Radar Tecnológico

Una guía de opinión sobre el entorno tecnológico actual

Volumen 32
Abril 2025

/thoughtworks

Diseño. Ingeniería. IA.

Sobre el Radar

Thoughtworkers son personas a las que les apasiona la tecnología. La construimos, la investigamos, la probamos, abogamos por el código abierto, escribimos sobre ella y constantemente tratamos de mejorarla para todas las personas. Nuestra misión es defender la excelencia del software y evolucionar la TI. Creamos y compartimos el Radar Tecnológico de Thoughtworks en apoyo de esa misión. El Technology Advisory Board de Thoughtworks, un grupo de líderes tecnológicos de alto nivel de Thoughtworks, crea el Radar. Se reúnen periódicamente para debatir la estrategia tecnológica global de Thoughtworks y las tendencias tecnológicas que tienen un impacto significativo en nuestra industria.

El Radar recoge el resultado de los debates del Technology Advisory Board en un formato que proporciona valor a una amplia gama de partes interesadas, desde las personas desarrolladoras hasta CTOs. El contenido pretende ser un resumen conciso.

Te animamos a explorar estas tecnologías. El Radar es de naturaleza gráfica y agrupa los elementos en técnicas, herramientas, plataformas, lenguajes y frameworks. Cuando los elementos del Radar podían aparecer en varios cuadrantes, elegimos el que nos pareció más apropiado. Además, agrupamos estos elementos en cuatro anillos para reflejar nuestra posición actual al respecto

Para más información sobre el Radar,consulta thoughtworks.com/es/radar/faq.

Un vistazo al Radar

El Radar se dedica a rastrear cosas interesantes, a las que nos referimos como blips. Organizamos los blips en el Radar utilizando dos elementos de categorización: cuadrantes y anillos. Los cuadrantes representan los diferentes tipos de blips. Los anillos indican nuestra recomendación para utilizar esa tecnología.

Un blip es una tecnología o técnica que desempeña un papel en el desarrollo de software. Los blips son cosas que están en "movimiento", es decir, que su posición en el Radar cambia a menudo, lo que suele indicar nuestra creciente confianza en recomendarlos a medida que avanzan por los anillos.

Adoptar: Estamos convencidas de que la industria debería adoptar estos ítems Nosotras los utilizamos cuando es apropiado en nuestros proyectos.

Probar: Vale la pena probarlos. Es importante entender cómo desarrollar estas capacidades. Las empresas deberían probar esta tecnología en proyectos en que se puede manejar el riesgo.

Evaluar: Vale la pena explorar con el objetivo de comprender cómo afectará a su empresa.

Resistir: Proceder con precaución.

Nuevo **Desplazado adentro /afuera** **Ningún cambio**

Nuestro Radar está orientado al futuro. Para dar paso a nuevos artículos, desvanecemos los que no se han movido recientemente, lo cual no es un reflejo de su valor, sino de nuestro limitado espacio en el Radar.

Contribuyentes

El Technology Advisory Board (TAB) es un grupo de 21 personas tecnólogas senior de Thoughtworks. El TAB se reúne dos veces al año en persona y virtualmente cada dos semanas. Su función principal es ser un grupo de asesoramiento para la CTO de Thoughtworks Rachel Laycock.

El TAB actúa como un organismo amplio que puede examinar los temas que afectan a la tecnología y a las personas tecnólogas en Thoughtworks. Esta edición del Radar Tecnológico de Thoughtworks es en base a la reunión virtual que TAB realizó remotamente en Febrero de 2025. .


Rachel Laycock (CTO)

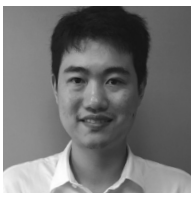

Martin Fowler (Chief Scientist)


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 producción

Grupo de personas editoras

- Alina Guerrero
- Elegardo Valdés
- Sergio Gutiérrez Santos

Marketing

- Daniel Negrete
- Magdalena Grondona

Grupo de personas traductoras

- Adrià López
- Alan Quimbita
- Alexandra Ortiz
- Andrea Astorga Bilbao
- Andrea Cuevas
- Andrea Peralta
- Andrea Velez
- Andrés Vázquez
- Ana López Estrella
- Anna Corral
- Antonia Castells
- Antonio Galisteo Cantero
- Ari Handler
- Carles Piqueras
- Carolina Melgarejo Pezoa
- Carolina Zhou Lin
- Carlos Fuentes
- Catalina Solis
- Cesar Abreu
- Dani Gomez
- David Arevalo
- Emmanuel Gomez Soler
- Erika Vacacela
- Esteban Moreno Arias
- Fernando Soto
- Fernanda Pérez
- Francis Alcántara
- Gabriel Salgado
- Gabriela Marques
- Ivanna Cevallos
- Javier Criollo
- Jan Gomez Roberts
- Joaquin Hervas
- Joangie Márquez
- Jordi Julià
- Jorge Arimany
- Jose María Alonso Montero
- Joseph Guerrero
- Jesús Cardenal Escribano
- Judit Navarro
- Juan Pablo Jorquera
- Juan Romero
- Leonardo Hidrovo
- Luis Ruiz Del Aguila
- Marc Palma
- Maria Laura Jaramillo
- Maria Martinez
- Mario Bizcocho
- Mercedes Crespo Jimenez
- Miguel Vela
- Milber Champutiz
- Nestor Gutierrez
- Noelia Martínez
- Oscar Barba Manzano
- Pablo Company Ramírez
- Pedro Vázquez
- Ricardo Casía
- Rodrigo Vallejo
- Sahard Hesamzadeh
- Sebastian Roman
- Sebastian Rueda
- Sendami Luque Liñan
- Valeria Zaldumbide
- Xavier Idrovo

Temas

Agentes supervisados en asistentes de códigos

Dos de nuestros temas destacan la rápida innovación en IA Generativa (GenIA) y uno de ellos trata sobre la aceleración de las capacidades de los asistentes de código. Cada vez son más las herramientas que permiten a developers generar código directamente desde un chat de IA dentro de su IDE; también llamado “basado en agentes”, “prompt-to-code” o “programación orientada por chat (CHOP por sus siglas en inglés)”. Con este enfoque, los asistentes de IA van más allá de responder preguntas o generar pequeños fragmentos de código (snippets); también navegan, modifican código, actualizan pruebas, ejecutan comandos y, en algunos casos, corrigen proactivamente errores de formato y compilación. Aunque seguimos siendo escépticos respecto a los agentes de codificación que prometen un desarrollo totalmente autónomo de grandes tareas, hemos visto resultados prometedores con este enfoque supervisado, en el que developers siguen guiando y supervisando las acciones del agente. Cursor, Cline y Windsurf lideran esta tendencia en el ámbito de las herramientas integradas en IDE, y GitHub Copilot también está avanzando. Asistentes ”orientados a agente” tales como aider, goose y Claude Code son alternativas para usar en terminales. A pesar de estos avances, seguimos siendo cautelosos sobre cómo esto aumenta la complacencia con código generado por IA, ya que a pesar de algunos resultados muy buenos, todavía vemos mucha necesidad de dirección y vigilancia en las revisiones de código. Un gran poder conlleva...

Evoluciones en la observabilidad

Se han visto movimientos significativos en el ámbito de la observabilidad, impulsados por la creciente complejidad de las arquitecturas distribuidas. Si bien la observabilidad siempre ha sido esencial, ésta ha seguido evolucionando junto con el resto del ecosistema de desarrollo de software. Uno de estos focos emergentes es la observabilidad de los grandes modelos de lenguaje (LLM, por sus siglas en inglés), pieza clave para la operacionalización de la IA. Hemos visto un aumento en las herramientas para monitorizar y evaluar el rendimiento de los LLM, incluyendo Weights & Biases Weave, Arize Phoenix, Helicone y HumanLoop. Otra tendencia es la integración de la observabilidad asistida por IA, en el que las herramientas aprovechan la inteligencia artificial para mejorar el análisis y la obtención de información. Además, la creciente adopción de OpenTelemetry está fomentando un panorama de observabilidad más estandarizado, permitiendo a los equipos mantenerse independientes de proveedores y ser más flexibles en la elección de sus herramientas.

Muchas de las principales herramientas de observabilidad; como Alloy, Tempo o Loki — ya son compatibles con OpenTelemetry. La súbita innovación en las herramientas de observabilidad demuestra una creciente concienciación de la industria sobre su importancia, creando un ciclo de retroalimentación en el que las prácticas y tecnologías en evolución se refuerzan mutuamente.

R de RAG

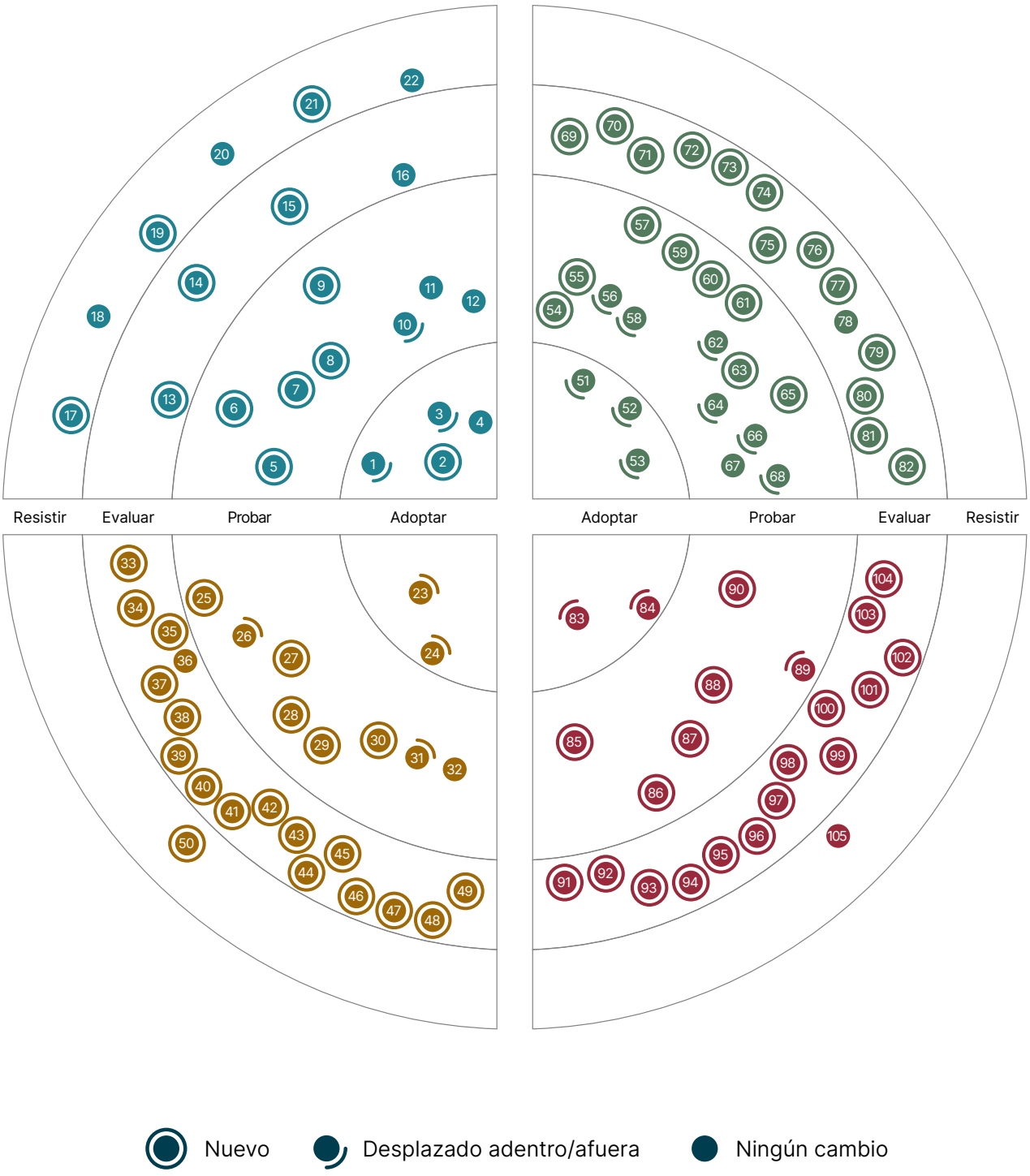
Esperamos que distintos aspectos del ecosistema de IA generativa evolucionen a ritmos variables, y en esta edición del Radar, vemos que esto ocurre con la R de RAG (retrieval-augmented generation, o generación aumentada por recuperación). Una de las interacciones clave con la caja negra del LLM es personalizar las entradas de los prompts para generar respuestas útiles y relevantes. La creciente necesidad de una recuperación efectiva en RAG ha llevado a la aparición de nuevas herramientas y técnicas presentadas en esta edición. Por ejemplo, discutimos RAG correctivo, el cual ajusta las respuestas de forma dinámica basándose en la retroalimentación o la heurística; Fusion-RAG, el cual combina varias fuentes y estrategias de recuperación para tener respuestas más exhaustivas y robustas; y Self-RAG, el cual evita por completo el paso de recuperación, obteniendo data bajo demanda. También resaltamos FastGraphRAG, el cual facilita la comprensión al crear grafos navegables por humanos. En base a las discusiones y nominaciones de nuestro equipo, la R de RAG es un tema candente y en rápida evolución.

Domando la frontera de los datos

Big data ha sido una preocupación para la industria durante mucho tiempo, pero las conversaciones en torno a esta edición del Radar se centraron no tanto en el tamaño, sino más específicamente en el manejo de datos ricos y complejos. Con la creciente presencia, e importancia, de los datos no estructurados en la empresa, garantizar que los datos se gestionen y empaqueten de manera efectiva para que puedan aprovecharse con éxito para todo, desde aplicaciones de IA hasta análisis de clientes, es hoy vital para las empresas.

Esto se refleja en varios blips: desde herramientas de bases de datos vectoriales hasta productos de análisis como Metabase, es sorprendente cuánto del ecosistema de software está siendo impulsado por lo que queremos y necesitamos hacer con los datos. Sin embargo, no se trata solo de herramientas, en esta edición señalamos el pensamiento de productos de datos, un marco de trabajo que alienta a los equipos a aplicar los principios del pensamiento de producto (product thinking) a las partes analíticas de su ecosistema. La aparición del pensamiento de producto de datos es, en cierta medida, el resultado del desafío continuo de aprovechar adecuadamente los datos (algo de lo que se ha hablado durante años, mucho antes del auge de la IA), el hecho de que haya encontrado su camino hacia el centro de atención (y nuestras conversaciones de Radar) demuestra que la necesidad de disciplina con el manejo de datos sigue siendo tan grande como siempre. Sin ella, las organizaciones pueden tener dificultades para innovar y bien podrían estar en desventaja comercial a medio y largo plazo.

El Radar



El Radar

Técnicas

Adoptar

1. Pensamiento de producto de datos
2. fuzz testing
3. Software Bill of Materials
4. Modelado de amenazas

Probar

5. Colección de peticiones API como artefactos de producto
6. Architecture advice process
7. GraphRAG
8. Gestión de acceso privilegiado justo a tiempo
9. Destilación de Modelos
10. Prompt Engineering (o ingeniería de instrucciones)
11. Modelos de lenguaje pequeños
12. Usando GenAI para entender bases de código legado

Evaluar

13. Diseño de código amigable con la IA
14. Pruebas de IU impulsadas por IA
15. Competence envelope como un modelo para comprender fallas de sistema
16. Salida estructurada de LLMs

Resistir

17. TI en la sombra acelerado por IA
18. Complacencia con código generado por IA
19. Asistentes de codeo local
20. Reemplazar codificación en parejas con IA
21. Reverse ETL
22. SAFe™

Plataformas

Adoptar

23. GitLab CI/CD
24. Trino

Probar

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

Evaluar

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 razonamiento
44. Restate
45. Supabase
46. Synthesized
47. Tonic.ai
48. turbopuffer
49. VectorChord

Resistir

50. Gestión de API híbridas con Tyk

Herramientas

Adoptar

51. Renovate
52. uv
53. Vite

Probar

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 Ingeniería de Software
67. Tuple
68. Turborepo

Evaluar

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

Resistir

—

Lenguajes y Frameworks

Adoptar

83. OpenTelemetry
84. React Hook Form

Probar

85. Effect
86. Motor GraphQL de Hasura
87. LangGraph
88. MarkItDown
89. Module Federation
90. Prisma ORM

Evaluar

91. .NET Aspire
92. SDK para Android XR
93. Browser Use
94. CrewAI
95. ElysiaJs
96. FastGraphRAG
97. Gleam
98. GoFr
99. Criptografía postcuántica con Java
100. Presidio
101. PydanticAI
102. Swift para aplicaciones con recursos limitados
103. Tamagui
104. torchtune

Resistir

105. Sobrecarga de Node

Técnicas

Adoptar

1. Pensamiento de producto de datos
2. fuzz testing
3. Software Bill of Materials
4. Modelado de amenazas

Probar

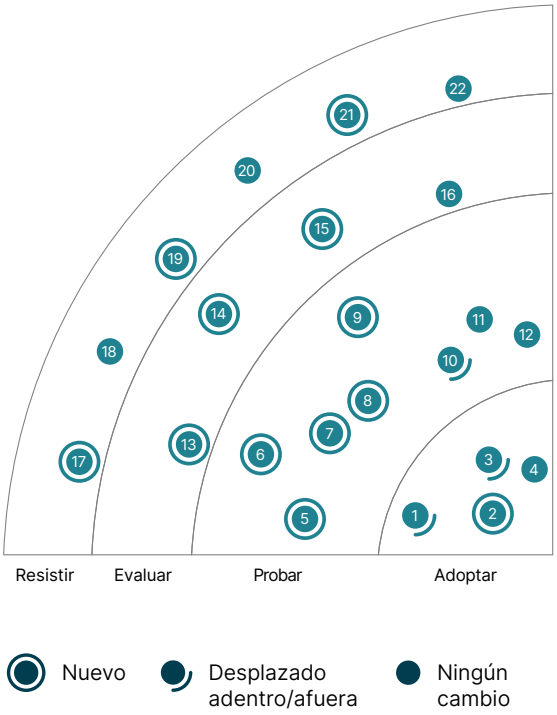
5. Colección de peticiones API como artefactos de producto
6. Architecture advice process
7. GraphRAG
8. Gestión de acceso privilegiado justo a tiempo
9. Destilación de Modelos
10. Prompt Engineering (o ingeniería de instrucciones)
11. Modelos de lenguaje pequeños
12. Usando GenAI para entender bases de código legado

Evaluar

13. Diseño de código amigable con la IA
14. Pruebas de IU impulsadas por IA
15. Competence envelope como un modelo para comprender fallas de sistema
16. Salida estructurada de LLMs

Resistir

17. TI en la sombra acelerado por IA
18. Complacencia con código generado por IA
19. Asistentes de codeo local
20. Reemplazar codificación en parejas con IA
21. Reverse ETL
22. SAFe™



1. Pensamiento de producto de datos

Adoptar

Las organizaciones adoptarán activamente pensamiento de producto de datos como práctica estándar para gestionar activos de datos. Este enfoque trata los datos como un producto con su propio ciclo de vida, estándares de calidad y un enfoque en conocer las necesidades del consumidor. Ahora lo recomendamos como la elección predeterminada para la gestión de datos, independientemente de si las organizaciones eligen arquitecturas como data mesh o lakehouse.

Enfatizamos la orientación al consumidor en el pensamiento de producto de datos para impulsar un mayor adopción y generación de valor. Esto significa diseñar productos de datos trabajando desde los casos de uso hacia atrás. También nos enfocamos en capturar y gestionar tanto metadatos relevantes del negocio como metadatos técnicos usando catálogos de datos modernos como DataHub, Collibra, Atlan e Informática. Estas prácticas mejoran el descubrimiento y usabilidad de los datos. Adicionalmente, aplicamos el pensamiento de producto de datos para escalar iniciativas de IA y crear datos listos para IA. Este enfoque incluye una gestión integral del ciclo de vida, asegurando que los datos no solo estén bien gobernados y sean de alta calidad, sino que también se retiren en cumplimiento con los requisitos legales y regulatorios cuando ya no sean necesarios.

2. Fuzz testing

Adoptar

o simplemente fuzzing, es una técnica de pruebas que ha existido durante mucho tiempo, pero sigue siendo una de las técnicas menos conocidas. El objetivo es proporcionar a un sistema de software todo tipo de entradas inválidas y observar su comportamiento. Para un endpoint HTTP, por ejemplo, las solicitudes incorrectas deberían resultar en errores 4xx, pero el fuzz testing a menudo provoca errores 5xx o peores. Bien documentado y soportado por herramientas, el fuzz testing es más relevante que nunca con mayor cantidad de código generado por IA y Complacencia con el código generado por IA. Esto significa que ahora es un buen momento para Adoptar el fuzz testing y asegurar que el código siga siendo robusto y seguro.

3. Software Bill of Materials

Adoptar

Desde nuestro primer blip sobre Software Bill of Materials (SBOM) en 2021, la generación de SBOM ha pasado de ser una práctica emergente a una práctica por defecto en nuestros proyectos. El ecosistema ha madurado significativamente, ofreciendo un conjunto de herramientas robusto y una integración fluida con CI/CD. Herramientas como Syft, Trivy, y Snyk permiten la generación integral de SBOM, desde el código fuente hasta las imágenes de contenedores, además de escaneo de vulnerabilidades. Plataformas como FOSSA y Chainloop mejoran la gestión de riesgos de seguridad al integrarse con los flujos de desarrollo y reforzar políticas de seguridad. Aunque un estándar universal para SBOM sigue en evolución, el amplio soporte para SPDX y CycloneDX ha reducido los desafíos de adopción. Los sistemas de IA también requieren SBOM, como lo demuestran el AI Cyber Security Code of Practice del gobierno del Reino Unido y el AI Cybersecurity Collaboration Playbook. de CISA. Seguiremos monitoreando los avances en este ámbito.

4. Modelado de amenazas

Adoptar

En el panorama en rápida evolución del desarrollo de software impulsado por IA, el modelado de amenazas es más crucial que nunca para crear software seguro, manteniendo la agilidad y evitando el security sandwich. El modelado de amenazas; un conjunto de técnicas para identificar y clasificar

potenciales amenazas, se aplica en diversos contextos, incluyendo aplicaciones de IA generativa, que introducen riesgos de seguridad únicos. Para que sea eficaz, debe realizarse con frecuencia a lo largo del ciclo de vida del software y funciona mejor junto con otras prácticas de seguridad. Entre ellas se incluye la definición de requisitos de seguridad interfuncionales para abordar riesgos comunes en las tecnologías del proyecto y el aprovechamiento de escáneres de seguridad automatizados para monitoreo continuo.

5. Colección de peticiones API como artefactos de producto

Probar

Tratar APIs como producto significa priorizar la experiencia del desarrollador(a), no sólo mediante un diseño razonable y estandarizado de dichas APIs sino también proporcionando una documentación completa y una experiencia de onboarding fluida. Aunque las especificaciones de OpenAPI (Swagger) pueden documentar interfaces de APIs de manera efectiva, el onboarding continúa siendo un desafío. Developers(as) necesitan acceso rápido a ejemplos funcionales, con autenticación pre-configurada y datos de prueba realistas. Con la evolución de herramientas de clientes API (como Postman, Bruno e Insomnia), recomendamos tratar las colecciones de peticiones API como artefactos del producto API. Las colecciones de peticiones API deben ser diseñadas cuidadosamente para guiar a developers(as) a través de flujos de trabajo clave, ayudándoles a comprender el lenguaje y funcionalidad del dominio de la API con el mínimo esfuerzo. Para mantener estas colecciones actualizadas, recomendamos almacenarlas en un repositorio e integrarlas dentro del pipeline de despliegue de la API.

6. Architecture advice process

Probar

Uno de los retos constantes en equipos grandes de software es determinar quién toma las decisiones arquitectónicas que dan forma a la evolución de los sistemas. El informe State of DevOps report revela que el enfoque tradicional de las Juntas de Revisión de Arquitectura es contraproducente, y a menudo obstaculiza el flujo de trabajo y se correlaciona con un bajo rendimiento organizacional. Una alternativa convincente es architectural advice process — un enfoque descentralizado donde cualquiera puede tomar una decisión arquitectónica, siempre que busque asesoramiento de los afectados y de aquellos con experiencia relevante. Este método permite a los equipos optimizar el flujo sin comprometer la calidad arquitectónica, tanto a pequeña como a gran escala. A primera vista, esta alternativa puede parecer controversial, pero prácticas como Architecture Decision Records y los foros de asesoramiento garantizan que las decisiones se mantengan informadas, mientras que también empoderan a quienes están más cerca del trabajo a tomar decisiones. Hemos visto este modelo tener éxito a escala en un gran número de organizaciones, incluyendo aquellas en industrias altamente reguladas.

7. GraphRAG

Probar

En nuestra última actualización de RAG, introdujimos GraphRAG , descrito originalmente en el artículo de Microsoft como un enfoque en dos pasos: (1) fragmentación de documentos y uso de análisis basado en LLM de los fragmentos para crear un grafo de conocimientos; (2) recuperación de fragmentos relevantes en el momento de la consulta mediante incrustaciones mientras se siguen las aristas del grafo de conocimiento para descubrir fragmentos relacionados adicionales, que se añaden al prompt aumentado. En muchos casos, este enfoque mejora las respuestas generadas por LLM. Hemos observado beneficios similares al utilizar IA generativa para comprender bases de código heredadas, donde utilizamos información estructural, como árboles sintácticos abstractos y dependencias, para construir el grafo de conocimiento. El patrón GraphRAG ha ganado adeptos, con

herramientas y frameworks como el paquete en Python GraphRAG de Neo4j, que están surgiendo para soportarlo. También consideramos que Graphiti se ajusta a una interpretación más amplia de GraphRAG como patrón.

8. Gestión de acceso privilegiado justo a tiempo

Probar

El principio de privilegio mínimo garantiza que los usuarios y sistemas solo tengan el acceso mínimo necesario para realizar las tareas. El abuso de credenciales privilegiadas es un factor clave en las brechas de seguridad, siendo la escalada de privilegios un vector de ataque común. Los atacantes suelen empezar con un acceso de bajo nivel y explotan vulnerabilidades del software o configuraciones erróneas para obtener privilegios de administrador, especialmente cuando las cuentas tienen permisos excesivos o innecesarios. Otro riesgo que a menudo se pasa por alto es el de los privilegios permanentes: accesos privilegiados disponibles de forma continua que amplían la superficie de ataque. La gestión de acceso privilegiado justo a tiempo mitiga este riesgo al conceder acceso solo cuando es necesario y revocarlo inmediatamente después, minimizando la exposición. Un modelo de seguridad de menor privilegio garantiza que los usuarios, aplicaciones y sistemas tengan únicamente los derechos imprescindibles durante el mínimo tiempo posible, un requisito fundamental para el cumplimiento normativo y la seguridad regulatoria. Nuestros equipos lo han implementado a través de un flujo de trabajo automatizado que activa un proceso de aprobación ligero, asigna roles temporales con acceso restringido e impone un tiempo de vida a cada rol, lo que garantiza que los privilegios expiren automáticamente una vez completada la tarea.

9. Destilación de Modelos

Probar

Las leyes de Escalabilidad han sido un factor clave del auge de la IA; el principio de que modelos más grandes, conjuntos de datos más extensos y mayores recursos de cómputo conducen a sistemas de IA más potentes. Sin embargo, el hardware de consumo y los dispositivos perimetrales a menudo carecen de la capacidad para soportar modelos a gran escala, lo que crea la necesidad de la destilación de modelos.

La destilación de modelos transfiere conocimiento de un modelo más grande y potente (maestro) a un modelo más pequeño y eficiente en coste (estudiante). El proceso suele implicar la generación de un conjunto de datos de muestra a partir del modelo maestro y el ajuste del modelo estudiante para que capture sus propiedades estadísticas. A diferencia de la poda o la cuantización, que se enfocan en la compresión de modelos suprimiendo parámetros, la destilación intenta retener conocimiento específico del dominio, minimizando la pérdida de precisión. También se puede combinar con la cuantización para una optimización adicional.

Propuesta originalmente por Geoffrey Hinton y otros, la destilación de modelos ha sido adoptada ampliamente. Un ejemplo destacado es Qwen/Llama la versión destilada de DeepSeek R1, que mantiene sólidas capacidades de razonamiento en modelos más pequeños. Con su creciente madurez, la técnica ya no se limita sólo a los laboratorios de investigación; ahora se aplica en una amplia variedad de proyectos, desde industriales hasta personales. Proveedores como OpenAI y Amazon Bedrock ofrecen guías para ayudar a developers a destilar sus propios modelos de lenguaje reducido (SLMs, small language model, por sus siglas en inglés). Creemos que la adopción de la destilación de modelos puede ayudar a las organizaciones a gestionar los costes de despliegue de los LLM al tiempo que libera el potencial de la inferencia LLM en dispositivos.

10. Prompt Engineering (o ingeniería de instrucciones)

Probar

Prompt engineering se refiere al proceso de diseñar y refinar los prompts (instrucciones) para modelos de IA generativa, con el objetivo de producir respuestas de alta calidad conscientes del contexto. Esto implica elaborar indicaciones claras, específicas y relevantes, ajustadas a la tarea o a la aplicación, para optimizar el resultado del modelo. A medida que las LLM evolucionan, particularmente con la llegada de los modelos de razonamiento, las prácticas del prompt engineering deben ser adaptadas. Basados en nuestra experiencia con la generación de código con IA, hemos observado que los prompts con pocos ejemplos pueden ofrecer un menor rendimiento comparado con prompts sin ejemplos al trabajar con modelos de razonamiento. Además, la técnica ampliamente usada de cadena de razonamiento o chain-of-thought (CoT) puede degradar el desempeño de los modelos de razonamiento, probablemente debido a que el aprendizaje por refuerzo ya ha afinado su mecanismo interno de CoT.

Nuestra experiencia práctica se alinea con lo que señala la investigación académica, que sugiere que "los modelos más avanzados podrían eliminar la necesidad del prompt engineering en el desarrollo de software". No obstante, las técnicas tradicionales del prompt engineering seguirán teniendo un rol crucial en reducir alucinaciones y mejorar la calidad de las respuestas, especialmente considerando las diferencias en tiempos de respuestas y costos de tokens entre los modelos de razonamiento y los LLM generales. Al crear aplicaciones con agentes autónomos, recomendamos elegir los modelos estratégicamente, con base en las necesidades y seguir refinando tanto las plantillas de prompts como sus técnicas correspondientes. Encontrar el equilibrio entre la calidad, el tiempo de respuesta y el costo de los tokens sigue siendo clave para maximizar la efectividad de los LLM.

11. Modelos de lenguaje pequeños

Probar

El reciente anuncio de DeepSeek R1 es un gran ejemplo de por qué los small language models (SLMs) siguen siendo interesantes. La versión completa de R1 tiene 671 mil millones de parámetros y requiere alrededor de 1.342 GB de VRAM para funcionar, algo que solo se logra utilizando un mini cluster de ocho GPUs NVIDIA de última generación. Pero DeepSeek también está disponible en versión "distilled" en Qwen y Llama — modelos más pequeños y open-weight —, transfiriendo efectivamente sus capacidades y permitiendo que se ejecute en hardware mucho más modesto. Aunque el modelo sacrifica algo de rendimiento en esos tamaños reducidos, aún permite un gran salto en rendimiento respecto a los SLMs anteriores. El campo de los SLM sigue innovando en otros ámbitos, también. Desde el último Radar, Meta introdujo Llama 3.2 en tamaños de 1B y 3B, Microsoft lanzó Phi-4, ofreciendo resultados de alta calidad con un modelo de 14B, y Google presentó PaliGemma 2, un modelo de visión-lenguaje en tamaños de 3B, 10B y 28B. Estos son solo algunos de los modelos que se están lanzando actualmente en tamaños más pequeños y, sin duda, es una tendencia importante a seguir.

12. Usando GenAI para entender bases de código legado

Probar

En los últimos meses, el uso de GenAI para comprender bases de código legado ha generado verdaderos progresos. Herramientas populares como GitHub Copilot se están promocionando como capaces de ayudar a modernizar bases de código legado. Herramientas como Cody de Sourcegraph facilitan a developers la navegación y comprensión de bases de código completas. Estas herramientas utilizan multitud de técnicas de GenAI para proporcionar ayuda contextual, simplificando el trabajo con sistemas legados complejos. Además, frameworks especializados como S3LLM están mostrando

cómo los modelos de lenguaje extensos (LLMs) pueden manejar software científico de gran escala -como los escritos en Fortran o Pascal- llevando la comprensión mejorada por GenAI a bases de código fuera de la TI empresarial tradicional. Creemos que esta técnica continuará ganando terreno dada la enorme cantidad de software legacy en el mundo.

13. Diseño de código amigable con la IA

Evaluar

Los agentes de ingeniería de software supervisados son cada vez más capaces de identificar actualizaciones necesarias e implementar cambios más extensos en una base de código. Al mismo tiempo, se observa un mayor nivel de complacencia con el código generado por IA, con desarrolladoras y developers que se resisten a revisar conjuntos de cambios extensos realizados por IA. Una justificación típica para esta actitud es la percepción de que la calidad del código orientado a humanos no es tan crítica porque la IA podrá encargarse de futuras modificaciones; sin embargo, los asistentes de codificación basados en IA funcionan mejor con bases de código bien estructuradas y diseñadas, lo que hace que un código amigable hacia la IA sea crucial para su mantenibilidad. Afortunadamente, un buen diseño de software para humanos también beneficia a la IA. Los nombres significativos proporcionan contexto de dominio y funcionalidad; la modularidad y las abstracciones permiten a la IA gestionar mejor el contexto al limitar las modificaciones necesarias; y el principio DRY (Don't Repeat Yourself o No te repitas) reduce el código duplicado, ayudando a que la IA tenga un comportamiento más predecible. Hasta ahora, los patrones más amigables con la IA coinciden con las mejores prácticas establecidas en el desarrollo de software. A medida que la IA siga evolucionando, es probable que surjan patrones aún más específicos para su optimización, por lo que tener esto en cuenta al diseñar código será de gran utilidad.

14. Pruebas de IU impulsadas por IA

Evaluar

Están surgiendo nuevas técnicas de asistencia impulsadas por IA en equipos de desarrollo de software, más allá de la mera generación de código. Un área que está ganando tracción es la de pruebas de IU impulsadas por IA , que se apoyan en la capacidad de los LLMs para interpretar interfaces gráficas de usuario. Existen diversas aproximaciones para esto. Hay una categoría de herramientas que usan LLMs multimodales ajustados para el procesamiento de instantáneas de la IU, que permiten a los scripts de prueba escritos en lenguaje natural, navegar por la aplicación. Ejemplos en este espacio son QA.tech o LambdaTests' KaneAI. combina modelos de base multimodal con las perspectivas que Browser Use, combina modelos de base multimodal con las perspectivas que Playwright's extrae de la estructura de la pagina web, en lugar de apoyarse en modelos finamente ajustados.

Al integrar pruebas de IU impulsadas por IA en una estrategia de pruebas, es crucial considerar donde producen más valor. Estos métodos pueden complementar las pruebas manuales exploratorias, y aunque la falta de determinismo de los LLMs puede introducir fallos aleatorios, su imprecisión puede ser una ventaja. Esto puede ser útil para probar aplicaciones heredadas con selectores faltantes o aplicaciones que cambian frecuentemente sus etiquetas y rutas de clic.

15. Competence envelope como un modelo para comprender fallas de sistema

Evaluar

La Teoría de la Extensibilidad Grácil define las reglas básicas que gobiernan sistemas adaptativos, incluyendo los sistemas socio-técnicos involucrados en software de construcción y operación. Un concepto clave en esta teoría es el de Competence envelope — el límite en el que un sistema puede funcionar robustamente frente a fallas. Cuando un sistema es llevado más allá de su competence envelope, se vuelve frágil y es más propenso a fallar. Este modelo provee un lente valioso para comprender fallas sistémicas, como se pudo observar en las fallas complejas que llevaron a la caída de Canva de 2024. La Teoría de la Residualidad, un desarrollo reciente en Software Architecture Thinking, ofrece una forma de poner a prueba el Competence Envelope de un sistema a partir de la introducción deliberada de factores de estrés a dicho sistema, y el posterior análisis de cómo éste se ha adaptado a estos estresores de manera histórica a lo largo del tiempo. Los enfoques se alinean con los conceptos de anti-fragilidad, resiliencia y robustez en sistemas socio-técnicos, y nos entusiasma ver aplicaciones prácticas emerger en este campo.

16. Salida estructurada de LLMs

Evaluar

La salida estructurada de LLMs se refiere a la práctica de restringir la respuesta de un modelo de lenguaje, a un esquema definido. Esto se puede lograr ya sea a través de instruir a un modelo generalizado que responda en un formato particular o realizando fine-tuning a un modelo para obtener una salida "nativa", por ejemplo, JSON. OpenAI ahora soporta salida estructurada, permitiendo a developers proporcionar un esquema JSON, pydantic o un objeto Zod para limitar las respuestas de un modelo. Esta capacidad es particularmente valiosa ya que permite llamadas a funciones, interacciones con una API e integraciones externas, donde la precisión y el cumplimiento de un formato son críticas. La salida estructurada no solo mejora la forma en que los LLMs pueden interactuar con el código, sino que también soporta un mayor cantidad de casos de uso, como generación de markup para el renderizado de gráficos. Adicionalmente, la salida estructurada demostró reducir las posibilidades de alucinaciones en la salida de un modelo.

17. TI en la sombra acelerado por IA

Resistir

La IA está reduciendo las barreras para que quienes no codifican puedan crear e integrar software por sí mismos, en lugar de esperar a que el departamento de TI se encargue de sus necesidades o requisitos. Aunque nos entusiasma el potencial que esto desbloquea, también nos preocupan los primeros indicios de TI en la sombra acelerado por IA. Las plataformas de automatización de flujos de trabajo sin código ahora admiten la integración de API de IA (por ejemplo, OpenAI o Anthropic), lo que hace tentador usar la IA como cinta adhesiva — uniendo integraciones que antes no eran posibles, como convertir mensajes de chat de un sistema en llamadas a la API de un ERP mediante IA. Al mismo tiempo, los asistentes de codificación impulsados por IA se están volviendo más autónomos, lo que permite a quienes no codifican, pero tienen una formación básica, la posibilidad de crear aplicaciones de utilidad internas.

Esto tiene todas las características de la próxima evolución de las hojas de cálculo que aún impulsan procesos críticos en algunas empresas — pero con un alcance mucho mayor. Si no se controla, esta nueva TI en la sombra podría provocar la proliferación de aplicaciones no gobernadas y potencialmente inseguras, dispersando los datos en cada vez más sistemas. Las organizaciones deben ser conscientes de estos riesgos y evaluar cuidadosamente las ventajas y desventajas entre la rápida resolución de problemas y la estabilidad a largo plazo.

18. Complacencia con código generado por IA

Resistir

A medida que los asistentes de IA para escritura de código ganan peso, también lo hace el número de investigaciones y datos que traen a la luz la preocupación por la complacencia con el código generado por IA. El último estudio de calidad de código de GitClear's muestra que, en el 2024, el código duplicado y la rotación de código aumentaron más de lo previsto, mientras la actividad de refactorización en el historial de commits disminuyó. Además, como reflejo de la complacencia en el código generado por IA, un estudio de Microsoft sobre trabajadores de conocimiento mostró que la confianza generada por la IA suele venir a costa del pensamiento crítico, un patrón que hemos observado a medida que la complacencia se instala con el uso prolongado de asistentes de programación. El auge de los agentes de ingeniería de software supervisados amplifica aún más los riesgos, ya que, cuando la IA genera conjuntos de cambios cada vez más grandes, developers se enfrentan a mayores desafíos a la hora de revisar los resultados. La aparición de la codificación por vibras, donde developers permiten que la IA genere código con una revisión mínima, ilustra la creciente confianza en los resultados generados por la IA. Si bien este enfoque puede ser adecuado para prototipos u otros tipos de código desechable, recomendamos encarecidamente no usarlo para código de producción.

19. Asistentes de codeo local

Resistir

Las organizaciones se mantienen cautelosas con respecto a las IAs asistentes de código de terceros, particularmente debido a preocupaciones con respecto a la confidencialidad del código. Como resultado, muchos developers están considerando asistentes de código locales , IAs que corren completamente en sus propias máquinas, eliminando la necesidad de enviar código a servidores externos. Sin embargo, los asistentes locales aún se quedan atrás de sus contrapartes de nube, que confían en modelos más grandes y más capaces. Incluso en máquinas de desarrollo de alta gama, los modelos más pequeños mantienen capacidades limitadas. Hemos encontrado que tienen dificultades con prompts complejos, carecen de la ventana de contexto necesaria para problemas más grandes y a menudo no pueden gatillar integraciones de herramientas o llamadas a funciones. Estas capacidades son especialmente esenciales para los flujos de trabajo agentivos, los cuales son la vanguardia en asistencia de código actualmente.

Así que, si bien recomendamos proceder con bajas expectativas, existen algunas capacidades que son válidas a nivel local. Ciertos IDEs populares actualmente incorporan modelos más pequeños en sus características centrales, tales como el completado de código predictivo de Xcode y el completado de líneas completas de código JetBrains. Y LLMs que puedan correr localmente como Qwen Coder son un paso hacia sugerencias locales en línea y manejo de queries simples de código. Se puede probar estas capacidades con Continue, que soporta la integración de modelos locales vía tiempo de ejecución como Ollama.

20. Reemplazar codificación en parejas con IA

Resistir

Cuando se habla de codificación en pares, inevitablemente surge el tema de pair programming Nuestra profesión tiene una relación de amor-odio con ella: algunos la adoran, otros no la soportan. Los codificadores en pares plantean ahora la siguiente pregunta: ¿puede un humano trabajar en par con la IA, en lugar de con otro humano, y obtener los mismos resultados para el equipo? GitHub

Copilot incluso se llama a sí mismo «tu codificador en parejas de IA». Aunque creemos que un asistente de programación puede aportar algunas de las ventajas de la programación en parejas, desaconsejamos totalmente reemplazar la programación en parejas por la IA. Enmarcar a los asistentes de programación como codificadores en pareja ignora uno de los principales beneficios de la programación en pareja: mejorar el equipo, no sólo a los colaboradores individuales. Los asistentes de programación pueden ser útiles para desbloquearse, aprender sobre una nueva tecnología, incorporarse o agilizar el trabajo táctico para que podamos centrarnos en el diseño estratégico. Pero no contribuyen a ninguno de los beneficios de la colaboración en equipo, como mantener bajo el trabajo en curso, reducir los traspasos y el re-aprendizaje, hacer posible la integración continua o mejorar la propiedad colectiva del código.

21. Reverse ETL

Resistir

Estamos observando una preocupante proliferación del llamado Reverse ETL. Los procesos ETL tradicionales tienen su lugar en las arquitecturas de datos convencionales, donde transfieren información desde sistemas de procesamiento transaccional hacia un sistema de análisis centralizado, como un data warehouse o un data lake. Si bien esta arquitectura tiene limitaciones bien documentadas, muchas de las cuales son abordadas por un data mesh, sigue siendo común en las empresas. En este contexto, mover datos desde un sistema de análisis centralizado de vuelta a un sistema transaccional puede tener sentido en ciertos casos, como cuando el sistema central puede agregar datos de múltiples fuentes o como parte de una arquitectura transicional en el proceso de migración hacia un data mesh. Sin embargo, estamos viendo una creciente tendencia en la que proveedores de productos utilizan Reverse ETL como excusa para trasladar cada vez más lógica de negocio a una plataforma centralizada: su propio producto. Este enfoque agrava muchos de los problemas que generan las arquitecturas de datos centralizadas, por lo que recomendamos tener extrema precaución al introducir flujos de datos desde una plataforma de datos centralizada y expansiva hacia sistemas de procesamiento transaccional.

22. SAFe™

Resistir

Observamos una adopción continua de SAFe™ (Scaled Agile Framework®). También seguimos observando que los procesos excesivamente estandarizados y basados en fases de SAFe generan fricción, que puede promover silos y que su control de arriba hacia abajo genera desperdicios en el flujo de valor y desalienta la creatividad del talento de ingeniería, mientras limita la autonomía y experimentación en los equipos. Una razón clave para su adopción es la complejidad de hacer que una organización se vuelva ágil, con empresas que esperan que un marco como SAFe ofrezca un atajo simple y basado en procesos para volverse ágiles. Dada la amplia adopción de SAFe -incluso entre nuestros clientes- hemos capacitado a más de 100 consultores de Thoughtworks para apoyarlos mejor. A pesar de este conocimiento profundo y los múltiples intentos, nuestra impresión es que a veces simplemente no hay una solución simple para un problema complejo, y seguimos recomendando un enfoque ágil, basado en entrega de valor y gobernanza que funcionen en conjunto con un programa integral de cambio.

Scaled Agile Framework® y SAFe™ son marcas comerciales de Scaled Agile, Inc.

Plataformas

Adoptar

23. GitLab CI/CD
24. Trino

Probar

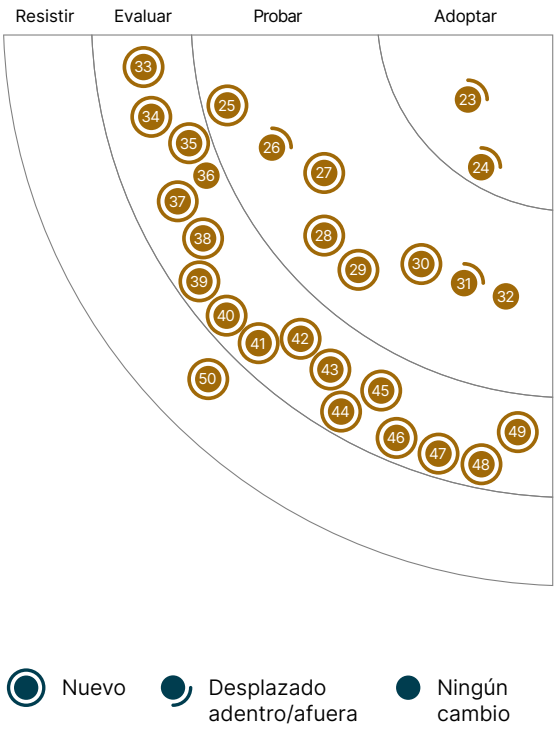
25. ABsmartly
26. Dapr
27. Grafana Alloy
28. Grafana Loki
29. Grafana Tempo
30. Railway
31. Unblocked
32. Weights & Biases

Evaluar

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 razonamiento
44. Restate
45. Supabase
46. Synthesized
47. Tonic.ai
48. turbopuffer
49. VectorChord

Resistir

50. Gestión de API híbridas con Tyk



23. GitLab CI/CD

Adoptar

GitLab CI/CD ha evolucionado hasta convertirse en un sistema completamente integrado en GitLab, cubriendo todo desde la integración y testeo de código hasta su despliegue y monitorización. Soporta complejos flujos de trabajo con funcionalidades como pipelines multi-etapa, caché, ejecución paralela y auto-escalado de ejecutores, adecuado para proyectos de gran escala y necesidades en pipelines complejas. Queremos destacar sus herramientas de seguridad y cumplimiento integradas (como análisis SAST y DAST), que lo hacen ideal para casos de uso con altos requisitos de cumplimiento. También se integra perfectamente con Kubernetes, dando soporte a flujos de trabajo nativos de la nube, y ofrece logging en tiempo real, reportes de tests y trazabilidad para una mejor observabilidad.

24. Trino

Adoptar

Trino es un motor de consultas SQL distribuido y de código abierto diseñado para consultas analíticas interactivas sobre big data. Está optimizado para ejecutarse tanto en entornos locales como en la nube, y permite consultar los datos donde residen, incluyendo bases de datos relacionales y diversos almacenes propietarios a través de conectores. Trino también puede consultar datos almacenados en formatos Parquet y formatos de tabla abiertos como Apache Iceberg. Sus capacidades integradas de federación de consultas permiten consultar datos de múltiples fuentes como si fueran una sola tabla lógica, lo que lo convierte en una excelente opción para cargas de trabajo analíticas que requieren agregar datos de diferentes orígenes. Trino es una parte clave de stacks populares como AWS Athena, Starburst y otras plataformas de datos protegidos. Nuestros equipos lo han utilizado con éxito en varios casos de uso, y cuando se trata de consultar conjuntos de datos de múltiples fuentes para análisis, Trino ha sido una opción confiable.

25. ABsmartly

Probar

ABsmartly es una plataforma avanzada de pruebas A/B, diseñada para tomar decisiones rápidas y confiables. Su característica más destacada es el motor de Pruebas Secuenciales Grupales (GST), que acelera los resultados de las pruebas hasta un 80% comparado con herramientas tradicionales de pruebas A/B. La plataforma ofrece informes en tiempo real, segmentación profunda de datos e integración completa y fluida en todo el ecosistema tecnológico mediante un enfoque API-first, lo que permite realizar experimentos en web, aplicaciones móviles, microservicios y modelos de Machine Learning (ML).

ABsmartly solventa problemas clave en experimentación escalable y basada en datos al permitir una iteración más rápida y un desarrollo de productos más ágil. Su ejecución con latencia cero, capacidad de segmentación profunda y soporte para experimentos en múltiples plataformas la convierten en una herramienta valiosa para organizaciones que buscan expandir su cultura de experimentación y priorizar la innovación basada en datos. Al reducir significativamente los ciclos de prueba y automatizar el análisis de resultados, ABsmartly nos ayudó a optimizar características y experiencias de usuario de manera más eficiente que las plataformas de pruebas A/B tradicionales.

26. Dapr

Probar

Dapr ha evolucionado significativamente desde la última vez que lo destacamos en nuestro Radar. Sus numerosas nuevas funcionalidades incluyen programación de tareas, actores virtuales, políticas de reintento más sofisticadas y componentes de observabilidad. Su catálogo de componentes sigue expandiéndose con nuevas capacidades como gestión de tareas, criptografía y mucho más. Nuestros equipos también destacan su creciente enfoque en configuraciones predeterminadas de seguridad, con soporte para mTLS e imágenes distroless. En general, hemos quedado satisfechos con Dapr y esperamos con interés su evolución futura.

27. Grafana Alloy

Probar

Anteriormente conocido como Grafana Agent, Grafana Alloy es un colector de código abierto de OpenTelemetry Alloy está diseñado para ser un colector de telemetría todo en uno para todos los datos de telemetría, incluyendo logs, métricas y trazas. Admite la recopilación de formatos de datos de telemetría comúnmente utilizados, como OpenTelemetry, Prometheus y Datadog. Con la reciente deprecación de Promtail, Alloy está emergiendo como la opción ideal para la recopilación de datos de telemetría, especialmente para logs, al usar la stack de observabilidad de Grafana.

28. Grafana Loki

Probar

Grafana Loki es un sistema de agregación de logs multi-tenant, altamente disponible y escalable horizontalmente, inspirado en Prometheus. Loki solo indexa metadatos sobre los logs como un conjunto de etiquetas para cada flujo de logs. Los datos de logs se almacenan en una solución de almacenamiento de bloques, como S3, GCS o Azure Blob Storage. El resultado que Loki promete es una reducción en la complejidad operativa y en los costos de almacenamiento en comparación con sus competidores. Como era de esperarse, se integra estrechamente con Grafana y Grafana Alloy, aunque se pueden usar otros mecanismos de recolección.

Loki 3.0 introdujo soporte nativo para OpenTelemetry, lo que hace que la ingesta y la integración con sistemas OpenTelemetry sea tan simple como configurar un endpoint. También ofrece características avanzadas de multi-tenant, como el aislamiento de tenants mediante shuffle-sharding, lo que evita que los tenants con comportamiento anómalo (por ejemplo, consultas pesadas o caídas) afecten a otros en un cluster. Si no has estado siguiendo los desarrollos en el ecosistema de Grafana, ahora es un buen momento para revisarlo, ya que está evolucionando rápidamente.

29. Grafana Tempo

Probar

Grafana Tempo es un backend de rastreo distribuído a gran escala, compatible con estándares abiertos como OpenTelemetry. diseñado para ser eficiente en cuanto a costes. Se basa en el almacenamiento de objetos para retener las trazas a largo plazo, lo que permite buscarlas, generar métricas span-based la correlación con logs y métricas. Por defecto, Tempo utiliza un formato de bloque columnar basado en Apache Parquet, lo que mejora el rendimiento de las consultas y permite a otras herramientas acceder a los datos de rastreo. Las consultas se ejecutan a través de TraceQL y Tempo CLI. También se puede configurar Grafana Alloy para recopilar y enviar trazas a Tempo. Nuestros equipos alojaron internamente Tempo en GKE, utilizando MinIO para el almacenamiento de objetos, recolectores de OpenTelemetry y Grafana para la visualización de las trazas.

30. Railway

Probar

Heroku solía ser una excelente opción para muchos developers que querían lanzar y desplegar sus aplicaciones rápidamente. En los últimos años, también hemos visto el auge de plataformas de despliegue como Vercel, que son más modernas, ligeras y fáciles de usar, pero diseñadas para aplicaciones front-end. Una alternativa full-stack en este ámbito es Railway, una plataforma en la nube tipo PaaS que simplifica todo, desde el despliegue con GitHub/Docker hasta la monitorización en producción.

Railway es compatible con la mayoría de los frameworks de programación más utilizados, bases de datos y despliegue mediante contenedores. Como plataforma de alojamiento a largo plazo para una aplicación, es importante comparar los costes entre diferentes plataformas de forma detallada. Actualmente, nuestro equipo tiene una buena experiencia con el despliegue y la observabilidad en Railway. Su operación es fluida y se integra bien con las prácticas de despliegue continuo que promovemos.

31. Unblocked

Probar

Unblocked es un asistente de IA para equipos. listo para su uso. Una vez integrado con repositorios de código, plataformas de documentación corporativa, herramientas de gestión de proyectos y herramientas de comunicación, Unblocked ayuda a responder preguntas complejas sobre conceptos técnicos y de negocio, diseño e implementación arquitectónica, así como procesos operativos. Esto es particularmente útil para navegar sistemas heredados (legacy) o muy grandes. Durante el uso de Unblocked, hemos observado que los equipos valoran el acceso rápido a la información contextual por encima de la generación de código e historias de usuario; para tales escenarios, especialmente los asistentes de programación, asistentes de ingeniería de software están mejor preparados.

32. Weights & Biases

Probar

Weights & Biases ha seguido evolucionando, añadiendo más funcionalidades centradas en LLM desde su última aparición en el Radar. Están ampliando Traces e introduciendo Weave, una plataforma completa que va más allá del seguimiento de sistemas basados en agentes LLM. Weave te permite crear evaluaciones de sistemas, definir métricas personalizadas, usar los LLMs como jueces para tareas como hacer resúmenes y guardar conjuntos de datos que capturan diferentes comportamientos para su análisis. Esto ayuda a optimizar los componentes LLM y a hacer un seguimiento del rendimiento tanto a nivel local como global. La plataforma también facilita el desarrollo iterativo y la depuración efectiva de sistemas basados en agentes, donde los errores pueden ser difíciles de detectar. Además, permite la recopilación de la valiosa retroalimentación humana, que puede utilizarse posteriormente para reajustar los modelos.

33. Arize Phoenix

Evaluar

Con la creciente popularidad de los modelos de lenguaje extenso (LLM) y de las aplicaciones impulsadas por agentes, la observabilidad de estos modelos cobra cada vez mayor relevancia. En el pasado, hemos recomendado plataformas como Langfuse y Weights & Biases (W&B). Arize Phoenix representa otra plataforma emergente en este ámbito, con la cual nuestro equipo ha tenido una experiencia positiva. Ofrece funciones estándar como seguimiento de modelos (LLM

tracing), evaluación y gestión de prompts, además de una integración ifluida con los principales proveedores y frameworks de LLM. Esto facilita la recopilación de información sobre la salida de los modelos, la latencia y el uso de tokens con una configuración mínima. Hasta el momento, nuestra experiencia se limita a la herramienta de código abierto, pero la plataforma más amplia de Arize ofrece funcionalidades más completas. Confiamos en poder explorar en mayor detalle en el futuro.

34. Chainloop

Evaluar

Chainloop es una plataforma para cadena de suministros de seguridad de código abierto que ayuda a los equipos de seguridad a hacer cumplir sus políticas y a la vez permite a los equipos de desarrollo su fácil integración con los pipelines de CI/CD. Consiste en un plano de control que actúa como única fuente de verdad para las políticas de seguridad, y una interfaz de comandos (CLI) que ejecuta certificaciones dentro de los flujos de trabajo de CI/CD para asegurar el cumplimiento. Los equipos de seguridad definen contratos de flujos de trabajo especificando qué artefactos — como los SBOMs o los reportes de vulnerabilidades — deben ser recogidos, dónde almacenarlos y como evaluar el cumplimiento de las políticas. Chainloop usa Rego la política de lenguaje OPA's, para validar certificaciones — por ejemplo, asegurando un estándar CycloneDX SBOM que cumpla con los requisitos de versión. Durante la ejecución del flujo de trabajo, artefactos de seguridad como SBOMs son adjuntados a una certificación y subidos al plano de control para su aplicación y auditoría. Esta estrategia asegura el cumplimiento de las políticas y puede ser impuesta de manera consistente y a escala a la vez que minimiza la fricción en los flujos de desarrollo. El resultado es un SLSA de nivel tres de conformidad de fuente de verdad única para metadatos, artefactos y certificaciones.

35. Deepseek R1

Evaluar

DeepSeek-R1 es la primera generación de modelos de razonamiento. de DeepSeek. A través de una progresión de modelos no basados en razonamiento, los ingenieros de DeepSeek diseñaron y utilizaron métodos para maximizar la utilización del hardware. Estos incluyen Multi-Head Latent Attention (MLA), Mixture of Experts (MoE) gating, 8-bit floating points training (FP8) y low-level PTX programming. Su enfoque de co-diseño de computación de alto rendimiento permite a DeepSeek-R1 competir con los modelos de vanguardia a un costo significativamente reducido para el entrenamiento y la inferencia. DeepSeek-R1-Zero destaca también por otra innovación: los ingenieros han podido obtener capacidades de razonamiento a partir de un modelo no basado en razonamiento utilizando un simple aprendizaje por refuerzo, sin necesidad de ajustes finos supervisados. Todos los modelos de DeepSeek son de open-weight, lo que significa que están disponibles gratuitamente, aunque el código de entrenamiento y los datos siguen siendo propietarios. El repositorio incluye seis modelos densos destilados de DeepSeek-R1, basados en Llama y Qwen, con DeepSeek-R1-Distill-Qwen-32B superando a OpenAI-o1-mini en varios puntos de referencia.

36. Deno

Evaluar

Creado por Ryan Dahl, el inventor de Node.js, Deno se diseñó para solucionar lo que él consideraba errores de Node.js. Cuenta con un sistema de sandboxing más estricto, gestor de dependencias integrado y compatibilidad nativa con Typescript, un aspecto clave para su base de usuarios. Preferimos Deno para proyectos de Typescript, ya que se siente un verdadero entorno de ejecución y conjunto de herramientas de Typescript, en lugar de un complemento de Node.js.

Desde su inclusion en the Radar in 2019, Deno ha realizado significativos avances, El lanzamiento de Deno 2 introduce la retrocompatibilidad con Node.js y librerías npm, versiones de soporte a largo plazo (LTS) y otras mejoras. Antes, una de las mayores barreras para Adoptar Node.js era la necesidad de reescribir las aplicaciones. Estas actualizaciones reducen la complejidad de la migración al tiempo que amplían las opciones de dependencia para herramientas y sistemas de apoyo. Dado el enorme ecosistema de Node.js y npm, estos cambios deberían impulsar una mayor aceptación.

Además, la biblioteca estándar de Deno se ha estabilizado, ayudando a combatir la proliferación de paquetes npm de bajo valor en todo el ecosistema. Sus herramientas y su biblioteca estándar hacen que TypeScript o JavaScript resulten más atractivos para el desarrollo del lado del servidor. Sin embargo, advertimos que no se debe elegir una plataforma únicamente para evitar programación políglota.

37. Graphiti

Evaluar

Graphiti crea grafos de conocimiento dinámicos y con conciencia temporal que capturan hechos y relaciones en evolución. Nuestros equipos utilizan GraphRAG para descubrir relaciones en los datos, lo que mejora la precisión en la recuperación y las respuestas. Dado que los conjuntos de datos evolucionan constantemente, Graphiti mantiene metadatos temporales en los bordes del grafo para registrar los ciclos de vida de las relaciones. Almacena datos estructurados y no estructurados en forma de episodes discretos y admite consultas mediante una fusión de algoritmos temporales, de texto completo, semánticos y de grafos. Para aplicaciones basadas en LLM — ya sea RAG o agentic — Graphiti permite la recuperación a largo plazo y el razonamiento basado en estados.

38. Helicone

Evaluar

Similar a Langfuse, Weights & Biases y Arize Phoenix, Helicone es una plataforma administrada de LLMOps diseñada para satisfacer la creciente demanda empresarial de gestión de costos de LLM, evaluación de ROI y mitigación de riesgos. De código abierto y centrada en developers, Helicone admite aplicaciones de IA listas para producción, ofreciendo experimentación con prompts, monitorización, depuración y optimización a lo largo de todo el ciclo de vida de los LLM. Permite el análisis en tiempo real de costos, utilización, rendimiento y trazas de pila de los agentes a través de varios proveedores de LLM. Aunque simplifica la gestión de operaciones de LLM, la plataforma aún está en desarrollo y puede requerir cierta experiencia para aprovechar completamente sus funciones avanzadas. Nuestro equipo la ha estado utilizando con una buena experiencia hasta ahora.

39. Humanloop

Evaluar

Humanloop es una plataforma emergente que busca hacer que los sistemas de IA sean más confiables, adaptables y alineados con las necesidades de las personas usuarias, integrando retroalimentación humana en puntos clave de decisión. Ofrece herramientas para el etiquetado humano, aprendizaje activo y ajuste fino con intervención humana, así como la evaluación de modelos de lenguaje (LLM) en función de los requisitos del negocio. Además, facilita la gestión rentable del ciclo de vida de las soluciones de IA generativa con mayor control y eficiencia. Humanloop promueve la colaboración a través de un espacio de trabajo compartido, gestión de prompts con control de

versiones e integración CI/CD para prevenir regresiones. También incluye funciones de observabilidad, como trazabilidad, logs, alertas y límites de seguridad para monitorear y optimizar el rendimiento de la IA. Estas capacidades hacen que Humanloop sea especialmente relevante para organizaciones que implementan IA en entornos regulados o de alto riesgo, donde la supervisión humana es clave. Con su enfoque en prácticas de IA responsable, Humanloop es una opción que vale la pena considerar para equipos que buscan construir sistemas de IA escalables y éticos.

40. Model Context Protocol (MCP)

Evaluar

Uno de los mayores desafíos relacionados al prompting (o interacción con IA a través de lenguaje natural) es garantizar que la herramienta de IA tenga acceso a todo el contexto relevante para la tarea. Frecuentemente este contexto existe dentro de los sistemas que utilizamos en nuestro día a día: wikis, aplicaciones de gestión de tareas, bases de datos o sistemas de observabilidad. La integración fluida entre las herramientas de IA y estas fuentes de información puede mejorar significativamente la calidad de los resultados generados por la IA.

El Model Context Protocol (MCP), un estándar abierto publicado por Anthropic, proporciona un marco de trabajo estandarizado para integrar aplicaciones LLM con fuentes de datos y herramientas externas. Define clientes y servidores MCP, donde los servidores acceden a las fuentes de datos y los clientes integran y usan estos datos para mejorar las solicitudes. Muchos asistentes de código ya han implementado la integración de MCP, permitiéndoles actuar como clientes MCP. Los servidores MCP pueden ejecutarse de dos maneras: localmente, como procesos Python o Node que se ejecutan en la máquina del usuario, o de forma remota, como un servidor al que el cliente MCP se conecta vía SSE (aunque todavía no hemos visto ningún uso de la variante de servidor remoto). Actualmente, MCP se usa principalmente de la primera manera, con developers(as) clonando implementaciones open-source del servidor. Si bien los servidores locales ofrecen una forma sencilla de evitar dependencias de terceros, siguen siendo menos accesibles para usuarios no técnicos e introducen desafíos tales como la gestión de actualizaciones y la gobernanza. Dicho esto, resulta sencillo imaginar cómo este estándar podría evolucionar hacia un ecosistema más maduro y accesible para sus usuarios en el futuro.

41. Open WebUI

Evaluar

Open WebUI es una plataforma de IA de código abierto y auto hospedada con un versátil conjunto de características. Soporta APIs compatibles con OpenAI y se integra con proveedores como OpenRouter y GroqCloud, entre otros. Puede ejecutarse completamente sin conexión conectándose a modelos locales o auto-hospedados a través de Ollama. Open WebUI incluye una capacidad integrada para RAG, lo que permite a los usuarios interactuar con documentos locales y web mediante chat. Ofrece controles RBAC granulares, habilitando diferentes modelos y capacidades de plataforma para diferentes grupos de usuarios. La plataforma es extensible a través de Funciones — bloques de construcción basados en Python que personalizan y mejoran sus capacidades. Otra característica clave es la evaluación de los modelos, que incluye un modo competitivo que permite comparar LLMs uno frente a otro en tareas específicas. Open WebUI puede ser desplegado a diversas escalas — como un asistente personal de IA, un asistente de colaboración en equipo o una plataforma de IA a nivel empresarial.

42. pg_mooncake

Evaluar

pg_mooncake es una extensión de PostgreSQL que añade almacenamiento orientado a columnas y ejecución vectorizada. Las tablas en formato columnar se almacenan como tablas Iceberg o Delta Lake en el sistema de archivos local o en cualquier almacenamiento en la nube compatible con S3. pg_mooncake tiene soporte para cargar datos desde formatos de archivo como Parquet, CSV o incluso conjuntos de datos de Hugging Face. Puede ser una buena opción para análisis de datos masivos que normalmente requieren almacenamiento columnar, ya que elimina la necesidad de incorporar tecnologías de almacenamiento columnar dedicadas en tu infraestructura.

43. Modelos de razonamiento

Evaluar

Uno de los avances más significativos en IA desde el último Radar es el descubrimiento y proliferación de los modelos de razonamiento. También comercializados como modelos de pensamiento, estos modelos han alcanzado un rendimiento a nivel humano en benchmarks como matemáticas avanzadas y programación.

Los modelos de razonamiento suelen entrenarse mediante aprendizaje por refuerzo o fine-tuning supervisado, mejorando capacidades como el raciocinio paso a paso (CoT), la exploración de alternativas (ToT) o la auto-corrección. Algunos ejemplos incluyen o1/o3 de OpenAI, DeepSeek R1 y Gemini 2.0 Flash Thinking. Sin embargo, estos modelos deben considerarse una categoría propia de LLMs en lugar de simples versiones más avanzadas.

Estas prestaciones mejoradas tienen un costo. Los modelos de razonamiento requieren un tiempo de respuesta y consumo de tokens mayor, lo que ha llevado a llamarlos de manera jocosaIA más lenta, del inglés “Slower AI” (como si la IA actual no fuera ya lo suficientemente lenta). No todas las tareas justifican este sacrificio. Para tareas más simples como la sumarización de texto, generación de contenido o chatbots de respuesta rápida, los LLMs de propósito general siguen siendo la mejor opción. Recomendamos usar modelos de razonamiento en campos del STEM, y en resolución de problemas complejos y toma de decisiones como, por ejemplo, usando LLMs como jueces o para mejorar la explicabilidad mediante salidas explícitas de CoT. Al momento de escribir esto, Claude 3.7 Sonnet, un modelo híbrido de razonamiento, acaba de ser lanzado, adelantando una posible fusión entre los LLMs tradicionales y los modelos de razonamiento.

44. Restate

Evaluar

Restate es una plataforma de ejecución duradera, similar a Temporal, desarrollada por los creadores originales de Apache Flink. Entre sus características, ofrece flujos de trabajo como código, procesamiento de eventos con estado, el patrón saga y máquinas de estado duraderas. Escrito en Rust y desplegado como un único binario, utiliza un registro distribuido para almacenar eventos, implementado mediante un algoritmo de consenso virtual basado en Flexible Paxos; lo que garantiza durabilidad incluso en caso de fallo de un nodo. Los SDKs están disponibles para los lenguajes más comunes: Java, Go, Rust y TypeScript. Seguimos sosteniendo que es mejor evitar transacciones distribuidas en sistemas distribuidos, debido tanto a la complejidad adicional como al inevitable sobrecoste operativo que implican. Sin embargo, esta plataforma merece ser evaluada si en tu entorno no puedes evitar el uso de transacciones distribuidas.

45. Supabase

Evaluar

Supabase es una alternativa de código abierto a Firebase para construir backends escalables y seguros. Ofrece un conjunto de servicios integrados, incluyendo una base de datos PostgreSQL, autenticación, APIs instantáneas, funciones Edge, suscripciones en tiempo real, almacenamiento y representaciones vectoriales. Supabase tiene como objetivo agilizar el desarrollo back-end, permitiendo a developers permitiendo a developers concentrarse en la creación de front-end mientras aprovechan el poder y la flexibilidad de tecnologías de código abierto. A diferencia de Firebase, Supabase está construido sobre PostgreSQL. Si estás trabajando en un prototipo o MVP, deberías considerar usar Supabase ya que será más sencillo migrar a otra solución en SQL después de la fase de prototipado.

46. Synthesized

Evaluar

Un desafío común en el desarrollo de software es la generación de datos de prueba para entornos de desarrollo y prueba. Idealmente, estos datos deberían asemejarse lo más posible a los de producción, asegurando al mismo tiempo que no se exponga información personal identificable ni datos sensibles. Aunque esto puede parecer sencillo, la generación de datos de prueba está lejos de serlo. De ahí el motivo de nuestro interés en Synthesized — una plataforma que permite enmascarar y crear subconjuntos de datos de producción existentes o generar datos sintéticos estadísticamente relevantes. Se integra directamente en los procesos de compilación y ofrece enmascaramiento de privacidad, proporcionando anonimización a nivel de atributo mediante técnicas irreversibles de ofuscación de datos como hashing, substitución aleatoria o descarte. Synthesized también puede generar grandes volúmenes de datos sintéticos para pruebas de carga. Aunque incluye las funcionalidades esperadas de GenAI, su propuesta central aborda un desafío real y persistente para los equipos de desarrollo, lo que la convierte en una opción que vale la pena explorar.

47. Tonic.ai

Evaluar

Tonic.ai forma parte de una tendencia creciente en plataformas diseñadas para generar datos sintéticos realistas y anonimizados para entornos de desarrollo, pruebas y control de calidad. Similar a Synthesized, Tonic.ai es una plataforma con un conjunto completo de herramientas que abordan diversas necesidades de síntesis de datos, en contraste con el enfoque basado en bibliotecas de Synthetic Data Vault. Tonic.ai genera datos tanto estructurados como no estructurados, manteniendo las propiedades estadísticas de los datos de producción al mismo tiempo que garantiza la privacidad y el cumplimiento mediante técnicas de privacidad diferencial. Las características clave incluyen la detección, clasificación y redacción automática de información sensible en datos no estructurados, junto con el aprovisionamiento de bases de datos bajo demanda a través de Tonic Ephemeral. También ofrece Tonic Textual, un lago de datos seguro que ayuda a developers de IA a aprovechar los datos no estructurados para sistemas de generación mejorada por recuperación (RAG) y el ajuste fino de LLM. Los equipos que buscan acelerar la velocidad de ingeniería mientras generan datos escalables y realistas — todo ello cumpliendo con estrictos requisitos de privacidad de datos — deberían considerar evaluar Tonic.ai.

48. turbopuffer

Evaluar

turbopuffer es un motor de búsqueda serverless y multi-tenant que integra a la perfección búsquedas vectoriales y de texto sobre almacenamiento de objetos. Nos gusta su arquitectura y sus decisiones de diseño, especialmente su foco en la durabilidad, escalabilidad y eficiencia de coste. Gracias a su uso del almacenamiento de objetos como log de escritura anticipada mientras mantiene sus nodos de consulta sin estado, es adecuado para cargas de búsqueda a gran escala.

Diseñado para el rendimiento y la precisión, turbopuffer ofrece una alta exhaustividad de forma predeterminada, incluso en búsquedas complejas basadas en filtros. Almacena en caché los resultados de consultas en SSD NVMe y mantiene en memoria los namespaces de acceso frecuente, permitiendo búsquedas de baja latencia sobre miles de millones de documentos. Esto lo hace ideal para la recuperación de documentos de gran escala búsqueda vectorial y generación mejorada por recuperación (RAG) de aplicaciones IA. Sin embargo, su dependencia en el almacenamiento de objetos introduce algunas desventajas en la latencia de consultas, haciéndolo más efectivo para cargas de trabajo que se benefician de la computación distribuida y sin estado. turbopuffer potencia sistemas de producción de gran escala como Cursor pero actualmente solo está disponible a través de referencia o invitación.

49. VectorChord

Evaluar

VectorChord es una extensión de PostgreSQL para búsquedas vectoriales por similaridad (vector similarity search), desarrollada por los creadores del pgvecto.rs como su sucesora. Es de código abierto, compatible con los tipos de datos de pgvector y diseñada para búsquedas vectoriales de alto rendimiento, eficientes en disco. Utiliza IVF (Inverted File Index - Indexado Invertido de Archivos) junto con la cuantificación de RaBitQ para permitir una búsqueda vectorial rápida, escalable y precisa mientras reduce significativamente la carga computacional. Como las demás extensiones de PostgreSQL en este ámbito, aprovecha el ecosistema PostgreSQL, permitiendo la búsqueda vectorial junto con operaciones transaccionales estándar. Aunque aún se encuentra en sus primeros pasos, vale la pena evaluar VectorChord para cargas de trabajo de búsqueda vectorial.

50. Gestión de API híbridas con Tyk

Resistir

Hemos observado múltiples equipos han tenido problemas con el gestor de API híbridas Tyk Si bien la idea de contar con un plano de control gestionado y planos de datos autogestionados ofrece flexibilidad para la configuración de infraestructura compleja (como el multi-cloud y de nube híbrida), los equipos han experimentado incidentes en el plano de control que fueron descubiertos internamente en lugar de por Tyk, demostrando posibles brechas de observabilidad en los ambientes alojados en AWS de Tyk. Es más, el nivel de soporte ante incidentes parece ser lento; la comunicación mediante tickets y correos electrónicos no es lo ideal en estas situaciones. Los equipos también han reportado problemas con la madurez de la documentación de Tyk, considerándolo a menudo inadecuado para escenarios y problemas complejos. Además, otros productos del ecosistema de Tyk que también parecen inmaduros, por ejemplo, se ha reportado que el portal para developers empresariales no es compatible con versiones anteriores y tiene capacidades de personalización limitadas. Especialmente en el caso de la configuración híbrida de Tyk, recomendamos proceder con cautela y continuaremos monitoreando su madurez.

Herramientas

Adoptar

51. Renovate
52. uv
53. Vite

Probar

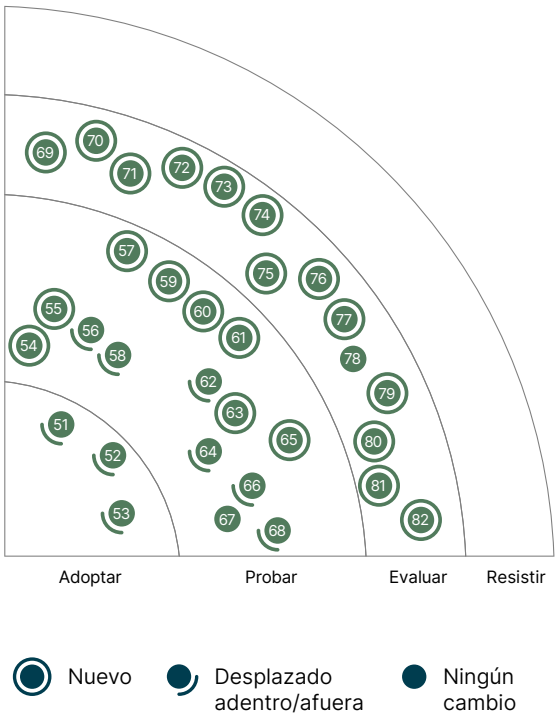
54. Claude Sonnet
55. Cline
56. Cursor
57. D2
58. Databricks Delta Live Tables
59. JSON Crack
60. MailSlurp
61. Metabase
62. NeMo Guardrails
63. Nyx
64. OpenRewrite
65. Plerion
66. Agentes de Ingeniería de Software
67. Tuple
68. Turborepo

Evaluar

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

Resistir

—



51. Renovate

Adoptar

Renovate se ha convertido en la herramienta preferida por muchos de nuestros equipos que buscan Adoptar un enfoque proactivo en la gestión de versiones de dependencias. Mientras Dependabot sigue siendo una opción segura por defecto para los repositorios alojados en GitHub, seguimos recomendando evaluar Renovate como una solución más completa y personalizable. Para maximizar los beneficios, configurar Renovate para monitorear y actualizar todas las dependencias, incluidas las herramientas, la infraestructura y las dependencias privadas o alojadas internamente. Para reducir la sobrecarga de las desarrolladoras, considerar la fusión automática de solicitudes de actualización de dependencias.

52. uv

Adoptar

Desde el último Radar, hemos ganado más experiencia con uv, y el feedback de los equipos han sido abrumadoramente positivos. uv es una herramienta de gestión de paquetes y proyectos de Python de la próxima generación escrita en Rust, con una propuesta de valor clave: es “extremadamente rápida”. Supera a otros gestores de paquetes de Python por un amplio margen en los benchmarks, acelerando los ciclos de construcción y pruebas y mejorando significativamente la experiencia del desarrollador. Más allá del rendimiento, uv ofrece un conjunto de herramientas unificado, reemplazando efectivamente herramientas como Poetry, pyenv y pipx. Sin embargo, nuestras preocupaciones sobre las herramientas de gestión de paquetes siguen siendo las mismas: un ecosistema fuerte, una comunidad madura y soporte a largo plazo son cruciales. Dado que uv es relativamente nuevo, moverlo al anillo de Adopción es atrevido. Sin embargo, muchos equipos de datos están ansiosos por abandonar el sistema heredado de gestión de paquetes de Python, y nuestros developers más cercanos al trabajo diario recomiendan consistentemente a uv como la mejor herramienta disponible hoy en día.

53. Vite

Adoptar

Desde la última vez que Vite fue mencionado en el Radar, ha cobrado aún más fuerza. Se trata de una herramienta de alto rendimiento para la construcción de front-end con un rápido hot-reloading. Está siendo Adoptarado y recomendado como la elección predeterminada de muchos frameworks de front-end, incluyendo Vue, SvelteKit y React, que recientemente deprecated create-react-app. Vite también recibió recientemente una importante inversión, lo que llevó a la fundación de VoidZero, una organización dedicada al desarrollo de Vite. Esta inversión debería acelerar el desarrollo y reforzar la sostenibilidad del proyecto a largo plazo.

54. Claude Sonnet

Probar

Claude Sonnet es un modelo de lenguaje avanzado que destaca en programación, redacción, análisis y procesamiento visual. Está disponible en navegadores, la terminal, la mayoría de los principales IDE e incluso se integra con GitHub Copilot. Hasta la fecha, las pruebas de rendimiento muestran que supera a los modelos anteriores con las versiones 3.5 y 3.7, incluyendo modelos previos de Claude. También es hábil en la interpretación de gráficos y la extracción de texto de imágenes, y ofrece una experiencia centrada en developers, con funciones comoArtifacts en la interfaz del navegador, que permite generar e interactuar con contenido dinámico como fragmentos de código y diseños en HTML.

Hemos utilizado la versión 3.5 de Claude Sonnet en el desarrollo de software y hemos encontrado que mejora significativamente la productividad en diversos proyectos. Destaca especialmente en proyectos que surgen desde cero, en particular para el diseño colaborativo de software y discusiones de arquitectura. Si bien aún es prematuro considerar a cualquier modelo de IA comoestable como asistente en programación, Claude Sonnet es uno de los modelos más fiables con los que hemos trabajado. En el momento de redactar este texto, también se ha lanzado Claude 3.7, con resultados prometedores, aunque aún no lo hemos probado en producción.

55. Cline

Probar

Cline es una extensión de código abierto para VSCode que actualmente es uno de los competidores más fuertes en el espacio de los agentes de ingeniería de software. supervisados. Permite a developers dirigir su implementación completamente desde el chat de Cline, integrándose sin problemas con el IDE que ya utilizan. Características clave, como el modo Plan & Act, el uso transparente de tokens y la integración de MCP ayudan a developers a interactuar de manera efectiva con los LLMs. Cline ha demostrado capacidades avanzadas en el manejo de tareas de desarrollo complejas, especialmente con Claude 3.5 Sonnet. Soporta grandes bases de código, automatiza las pruebas de navegadores headless y corrige errores de forma proactiva. A diferencia de soluciones basadas en la nube, Cline intensifica la privacidad al almacenar datos localmente. Su naturaleza de código abierto no solo asegura una mayor transparencia, sino que también permite mejoras impulsadas por la comunidad. Sin embargo, developers deben ser conscientes del costo del uso de tokens, ya que la orquestación de contexto de código de Cline, aunque muy efectiva, es intensiva en recursos. Otro cuello de botella potencial es el rate limiting, que puede ralentizar los flujos de trabajo. Hasta que esto se resuelva, es aconsejable utilizar proveedores de API como OpenRouter, que ofrecen mejores reductores de tráfico.

56. Cursor

Probar

Continuamos impresionados por el editor de código basado en Inteligencia Artificial Cursor, el cual sigue siendo líder en el competitivo espacio de la asistencia de código con IA. Su orquestación del contexto del código es muy efectiva y admite una amplia gama de modelos, incluyendo la opción de utilizar una clave de API personalizada. El equipo de Cursor a menudo introduce características innovadoras de experiencia de usuario antes que otros proveedores e incluye una lista extensa de proveedores de contexto en su chat, como la referencia a diferencias en git, conversaciones previas con la IA, búsquedas web, documentación de librerías e integración con MCP. Junto con herramientas como Cline y Windsurf, Cursor se destaca por su poderoso modo de agente de codificación. Este modo permite a developers guiar su implementación directamente desde una interfaz de chat basada en IA, donde la herramienta lee y modifica archivos de forma autónoma, así como también ejecuta comandos. También valoramos la capacidad de Cursor para detectar errores de linting y de compilación en el código generado, y corregirlos proactivamente.

57. D2

Probar

D2 es una herramienta de código abierto de tipo diagramas como código que ayuda a los usuarios a crear y personalizar diagramas a partir de texto. Introduce el lenguaje de scripting de diagramas D2, que prioriza la legibilidad frente a la compacidad con una sintaxis simple y declarativa. D2 incluye por defecto un tema y utiliza el mismo motor de diseño que Mermaid. Nuestros equipos valoran su sintaxis ligera, que está diseñada específicamente para documentación de software y diagramas de arquitectura.

58. Databricks Delta Live Tables

Probar

Delta Live Tables (DLT) sigue demostrando su valor a la hora de simplificar y agilizar la gestión de pipelines de datos, soportando tanto el streaming en tiempo real como el procesamiento por batch a través de un enfoque declarativo. Al automatizar tareas complejas de ingeniería de datos, como la gestión manual de puntos de control, DLT reduce la sobrecarga operativa y garantiza un sistema robusto de punta a punta. Su capacidad para orquestar pipelines sencillos con una intervención manual mínima mejora la fiabilidad y flexibilidad, mientras que funciones como las vistas materializadas proporcionan actualizaciones incrementales y optimización del rendimiento para casos de uso específico.

Sin embargo, los equipos deben comprender los matices de DLT para aprovechar plenamente sus ventajas y evitar posibles dificultades. DLT gestiona sus propias tablas y restringe la inserción de datos a un único pipeline simultáneamente. Las tablas de streaming son de adición, lo que requiere consideraciones de diseño cuidadosas. Además, al borrar un pipeline DLT también se borran la tabla y los datos subyacentes, lo que puede crear problemas operativos.

59. JSON Crack

Probar

JSON Crack es una extensión de Visual Studio Code que genera gráficos interactivos a partir de datos en formato textual. A pesar de su nombre, admite múltiples formatos, incluyendo YAML, TOML y XML. A diferencia de Mermaid y D2, donde la representación textual es un medio para crear un gráfico visual específico, JSON Crack es una herramienta para visualizar datos que se encuentran en formato de texto. Su algoritmo de diseño funciona bien y permite ocultar selectivamente ramas y nodos, lo que lo convierte en una excelente opción para explorar conjuntos de datos. También esta disponible una herramienta web complementaria, pero en este caso tenemos dudas sobre depender de servicios online para el formatear o parsear código JSON Crack tiene un límite en la cantidad de nodos a procesar y redirige a los usuarios a una herramienta comercial relacionada para manejar archivos con más de unos pocos cientos de nodos.

60. MailSlurp

Probar

Los flujos de trabajo de pruebas que implican el correo electrónico suelen ser complejos y requieren mucho tiempo. Los equipos de desarrollo deben construir clientes de API de correo electrónico personalizados para la automatización, a la vez que configuran buzones de entrada temporales para escenarios de prueba manual, tales como pruebas de usabilidad o formación interna del producto antes de las liberaciones importantes. Estos desafíos se vuelven aún más pronunciados al desarrollar productos de integración de clientes. Hemos tenido una experiencia positiva con MailSlurp, un servidor de correo y servicio de API SMS. Proporciona API REST para crear buzones de entrada y números de teléfono, así como para validar correos electrónicos y mensajes directamente en el código, y su tablero sin código también es útil para la preparación de pruebas manuales. Otras características, como dominios personalizados, webhooks, respuesta automática y reenvío, merecen ser consideradas para escenarios más complejos.

61. Metabase

Probar

Metabase es una herramienta de análisis e inteligencia de negocio de código abierto que permite a los usuarios visualizar y analizar datos provenientes de diversas fuentes, incluidas bases de

datos relacionales y NoSQL. La herramienta facilita la creación de visualizaciones e informes, su organización en tableros y el intercambio de información de forma sencilla. Además, ofrece un SDK para incrustar tableros interactivos en aplicaciones web, adaptándose al estilo y tema de la aplicación, lo que resulta muy conveniente para developers. Con conectores de datos respaldados oficialmente por la comunidad, Metabase es versátil en diversos entornos. Como herramienta de BI ligera, nuestros equipos la encuentran útil para gestionar tableros interactivos e informes en sus aplicaciones.

62. NeMo Guardrails

Probar

NeMo Guardrails es un kit de herramientas de código abierto de fácil uso de NVIDIA que permite a developers implementar barreras de protección para modelos de lenguaje de gran tamaño (LLM) utilizados en aplicaciones conversacionales. Desde que lo mencionamos por última vez en el Radar, NeMo ha experimentado una adopción significativa en nuestros equipos y continúa mejorando. Muchas de las últimas mejoras de NeMo Guardrails se centran en expandir las integraciones y fortalecer la seguridad, los datos y el control, alineándose con el objetivo principal del proyecto.

Una actualización importante de NeMo documentation es que ha mejorado la usabilidad y se han añadido nuevas integraciones, incluyendo AutoAlign y Patronus Lynx, junto con soporte para Colang 2.0. Las actualizaciones clave incluyen mejoras en la seguridad y protección del contenido así como una versión reciente que admite la transmisión de contenido LLM a través de rieles de salida para un rendimiento mejorado. También hemos visto soporte adicional para Prompt Security. Además, Nvidia lanzó tres nuevos microservicios: el microservicio NIM de seguridad de contenido, el microservicio NIM de control de temas y la detección de jailbreak, todos los cuales se han integrado con NeMo Guardrails. Debido a su creciente conjunto de características y al aumento de su uso en producción, estamos moviendo NeMo Guardrails a la fase de Probar. Recomendamos revisar las últimas notas de la versión para obtener una visión general completa de los cambios desde nuestro último blip.

63. Nyx

Probar

Nyx Es una herramienta versátil de lanzamiento semántico que admite una amplia gama de proyectos de ingeniería de software. Es independiente del lenguaje y funciona con todas las principales plataformas de CI y SCM, lo que la hace altamente adaptable. Aunque muchos equipos utilizan versionado semántico en el Desarrollo basado en trunk, Nyx también es compatible con flujos de trabajo como Gitflow, OneFlow y GitHub Flow. Una ventaja clave de Nyx en producción es su generación automática de changelogs, con soporte integrado para Comits convencionales.

Como se ha señalado en ediciones anteriores del Radar, advertimos sobre los patrones de desarrollo que dependen de Ramas de larga duración (Por ejemplo, Gitflow, GitOps), ya que introducen desafíos que incluso herramientas potentes como Nyx no pueden mitigar. Recomendamos encarecidamente probar Nyx en flujos de trabajo CI/CD, especialmente en desarrollo basado en trunk, donde hemos visto repetidos casos de éxito.

64. OpenRewrite

Probar

OpenRewrite sigue siendo una herramienta muy útil para refactorizaciones a gran escala que siguen un conjunto de reglas, tal como la transición a una versión más reciente de la API de una librería ampliamente utilizada o la aplicación de actualizaciones a múltiples servicios que fueron creados a partir de la misma plantilla. Se ha introducido soporte para lenguajes más allá de Java, notablemente

JavaScript. Con ciclos de lanzamientos LTS cortos en frameworks como Angular, mantener los proyectos actualizados a versiones más recientes es cada vez más importante. OpenRewrite facilita este proceso de forma eficaz. Utilizar un asistente de codificación basado en IA es una alternativa, pero para cambios basados en reglas, generalmente es más lento, más costoso y menos fiable. Nos gusta que OpenRewrite venga con un catálogo de recetas (reglas), que describen los cambios a realizar. El motor de refactorización, las recetas incluidas y los plugins de herramientas de construcción son software de código abierto, lo que facilita que los equipos recurran a OpenRewrite cuando lo necesiten.

65. Plerion

Probar

Plerion es una plataforma de seguridad en la nube centrada en AWS que se integra con los proveedores de alojamiento para descubrir riesgos, errores de configuración y vulnerabilidades en su infraestructura en la nube, servidores y aplicaciones. Similar a Wiz, Plerion utiliza una priorización basada en riesgos para los problemas detectados, prometiendo permitircentrarse en el 1% que importa". Nuestros equipos informan de experiencias positivas con Plerion, señalando que ha proporcionado a nuestros clientes perspectivas significativas y ha reforzado la importancia del monitoreo proactivo de la seguridad para sus organizaciones.

66. Agentes de Ingeniería de Software

Probar

Desde que escribimos sobre agentes de ingeniería de software hace seis meses, la industria aún carece de una definición compartida del términoagente. Sin embargo, ha surgido un desarrollo importante, no en agentes de codificación completamente autónomos (que siguen sin ser convincentes), sino en modos de agentes supervisados dentro del IDE. Estos modos permiten a developers dirigir la implementación a través de chat, con herramientas que no solo modifican el código en múltiples archivos, sino que también ejecutan comandos, realizan pruebas y responden a la retroalimentación del IDE, como errores de estilo o compilación.

Este enfoque, a veces llamadoprogramación orientada por chat (CHOP por sus siglas en inglés) oprompt-to-code (indicación a código), mantiene el control en developers mientras se transfiere más responsabilidad a la IA que los asistentes de codificación tradicionales, como las sugerencias automáticas. Las herramientas líderes en este espacio incluyen Cursor, Cline y Windsurf, con GitHub Copilot ligeramente rezagado pero alcanzando a los otros rápidamente. La utilidad de estos modos de agentes depende tanto del modelo utilizado (con Claude's Sonnet series siendo actualmente la referencia) como qué tan bien se integra con el IDE para proporcionar una buena experiencia de desarrollo.

Hemos encontrado estos flujos de trabajo interesantes y prometedores, con un notable aumento en la velocidad de codificación. Sin embargo, mantener pequeños los alcances de los problemas ayuda a developers a revisar mejor los cambios generados por la IA. Esto funciona mejor con indicaciones de baja abstracción y bases de código amigables con la IA que estén bien estructuradas y debidamente probadas. A medida que estos modos mejoren, también aumentarán el riesgo de complacencia con el código generado por la IA. Para mitigar esto, recomendamos la programación en pareja y otras prácticas de revisión disciplinadas, especialmente para el código de producción.

67. Tuple

Probar

Tuple, una herramienta optimizada para la programación en pareja de forma remota, fue diseñada originalmente para llenar el vacío que dejó Screenhero de Slack. Desde que la mencionamos en el Radar, ha ganado más adopción, ha corregido problemas y limitaciones y ahora es compatible con Windows. Una mejora clave es la optimización del uso compartido de escritorio con una función de privacidad integrada, que permite ocultar ventanas de aplicaciones privadas (como mensajes de texto) mientras se comparten herramientas como la ventana del navegador. Antes, las limitaciones de la interfaz de usuario hacían que Tuple se sintiera más como una herramienta específica para programación en pareja que como una solución de colaboración general. Con estas actualizaciones, ahora es posible colaborar en más contenidos fuera del IDE.

Sin embargo, es importante destacar que la pareja remota tiene acceso a todo el escritorio. Si no se configura correctamente, esto podría representar un riesgo de seguridad, especialmente si la otra persona no es de confianza.

Recomendamos encarecidamente educar a los equipos sobre la configuración de privacidad, buenas prácticas y código de conducta en Tuple antes de implementarlo. Animamos a los equipos a probar la última versión de Tuple en su flujo de trabajo de desarrollo. Se alinea con nuestra recomendación de programación en pareja remota con pragmatismo ofreciendo baja latencia, una experiencia de usuario intuitiva y mejoras significativas en usabilidad.

68. Turborepo

Probar

Turborepo ayuda a gestionar grandes mono repositorios de JavaScript o TypeScript mediante el análisis, almacenamiento en caché, paralelización y optimización de las tareas de construcción para acelerar el proceso. En mono repositorios grandes, los proyectos suelen depender unos de otros; reconstruir todas las dependencias por cada cambio es ineficiente y consume mucho tiempo, pero Turborepo facilita este proceso. A diferencia de Nx, la configuración predeterminada de Turborepo utiliza múltiples archivos package.json; uno por proyecto, lo que permite tener dependencias con diferentes versiones (múltiples versiones de React, por ejemplo) en un único mono repositorio, algo que Nx desaconseja. Aunque esto podría considerarse un anti-patrón, resuelve ciertos casos de uso, como la migración de múltiples repositorios a uno único, donde los equipos pueden necesitar temporalmente varias versiones de dependencias. En nuestra experiencia, Turborepo es bastante sencillo de configurar y tiene un buen desempeño.

69. AnythingLLM

Evaluar

AnythingLLM es una aplicación de escritorio de código abierto que permite chatear con documentos o fragmentos de contenido extensos, respaldada por una integración lista para usar con modelos de lenguaje de gran tamaño (LLMs) y bases de datos vectoriales. Cuenta con una arquitectura modular para modelos de incrustación y puede utilizarse con la mayoría de los LLM comerciales, así como con modelos de código abierto gestionados por Ollama. Además de RAG, se pueden crear diferentes habilidades y organizarlas como agentes para realizar tareas y flujos de trabajo personalizados.
La aplicación permite a los usuarios organizar documentos e interacciones en distintos espacios de trabajo, que funcionan como hilos de conversación persistentes con diferentes contextos.
Recientemente, también se ha añadido la posibilidad de desplegarlo como una aplicación web multiusuario mediante una imagen de Docker. Algunos de nuestros equipos lo están utilizando como asistente personal local y lo consideran una herramienta potente y útil.

70. Gemma Scope

Evaluar

La interpretabilidad mecánica, el entendimiento del funcionamiento interno de los grandes modelos de lenguaje, se está convirtiendo en un campo cada vez más importante. Herramientas como Gemma Scope y librerías open-source Mishax proporcionan perspectivas sobre la familia Gemma2 de modelos abiertos. Las herramientas de interpretabilidad desempeñan un papel crucial a la hora de depurar comportamientos inesperados, identificando los componentes responsables de alucinaciones, sesgos u otros casos que causan fallos y, en última instancia, generan confianza ofreciendo una visibilidad más detallada de los modelos. Aunque este campo puede ser de especial interés para los investigadores, cabe señalar que con la reciente publicación de DeepSeek-R1, el entrenamiento de modelos es cada vez más factible para las empresas más allá de los actores establecidos. A medida que la GenAI siga evolucionando, tanto la interpretabilidad como la seguridad irán ganando importancia.

71. Hurl

Evaluar

Hurl es una herramienta versátil para realizar secuencias de solicitudes HTTP, definidas en archivos de texto plano utilizando una sintaxis específica de Hurl. Además de enviar solicitudes, Hurl puede validar respuestas, asegurando que una solicitud devuelva un código de estado HTTP específico; comprobar condiciones en los encabezados de respuesta o en el contenido usando XPATH, JSONPath o expresiones regulares; y extraer datos de la respuesta en variables, que luego pueden usarse para encadenar solicitudes.

Gracias a su conjunto de características, Hurl es útil para automatizaciones de API simples, pero también sirve como una herramienta de pruebas de API automatizada. Su capacidad para generar informes de pruebas detallados en formato HTML o JSON mejora su utilidad en flujos de trabajo de pruebas. Aunque herramientas especializadas como Bruno y Postman ofrecen interfaces gráficas de usuario (GUIs) y características adicionales, nos gusta Hurl por su simplicidad. Al igual que Bruno, que también usa archivos de texto plano, las pruebas de Hurl pueden almacenarse en el repositorio de código.

72. Jujutsu

Evaluar

Git es el principal sistema de control de versiones distribuido (VCS), acumulando la vasta mayoría de la cuota de mercado. Aún así, y pese a más de una década de liderazgo, developers aún encuentran dificultades con los complejos flujos de trabajo para la gestión de ramas, la fusión, el rebase y la resolución de conflictos. Esta frustración constante ha dado lugar a una serie de herramientas diseñadas para aliviar este problema; algunas ofreciendo ayudas visuales para aclarar la complejidad, otras a través de interfaces gráficas que abstraen la misma por completo.

Jujutsu va un paso más allá, ofreciendo una alternativa completa a Git, pero manteniendo la compatibilidad al usar repositorios de Git como almacenamiento de backend. Esto permite a developers seguir utilizando servidores y servicios Git existentes mientras se benefician de los flujos de trabajo optimizados de Jujutsu. Posicionado como “simple y poderoso”, Jujutsu enfatiza la facilidad de uso para developers de todos los niveles de experiencia. Una de sus características destacadas es la resolución de conflictos de primera clase, con el potencial de mejorar significativamente la experiencia de desarrollo.

73. kubenetmon

Evaluar

La monitorización y la comprensión del tráfico de red asociado a Kubernetes puede ser desafiante, particularmente cuando tu infraestructura se extiende a múltiples zonas, regiones o nubes. kubenetmon, creado por ClickHouse y recientemente disponible en código abierto, tiene la esperanza de resolver este problema ofreciendo mediciones detalladas del tráfico de datos de Kubernetes a través de los mayores proveedores de servicios en la nube. Si usas Kubernetes y te has frustrado por costos opacos de transferencia de datos en tu factura, puede merecer la pena que explores kubenetmon.

74. Mergiraf

Evaluar

Resolver conflictos de fusión es probablemente una de las actividades menos apreciadas en el desarrollo de software. Y aunque existen técnicas que reducen la complejidad de las fusiones —por ejemplo practicar la integración continua, en el sentido original de fusionar con una rama principal compartida, al menos una vez al día—, vemos que se dedica demasiado esfuerzo a las fusiones. Long-lived feature branches es uno de los culpables, pero la codificación asistida por IA también tiende a incrementar el tamaño de los conjuntos de cambios. La ayuda podría llegar en forma de Mergiraf, una nueva herramienta que resuelve conflictos de fusión analizando el árbol sintáctico en lugar de tratar el código como líneas de texto. Como un controlador de fusión para git, puede configurarse para que subcomandos como `merge` y `cherry-pick` usen automáticamente Mergiraf en lugar de los métodos predeterminados.

75. ModernBERT

Evaluar

El sucesor de BERT (Bidirectional Encoder Representations from Transformers), ModernBERT constituye la nueva generación de modelos transformadores centrados exclusivamente en la fase de codificación diseñados para abordar un amplio espectro de tareas de procesamiento del lenguaje natural (PLN). Como sustituto directo de BERT, ModernBERT no solo incrementa el rendimiento y la precisión, sino que también solventa algunas de sus limitaciones; destacando especialmente su capacidad de manejar longitudes de contexto notablemente mayores gracias a la técnica denominadaAlternating Attention. Para los equipos que requieran soluciones de PLN, resulta recomendable evaluar ModernBERT antes de Adoptar un modelo generativo de propósito general.

76. OpenRouter

Evaluar

OpenRouter es una API unificada para acceder a múltiples modelos de lenguaje extenso. Ofrece un punto de integración único para principales proveedores de LLM, simplificando la experimentación, reduciendo la dependencia de un proveedor y optimizando los costos al enrutar las solicitudes al modelo más apropiado. Herramientas populares como Cline y Open WebUI utilizan OpenRouter como su punto de acceso primario. Durante el análisis del Radar, cuestionamos la necesidad real de alternar entre modelos en la mayoría de los proyectos, dado que OpenRouter debe añadir un margen de precio como modelo de negocio sobre esta capa de encapsulación. Sin embargo, también reconocemos que OpenRouter proporciona diversas estrategias de balanceo de carga para ayudar a optimizar los costos. Una característica particularmente útil es su capacidad para evitar los límites de tasa de las APIs. Si una aplicación excede la tasa límite de un solo proveedor de LLM, OpenRouter puede ayudar a evitar esta restricción y lograr un mejor rendimiento.

77. Redactive

Evaluar

Redactive es una plataforma empresarial de habilitación de IA diseñada para ayudar a organizaciones reguladas a preparar de forma segura datos no estructurados para aplicaciones de inteligencia artificial, como asistentes y copilotos potenciados por IA. Se integra con plataformas de contenido como Confluence, creando índices de texto seguros para búsquedas con generación mejorada por recuperación (RAG). Al servir únicamente datos en vivo y aplicar en tiempo real los permisos de usuario desde los sistemas fuente, Redactive garantiza que los modelos de IA accedan a información precisa y autorizada sin comprometer la seguridad. Además, proporciona a los equipos de ingeniería herramientas para construir casos de uso de IA de forma segura usando cualquier LLM. Para las organizaciones que están explorando soluciones impulsadas por IA, Redactive ofrece un enfoque simplificado para la preparación de datos y el cumplimiento, equilibrando seguridad y accesibilidad para equipos que experimentan con capacidades de IA en un entorno controlado.

78. System Initiative

Evaluar

Seguimos entusiasmados con System Initiative. Esta herramienta experimental representa una radical nueva dirección para el trabajo en DevOps. Nos gusta mucho el pensamiento creativo detrás de esta herramienta y esperamos que motive a otros a romper con el status quo de los enfoques de infraestructura como código. System Initiative ha superado la fase beta, ahora está disponible de forma gratuita y open source bajo licencia Apache 2.0. Aunque sus developers la utilizan para gestionar su infraestructura en producción, aún le falta camino por recorrer antes de poder escalar y satisfacer las demandas de grandes empresas. Sin embargo, seguimos creyendo que vale la pena probarla para experimentar un enfoque completamente diferente a otras herramientas de DevOps.

79. TabPFN

Evaluar

TabPFN es un modelo basado en transformadores, diseñado para una clasificación rápida y precisa de conjuntos pequeños de datos tabulares. Aprovecha el aprendizaje contextual (ICL) para hacer predicciones directamente a partir de ejemplos etiquetados sin ajustar hiperparámetros ni entrenamiento adicional. Pre entrenado en millones de conjuntos de datos sintéticos, TabPFN generaliza bien a través de diversas distribuciones de datos y maneja eficazmente valores ausentes y valores atípicos. Sus fortalezas incluyen procesamiento eficiente de datos heterogéneos y robustez frente a características poco informativas.

TabPFN es especialmente adecuado para aplicaciones a pequeña escala en las que la velocidad y la precisión son cruciales. Sin embargo, se enfrenta a desafíos de escalabilidad con conjuntos de datos más grandes y tiene limitaciones en el manejo de tareas de regresión. Como una solución vanguardista, merece la pena evaluar TabPFN por su potencial para superar a los modelos de clasificación tabular tradicionales, especialmente cuando los transformadores se aplican con menos frecuencia.

80. v0

Evaluar

v0 de Vercel es una herramienta de IA para generar código de interfaz web a partir de una captura de pantalla, diseño de Figma o un simple prompt. Soporta React, Vue, shadcn y Tailwind entre otros frameworks de interfaz web. Más allá del código generado por IA, v0 proporciona una gran experiencia de usuario, que incluye la posibilidad de previsualizar el código generado y desplegarlo en Vercel en un solo paso. Aunque la construcción de una aplicación real implica la integración de múltiples funcionalidades más allá de una única pantalla, v0 proporciona una manera sólida de prototipar y puede usarse para inicializar un punto de partida para el desarrollo de aplicaciones complejas.

81. Windsurf

Evaluar

Windsurf es un asistente de código con IA desarrollado por Codeium que destaca por sus habilidades agénticas. Es similar a Cursor y Cline, permitiendo a developers dirigir su implementación a través de un chat con IA que navega y modifica el código, además de ejecutar comandos. Con frecuencia, lanza nuevas funciones e integraciones para su modo agéntico. Recientemente, por ejemplo, ha introducido una vista previa del navegador que facilita el acceso del agente a los elementos del DOM y la consola del navegador, así como una capacidad de investigación web que le permite buscar documentación y soluciones en internet cuando sea necesario. Windsurf ofrece acceso a una variedad de modelos populares y permite a los usuarios activar y referenciar la búsqueda en la web, la documentación de bibliotecas y la integración con MCP como proveedores de contexto adicionales.

82. YOLO

Evaluar

La serie YOLO (You Only Look Once), desarrollada por Ultralytics, sigue avanzando en los modelos de visión por computadora. La última versión, YOLO11, ofrece mejoras significativas tanto en precisión como en eficiencia en comparación con sus versiones anteriores. YOLO11 es capaz de realizar clasificación de imágenes a alta velocidad con recursos mínimos, lo que la hace idónea para aplicaciones en tiempo real en dispositivos periféricos. También, encontramos que la capacidad de utilizar este mismo framework para llevar a cabo la estimación de posturas, detección de objetos, segmentación de imágenes y otras tareas es muy poderosa. Este importante desarrollo también nos recuerda que emplear modelos de aprendizaje automáticos “tradicionales” para tareas específicas puede ser más eficaz que utilizar modelos generales de Inteligencia Artificial, como los LLMs.

Lenguajes y Frameworks

Adoptar

83. OpenTelemetry
84. React Hook Form

Probar

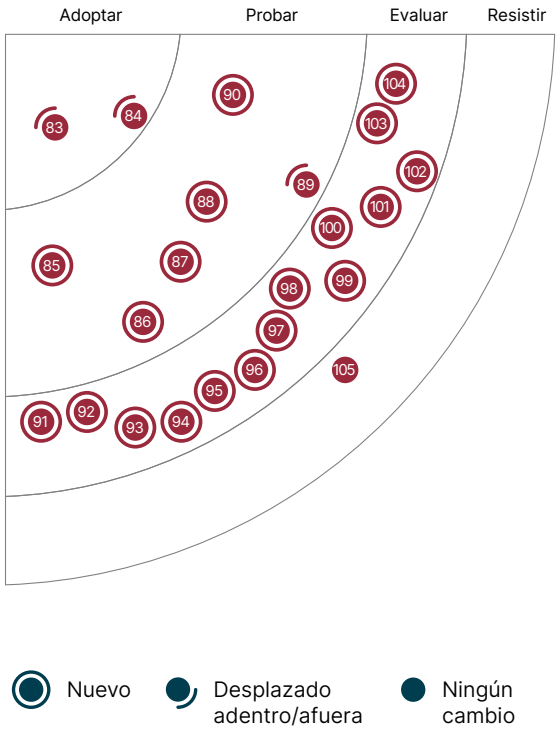
85. Effect
86. Motor GraphQL de Hasura
87. LangGraph
88. MarkItDown
89. Module Federation
90. Prisma ORM

Evaluar

91. .NET Aspire
92. SDK para Android XR
93. Browser Use
94. CrewAI
95. ElysiaJs
96. FastGraphRAG
97. Gleam
98. GoFr
99. Criptografía postcuántica con Java
100. Presidio
101. PydanticAI
102. Swift para aplicaciones con recursos limitados
103. Tamagui
104. torchtune

Resistir

105. Sobrecarga de Node



83. OpenTelemetry

Adoptar

OpenTelemetry se está convirtiendo rápidamente en el estándar de la industria para la observabilidad. El lanzamiento de la especificación del protocolo OpenTelemetry (OTLP) estableció una forma estandarizada de gestionar trazas, métricas y registros, reduciendo la necesidad de múltiples integraciones o grandes reescrituras a medida que aumentan los requisitos de interoperabilidad y las soluciones de monitorización distribuidas. A medida que OpenTelemetry se expande para admitir registros y perfiles, OTLP garantiza un formato de transporte consistente en todos los datos de telemetría, simplificando la instrumentación y haciendo que la observabilidad full-stack sea más accesible y escalable para las arquitecturas de microservicios.

Adoptarado por proveedores como Datadog, Nuevo Relic y Grafana, OTLP permite a las organizaciones crear conjuntos de datos de observabilidad flexibles e independientes de cada proveedor, sin depender de soluciones patentadas o privadas. Admite compresión gzip y zstd, reduciendo el tamaño de los datos de telemetría y el uso de ancho de banda — una ventaja clave para entornos que gestionan grandes volúmenes de datos de telemetría. Diseñado para el crecimiento a largo plazo, OTLP garantiza que OpenTelemetry siga siendo un estándar robusto y preparado para el futuro, consolidando su posición como la opción predilecta para el transporte de telemetría.

84. React Hook Form

Adoptar

Hemos identificado React Hook Form como una alternativa a Formik. Al utilizar componentes no controlados por defecto, ofrece un rendimiento considerablemente superior sin configuración adicional, especialmente para formularios de gran tamaño. React Hook Form está bien integrado con varias librerías de validación basadas en esquemas, incluyendo Yup, Zod y más. Adicionalmente, React Hook Form ofrece mucha flexibilidad, facilitando la integración con código fuentes existentes y otras librerías. Puedes usar React Hook Form con librerías de componentes controlados externas como shadcn o AntD. Con un rendimiento sólido, una integración fluida y un desarrollo activo, es una opción confiable para la creación de aplicaciones con formularios extensos o con una gran cantidad de los mismos.

85. Effect

Probar

Effect es una potente librería de TypeScript para construir complejos programas síncronos y asíncronos. El desarrollo de aplicaciones web a menudo requiere código repetitivo para tareas relacionadas con asincronía, concurrencia, gestión de estados y manejo de errores. Effect-TS agiliza estos procesos utilizando un enfoque de programación funcional. Aprovechando el sistema de tipos de TypeScript, Effect ayuda a encontrar problemas difíciles de detectar en tiempo de compilación. Nuestro equipo utilizaba anteriormente fp-ts para la programación funcional, pero descubrió que Effect-TS proporciona abstracciones que se ajustan más a las tareas diarias. También facilita la combinación y comprobación del código. Mientras que los enfoques tradicionales como `Promise/try-catch` o `async/await` pueden manejar estos escenarios, después de usar Effect, nuestro equipo no encontró ninguna razón para volver atrás.

86. Motor GraphQL de Hasura

Probar

El motor GraphQL de Hasura es una capa universal de acceso a datos que simplifica la creación, ejecución y gestión de APIs de alta calidad en diferentes fuentes de datos. Proporciona APIs GraphQL instantáneas sobre varias bases de datos (incluyendo PostgreSQL, MongoDB y ClickHouse) y fuentes de datos, permitiendo a developers obtener sólo los datos que necesitan de forma rápida y segura. Encontramos que Hasura es un GraphQL fácil de implementar en la agregación de recursos del lado del servidor y lo hemos aplicado en múltiples proyectos de productos de datos. Sin embargo, seguimos siendo cautos con respecto a su potente gestión de consultas federadas y esquemas unificados. Una adición reciente destacable es la función PromptQL de Hasura, que permite a developers aprovechar LLMs para lograr interacciones de datos más naturales e intuitivas.

87. LangGraph

Probar

LangGraph es un framework de orquestación diseñado para construir aplicaciones multiagente con estados utilizando modelos LLM. Esta tecnología proporciona un conjunto de primitivas de bajo nivel como aristas y vértices en lugar de las abstracciones de alto nivel de LangChain's lo que permite a developers un control más detallado sobre los flujos de trabajo de los agentes, la gestión de memoria y la persistencia del estado. Este enfoque basado en grafos garantiza flujos de trabajo predecibles y personalizables, facilitando la depuración, el escalado y el mantenimiento de aplicaciones en producción. Aunque presenta una curva de aprendizaje más pronunciada, su diseño ligero y modular lo convierte en un framework potente para la creación de aplicaciones con agentes autónomos.

88. MarkItDown

Probar

MarkItDown convierte varios formatos (PDF, HTML, PowerPoint, Word) en Markdown, mejorando la legibilidad del texto y manteniendo el contexto. Ya que los grandes modelos de lenguaje derivan el contexto de formatting cues como los encabezados y secciones, Markdown ayuda a preservar la estructura para una mejor comprensión. En las aplicaciones basadas en RAG, nuestros equipos usaron MarkItDown para preprocesar documentos a Markdown, asegurándose que los marcadores lógicos (encabezados, subsecciones) quedaran intactos. Antes de incorporar la generación, la fragmentación consciente de la estructura ayudaba a mantener el contexto de la sección lo cual mejoraba la claridad de las respuestas a las consultas especialmente para documentos complejos. Markdown es ampliamente utilizado para documentación y también convierte la CLI de MarkItDown en una valiosa herramienta de productividad para developers.

89. Module Federation

Probar

Module Federation permite la especificación de módulos compartidos y la deduplicación de dependencias en micro frontends. Con la versión 2.0, ha evolucionado para funcionar de forma independiente de webpack. Esta actualización introduce características clave, incluyendo un tiempo de ejecución de federación, una nueva API de plugins y soporte para frameworks populares como React y Angular, así como otros empaquetadores populares como Rspack y Vite. Al Adoptar Module Federation, las aplicaciones web de gran tamaño pueden dividirse en micro frontends más pequeños y manejables, lo que permite que diferentes equipos desarrollen, desplieguen y escalen de forma independiente, a la vez que comparten dependencias y componentes de manera eficiente.

90. Prisma ORM

Probar

Prisma ORM es una herramienta de código abierto para bases de datos que simplifica el trabajo con éstas en aplicaciones de Node.js y TypeScript. Ofrece un enfoque moderno y de tipado seguro para acceder a bases de datos, automatiza las migraciones de esquemas de bases de datos y proporciona una API de consulta intuitiva. A diferencia de los ORM tradicionales, Prisma ORM utiliza objetos de JavaScript estándar para definir los tipos de base de datos, sin necesidad de decoradores ni clases. Nuestra experiencia con Prisma ORM ha sido positiva; consideramos que no solo se alinea mejor con el ecosistema de desarrollo en TypeScript, sino que también se integra impecablemente con el paradigma de programación funcional.

91. .NET Aspire

Evaluar

.NET Aspire está diseñado para simplificar la orquestación de aplicaciones distribuidas en la máquina local de un desarrollador. Aspire permite orquestar múltiples servicios en un entorno de desarrollo local; incluyendo múltiples proyectos .NET, bases de datos dependientes y contenedores Docker, todo con un solo comando. Además, Aspire proporciona herramientas de observabilidad; incluyendo logs, trazabilidad y paneles de métricas, para el desarrollo local, desacopladas de las herramientas utilizadas en entornos de pruebas o producción. Esto mejora significativamente la experiencia del desarrollador al crear, ajustar y depurar los aspectos de observabilidad de cualquier sistema en el que esté trabajando.

92. SDK para Android XR

Evaluar

Google, en colaboración con Samsung y Qualcomm presentó Android XR, un nuevo sistema operativo diseñado para visores XR. El soporte ha sido planificado para gafas y otros dispositivos. La mayoría de las aplicaciones Android son compatibles con pocos o ningún cambio, pero la idea es construir nuevas aplicaciones espaciales desde cero o “espacializar” las aplicaciones existentes. El nuevo Android XR SDK se posiciona como el SDK de referencia para este tipo de proyectos y Google proporciona una guía sobre cómo escoger herramientas y tecnologías que forman parte del SDK. Actualmente está disponible una versión en vista previa para developers.

93. Browser Use

Evaluar

Browser Use es una biblioteca de código abierto que permite a los agentes de IA basados en LLM, usar navegadores web para acceder a aplicaciones web. Puede controlar el navegador y realizar acciones que incluyen navegaciones, entrada y extracción de textos. Con la capacidad de manejar múltiples pestañas, puede orquestar acciones coordinadas a través de múltiples aplicaciones web. Es útil en escenarios donde los agentes basados en LLM necesitan acceso a contenido web, realizar acciones en él y obtener los resultados. La biblioteca puede trabajar con una variedad de LLMs. Utiliza Playwright para controlar el navegador web, combinando la comprensión visual con la extracción de estructura HTML para mejorar la interacción web. Esta biblioteca está ganando terreno en entornos multiagente, permitiendo a los agentes colaborar en flujos de trabajo complejos que involucran interacciones web.

94. CrewAI

Evaluar

CrewAI es una plataforma diseñada para ayudarte a construir y gestionar agentes de IA que pueden trabajar juntos para llevar a cabo tareas complejas. Piénsalo como una forma de crear una tripulación de trabajadores de IA, cada uno con sus propias habilidades especiales, que pueden colaborar para alcanzar un objetivo común. Hemos mencionado anteriormente en el Radar bajo agentes autónomos impulsados por LLM. Además de la biblioteca de Python de código abierto, CrewAI ahora cuenta con una solución empresarial para que las organizaciones puedan crear aplicaciones basadas en agentes para casos de negocio reales, ejecutarlas en su infraestructura en la nube y conectarlas a fuentes de datos existentes, como Sharepoint o JIRA. Hemos utilizado CrewAI en múltiples ocasiones para abordar desafíos de producción, desde la validación automatizada de códigos promocionales hasta la investigación de fallas en transacciones y consultas de soporte al cliente. Mientras el panorama de los agentes de IA sigue evolucionando rápidamente, estamos seguros de ubicar a CrewAI en Evaluar.

95. ElysiaJs

Evaluar

ElysiaJS es un framework web con seguridad de tipos de extremo a extremo para TypeScript, diseñado principalmente para Bun pero también compatible con otros entornos de ejecución de JavaScript. A diferencia de alternativas como tRPC, que impone estructuras específicas de interfaz de API, ElysiaJS no impone ninguna estructura de interfaz de API. Esto permite a developers crear APIs que sigan prácticas establecidas en la industria, como RESTful, JSON: API u OpenAPI, y también proporciona seguridad de tipos de extremo a extremo. ElysiaJS ofrece un alto rendimiento cuando se utiliza con el entorno de ejecución de Bun, llegando a ser comparable a frameworks web de Java o Go en algunos benchmarks. ElysiaJS es una opción que vale la pena considerar, especialmente al crear un backend-for-frontend (BFF).

96. FastGraphRAG

Evaluar

FastGraphRAG es una implementación de código abierto de GraphRAG diseñada para ofrecer gran precisión y alto rendimiento en la recuperación de información. Emplea Personalized PageRank para limitar la navegación del grafo únicamente a los nodos más relevantes, mejorando así la precisión de la recuperación y la calidad de las respuestas del LLM. También proporciona una representación visual del grafo, ayudando a los usuarios a comprender mejor las relaciones entre nodos y el proceso de búsqueda. Con compatibilidad para actualizaciones incrementales, se adapta muy bien a conjuntos de datos dinámicos y en constante evolución. Optimizado para casos de uso de GraphRAG a gran escala, FastGraphRAG mejora el rendimiento al tiempo que minimiza el consumo de recursos.

97. Gleam

Evaluar

Erlang/OTP es una plataforma potente para construir sistemas distribuidos altamente concurrentes, escalables y tolerantes a fallos. Tradicionalmente, sus lenguajes han sido de tipado dinámico, pero Gleam introduce seguridad de tipos a nivel de lenguaje. Construido sobre BEAM, Gleam combina la expresividad de la programación funcional con la seguridad de tipos en tiempo de compilación, reduciendo errores en tiempo de ejecución y mejorando la mantenibilidad. Con una sintaxis moderna, se integra bien con el ecosistema OTP, aprovechando las fortalezas de Erlang y Elixir al mismo tiempo que garantiza una interoperabilidad sólida. La comunidad de Gleam es activa y acogedora, y esperamos con interés su desarrollo continuo.

98. GoFr

Evaluar

GoFr es un framework para construir microservicios en Golang, diseñado para simplificar el desarrollo al abstraer el código repetitivo de funcionalidades comunes de microservicios, como el registro de logs, trazabilidad, métricas, gestión de configuración y documentación de API con Swagger. Soporta múltiples bases de datos, gestiona migraciones y facilita la comunicación pub/sub con brokers como Kafka y NATs. Además, GoFr incluye la programación de tareas con cron jobs. Reduce la complejidad de crear y mantener microservicios, y permite a developers centrarse en escribir la lógica de negocio en lugar de preocuparse por la infraestructura. Aunque existen otras librerías populares de Go para construir APIs web, GoFr está ganando reconocimiento y vale la pena explorarlo para microservicios basados en Golang.

99. Criptografía postcuántica con Java

Evaluar

La criptografía asimétrica, que hace seguras la mayoría de las comunicaciones modernas, está basada en la resolución de un problema matemáticamente difícil. Sin embargo, el problema utilizado en los algoritmos actuales será fácil de resolver con computadoras cuánticas, lo que impulsa la investigación en alternativas. La criptografía basada en retículos es actualmente la candidata más prometedora. Aunque a las computadoras cuánticas les faltan años para ser criptográficamente relevantes, vale la pena considerar la criptografía postcuántica para aplicaciones que deben permanecer seguras durante décadas. También existe el riesgo de que datos cifrados estén siendo recolectados ahora mismo para ser descifrados una vez que se pueda hacer uso de las computadoras cuánticas.

La criptografía postcuántica en Java dará sus primeros pasos en JDK 24, que estará disponible de forma general a finales de marzo. Esta versión de lanzamiento incluye JEP 496 y JEP 497 que implementan un mecanismo de encapsulación de claves y un algoritmo de firma digital, ambos basados en estándares y diseñados para ser resistentes a futuros ataques de computación cuántica. Aunque liboqs parte del proyecto Open Quantum Safe, proporciona implementaciones en C con un contenedor JNI, es grato también ver emerger una implementación nativa en Java.

100. Presidio

Evaluar

Presidio es un SDK de protección de datos para identificar y anonimizar sdatos sensibles en texto estructurado y no estructurado. Presidio detecta información de identificación personal (PII) como números de tarjetas de crédito, nombres y ubicaciones, usando reconocimiento de nombres, expresiones regulares y lógica basada en reglas. Presidio admite reconocimiento de entidades de PII personalizable, lo que permite a las empresas adaptarlo a sus requisitos específicos de privacidad. A pesar de que, Presidio automatiza la identificación de información sensible, no es infalible y puede perder o identificar erróneamente los datos. Tenga cuidado al confiar en sus resultados.

101. PydanticAI

Evaluar

A medida que las tecnologías para construir aplicaciones y agentes basados en LLM continúan evolucionando rápidamente, los frameworks para construir y orquestar tales aplicaciones a menudo luchan por mantenerse actualizados o encontrar las abstracciones correctas y eternas. PydanticAI es el último participante en este espacio, con el objetivo de simplificar las implementaciones mientras

se evita complejidad innecesaria. Desarrollado por los creadores del popular Pydantic, se basa en las lecciones aprendidas de marcos anteriores, muchos de los cuales ya dependen de Pydantic. En lugar de intentar ser aplicable para toda situación, PydanticAI ofrece un enfoque ligero pero poderoso. Se integra con todas las principales APIs de modelos e incluye herramientas integradas de salida estructurada de LLMs e introduce una abstracción basada en grafos para gestionar flujos de trabajo complejos de agentes.

102. Swift para aplicaciones con recursos limitados

Evaluar

Desde el lanzamiento de Swift 6.0, el lenguaje ha expandido su alcance más allá del ecosistema de Apple con un soporte mejorado para los principales sistemas operativos, lo que hace que sea más viable usar Swift para aplicaciones con recursos limitados. Tradicionalmente, este espacio ha sido dominado por C, C++ y, más recientemente, Rust, debido a su control de bajo nivel, alto rendimiento y disponibilidad de compiladores y bibliotecas certificadas que cumplen con los estándares como MISRA, ISO 26262 y ASIL. Mientras Rust ha comenzado a obtener certificaciones similares, Swift aún no ha iniciado este proceso, lo que limita su uso en aplicaciones críticas para la seguridad.

La creciente adopción de Swift se debe a su equilibrio entre rendimiento y características de seguridad, que incluyen una robusta seguridad de tipos y el recuento automático de referencias para la gestión de memoria. Mientras el modelo de propiedad de Rust ofrece garantías más fuertes de seguridad de memoria, Swift ofrece un enfoque diferente que algunos developers encuentran más accesible. Tanto Swift como Rust comparten el backend del compilador LLVM/Clang, lo que permite que los avances en uno beneficien al otro. Con su capacidad para compilar código de máquina optimizado, su desarrollo de código abierto y su creciente soporte multiplataforma, Swift está emergiendo como un contendiente para una gama más amplia de aplicaciones — mucho más allá de sus raíces en iOS.

103. Tamagui

Evaluar

Tamagui es una biblioteca para compartir estilos de manera eficiente entre React web y React Native. Ofrece un sistema de diseño con componentes reutilizables, tanto con estilos como sin ellos, que se renderizan perfectamente en diversas plataformas. Su compilador optimizador opcional mejora el rendimiento al convertir los componentes con estilo en CSS atómico con divs en la web y objetos de estilo elevados en vistas nativas.

104. torchtune

Evaluar

torchtune es una librería de PyTorch para la autoría, post-entrenamiento y experimentación con LLMs. Soporta configuraciones individuales y multi-GPU y habilita entrenamiento distribuído con FSDP2. La librería provee de recipes basadas en YAML para tareas como ajustes finos, inferencia, evaluación y entrenamiento consciente de la cuantización. Cada receta ofrece un set de características enfocado, evitando configuraciones complejas basadas en flags. Prioriza simpleza, favoreciendo la caridad del código por sobre abstracciones excesivas. Además incluye un CLI para descargar modelos, administrando recetas y ejecutando experimentos de forma eficiente.

105. Sobrecarga de Node

Resistir

Hace unos años, observamos una sobrecarga de Node : Node.js se utilizaba a menudo por razones cuestionables o sin tener en cuenta otra alternativa. Aunque entendemos que algunos equipos prefieren un stack de un solo lenguaje, a pesar de las desventajas, seguimos defendiendo la programación políglota. En aquel momento, señalamos que Node.js tenía una merecida reputación de eficiencia en cargas de trabajo con uso intensivo de E/S, pero mencionamos que otros frameworks se habían puesto al día y ofrecían mejores APIs y un rendimiento general superior. También advertimos que Node.js nunca fue adecuado para cargas de trabajo de cómputo intensivo, una limitación que sigue siendo un desafío significativo. Ahora, con el auge de las cargas de trabajo con uso intensivo de datos, también vemos a los equipos tener dificultades con estas.

Mantente al día de todas las noticias y opiniones relacionadas con Radar

Suscríbete al Radar Tecnológico para recibir correos electrónicos cada dos meses con información sobre tecnología de Thoughtworks y futuras publicaciones del Radar Tecnológico.

Suscríbete ahora

Somos una consultora tecnológica global que genera un impacto extraordinario al combinar experiencia en diseño, ingeniería e IA.

Durante más de 30 años, nuestra cultura de innovación y excelencia tecnológica ha ayudado a nuestra base de clientes a fortalecer sus sistemas empresariales, escalar con agilidad y crear experiencias digitales integradas.

Nos dedicamos a resolver los desafíos más críticos de nuestra base de clientes, combinando la inteligencia artificial con la creatividad humana para convertir sus ideas más ambiciosas en realidad.

/thoughtworks

Diseño. Ingeniería. IA.