PROFESSIONAL AGENTIC ARCHITECT · CHEAT SHEET · CERTIFYDECK

Google Cloud Certified Professional Agentic Architect

Dominios, pesos oficiales y habilidades clave del temario de Professional Agentic Architect, para repasar antes del examen.

Nivel: Advanced (Professional) Duración: 120 min Fuente: guía de estudio oficial de Microsoft Learn

1. Building agents using low-code tools

13–13%

Configuring agentic workflows and behavior using low-code tools

- En los flujos de Dialogflow CX cada página es un estado: las rutas (de intención o de condición) y los controladores de eventos deciden respuestas y transiciones según su ámbito y orden de evaluación. CX Agent Studio reutiliza esos flujos como agentes basados en flujos, con variables de entrada y salida explícitas, y Workflow Builder (antes Agent Designer) encadena activadores, pasos, condiciones y aprobaciones humanas.
- Las instrucciones del sistema fijan rol, objetivos, restricciones y formato, pero no impiden del todo las filtraciones. CX Agent Studio reestructura las instrucciones en XML, admite instrucciones globales, componentes modulares y ejemplos few-shot con tool_code y tool_outputs. Los ejemplos deben usarse con moderación y formato coherente, y en la cadena de pensamiento el razonamiento va antes de la respuesta final.

Connecting enterprise data to Gemini Enterprise

- Gemini Enterprise conecta datos mediante almacenes de datos: la federación consulta en origen con las credenciales del usuario y mantiene sus permisos, mientras que la ingesta indexa para mejorar la búsqueda pero no copia los permisos de BigQuery o Cloud SQL. El control de acceso exige un proveedor de identidad y ACL, y las restricciones de organización, VPC Service Controls y Agent Gateway gobiernan qué fuentes y servidores MCP se conectan.
- Para documentos con imágenes, Gemini Enterprise ofrece el analizador OCR (más barato, solo PDF) y el analizador de diseño, que requiere fragmentación y puede anotar imágenes y tablas. Los modelos Gemini de Agent Platform procesan vídeo y audio desde Cloud Storage, con recortes, fps ajustables y marcas de tiempo; los permisos del bucket no se importan.

2. Using coding agents for application development

17–17%

Using coding agents effectively

- Antigravity define servidores MCP en mcp_config.json (stdio local o Streamable HTTP remoto), con ADC, OAuth, enabledTools y reglas Deny > Ask > Allow; las habilidades del espacio de trabajo van en .agents/skills. Los servidores MCP de Google Cloud exigen mcp.tools.call y permisos sobre el recurso, admiten toolsets y políticas de denegación de IAM; Claude Code usa Agent Platform con roles/aiplatform.user.
- Agent Sandbox en GKE aísla el código no fiable con gVisor o microVM, usa SandboxTemplate, SandboxWarmPool y SandboxClaim y crea una NetworkPolicy de denegación por plantilla; Agent Substrate suspende agentes inactivos. Cloud Workstations en un perímetro de VPC Service Controls exige clúster privado sin IP públicas, y el entorno aislado de Antigravity bloquea archivos sensibles y limita la red.
- En Antigravity, /plan explora el código y genera un plan revisable antes de editar, /boost resuelve con varios agentes errores de concurrencia, optimizaciones y refactorizaciones complejas, y /teamwork coordina migraciones de varios días. CodeMender encuentra vulnerabilidades, las verifica con un exploit de prueba de concepto y genera parches probados; el análisis diff controla las solicitudes de cambio y el profundo sirve para auditorías.

Customizing coding agents for enterprise workflows

- En Antigravity, las reglas (AGENTS.md o .agents/rules con disparadores always_on, model_decision, glob o manual) fijan restricciones y las habilidades (carpetas con SKILL.md) describen flujos de varios pasos. Los hooks de hooks.json ejecutan scripts antes o después de las acciones del agente, los subagentes se definen en .agents/agents y un plugin con plugin.json empaqueta todo en una unidad versionada.
- Agents CLI instala en Antigravity y otros agentes de programación siete habilidades del ciclo de vida de ADK y también funciona solo desde la terminal. Sus comandos generan el proyecto, evalúan, despliegan en Agent Runtime, Cloud Run o GKE con Agent Identity y Agent Gateway, publican en Gemini y activan trazas y analítica en BigQuery; las extensiones sustituyen o añaden comandos fijados a un commit.

3. Developing custom agents

33–33%

Designing and building agentic workflows in code

- Elige el modelo más barato que cumpla calidad y latencia y enruta: las tareas sencillas a un modelo pequeño y el razonamiento complejo a uno mayor. Los modelos gestionados (Gemini, partners, MaaS) minimizan la operación y encajan con tráfico variable; autodesplegar un modelo abierto desde Model Garden o en GKE da control de pesos, hardware y red para normativas estrictas y volumen predecible.
- En ADK, un LlmAgent combina modelo, instrucciones, herramientas y esquemas de salida; el LLM elige herramientas por su nombre, docstring y parámetros. Los callbacks (before/after de agente, modelo y herramienta) observan o sustituyen pasos para crear barreras deterministas, y los plugins se registran una vez en el Runner o App para políticas globales. Los artefactos guardan archivos binarios.
- La sesión guarda eventos y estado de una conversación; en producción conviene un servicio persistente como Agent Platform Sessions, automático al desplegar en Agent Runtime. Los prefijos user:, app: y temp: fijan el alcance del estado. Memory Bank extrae con un LLM hechos relevantes, los consolida por ámbito y los recupera por similitud, con temas, TTL y generación asíncrona configurables.
- Agents CLI funciona con un agente de programación, al que agents-cli setup instala siete habilidades (global por defecto, --workspace para el proyecto, --agent para elegir agente), o sin él, desde el terminal: -i activa las preguntas para uso humano y -y las omite. Las extensiones sustituyen o añaden comandos y copian habilidades en .agents/skills, fijadas a un commit que agents-cli install restaura.

Integrating enterprise domain knowledge

- RAG Engine cubre ingesta, fragmentación (tamaño y solapamiento), embeddings, indexación en un corpus y recuperación, con reordenación por la API de ranking o un LLM; el modelo de embeddings queda fijo para el corpus. Vector Search ofrece índices ANN con filtros y búsqueda híbrida densa y dispersa; Agent Retrieval gestiona colecciones de objetos con embeddings automáticos, fusión RRF y VertexRanker.
- Agent Identity da a cada agente una identidad SPIFFE propia, no compartida, sin claves de larga duración y que no se puede suplantar; se le conceden roles con políticas de permisos sobre su principal (principalSet para todo el proyecto) y se le niegan con políticas de denegación. Cada redespliegue crea un principal nuevo, y borrar el agente no elimina sus vinculaciones de IAM.
- Los servidores MCP remotos de Google Cloud (Cloud SQL, BigQuery…) se activan con su API, se registran solos en Agent Registry y se controlan con IAM (rol MCP Tool User, políticas de denegación de solo lectura) y Model Armor. Las API propias se publican como MCP con Apigee o Cloud Run; los servidores externos se registran a mano con toolspec.json y sus credenciales van en auth manager mediante vinculaciones.

Orchestrating and coordinating agentic workflows

- MCP conecta un agente con sus herramientas y recursos; A2A conecta agentes independientes y opacos que no comparten memoria ni herramientas, mediante tarjetas de agente, tareas con estado, contextId, streaming SSE y notificaciones push. En ADK, to_a2a() expone un agente y RemoteA2aAgent lo consume; los subagentes locales siguen siendo mejores para lógica interna de baja latencia y estado compartido.
- ADK ofrece SequentialAgent, ParallelAgent y LoopAgent deterministas, flujos en grafo con rutas, JoinNode y nodos de intervención humana, y equipos colaborativos con modos chat, task o single_turn. En Google Cloud, Agent Registry resuelve subagentes y herramientas en tiempo de ejecución, Agent Identity da a cada agente su propio principal de IAM y Agent Gateway bloquea por defecto lo que una política no permita.

4. Evaluating and deploying agentic workflows

22–22%

Evaluating agents in development and in production

- En ADK, cada caso de un evalset guarda por turno la consulta, la trayectoria de herramientas esperada, las respuestas intermedias y la respuesta final, más un estado inicial de sesión; adk web convierte sesiones reales en casos. Para casos límite sin escribirlos a mano, la simulación de usuario genera escenarios (mensaje inicial, plan y perfil) y la simulación del entorno inyecta errores de herramientas.
- En CI, ADK evalúa con pytest (AgentEvaluator), adk eval o adk conformance test; tool_trajectory_avg_score comprueba la trayectoria con coincidencia EXACT, IN_ORDER o ANY_ORDER. En producción, los monitores online de Agent Platform muestrean trazas de Cloud Trace, puntúan métricas como Tool Use Quality y exportan las puntuaciones a Cloud Monitoring, donde las políticas de alertas avisan de la deriva.
- Los evalsets de ADK sirven para evaluar agentes en desarrollo con criterios deterministas (trayectoria, ROUGE-1) o con un LLM como juez (rúbricas, alucinaciones). Gen AI evaluation service ofrece rúbricas adaptativas (recomendadas), estáticas y métricas basadas en cálculo; para criterios propios se registran métricas LLM o de código personalizadas en el Metric Registry y se reutilizan offline y online.
- En ADK, el conjunto de datos de referencia se guarda en archivos de prueba (una sesión, pruebas unitarias) o en evalsets (varias sesiones largas, integración), con las llamadas a herramientas esperadas y la respuesta de referencia. Los criterios tool_trajectory_avg_score (EXACT, IN_ORDER, ANY_ORDER), final_response_match_v2 y hallucinations_v1 miden la recuperación, el significado y la fundamentación.

Deploying and scaling production workloads

- Agent Runtime es el entorno totalmente gestionado con Sessions, Memory Bank y observabilidad integrados, pero no aloja servidores MCP propios. Cloud Run ofrece contenedores sin servidor que escalan a cero y necesitan almacenamiento externo para el estado. GKE da el máximo control (nodos, aceleradores, políticas de red) a cambio de más carga operativa.
- Las trazas de Cloud Trace muestran dónde se va el tiempo entre llamadas al modelo y a herramientas; las herramientas síncronas impiden la ejecución en paralelo. Los bucles se acotan con max_iterations o escalate en LoopAgent y con max_llm_calls en RunConfig. La deriva se detecta con monitores online y alertas de calidad, y los fallos de despliegue se diagnostican en los registros enlazados.
- Agent Platform ofrece vistas de observabilidad (latencia por herramienta y modelo, tokens, uso) y trazas con entradas, respuestas y llamadas a herramientas para hallar errores de lógica. En Agent Runtime, min_instances mitiga el arranque en frío y container_concurrency las colas. Monitores online y alertas de calidad detectan deriva y alucinaciones; en ADK, hallucinations_v1 y los criterios de eficiencia miden fundamentación y coste.

5. Securing and governing agentic workflows

15–15%

Configuring agent security and governance

- Agent Identity da a cada agente una identidad SPIFFE no compartida, sin claves de larga duración y con tokens vinculados a su certificado. Con autoridad propia usa su identidad ante Google Cloud, tokens de ID de OIDC, OAuth de 2 patas o claves de API; en nombre de un usuario, OAuth de 3 patas con consentimiento. Auth manager guarda las credenciales, automatiza consentimiento y renovación, y ADK inyecta las cabeceras.
- Una política de límite de acceso de principales (PAB) se vincula a conjuntos de principales, como las identidades de agente de un proyecto, y define a qué recursos pueden acceder, aunque tengan roles en otros sitios. No concede acceso, es aditiva, falla en cerrado y depende de su versión de aplicación. En los agentes se evalúa en las llamadas salientes y Agent Gateway puede aplicarla mediante IAP.
- Agent Gateway es el punto de entrada y salida del tráfico agéntico: exige una Agent Identity por agente, comprueba los destinos en Agent Registry y deniega por defecto lo que ninguna política de IAM permite. Conviene probar las políticas en modo de prueba, que solo registra infracciones, antes de ENFORCE. Sus logs registran decisiones, herramienta MCP y recurso del registro, y el trazado con OpenTelemetry se correlaciona con ellos.
- Agent Registry cataloga agentes, servidores MCP, endpoints y skills; las políticas de IAM se vinculan al nombre de recurso registrado y al principal del agente, nunca al URN, y un límite de gestión agrega componentes de varios proyectos. Model Armor analiza peticiones y respuestas con plantillas; los ajustes mínimos (floor settings) impiden plantillas menos estrictas y conviene empezar con Inspect only antes de bloquear.

Implementing secure agent behavior and execution

- Agent Gateway aplica las políticas: IAP con REQUEST_AUTHZ para la identidad, y Model Armor o la gobernanza semántica con CONTENT_AUTHZ; se prueba en DRY_RUN antes de ENFORCE. La gobernanza semántica evalúa cada llamada frente a la intención del usuario y reglas en lenguaje natural. Para la intervención humana, ADK ofrece confirmación de herramientas y nodos RequestInput, y Workflow Builder pasos de revisión.
- Agent Identity separa tres identidades: la del usuario, verificada en la entrada por Agent Gateway; la del agente, para su propia autoridad; y la delegada, con OAuth de 3 pasos gestionado por auth manager, que exige cumplir a la vez las políticas del agente y los permisos del usuario y registra ambos. Agent Registry vincula agentes con endpoints y proveedores de autenticación sin secretos en el código.