



## **Desafío Abierto de Agente de IA: Asistente para Incubadoras**

08/08/2025

### **Estudiantes**

Diego Handalian

Juan Darracq

Manuel Mongelós

Alejandro Piccardo

Felipe Villaronga

|                                     |           |
|-------------------------------------|-----------|
| Introducción                        | 3         |
| Arquitectura de la Solución         | 4         |
| Módulos y Alternativas Consideradas | 6         |
| Responsible AI                      | 8         |
| Gemini Vision & Multimodal          | 8         |
| Pruebas y Resultados                | 9         |
| Conclusiones y Futuras Mejoras      | 12        |
| Referencias                         | <b>13</b> |

## Introducción

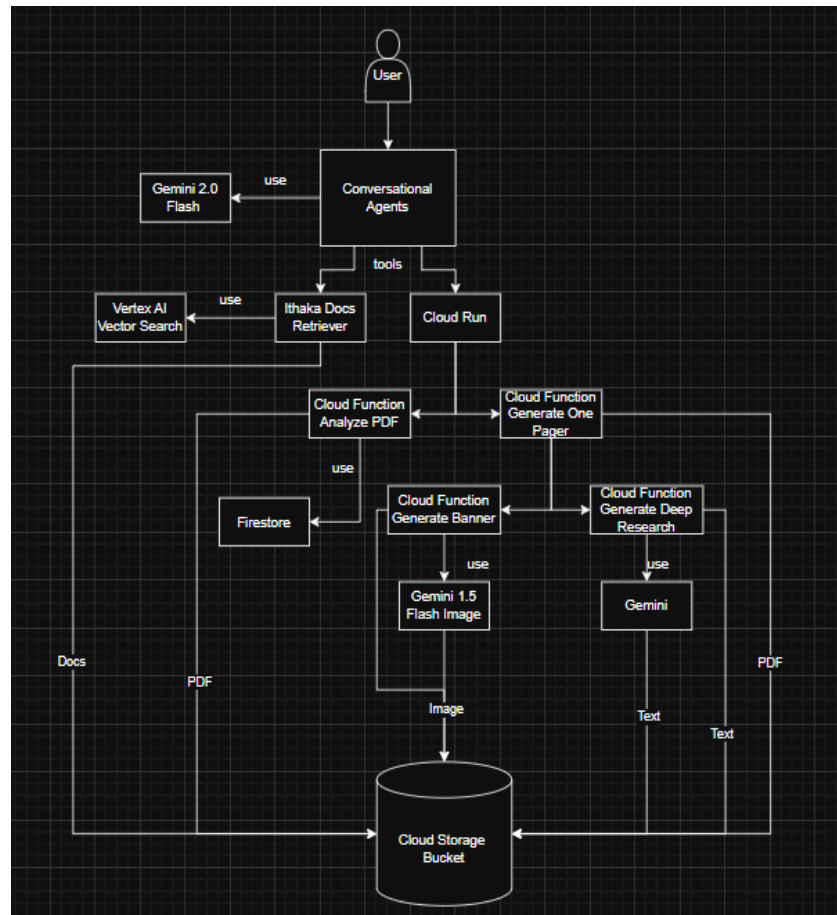
Ithaka es el Centro de Emprendimiento e Innovación de la Universidad Católica del Uruguay, busca fomentar la creación y el crecimiento de proyectos emprendedores a través de programas de incubación y mentorías. Cada año, decenas de emprendedores intentan postular a Ithaka sin tener claridad sobre el proceso, los criterios o cómo redactar adecuadamente sus respuestas.

Ante esta problemática, el objetivo de este reto fue desarrollar un agente conversacional que acompañe y oriente al postulante durante todo el proceso de inscripción. Este agente debe ser capaz de responder preguntas frecuentes, ofrecer recomendaciones personalizadas y guiar al usuario en la preparación de su postulación, minimizando errores y aumentando la calidad de las propuestas recibidas.

Nuestra solución permite automatizar la orientación inicial que hoy en día se brinda de forma manual, asegurando una respuesta consistente, precisa y disponible las 24 horas. Además de su funcionalidad principal, el asistente es capaz de comprender consultas en múltiples idiomas y de interpretar el contexto a lo largo de conversaciones prolongadas. Esto no solo contribuye a una experiencia de usuario más fluida y personalizada, sino que también minimiza los errores en la postulación y, en última instancia, aumenta la calidad y el potencial de los proyectos recibidos por Ithaka.

## Arquitectura de la Solución

En el siguiente diagrama se muestran los distintos componentes de la arquitectura y los flujos de datos de la solución construida en Google Cloud Platform. La misma utiliza servicios serverless y modelos de IA avanzados para gestionar la interacción con el usuario y las tareas complejas de forma eficiente.



### Componentes Principales

- **User:** El punto de entrada al sistema, que interactúa con el asistente a través de una interfaz de chat.
- **Conversational Agents:** Este es el cerebro del sistema, un agente de **Vertex AI Conversation** que orquesta toda la lógica. Recibe la entrada del usuario, detecta la intención y decide qué "herramienta" o acción ejecutar.
- **Gemini 2.0 Flash:** El modelo de lenguaje base utilizado por el agente para la lógica conversacional, la generación de respuestas y el análisis de texto.
- **Herramientas (tools):** Son las funcionalidades que el agente puede invocar. Se dividen en dos categorías principales:

- **Ithaka Docs Retriever:** El sistema RAG que utiliza **Vertex AI Vector Search** para buscar información relevante en los documentos de Ithaka, almacenados en un **Cloud Storage Bucket**.
- **Cloud Functions:** Un conjunto de funciones serverless que gestionan las tareas asíncronas y computacionalmente intensivas.

### Ejemplo de Flujo

#### **Flujo de Consulta (RAG):**

1. Un usuario hace una pregunta sobre los documentos de Ithaka.
2. El agente conversacional invoca el **Ithaka Docs Retriever**.
3. El Retriever consulta **Vertex AI Vector Search** para encontrar fragmentos de texto relevantes en el bucket de Cloud Storage.
4. El agente utiliza estos fragmentos como contexto para generar una respuesta precisa y contextualizada con **Gemini**.

### Prompting

Se utilizó una combinación de técnicas de ingeniería de prompts para ejecutar distintas tareas que debe cumplir el agente conversacional.

- **Zero-Shot Prompting:** Permite que el modelo genere una respuesta o realice una tarea sin recibir ejemplos específicos en el prompt. Es ideal para tareas sencillas que no requieren contexto previo.
  - **Uso en el proyecto:** El agente utiliza Zero-Shot para tareas como el análisis de tono del mensaje de un usuario ("¿Este usuario parece frustrado o satisfecho?"). Simplemente se le pide al modelo que realice una clasificación sin darle ejemplos de lo que significa "frustrado" o "satisfecho" en este contexto. El modelo aprovecha su conocimiento preexistente para dar la respuesta.
- **Few-Shot Prompting:** Se utiliza cuando la tarea es más específica y requiere un formato de salida particular. Se proporcionan algunos ejemplos de la tarea dentro del prompt para guiar al modelo.
  - **Uso en el proyecto:** Es muy útil para tareas como la generación de un One Pager. El prompt no solo le da la información clave, sino que también le muestra ejemplos de cómo debe estar estructurado el resultado final (p. ej., "Problema:..., Solución:..., Mercado:..."). Esto asegura que el formato de salida sea consistente con los estándares de Ithaka.

- **Chaining (Encadenamiento de Prompts):** Esta es la técnica más avanzada y se utiliza para dividir una tarea compleja en pasos más pequeños y manejables, donde la salida de un paso se convierte en la entrada del siguiente.
  - **Uso en el proyecto:** El flujo de procesamiento de un PDF es un excelente ejemplo de encadenamiento de prompts:
    - **Paso 1 (OCR):** Se utiliza un prompt para que **Gemini Vision** extraiga todo el texto y los elementos visuales clave de un documento.
    - **Paso 2 (Resumen):** La salida del primer paso se toma como entrada para un segundo prompt, que le pide al modelo que resuma el texto en puntos clave.
    - **Paso 3 (Extracción de Información):** Finalmente, la salida del resumen se utiliza en un tercer prompt que le pide al modelo que responda a preguntas específicas del usuario ("¿Cuál es la propuesta de valor del proyecto?").

## Módulos y Alternativas Consideradas

### Módulo del Agente Conversacional

**Conversational Agents (Vertex AI Conversation):** Esta plataforma es el núcleo del agente conversacional. Proporciona una estructura completa y gestionada para la detección de intenciones, la gestión del diálogo y la integración de herramientas a través de webhooks. Esto nos permite centrarnos en la lógica de negocio sin tener que construir la capa conversacional desde cero.

Entre las cosas que nos proveyó Conversational Agents, se encuentra la posibilidad de recibir audio de forma nativa y que el agente pueda responder a este. Esto ya viene integrado en la interfaz provista, pero de otra forma, también se podría lograr subiendo archivos mp4 a un bucket y el agente puede tomarlos desde ahí para responder.

Otro de los atractivos es su forma de presentar el área de *prompting*, donde el agente no recibe un solo prompt, sino que la interfaz ya nos lleva a las buenas prácticas de prompting que vimos en el curso. Una de las secciones para promptear al agente se trata de su contexto, su rol y como será su objetivo. La segunda área es para las instrucciones, cómo reaccionar frente a distintas interacciones, con un formato dado para el uso de sus herramientas. A esto se le suma la posibilidad de sumar ejemplos, lo cual termina mostrándonos un panorama donde la interfaz invita al usuario a usar técnicas de brindar contexto, instrucciones, condiciones con ejemplos, facilitando el lograr un buen resultado para el usuario.

A través del prompt y la posibilidad de instrucciones para reaccionar a distintos pedidos, pudimos fácilmente implementar que el agente hablara principalmente en español pero tradujera a cualquier idioma si fuese pedido por el usuario.

#### **Alternativa:**

- **Agente con LangChain:** Un enfoque común y muy potente en la industria, que permite una gran flexibilidad en la creación de cadenas de prompts y la conexión de herramientas. La principal desventaja es la necesidad de un mayor esfuerzo de desarrollo y gestión de la infraestructura (por ejemplo, alojarlo en una VM o Cloud Function) en comparación con una plataforma gestionada como Vertex AI Conversation, que ya incluye el frontend y el orquestador.

#### Módulo de Optimización de Modelos

**Uso de modelos gestionados de Vertex AI:** Este enfoque se basa en la fiabilidad, facilidad de uso y escalabilidad. No requiere fine tuning, lo que reduce la complejidad y los costos iniciales de entrenamiento.

#### **Alternativa:**

- **Fine Tuning de modelos más pequeños:** Una estrategia para optimizar costos y latencia a largo plazo. Se podría tomar un modelo más pequeño y entrenarlo con los datos específicos de Ithaka para mejorar su rendimiento en tareas muy concretas. Esto requiere una inversión inicial en datos y computación, pero podría resultar en una solución más eficiente en un entorno de producción con un alto volumen de uso.

#### Arquitectura

**Patrón Sincrónico:** Es el patrón de arquitectura utilizado, donde una solicitud del agente invoca una función específica en Cloud Run y espera una respuesta, es directo y fácil de implementar para el desarrollo inicial.

#### **Alternativas:**

En busca de una mayor escalabilidad y robustez se buscaron otras alternativas que luego se descartaron principalmente por su complejidad.

- **Arquitectura "Agente de Agentes":** En lugar de un agente que llama a herramientas simples, se establece un agente enrutador/controlador principal que se encarga de dirigir la conversación al agente secundario más apropiado para la tarea. Sin embargo el mismo tiene una mayor complejidad y latencia, y depende de la capacidad del agente enrutador.

- **Patrón de Llamadas Múltiples (MCP):** En este patrón, el agente puede invocar a varias "herramientas" pequeñas y especializadas de forma independiente y en paralelo. Estas herramientas se alojarían como endpoints en Cloud Functions o en un servicio de Cloud Run. Esto ofrece ventajas como la modularidad, escalabilidad y tolerancia a fallos. Como desventaja tiene una mayor complejidad, aumento de latencia y mayor costo operacional.



## Responsible AI

Responsible AI busca que la solución sea segura, ética, inclusiva y transparente. Estos principios son esenciales para nuestro asistente conversacional Ithaka debido a que el público objetivo son estudiantes y emprendedores con distintos niveles de experiencia, conocimientos y necesidades.

El modelo va a ser la puerta de nuevos emprendedores, por lo que su comportamiento influirá en la percepción inicial que estos tengan de Ithaka. El sistema debe dar información clara, útil y confiable para fortalecer la confianza y motivar a más personas a postularse. Por el contrario, respuestas ambiguas o incorrectas podrían generar frustración y desincentivar la participación.

En segundo lugar, la equidad es crucial. El asistente debe orientar de la misma forma a todos los postulantes, sin favorecer a los que ya poseen más conocimientos técnicos o experiencia previa. Esto implica mantener un lenguaje accesible, explicar cuando sea necesario y ofrecer ejemplos prácticos.

En los cursos Intro to Generative AI, Generative AI for Developers y Gemini for Google Cloud, se enfatizó que la IA responsable no es opcional, sino que un componente central del diseño y despliegue de cualquier agente conversacional.

Partiendo de esto, en el uso de Conversational Agents y las herramientas de RAG de Google, encontramos un gran valor agregado en que al agente usar las tools, te muestra qué obtuvieron de estas. Sumado a como funcionan las herramientas de RAG, que te muestran lo que obtuvieron como similar a la búsqueda del texto, podemos ver un flujo donde se podrían detectar alucinaciones si ocurrieran, por lo transparente del proceso. Para una organización institucional, creemos esto puede tener un gran valor y seguridad, dándoles mucho más control sobre lo que pasa con su agente.

## Gemini Vision & Multimodal

Para procesar documentos se integró el modelo Gemini 2.0 Flash con su versión multimodal, permitiendo al chatbot procesar archivos como PDFs o imágenes que el usuario sube durante la conversación. Esto se implementó mediante una función desplegada en Google Cloud Functions, que toma el archivo desde un bucket de Cloud Storage y lo analiza con Gemini a partir de un prompt contextual.

Por lo que, por ejemplo, un usuario puede subir un documento con su propuesta de emprendimiento y el asistente sería capaz de leerlo, resumirlo y responder preguntas específicas del PDF.

También se utilizó Gemini 1.5 Image para generar banners personalizados a partir de un prompt de un usuario, subiendo la imagen también al bucket de Cloud Storage. Además, se utilizó Custom

Search API para realizar una búsqueda contextual, enviárselo a un prompt de Gemini 2.0 Flash y generar un One Pager incluye problema, solución, mercado y competencia

Esto fue posible en parte gracias a los cursos vistos, donde se explica cómo trabajar con distintos tipos de inputs

# Pruebas y Resultados

## Consultas al Agente Conversacional

Algunas de las pruebas que se le hicieron al agente fue realizar preguntas que haría un emprendedor, cómo pedir ayuda sobre un emprendimiento, validar ideas o generar un One Pager.



Quiero ayuda con mi emprendimiento



¡Hola! Como tu asistente de IA de Ithaka, estoy aquí para ayudarte con todas las preguntas relacionadas con el proceso de postulación a la incubadora.

Veo que quieres ayuda con tu emprendimiento. Puedo ofrecerte generar un "one pager" para tu idea. Esto incluiría un estudio de mercado y un banner. Solo necesito que me des una oración corta que describa tu emprendimiento y lo que quieres lograr.

¿Cómo te gustaría que te ayude hoy?

Latency: 0.810s



Quiero lograr una app que te ayude a saber que cocinar con lo que tenes en la heladera, usando IA



¡Excelente idea! Una aplicación así sería muy útil.

Para generar el "one pager" y el banner de tu aplicación, necesito una descripción un poco más detallada de la idea. ¿Podrías darme una oración corta que resuma lo que quieres hacer con esta aplicación? Por ejemplo, "Una aplicación móvil que sugiere recetas personalizadas usando IA, basándose en los ingredientes disponibles en tu nevera, para reducir el desperdicio de alimentos y facilitar la planificación de comidas."

Latency: 0.908s



Eso mismo!



¡Perfecto! Con esa descripción, puedo generar el one-pager y el banner para tu aplicación.



onepager-generation generateBanner  
Tool Action

1  
Input parameters

1  
Output parameters



¡Excelente! He generado el one-pager y el banner para tu aplicación. Puedes acceder a ellos a través de este enlace:

[https://storage.googleapis.com/ithaka-one-pagers/one-pagers/title\\_a836f29f-d435-4d20-b806-753d1c4e4980.pdf?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=cs-invoker-onepager%40valiant-complex-467519-d3.iam.gserviceaccount.com%2F20250808%2Fauto%2Fstorage%2Fgoog4\\_request&X-Goog-Date=20250808T025538Z&X-Goog-Expires=600&X-Goog-SignedHeaders=host&X-Goog-Signature=7db2a1f320dae8f2f1afc3cc91fabea810ef7e1b424a11a1e4c704f2a47fb678e0f0a8c5c211d3b9690bc7e9794ace56ef937ef398bb2ba21e0ddd3533e7624283a5e61113a0062b8cd6bafb5d9568c0fad1ee4a88ec4d231689d9f917b4eae82a10932b57c5dfa58dfe92bc094c95206c6d4fc0337d85f0f64fb98c34bf694487e13dea2d1361e7f91fd5112fa9892a6921990ef](https://storage.googleapis.com/ithaka-one-pagers/one-pagers/title_a836f29f-d435-4d20-b806-753d1c4e4980.pdf?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=cs-invoker-onepager%40valiant-complex-467519-d3.iam.gserviceaccount.com%2F20250808%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20250808T025538Z&X-Goog-Expires=600&X-Goog-SignedHeaders=host&X-Goog-Signature=7db2a1f320dae8f2f1afc3cc91fabea810ef7e1b424a11a1e4c704f2a47fb678e0f0a8c5c211d3b9690bc7e9794ace56ef937ef398bb2ba21e0ddd3533e7624283a5e61113a0062b8cd6bafb5d9568c0fad1ee4a88ec4d231689d9f917b4eae82a10932b57c5dfa58dfe92bc094c95206c6d4fc0337d85f0f64fb98c34bf694487e13dea2d1361e7f91fd5112fa9892a6921990ef)

¿Hay algo más en lo que pueda ayudarte hoy?

## Generación de Banner

Dado un prompt explicando en pocas palabras lo que se quiere, por ejemplo: “Pato-Pato-IA, haciendo tu patito de hule como lo quieras”.



Otros ejemplos de los banners generados:



## Generación de One Pagers

Dado el prompt: “Una aplicación móvil que usa IA para escanear el contenido de tu refrigerador y te sugiere recetas saludables para la semana, ayudando a reducir el desperdicio de comida.” Se genera



One-Pager: **NutriGenius: Tu Chef Personal con IA**

1. El Problema Las personas se enfrentan a desafíos significativos al intentar mantener una dieta saludable: **Sobrecarga de Información:** La vasta cantidad de recetas online puede ser abrumadora y difícil de filtrar. **Falta de Tiempo:** Planificar comidas y buscar recetas requiere tiempo y esfuerzo que muchos no tienen. **Dificultad para Adaptar Recetas:** Ajustar recetas para restricciones dietéticas, alergias o preferencias personales es complicado. **Inspiración Culinaria Limitada:** Es fácil caer en la rutina y comer las mismas comidas saludables repetidamente, llevando a la monotonía y al abandono de la dieta. **Desperdicio de Alimentos:** La falta de planificación conduce a la compra excesiva de ingredientes y al desperdicio de alimentos.

2. La Solución NutriGenius es una aplicación móvil impulsada por IA que simplifica la alimentación saludable al:

- Organizar Recetas Inteligente:** Ofrece una amplia base de datos de recetas saludables, curadas y organizadas según criterios como tipo de dieta (vegetariana, vegana, keto, etc.), alergias, preferencias de sabor, y tiempo de preparación.
- Recomendaciones Personalizadas:** Utiliza algoritmos de IA para recomendar recetas basadas en el perfil del usuario, incluyendo restricciones dietéticas, objetivos de salud, y preferencias alimentarias.
- Planificación de Comidas Automatizada:** Genera planes de comida semanales personalizados, considerando las necesidades nutricionales y el estilo de vida del usuario.
- Lista de Compras Inteligente:** Crea listas de compras optimizadas basadas en el plan de comidas, minimizando el desperdicio de alimentos.
- Integración con Dispositivos Wearables (opcional):** Incorpora datos de actividad física y salud del usuario para ajustar las recomendaciones de recetas y planes de comidas.

3. Mercado Potencial NutriGenius se dirige a los siguientes segmentos de mercado:

- Individuos Conscientes de la Salud:** Personas que buscan mejorar su salud y bienestar a través de una alimentación equilibrada.
- Personas con Restricciones Dietéticas:** Individuos que necesitan adaptar su dieta debido a alergias, intolerancias alimentarias o condiciones médicas como diabetes o celiaquía.
- Profesionales Ocupados:** Personas que tienen poco tiempo para planificar comidas y buscar recetas saludables.
- Familias:** Padres y madres que buscan opciones saludables y fáciles de preparar para sus hijos.
- Entusiastas del Fitness:** Individuos que buscan optimizar su dieta para mejorar su rendimiento.

## Conclusiones y Futuras Mejoras

Este proyecto ha demostrado el potencial de un agente conversacional para transformar la interacción entre una incubadora y sus postulantes. La arquitectura serverless de Google Cloud, junto con las capacidades multimodales de Gemini, permitió construir una solución robusta y escalable que no solo responde preguntas, sino que también genera contenido y procesa documentos de manera inteligente. Se logró un sistema que no solo responde preguntas frecuentes, sino que también gestiona tareas complejas como el análisis de documentos y la generación de contenido.

Como futuras mejoras, se podrían considerar:

1. **Pruebas de Prompts:** Realizar más y distintas pruebas de prompt, como por ejemplo un enfoque de pruebas A/B, en busca de analizar cómo se obtienen mejores resultados e ir recolectando feedback a través de otro LLM.
2. **Inclusión de Coding Agent:** Este módulo sería incluido para interpretar la descripción de una idea de negocio proporcionada por el usuario y, utilizando modelos de IA especializados en la generación de código, crear un prototipo básico y funcional, ofreciendo a los emprendedores una visualización temprana de sus ideas a través de mini-demos funcionales con datos de ejemplo.
3. **Integración con un CRM:** Se podría integrar el agente con un sistema de gestión de relaciones con el cliente (CRM) o una base de datos de postulantes. Esto permitiría al agente registrar automáticamente la información capturada durante la conversación (como el resumen de un plan de negocio o los datos de contacto) y monitorear el progreso del usuario en el proceso de postulación.
4. **Generación de Diagramas:** Expandir la funcionalidad multimodal para que el agente pueda generar diagramas de arquitectura o de flujo de datos a partir de descripciones de texto. Esto proveería un gran valor a los emprendedores que están definiendo sus modelos de negocio o arquitectura tecnológica.
5. **Sistema de Recomendación Avanzado:** Mejorar el módulo de recomendaciones para que analice el perfil del emprendedor (sector, etapa, necesidades) para ofrecer recomendaciones más precisas sobre fondos de inversión o mentores específicos dentro de la red de Ithaka.

En definitiva, el proyecto cumple con su objetivo de reducir barreras para los emprendedores y optimizar el tiempo del equipo de Ithaka. Mientras que algunas de las alternativas y puntos anteriormente mencionados son posibles mejoras que se pueden implementar en una futura iteración de desarrollo para poder aumentar aún más la productividad del Centro de Emprendimiento e Innovación de la Universidad Católica del Uruguay.

## Referencias

- Google Cloud Skills Boost. (s.f.-a). *Beginner: Introduction to Generative AI Learning Path*. Google. Recuperado el 8 de agosto de 2025, de <https://www.cloudskillsboost.google/paths/118>
- Google Cloud Skills Boost. (s.f.-b). *Advanced: Generative AI for Developers Learning Path*. Google. Recuperado el 8 de agosto de 2025, de <https://www.cloudskillsboost.google/paths/183>
- Google Cloud Skills Boost. (s.f.-c). *Gemini for Google Cloud Learning Path*. Google. Recuperado el 8 de agosto de 2025, de <https://www.cloudskillsboost.google/paths/236>
- Centro Ithaka – Universidad Católica del Uruguay (UCU). (s.f.). *Centro de emprendimiento e innovación*. UCU. Recuperado el 8 de agosto de 2025, de <https://www.ucu.edu.uy/categoria/Ithaka-340>