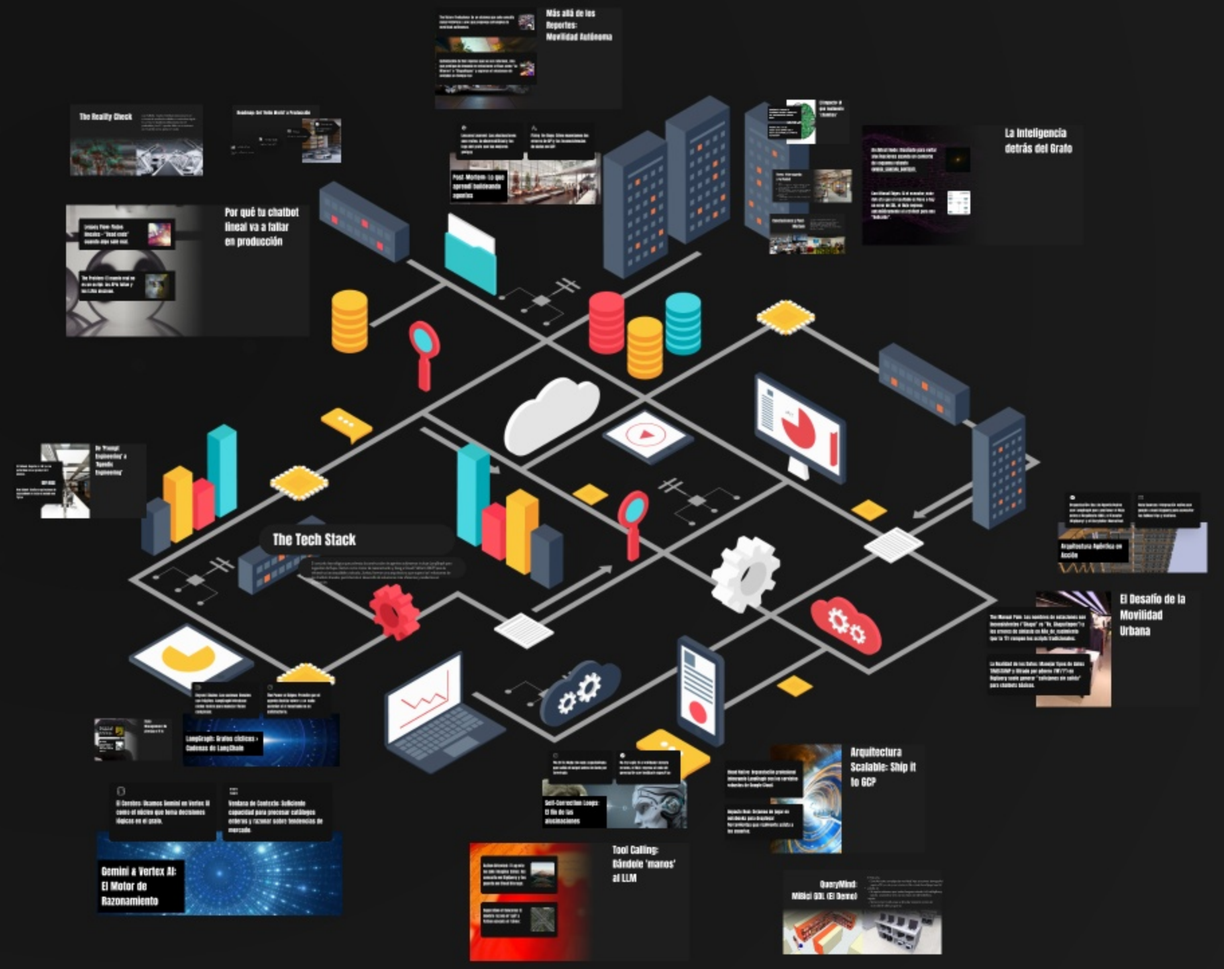
Más allá del Chat: Construyendo Agentes Autónomos con Python y Google Cloud

Presentador: Jose Muñoz



Roadmap: Del 'Hello World' a Producción

The Reality Check

Por qué el chat lineal está muerto en prod-

The Tech Stack

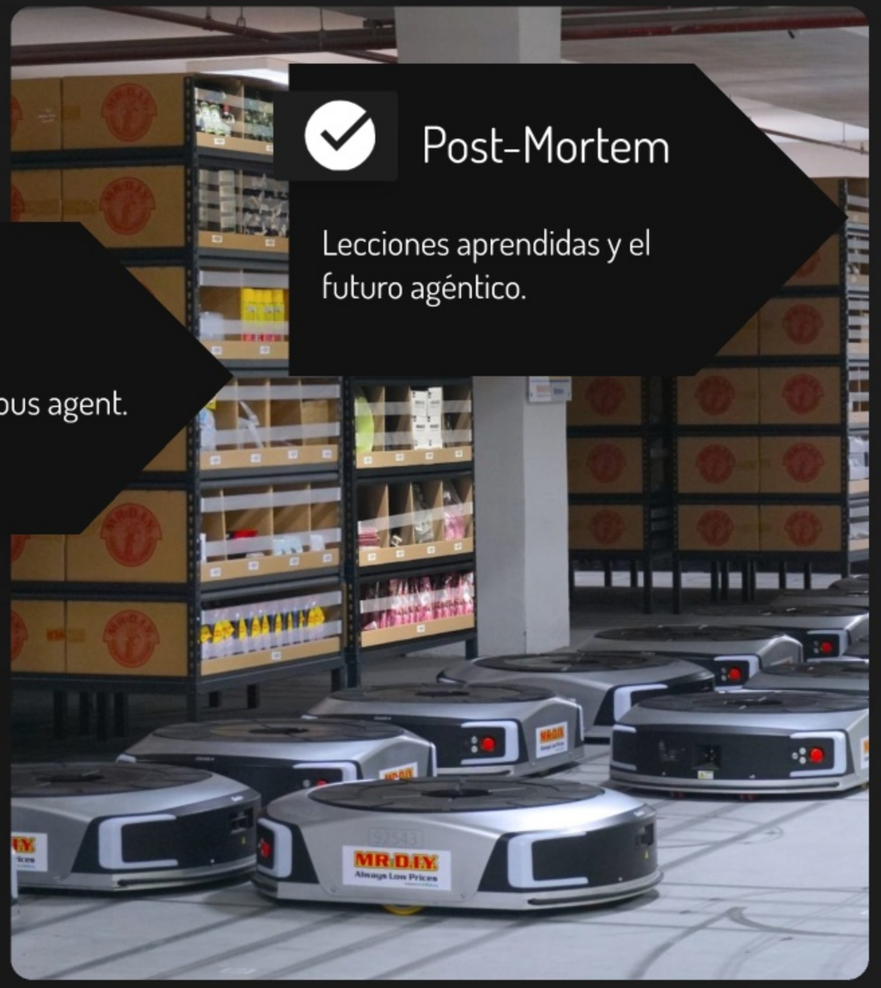
LangGraph + Gemini + GCP

The Lab

Building a real autonomous agent.

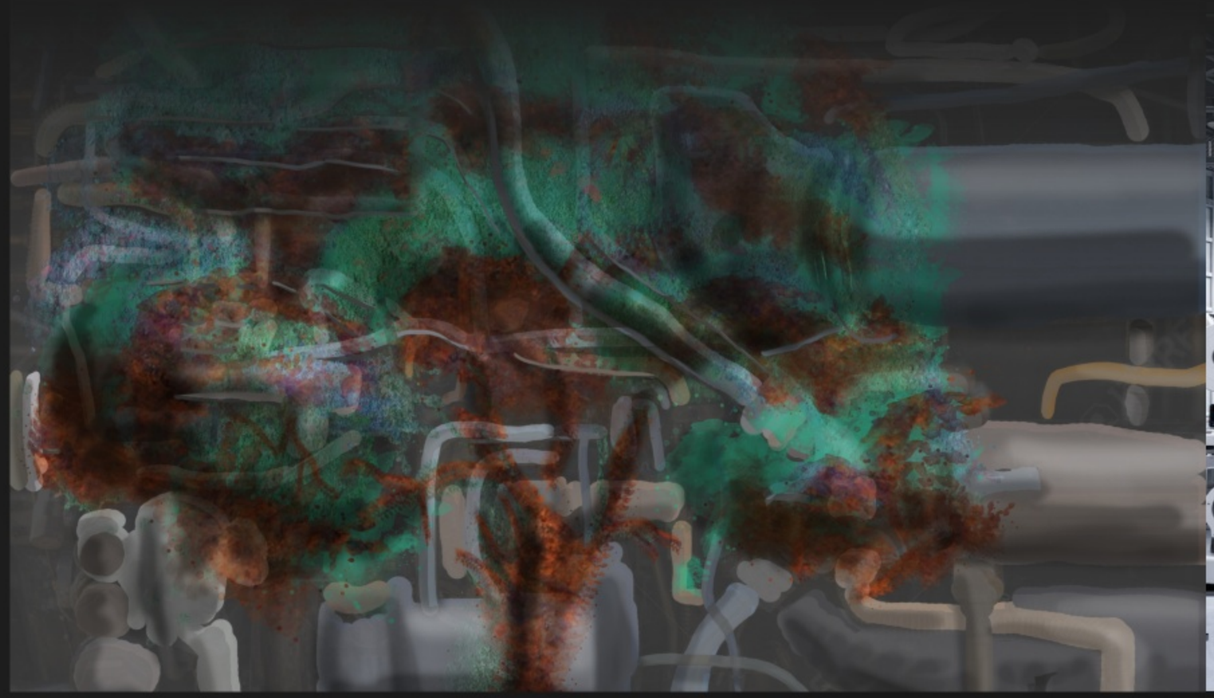
Post-Mortem

Lecciones aprendidas y el futuro agéntico.



The Reality Check

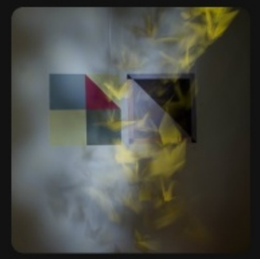
Los chatbots lineales enfrentan serios desafíos en entornos de producción debido a su estructura rígida. En un mundo donde las interacciones no son predecibles y las APIs pueden fallar, estos sistemas se convierten en callejones sin salida.



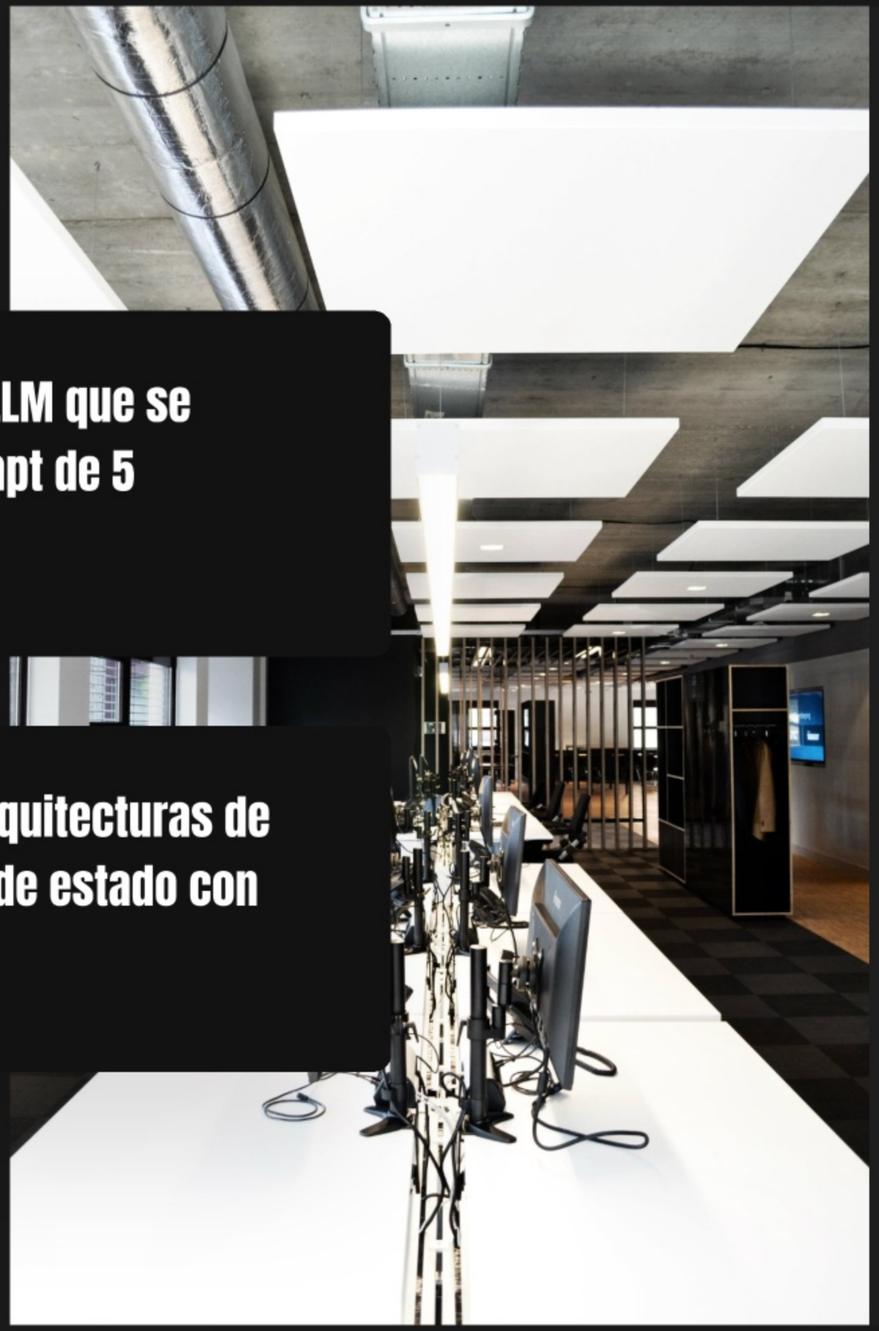
Por qué tu chatbot lineal va a fallar en producción

Legacy Flow: Flujos lineales = "Dead ends" cuando algo sale mal.

The Problem: El mundo real no es un script; las APIs fallan y los LLMs alucinan.



De 'Prompt Engineering' a 'Agentic Engineering'



Old School: Rogarle al LLM que se porte bien con un prompt de 5 páginas.

New School: Diseñar arquitecturas de razonamiento y ciclos de estado con Python.

The Tech Stack

El conjunto tecnológico que potencia la construcción de agentes autónomos incluye LangGraph para la gestión de flujos, Gemini como motor de razonamiento y Google Cloud Platform (GCP) para la infraestructura escalable y robusta. Juntos, forman una arquitectura que supera las limitaciones de los chatbots lineales, permitiendo el desarrollo de soluciones más eficientes y resilientes en producción.

Beyond Chains: Las cadenas lineales son frágiles. LangGraph introduce ciclos reales para manejar flujos complejos.

The Power of Edges: Permite que el agente decida volver a un nodo anterior si el resultado no es satisfactorio.

LangGraph: Grafos cíclicos > Cadenas de LangChain

State Management: No pierdas el hilo

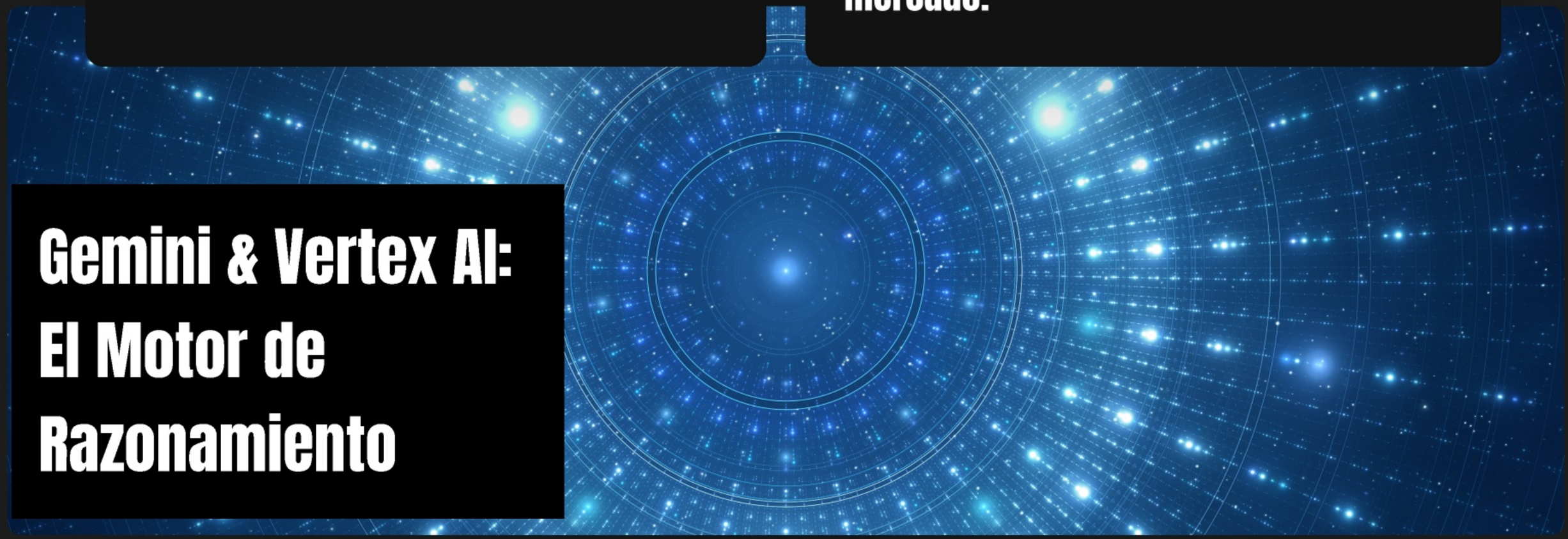
Shared State: Un objeto de estado persistente que evoluciona con cada paso del agente.

Snapshots:
Capacidad de pausar, revisar y retomar el proceso sin perder el progreso.

El Cerebro: Usamos Gemini en Vertex AI como el núcleo que toma decisiones lógicas en el grafo.

0101
1001

Ventana de Contexto: Suficiente capacidad para procesar catálogos enteros y razonar sobre tendencias de mercado.



Gemini & Vertex AI: El Motor de Razonamiento

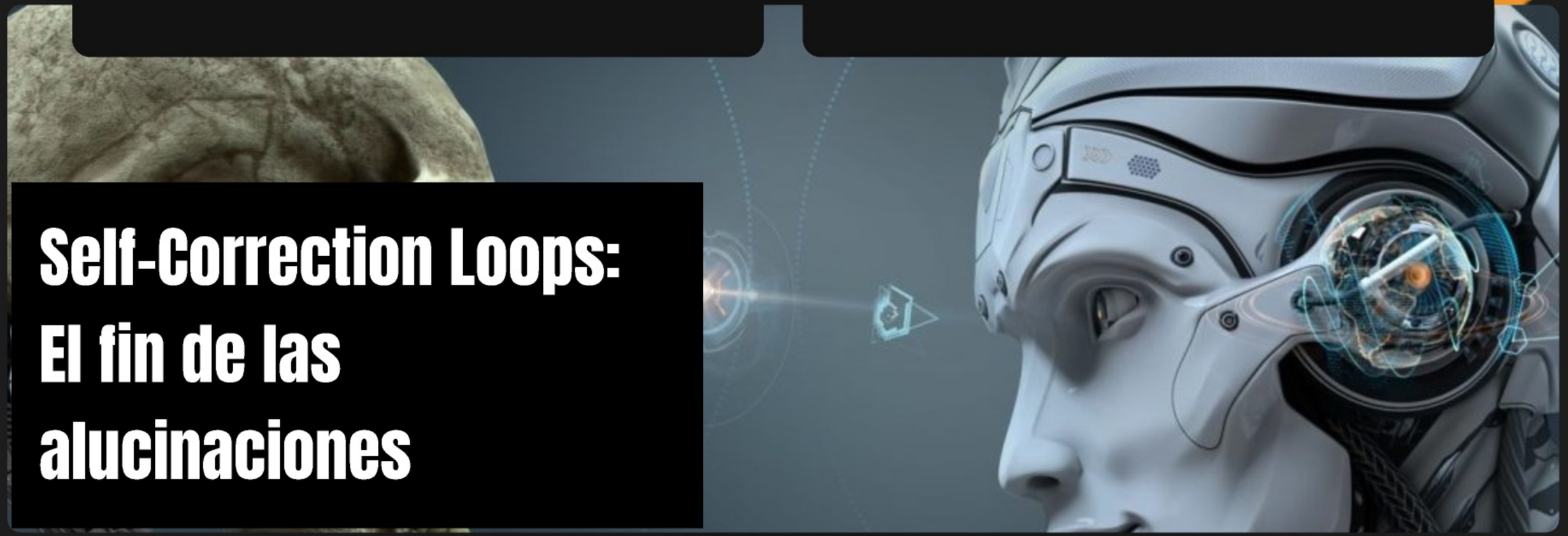
Tool Calling: Dándole 'manos' al LLM

Action Oriented: El agente no solo imagina datos; los consulta en BigQuery y los guarda en Cloud Storage.

Separation of Concerns: El modelo razona el 'qué' y Python ejecuta el 'cómo'.

The Critic Node: Un nodo especializado que valida el output antes de darlo por terminado.

Re-try Logic: Si el validador detecta errores, el flujo regresa al nodo de generación con feedback específico.



Self-Correction Loops: El fin de las alucinaciones

Arquitectura Scalable: Ship it to GCP

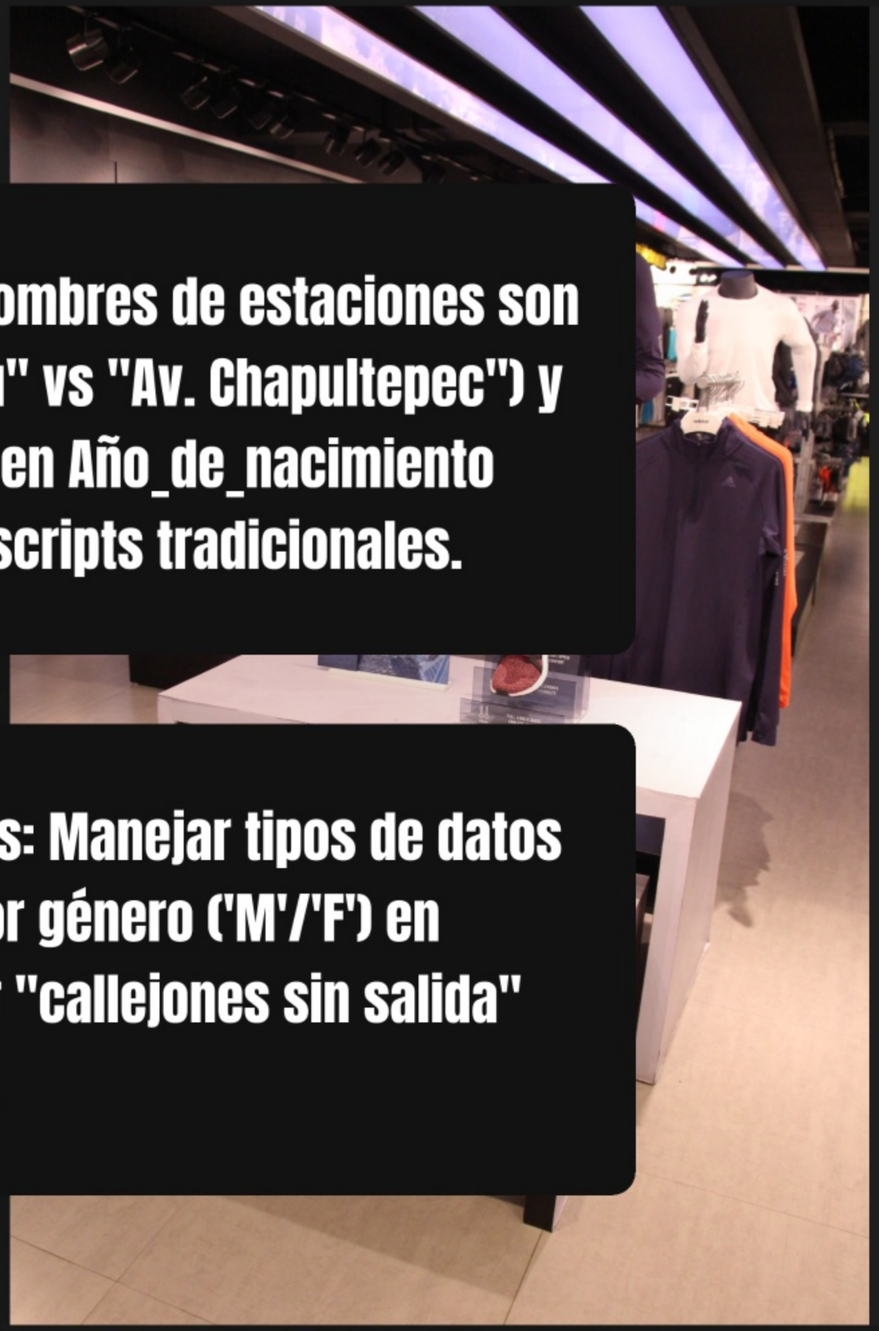
Cloud Native: Orquestación profesional integrando LangGraph con los servicios robustos de Google Cloud.

Impacto Real: Dejamos de jugar en notebooks para desplegar herramientas que realmente asista a los usuarios.

QueryMind: MiBici GDL (El Demo)

- El Problema:
  - Consultar datos complejos de movilidad (trips, estaciones, demografía) requiere SQL preciso y conocimiento del contexto local (jerga tapatía).
- La Solución:
  - Un agente autónomo que traduce lenguaje natural a SQL de BigQuery, ejecuta la consulta y narra los resultados con identidad local.
- Impacto:
  - Democratización del acceso a datos de transporte público sin necesidad de saber programar.

El Desafío de la Movilidad Urbana



The Manual Pain: Los nombres de estaciones son inconsistentes ("Chapu" vs "Av. Chapultepec") y los errores de sintaxis en Año_de_nacimiento (por la 'ñ') rompen los scripts tradicionales.

La Realidad de los Datos: Manejar tipos de datos TIMESTAMP y filtrado por género ('M'/'F') en BigQuery suele generar "callejones sin salida" para chatbots básicos.

Orquestación: Uso de AgenticEngine con LangGraph para gestionar el flujo entre el Arquitecto (SQL), el Ejecutor (BigQuery) y el Storyteller (Narrativa).

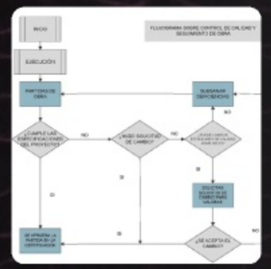
Data Sources: Integración nativa con google-cloud-bigquery para consultar las tablas trips y stations.

Arquitectura Agéntica en Acción

La Inteligencia detrás del Grafo

Architect Node: Diseñado para evitar alucinaciones usando un contexto de esquema robusto (MIBICI_SCHEMA_CONTEXT).

Conditional Edges: Si el executor_node detecta que el resultado es None o hay un error de SQL, el flujo regresa automáticamente al architect para una "Reflexión".



Demo: Interrogando a la Ciudad

- Dinámica:
  - Se solicita a la audiencia una pregunta compleja (ej: "¿Cuántas mujeres usaron MiBici cerca de la Minerva emn 2024?").
- Observabilidad en Vivo:
  - La terminal mostrará en tiempo real cómo el agente decide usar LIKE '%Vallarta%' para encontrar la Minerva y cómo calcula la duración del viaje.
- Identidad Local:
  - El nodo storyteller procesa los datos finales para entregar una respuesta amable.

Conclusiones y Post-Mortem

A lo largo del proceso de construcción de agentes autónomos, se han obtenido valiosas lecciones sobre la implementación efectiva de IA en producción. Las experiencias vividas subrayan la importancia de la observabilidad y la gestión de errores para crear sistemas de IA confiables y funcionales.

El Impacto: IA que realmente 'chambea'

Operational AI: Pasamos de experimentos de chat a herramientas que resuelven cuellos de botella reales.

Reliability: Los ciclos de autocorrección permiten que el sistema sea confiable para ambientes de producción.

Lessons Learned: Las alucinaciones son reales; la observabilidad y los logs del grafo son tus mejores amigos.

Fixing the Bugs: Cómo manejamos los errores de API y las inconsistencias de datos en GCP.

Post-Mortem: Lo que aprendí buildeando agentes

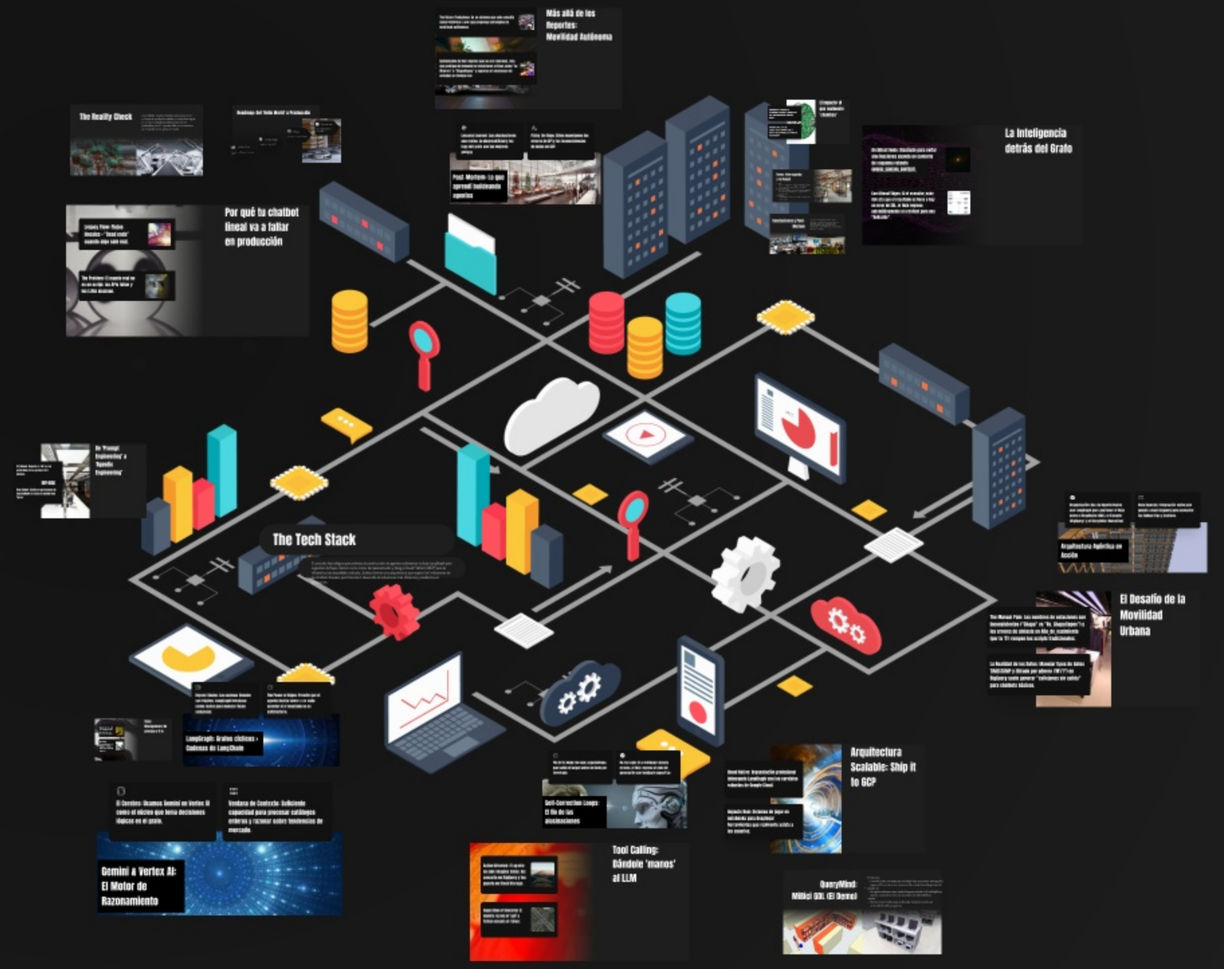
Más allá de los Reportes: Movilidad Autónoma

The Vision: Evolucionar de un sistema que solo consulta datos históricos a uno que proponga estrategias de movilidad autónomas.

Optimización de Red: Agentes que no solo informen, sino que predigan la demanda en estaciones críticas como "La Minerva" o "Chapultepec" y sugieran el rebalanceo de unidades en tiempo real.

Más allá del Chat: Construyendo Agentes Autónomos con Python y Google Cloud

Presentador: Jose Muñoz



Take this with you.
Revisit anytime.

Missed something? Want to explore further?
Scan or click below to open this presentation.
Anytime, anywhere.

View presentation