

# Guía para Desarrollo de Agentes de IA en Google Colab con Google ADK

12 de enero de 2026

## 1. Requerimientos Previos

### 1.1. Requerimientos de Hardware en Google Colab

- **Tipo de entorno:** Se recomienda utilizar el entorno de ejecución con GPU (NVIDIA Tesla T4, P100, V100 o similar) para tareas que involucren modelos grandes.
- **Memoria RAM:** Mínimo 12 GB (preferiblemente 25+ GB para agentes complejos).
- **Almacenamiento:** El espacio gratuito es limitado (~80 GB). Para proyectos grandes, considere montar Google Drive.

### 1.2. Requerimientos de Software y Configuración

- **Cuenta de Google:** Necesaria para acceder a Colab y Google Cloud APIs.
- **Google Cloud Project:** Se requiere un proyecto en Google Cloud Platform (GCP) con las APIs habilitadas.
- **Autenticación:** Credenciales de servicio GCP en formato JSON.
- **Versión de Python:** 3.9 o superior (predeterminado en Colab).
- **Sistema operativo:** Ubuntu 20.04+ (base de Colab).

## 2. Paso a Paso de Configuración y Desarrollo

### 2.1. Paso 1: Configuración Inicial del Entorno Colab

1. Abre Google Colab: <https://colab.research.google.com/>
2. Crea un nuevo notebook: *Archivo* → *Nuevo notebook*
3. Cambia el entorno de ejecución: *Runtime* → *Change runtime type* → *GPU (T4)*
4. Verifica la GPU asignada:

```
1 !nvidia-smi -L
```

## 2.2. Paso 2: Instalación de Dependencias

```
1 # Actualizar pip e instalar ADK
2 !pip install --upgrade pip
3 !pip install google-cloud-aiagent
4 !pip install google-cloud-aiplatform
5
6 # Instalar dependencias adicionales
7 !pip install langchain
8 !pip install google-generativeai
9 !pip install langchain-google-vertexai
```

## 2.3. Paso 3: Autenticación en Google Cloud

```
1 from google.colab import auth
2 auth.authenticate_user()
3
4 # Montar Google Drive (opcional)
5 from google.colab import drive
6 drive.mount('/content/drive')
7
8 # Configurar variables de entorno
9 import os
10 os.environ['GOOGLE_CLOUD_PROJECT'] = 'tu-project-id' # Reemplaza con tu ID
11
12 # Autenticar con cuenta de servicio (alternativa)
13 # Sube tu archivo JSON de credenciales
14 from google.cloud import aiplatform
15 from google.oauth2 import service_account
16
17 credentials = service_account.Credentials.from_service_account_file(
18     '/content/credenciales.json' # Ruta a tu archivo JSON
19 )
20 aiplatform.init(project='tu-project-id', credentials=credentials)
```

## 2.4. Paso 4: Verificar Versiones y Accesos

```
1 import sys
2 print(f"Python: {sys.version}")
3
4 import google.cloud.aiagent as aiagent
5 print(f"ADK Version: {aiagent.__version__}")
6
7 # Verificar APIs habilitadas
8 !gcloud services list --enabled --project=$GOOGLE_CLOUD_PROJECT
```

## 2.5. Paso 5: Crear un Agente Básico con ADK

```
1 from google.cloud import aiagent
2 from google.cloud.aiagent import components
3
4 # 1. Inicializar el agente
5 agent = aiagent.Agent(
6     name="mi-primer-agente",
7     model="gemini-1.5-pro", # o "gemini-1.5-flash"
8     instructions="Eres un asistente util especializado en analisis de datos."
9 )
10
11 # 2. Agregar herramientas (ejemplo: busqueda web)
12 search_tool = components.WebSearchTool(
13     api_key="TU_API_KEY", # Reemplazar con clave real
```

```

14     custom_search_engine_id="TU_CSE_ID"
15 )
16 agent.add_tool(search_tool)
17
18 # 3. Agregar memoria
19 memory = components.ConversationMemory(
20     max_turns=10,
21     storage_path="/content/memory.json"
22 )
23 agent.add_component(memory)
24
25 # 4. Configurar el planificador
26 planner = components.ReActPlanner()
27 agent.add_component(planner)

```

## 2.6. Paso 6: Probar el Agente

```

1 # Probar con una consulta simple
2 response = agent.run("Cual es el precio actual de Bitcoin?")
3 print("Respuesta del agente:", response.text)
4
5 # Ver historial de conversacion
6 history = memory.get_history()
7 for turn in history:
8     print(f"{turn.role}: {turn.content}")

```

## 2.7. Paso 7: Despliegue en Vertex AI (Opcional)

```

1 # Exportar el agente
2 agent.export_to_vertex(
3     project_id=os.environ['GOOGLE_CLOUD_PROJECT'],
4     region="us-central1",
5     display_name="MiAgenteColab"
6 )
7
8 # O crear endpoint directo
9 endpoint = agent.deploy(
10     machine_type="n1-standard-4",
11     min_replica_count=1,
12     max_replica_count=3
13 )
14 print(f"Endpoint creado: {endpoint.resource_name}")

```

## 3. Consideraciones Finales

- **Costos:** Monitorea el uso en Google Cloud Console. Los modelos Gemini y Vertex AI tienen costos asociados.
- **Límites de Colab:** Las sesiones gratuitas expiran a las 12 horas. Guarda el estado en Drive.
- **Mejores prácticas:**
  1. Guarda el notebook frecuentemente (Ctrl+S).
  2. Usa celdas de markdown para documentación.
  3. Prueba con consultas simples antes de implementar lógica compleja.
  4. Revoca las credenciales de servicio cuando no las uses.
- **Solución de problemas:**

- Error de autenticación: Verifica que las APIs estén habilitadas.
- Falta de memoria: Reduce el tamaño del modelo o usa batch processing.
- Timeout: Usa `!kill -9 -1` para reiniciar el entorno.

## 4. Recursos Adicionales

- Documentación oficial ADK: <https://cloud.google.com/agent-development-kit>
- Repositorio GitHub: <https://github.com/google-cloud-ai/agent-development-kit>
- Ejemplos de Colab: <https://github.com/googlecolab/colabtools>
- Foro de soporte: <https://stackoverflow.com/questions/tagged/google-colab>