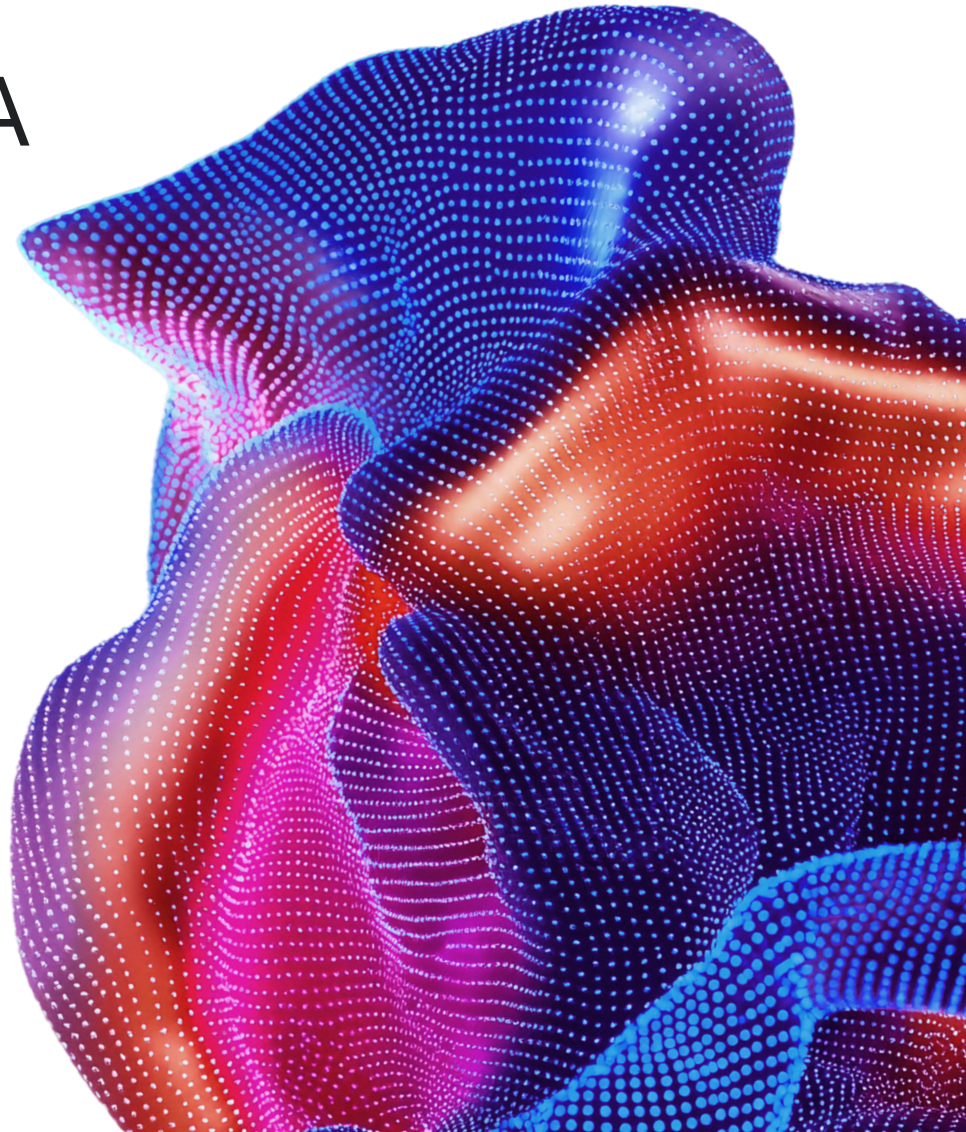
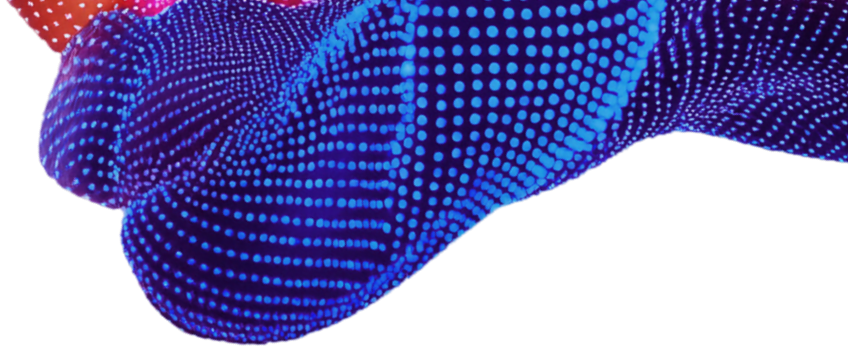
Google Cloud

Guía técnica para Startups

Agentes de IA





Índice

Introducción

El desarrollo de agentes de IA representa un cambio de paradigma en la ingeniería de software, que permite a las startups automatizar flujos de trabajo complejos, crear experiencias de usuario novedosas y resolver problemas de negocios que hasta ahora eran técnicamente inviables.

Pero pasar de un prototipo prometedor a un agente listo para producción implica resolver una serie de desafíos nuevos. ¿Cómo lidiar con el comportamiento no determinista de esos agentes? ¿Cómo verificar sus complejas rutas de razonamiento? Y, fundamentalmente, ¿por dónde empezar?

Esta guía técnica ayudará a responder preguntas como estas y brinda una hoja de ruta sistemática, centrada en la práctica, para orientar a quienes se inician en este nuevo universo. Está dirigida a startups y desarrolladores que quieren capitalizar el potencial de los sistemas de agentes.

Aprenderás los conceptos básicos de los sistemas de agentes, desde sus principales componentes arquitectónicos hasta los principios que garantizan una operación fiable y responsable en el entorno de producción. También conocerás el conjunto completo de herramientas que hacen más eficiente la creación y el uso de agentes en Google Cloud, desde el desarrollo centrado en código (code-first) usando el Agent Development Kit (ADK) y la automatización operativa con el Agent Starter Pack, hasta el diseño de agentes sin programar con Gemini Enterprise.

Esta guía será de gran utilidad en todas las etapas de tu proyecto, desde la validación de la idea y la construcción de un producto mínimo viable (MVP), hasta el soporte de un producto en producción.

Cómo usar esta guía

¿Recién te inicias en el mundo de los agentes de IA?

Comienza con la Sección 1 para familiarizarte con los conceptos básicos.

¿Todo listo para desarrollar?

Ve directo a la Sección 2 y crea tu primer agente usando el ADK.

¿Ya creaste un agente?

Ve a la Sección 3 para optimizar su seguridad, estabilidad y escalabilidad.

¿Buscas impulso adicional?

Utiliza el Gemini Kit para prototipar con mayor agilidad y solicita la inscripción al programa Google Cloud for Startups para recibir orientación especializada y hasta $350,000 USD en créditos de Cloud.

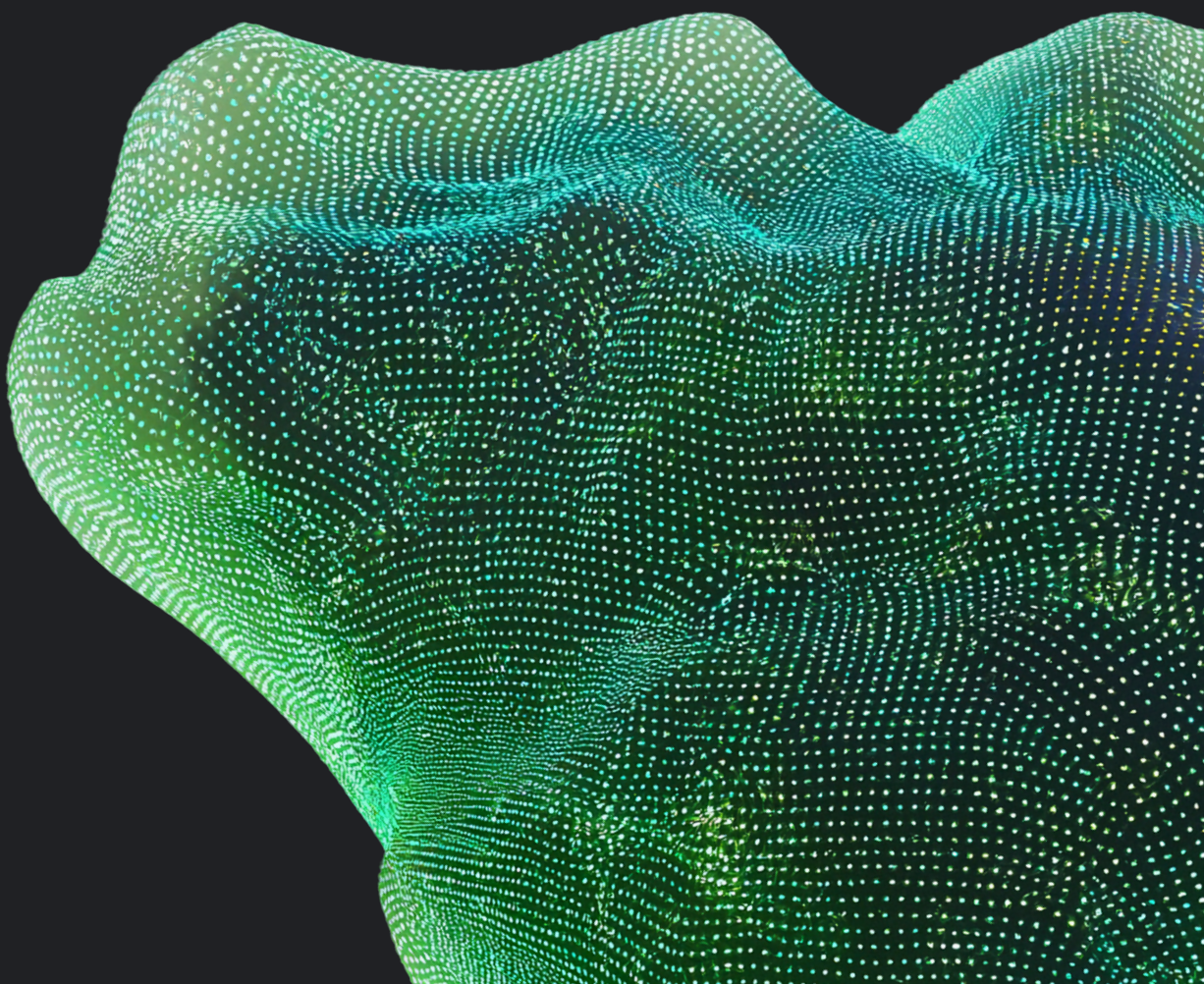
El enfoque de esta guía

El ecosistema de IA basada en agentes ofrece una amplia gama de herramientas, bibliotecas y enfoques para desarrollar arquitecturas cognitivas. Google ofrece frameworks de código abierto como Genkit y soluciones de IA conversacional en Google Cloud, que coexisten con bibliotecas populares como LangChain y CrewAI.

Esta guía se centra primordialmente en el ADK, y comparte conceptos y patrones arquitectónicos diseñados para crear agentes robustos y escalables en Google Cloud, manteniendo la flexibilidad necesaria para integrar cualquier otra herramienta o biblioteca de tu preferencia.

Sección 1

Conceptos básicos de los agentes de IA



El campo de la IA basada en agentes está evolucionando vertiginosamente. Esta sección presenta los fundamentos de los agentes de IA y explica sus conceptos básicos, propósitos y funcionamiento. También detalla las herramientas y los servicios relevantes disponibles dentro de Google Cloud.

¿Prefieres audio? Escucha la versión en podcast de esta sección, creada con NotebookLM.

Este podcast fue creado usando NotebookLM con el siguiente prompt: “Como presentador de podcast, genera un episodio conversacional y educativo sobre la ‘Guía técnica para startups: Agentes de IA’, dirigido a un público técnico de desarrolladores y fundadores de startups. La sesión debe cubrir los tres pilares del uso de agentes de IA (desarrollo, aplicación, integración) y profundizar sobre herramientas como el Agent Development Kit (ADK) y los agentes Gemini preconfigurados.

“Luego explica los componentes principales de un agente, como los modelos, las herramientas, la orquestación y el entorno de ejecución. También menciona cómo garantizar la confianza y el desempeño a través de técnicas como la fundamentación (grounding) con Generación Aumentada por Recuperación (RAG) y el uso de la multimodalidad. Concluye con un resumen de los puntos clave y un llamado a la acción claro que invite a los oyentes a explorar los recursos de Google”.

1.1 Descripción general del ecosistema de agentes de Google Cloud

> **El flujo de trabajo basado en agentes representa el nuevo paradigma. No se trata simplemente de hacer una pregunta y obtener una respuesta; se trata de asignar a la IA un objetivo complejo —como 'planificar este lanzamiento de producto' o 'resolver esta interrupción en la cadena de suministro' — y permitirle orquestar los múltiples pasos necesarios para lograrlo. Esto transformará radicalmente la productividad."**
>
> **Thomas Kurian**
> CEO de Google Cloud

La creación de agentes de IA listos para producción requiere algo más que elegir un gran modelo de lenguaje. Una solución completa exige una infraestructura escalable, herramientas robustas de integración de datos y patrones arquitectónicos que se adapten a diversos requisitos técnicos.

Google Cloud respalda el desarrollo integral de sistemas de agentes, ya sea para crear tus propios agentes, usar agentes de Google Cloud prediseñados o integrar agentes de terceros. Respaldado por el Protocolo de Contexto de Modelo (MCP) y el protocolo Agent2Agent (A2A), este framework común fue diseñado para garantizar la interoperabilidad. De esta manera, independientemente de su origen o arquitectura, tus agentes pueden colaborar dentro del ecosistema de Google Cloud.[1]

1. Los protocolos MCP y A2A se analizan en profundidad en la Sección 2 de esta guía.

Crea tus propios agentes

Si buscas desarrollar agentes personalizados para tareas específicas, este es el camino ideal. Existen dos enfoques posibles: uno centrado en código (code-first), que ofrece control total, y otro centrado en la aplicación (application-first), que acelera los tiempos de desarrollo.

Agent Development Kit: desarrollo personalizado centrado en código

Este enfoque es ideal para desarrolladores, startups técnicas y equipos que precisan un control granular sobre el comportamiento de los agentes. El Agent Development Kit (ADK) de Google Cloud fue diseñado específicamente para este enfoque personalizado.

El ADK permite a los desarrolladores construir, gestionar, evaluar y desplegar agentes con IA. Proporciona un entorno robusto y flexible para crear agentes tanto conversacionales como no conversacionales, capaces de manejar tareas y flujos de trabajo complejos.

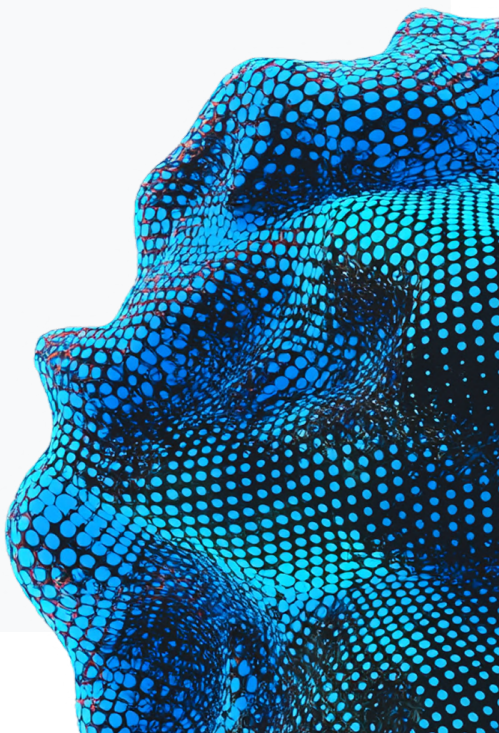
Los agentes creados con el ADK pueden desplegarse fácilmente en Vertex AI Agent Engine, un entorno gestionado y escalable diseñado para esta carga de trabajo. Dado que estos agentes están contenerizados, también ofrecen la flexibilidad de ejecutarse en cualquier entorno compatible con contenedores, como Cloud Run y Google Kubernetes Engine (GKE).

Principales características

- **Lógica de orquestación:** El motor de razonamiento central del agente, como el framework ReAct (ver la Sección 1.2), permite planificar y ejecutar secuencias de llamadas a herramientas y acciones para lograr objetivos complejos.
- **Definición y registro de herramientas:** Una interfaz para definir funciones y API personalizadas, que facilita la interacción del agente con datos, API y sistemas externos.
- **Gestión de contexto:** Un sistema que proporciona memoria al agente y le permite retener las preferencias del usuario y el historial de conversaciones en múltiples interacciones para garantizar una experiencia coherente.
- **Evaluación y observabilidad**: Conjunto de herramientas integradas para realizar pruebas de calidad rigurosas del agente, depurar el razonamiento paso a paso y monitorear el rendimiento en entornos de producción.
- **Contenerización**: Permite empaquetar al agente en un contenedor estándar y portátil, preparándolo para un despliegue ágil en cualquier entorno de nube compatible.
- **Composición multiagente:** Facilita la creación de sistemas donde múltiples agentes especializados colaboran, delegan tareas y trabajan juntos para resolver problemas.

Por qué es importante para las startups

- **Automatiza flujos de trabajo, no solo conversaciones:** Implementa lógica de orquestación de varios pasos para resolver problemas de negocio complejos. Esto genera la eficiencia operativa que un equipo pequeño necesita para escalar sin límites.
- **Desarrolla un producto competitivo:** Conecta agentes directamente a tus API propietarias y datos internos para crear un producto con una ventaja competitiva real.
- **Recuerda a tus clientes para ofrecer una experiencia realmente personalizada**: Integra de forma fluida el contexto de las conversaciones recientes con el conocimiento acumulado a largo plazo, para lograr que tu agente recuerde interacciones pasadas y construya una relación genuina con el cliente.
- **Realiza el lanzamiento con confianza:** Aprovecha las herramientas de evaluación y observabilidad integradas para probar y depurar de forma rigurosa a tu agente, y garantizar un producto fiable y listo para producción.
- **Se centra en el producto, no en la infraestructura:** Empaqueta tu agente en un contenedor estándar para llegar a producción de forma más rápida y fiable, utilizando las prácticas estándar de DevOps.



Gemini Enterprise: desarrollo centrado en la aplicación

La segunda alternativa para crear agentes es usar Gemini Enterprise. A diferencia del ADK que tiene un enfoque centrado en código, Google Agentspace permite orquestar toda la fuerza de trabajo de IA y facilita que perfiles no técnicos creen agentes personalizados sin necesidad de escribir código.

Este enfoque de plataforma es ideal para gestionar múltiples agentes y escalar su uso a medida que tu startup crece y tu portafolio de aplicaciones SaaS se expande.

Principales características

- **Búsqueda unificada en toda la empresa:** Conéctate y realiza búsquedas en distintas aplicaciones SaaS.
- **Síntesis de datos multimodales:** Entiende y sintetiza la información de textos, imágenes, gráficos y videos respetando los permisos de datos.
- **Biblioteca de agentes preconfigurados:** Ofrece un conjunto de agentes listos para usar en tareas complejas, como investigación profunda o generación de ideas.
- **Creación de agentes personalizados sin código:** Integra Agent Designer, que permite a los perfiles no técnicos crear agentes mediante una interfaz basada en prompts.

Por qué es importante para las startups

- **Elimina los silos de datos:** Los equipos sin desarrolladores pueden crear y desplegar agentes capaces de acceder y operar a través de aplicaciones y fuentes de datos fragmentadas.
- **Automatiza los flujos de trabajo:** Crea flujos de trabajo multiplataforma sin consumir los escasos recursos de ingeniería para que el equipo pueda centrarse en el desarrollo del producto principal.

> “Convertir a Gemini en un modelo mundial es un paso crucial en el desarrollo de un nuevo tipo de IA más general y útil: un asistente universal de IA. Se trata de una IA inteligente que entiende el contexto en el que te encuentras y que puede planificar y actuar en tu nombre, desde cualquier dispositivo”.
>
> **Demis Hassabis**
> CEO de Google DeepMind

Usa agentes de Google Cloud

Con un prototipado rápido y formas sencillas de integrar la IA en tus aplicaciones existentes, los agentes gestionados te permiten centrarte en la lógica principal de negocio en lugar de gestionar la infraestructura. También son ideales si tus recursos de ingeniería son limitados.

Gemini Code Assist

Gemini Code Assist es un asistente con tecnología de IA para desarrolladores que se integra en múltiples puntos del ciclo de vida del desarrollo de software y ofrece soporte a través de extensiones de IDE, una interfaz de línea de comandos, integración con GitHub y dentro de varios servicios de Google Cloud.

Principales recursos

- **Integración con IDE:** Dentro de los IDE populares (VS Code, JetBrains IDEs, Android Studio), permite el completado automático de código, generación de funciones bajo demanda y una interfaz de chat. Usa la gran ventana de contexto de Gemini para proporcionar respuestas relevantes al código base abierto. Las ediciones Enterprise pueden estar conectadas a repositorios de código fuente privados para ofrecer sugerencias más personalizadas.
- **Interfaz de línea de comandos:** La CLI de Gemini es un agente de IA de código abierto que lleva las capacidades de Gemini directamente a la terminal para tareas como comprensión de código, manipulación de archivos y resolución dinámica de problemas.
- **Integración con GitHub:** En GitHub, Gemini Code Assist puede revisar automáticamente los pull request para identificar errores y problemas de estilo, sugiriendo cambios de código específicos.
- **Desarrollo guiado por agentes:** Despliega agentes de IA capaces de realizar ediciones complejas en varios archivos dentro del contexto completo de un proyecto. Estos flujos de trabajo agénticos incorporan la supervisión humana (HITL) y pueden integrarse con herramientas del ecosistema que siguen el MCP.
- **Integración de servicios de Google Cloud:** Ofrece asistencia de IA directamente en servicios como Firebase (análisis de errores de aplicaciones, información sobre rendimiento), Colab Enterprise (generación de código Python), BigQuery (lenguaje natural a SQL, optimización de consultas), Cloud Run y Apigee.

Por qué es importante para las startups

Gemini Code Assist actúa como un multiplicador de fuerza. Puede manejar las tareas de compilación en todo el ciclo de vida de desarrollo del software, desde tareas rutinarias como escribir código repetitivo hasta operaciones más complejas como la refactorización de múltiples archivos.

Puedes delegar una amplia gama de tareas a Gemini Code Assist. A continuación, incluimos algunos ejemplos que muestran sus posibles usos.

- **Automatización de código repetitivo:** Genera una Cloud Function en Python que se active con una solicitud HTTP. Debe analizar una carga útil JSON para un userId y documentId, luego debe usar la biblioteca cliente google-cloud-firestore para buscar un documento específico en una colección 'users' y regresarlo como una respuesta JSON.

- **Pruebas integrales:** Usa una de tus funciones existentes y pide a Code Assist que genere un conjunto completo de pruebas, incluidos los objetos simulados necesarios para servicios de Google Cloud, como Cloud Storage o Firestore.

- **Refactorización a gran escala con Gemini:** Pídele que analice múltiples servicios en tu base de código y que genere un plan estratégico. Por ejemplo: "Dado nuestro 'user-service' y 'auth-service', sugiere un plan paso a paso para refactorizar la lógica de autenticación en una biblioteca compartida única, y detalla las ventajas y desventajas de este enfoque".

Gemini Cloud Assist

Gemini Cloud Assist es un experto en IA para tu entorno de Google Cloud que ofrece asistencia contextual para la gestión de infraestructura y operaciones de aplicaciones. Usa el contexto de tu proyecto, como los ID de proyectos de Google Cloud y la página de producto específica que se está viendo en la consola, para adaptar la asistencia.[2]

Principales características

- **Diseño y despliegue:** Dentro del Application Design Center, puedes describir la infraestructura deseada en lenguaje natural. Gemini Cloud Assist genera diagramas de arquitectura y plantillas de aplicaciones exportables como Terraform para facilitar su integración con los flujos de trabajo existentes de Infraestructura como Código (IaC).
- **Diagnóstico y resolución de problemas:** Se integra con Cloud Observability para resumir registros complejos y explicar mensajes de error. Ante problemas más profundos, permite iniciar investigaciones, en las cuales Gemini analiza registros y métricas para identificar la causa raíz.
- **Configuración y optimización:** Ofrece recomendaciones personalizadas de costo y rendimiento a través del FinOps Hub y del panel de Optimización de Costos.
- **Seguridad y análisis:** Facilita la investigación de flujos de red y registros en lenguaje natural. Brinda orientación en tareas críticas de seguridad como el cifrado de datos, la gestión de secretos y la validación de políticas organizativas personalizadas. Además, recomienda roles de IAM y diagnostica errores de permisos.

Por qué es importante para las startups

- **Ahorra tiempo:** La administración de la nube puede consumir mucho tiempo de ingeniería. Gemini Cloud Assist te permite dedicar ese tiempo a crear tu producto.

Prueba estos prompts en Gemini Cloud Assist:

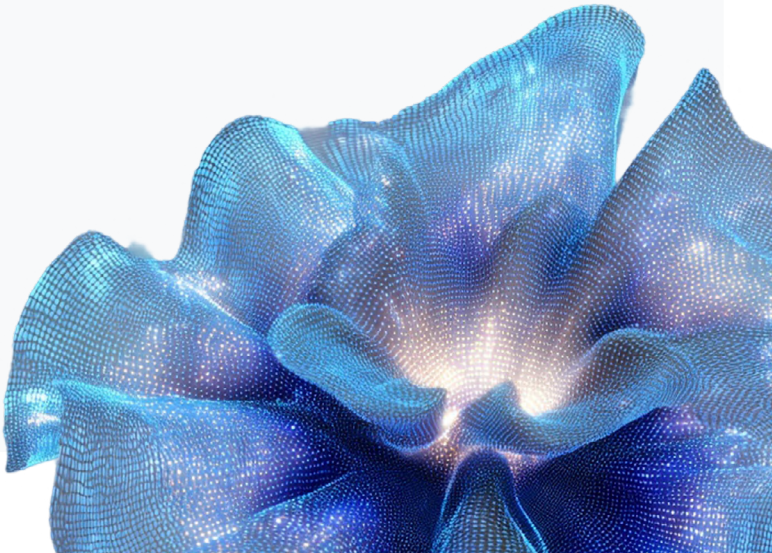
> ¿Cómo uso Vertex AI para desplegar un modelo?

> Crea un plan de alto nivel para diseñar, desarrollar y desplegar una aplicación web en Google Cloud.

> Haz una lista de todos los buckets de Cloud Storage en el proyecto prod-v1 que no tengan habilitado el Control de versiones de objetos.

> ¿Cuáles son las reglas de firewall dirigidas al público que se aplican a instancias con la etiqueta de red external-web-server?

> Muéstrame todos las funciones de IAM otorgadas a la cuenta de servicio data-pipeline@my-project.iam.gserviceaccount.com



2. Para obtener información sobre cómo se fundamenta Gemini Cloud Assist, consulta la documentación oficial.

Gemini en Colab Enterprise

Si tu startup está trabajando en ciencia de datos, aprendizaje automático o análisis de datos, Gemini en Colab Enterprise convierte cada notebook en un espacio de trabajo colaborativo con IA. Está diseñado para generar, explicar y depurar código Python en contexto.

Principales características

- Finalización y generación automática de código Python dentro de Cola
- Explicación de la lógica del código y errores en lenguaje simple
- Filtrado, transformación y visualización de datos
- Recomendación de conjuntos de datos públicos y recursos de investigación
- Resumen de notebooks enteros o celdas de código

Prueba estos prompts en Gemini usando Colab Enterprise:

> ¿Cómo filtro un DataFrame de Pandas?
>
> Haz un gráfico del ingreso promedio por región.
>
> Muéstrame una lista de conjuntos de datos disponibles públicamente para tecnología climática
>
> Resume el objetivo de este notebook.

Por qué es importante para las startups

- **Acelera la investigación y el desarrollo:** Automatiza los aspectos más tediosos de la preparación, el análisis y la visualización de datos, y permite a los desarrolladores iterar en nuevos modelos e ideas de forma mucho más rápida.
- **Reduce la barrera de entrada:** Los ingenieros que están dando sus primeros pasos en ciencia de datos pueden comenzar más rápido, mientras que los profesionales con mayor experiencia pueden dedicar más tiempo a la experimentación de modelos y menos tiempo a la manipulación de datos.

Incorpora agentes de socios

Si tu caso de uso es más especializado, puedes integrar fácilmente agentes de terceros o de código abierto en tu stack utilizando el ecosistema abierto de Google Cloud y el Google Cloud Marketplace.

Explora Agent Garden para implementar agentes de ADK prediseñados compatibles con el razonamiento de datos y la colaboración entre agentes. Puedes mezclarlos y combinarlos con los agentes que construyas, acelerando el tiempo de impacto.

1.2 Principales componentes de cada agente

Modelos: selección y ajuste

Piensa en el modelo como el cerebro de tu agente. Puedes usarlo para leer solicitudes de usuarios, entender qué debería suceder y generar respuestas inteligentes.

Cómo elegir el modelo correcto

Elegir el modelo correcto no es solo seleccionar el más potente disponible, sino encontrar el equilibrio óptimo entre capacidad, velocidad y costo para tu caso de uso. Cada modelo puede ser evaluado tomando en cuenta esas tres características conflictivas, y el objetivo siempre es identificar la opción más eficiente para una tarea específica.

A medida que la capacidad de un modelo aumenta, su costo y latencia generalmente aumentan también. El error más común es invertir de más en capacidad cuando un caso de uso no lo requiere, algo que termina generando un gasto innecesario y un rendimiento más lento. La estrategia ideal es seleccionar el modelo más eficiente para cada tarea particular.

Este principio es aún más importante cuando se aplica a nivel de sistema. Las arquitecturas cognitivas robustas utilizan múltiples agentes especializados, donde cada uno selecciona dinámicamente el modelo más ligero para su subtarea específica. Esto asegura, por ejemplo, que un modelo más pesado se reserve para los razonamientos complejos, mientras que un modelo ligero se ocupe de consultas rutinarias. Este enfoque multiagente ofrece la flexibilidad de arquitectura necesaria para optimizar el costo y el rendimiento de todo el sistema, y no solo de un componente aislado.

> “Los agentes de IA son sistemas que combinan la inteligencia de modelos de IA avanzados con el acceso a herramientas para que actúen en tu nombre, siempre bajo tu supervisión”.
>
> **Sundar Pichai**
> CEO de Google e Alphabet

Casos de uso

Prototipado inicial y tareas a gran escala

- **Perfil del modelo:** Un modelo ligero y de bajo costo como Gemini 2.5 Flash-Lite.
- **Justificación:** Este es el modelo 2.5 más rápido y eficiente en términos de costo, ideal para tareas de gran volumen y sensibles a la latencia, como traducción y clasificación.

Aplicaciones de alto volumen y alta calidad

- **Perfil del modelo:** Un modelo equilibrado de gama media como Gemini 2.5 Flash.
- **Justificación:** Fue diseñado para controlar el equilibrio entre calidad, costo y velocidad. Ofrece un rendimiento sólido en tareas complejas a un precio más bajo que la versión Pro, que lo hace ideal para aplicaciones de producción que requieren inteligencia y economía.

Razonamiento complejo de múltiples pasos y generación de código avanzada

- **Perfil del modelo:** Un modelo de razonamiento avanzado como Gemini 2.5 Pro.
- **Justificación:** Este es el modelo con más capacidad para las tareas difíciles donde el rendimiento no es negociable.[3]

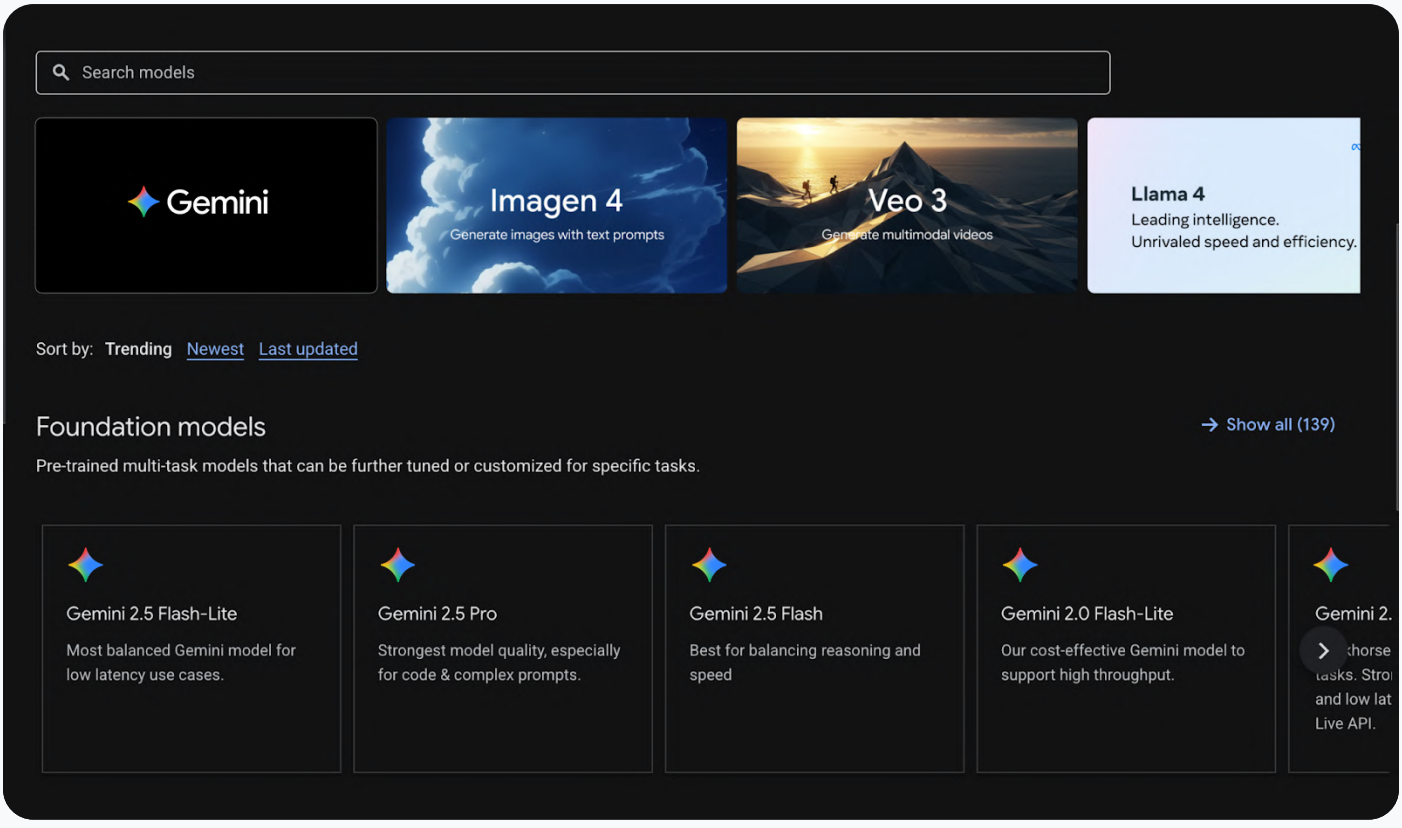
3. Gemini 2.5 Pro alcanzó resultados líderes en benchmarks avanzados: 82.2% en Aider Polyglot (programación de código) y 86.4% en GPQA diamond (razonamiento). Los datos corresponden a junio de 2025.

Puedes usar la familia de modelos Gemini 2.5 para desglosar problemas, formular planes y usar herramientas. Este proceso de razonamiento es configurable: al asignar más tokens de razonamiento a una llamada específica, el desarrollador puede indicar al modelo que dedique un mayor esfuerzo computacional. Esto supone asumir un aumento predecible en la latencia y los costos a cambio de una mejora potencial en la precisión.

Este control a nivel de token, combinado con la selección de modelos y modos de razonamiento configurables, ofrece a los desarrolladores un conjunto dinámico de herramientas para una optimización avanzada. El costo y rendimiento de un sistema multiagente pueden ajustarse para cumplir con requisitos técnicos y comerciales específicos.

Consejo experto

Usa Model Garden en Vertex AI para descubrir, personalizar y desplegar modelos fundacionales desde una plataforma única y centralizada. Ofrece una selección curada de más de 200 modelos de Google, socios como Anthropic y una amplia variedad de modelos abiertos de proveedores como Meta (familia Llama) y Mistral. En lugar de gestionar la infraestructura de forma manual, puedes desplegar los modelos en aplicaciones con un solo clic y escalarlos mediante las funcionalidades integradas de MLOps de extremo a extremo.



Ajuste del modelo

Una vez que seleccionas un modelo que se ajuste a tus necesidades de costo, latencia y calidad, tienes la opción de realizar un ajuste fino (fine-tuning). Este proceso especializa el conocimiento y el estilo del modelo para las necesidades específicas de tu negocio mediante un conjunto de datos refinado con tus propios ejemplos de alta calidad.

La disponibilidad del ajuste de modelos se determina de forma individual. Dentro del catálogo de modelos de Google, esta funcionalidad es compatible con la familia Gemma (modelos abiertos) y con versiones específicas de Gemini. Resulta fundamental revisar la documentación y el acuerdo de licencia de cada modelo para verificar si el ajuste está permitido y es técnicamente viable.

Consejo experto

Para saber qué modelos pueden ajustarse en Vertex AI, consulta la documentación oficial.

Caso de uso

Ajuste fino de un agente de atención al cliente

Supongamos que estás construyendo un agente de atención al cliente para tu producto SaaS. Podrías aplicar un ajuste fino utilizando un conjunto de datos con miles de tickets de soporte previos y sus resoluciones ideales. Esto ayuda al modelo a aprender sobre los problemas comunes y a responder con un tono alineado al estilo de tu equipo de atención.

Nota

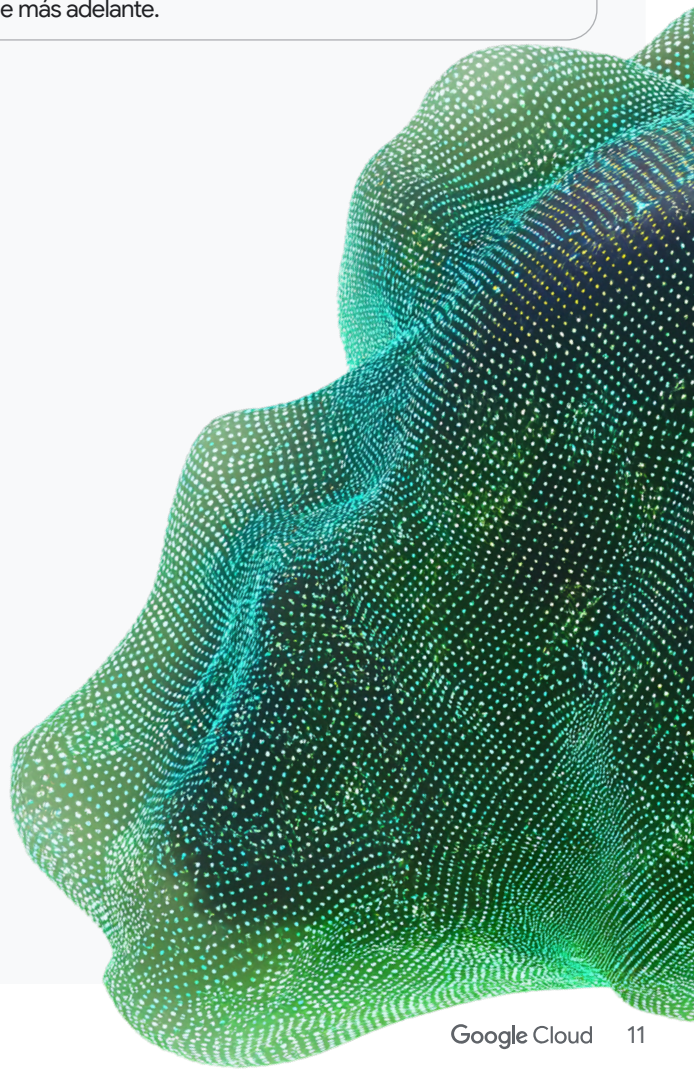
El ajuste fino no es lo mismo que fundamentación. El ajuste fino adapta el estilo de un modelo y refina su conocimiento en una tarea específica. La fundamentación (grounding) conecta el modelo a fuentes de datos verificables en tiempo real para garantizar que sus respuestas sean objetivamente correctas. La fundamentación del modelo se explica en detalle más adelante.

Herramientas: habilitar la acción de agentes

Las herramientas son capacidades definidas que permiten a un agente hacer más que las funciones nativas de su modelo de razonamiento central, desde realizar cálculos internos simples hasta interactuar con sistemas externos a través de llamadas a API. Cubren la brecha entre el razonamiento del agente y su capacidad para obtener nueva información o ejecutar operaciones con estado.

Las herramientas pueden incluir una amplia variedad de componentes:

- **Funciones y servicios internos:** Lógica propietaria elaborada por tu propio equipo.
- **API:** Conexiones tanto a servicios internos como a servicios externos de terceros.
- **Fuentes de datos:** La capacidad de consultar bases de datos, almacenes vectoriales u otros repositorios de información.
- **Otros agentes:** En sistemas multiagentes, un agente puede utilizar otro agente especializado como herramienta.



Arquitectura de datos para sistemas de agentes

Los datos son la base para la memoria a corto y largo plazo de un agente. Una arquitectura de datos robusta debe resolver tres necesidades distintas: almacenamiento persistente para la recuperación de conocimiento a largo plazo, acceso de baja latencia para contexto conversacional a corto plazo y un registro inmutable para auditoría transaccional. Al mapear servicios específicos de Google Cloud a cada una de estas necesidades, se garantiza que cada decisión de arquitectura sea rentable y escalable, y que atienda las necesidades inmediatas del negocio y los objetivos de tiempo de lanzamiento.

1. Base de conocimientos a largo plazo (fundamentación y recuperación)

La memoria a largo plazo de un agente es la base de su inteligencia, fundamentación y personalización. Se diferencia del contexto rápido y a corto plazo de una conversación en vivo. Una arquitectura robusta de memoria a largo plazo debe comprender tres componentes centrales: una base de conocimientos estructurada para la fundamentación basada en hechos a través de la Generación Aumentada por Recuperación (RAG); un almacenamiento persistente del historial de interacciones con el usuario para permitir una experiencia continua y personalizada; y un lago de datos operativo para elementos sin procesar, como las transcripciones de conversaciones y los estados de flujos de trabajo, destinados a procesos cognitivos más complejos y análisis futuro.

Servicio de datos	Descripción general	Casos de uso para startups
Vertex AI Search	Un servicio gestionado para compilar aplicaciones de búsqueda vectorial de alto rendimiento. Es la herramienta principal para habilitar la comprensión semántica y la recuperación de información en grandes conjuntos de datos no estructurados.	Encuentra respuestas al instante dentro de la documentación interna de tu producto, los registros de chat de atención al cliente y las publicaciones de los foros de la comunidad, para que tu agente pueda brindar un soporte preciso y contextual a los nuevos usuarios. Esto reduce la carga de trabajo de tu pequeño equipo de soporte.
Firestore	Una base de datos de documentos NoSQL sin servidor con capacidades de sincronización en tiempo real.	Mantén el estado en tiempo real de un flujo de incorporación de usuario de múltiples pasos guiado por el agente. A medida que el usuario completa cada paso (p. ej., "crear perfil", "conectar API", "invitar a miembro del equipo"), el agente actualiza un documento de Firestore. Los desarrolladores pueden entonces observar el progreso de la tarea del agente en tiempo real, y el usuario puede reanudar el proceso de forma fluida a través de diferentes sesiones.
Vertex AI Memory Bank (Vista previa)	Un servicio gestionado en Vertex AI Agent Engine y diseñado específicamente para generar, almacenar y recuperar dinámicamente memorias a largo plazo de conversaciones de usuarios.	En lugar de desarrollar manualmente una lógica para extraer las preferencias del usuario, un agente puede llamar automáticamente a GenerateMemories sobre un historial de conversación. Esto extrae hechos clave de forma asíncrona (p. ej., "el usuario prefiere vuelos sin escalas", "el perro del usuario se llama Fido") y los almacena. En sesiones futuras, el agente puede recuperar estas memorias mediante una búsqueda por similitud para ofrecer una experiencia profundamente personalizada y continua con un mínimo de código personalizado.
Cloud Storage	Un almacén de objetos altamente escalable y duradero para datos fuente no estructurados sin procesar (ej. PDF, imágenes), que alimenta a otros servicios para la indexación y el procesamiento.	Funciona como plataforma duradera y de bajo costo para todos los documentos subidos por los usuarios, imágenes de reportes de errores o grabaciones de llamadas de feedback de clientes. Estos datos brutos luego son procesados e indexados por servicios como Vertex AI Search para enriquecer el conocimiento de tu agente.
BigQuery	Un almacén de datos totalmente gestionado y sin servidor para almacenar y analizar conjuntos de datos masivos estructurados y semiestructurados, que permite a los agentes disponer de herramientas para ejecutar consultas analíticas complejas.	Un agente puede hacer peticiones como: "Resume los patrones de interacción de los usuarios con la nueva funcionalidad que lanzamos la semana pasada" o "¿Qué cohortes de clientes tienen el mayor riesgo de abandono tomando como base la actividad reciente?". BigQuery brinda inteligencia empresarial al instante.

2. Memoria de trabajo (contexto conversacional y estado a corto plazo)

Esta capa gestiona la información transitoria requerida para una tarea o conversación en curso. Se debe proporcionar un acceso de latencia extremadamente baja para mantener una experiencia de usuario rápida y fluida.

Servicio de datos	Descripción general	Casos de uso para startups
Memorystore	Un almacén de datos totalmente gestionado en memoria que proporciona una latencia inferior a milisegundos. Es ideal para el almacenamiento en caché de datos de acceso frecuente y la gestión del estado de la sesión.	Su función principal es el almacenamiento en caché de alta velocidad para guardar los resultados de cualquier operación computacionalmente de alto costo o de alta latencia. En lugar de ejecutar repetidamente una tarea costosa, como una llamada a la API de un LLM, una consulta de base de datos compleja o una llamada a un servicio de terceros, el agente verifica primero en Memorystore si hay un resultado en caché. Esto reduce drásticamente tanto la latencia de respuesta como los costos operativos recurrentes, ambos críticos para el sistema de agentes de una startup.

3. Memoria transaccional (gestión de estado y auditoría de acciones)

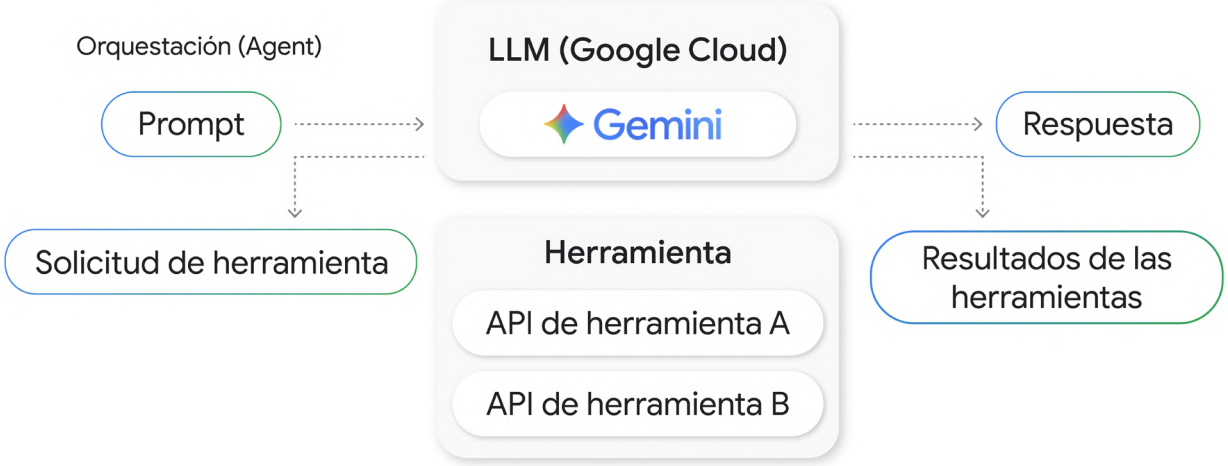
Esta capa es responsable de registrar las acciones y los cambios de estado con una fuerte consistencia e integridad. Funciona como el sistema de registro y exige garantías ACID frecuentes para asegurar la fiabilidad.

Servicio de datos	Descripción general	Casos de uso para startups
Cloud SQL	Un servicio totalmente gestionado para bases de datos relacionales tradicionales. Ofrece una fuerte consistencia para cargas de trabajo transaccionales de una sola región. Es la opción estándar para una gestión de estado fiable.	Cuando un agente ejecuta con éxito una acción de negocios crítica, como procesar el pago de una suscripción o aprovisionar un nuevo servicio para un usuario a través de una llamada a la API, registra esa acción en una base de datos de Cloud SQL. Esto crea un registro de auditoría permanente y compatible con ACID, que garantiza que cada acción importante impulsada por el agente sea rastreada de forma fiable y verificable.
Cloud Spanner	Una base de datos relacional distribuida globalmente, que ofrece una fuerte consistencia y escalabilidad horizontal. Está diseñada para aplicaciones de misión crítica que requieren alta disponibilidad e integridad transaccional en múltiples regiones geográficas.	Una startup típicamente migraría de Cloud SQL a Spanner solo después de que el producto esté bien establecido en el mercado y comience a atraer a usuarios de diferentes partes del mundo. Por ejemplo, una aplicación de viajes o de comercio electrónico que inicialmente usaba Cloud SQL ahora necesita procesar reservas o pedidos de usuarios en Norteamérica, Europa y Asia simultáneamente sin conflictos de datos. La consistencia transaccional global de Spanner ofrece soporte a esta escala.

Orquestación de agentes: la función ejecutiva

La orquestación es el núcleo operativo que guía a un agente a través de una tarea de múltiples pasos. Para cualquier proceso que requiera más de una sola acción, determina qué herramientas se necesitan, en qué secuencia deben usarse y cómo deben combinarse los datos generados para alcanzar un objetivo final.

Como función ejecutiva del agente, la orquestación puede aplicarse para asumir responsabilidad por la planificación y la toma de decisiones. Y, al automatizar procesos de negocio complejos, optimiza la eficiencia de los equipos pequeños de las startups.



Conceptos de orquestación y arquitectura cognitiva

Un patrón de orquestación común y efectivo es **ReAct (Razonamiento + Acción)**, un framework que combina sinérgicamente las capacidades de razonamiento y acción de los grandes modelos de lenguaje.[4]

ReAct establece un ciclo dinámico de múltiples etapas, donde el modelo genera trazas de razonamiento (pensamientos) y acciones específicas de la tarea, de manera intercalada. Esto permite una mayor sinergia: el razonamiento ayuda al modelo a rastrear y actualizar los planes de acción, mientras que las acciones recopilan información de herramientas externas para alimentar el proceso de razonamiento.

Así es como funciona:

1. **Razonamiento**: El agente evalúa el objetivo y el estado actual, formando una hipótesis sobre el siguiente mejor paso a seguir y si es necesario usar una herramienta.
2. **Acción**: El agente selecciona e invoca la herramienta adecuada.
3. **Observación**: El agente recibe los datos de salida de la herramienta. Esta nueva información se integra en el contexto del agente y alimenta el siguiente paso de razonamiento del ciclo.

4. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. Publicado como artículo de conferencia en ICLR 2023.

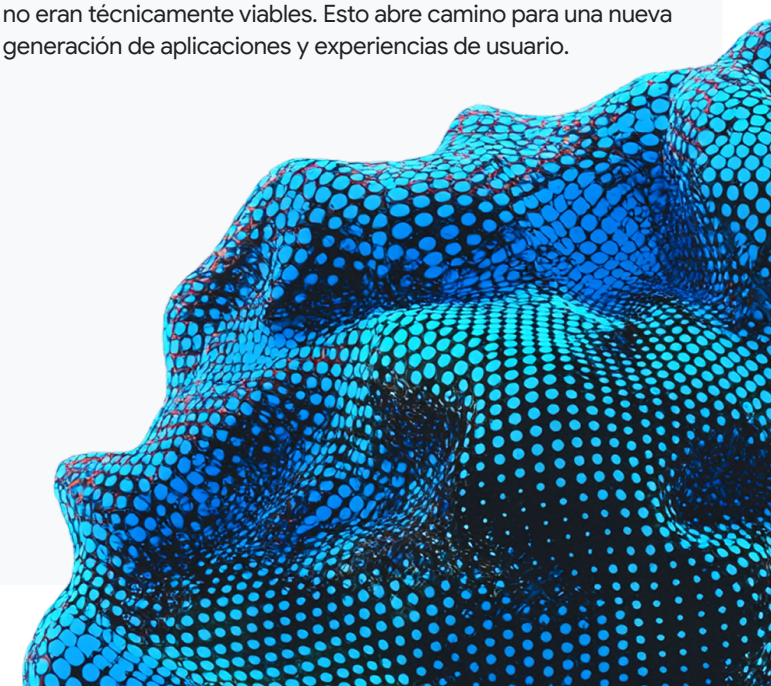
Ejemplo: Procesamiento de un reembolso con orquestación ReAct

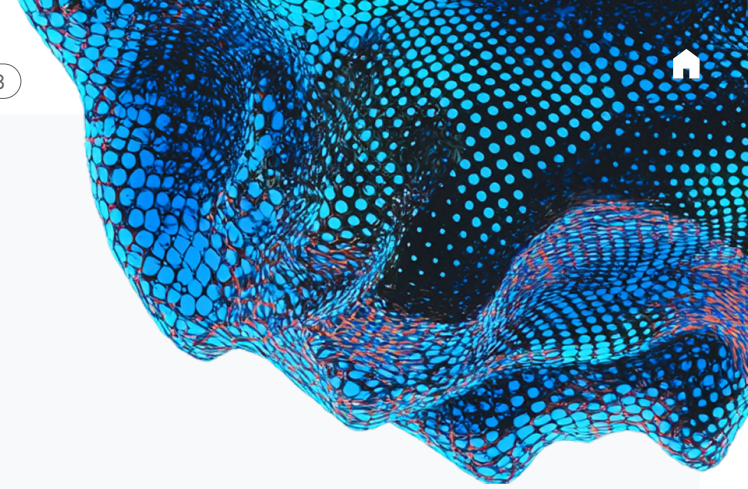
- **Razonamiento:** El usuario quiere un reembolso. El primer paso es comprender las reglas de la empresa para los reembolsos.
- **Acción:** Usar la herramienta semantic_search para consultar la base de conocimientos interna con el término “política de reembolso”.
- **Observación:** La herramienta devuelve: “Se encuentran disponibles reembolsos completos para todos los productos dentro de los 30 días posteriores a la fecha de compra”.
- **Razonamiento:** La política exige la fecha de compra. Esta información para el pedido específico del usuario debe provenir del CRM.
- **Acción:** Llamar a la función get_order_details de la herramienta CRM con el ID del usuario.
- **Observación:** La función devuelve un objeto de pedido, incluyendo purchase_date: ‘2025-07-20’.
- **Razonamiento:** La fecha actual es 29 de julio de 2025. La compra fue hace 9 días, lo cual está dentro de la ventana de 30 días. Se cumplen los criterios y se puede iniciar el reembolso.
- **Acción:** Llamar a la herramienta process_refund con el ID del pedido y el monto del reembolso.
- **Observación:** La herramienta devuelve status: ‘success’.
- **Respuesta final:** “Tu reembolso ha sido procesado con éxito. Deberías ver el monto acreditado en tu cuenta en un plazo de 3 a 5 días hábiles”.

Casos de uso

- **Onboarding automatizado de clientes:** Un agente podría ser orquestado para guiar a un nuevo usuario durante la configuración inicial. Primero, podría usar una herramienta para crear una nueva cuenta vía API, luego usar una herramienta send_email para enviar un mensaje de bienvenida y, finalmente, usar una herramienta para consultar la base de datos y verificar que el usuario haya completado su primera acción y, en caso de no haberlo hecho, activar un recordatorio.
- **Monitoreo y remediación proactiva del sistema:** Una alerta de monitoreo puede activar la orquestación. Primero, el agente obtiene contexto adicional usando una herramienta para consultar los registros en Cloud Logging. A partir de ese análisis, decide usar una herramienta kubectl para reiniciar un pod específico en GKE y, finalmente, usa una herramienta slack_notification para enviar una notificación en el canal de guardia a través de Slack.
- **Cualificación avanzada de prospectos (leads):** Un agente de ventas podría ser orquestado para enriquecer el email de un nuevo lead con datos corporativos desde una API. Invoca una herramienta para consultar el CRM interno y verificar si el lead ya es un cliente existente. Finalmente, procesa la información recopilada para decidir si asigna el lead a un representante de ventas senior o lo agrega a una campaña de emails de captación.

Dominar la orquestación es la clave para ir más allá de los agentes simples que ejecutan una única tarea. Cuando se hace correctamente, permite crear sistemas sofisticados y autónomos capaces de resolver problemas que, anteriormente, no eran técnicamente viables. Esto abre camino para una nueva generación de aplicaciones y experiencias de usuario.





Entorno de ejecución: despliegue de agentes a gran escala

El despliegue de un prototipo de agente funcional en un entorno de producción requiere una infraestructura de ejecución robusta. Este entorno de ejecución facilita el despliegue de agentes a gran escala, transformando un prototipo en un producto fiable capaz de gestionar requisitos operativos complejos, tales como la seguridad, el balanceo de carga y el manejo de errores, especialmente ante un crecimiento impredecible de la base de usuarios.

Conceptos y arquitectura del entorno de ejecución

Un entorno de ejecución de grado de producción para agentes de IA debe ofrecer varias capacidades fundamentales

- **Escalabilidad:** La infraestructura debe escalar automáticamente para manejar cargas variables, desde cero hasta millones de solicitudes. Esto incluye tanto el balanceo de carga basado en solicitudes como el escalado automático basado en recursos para gestionar las demandas computacionales de manera eficiente.
- **Seguridad:** El entorno de ejecución debe garantizar un procesamiento seguro y encargarse de gestionar la identidad, los controles de acceso a la red y los canales de comunicación seguros (como TLS) para proteger tanto al agente como a los datos a los que accede.
- **Fiabilidad y observabilidad:** El sistema debe incluir mecanismos para el manejo de errores, reintentos automáticos y un monitoreo integral. Esto implica registrar las acciones del agente y los datos generados por las herramientas, así como recopilar métricas sobre el rendimiento y la utilización de recursos para diagnosticar y resolver problemas.

Casos de uso

Tu elección del entorno de ejecución impacta directamente en la carga operativa y en tu capacidad para escalar

- **Vertex AI Agent Engine:** Una startup en etapa semilla con un equipo de ingeniería pequeño despliega su primer agente de atención al cliente y pasa de un prototipo funcional a un endpoint de producción escalable y seguro en días en vez de semanas.
- **Cloud Run:** Una startup que experimenta un crecimiento rápido pero impredecible en su nueva funcionalidad impulsada por IA despliega su agente del ADK en esta arquitectura sin servidor. De este modo, solo pagan por la capacidad de cómputo cuando el agente procesa solicitudes activamente. Es una forma rentable de manejar picos de tráfico sin sobreaprovisionar la infraestructura.
- **Google Kubernetes Engine (GKE):** Una startup de Serie B con un equipo de ingeniería de plataforma consolidado y docenas de microservicios decide alojar su nuevo agente interno de automatización en su clúster de GKE existente. De esta manera, pueden utilizar los procesos de CI/CD, las políticas de seguridad y los paneles de monitoreo ya establecidos, y garantizar que el agente cumpla con los mismos estándares operativos que el resto de sus servicios en producción.

1.3 El rol de la fundamentación en sistemas de agentes

La credibilidad y utilidad de un agente dependen de su capacidad para generar respuestas precisas y confiables basadas en hechos verificables, un proceso conocido como fundamentación o grounding. Esta sección explora la evolución de las técnicas de fundamentación y brinda una hoja de ruta para crear agentes cada vez más sofisticados y fiables.

Comenzaremos con el patrón fundamental de RAG, que fundamenta a un agente recuperando texto basado en similitudes semánticas. Luego, analizaremos GraphRAG, que enriquece la fundamentación al comprender las relaciones explícitas entre los puntos de datos en un grafo de conocimiento. Finalmente, cubriremos Agentic RAG, donde el agente ya no es un receptor pasivo de información, sino un participante activo y con capacidad de razonamiento en el propio proceso de búsqueda, capaz de ejecutar estrategias de múltiples pasos para encontrar la mejor respuesta posible.

RAG: un primer paso fundamental

El primer paso en el camino hacia una fundamentación sofisticada es el patrón arquitectónico de RAG (Generación Aumentada por Recuperación). Este enfoque mejora las respuestas de un LLM al recuperar información relevante de una base de conocimientos externa antes de generar una respuesta. En vez de depender únicamente del conocimiento adquirido durante el entrenamiento, el agente realiza una búsqueda semántica para encontrar datos verificables, que luego se pasan al LLM como contexto. Esto garantiza una base de respuestas fundamentadas y verificables.

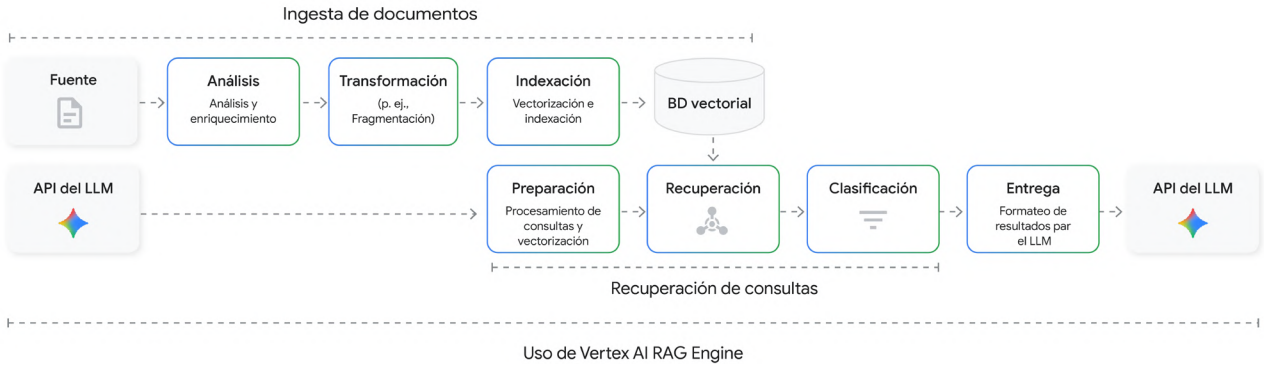
Aunque es un punto de partida fundamental, el proceso básico de “recuperación y generación” trata el conocimiento como un inventario plano de hechos aislados. Si bien es una técnica eficaz para consultas directas, resulta insuficiente ante preguntas complejas que exigen comprender las interconexiones semánticas entre los datos.

Beneficios de la RAG para los sistemas de agentes

- **Los agentes pueden acceder a la información más reciente:** Los datos obtenidos son más actuales que la última fecha de entrenamiento, lo que permite un comportamiento agéntico más oportuno y relevante.
- **Los agentes son más precisos:** La RAG reduce significativamente el riesgo de respuestas que podrían conducir a acciones agénticas incorrectas o inapropiadas.
- **Respuestas más rápidas:** Los embeddings vectoriales y las bases de datos especializadas permiten búsquedas semánticas ultrarrápidas en conjuntos de datos masivos. Esto permite a los agentes ofrecer decisiones más oportunas y con mayor capacidad de respuesta.
- **Más conciencia del agente:** El flujo de trabajo de RAG, que consiste en la ingesta, el análisis, la fragmentación, la vectorización, el almacenamiento y la recuperación, se puede aplicar a textos, imágenes y otros tipos de datos. Con esta comprensión más profunda, los agentes pueden realizar tareas de razonamiento más complejas en varios pasos.

Vertex AI Search es la solución RAG gestionada y lista para usar de Google Cloud. Esta solución agiliza el proceso de integración de fuentes de datos y permite además el uso de herramientas de código abierto o de terceros. Por otro lado, Vertex AI RAG Engine brinda un framework de datos para desarrollar aplicaciones de LLM aumentadas por contexto capaces de generar respuestas precisas, controladas y alineadas con conocimientos y políticas específicas. Es la arquitectura ideal para aplicaciones críticas de startups, como atención al cliente, gestión del conocimiento interno y tareas relacionadas con el cumplimiento normativo.

Vertex AI RAG Engine en acción



Herramienta

Usa Vertex AI Search y Vertex AI RAG Engine para fundamentar las respuestas utilizando tu contenido propietario.

Consejo experto

Usa la API de verificación de fundamentación para comprobar si las respuestas de la IA se basan en información fundamentada y actualizada.

Bases de datos vectoriales: búsqueda por significado

La capacidad de buscar por significado, no solo por palabras clave, es posible gracias a los embeddings vectoriales. Estas representaciones numéricas capturan la esencia conceptual de los datos (como texto e imágenes), lo que permite que un sistema encuentre información relevante sin importar cómo se formule la pregunta. Las bases de datos vectoriales son la infraestructura que permite esto a gran escala. Son sistemas altamente especializados diseñados para almacenar, indexar y consultar millones de estos embeddings con la latencia ultrabaja que requiere un sistema de agentes con alta capacidad de respuesta.

Así es como funciona:

1. **Los datos se transforman en embeddings vectoriales:** El modelo de Machine Learning coloca los elementos semánticamente similares muy cerca unos de otros dentro de un espacio vectorial multidimensional.
2. **Almacenamiento e indexación:** La base de datos vectorial almacena estos embeddings y construye índices especializados para permitir búsquedas rápidas y eficientes por solicitud.
3. **Consulta:** La consulta del usuario se convierte en un embedding utilizando el mismo modelo. Luego, la base de datos busca en su índice los embeddings que están más cerca del embedding de la consulta, recuperando efectivamente la información semánticamente más relevante para fundamentar la respuesta del modelo.

Caso de uso

Mejora de la atención al cliente

Una empresa de calzado utiliza una base de datos vectorial con búsqueda semántica para alimentar su chatbot de atención al cliente:

- Las descripciones de productos, la información de garantía y las preguntas frecuentes se convierten en embeddings y se almacenan.
- La base de datos vectorial comprende que “bueno para personas con pies anchos” está semánticamente relacionado con conceptos como “calce ancho”, “extra ancho” o “cómodo para pies anchos”.
- Recupera las recomendaciones de productos relevantes y ofrece una experiencia mucho mejor al cliente.

Compara esto con el supuesto caso en que la empresa de calzado utilizara una **base de datos tradicional**. Una consulta usando `LIKE '%good for people with wide feet%'` no devolvería ningún resultado porque esa frase exacta no existe en la base de datos.

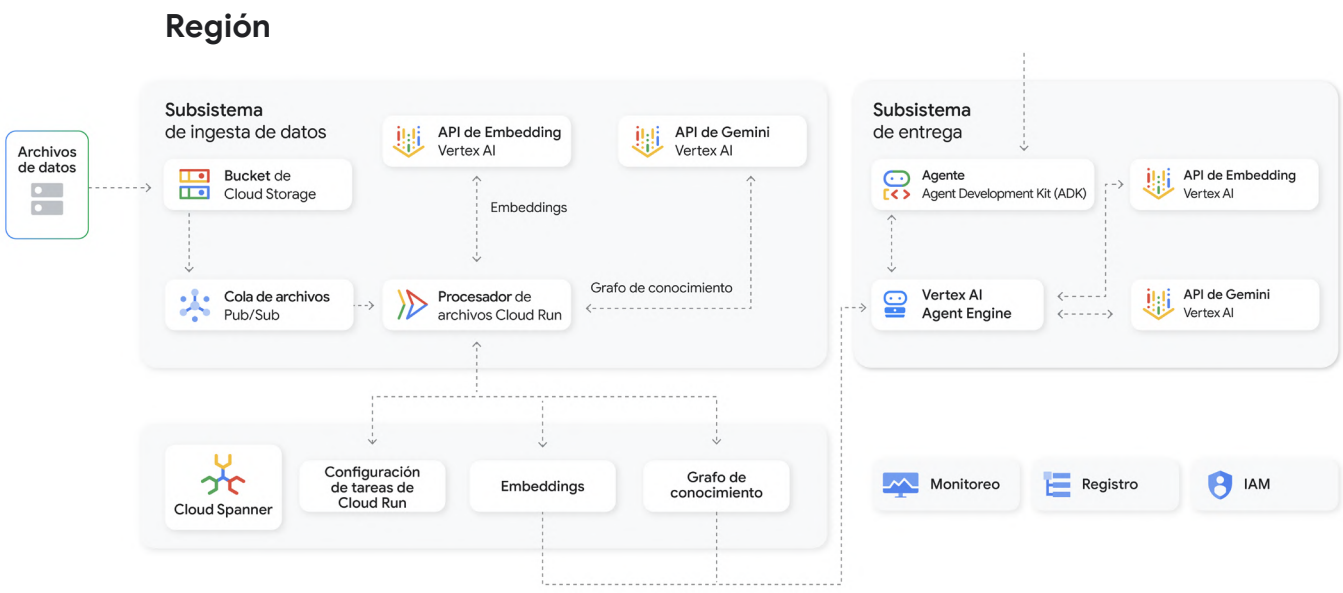
GraphRAG: una fundamentación más inteligente

El GraphRAG crea un grafo de conocimiento de modo que, en lugar de solo hacer coincidir frases similares, permite a tu agente comprender cómo se relacionan los conceptos.

Caso de uso

Un asistente de IA médico que necesita conocer **"síntomas → causas → tratamientos"** y no solo recuperar fragmentos de texto relacionados.

La jerarquía del conocimiento en GraphRAG



Agentic RAG: razonamiento dinámico y búsqueda inteligente

El enfoque más poderoso para la fundamentación es Agentic RAG, una técnica que te ayuda a transformar al agente de un receptor pasivo de datos recuperados a un participante activo y pensante en la búsqueda de conocimiento. Siguiendo estructuras como ReAct, el agente puede analizar una consulta compleja, formular un plan de múltiples pasos y ejecutar diversas llamadas a herramientas en secuencia para encontrar la mejor información posible.

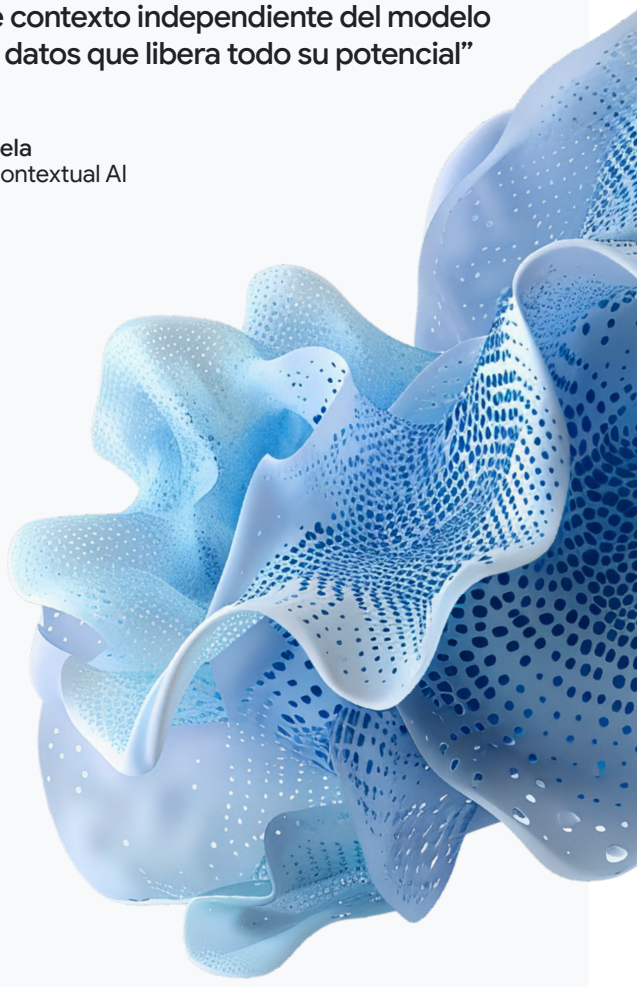
Un excelente ejemplo de este patrón agéntico es la fundamentación con Búsqueda de Google. Puedes usar los modelos de la familia Gemini 2.5 para integrar el razonamiento avanzado, permitiéndoles intercalar las capacidades de búsqueda con procesos internos de pensamiento para responder consultas complejas, con múltiples saltos de lógica, y ejecutar tareas de largo alcance. El agente puede ayudar a manejar todo el flujo de trabajo automáticamente: analiza el pedido, formula y ejecuta consultas de búsqueda precisas, y sintetiza una respuesta final fundamentada con sus fuentes.

Un agente creado con los modelos de la familia Gemini hace mucho más que un simple reconocimiento de contenido, resuelve de forma activa problemas en múltiples pasos. Por ejemplo, un agente podría:

- Analizar una foto para identificar una especie específica de planta y luego buscar de forma autónoma las instrucciones detalladas de cuidado.
- Procesar una transmisión de audio de una llamada de atención al cliente para no solo transcribir las palabras, sino también detectar cómo se siente el cliente, por ejemplo, si se siente frustrado, para escalar el ticket a quien corresponde.

Esta capacidad de percibir y razonar a través de diferentes tipos de datos transforma al agente de un mero procesador de datos a una herramienta de resolución de problemas que entiende e interactúa con el mundo de una manera más completa.

> “La creencia convencional era que el rendimiento de los modelos fundacionales mejoraría exponencialmente, pero estamos llegando a un punto de inflexión donde esa curva se está aplanando y la verdadera diferenciación radica en la especialización y la ingeniería de contexto. El Agentic RAG constituye un pilar central de esa capa de contexto que permite que los agentes de IA busquen, recuperen y razonen iteractivamente sobre los datos de referencia antes de generar una respuesta final.
>
> El futuro es multi-LLM: diferentes modelos para diferentes tareas, conectados por una capa de contexto independiente del modelo y de los datos que libera todo su potencial”
>
> **Douwe Kiela**
> CEO de Contextual AI



Ejemplo: Verificación de inventario en tiempo real

Define una función llamada check_inventory que tome un product_ID y devuelva los niveles de stock actuales desde tu sistema de inventario en tiempo real. De manera similar, otra función, check_warranty_status, podría tomar un product_ID y devolver la información de su garantía directamente desde tu sistema de gestión de garantías.

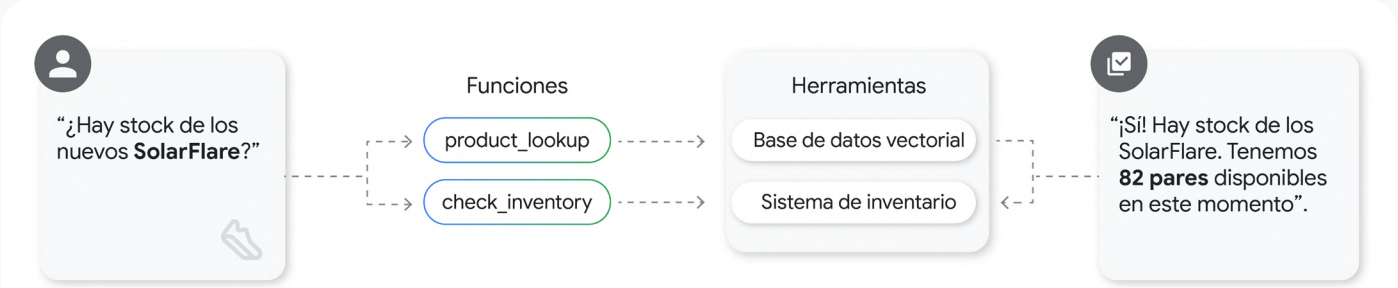
Luego, cuando un cliente pregunta sobre la disponibilidad de un producto específico, el agente de IA:

1. **Identifica el producto:** Utiliza la búsqueda semántica (alimentada por la base de datos vectorial) para identificar con precisión el modelo específico de calzado por el que pregunta el cliente, incluso si lo describe de forma vaga.
2. **Activa la acción:** Reconoce la necesidad de obtener información de stock en tiempo real y utiliza la llamada a funciones para invocar tu función check_inventory.
3. **Brinda una respuesta en tiempo real:** La función check_inventory se ejecuta, obtiene los datos de stock en vivo de tu sistema de inventario y se los devuelve al agente de IA. El agente luego le proporciona al cliente una respuesta inmediata y precisa sobre la disponibilidad.

Esta combinación de recuperación (saber qué información es relevante) y acciones (realizar operaciones en tiempo real) hace que tus agentes de IA sean más inteligentes, rápidos y mucho más útiles.

Consejo experto

Usa Vertex AI y la búsqueda vectorial de Google Cloud para agregar esta funcionalidad a tu agente.



El flujo de trabajo de Agentic RAG en Google Cloud

Google Cloud brinda servicios gestionados que manejan todo el flujo de trabajo de Agentic RAG:

- **Generación de embeddings e indexación:** El primer paso es convertir tus datos en embeddings vectoriales. Una opción es el modelo Gemini Embedding, que está disponible tanto en Vertex AI como en la API de Gemini y es compatible con más de 100 idiomas. Este y otros modelos forman parte de la suite más amplia de las API de embeddings de Vertex AI.
- **Almacenamiento e indexación:** Los embeddings vectoriales luego se almacenan y se indexan en Vertex AI Vector Search. Se trata de una base de datos vectorial de alto rendimiento y totalmente gestionada que crea automáticamente los índices especializados necesarios para búsquedas rápidas y eficientes de similitudes a gran escala.
- **Recuperación y razonamiento:** Cuando un usuario envía una consulta, esta se convierte en un embedding y Vertex AI Vector Search la utiliza para encontrar la información más relevante. Este contexto específico luego se pasa al LLM para generar la respuesta final fundamentada.

Consejo experto

Usa el enfoque de recuperación y reclasificación. Gestiona el equilibrio entre un búsqueda exhaustiva (encontrar todos los documentos relevantes) y la precisión (garantizar que cada documento recuperado sea relevante) utilizando el enfoque de dos pasos "recuperación y reclasificación" (mostrado en este ejemplo de agentic_rag). En primer lugar, se amplía el alcance de la búsqueda al configurar Vertex AI Vector Search para recuperar un conjunto de documentos candidatos mayor al estrictamente necesario. En segundo lugar, ese conjunto más grande se envía al LLM o a un servicio de reclasificación especializado, el cual identifica los documentos más relevantes y descarta aquellos que sean irrelevantes o semánticamente opuestos.

De la recuperación al razonamiento: una ventaja estratégica

El Agentic RAG representa un salto fundamental, pasando de la simple recuperación de información a la resolución genuina de problemas. Al permitir que el agente sea un participante activo y pensante, los desarrolladores pueden diseñar sistemas capaces de ejecutar consultas complejas de varios pasos y tareas a largo plazo, dos de las características centrales de las capacidades agénticas de la nueva generación.

Para una startup, dominar este enfoque es fundamental para crear un producto auténticamente inteligente y competitivo, capaz de impulsar experiencias de usuario y flujos de trabajo novedosos

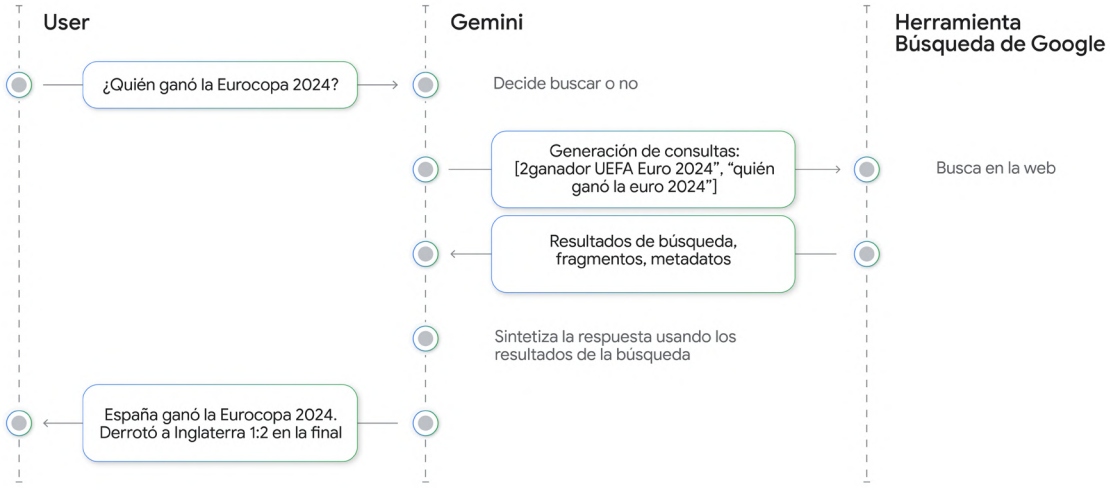
Otros métodos de fundamentación

Si bien la RAG es una técnica base, Vertex AI ofrece otras formas de garantizar que tus agentes ofrezcan respuestas precisas y confiables. Por ejemplo.

- **Fundamentación con Búsqueda de Google:** Conecta tu modelo al conocimiento mundial y a una amplia variedad de temas.
- **Fundamentación con Google Maps:** Usa los datos de Google Maps con tu modelo para brindar respuestas más precisas y contextualizadas.
- **Fundamentación de Gemini con tus datos:** Usa RAG para conectar tu modelo con los datos de tu sitio web o tus conjuntos de documentos.

Ejemplo: Fundamentación con Google Search

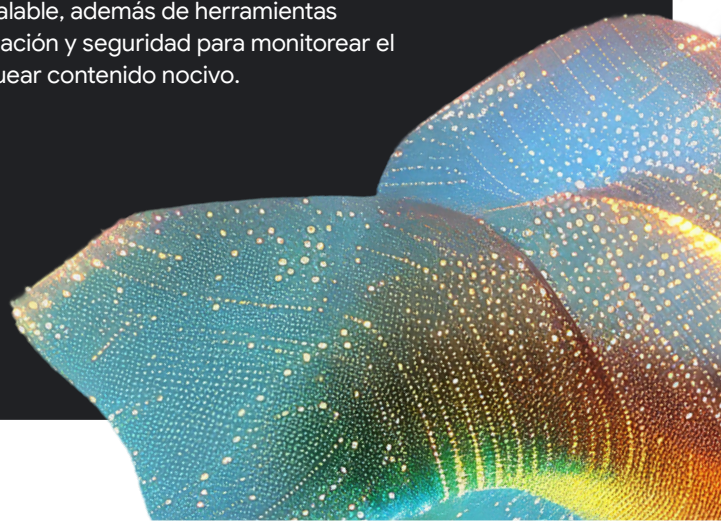
Cuando habilitas la herramienta Google Search, el modelo controla todo el flujo de trabajo de búsqueda, procesamiento y citación de información de forma automática.

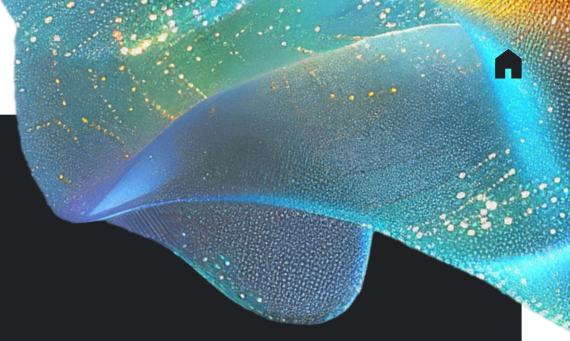


1. **Prompt del usuario:** La aplicación envía el prompt a la API de Gemini con la herramienta Google Search habilitada.
2. **Análisis del prompt:** El modelo analiza el prompt y determina si una búsqueda en Google puede mejorar la respuesta.
3. **Búsqueda en Google:** Si es necesario, el modelo genera automáticamente una o varias consultas de búsqueda y las ejecuta.
4. **Procesamiento de los resultados de búsqueda:** El modelo procesa los resultados de la búsqueda, sintetiza la información y formula una respuesta.
5. **Respuesta fundamentada:** La API devuelve al usuario una respuesta final, fácil de entender, que está fundamentada en los resultados de la búsqueda. Esta respuesta incluye la respuesta de texto del modelo y metadatos de fundamentación que contienen las consultas de búsqueda, los resultados encontrados en web y las citas.

Puntos clave: cómo elegir los componentes de tu agente de IA

Tu objetivo	Mejor opción
Elegir la inteligencia central del agente.	Selecciona un modelo (p. ej., Gemini) basado en tu caso de uso y haz un ajuste fino con tus datos específicos.
Hacer que tu agente sea fiable y se base en hechos.	Usa técnicas de fundamentación como la RAG con una base de datos vectorial para que verifique los hechos, y no solo adivine.
Gestionar una tarea compleja de múltiples pasos.	Usa la orquestación para crear un plan que determine qué herramienta usar, en qué orden y cómo combinar los resultados.
Conectarte a datos y servicios públicos en tiempo real.	Usa extensiones prediseñadas para conectarte fácilmente a las API de terceros.
Conectarte de forma segura a tus propias herramientas internas.	Escribe funciones personalizadas para darle al agente acceso controlado a tus bases de datos privadas, CRM u otros sistemas internos.
Desplegar un producto fiable y seguro a gran escala.	Usa un tiempo de ejecución gestionado para una infraestructura escalable, además de herramientas integradas de evaluación y seguridad para monitorear el rendimiento y bloquear contenido nocivo.





¿Todo listo para convertir tu visión de IA en realidad? Estamos aquí para ayudarte.

Aprende a desarrollar más aplicaciones de IA generativa con las sesiones bajo demanda de Startup School.

Empieza ahora

Recibe hasta $350,000 USD en créditos de Google Cloud con el programa Google Cloud for Startups.

Solicita la inscripción ahora

Habla con nuestro equipo especializado en startups.

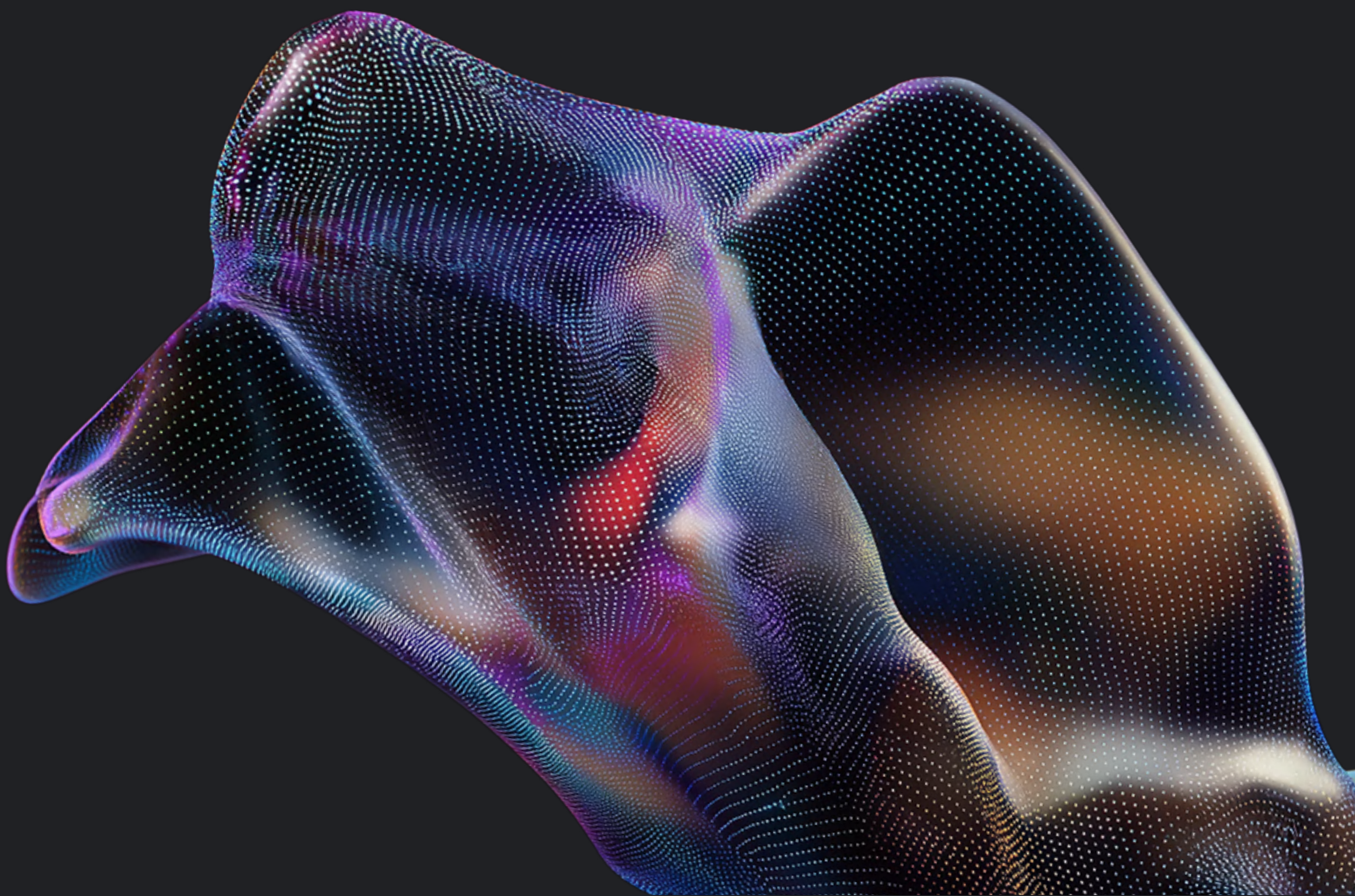
Comunícate con nosotros

Mantente al tanto de las últimas novedades y recibe todas nuestras actualizaciones suscribiéndote al boletín informativo de Google Cloud Startup.

Suscribirse

Sección 2

Cómo crear agentes de IA



La sección anterior definió los componentes fundamentales de un sistema de agentes moderno: un modelo de razonamiento central, un conjunto de herramientas para habilitar la acción, opciones de arquitectura de datos para mantener la memoria del agente a corto y largo plazo, un mecanismo de fundamentación para garantizar la precisión factual de la información y opciones de despliegue.

Una vez establecido este framework conceptual, ahora podemos pasar a centrarnos en cómo crear un agente. Esta sección presenta una guía práctica, con una opinión definida, sobre las decisiones de arquitectura involucradas en la creación de un agente listo para la producción.

El ecosistema de Google Cloud, abierto y flexible, reconoce que la industria ofrece muchos frameworks excelentes. Aquí, sin embargo, nuestra intención es centrarnos en el ADK (Agent Development Kit), una implementación completa y robusta que encaja perfectamente en el ecosistema de Google Cloud. También damos algunas recomendaciones específicas basadas en estándares abiertos de la industria como el Protocolo de Contexto de Modelo (MCP) y el protocolo Agent2Agent (A2A).

Consejo experto

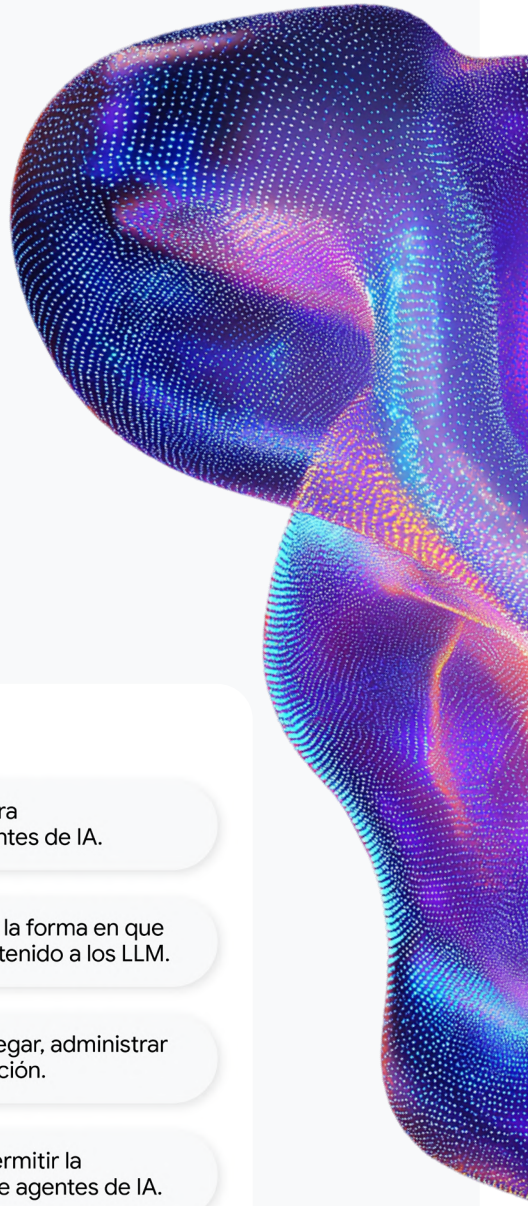
Antes de crear tu agente, explora +1000 casos reales de uso de agentes adoptados por organizaciones líderes del mundo.

¿Prefieres audio? Escucha la versión en podcast de esta sección, creada con NotebookLM.

Este podcast fue creado usando NotebookLM con el siguiente prompt: “Como presentador de podcast, genera un episodio práctico dirigido a desarrolladores y fundadores técnicos. El podcast debe presentar el Agent Development Kit (ADK) como una solución a los desafíos comunes en la creación de agentes. Debe detallar los beneficios centrales del ADK y explicar sus tipos de agentes principales, como el LlmAgent inteligente y los Agentes de Flujo de Trabajo estructurados (p. ej., Secuencial, Paralelo y en Bucle).

“El podcast debe luego cubrir el ecosistema más amplio, como el Protocolo de Contexto de Modelo (MCP), Vertex AI Agent Engine para el despliegue y el protocolo Agent2Agent (A2A) para la comunicación. Menciona brevemente herramientas alternativas como Gemini Enterprise, Firebase Studio y Gemini CLI, y concluye con un resumen y un llamado a la acción que invite a los oyentes a explorar los recursos de Google para startups”.

2.1 Un kit de herramientas completo para crear agentes de IA



A la hora de crear un agente de IA personalizado para tu startup, los fundadores se enfrentan a un dilema importante: velocidad de desarrollo versus flexibilidad.

En un extremo, tienes soluciones fáciles de usar como plataformas con poco código o productos ya listos. Son rápidos de implementar, pero te dan menos control, lo que los hace mejores para problemas comunes. En el otro extremo, tienes frameworks altamente flexibles y la opción de crear todo desde cero. Si bien ofrecen la máxima capacidad de personalización, exigen muchos más recursos de desarrollo y una profunda experiencia técnica. El ADK se sitúa en el medio de este escenario de desarrollo.

Explora el ADK

Principales componentes para crear agentes de IA

Agent Development Kit	Herramienta de código abierto para construir, evaluar y desplegar agentes de IA.
Protocolo de Contexto de Modelo	Protocolo abierto que estandariza la forma en que las aplicaciones proporcionan contenido a los LLM.
Vertex AI Agent Engine	Plataforma gestionada para desplegar, administrar y escalar agentes de IA en producción.
Protocolo Agent2Agent (A2A)	Estándar abierto diseñado para permitir la comunicación y colaboración entre agentes de IA.

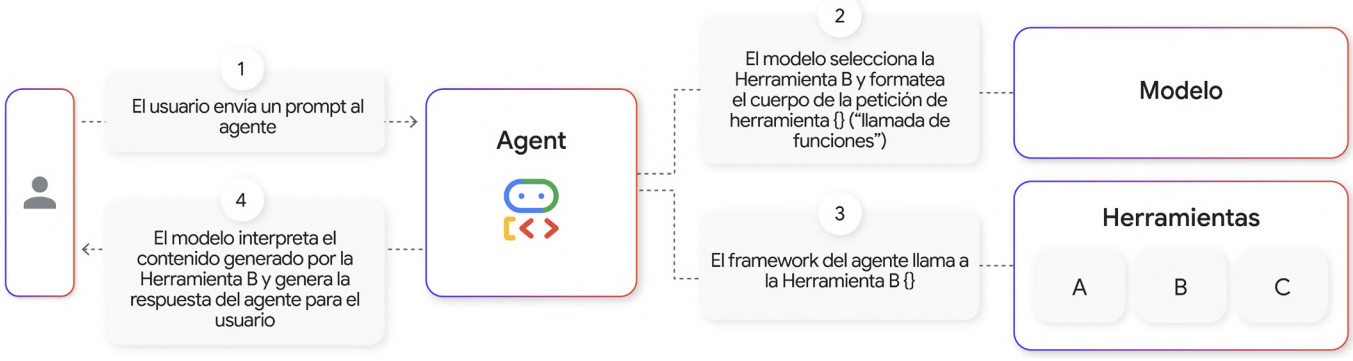
Desarrollo con ADK

Si necesitas más potencia que una simple herramienta con poco código, pero quieres un proceso de desarrollo acelerado, el ADK te ofrece el control para desarrollar un sistema único de agentes colaborativos y simplifica los desafíos técnicos complejos. Por ejemplo, protocolos específicos como MCP y A2A facilitan la ampliación de capacidades de los agentes (como analizaremos a continuación).

Además, el ADK complementa las herramientas que tu equipo ya pueda estar utilizando y se integra con el ecosistema más amplio de Google Cloud, por lo que no quedas atado a un único enfoque. Esta flexibilidad también se aplica a la forma en que ejecutas tus agentes una vez creados, con opciones para desplegar en un servicio totalmente gestionado como Vertex AI Agent Engine o en una plataforma versátil sin servidor como Cloud Run. Lo importante es elegir la base adecuada para tus necesidades operativas específicas.

> **Nos centramos en construir un ecosistema floreciente de agentes de IA. El Agent Development Kit, de código abierto, tuvo más de un millón de descargas en menos de cuatro meses".**
>
> **Sundar Pichai**
> CEO de Google y Alphabet

Desarrollar flujos de trabajo complejos es más fácil con el ADK



Lo que puedes hacer con el ADK

1. Desarrollar sistemas de IA complejos y colaborativos

El ADK es multiagente por diseño. Es fácil crear soluciones de IA altamente especializadas que automatizan flujos de trabajo complejos de múltiples pasos. Además, con una orquestación flexible (secuencial, paralela o dinámica), puedes comenzar con automatizaciones simples y evolucionar hacia sistemas altamente adaptativos. Por ejemplo, puedes desarrollar un sistema inteligente de gestión de proyectos con un "Agente de Desglose de Tareas" que delega subtareas a "Agentes de Generación de Código", "Agentes de Diseño" y "Agentes de Documentación" especializados.

2. Integrar la IA en herramientas, agentes y flujos de trabajo existentes

El ADK está construido en torno a un rico ecosistema de herramientas que permite que tus agentes interactúen con todas tus herramientas y datos existentes. Puedes conectar tus agentes a herramientas de productividad que ya usas, como Notion, Slack o un CRM, así como a frameworks de herramientas, como LangChain y LlamaIndex, o frameworks de agentes como LangGraph o CrewAI. Las herramientas se pueden compartir vía MCP y los agentes que crees se pueden compartir vía A2A. De esta manera, puedes inyectar inteligencia artificial en cada fase de tus operaciones, mejorando las herramientas y los sistemas que ya tienes sin necesidad de reformular toda la arquitectura.

3. Asegurar calidad y fiabilidad desde el primer día

Para ganarse la confianza de los usuarios, es fundamental probar y evaluar cuidadosamente tus agentes antes de desplegarlos a producción. Las herramientas integradas de observabilidad y evaluación del ADK te ayudan a:

- **Iterar rápidamente:** Antes del despliegue, puedes probar sistemáticamente cómo responden tus agentes a diversos escenarios y qué tan bien ejecutan tareas complejas.
- **Depurar el comportamiento del agente:** Inspecciona la traza de ejecución completa de tu agente, incluido su razonamiento (pensamientos), llamadas a herramientas y observaciones para comprender su proceso de toma de decisiones y depurar flujos de trabajo complejos de múltiples pasos.
- **Comparar los agentes:** Evalúa diferentes diseños de agentes o actualizaciones de modelos comparándolos con métricas predefinidas, usando un enfoque basado en datos para mejorar continuamente el rendimiento del agente.

Esto lleva tu proceso más allá de las simples pruebas superficiales basadas en la intuición (vibe-testing), lo que te permite lanzar rápidamente agentes de nivel profesional, ganar la confianza del usuario a través de una fiabilidad probada e iterar con la seguridad que dan los datos.

4. Escalar la IA con confianza

A medida que creces, tus soluciones de IA deben escalar de forma fluida sin convertirse en un cuello de botella. El ADK acelera el camino a producción utilizando AgentOps (se describe en detalle a continuación) para cerrar la brecha entre el desarrollo local y el despliegue. El framework expone a los agentes como servicios web estándar utilizando FastAPI, los cuales luego pueden ser contenerizados. Esto evita que tus desarrolladores tengan que construir un infraestructura personalizada de implementación. Pueden desplegar en cualquier lugar, desde pruebas locales hasta entornos de ejecución totalmente gestionados y escalables automáticamente, como Vertex AI Agent Engine o Cloud Run.

Núcleo del ADK: arquitecturas de agentes

Un paso fundamental en la construcción con el ADK es seleccionar la arquitectura de agente adecuada. Las distintas clases de agentes están diseñadas para diferentes patrones de ejecución, y tu elección determinará cómo razona y opera tu agente. Por lo general, se trata de buscar el equilibrio entre el poder flexible y no determinista de un LLM y el control predecible y determinista de la lógica con codificación rígida. Comprender la interacción entre estas clases de agentes es clave para desarrollar sistemas de IA robustos y eficaces.

Los tipos de agentes del ADK se organizan en tres categorías

1 Motor central: LLM

Determinismo: No determinista (flexible)

Este es el tipo de agente más común y generalmente se lo conoce simplemente como “Agente”. Utiliza un LLM como Gemini para el razonamiento complejo, la toma de decisiones dinámica y la comprensión del lenguaje natural. Constituye el núcleo de la mayoría de los agentes conversacionales y de resolución de problemas.

2 Agente de flujo de trabajo (SequentialAgent, ParallelAgent, LoopAgent)

Motor central: Lógica predefinida

Determinismo: Determinista (predecible)

Estos orquestadores controlan de forma determinista cómo se ejecutan otros agentes en patrones predefinidos. Se utilizan para procesos estructurados.

Agentes secuenciales (SequentialAgent)

Ejecuta subagentes en un orden fijo, y pasa los datos de salida de uno como datos de entrada al siguiente. El Agente A completa la Tarea 1, luego pasa sus datos de salida al Agente B para la Tarea 2.

Ejemplo: Quieres crear un agente que pueda resumir cualquier página web, utilizando dos herramientas: Page Contents y Summarize Page. Debido a que no puedes resumir de la nada, el agente siempre debe llamar a Get Page Contents antes de llamar a Summarize Page.

Obtén el código completo

Agentes paralelos (ParallelAgent)

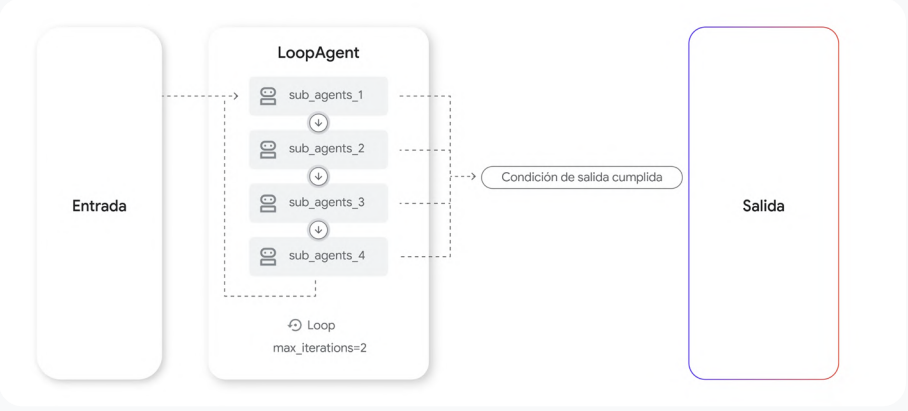
Este ejecuta múltiples subagentes simultáneamente. Se utiliza para la optimización del rendimiento cuando las tareas son independientes. El Agente A y el Agente B trabajan en subtareas independientes al mismo tiempo, y sus resultados se combinan.

Ejemplo: Para operaciones como la recuperación de datos de múltiples fuentes o cálculos pesados, la paralelización produce ganancias sustanciales de rendimiento. Es importante destacar que esta estrategia asume que no hay una necesidad inherente de estado compartido o intercambio directo de información entre los agentes que se ejecutan concurrentemente.

Obtén el código completo

Agentes de bucle (LoopAgent)

Un agente de flujo de trabajo ejecuta sus subagentes en un bucle (iterativamente). Ejecuta de forma repetida una secuencia de agentes durante un número específico de iteraciones o hasta que se cumpla una condición de salida. Usa el LoopAgent cuando tu flujo de trabajo implique repetición o refinamiento iterativo, como la revisión de código.



Ejemplo: Quieres crear un agente que pueda generar imágenes que contengan cantidades específicas de comida (p. ej., cinco bananas), utilizando dos herramientas: Generar Imagen, Contar artículos de comida. Debido a que quieres seguir generando imágenes hasta que se genere correctamente el número especificado de elementos, o después de un cierto número de iteraciones, debes crear tu agente usando un LoopAgent.

Obtén el código completo

3 Agente personalizado (Subclase de BaseAgent)

Motor central: Código Python personalizado

Determinismo: Puede ser cualquiera de los dos, según la implementación

Para requerimientos únicos y flujos de trabajo a medida que van más allá de un bucle de razonamiento estándar, puedes crear un agente personalizado heredando directamente de BaseAgent y escribiendo lógica Python personalizada para controlar su comportamiento. Este enfoque es necesario cuando las acciones de un agente no están determinadas por un LLM, sino por reglas específicas de codificación rígida.

Cómo se hace: Un desarrollador crea una nueva clase de Python que hereda de la clase BaseAgent. Luego, debe implementar el método _run_async_impl, que contiene la lógica única que ejecutará el agente. Este método tiene acceso completo al estado de la sesión y puede generar eventos para comunicarse con otros agentes o terminar un flujo de trabajo.

Consulta la guía de inicio rápido de Gemini Fullstack Agent Development Kit (ADK) para ver un ejemplo de cómo implementar un agente personalizado.

Orquestación del ADK: implementación del bucle ReAct

Como analizamos en la Sección 1, el paradigma ReAct es un patrón fundamental para los sistemas de agentes. El ADK proporciona las abstracciones y clases centrales necesarias para implementar este proceso dinámico y cíclico de una manera estructurada. Su LlmAgent está diseñado específicamente para ejecutar este bucle y gestionar las transiciones entre sus etapas principales:

- **Razonamiento (pensamiento):** La clase LlmAgent gestiona esta etapa internamente. Recibe el prompt del usuario y su estado interno actual, luego llama al modelo de lenguaje subyacente para formar una hipótesis y determinar la siguiente mejor acción.
- **Acción (uso de herramientas y delegación de agentes)**: El ADK habilita esta etapa a través de su sistema flexible de herramientas. Cuando el LlmAgent decide actuar, puede invocar una función de Python simple o, para tareas más complejas, delegar el trabajo a otro subagente especializado utilizando el patrón Agent-as-a-Tool.
- **Observación**: El ADK captura automáticamente el diccionario devuelto por la herramienta o el subagente y lo devuelve al LlmAgent. Estos datos de salida se convierten en la nueva información que el agente integra en su contexto y alimentan al siguiente paso de Razonamiento del ciclo.

Al ofrecer una implementación nativa de este patrón esencial, el ADK abstrae el código repetitivo, para que puedas transformar rápidamente el poderoso concepto de bucle ReAct en un agente funcional de múltiples pasos.

Herramientas del ADK: un framework para la acción agéntica

En el ADK, un agente puede usar herramientas para realizar acciones que van más allá de las capacidades nativas de su modelo de razonamiento central. Estas capacidades definidas permiten que un agente ejecute código, interactúe con sistemas externos y actúe fuera de su propio contexto de ejecución inmediato. Una herramienta es una función de Python (o un método de Java) que puede implementar una lógica autónoma o actuar como un wrapper para operaciones más complejas, como realizar llamadas a una API, usar un MCP para acceder a diversos sistemas externos o delegar una tarea a otro agente especializado de forma local o remota a través de A2A

Esta sección describe cómo diseñar herramientas efectivas y presenta la taxonomía de los tipos de herramientas disponibles.

Consejo experto

Para una análisis completo, incluidos los ejemplos de código y patrones de uso avanzados, consulta la documentación del ADK.

Diseño de herramientas efectivas: el contrato de API para el modelo

Para que un modelo use una herramienta correctamente, su definición debe servir como un **contrato de API** claro e inequívoco, compuesto por:

- **Firma de la función:** Usa nombres descriptivos para las herramientas y sus parámetros. Las anotaciones de tipo de Python son obligatorias, ya que proporcionan el esquema estructural que utiliza el modelo para generar argumentos válidos.
- **Docstring (el núcleo semántico):** Esta es la principal fuente de información semántica para el modelo. Un docstring bien escrito debe definir con precisión la función de la herramienta, los criterios de uso, los parámetros y el esquema de retorno esperado.
- **Esquema de retorno:** Una herramienta debe devolver un diccionario. Aunque no es un requisito sintáctico estricto, es una buena práctica incluir una clave de estado (status) (p. ej., success o error) en este diccionario. Esta estructura es esencial para que el agente distinga de forma fiable entre resultados exitosos y con error en la etapa de Observación y pueda razonar sobre cómo proceder.
- **Herramientas con estado y ToolContext:** Para las herramientas que necesitan leer o escribir en un estado de sesión persistente, se puede agregar un parámetro opcional tool_context: ToolContext a la firma de la función. El agente inyecta automáticamente este objeto, dándole a la herramienta acceso a un diccionario de estado de la sesión.

Consejo experto

Para conocer las mejores prácticas y ejemplos de cómo definir parámetros y esquemas de herramientas, estructurar prompts eficaces e implementar flujos de trabajo complejos de múltiples agentes, consulta el repositorio ADK Samples.

Una taxonomía de las herramientas del ADK

El ADK ofrece una arquitectura flexible para implementar herramientas, que van desde funciones simples hasta sistemas multiagente interoperables.

Conjuntos de herramientas: agrupación de capacidades relacionadas

Un patrón central del ADK es el Toolset, una clase que agrupa un conjunto de herramientas relacionadas en un objeto único y configurable (p. ej., BigQueryToolset, MCPToolset).

Herramientas de función personalizadas

Este es el método más directo para extender un agente con lógica propietaria.

- FunctionTool: El wrapper estándar para funciones síncronas de Python.
- LongRunningFunctionTool: Una herramienta especializada para tareas asíncronas o flujos de trabajo que requieren intervención humana.

Herramientas jerárquicas y remotas

El ADK permite la creación de sistemas complejos mediante la composición de agentes.

- **Agente como herramienta:** Un patrón de delegación donde un agente principal (padre) usa otro agente especializado. Esto permite que el padre invoque a otro agente, reciba una respuesta y conserve el control para gestionar futuras entradas. (Esto es distinto a un modelo de delegación de subagentes, donde la totalidad del control conversacional se pasa a un subagente y todas las entradas posteriores son manejadas por el subagente).
- RemoteA2aAgent: Para la comunicación entre agentes en diferentes procesos, el ADK ofrece la clase RemoteA2aAgent, que utiliza el protocolo **Agent2Agent** (A2A) para integrar sistemas distribuidos de manera transparente.

Herramientas preconstruidas e integradas

El ADK incluye un conjunto de herramientas y wrappers para acelerar el desarrollo.

- **Herramientas integradas:** Herramientas listas para usar como Google Search y Code Execution.
- **Conjuntos de datos de Google Cloud:** Integraciones ricas para servicios como Vertex AI Search y BigQuery.
- **Interoperabilidad de terceros:** Wrappers como LangchainTool y CrewaiTool permiten la reutilización directa de herramientas de ecosistemas de código abierto populares

Estandarización con el Protocolo de Contexto de Modelo

El Protocolo de Contexto de Modelo (MCP) es un estándar abierto emergente para conectar la IA y los LLM con fuentes de datos y herramientas externas. Permite conectar tus aplicaciones de IA a varias fuentes de datos y herramientas sin tener que crear integraciones punto a punto personalizadas para cada una.

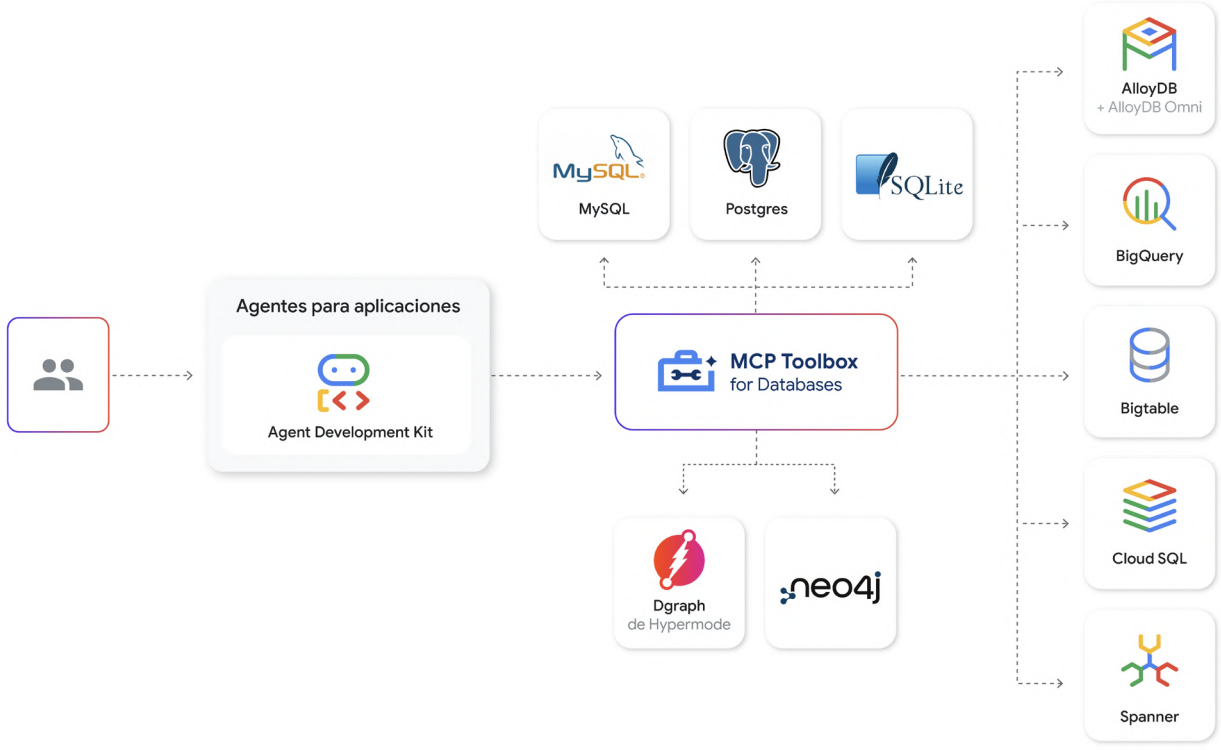
Con el ADK, tus agentes pueden participar en este ecosistema de dos maneras:

- **Uso de herramientas externas:** Un agente del ADK puede actuar como un cliente MCP, lo que le permite usar cualquier herramienta expuesta por un servidor MCP de terceros.
- **Exposición de herramientas nativas:** Los desarrolladores pueden empaquetar sus herramientas del ADK en un servidor MCP y permitirles estar disponibles de forma segura para cualquier otro agente o aplicación compatible con MCP.

Consejo experto

Usa el MCP Toolbox for Databases de código abierto para conectar tus agentes de manera fácil y segura a una gran variedad de fuentes de datos populares.

El MCP es como un adaptador universal para las herramientas y fuentes de datos de un agente.



Gestiona tus datos con los servicios de Google Cloud

Como se describió en la sección anterior, la memoria a largo plazo, la de trabajo y la transaccional desempeñan un papel distinto en la arquitectura de datos de un agente. Aquí explicamos cómo construirla. El ADK ofrece los patrones e integraciones necesarias para mapear esta arquitectura conceptual directamente a servicios de datos específicos y escalables de Google Cloud.

1. Base de conocimiento a largo plazo (fundamentación, contexto y analítica)

Se trata de la memoria permanente del agente, que combina una biblioteca de conocimientos con capacidad de búsqueda, un registro de las interacciones del usuario y un repositorio para analítica.

- **Vertex AI Search:** Funciona como la biblioteca de conocimiento que el agente puede consultar para información no estructurada. En el ADK, un VertexAISearchToolset permite que un agente fundamente sus respuestas obteniendo información relevante de un conjunto específico de documentos.
- **Firestore**: Funciona como la memoria de usuario persistente del agente. En el ADK, se utiliza para almacenar y recuperar el historial de conversaciones y el estado de las tareas de larga duración, lo cual genera una experiencia continua y personalizada que se puede reanudar en las distintas sesiones.
- **Cloud Storage**: Actúa como el sistema de archivos duradero del agente. El ADK lo usa como la fuente de verdad para documentos sin procesar (p. ej., PDF, imágenes) que luego son indexados por servicios como Vertex AI Search.
- **BigQuery:** Funciona como la base de datos analítica del agente. El BigQueryToolset en el ADK permite a los agentes responder preguntas ejecutando consultas analíticas complejas en grandes conjuntos de datos estructurados.

2. Memoria de trabajo (caché y estado de sesión)

Se trata de la memoria transitoria de alta velocidad del agente para gestionar el contexto inmediato de una conversación en vivo.

- **Memorystore:** Proporciona una caché de alta velocidad para el agente. En el ADK, su función principal es almacenar los resultados de las llamadas a herramientas frecuentes o costosas, para reducir drásticamente la latencia y los costos operativos.

3. Memoria transaccional (auditoría y ejecución fiable)

Se trata del registro persistente del agente destinado a grabar acciones críticas y cambios de estado con un alto nivel de integridad.

- **Cloud SQL:** Funciona como el sistema de registro fiable del agente. El ADK habilita patrones donde las herramientas registran sus acciones en Cloud SQL, creando una pista de auditoría permanente y compatible con ACID para cada acción importante impulsada por el agente.
- **Cloud Spanner**: Actúa como un backend globalmente consistente para acciones de misión crítica del agente. En una implementación avanzada del ADK, una herramienta que representa un proceso de negocio central (p. ej., process_global_order) dispararía una transacción en un sistema basado por Spanner para garantizar la integridad global.

4. La próxima frontera: memoria conversacional destilada

A medida que el historial de interacciones de un agente con un usuario aumenta con el paso de las semanas o los meses, enviar todo el contexto sin procesar al modelo para cada consulta se torna ineficiente y tiene un costo prohibitivo. Además, los modelos pueden perder precisión

La destilación de memoria es la próxima frontera. Esta técnica utilizar un LLM para sintetizar de forma dinámica y continua largos historiales de conversación y transformarlos en un conjunto compacto y estructurado de datos y preferencias esenciales. La memoria a largo plazo depurada que se obtiene es mucho más eficiente a la hora de recuperar y usar.

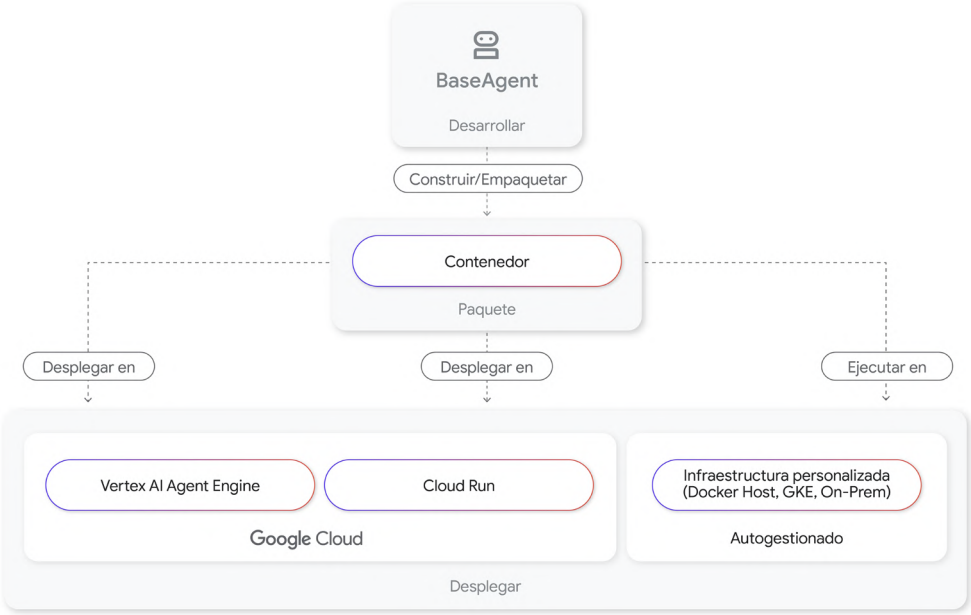
Se trata es un área que está en investigación, pero ya están surgiendo los primeros patrones. Un ejemplo es Vertex AI Memory Bank, un servicio gestionado en Vertex AI Agent Engine, que ofrece mecanismos para implementar la destilación de memoria:

- **Destilación automatizada:** Puede procesar de forma asíncrona los historiales de conversación para extraer y generar automáticamente una lista de hechos relevantes sobre el usuario (GenerateMemories).
- **Destilación dirigida por el agente:** Para tener más control, un agente puede usar la memoria como herramienta para decidir qué información específica es lo suficientemente importante como para escribirse explícitamente en el banco de memoria (CreateMemory).

Trabajar con una memoria destilada en lugar de usar el historial sin procesar resulta más escalable, eficiente y "humano"; es el enfoque ideal para la próxima generación de sistemas de agentes.

Despliegue en el entorno de ejecución gestionado con Vertex AI Agent Engine

El ADK es, por diseño, independiente de la plataforma de despliegue. La lógica central del agente que defines en Python está desacoplada de la infraestructura de servicio, lo que te permite desarrollar y probar de forma local para luego desplegar el mismo agente en diversos entornos de producción.



Para su despliegue, los agentes del ADK se exponen como servicios web estándar mediante FastAPI. El comando adk api_server encapsula automáticamente tu agente en un servidor de API listo para producción, el cual luego puede ejecutarse en contenedores.

Si bien este contenedor podría desplegarse en varios servicios de Google Cloud, los tres principales destinos de despliegue gestionado para los agentes del ADK son:

- **Cloud Run:** Una plataforma de computación gestionada para ejecutar tu agente como una aplicación basada en contenedores. Es una excelente opción para integrar tu agente en una arquitectura de microservicios existente o para casos de uso que requieran configuraciones de contenedor personalizadas.
- **Vertex AI Agent Engine:** Un servicio totalmente gestionado y de escalado automático en Google Cloud, diseñado específicamente para desplegar, gestionar y escalar agentes de IA creados con frameworks como el ADK. Ofrece una profunda integración con el ecosistema de Vertex AI para MLOps, monitoreo y seguridad.
- **Google Kubernetes Engine (GKE):** Este servicio gestionado de Kubernetes es la mejor opción si tienes una infraestructura existente basada en Kubernetes o estás tomando una decisión estratégica "desde el primer día" para priorizar la portabilidad a largo plazo, el control arquitectónico profundo y el ecosistema de código abierto de Kubernetes. Ofrece el control más granular sobre las redes, las cargas de trabajo con estado y el hardware especializado (como las GPU y TPU), lo que lo hace ideal para equipos con experiencia en ingeniería de plataformas o aquellos que construyen aplicaciones complejas de múltiples servicios que necesitarán escalar.

Nota

Es importante comprender la relación entre las herramientas de desarrollo de agentes de Google Cloud. **Vertex AI Agent Builder** es la plataforma integral que abarca todo el ciclo de vida del agente, desde el descubrimiento hasta el despliegue. Un componente central de esta plataforma es Vertex AI Agent Engine, el servicio gestionado diseñado específicamente para desplegar, administrar y escalar tus agentes en producción.

En el contexto de esta guía, cuando hablamos del entorno de ejecución en producción, nos referimos al Vertex AI Agent Engine.

Para las startups que utilizan el ADK, este es el destino de despliegue recomendado. Está optimizado específicamente para ser una solución rentable y de escalado automático, lo que ofrece el camino más fácil y directo hacia un agente escalable y listo para producción. Al ser un servicio totalmente gestionado, abstrae la infraestructura subyacente y permite que tus ingenieros se concentren en la lógica central del agente en lugar de en la sobrecarga operativa.

Como un servicio diseñado para cargas de trabajo agénticas, Vertex AI Agent Engine ofrece varios beneficios clave:

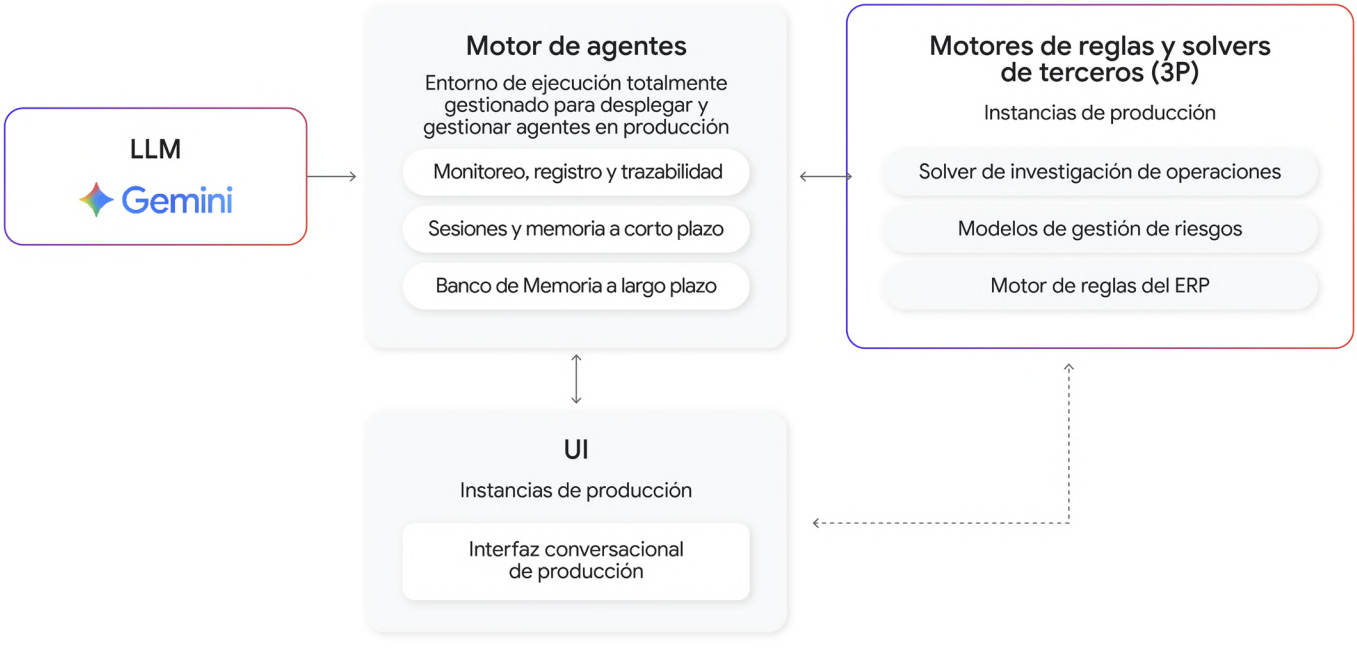
Capacidades centrales

- **Escalabilidad automatizada:** Maneja automáticamente el escalado para satisfacer cargas de usuarios variables.
- **Seguridad y autenticación:** Proporciona gestión integrada de identidad y acceso (IAM).
- **Independiente del framework:** Soporta agentes creados con diversos frameworks, no solo el ADK.
- **Gestión del ciclo de vida del agente:** Ofrece APIs para crear, leer, actualizar y eliminar (CRUD) tus agentes desplegados.

Funcionalidades agénticas especializadas

- Memory Bank: Un servicio gestionado para generar y recuperar dinámicamente registros de memorias a largo plazo personalizados y basados en las conversaciones de los usuarios.
- Example Store: Permite a los desarrolladores proporcionar y administrar ejemplos few-shot para mejorar y guiar el rendimiento del agente en tareas específicas.

Arquitectura del sistema para un motor de agentes impulsado por Gemini



Colaboración con la comunicación Agent2Agent

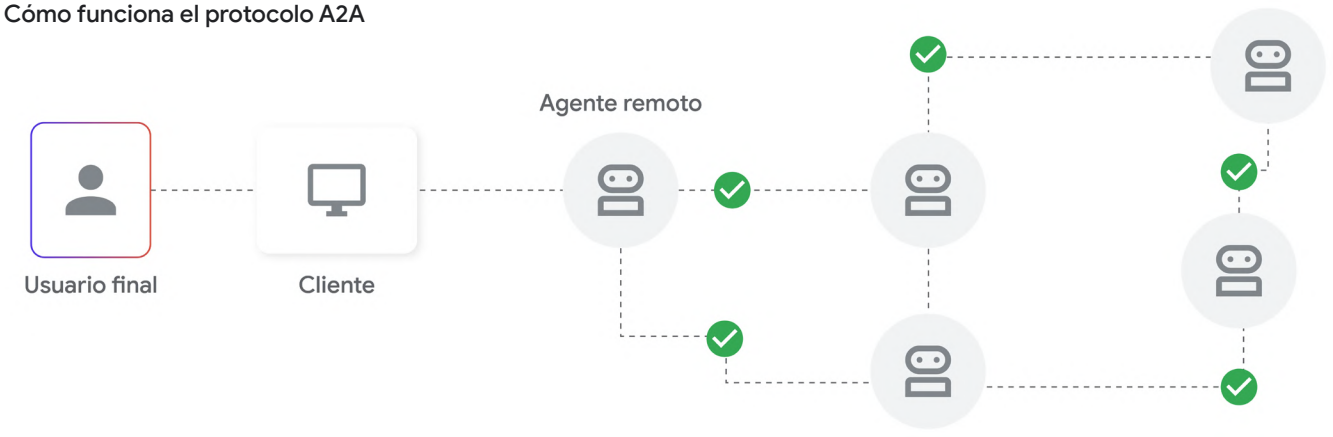
El verdadero potencial de los agentes especializados se libera cuando pueden colaborar entre sí. Para facilitar esto, Google promueve el **protocolo Agent2Agent (A2A)**, un estándar abierto que garantiza que los agentes que desarrolles hoy puedan descubrir, comunicarse y coordinar acciones de forma segura con otros agentes, independientemente de quién los haya creado o del framework que utilicen. Este compromiso con un ecosistema abierto e interoperable es fundamental para la estrategia de agentes de Google Cloud.

Los conceptos clave del protocolo A2A incluyen:

- **Tarjeta del agente:** Una "tarjeta de presentación" digital (típicamente un archivo JSON en un endpoint conocido) que un agente utiliza para comunicar sus capacidades, la URL de su endpoint y los requisitos de autenticación, lo que facilita que otros agentes lo descubran.
- **Arquitectura orientada a tareas:** Las interacciones se estructuran como "tareas". Un agente cliente envía una solicitud de tarea a un agente servidor, el cual la procesa y devuelve una respuesta. Un agente puede actuar tanto como cliente como servidor.
- **Independiente de la modalidad:** El protocolo A2A es compatible con la comunicación mediante texto, audio y video, reflejando así la naturaleza multimodal y en constante evolución de las interacciones entre agentes.

Ecosistema de socios robustos del protocolo A2A

Cómo funciona el protocolo A2A



Los agentes del ADK pueden integrarse de forma nativa en este ecosistema. Exponen un endpoint HTTP estándar y un archivo agent.json, lo que facilita su descubrimiento y comunicación con cualquier otro agente compatible con A2A.

Consejo experto

Explora estos recursos de A2A para comenzar:

- Organización en Github del Proyecto A2A
- Documentación del protocolo A2A
- Especificación del protocolo A2A

Historia del cliente

Cómo BioCorteX utiliza el protocolo A2A para acelerar el descubrimiento de fármacos.

BioCorteX utiliza grafos de conocimiento y simulaciones in silico para modelar las complejas interacciones entre las bacterias, los fármacos y el huésped. De este modo, logran descubrir interacciones ocultas y mitigar el riesgo en el proceso de desarrollo de medicamentos para sus socios farmacéuticos.

Situación

En las ciencias de la vida, conectar y transformar conjuntos de datos dispares en conocimiento de relevancia comercial es un proceso lento e incierto. La comprobación de hipótesis puede llevar años, obstaculizada por modelos deficientes, teorías contradictorias y la enorme complejidad de la biología.

Solución

BioCorteX desarrolló un sistema multiagente en Google Cloud que evalúa hipótesis desde tres dimensiones: plausibilidad biológica, relevancia clínica en el mundo real e importancia comercial. Utiliza agentes impulsados por Gemini, el ADK y una arquitectura GraphRAG para navegar por su grafo de conocimiento de 44 mil millones de conexiones de muestras globales, todo orquestado a través de A2A.

Impacto

Lo que antes llevaba años, ahora toma días. Los agentes de grafos de BioCorteX ofrecen a los tomadores de decisiones una planificación de escenarios totalmente transparente en todo su portafolio. Esto permite respaldar las consideraciones comerciales de alto nivel con un profundo conocimiento científico, lo que acelera las pruebas de nuevos mecanismos y áreas terapéuticas al tiempo que reduce el desperdicio en todo el pipeline de desarrollo.

> “En BioCorteX, nuestros agentes Carbon Graph son distintos. A diferencia de otros agentes, estos se basan en hechos, no en opiniones ni en meras asociaciones. Al recorrer el grafo de conocimiento basado en biología mecanicista más grande del mundo [en lugar de utilizar un LLM], los agentes Carbon Graph no se limitan a sugerir hipótesis, sino que evalúan su plausibilidad, relevancia clínica y viabilidad comercial. Esto permite una alineación total entre los equipos de I+D, Asuntos Regulatorios y el área Comercial de la organización.”

2.2 Guía paso a paso: cómo definir un agente LLM

El desarrollo de un agente de IA es un proceso iterativo de definición, pruebas y despliegue. Esta sección se enfoca en la primera fase y la más crítica: definir la identidad central, las instrucciones y las capacidades del agente.

Si bien el repositorio de código abierto ADK Samples ofrece una biblioteca completa de agentes listos para usar, su propósito es mostrarte la estructura de un agente terminado. Esta sección, por el contrario, está diseñada para enseñarte a pensar en el desarrollo de agentes. Explica los principios arquitectónicos y el "porqué" estratégico detrás de cada componente central. De este modo, adquirirás el conocimiento fundamental necesario para utilizar los ejemplos de código con eficacia y desarrollar tus propias soluciones personalizadas.

Para llevar esto a la práctica, analicemos el proceso de desarrollo de un Asistente de Errores de Software. Se trata de un agente LlmAgent diseñado para ayudar a un equipo de soporte a clasificar y priorizar los nuevos informes de incidencias.

1. Define la identidad del agente

Primero, debes establecer qué es el agente y para qué sirve. Esto se hace con tres parámetros clave:

- **name (obligatorio)**: Un identificador de cadena único, crucial para las operaciones internas y la delegación multiagente. Para nuestro ejemplo: software_bug_triage_agent.
- **description (recomendado)**: Un resumen conciso de sus capacidades, utilizado por otros agentes para decidir cuándo enrutar tareas. Para nuestro ejemplo: "Analiza nuevos reportes de errores de software, categoriza su prioridad y los asigna al equipo de ingeniería correcto".
- **model (obligatorio)**: El LLM subyacente que impulsa el razonamiento del agente, como gemini-2.5-flash.

2. Guía al agente con instrucciones

El parámetro instruction es el componente más crítico para definir el comportamiento de un agente. Este parámetro establece la tarea principal del agente, su rol (persona), sus limitaciones y la manera en que debe utilizar sus herramientas. En el caso de nuestro Asistente de Errores de Software, le indicaríamos que actúe como un gerente de ingeniería experto, le explicaríamos cómo buscar datos de los usuarios mediante el uso de sus herramientas y especificaríamos que su salida final debe ser un objeto JSON para nuestro sistema de tickets.

Una instrucción efectiva debe:

- **Ser clara y específica** sobre los resultados deseados.
- **Brindar ejemplos** (few-shot prompting) para tareas complejas.
- **Guiar el uso de herramientas** explicando cuándo y por qué deben usarse.
- **Inyectar datos dinámicos** del estado del agente usando la sintaxis{variable}.

Consejo experto

Sé sumamente preciso. Ten en cuenta que toda la definición actúa como un prompt.

Un LLM utiliza cada elemento de la definición de un agente para su razonamiento, desde su nombre y descripción hasta los nombres y descripciones de sus herramientas. Además, el modelo interpreta esta información de forma sumamente literal. Evita utilizar nombres y descripciones ambiguos, poco claros o contradictorios, ya que esto puede provocar un "envenenamiento del contexto" y hacer que el agente se confunda, persiga objetivos incorrectos o falle al utilizar sus herramientas. Por el contrario, considera cada cadena de texto de la configuración como una instrucción minuciosamente elaborada para el modelo.

Nota

Los agentes de IA y sus frameworks subyacentes están evolucionando a un ritmo vertiginoso. Si bien esta guía se centra en los principios arquitectónicos y patrones perdurable para el desarrollo de sistemas de agentes, los fragmentos de código y los detalles específicos de las API que se muestran a continuación representan una "instantánea temporal". Nuestro objetivo es transmitir el "porqué" y el "cómo" del diseño de agentes, y no proporcionar código fuente para copiar y pegar directamente en un entorno de producción.

Para obtener los detalles de implementación más actuales, las firmas de API y las mejores prácticas, consulta siempre la documentación oficial del ADK y repositorio de ADK Samples.

3. Dota al agente de herramientas

Las herramientas dan a tu agente de capacidades que van más allá de su razonamiento integrado, lo que le permite interactuar con el mundo exterior. Para cumplir su función, nuestro Asistente de Errores de Software necesitaría varias herramientas, tales como:

- Una función para obtener información sobre el usuario que reporta el error (get_user_details(user_id)).
- Una función para buscar en el código base archivos relevantes (search_codebase(file_name)).
- Una función para crear un ticket en un sistema de gestión de proyectos (create_jira_ticket(..)).

El LLM utiliza el nombre, el docstring y el esquema de parámetros de la herramienta para determinar a cuál debe invocar.

Consejo experto

Sé breve y directo.

Cada herramienta que defines agrega una nueva opción para que el modelo la considere. Especialmente cuando un agente tiene muchas herramientas, cualquier ambigüedad o superposición en sus descripciones puede confundir al modelo, lo que lleva a comportamientos de bucle o a una selección incorrecta de la herramienta. Para asegurar que el modelo pueda elegir correctamente, haz que el nombre y la descripción de cada herramienta sean una señal clara y única de su propósito.

4. Completa el ciclo de vida de desarrollo

Ahora estás listo para probar y evaluar el rendimiento de tu agente. El objetivo de esta fase es analizar la calidad de las respuestas del agente examinando su ejecución paso a paso (su "trayectoria"). La siguiente sección cubre en detalle el aspecto fundamental de la evaluación de agentes.

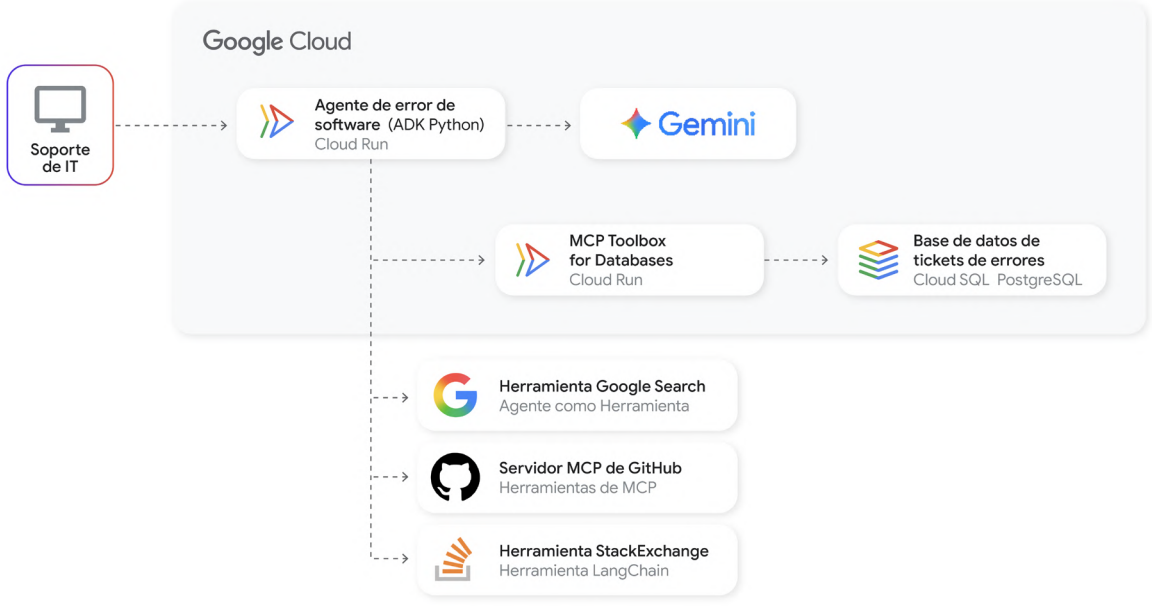
Una vez que hayas verificado que su funcionamiento es óptimo, necesitas un flujo simplificado para su despliegue. En esta etapa, tu prototipo se convierte en una aplicación lista para producción para tu equipo o clientes, transformando al agente en una herramienta empresarial fiable.

Consejo experto

Prueba, prueba y vuelve a probar

Los sistemas de agentes son no deterministas y pueden exhibir comportamientos emergentes, por lo que las pruebas unitarias estándar resultan insuficientes. La evaluación rigurosa es la única forma de garantizar la calidad y la fiabilidad de tu agente. Centra tus pruebas en dos áreas clave: la trayectoria de razonamiento (la lógica paso a paso y el uso de herramientas) y la calidad de la salida final (su precisión, utilidad y fundamentación). Como demuestran las exhaustivas pruebas de rendimiento, incluso los modelos de última generación pueden producir alucinaciones o quedarse atrapados en bucles de razonamiento, lo que hace que la evaluación continua sea una parte crítica del ciclo de vida de desarrollo.

Arquitectura de un asistente de errores de software con ADK Python



Historia de cliente

Cómo Box utiliza el ADK y el protocolo A2A para acelerar el desarrollo de contenido.

Box es una plataforma de Gestión de Contenido Inteligente que permite a las organizaciones impulsar la colaboración, gestionar todo el ciclo de vida del contenido, proteger el contenido crítico y transformar los flujos de trabajo de negocio.

Situación

Los procesos de negocio críticos como las verificaciones de cumplimiento, la gestión de contratos y las aprobaciones de préstamos se ralentizan porque los empleados tienen que buscar e interpretar grandes cantidades de información almacenada en documentos de Box. Esto crea ineficiencias y retrasa decisiones críticas.

Solución

Box presenta un agente compatible con A2A, desarrollado con el ADK de Google e impulsado por Gemini. El agente se integra directamente con Box Intelligent Content Cloud, lo que permite a los usuarios formular preguntas complejas en lenguaje natural y obtener al instante respuestas contextualizadas, resúmenes y estadísticas de sus documentos.

Impacto

Esto acelera drásticamente los flujos de trabajo centrados en el contenido y mejora la calidad de la toma de decisiones. Además, sienta las bases para agentes transaccionales más avanzados capaces de gobernar, gestionar e iniciar procesos como firmas electrónicas y aprobaciones, lo que transforma fundamentalmente la forma de trabajar en el entorno empresarial.

> “Estamos entrando en una nueva era donde los agentes de IA transformarán la forma de trabajar, y el contenido se encuentran en el centro de todo. Gracias a Box, que actúa como capa de contenido segura, y al protocolo A2A de Google Cloud, que facilita una colaboración fluida en todo el ecosistema, estamos desbloqueando nuevas y poderosas formas de automatizar los procesos empresariales, acelerar la toma de decisiones y generar resultados tangibles para nuestros clientes”.

box

2.3 Gobierna y escala tu fuerza de trabajo de agentes con Gemini Enterprise

A medida que tu startup pasa de crear un único agente a desplegar un portafolio de agentes especializados, te enfrentas a un nuevo conjunto de desafíos: ¿Cómo gestionarlos? ¿Cómo facilitar su adopción por parte de los perfiles no técnicos del equipo? ¿Cómo gobernar su acceso a los datos y herramientas?

Gemini Enterprise resuelve estos problemas de escalabilidad. Esta plataforma unificada y segura permite crear, gobernar y orquestar toda tu fuerza de trabajo de agentes de IA, integrando aplicaciones y fuentes de datos dispares. Esta solución complementa el desarrollo orientado al código del ADK, ya que proporciona el framework necesario para escalar el uso de agentes en toda tu organización y gestionarlos de manera eficaz.

Puedes usar Gemini Enterprise para

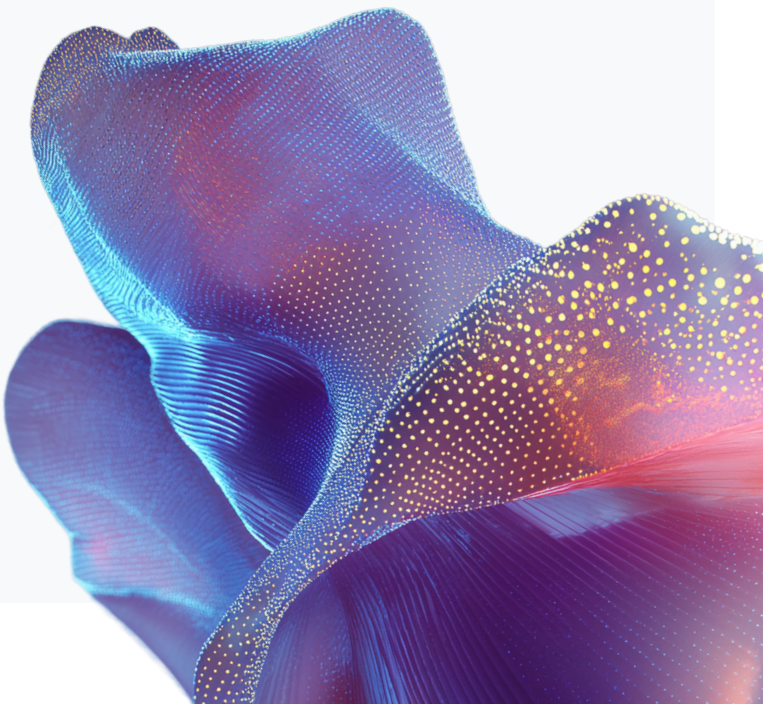
- **Unificar y acceder a los datos de la empresa:** Gemini Enterprise elimina los silos de datos utilizando conectores listos para usar en tus aplicaciones empresariales existentes (como Microsoft SharePoint, Google Workspace, Jira). Aplica la tecnología de búsqueda multimodal de Google a través de estos datos conectados, lo cual permite a cualquier empleado obtener respuestas instantáneas y sintetizar información de una fuente de verdad central, respetando todos los controles de acceso existentes.
- **Impulsar la automatización en todo el equipo**: Si bien el ADK es ideal para el desarrollo complejo de agentes orientados al código, Gemini Enterprise permite que toda tu organización automatice sus flujos de trabajo. Los expertos de dominio en producto, marketing y operaciones pueden utilizar el Agent Designer, una herramienta sin código, para desarrollar sus propios agentes personalizados mediante una interfaz basada en prompts. Esto convierte su conocimiento específico en soluciones automatizadas sin depender de recursos de ingeniería.
- **Gobernar y orquestar una flota de agentes**: Gemini Enterprise proporciona una plataforma unificada para gestionar y gobernar los agentes creados usando el ADK, el Agent Designer sin código o por socios externos. La Agent Gallery funciona como un portal centralizado para que tu equipo descubra, gestione y despliegue todos estos agentes. Esto incluye tanto tus soluciones personalizadas como los Agentes de Google listos para usar, diseñados para tareas complejas como la investigación profunda o la generación de ideas.

Prueba estos prompts en Gemini Enterprise:

> Programa nuestra reunión semanal de sincronización del equipo para el jueves a las 10 a.m.

> Resume las novedades de esta semana del canal de Slack #product.

> Crea una agenda de reunión para discutir la preparación para inversores.



Historia de cliente

Cómo el agente de IA de Zoom programa automáticamente reuniones de Zoom desde el contexto de Gmail, con integración a Gemini Enterprise

Zoom es una empresa de tecnología de comunicación que ofrece una plataforma de trabajo abierta basada en IA (AI-first) que se utiliza para reuniones virtuales, webinars, chat, colaboración en línea, experiencia del cliente y mucho más.

Situación

La estrategia AI-first de Zoom se centra en transformar AI Companion en un framework totalmente agéntico. Este no solo es capaz de realizar un razonamiento avanzado y una orquestación de tareas, sino que también se integra a la perfección con los sistemas de terceros clave para el cliente. AI permitir la colaboración con otros agentes de IA, Zoom impulsa resultados laborales más significativos a través de un ecosistema abierto e interoperable.

Solución

AI Companion de Zoom se integrará con Gemini Enterprise para agilizar la programación de reuniones. Con su lanzamiento previsto para finales de este verano, esta colaboración permitirá a los agentes de IA compatibles con el protocolo A2A programar automáticamente reuniones de Zoom directamente desde el contexto de Gmail, actualizar Google Calendar y mantener informados a los participantes. De este modo, se elimina el ida y vuelta propio de la programación manual.

Impacto

- Reducción de barreras técnicas en la integración de IA entre plataformas.
- Interacción fluida entre AI Companion de Zoom y agentes externos compatibles con el protocolo A2A sin código personalizado.
- Mejora en la automatización de flujos de trabajo y mayor eficiencia para clientes empresariales.
- Soporte futuro para interacciones de IA multiagente más sofisticadas.

> “Nuestra contribución al protocolo A2A permite una integración más profunda con Google Cloud y otras plataformas de terceros, lo que ofrece a los clientes mayor flexibilidad y capacidad de elección”.

zoom

2.4 Otras opciones para crear agentes

Experimenta la CLI de Gemini

Para las startups que necesitan una forma inmediata y rentable de experimentar con IA, la CLI de Gemini es un agente de código abierto que lleva a Gemini directamente a tu terminal. Esta herramienta ofrece:

- **Ahorro significativo:** Obtén acceso gratuito a Gemini 2.5 con límites de uso generosos (ventana de contexto de 1 millón de tokens, 60 consultas por minuto).
- **Productividad mejorada**: Al integrarse en los flujos de trabajo existentes de los desarrolladores, acelera la codificación, la depuración y la documentación.
- **Flexibilidad total**: Al ser una herramienta de código abierto (Apache 2.0), puedes auditarla, modificarla e integrarla en tu cadena de herramientas, sin quedar atado a un proveedor y con la opción de realizar una personalización profunda.

Consejo experto

Echa un vistazo a la CLI de Gemini configurada para el desarrollo con ADK.

Acelera el desarrollo con Firebase Studio

Un agente de backend, incluso uno poderoso construido con el ADK, es solo una parte de un producto completo. Para darle vida, necesitas desarrollar una aplicación full stack a su alrededor, como la interfaz de usuario, bases de datos y hosting. Firebase Studio es un espacio de trabajo integrado en la nube que utiliza IA agéntica para acelerar todo el ciclo de vida de desarrollo. Los equipos pueden usarlo para manejar todo, desde el prototipado de UI y la generación de código hasta el despliegue seguro en la infraestructura de Google Cloud.

Juntos, el ADK para la lógica de backend del agente, el Agent Starter Pack para la infraestructura de producción (que analizaremos en la sección 3) y Firebase Studio para la aplicación completa ofrecen un kit de herramientas completo de extremo a extremo para que una startup desarrolle y despliegue un sistema de agentes potente y de vanguardia.

Firebase Studio acelera todo el ciclo de vida de desarrollo con IA:

- **Configuración rápida:** Usa el App Prototyping Agent para crear un nuevo proyecto usando lenguaje natural, maquetas o capturas de pantalla. Comienza seleccionando opciones desde un amplio catálogo de plantillas para frameworks y lenguajes populares, o importa un proyecto existente.
- **Gemini en Firebase**: Aprovecha la asistencia de IA en tareas como codificación, depuración, pruebas, refactorización, explicación y documentación de código.
- **Colaboración**: Comparte espacios de trabajo con los miembros del equipo y proporciona una URL a los primeros testers para que obtengan una vista previa de las aplicaciones.
- **Optimización**: Previsualiza las aplicaciones tal como las verán los usuarios con vistas previas web integradas y emuladores de Android. Además, prueba y optimiza tu código con acceso a miles de extensiones en el registro Open VSX.
- **Despliegue**: Publica en Firebase App Hosting con unos pocos clics o despliega aplicaciones listas para producción en Cloud Run, Firebase Hosting o en tu propia infraestructura personalizada.

Prueba estos prompts con el agente App Prototyping:

> Genera un panel de atención al cliente que ingiera datos de Zendesk y muestre métricas clave como el volumen de tickets y el tiempo de resolución.

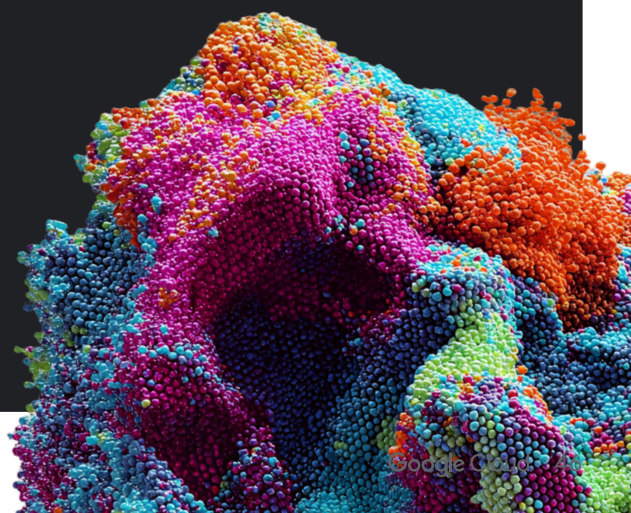
> Crea una aplicación SaaS B2B con autenticación de usuarios, una base de datos PostgreSQL y una página de facturación por suscripción.

> Construye una aplicación full stack para un sistema interno de seguimiento de errores con un formulario para su envío y un tablero Kanban para ver el estado de los tickets.

También puedes empezar seleccionando opciones desde un amplio catálogo de plantillas para frameworks y lenguajes populares, o importar un proyecto existente.

Puntos clave: desde la construcción hasta el escalado

Tu objetivo	Mejor opción
Construir un sistema multiagente personalizado desde código.	Usa el Agent Development Kit (ADK) de código abierto.
Desplegar, escalar y gestionar tu agente en producción.	Despliégalo en Vertex AI Agent Engine.
Darle a tu agente el poder de la memoria a largo plazo.	Usa la funcionalidad Memory Bank dentro de Vertex AI Agent Engine.
Permitir que tu agente descubra y hable con otros agentes.	Permite que tu agente descubra y hable con otros agentes.
Construir una aplicación completa impulsada por IA desde un prompt.	Usa Firebase Studio para un espacio de trabajo de desarrollo asistido por IA.
Experimentar rápidamente con Gemini en tu terminal.	Usa la CLI de Gemini para una interfaz de línea de comandos simple.



¿Todo listo para convertir tu visión de IA en realidad? Estamos aquí para ayudarte.

Aprende a desarrollar más aplicaciones de IA generativa con las sesiones bajo demanda de Startup School.

Empieza ahora

Recibe hasta $350,000 USD en créditos de Google Cloud con el programa Google Cloud for Startups.

Solicita la inscripción ahora

Habla con nuestro equipo especializado en startups.

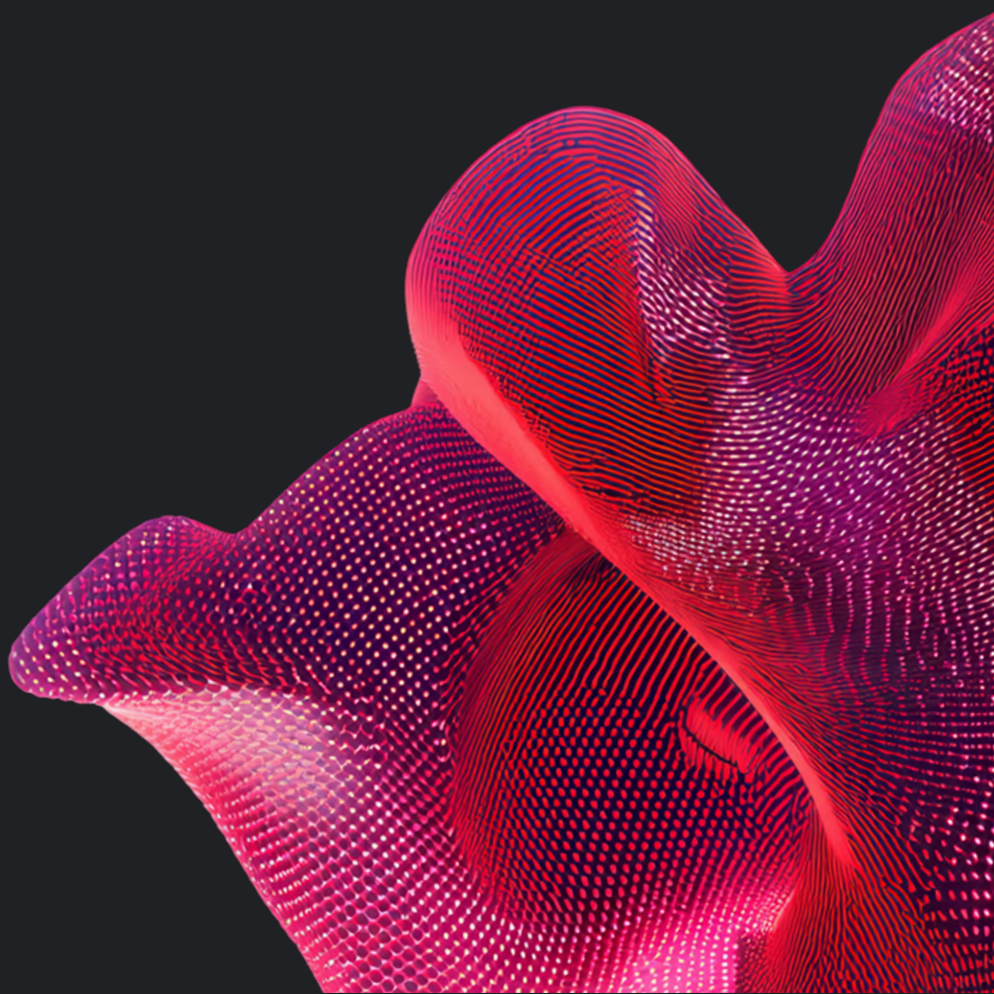
Comunícate con nosotros

Mantente al tanto de las últimas novedades y recibe todas nuestras actualizaciones suscribiéndote al boletín informativo de Google Cloud Startup.

Suscribirse

Sección 3

Cómo garantizar agentes de IA fiables y responsables



Debido a la naturaleza no determinista de los sistemas basados en LLM, a menudo resulta difícil lograr una fiabilidad de grado de producción. Ir más allá de una prueba superficial informal (vibe-testing) requiere un enfoque de ingeniería riguroso para garantizar que un agente opere de forma segura y genere valor de manera consistente.

Esta sección detalla las metodologías y herramientas necesarias para abordar estos desafíos, centrándose en tres áreas clave:

- **Corrección y fiabilidad:** Evaluar la precisión de las salidas finales y la validez de los pasos de razonamiento intermedios.
- **Rendimiento y escalabilidad:** Medir y optimizar la latencia y el rendimiento del agente bajo carga.
- **Seguridad y responsabilidad:** Implementar salvaguardas, monitorear comportamientos indeseables y garantizar que el agente opere dentro de los límites definidos.

Estas prácticas materializan el compromiso de Google con una IA responsable, lo que permite a las startups desarrollar agentes robustos y fiables, estrictamente alineados con los principios de seguridad líderes en la industria.

¿Prefieres audio? Escucha la versión en podcast de esta sección, creada con NotebookLM.

> **Los agentes poseen la clave para alcanzar un nuevo nivel de productividad, pero su éxito depende, en última instancia, de nuestra guía".**
>
> **Harrison Chase**
> CEO y cofundador de LangChain

Consejo experto

Alcanzar una observabilidad de grado de producción implica ir más allá de las métricas a nivel de aplicación; requiere la medición de métricas operativas de bajo nivel, como la utilización de CPU y memoria. El monitoreo riguroso del consumo de recursos es fundamental para diagnosticar cuellos de botella en el rendimiento, optimizar el tiempo de ejecución y reducir de forma directa los costos operativos. El ADK y el Agent Starter Pack proporcionan soporte nativo para OpenTelemetry, lo que te permite exportar estos datos operativos críticos directamente hacia tus plataformas de monitoreo existentes.

Este podcast fue creado usando NotebookLM con el siguiente prompt: "Como presentador de podcast, genera un podcast dirigido a desarrolladores y fundadores técnicos. Presenta AgentOps como el framework para desarrollar una IA fiable, dejando a un lado las pruebas informales para adoptar un proceso riguroso y automatizado.

"Explica cómo AgentOps evalúa el razonamiento, la precisión y la seguridad de un agente, y cómo mitiga riesgos como la desinformación y las vulnerabilidades de seguridad. Describe cómo el Agent Starter Pack, al integrarse con el Agent Development Kit (ADK), implementa esto rápidamente mediante herramientas preconfiguradas para la infraestructura, CI/CD (Integración y Despliegue Continuos) y la evaluación continua. Concluye destacando que este enfoque disciplinado representa una ventaja competitiva e invita a los oyentes a explorar los recursos de Google para startups".

3.1 AgentOps: un framework para agentes listos para producción

Operaciones de Agentes (AgentOps) es una metodología operativa que aborda los desafíos de fiabilidad y responsabilidad en producción. Adapta los principios de DevOps, MLOps y DataOps a los desafíos únicos de desarrollar, desplegar y gestionar agentes de IA a lo largo de su ciclo de vida. Además, brinda un framework sistemático, automatizado y reproducible para gestionar las complejidades de los sistemas no deterministas basados en LLM en entornos de producción.

Una estrategia robusta de AgentOps sistematiza el proceso de desarrollo, proporcionando bucles de retroalimentación continuos para mejorar la fiabilidad, la seguridad y el rendimiento de un agente en todas sus herramientas, sus capacidades de razonamiento y sus modelos subyacentes.

Un framework sistemático para la evaluación de agentes

La evaluación de sistemas de agentes no deterministas representa uno de los desafíos más complejos en la ingeniería de software moderna. Mientras que las pruebas tradicionales suelen centrarse en la corrección léxica, la evaluación de agentes debe abordar dos dimensiones mucho más críticas: la corrección semántica (¿el agente comprendió la intención del usuario y generó una respuesta verdaderamente útil?) y la corrección de razonamiento (¿el agente siguió una trayectoria lógica y eficiente para llegar a su conclusión?).

Como ya explicamos en la sección 1, la arquitectura cognitiva que gobierna este razonamiento suele basarse en un framework como ReAct, el cual establece un bucle dinámico donde el agente intercala pensamiento y acción. Un fallo en cualquier punto de este bucle puede derivar en un resultado incorrecto. Por consiguiente, se requiere un framework de evaluación riguroso y de múltiples capas. Esta arquitectura se implementa mediante la combinación del ADK —para la lógica central y la instrumentación del agente— y el Agent Starter Pack para la infraestructura de grado de producción, la automatización y observabilidad.

Capa 1: Evaluación a nivel de componente (pruebas unitarias deterministas)

Esta capa se centra en los componentes predecibles y no basados en LLM del sistema del agente.

- **Objetivo:** Verificar la corrección léxica de los bloques de construcción individuales y garantizar que los fallos del agente no provengan de bugs simples en sus componentes.
- **Qué evaluar:**
  - **Herramientas:** Comportamiento esperado con entradas válidas, inválidas y de casos límite.
  - **Procesamiento de datos:** Robustez de las funciones de análisis y serialización.
  - **Integraciones de API:** Manejo de condiciones de éxito, error y tiempo de espera.
- **Implementación:**
  - El ADK define las herramientas del agente como funciones de Python (o métodos de Java). Estas funciones son los sujetos directos de las pruebas a nivel de componente.
  - El Agent Starter Pack proporciona la infraestructura de pruebas. Genera un proyecto con un entorno pytest estándar configurado en el directorio tests/unit/. Los desarrolladores pueden escribir inmediatamente pruebas unitarias para sus herramientas definidas en el ADK y ejecutarlas a través del comando make test.

Capa 2: Evaluación de trayectoria (corrección procedimental)

Esta es la capa más crítica para evaluar el proceso de razonamiento del agente. Una “trayectoria” es la secuencia completa de pasos de Razonamiento, Acción y Observación que toma el agente para completar una tarea.

- **Objetivo:** Verificar la corrección de razonamiento del agente dentro del ciclo ReAct.
- **Qué evaluar:**
  - **Paso de Razonamiento:** ¿El agente evalúa

correctamente el objetivo y el estado actual para formar una hipótesis lógica para el siguiente paso?

- **Paso de Acción:** ¿Selecciona la herramienta correcta (Selección de Herramienta) y extrae y formatea correctamente los argumentos para esa herramienta (Generación de Parámetros)?
- **Paso de Observación:** ¿Integra correctamente los datos de salida de la herramienta para informar el siguiente paso de Razonamiento del ciclo?

- **Implementación:**
  - El entorno de ejecución central del ADK ejecuta el bucle ReAct del agente y se integra directamente con Google Cloud Trace para instrumentar cada paso de Razonamiento, Acción y Observación. Esta integración permite a los desarrolladores visualizar toda la trayectoria, inspeccionar las entradas y salidas de las herramientas y examinar la cadena de pensamiento del modelo para depurar su razonamiento.
  - El Agent Starter Pack automatiza y escala la evaluación de la trayectoria. Un directorio tests/integration/ crea un conjunto de referencia de prompts con las trayectorias ReAct esperadas. El pipeline automatizado de CI/CD (configurado con agent-starter-pack setup-cicd) ejecuta estas pruebas en cada pull request para prevenir regresiones. Además, la infraestructura de observabilidad del Starter Pack es la que captura los datos de la traza emitidos por el agente del ADK.

Capa 3: Evaluación de resultados (corrección semántica)

Esta capa evalúa la respuesta final dirigida al usuario, la cual se genera una vez que el bucle ReAct ha concluido.

- **Objetivo:** Verificar la corrección semántica, la precisión factual y la calidad general de la respuesta final.

- **Qué evaluar:**
  - **Precisión factual y fundamentación:** ¿La respuesta es correcta y puede verificarse que se basa en la información recopilada durante los pasos de Observación?
  - **Utilidad y tono:** ¿La respuesta satisface completamente la necesidad del usuario usando el estilo apropiado?
  - **Completitud:** ¿La respuesta contiene toda la información necesaria?
- **Implementación:**
  - El conjunto de herramientas del ADK es clave para verificar la precisión factual. Los desarrolladores pueden crear herramientas especializadas o usar los API para la verificación de fundamentación. Estas herramientas, invocadas durante el paso de Acción, verifican programáticamente si la respuesta final del agente está respaldada por el contexto que recuperó, lo que proporciona una métrica cuantitativa contra la alucinación.
  - El Agent Starter Pack proporciona la plataforma para ejecutar estas evaluaciones a escala. Se integra con el servicio de evaluación de IA Generativa de Vertex AI para realizar la puntuación mediante la técnica de LLM como juez. Su entorno de pruebas de UI integrado incluye mecanismos de feedback que registran las calificaciones humanas directamente en BigQuery, lo que permite una evaluación de alta fidelidad con intervención humana.

Capa 4: Monitoreo a nivel de sistema (en producción)

La evaluación no termina con el despliegue. El monitoreo continuo del rendimiento del agente en tiempo real es fundamental.

- **Objetivo:** Supervisar el rendimiento en el mundo real y detectar fallos operativos o la deriva de comportamiento.
- **Qué monitorear:** Las tasas de fallo de las herramientas, las puntuaciones de feedback de los usuarios, las métricas de trayectoria (como el número de ciclos ReAct por tarea) y la latencia de extremo a extremo.

  Implementación:

- El agente del ADK, que se ejecuta en producción, es la fuente de los datos operativos, y emite eventos y trazas para cada interacción de usuario en tiempo real.
- El Agent Starter Pack proporciona un stack de observabilidad de grado de producción listo para usar. Configura automáticamente OpenTelemetry y un enrutador de registros (Log Router) hacia BigQuery, además de proporcionar plantillas para paneles de Looker Studio. Esto permite a los equipos supervisar inmediatamente el rendimiento del agente, analizar tendencias y depurar problemas utilizando datos del uso en el mundo real, sin requerir configuración adicional.

Esta metodología integral y práctica para la evaluación de agentes representa la implementación tangible de una estrategia robusta de AgentOps, que impulsa a los equipos más allá de una prueba superficial informal, guiándolos hacia un proceso sistemático, automatizado y reproducible. Al diseccionar la evaluación en componentes clave (trayectoria, resultados y monitoreo a nivel de sistema), se abordan directamente los dominios centrales de AgentOps.

La adopción de un framework de evaluación sistemático no es simplemente una buena práctica: es una ventaja competitiva. Establece un proceso riguroso, automatizado y basado en datos que permite a los equipos innovar más rápido, desplegar con total confianza y desarrollar agentes cuyo nivel de seguridad y eficacia es demostrable.

Kit de herramientas AgentOps: ADK y Agent Starter Pack

Los pipelines automatizados de CI/CD implementan los principios de AgentOps, de modo que cualquier cambio en el código, las herramientas o los prompts del agente activa un proceso estandarizado de compilación, pruebas unitarias y evaluación cuantitativa contra un conjunto de datos predefinido. Esta etapa de evaluación automatizada es esencial para prevenir regresiones y brindar un feedback continuo y objetivo sobre el rendimiento y la seguridad del agente antes del despliegue.

Para acelerar la adopción de los principios de AgentOps, el Agent Starter Pack ofrece una implementación de referencia lista para producción. Sus plantillas holísticas resuelven los desafíos comunes (como despliegue y operaciones, evaluación, personalización y observabilidad) asociados al desarrollo y el despliegue de agentes de IA. En resumen, esta herramienta inicializa un nuevo proyecto de agente con la infraestructura y los pipelines necesarios, lo que permite a los desarrolladores centrarse exclusivamente en la lógica central.

Consejo experto

Puedes crear un nuevo proyecto de agente listo para producción con un solo comando: uvx agent-starter-pack create my-agent -a adk@gemini-fullstack.

El Agent Starter Pack incluye los siguientes componentes clave:

- **Infraestructura como Código (Terraform):** Proporciona plantillas reproducibles para aprovisionar y gestionar el entorno cloud del agente, lo que incluye servicios como Cloud Run, permisos de IAM y redes.
- **Pipelines de CI/CD (Cloud Build):** Un archivo cloudbuild.yaml preconfigurado automatiza la compilación, las pruebas unitarias, la evaluación cuantitativa y el despliegue, e implementa directamente el flujo de trabajo de CI/CD de AgentOps.
- **Observabilidad y registro (Cloud Trace y Cloud Logging):** Establece la base para el monitoreo y la depuración mediante la integración con Cloud Trace para un análisis profundo de las trazas de ejecución del agente, y con Cloud Logging para una gestión centralizada de los registros.
- **Integración de datos (BigQuery):** Incluye componentes fundamentales para los agentes que necesitan conectarse y analizar datos empresariales estructurados utilizando BigQuery.
- **Evaluación continua (evaluación de Vertex AI):** Se integra con Vertex AI para ejecutar conjuntos de datos de evaluación contra los cambios del agente, lo que permite medir continuamente el rendimiento frente a los dominios clave discutidos anteriormente.

Arquitectura de alto nivel del Agent Starter Pack

Mejores juntos: ADK y Agent Starter Pack

El ADK y el Agent Starter Pack están diseñados para proporcionar una clara separación entre la lógica de aplicación de un agente y su ciclo de vida operativo, lo que permite un proceso de desarrollo robusto y escalable.

En esencia, el ADK se utiliza para escribir el código de aplicación del agente, mientras que el Agent Starter Pack proporciona la línea base operativa, lista para producción, que permite ejecutar y gestionar ese código a escala.

- **El ADK gestiona el comportamiento de ejecución del agente:** Al funcionar como un SDK de Python/Java, el ADK proporciona las APIs y las abstracciones centrales para definir la lógica de aplicación de un agente. Los desarrolladores lo utilizan para implementar flujos de orquestación, definir herramientas y configurar interacciones con los LLM.
- **El Agent Starter Pack gestiona el entorno operativo:** Al funcionar como una herramienta de andamiaje, genera la infraestructura como código (Terraform) para aprovisionar los entornos de despliegue (p. ej., Cloud Run) y las configuraciones del pipeline de CI/CD (Cloud Build) con el fin de automatizar todo el ciclo de vida.

Esta separación se manifiesta en un flujo de trabajo de cinco pasos:

1. **Inicialización con el Agent Starter Pack:** Un desarrollador ejecuta un solo comando para generar un nuevo proyecto que contiene todos los componentes operativos necesarios, incluidos los archivos de Terraform para la infraestructura, las configuraciones de Cloud Build para CI/CD y los archivos base para los conjuntos de datos de evaluación.
2. **Desarrollo con el ADK**: Dentro de esta estructura, el desarrollador utiliza el ADK para escribir la lógica de aplicación del agente, implementando herramientas personalizadas, componiendo agentes y redactando las instrucciones centrales.
3. **Commit y automatización**: Cuando el código se confirma y se sube al repositorio fuente, se activa automáticamente el pipeline de CI/CD, el cual está preconfigurado y gestionado por Cloud Build.
4. **Evaluación continua**: El pipeline construye el agente del ADK en un contenedor y luego ejecuta una evaluación cuantitativa contra un conjunto predefinido de pruebas. De este modo, se valida programáticamente el rendimiento y la seguridad del agente.
5. **Despliegue con total confianza**: Tras una evaluación exitosa, el pipeline despliega automáticamente la nueva versión validada del agente en su entorno de producción final.

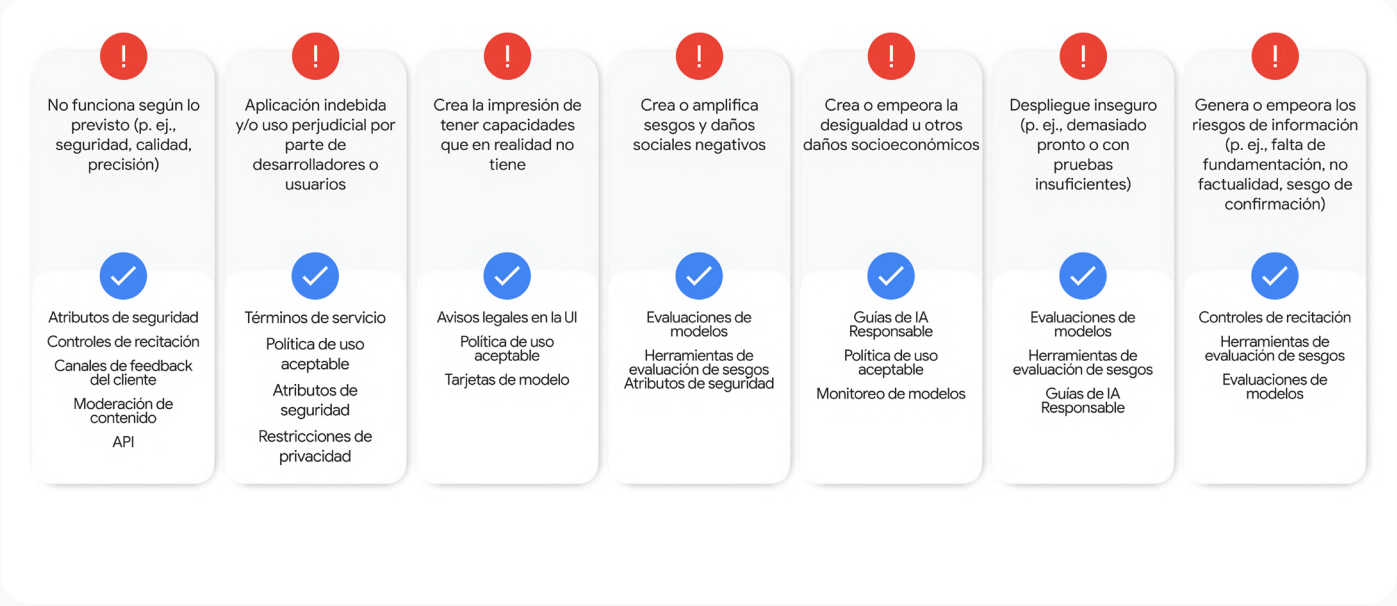
Al integrar el framework de desarrollo del ADK con la automatización operativa del Agent Starter Pack, se establece un proceso completo de MLOps/DevOps de extremo a extremo diseñado específicamente para crear y gestionar agentes de IA de grado de producción. Esto es AgentOps a escala.

3.2 Cómo desarrollar agentes de IA responsables y seguros con AgentOps

El desarrollo de agentes de IA avanzados conlleva la responsabilidad no negociable de garantizar que sean seguros, fiables y que estén alineados con los principios éticos. Esto significa diseñarlos desde su concepción con salvaguardas para prevenir resultados nocivos o no deseados, tales como sesgos injustos, violaciones a la privacidad y vulnerabilidades de seguridad.

Abordar este desafío requiere un enfoque estructurado. El siguiente diagrama proporciona una descripción general de alto nivel de los riesgos comunes, así como de los controles técnicos y de procedimiento utilizados para mitigarlos. Si bien este es un valioso punto de partida, para obtener una guía completa sobre estándares y mejores prácticas, recomendamos consultar el Secure AI Framework (SAIF) de Google.

> **A medida que los agentes de IA se integran en nuestras vidas, resulta crucial abordar los nuevos desafíos en torno a la confianza, la privacidad y la seguridad. Es imperativo pensar en estos factores desde el principio y preguntarnos constantemente: ¿cómo desarrollamos productos fiables?"**
>
> **Jia Li**
> Cofundadora, Presidenta y Directora de IA de LiveX AI

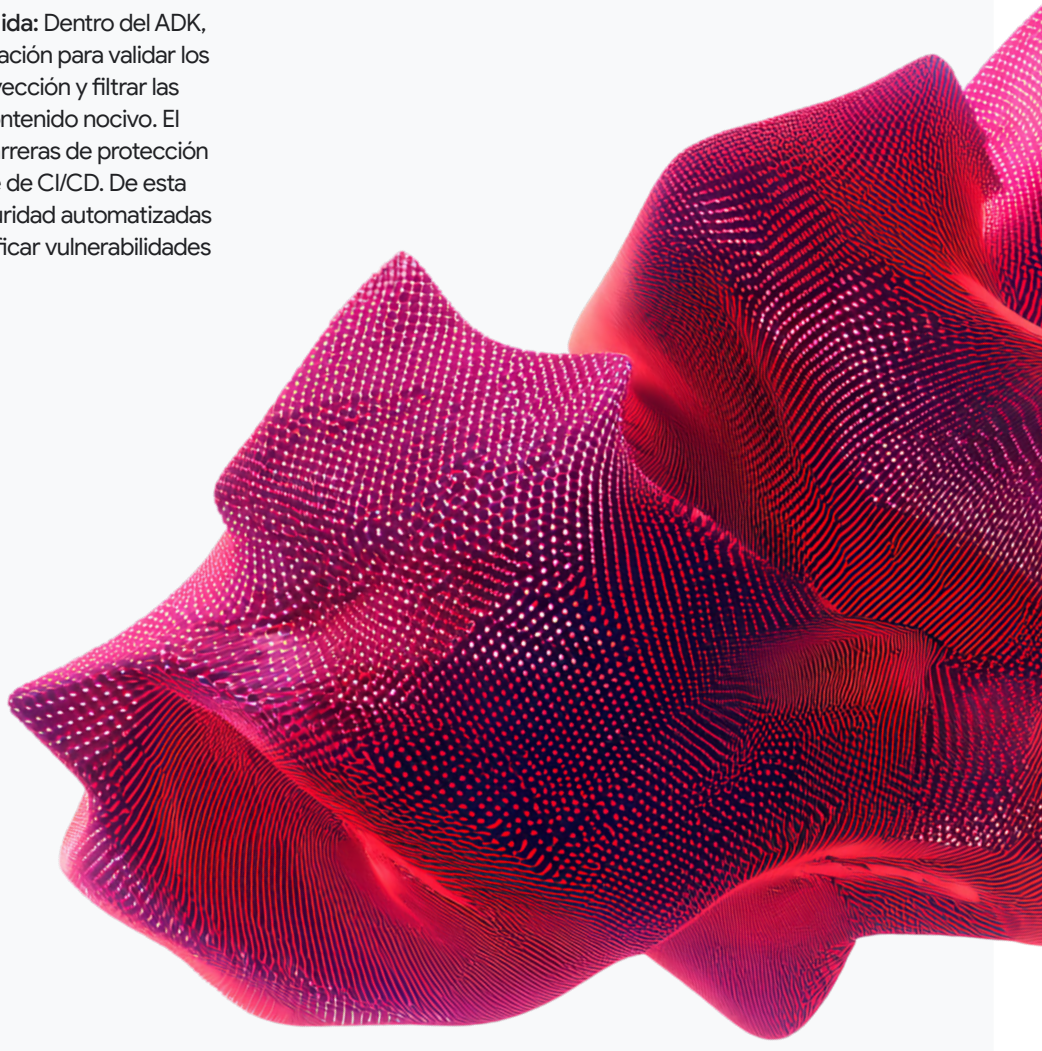


El ADK y el Agent Starter Pack proporcionan una estrategia de defensa en profundidad para esta área crítica. En primer lugar, el ADK permite implementar controles de seguridad granulares a nivel de aplicación. En segundo lugar, el Agent Starter Pack automatiza el despliegue de una infraestructura de nube fortificada que garantiza el cumplimiento de estos controles a escala.

Este enfoque combinado aborda aspectos clave de la seguridad y el cumplimiento normativo:

- **Infraestructura segura y control de acceso:** El Agent Starter Pack utiliza Terraform para aprovisionar una base segura, lo que permite desplegar tu agente en entornos como Cloud Run y configurar roles de IAM específicos para aplicar el principio de mínimo privilegio. Las herramientas que defines en el ADK operan, por tanto, bajo estos estrictos permisos a nivel de nube, lo que garantiza que el agente no pueda acceder a recursos no autorizados, incluso si su propia lógica se ve comprometida.
- **Barreras de protección de entrada y salida:** Dentro del ADK, es posible implementar la lógica de aplicación para validar los prompts frente a posibles ataques de inyección y filtrar las salidas finales del agente en busca de contenido nocivo. El Agent Starter Pack asegura que estas barreras de protección sean robustas y las integra en su pipeline de CI/CD. De esta forma, puedes ejecutar pruebas de seguridad automatizadas contra cada cambio de código para verificar vulnerabilidades antes de que lleguen a producción.
- **Auditoría y monitoreo:** La observabilidad detallada en el ADK crea una traza granular de cada pensamiento y llamada a herramienta que hace el agente. El Agent Starter Pack operacionaliza esto mediante la configuración automática de sumideros de registros que enrutan estos datos hacia BigQuery para un almacenamiento seguro a largo plazo. Esto crea la pista de auditoría persistente necesaria para las revisiones de cumplimiento normativo y la respuesta a incidentes.

La seguridad es una responsabilidad compartida. Si bien el ADK proporciona el framework para la arquitectura cognitiva de un agente y el Agent Starter Pack proporciona los componentes para desplegarlo, operan dentro del ecosistema más amplio de Google Cloud. Todo esto aporta una postura de seguridad formidable, desarrollada sobre una base segura desde el diseño, con controles integrados diseñados para defender cualquier carga de trabajo.



Puntos clave: construcción de agentes fiables

Tu objetivo	Mejor opción
Gestionar el ciclo de vida de tu agente de forma profesional.	Adopta AgentOps para automatizar procesos desde el desarrollo hasta el despliegue y el monitoreo.
Garantizar que tu agente sea preciso y seguro antes de lanzarlo a producción.	Implementa una evaluación automatizada en tu pipeline de CI/CD para probar con rigurosidad la calidad, la fundamentación y la seguridad.
Supervisar el rendimiento, el costo y los errores de tu agente en el mundo real.	Configura el monitoreo utilizando herramientas de observabilidad para obtener datos en tiempo real sobre la latencia, el uso de tokens y las tasas de éxito de las llamadas a herramientas.
Averiguar por qué tu agente tomó una decisión específica.	Inspecciona la trayectoria del agente (su "cadena de pensamiento") utilizando herramientas de registro y trazabilidad para depurar su proceso de razonamiento.
Proteger tu agente, sus datos y su acceso a herramientas.	Aplica los principios de seguridad de AgentOps, que incluyen seguridad de la infraestructura, gobernanza de datos y controles de cumplimiento.
Empezar rápidamente con AgentOps.	Usa el Agent Starter Pack para obtener plantillas preconfiguradas para CI/CD, evaluación e infraestructura.

¿Todo listo para convertir tu visión de IA en realidad? Estamos aquí para ayudarte.

Aprende a desarrollar más aplicaciones de IA generativa con las sesiones bajo demanda de Startup School.

Empieza ahora

Recibe hasta $350,000 USD en créditos de Google Cloud con el programa Google Cloud for Startups.

Solicita la inscripción ahora

Habla con nuestro equipo especializado en startups.

Comunícate con nosotros

Mantente al tanto de las últimas novedades y recibe todas nuestras actualizaciones suscribiéndote al boletín informativo de Google Cloud Startup.

Suscribirse

Más sobre el stack completo de IA de Google

Desarrolla rápidamente con Gemini en Google AI Studio.

Ver todos los modelos de Gemini disponibles.

Explorar

Destacado

Gemini 2.5 Flash Image (también conocido como Nano Banana)

Este modelo de generación y edición de imágenes permite combinar múltiples imágenes en una sola, mantener la consistencia del personaje para una narrativa enriquecida, realizar transformaciones dirigidas usando lenguaje natural y aprovechar el vasto conocimiento mundial de Gemini para generar y editar imágenes.

Destacado

Veo e Imagen

Estos modelos de vanguardia permiten generar videos e imágenes de alta calidad a partir de prompts de texto, editar elementos visuales existentes mediante instrucciones en lenguaje natural y crear experiencias de narrativa inmersivas gracias a capacidades avanzadas de síntesis visual.

PROMPT: El hombre mira hacia arriba y le sonríe a la cámara

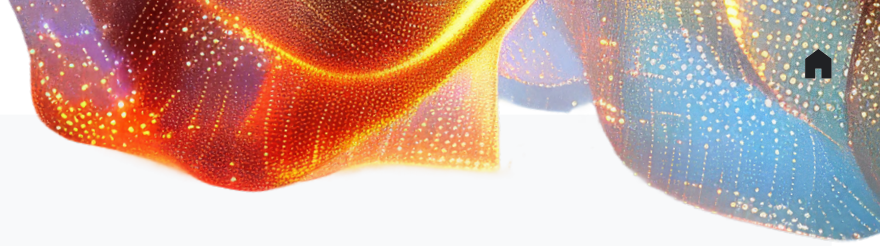
PROMPT: El perro se levanta, moviendo la cola, feliz y mirando a la cámara.

Conclusión

El recorrido desde un prototipo hasta un sistema de grado de producción requiere de una ingeniería disciplinada. Mediante la adopción de un framework centrado en el código como el ADK y los principios operativos de esta guía, es posible dejar a un lado la prueba superficial informal y seguir un proceso riguroso y fiable para desarrollar y gestionar todo el ciclo de vida de tu agente.

Para tu startup, este enfoque disciplinado se convierte en una poderosa ventaja competitiva. Tu equipo puede iterar e innovar más rápido al automatizar evaluaciones que consumen demasiados recursos en el camino. Además, esta metodología permite escalar con total confianza, sin comprometer la seguridad o la fiabilidad del sistema.

Como ha demostrado esta guía, Google Cloud respalda esta innovación: desde su hardware de IA diseñado específicamente para este propósito y su plataforma de datos unificada, hasta los modelos, servicios y herramientas necesarias para transformar tu concepto en un sistema de IA sofisticado. La plataforma es el cimiento; tu visión única y los principios descritos en esta guía son el plano arquitectónico. Juntos, forman la base para desarrollar la próxima generación de sistemas inteligentes que impulsarán el crecimiento de tu startup.



Recursos

- AdkApp: Desarrolla y despliega agentes en Vertex AI Agent Engine.
- Agent Development Kit (ADK): El ADK es un framework flexible y modular para desarrollar y desplegar agentes de IA.
- Agent2Agent (A2A): Un protocolo abierto que permite la comunicación e interoperabilidad entre aplicaciones agénticas opacas.
- Agent Starter Pack: Obtén agentes listos para producción en Google Cloud, más rápido. Pasa de la idea al despliegue de forma más rápida con plantillas y herramientas preconstruidas.
- BigQuery: BigQuery es el almacén de datos analítico totalmente gestionado, a escala de petabytes y rentable de Google Cloud que te permite ejecutar análisis sobre grandes cantidades de datos en tiempo casi real.
- Check grounding API: Como parte de tu experiencia RAG en aplicaciones de IA, puedes verificar la fundamentación para determinar qué tan fundamentado está un fragmento de texto (llamado "candidato a respuesta") en un conjunto determinado de textos de referencia (llamados "hechos").
- Cloud Functions API: Esta API gestiona funciones ligeras proporcionadas por el usuario que se ejecutan en respuesta a eventos.
- Cloud Run: Ejecuta servicios de frontend y backend, trabajos por lotes, aloja los LLM y pone en cola cargas de trabajo de procesamiento, sin la necesidad de gestionar infraestructura.
- Cloud Storage bucket: Los buckets son los contenedores básicos que guardan tus datos. Todo lo que almacenes en Cloud Storage debe estar contenido en un bucket.
- Colab Enterprise: Colab Enterprise es un entorno de notebooks gestionado y colaborativo con las capacidades de seguridad y cumplimiento normativo de Google Cloud.
- Example Store: El Almacén de Ejemplos te permite almacenar y recuperar dinámicamente ejemplos few-shot.
- Firestore: Firestore es una base de datos NoSQL altamente escalable para tus aplicaciones web y móviles.
- Gemini 2.5 Flash: Gemini 2.5 Flash está diseñado para buscar el equilibrio entre calidad, costo y velocidad.
- Gemini 2.5 Flash Image: (también conocido como Nano Banana): Gemini puede generar y procesar imágenes de forma conversacional. Puedes enviar prompts a Gemini con texto, imágenes o una combinación de ambos, lo que te permite crear, editar e iterar en visuales con un control sin precedentes.
- Gemini 2.5 Pro: Gemini 2.5 Pro es nuestro modelo Gemini de razonamiento más avanzado, capaz de resolver problemas complejos.
- Gemini CLI: Gratuito y de código abierto, lleva a Gemini 2.5 directamente a las terminales de los desarrolladores, con un acceso inigualable para individuos.
- Gemma: Un conjunto de modelos abiertos ligeros y de vanguardia desarrollado a partir de la misma tecnología que impulsa nuestros modelos Gemini.
- Servicio de evaluación de IA generativa: El servicio de evaluación de IA generativa en Vertex AI te permite evaluar cualquier modelo o aplicación generativa y comparar los resultados de la evaluación con tu propio juicio, utilizando tus propios criterios de evaluación.
- Google AI Studio: Google AI Studio es la forma más rápida de comenzar a desarrollar con Gemini, nuestra familia de modelos de IA generativa multimodal de última generación.
- Google Cloud Observability: Google Cloud Observability incluye servicios de observabilidad que te ayudan a comprender el comportamiento, la salud y el rendimiento de tus aplicaciones.
- Google Kubernetes Engine (GKE): GKE es el servicio de Kubernetes más escalable y totalmente automatizado. Pon tus contenedores en piloto automático y ejecuta de forma segura tus cargas de trabajo empresariales a escala, con poca o ninguna experiencia en Kubernetes.
- GraphRAG: GraphRAG en Google Cloud combina grafos de conocimiento con la Generación Aumentada por Recuperación (RAG) para mejorar la precisión, el contexto y la explicabilidad de los grandes modelos de lenguaje (LLM).
- Imagen: Imagen en Vertex AI lleva las capacidades avanzadas de IA generativa de imágenes de Google a los desarrolladores de aplicaciones.
- MCP Toolbox for Databases: Este es un servidor MCP de código abierto que te ayuda a desarrollar herramientas de IA generativa para que tus agentes puedan acceder a los datos en tu base de datos.
- Protocolo de Contexto de Modelo (MCP): El MCP es un protocolo abierto que estandariza la forma en que las aplicaciones proporcionan contexto a los LLM.

- Evaluación de modelos en Vertex AI: El servicio de evaluación de IA predictiva te permite evaluar el rendimiento del modelo en casos de uso específicos.
- Model Garden nn Vertex AI: Inicia tu proyecto de ML con un único lugar para descubrir, personalizar y desplegar una amplia variedad de modelos de Google y de sus socios.
- Ajuste fino de modelos: El ajuste de modelos es un proceso crucial en la adaptación de Gemini para realizar tareas específicas con mayor precisión y exactitud.
- ReAct: La orquestación con un agente ReAct (razonamiento + acción) implica una interacción de múltiples turnos entre una aplicación y un modelo (o modelos) donde el agente gestiona conversaciones, transacciones e instrucciones del LLM.
- IA Responsible (IAR): Para ayudar a los desarrolladores, el Vertex AI Studio tiene filtrado de contenido integrado, y nuestras APIs de IA generativa tienen una puntuación de atributos de seguridad para ayudar a los clientes a probar los filtros de seguridad de Google y definir los umbrales de confianza que son adecuados para su caso de uso y negocio.
- Generación Aumentada por Recuperación: La RAG es un framework de IA que combina las fortalezas de los sistemas tradicionales de recuperación de información (como búsquedas y bases de datos) con las capacidades de los LLM generativos.
- Base de datos vectorial: Una base de datos vectorial es cualquier base de datos que te permita almacenar, indexar y consultar embeddings vectoriales, o representaciones numéricas de datos no estructurados, como texto, imágenes o audio.
- Veo: Puedes usar Veo en Vertex AI para generar nuevos videos a partir de un prompt de texto o un prompt de imagen.
- Vertex AI Agent Engine: Vertex AI Agent Engine es un conjunto de servicios que permite a los desarrolladores desplegar, gestionar y escalar agentes de IA en producción.
- Vertex AI notebooks: Accede a todas las capacidades en la plataforma Vertex AI para trabajar en todo el flujo de trabajo de la ciencia de datos, desde la exploración de datos hasta el prototipo y la producción.
- Vertex AI Platform: Vertex AI es una plataforma de desarrollo de IA unificada y totalmente gestionada para crear y usar IA generativa.
- Vertex AI RAG Engine: Vertex AI RAG Engine es un framework de datos para desarrollar aplicaciones LLM aumentadas por contexto.
- Vertex AI Search: Vertex AI Search reúne el poder de la recuperación de información profunda, el procesamiento del lenguaje natural de vanguardia y lo último en procesamiento LLM para comprender la intención del usuario y devolver los resultados más relevantes para él.
- Vertex AI Studio: Optimiza tus flujos de trabajo de modelos fundacionales con Vertex AI Studio. Crea prototipos, refina y despliega modelos sin problemas en tus aplicaciones.

Google Cloud

¿Preguntas?

Habla con nuestro equipo especializado en startups.

Comunícate con nosotros