão mais recente do Tuple no fluxo de trabalho de desenvolvimento. A ferramenta está alinhada com nossa recomendação de programação em pares remota pragmática, oferecendo baixa latência, uma experiência intuitiva e melhorias significativas na usabilidade.

68. Turborepo

Experimente

Turborepo ajuda a gerenciar grandes monorepos feitos em JavaScript ou TypeScript, através da análise, armazenamento em cache, paralelismo e otimização de tarefas de build para acelerar o processo. Em grandes monorepos, os projetos geralmente dependem uns dos outros; refazer o build de todas as dependências a cada alteração é ineficiente e demorado, mas o Turborepo facilita as coisas. Ao contrário do Nx, a configuração padrão do Turborepo usa vários arquivos package.json — um por projeto — o que permite ter dependências com versões diferentes (várias versões do React, por exemplo) em um único monorepo, algo que o Nx desencoraja. Embora possa ser considerado um antipadrão, isso resolve alguns casos de uso, como migrar de multi para monorepo, onde as equipes podem temporariamente precisar de várias versões de dependências. Na nossa experiência, o Turborepo é bastante simples de configurar e tem um bom desempenho.

69. AnythingLLM

Avalie

O AnythingLLM é um aplicativo de desktop de código aberto usado para interagir com grandes documentos ou trechos de conteúdo, contando com integração nativa com modelos de linguagem de grande porte (LLMs) e bancos de dados vetoriais. Ele possui uma arquitetura modular para modelos de embedding e pode ser utilizado tanto com LLMs comerciais quanto com modelos de código aberto gerenciados pelo Ollama. Além do suporte a RAG, é possível criar e organizar diferentes habilidades

na forma de agentes para executar tarefas e fluxos de trabalho personalizados. As usuárias podem organizar documentos e interações dentro de diferentes espaços de trabalho, que funcionam como sessões persistentes com contextos distintos. Recentemente, também se tornou possível implantá-lo como uma aplicação web multiusuário utilizando uma simples imagem Docker. Algumas de nossas equipes estão usando o AnythingLLM como assistente pessoal local e o consideram uma ferramenta poderosa e útil.

70. Gemma Scope

Avalie

A interpretabilidade mecanicista — compreender o funcionamento interno dos modelos de linguagem de grande porte (LLMs) — está se tornando um campo cada vez mais relevante. Ferramentas como Gemma Scope e a biblioteca de código aberto Mishax fornecem insights sobre a família de modelos abertos Gemma2. Ferramentas de interpretabilidade desempenham um papel essencial na depuração de comportamentos inesperados, identificando os componentes responsáveis por alucinações, vieses ou demais falhas, e na construção de confiança ao oferecer mais visibilidade sobre os modelos. Embora esse campo seja de interesse particular para pesquisadoras, vale destacar que com o recente lançamento do DeepSeek-R1, o treinamento de modelos tem se tornado mais viável à outras empresas além dos principais players do mercado. À medida que a IA generativa continua evoluindo, tanto a interpretabilidade quanto a segurança ganharão ainda mais importância.

71. Hurl

Avalie

O Hurl é um canivete suíço para fazer sequências de solicitações HTTP, definidas em arquivos de texto simples usando a sintaxe específica do Hurl. Além de enviar solicitações, o Hurl pode validar respostas, garantindo que uma solicitação retorne um código de status HTTP específico; verificar condições nos cabeçalhos da resposta ou no conteúdo usando XPATH, JSONPath ou expressões regulares; e extrair dados da resposta em variáveis, que podem ser usadas para encadear solicitações.

Com seu conjunto de recursos, o Hurl é útil para automações simples de API, mas também serve como uma ferramenta de teste automatizado de API. Sua capacidade de gerar relatórios de teste detalhados em HTML ou JSON aumenta sua utilidade para testar workflows. Apesar de ferramentas dedicadas como Bruno e Postman oferecerem GUIs e recursos adicionais, gostamos do Hurl por sua simplicidade. Assim como o Bruno, que também usa arquivos de texto simples, os testes do Hurl podem ser armazenados no repositório de código.

72. Jujutsu

Avalie

Git é o sistema de controle de versão distribuído (SCV) dominante, detendo a maior parte do mercado. Entretanto, apesar de mais de uma década de dominância, as desenvolvedoras ainda continuam enfrentando dificuldades com seus workflows complexos para branching, merging, rebasing e resolução de conflitos. Essa frustração contínua tem alimentado uma onda de ferramentas projetadas para aliviar essa dor — algumas oferecendo visualizações para esclarecer a complexidade, outras fornecendo suas próprias interfaces gráficas para abstraí-la completamente.

Jujutsu dá um passo adiante, oferecendo uma alternativa completa ao Git, mantendo a compatibilidade ao usar repositórios Git como backend de armazenamento. Isso permite que desenvolvedoras continuem usando os servidores e serviços Git existentes enquanto se beneficiam

dos workflows simplificados do Jujutsu. Posicionado como simples e poderoso ao mesmo tempo, Jujutsu enfatiza facilidade de uso para desenvolvedoras de todos os níveis de experiência. Um dos seus recursos de destaque é a resolução de conflitos de primeira classe, que tem potencial para melhorar significativamente a experiência da desenvolvedora.

73. kubenetmon

Avalie

Monitorar e entender o tráfego de rede associado ao Kubernetes pode ser um desafio, especialmente quando sua infraestrutura abrange várias zonas, regiões ou nuvens. kubenetmon, desenvolvido pela ClickHouse e recentemente disponibilizado como código aberto, busca resolver esse problema ao oferecer uma medição detalhada da transferência de dados do Kubernetes entre as principais provedoras de nuvem. Se você está rodando Kubernetes e tem se frustrado com custos obscuros de transferência de dados na sua fatura, pode valer a pena explorar o kubenetmon.

74. Mergiraf

Avalie

Resolver conflitos de merges é provavelmente uma das atividades menos apreciadas no desenvolvimento de software. Ainda que existam técnicas que reduzam a complexidade dos merges — por exemplo, praticando integração contínua no sentido original de realizar o merge para a branch principal compartilhada diariamente — acabamos vendo muitos esforços gastos com merges. Branches de funcionalidades de longa duração são uma das culpadas, mas o código assistido por IA também possui uma tendência em aumentar o tamanho das mudanças de código. A ajuda pode vir na forma de Mergiraf, uma nova ferramenta que resolve conflitos de merges considerando a árvore sintática ao invés de tratar código apenas como linhas de texto. Como um controlador de merges do git, pode ser configurado para que sub-comandos git como `merge` e `cherry-pick` automaticamente utilizem Mergiraf ao invés das heurísticas padrões.

75. ModernBERT

Avalie

Sendo sucessor do BERT (Bidirectional Encoder Representations from Transformers), ModernBERT é uma família de modelos transformer enconder-only de última geração projetados para uma ampla gama de tarefas de processamento de linguagem natural (PLN). Como um substituto imediato, ModernBERT melhora o desempenho e a precisão, ao mesmo tempo que aborda algumas das limitações do BERT — notavelmente incluindo suporte para comprimentos de contexto muito maiores, graças à Atenção Alternada. Equipes com necessidades de PNL devem considerar o ModernBERT antes de adotar um modelo generativo de propósito geral.

76. OpenRouter

Avalie

OpenRouter é uma API unificada para acessar vários modelos de linguagem de grande porte (LLMs). Ela fornece um único ponto de integração para provedores LLM tradicionais, simplifica a experimentação, reduz o bloqueio de fornecedores, e otimiza custos ao encaminhar solicitações para o modelo mais apropriado. Ferramentas populares como Cline e Open WebUI usam OpenRouter como seu ponto de acesso. Durante nossa discussão no Radar, questionamos se a maioria dos projetos realmente precisam alternar entre modelos, já que OpenRouter deve adicionar margem de lucro como um modelo de lucro sobre essa camada de encapsulamento. No entanto, também reconhecemos que

o OpenRouter fornece várias estratégias de balanceamento de carga para ajudar a otimizar os custos. Um recurso particularmente útil é sua capacidade de ignorar limites de taxa de API. Se sua aplicação exceder o limite de taxa de um único provedor LLM, OpenRouter pode ajudar você a superar essa limitação e obter melhor rendimento.

77. Redactive

Avalie

O Redactive é uma plataforma corporativa voltada à adoção segura de inteligência artificial, projetada para ajudar organizações reguladas a preparar dados não estruturados para aplicações de IA, como assistentes e copilotos baseados em inteligência artificial. Ela se integra a plataformas de conteúdo como o Confluence, criando índices seguros de texto para buscas com geração aumentada de recuperação (RAG). Ao fornecer apenas dados atualizados e aplicar permissões das usuárias em tempo real nos sistemas de origem, o Redactive garante que os modelos de IA tenham acesso a informações precisas e autorizadas sem comprometer a segurança. Além disso, ele oferece às equipes de engenharia ferramentas para construir casos de uso de IA com segurança utilizando qualquer modelo de linguagem de grande porte (LLM). Para organizações que estão explorando soluções impulsionadas por IA, o Redactive fornece uma abordagem simplificada para a preparação de dados e compliance, equilibrando segurança e acessibilidade para equipes que experimentam capacidades da IA em um ambiente controlado.

78. System Initiative

Avalie

Continuamos empolgados com o System Initiative. Essa ferramenta experimental representa uma direção radicalmente nova para o trabalho em DevOps. Gostamos muito da abordagem criativa por trás dessa ferramenta e esperamos que ela incentive outras iniciativas a quebrar o status quo das abordagens de infraestrutura-como-código. O System Initiative saiu da fase beta e agora está disponível gratuitamente como código aberto, sob a licença Apache 2.0. Embora suas desenvolvedoras já o utilizem para gerenciar infraestrutura em produção, ainda existe um caminho a percorrer para atender às demandas de grandes empresas. No entanto, acreditamos que vale a pena explorá-lo para conhecer uma abordagem completamente diferente das ferramentas de DevOps.

79. TabPFN

Avalie

TabPFN é um modelo baseado em transformadores projetado para classificação rápida e precisa em pequenos conjuntos de dados tabulares. Ele utiliza aprendizado em contexto (in-context learning, ou ICL) para fazer previsões diretamente a partir de exemplos rotulados, sem necessidade de ajuste de hiperparâmetros ou treinamento adicional. Pré-treinado em milhões de conjuntos de dados sintéticos, o TabPFN generaliza bem em diversas distribuições de dados e lida de forma eficaz com valores ausentes e atípicos. Seus pontos fortes incluem o processamento eficiente de dados heterogêneos e a robustez contra características não informativas.

O TabPFN é particularmente adequado para aplicações de pequeno porte, onde velocidade e precisão são cruciais. No entanto, enfrenta desafios de escalabilidade com conjuntos de dados maiores e tem limitações no tratamento de tarefas de regressão. Como uma solução inovadora, o TabPFN vale a pena ser avaliado pelo seu potencial de superar modelos tradicionais em classificação tabular, especialmente em cenários onde transformadores são menos comumente aplicados.

80. v0

Avalie

v0, da Vercel, é uma ferramenta de IA para gerar código frontend a partir de uma captura de tela, design no Figma ou um simples comando. Ela dá suporte a React, Vue, shadcn e Tailwind, entre outros frameworks de frontend. Além do código gerado por IA, o v0 oferece uma ótima experiência de usuário, incluindo a capacidade de visualizar o código gerado e implantá-lo na Vercel em um único passo. Embora a construção de aplicações para o mundo real envolva a integração de múltiplas funcionalidades além de uma única tela, o v0 oferece uma maneira sólida de prototipar e pode ser usado para iniciar o desenvolvimento de aplicações complexas.

81. Windsurf

Avalie

Windsurf é um assistente de programação de IA da Codeium que se destaca por suas capacidades "agênticas", ou autônomas. Semelhante ao Cursor e Cline, ele permite que desenvolvedoras conduzam sua implementação a partir de um chat de IA que navega e altera o código e executa comandos. Ele frequentemente lança novos recursos e integrações interessantes para o modo agêntico. Recentemente, por exemplo, lançou uma visualização de navegador que facilita o acesso do agente aos elementos DOM e ao console do navegador, e uma capacidade de pesquisa na web que permite ao Windsurf procurar documentação e soluções na internet quando apropriado. O Windsurf fornece acesso a uma variedade de modelos populares, e as usuárias podem ativar e referenciar pesquisa na web, documentação da biblioteca e integração Model Context Protocol (MCP) – um protocolo para o intercâmbio de contexto entre diferentes modelos de IA – como provedores de contexto adicionais.

82. YOLO

Avalie

A série YOLO (You Only Look Once), criada por Joseph Redmon e Ali Farhadi em 2015 durante o seu doutorado e desenvolvida pela Ultralytics, continua avançando nos modelos de visão computacional. A versão mais recente, YOLO11, oferece melhorias significativas em termos de precisão e eficiência em relação às versões anteriores. O YOLO11 pode realizar a classificação de imagens em alta velocidade com recursos mínimos, o que o torna adequado para aplicações em tempo real em dispositivos de ponta. Também descobrimos que a capacidade de usar a mesma estrutura para fazer estimativa de pose, detecção de objetos, segmentação de imagens e outras tarefas é muito poderosa. Esse desenvolvimento significativo também nos lembra que o uso de modelos "tradicionais" de aprendizado de máquina para tarefas específicas pode ser mais eficiente do que os modelos gerais de IA, como os modelos de linguagem de grande porte (LLMs).

Linguagens e Frameworks

Adote

83. OpenTelemetry
84. React Hook Form

Experimente

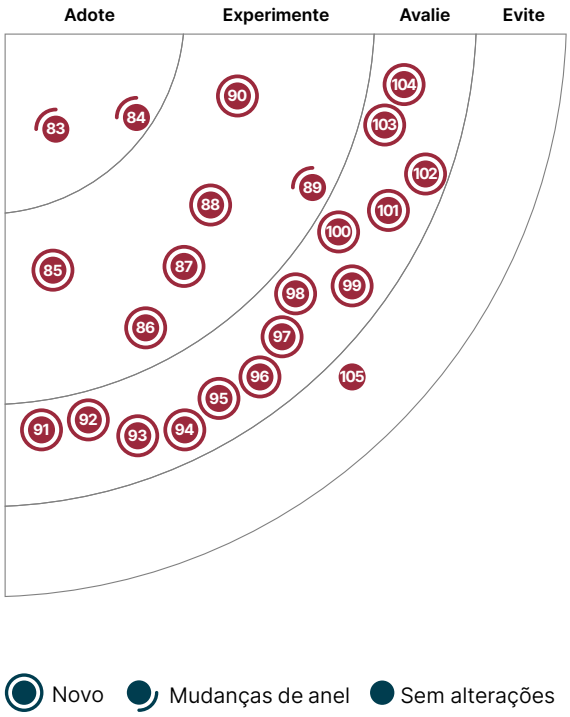
85. Effect
86. Hasura GraphQL engine
87. LangGraph
88. MarkItDown
89. Federação de módulos
90. Prisma ORM

Avalie

91. .NET Aspire
92. Android XR SDK
93. Browser Use
94. Crew AI
95. ElysiaJs
96. FastGraphRAG
97. Gleam
98. GoFr
99. Criptografia pós-quântica em Java
100. Presidio
101. PydanticAI
102. Swift para aplicações com restrições de recursos
103. Tamagui
104. torchtune

Evite

105. Uso excessivo de Node



83. OpenTelemetry

Adote

O OpenTelemetry está rapidamente se tornando o padrão da indústria para observabilidade. O lançamento da especificação do protocolo OpenTelemetry (OTLP) estabeleceu um padrão para lidar com traces, métricas e logs, reduzindo a necessidade de várias integrações ou grandes reescritas ao passo que o monitoramento de soluções distribuídas e os requisitos de interoperabilidade continuam crescendo. À medida que o OpenTelemetry se expande para suportar logs e profiling, o OTLP garante um formato de transporte consistente para todos os dados de telemetria, simplificando a instrumentação e tornando a observabilidade full-stack mais acessível e escalável para arquiteturas de microsserviços. Adotado por fornecedores como Datadog, New Relic e Grafana, o OTLP permite que as organizações construam pilhas de observabilidade flexíveis e independentes de um fornecedor específico, sem ficarem presas a soluções de um proprietário. Ele suporta compressão gzip e zstd, reduzindo o tamanho dos dados de telemetria e diminuindo o uso de largura de banda — uma vantagem chave para ambientes que lidam com grandes volumes de dados de telemetria. Projetado para crescimento de longo prazo, o OTLP garante que o OpenTelemetry continue sendo um padrão robusto e à prova de futuro, consolidando sua posição como a escolha de fato para transporte de telemetria.

84. React Hook Form

Adote

Destacamos o React Hook Form como uma alternativa ao Formik. Ao usar componentes não controlados por padrão, ele oferece um desempenho significativamente melhor, especialmente para formulários grandes. React Hook forms é bem integrado a diversas bibliotecas de validação baseadas em esquema, incluindo Yup, Zod e outras. Além disso, React Hook Form oferece muita flexibilidade, facilitando a integração com bases de código existentes e outras bibliotecas.

É possível usá-lo com bibliotecas externas de componentes controlados, como shadcn ou AntD. Com alto desempenho, integração fluida e desenvolvimento ativo, é uma ótima opção para construir formulários grandes ou aplicações com muitos formulários.

85. Effect

Experimente

Effect é uma poderosa biblioteca de TypeScript para construir programas síncronos e assíncronos complexos. O desenvolvimento de aplicações web frequentemente exige código repetitivo para tarefas como assincronia, concorrência, gerenciamento de estado e tratamento de erros. O Effect-TS simplifica esses processos utilizando uma abordagem de programação funcional. Aproveitando o sistema de tipos do TypeScript, o Effect ajuda a identificar problemas difíceis de detectar durante a compilação.

Nossa equipe anteriormente utilizava o TypeScript para programação funcional, mas descobriu que o Effect-TS oferece abstrações que se alinham melhor às tarefas do dia a dia. Além disso, ele facilita a composição e os testes do código. Embora abordagens tradicionais como `Promise/try-catch` ou `async/await` possam lidar com esses cenários, depois de usar o Effect, nossa equipe não encontrou motivo para voltar atrás.

86. Hasura GraphQL engine

Experimente

O Hasura GraphQL engine é uma camada de acesso a dados universal que simplifica a construção, execução e governança de APIs de alta qualidade em diferentes fontes de dados. Proporciona acesso instantâneo as APIs GraphQL sobre várias bases de dados (incluindo PostgreSQL, MongoDB e ClickHouse) e fontes de dados, permitindo que desenvolvedoras busquem apenas os dados de que precisam de forma rápida e segura. Achamos o Hasura fácil de implementar GraphQL para agregação de recursos no servidor e o aplicamos em múltiplos projetos de produtos de dados. No entanto, nós permanecemos cautelosas à sua poderosa consulta federada e gestão de esquema unificado. Uma adição recente notável é o recurso PromptQL do Hasura, que permite às desenvolvedoras aproveitar modelos de linguagem de grande porte (LLMs) para interações de dados mais naturais e intuitivas.

87. LangGraph

Experimente

LangGraph é um framework de orquestração projetado para criar aplicações multiagente com persistência de estado usando modelos de linguagem de grande porte (LLMs). Ele fornece um conjunto de primitivas de nível mais baixo, como nós e arestas, em comparação com as abstrações de nível mais alto do LangChain, oferecendo às desenvolvedoras controle mais detalhado sobre fluxos de agentes, gerenciamento de memória e persistência de estado. Essa abordagem baseada em grafos garante fluxos de trabalho previsíveis e personalizáveis, facilitando a depuração, escalabilidade e manutenção de aplicações em produção. Embora tenha uma curva de aprendizado mais íngreme, o design leve e modular do LangGraph o torna um framework poderoso para a criação de aplicações agentivas.

88. MarkItDown

Experimente

MarkItDown converte vários formatos (PDF, HTML, PowerPoint, Word) em Markdown, melhorando a legibilidade do texto e a retenção de significado. Como os modelos de linguagem de grande porte (LLMs) extraem o contexto a partir de pistas de formatação, como títulos e seções, o Markdown ajuda a preservar a estrutura para uma melhor compreensão. Em aplicações baseadas em RAG, nossas equipes usaram o MarkItDown para pré-processar documentos em Markdown, garantindo que marcadores lógicos (títulos, subseções) permanecessem intactos. Antes da geração de embeddings, a segmentação estruturada ajudou a manter o contexto complexo das seções, melhorando a clareza das respostas às consultas, especialmente para documentos complexos. Amplamente utilizado para documentação, o Markdown também torna a CLI do MarkItDown uma ferramenta valiosa para a produtividade de desenvolvedoras.

89. Federação de módulos

Experimente

A federação de módulos permite especificar módulos compartilhados e a deduplicação de dependências entre micro frontends. Com a versão 2.0, a federação de módulos evoluiu para funcionar de forma independente do webpack. Esta atualização introduz funcionalidades importantes, incluindo um runtime próprio, uma nova API de plugins e suporte para frameworks populares como React e Angular, além de empacotadores conhecidos, como Rspack e Vite. Ao adotar a federação de módulos, grandes aplicações web podem ser divididas em micro frontends menores e mais gerenciáveis, permitindo que diferentes equipes desenvolvam, implantem e escalem de forma independente, enquanto compartilham dependências e componentes de maneira eficiente.

90. Prisma ORM

Experimente

O Prisma ORM é um kit de ferramenta de banco de dados de código aberto que simplifica o trabalho com bancos de dados em aplicações Node.js e TypeScript. Ele oferece uma abordagem moderna e segura para acesso a bancos de dados, automatiza a migração de esquema de bancos de dados e fornece uma API de consulta intuitiva. Ao contrário dos ORMs típicos, o PrismaORM usa objetos JavaScript simples para definir os tipos de bancos de dados sem usar decorators ou classes. Nossa experiência com o PrismaORM é positiva; achamos que ele não apenas se alinha melhor com o panorama geral de desenvolvimento do TypeScript, como também se integra bem com o paradigma de programação funcional.

91. .NET Aspire

Avalie

.NET Aspire foi projetado para simplificar a orquestração de aplicações distribuídas na máquina local da desenvolvedora. Ele permite orquestrar vários serviços em um ambiente de desenvolvimento local — incluindo múltiplos projetos .NET, bancos de dados dependentes e contêineres Docker — com um único comando. Além disso, Aspire oferece ferramentas de observabilidade — como logs, rastreamento e painéis de métricas — para desenvolvimento local, separadas das usadas em ambientes de teste ou produção. Isso melhora significativamente a experiência da desenvolvedora no momento de criar, ajustar e depurar os aspectos de observabilidade de qualquer sistema em que esteja trabalhando.

92. Android XR SDK

Avalie

O Google, em parceria com a Samsung e a Qualcomm, lançou o Android XR, um novo sistema operacional projetado para headsets XR. O suporte está planejado para óculos e outros dispositivos. A maioria dos aplicativos Android é compatível, necessitando de poucas ou nenhuma modificação, mas a ideia é criar novos aplicativos espaciais do zero ou espacializar aplicativos existentes. O novo SDK Android XR é colocado como o kit de desenvolvimento de software (SDK) ideal para esses projetos, e o Google fornece orientações sobre como escolher as ferramentas e tecnologias incluídas no SDK. No momento, ele está disponível em versão prévia para desenvolvedoras.

93. Browser Use

Avalie

Browser Use é uma biblioteca python de código aberto que permite que agentes de IA baseados em modelos de linguagem de grande porte (LLMs) usem navegadores da web e acessem aplicações web. Ela pode controlar o navegador e executar etapas que incluem exploração de páginas, entrada de dados e extração de textos. Com a capacidade de gerenciar várias guias, ela pode executar ações coordenadas em diferentes aplicações web. Isso é útil para cenários onde agentes baseados em LLM precisam acessar conteúdo da web, executar ações nele e obter resultados. A biblioteca pode trabalhar com uma variedade de LLMs. Ela aproveita o Playwright para controlar o navegador, combinando compreensão visual com extração de estrutura HTML para melhor interação na web. Esta biblioteca está ganhando força em cenários multiagentes, permitindo que os agentes colaborem em fluxos de trabalho complexos envolvendo interações na web.

94. Crew AI

Avalie

CrewAI é uma plataforma projetada para ajudar a construir e gerenciar agentes de IA que podem trabalhar em conjunto para executar tarefas complexas. Pense nela como uma forma de criar um grupo de agentes de IA, cada um com suas próprias especialidades, colaborando para alcançar um objetivo comum. Nós já a mencionamos previamente no Radar, no tópico Agentes autônomos impulsionados por LLM. Além da biblioteca de código aberto em Python, a CrewAI agora oferece uma solução empresarial para que organizações possam criar aplicações baseadas em agentes voltadas a cenários reais de negócio, executá-las em sua própria infraestrutura de nuvem e conectá-las a fontes de dados existentes como Sharepoint ou JIRA. Já utilizamos CrewAI em vários desafios em produção, desde a validação automatizada de códigos promocionais até a investigação de falhas em transações e consultas de suporte ao cliente. Embora o cenário de agentes inteligentes continue evoluindo rapidamente, estamos confiantes em colocar a CrewAI na categoria Avalie.

95. ElysiaJs

Avalie

ElysiaJS é um framework ponta a ponta com tipagem segura para TypeScript, projetado primariamente para Bun mas também compatível com outros runtimes JavaScript. Diferente de alternativas como tRPC que impõe estruturas de interface de API específicas, ElysiaJS não impõe nenhuma estrutura de interface de API. Isso permite que desenvolvedoras criem APIs que seguem práticas estabelecidas pela indústria como RESTful, JSON: API ou OpenAPI, garantindo ao mesmo tempo a segurança de tipos de ponta a ponta. Quando usado com o Bun, ElysiaJS apresenta alto desempenho, sendo comparável a frameworks web de Java ou Go em alguns benchmarks. Vale a pena considerar o ElysiaJS, especialmente ao desenvolver um backend-for-frontend (BFF).

96. FastGraphRAG

Avalie

FastGraphRAG é uma implementação de código aberto do GraphRAG projetada para alta precisão e desempenho na recuperação de informações. Ele utiliza o Personalized PageRank para limitar a navegação no gráfico apenas aos nós mais relevantes entre todos os nós relacionados, aumentando a precisão da recuperação e melhorando a qualidade das respostas de modelos de linguagem de grande porte (LLMs). Ele também fornece uma representação visual do gráfico, ajudando as usuárias a entender os relacionamentos dos nós e o processo de recuperação. Com suporte para atualizações incrementais, é adequado para conjuntos de dados dinâmicos e em evolução. Otimizado para casos de uso do GraphRAG em larga escala, o FastGraphRAG melhora o desempenho ao mesmo tempo em que minimiza o consumo de recursos.

97. Gleam

Avalie

O Erlang/OTP é uma plataforma poderosa para construir sistemas distribuídos altamente concorrentes, escaláveis e tolerantes a falhas. Tradicionalmente, suas linguagens são dinamicamente tipadas, mas o Gleam introduz, a nível de linguagem, segurança de tipos. Construído sobre o BEAM, o Gleam combina a expressividade da programação funcional com segurança de tipo em tempo de compilação, reduzindo os erros em tempo de execução e melhorando a manutenibilidade. Com uma sintaxe moderna, ele integra bem com o ecossistema OTP, aproveitando os pontos fortes do Erlang e do Elixir enquanto garante forte interoperabilidade. A comunidade do Gleam é ativa e acolhedora, e estamos ansiosas para o seu desenvolvimento contínuo.

98. GoFr

Avalie

GoFr é um framework para a construção de microsserviços em Golang, projetado para simplificar o desenvolvimento ao abstrair o código repetitivo necessário para funcionalidades comuns de microsserviços, como logging, rastreamento, métricas, gerenciamento de configuração e documentação de API com Swagger. Ele suporta múltiplos bancos de dados, gerencia migrações de banco e facilita a comunicação pub/sub com brokers como Kafka e NATs. Além disso, o GoFr inclui agendamento de tarefas com cron jobs. Isso reduz a complexidade da criação e manutenção de microsserviços, permitindo que as pessoas desenvolvedoras se concentrem na escrita da lógica de negócio em vez de lidar com preocupações relacionadas à infraestrutura. Embora existam outras bibliotecas populares para construção de APIs web em Go, o GoFr está ganhando popularidade e vale a pena explorá-lo para microsserviços baseados em Golang.

99. Criptografia pós-quântica em Java

Avalie

No cerne da criptografia assimétrica, que garante a segurança da maioria das comunicações modernas, está um problema matematicamente complexo. No entanto, o problema usado nos algoritmos atuais será facilmente resolvido por computadores quânticos, o que impulsiona a pesquisa por alternativas. A criptografia baseada em rede é atualmente a candidata mais promissora. Embora computadores quânticos relevantes para a criptografia ainda estejam a anos de distância, a criptografia pós-quântica merece consideração para aplicações que precisam permanecer seguras por décadas. Há também o risco de que dados criptografados sejam registrados hoje para serem descriptografados quando os computadores quânticos estiverem disponíveis.

A criptografia pós-quântica em Java dá seus primeiros passos no JDK 24, definido para disponibilidade geral no final de março. Esta versão inclui o JEP 496 e JEP 497, que implementam um mecanismo de encapsulamento de chaves e um algoritmo de assinatura digital, ambos baseados em padrões e projetados para serem resistentes a futuros ataques de computação quântica. Embora a liboqs do projeto Open Quantum Safe forneça implementações baseadas em C com um wrapper JNI, é encorajador ver uma implementação nativa em Java surgindo também.

100. Presidio

Avalie

Presidio é um kit de desenvolvimento de software (SDK) de proteção de dados para identificar e anonimizar dados confidenciais em texto estruturado e não estruturado. Ele detecta informações de identificação pessoal (PII) — ou seja, dados sensíveis que podem identificar um indivíduo —, como números de cartão de crédito, nomes e locais, usando reconhecimento de entidade nomeada, expressões regulares e lógica baseada em regras. O Presidio suporta reconhecimento e desidentificação de entidades PII, permitindo que as empresas o adaptem aos seus requisitos de privacidade específicos. Embora o Presidio automatize a identificação de informações confidenciais, não é infalível e pode perder ou identificar dados incorretamente. Tenha cuidado ao confiar em seus resultados.

101. PydanticAI

Avalie

Conforme as tecnologias para construir aplicativos e agentes baseados em modelos de linguagem de grande porte (LLMs) evoluem rapidamente, os frameworks para construir e orquestrar essas aplicações muitas vezes têm dificuldade em acompanhar ou encontrar as abstrações corretas e

duradouras. PydanticAI é o mais novo membro deste espaço, buscando simplificar implementações, evitando complexidades desnecessárias. Desenvolvido pelas criadoras do popular Pydantic, ele é construído baseado nas lições aprendidas com frameworks anteriores — muitos dos quais já dependem do Pydantic. Ao invés de ser um canivete suíço, o PydanticAI oferece uma abordagem leve, porém poderosa. Ele se integra com todas as principais APIs de modelos, inclui gestão integrada de saída estruturada de LLMs e introduz uma abstração baseada em grafos para gerenciar workflows complexos de agentes.

102. Swift para aplicações com restrições de recursos

Avalie

Desde o lançamento do Swift 6.0, a linguagem se expandiu além do ecossistema da Apple, com suporte aprimorado para sistemas operacionais importantes, tornando-a mais viável para uso em aplicações com restrições de recursos. Tradicionalmente, esse espaço foi dominado por C, C++ e, mais recentemente, Rust, devido ao seu controle de baixo nível, alto desempenho e disponibilidade de compiladores e bibliotecas certificados que cumprem padrões como MISRA, ISO 26262 e ASIL. Enquanto o Rust começou a alcançar certificações semelhantes, o Swift ainda não iniciou esse processo, limitando seu uso em aplicações críticas de segurança.

A crescente adoção do Swift é impulsionada por seu equilíbrio entre desempenho e recursos de segurança, incluindo forte segurança de tipo e contagem automática de referência para gerenciamento de memória. Enquanto o modelo de propriedade do Rust oferece garantias de segurança de memória mais fortes, o Swift oferece um compromisso diferente que algumas desenvolvedoras acham mais acessível. Ambos, Swift e Rust, compartilham o backend do compilador LLVM/Clang, permitindo que avanços em um beneficiem o outro. Com sua capacidade de compilar para código de máquina otimizado, seu desenvolvimento de código aberto e seu suporte expansivo multiplataforma, o Swift está emergindo como um concorrente para uma gama mais ampla de aplicações — muito além de suas raízes no iOS.

103. Tamagui

Avalie

Tamagui é uma biblioteca que facilita o compartilhamento eficiente de estilos entre React web e React Native. Ela oferece um design system com componentes reutilizáveis, estilizados ou não, que funcionam perfeitamente em diferentes plataformas. Seu compilador de otimização opcional impulsiona o desempenho, convertendo componentes estilizados em CSS atômico com divs na web e objetos de estilo otimizados em visualizações nativas.

104. torchtune

Avalie

torchtune é uma biblioteca do PyTorch para criação, pós-treinamento e experimentação com modelos de linguagem de grande porte (LLMs). Ela oferece suporte a configurações com uma ou múltiplas GPUs e permite treinamento distribuído com FSDP2. A biblioteca fornece receitas baseadas em YAML para tarefas como ajuste fino, inferência, avaliação e treinamento consciente de quantização. Cada receita apresenta um conjunto de recursos focado, evitando configurações complexas baseadas em flags. Ela prioriza a simplicidade, favorecendo a clareza do código em vez de abstrações excessivas. Além disso, inclui uma CLI para baixar modelos, gerenciar receitas e executar experimentos de forma eficiente.

105. Uso excessivo de Node

Evite

Alguns anos atrás, observamos o uso excessivo de Node: o Node.js era frequentemente utilizado por razões questionáveis ou sem sequer considerar alternativas. Embora entendamos que algumas equipes prefiram uma stack de linguagem única — apesar dos trade-offs — continuamos a defender a programação poliglota. Na época, destacamos que o Node.js tinha uma reputação merecida por sua eficiência em workloads intensivos em IO, mas mencionamos que outros frameworks haviam alcançado esse nível, oferecendo APIs melhores e um desempenho geral superior. Também alertamos que o Node.js nunca foi adequado para workloads computacionalmente intensivos, uma limitação que continua sendo um grande desafio. Agora, com o crescimento de workloads intensivas em dados, vemos equipes enfrentarem dificuldades também nesse aspecto.

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.