



Extracción Completa de la Documentación del Agent Development Kit (ADK) de Google

En esta extracción exhaustiva de la documentación del Agent Development Kit (ADK) de Google, se presenta un análisis detallado de esta plataforma integral diseñada para el desarrollo y despliegue de agentes de IA. El ADK representa una innovación significativa en el panorama de frameworks para agentes, ofreciendo una aproximación **code-first** que permite a desarrolladores crear sistemas agenticos sofisticados desde aplicaciones simples hasta arquitecturas multi-agente complejas.^{[1] [2]}

¿Qué es el Agent Development Kit?

El Agent Development Kit (ADK) es un framework **flexible y modular** para desarrollar y desplegar agentes de IA. Aunque está optimizado para Gemini y el ecosistema de Google, el ADK es **model-agnostic** (agnóstico de modelo), **deployment-agnostic** (agnóstico de despliegue), y está construido para la **compatibilidad con otros frameworks**. El ADK fue diseñado para hacer que el desarrollo de agentes se sienta más como desarrollo de software tradicional, facilitando a los desarrolladores crear, desplegar y orquestar arquitecturas agenticas que van desde tareas simples hasta flujos de trabajo complejos.^{[3] [1]}

Características Clave

Orquestación Flexible: Define flujos de trabajo usando workflow agents (Sequential, Parallel, Loop) para pipelines predecibles, o aprovecha el enrutamiento dinámico impulsado por LLM (LLMAgent transfer) para comportamiento adaptativo.^{[1] [3]}

Arquitectura Multi-Agente: Construye aplicaciones modulares y escalables componiendo múltiples agentes especializados en una jerarquía. Permite coordinación y delegación complejas.^{[3] [1]}

Rico Ecosistema de Herramientas: Equipa agentes con diversas capacidades: usa herramientas pre-construidas (Search, Code Exec), crea funciones personalizadas, integra bibliotecas de terceros (LangChain, CrewAI), o incluso usa otros agentes como herramientas.^{[1] [3]}

Listo para Despliegue: Conteneriza y despliega tus agentes en cualquier lugar – ejecuta localmente, escala con Vertex AI Agent Engine, o integra en infraestructura personalizada usando Cloud Run o Docker.^{[3] [1]}

Evaluación Integrada: Evalúa sistemáticamente el rendimiento del agente evaluando tanto la calidad de respuesta final como la trayectoria de ejecución paso a paso contra casos de prueba predefinidos.^{[1] [3]}

Tipos de Agentes en ADK

ADK proporciona diferentes categorías de agentes para construir aplicaciones sofisticadas.^[4]

Agentes LLM (LlmAgent, Agent)

Estos agentes utilizan Modelos de Lenguaje Grandes (LLMs) como su motor principal para entender lenguaje natural, razonar, planificar, generar respuestas y decidir dinámicamente cómo proceder o qué herramientas usar, haciéndolos ideales para tareas flexibles centradas en lenguaje.^[4]

Agentes de Flujo de Trabajo (SequentialAgent, ParallelAgent, LoopAgent)

Estos agentes especializados controlan el flujo de ejecución de otros agentes en patrones predeterminados y determinísticos (secuencia, paralelo, o bucle) sin usar un LLM para el control de flujo en sí, perfectos para procesos estructurados que requieren ejecución predecible.^[4]

Agentes Personalizados

Creados extendiendo BaseAgent directamente, estos agentes permiten implementar lógica operacional única, flujos de control específicos, o integraciones especializadas no cubiertas por los tipos estándar.^[4]

Ecosistema de Herramientas

En el contexto de ADK, una **Herramienta (Tool)** representa una capacidad específica proporcionada a un agente de IA, permitiéndole realizar acciones e interactuar con el mundo más allá de sus habilidades básicas de generación de texto y razonamiento.^[5]

Tipos de Herramientas

Herramientas de Función: Herramientas creadas por el desarrollador, adaptadas a las necesidades específicas de la aplicación, incluyendo funciones/métodos síncronos, agentes-como-herramientas, y herramientas de función de larga ejecución.^[5]

Herramientas Integradas: Herramientas listas para usar proporcionadas por el framework para tareas comunes, como Google Search, Ejecución de Código, y Generación Aumentada por Recuperación (RAG).^[5]

Herramientas de Terceros: Integración de herramientas de bibliotecas externas populares como LangChain Tools y CrewAI Tools.^[5]

Contexto de Herramientas (ToolContext)

Para escenarios avanzados, ADK permite acceso a información contextual adicional mediante el parámetro especial `tool_context: ToolContext`. Este contexto proporciona:^[5]

- **Gestión de Estado:** Acceso de lectura y escritura al estado de la sesión actual
- **Control de Flujo del Agente:** Capacidad de influir en las acciones subsiguientes del agente

- **Autenticación:** Manejo de credenciales y flujos de autenticación
- **Acceso a Servicios:** Interacción con Artifacts y Memory services

Opciones de Despliegue

El ADK ofrece múltiples opciones de despliegue basadas en las necesidades de preparación para producción o flexibilidad personalizada:^[1]

Vertex AI Agent Engine

Agent Engine es un servicio completamente gestionado y de auto-escalado en Google Cloud específicamente diseñado para desplegar, gestionar y escalar agentes de IA contruidos con frameworks como ADK. Es un servicio totalmente gestionado que maneja la infraestructura para escalar agentes en producción, permitiendo a los desarrolladores enfocarse en crear aplicaciones inteligentes e impactantes.^[6] ^[1]

Para desplegar en Agent Engine:

```
from vertexai import agent_engines

remote_app = agent_engines.create(
    agent_engine=app,
    requirements=["google-cloud-aiplatform[adk,agent_engines]"]
)
```

Cloud Run

Cloud Run es una plataforma gestionada de auto-escalado en Google Cloud que permite ejecutar agentes como aplicaciones basadas en contenedores. Se puede desplegar usando:^[1]^[5]

1. **CLI de ADK:** `adk deploy cloud_run` (recomendado para Python)
2. **gcloud CLI:** `gcloud run deploy` para mayor control

Google Kubernetes Engine (GKE)

GKE es un servicio gestionado de Kubernetes de Google Cloud que permite ejecutar agentes en un entorno contenerizado. Es una buena opción cuando se necesita más control sobre el despliegue o para ejecutar modelos abiertos.^[7] ^[1]

Infraestructura Compatible con Contenedores

Los agentes pueden empaquetarse manualmente en una imagen de contenedor y ejecutarse en cualquier entorno que soporte imágenes de contenedor, como Docker o Podman localmente.^[1]

Sistemas Multi-Agente

Los sistemas multi-agente en ADK son aplicaciones donde diferentes agentes, a menudo formando una jerarquía, colaboran o coordinan para lograr un objetivo más amplio. Estructurar aplicaciones de esta manera ofrece ventajas significativas como modularidad mejorada, especialización, reutilización, mantenibilidad y la capacidad de definir flujos de control estructurados.^[8]

Primitivos ADK para Composición de Agentes

ADK proporciona bloques de construcción centrales que permiten estructurar y gestionar interacciones dentro del sistema multi-agente:

- **Jerarquías de Agentes:** Usando el parámetro `sub_agents`
- **Agentes de Flujo de Trabajo:** Para control de ejecución estructurada
- **Mecanismos de Transferencia:** Para coordinación dinámica entre agentes

Framework de Evaluación

¿Por Qué Evaluar Agentes?

A diferencia del desarrollo de software tradicional donde las pruebas unitarias y de integración proporcionan una señal clara de "pasa/falla", los agentes LLM introducen un nivel de variabilidad que hace insuficientes los enfoques de prueba tradicionales. Debido a la naturaleza probabilística de los modelos, las afirmaciones determinísticas "pasa/falla" son a menudo inadecuadas para evaluar el rendimiento del agente.^[9]

Qué Evaluar

La evaluación de agentes puede desglosarse en dos componentes:^[9]

Evaluación de Trayectoria y Uso de Herramientas: Analizando los pasos que toma un agente para llegar a una solución, incluyendo su elección de herramientas, estrategias, y la eficiencia de su aproximación.^[9]

Evaluación de la Respuesta Final: Evaluando la calidad, relevancia, y corrección de la salida final del agente.^[9]

Métodos de Evaluación

ADK ofrece dos métodos para evaluar el rendimiento del agente:^[9]

1. **Usar un Archivo de Prueba:** Para pruebas unitarias durante desarrollo activo
2. **Usar un Archivo Evalset:** Para pruebas de integración más complejas

Criterios de Evaluación

- **tool_trajectory_avg_score:** Compara el uso real de herramientas del agente contra el uso esperado
- **response_match_score:** Compara la respuesta final del agente usando métricas ROUGE

Instalación y Configuración

Instalación Básica

```
# Crear un entorno virtual (recomendado)
python -m venv .venv
source .venv/bin/activate # En macOS/Linux

# Instalar ADK
pip install google-adk
```

Variables de Entorno

```
export GOOGLE_CLOUD_PROJECT=your-project-id
export GOOGLE_CLOUD_LOCATION=us-central1
export GOOGLE_GENAI_USE_VERTEXAI=True
```

Ejemplo Básico de Agente

```
from google.adk.agents import LlmAgent
from google.adk.tools import google_search

dice_agent = LlmAgent(
    model="gemini-2.0-flash-exp",
    name="question_answer_agent",
    description="A helpful assistant agent that can answer questions.",
    instruction="Respond to the query using google search",
    tools=[google_search],
)
```

Seguridad y Mejores Prácticas

ADK incluye características de seguridad integradas: [\[10\]](#)

- **Conformidad VPC-SC:** Para fortalecer la seguridad de datos
- **Configuración de Autenticación e IAM:** Control de acceso granular
- **Patrones de Seguridad:** Implementación de mejores prácticas de seguridad en el diseño del agente

Monitoreo y Observabilidad

El ADK proporciona capacidades integradas de monitoreo:^[10]

- **Google Cloud Trace:** Soporte para OpenTelemetry
- **Cloud Monitoring:** Métricas centralizadas
- **Cloud Logging:** Registro de eventos detallado
- **Soporte de Proveedores Externos:** Como AgentOps para observabilidad adicional

Casos de Uso y Aplicaciones

El ADK es utilizado internamente por productos de Google como Agentspace y Google Customer Engagement Suite (CES). Las aplicaciones típicas incluyen:^[11]

- **Sistemas de Soporte al Cliente:** Agentes que pueden acceder a bases de conocimiento
- **Asistentes de Investigación:** Agentes multi-herramienta que buscan y analizan datos
- **Aplicaciones de Voz/Video:** Aplicaciones interactivas usando streaming
- **Flujos de Trabajo Empresariales:** Procesos comerciales complejos con equipos de agentes especializados
- **Agentes de Análisis de Datos:** Sistemas que procesan y visualizan datos estructurados

Limitaciones y Consideraciones

- **Soporte de Versiones:** Agent Engine sólo soporta Python versión ≥ 3.9 y ≤ 3.13 ^[6]
- **Dependencias:** Los agentes requieren gestión cuidadosa de dependencias para builds reproducibles
- **Costos:** Algunos servicios como Vertex Gen AI Evaluation Service API requieren servicios pagos^[9]
- **Complejidad:** Los sistemas multi-agente requieren diseño y orquestación cuidadosa

Conclusión

El Agent Development Kit de Google representa una plataforma madura y comprensiva para el desarrollo de agentes de IA. Su aproximación code-first, combinada con capacidades robustas de despliegue y herramientas de evaluación integradas, lo posiciona como una solución empresarial para organizaciones que buscan implementar sistemas agenticos escalables. La flexibilidad del framework para soportar desde agentes simples hasta arquitecturas multi-agente complejas, junto con sus opciones de despliegue versátiles, lo hace adecuado para una amplia gama de casos de uso empresariales.^{[2] [1]}

La documentación revela un framework bien estructurado que aborda las necesidades reales del desarrollo de agentes en producción, desde la creación inicial hasta el despliegue, monitoreo y evaluación continua. Con su enfoque en la compatibilidad con otros frameworks y su naturaleza agnóstica de modelos, el ADK se presenta como una opción estratégica para organizaciones que buscan invertir en tecnologías de agentes de IA a largo plazo.

*
**

1. <https://google.github.io/adk-docs/deploy/>
2. <https://github.com/google/adk-docs>
3. <https://google.github.io/adk-docs/>
4. <https://google.github.io/adk-docs/agents/>
5. <https://google.github.io/adk-docs/deploy/cloud-run/>
6. <https://google.github.io/adk-docs/deploy/agent-engine/>
7. <https://google.github.io/adk-docs/deploy/gke/>
8. <https://google.github.io/adk-docs/agents/multi-agents/>
9. <https://google.github.io/adk-docs/evaluate/>
10. <https://cloud.google.com/vertex-ai/generative-ai/docs/agent-engine/overview>
11. <https://developers.googleblog.com/en/agent-development-kit-easy-to-build-multi-agent-applications/>