Reporte Proyecto :
ChatBot LCC con AWS

Autores: Balderrama Dominguez Gael, Fimbres Delgado Silvia Mariana, Montoya Valencia Lenika Elizab

Diciembre de 2025

Resumen

Este proyecto consistió en el desarrollo de un asistente inteligente para la Licenciatura en Ciencias de la Computación (LCC) de la Universidad de Sonora, utilizando servicios administrados de AWS. El objetivo principal fue crear un chatbot capaz de responder preguntas sobre reglamentos, planes de estudio y documentos oficiales. La solución emplea un sistema **RAG** (Retrieval-Augmented Generation) conectado a un modelo de lenguaje desplegado en Amazon SageMaker, todo accesible mediante una interfaz web estática. Esta propuesta surgió para apoyar a la comunidad LCC, facilitando la consulta de información oficial y segura.

Logros Clave

- Implementación de un **Chatbot Inteligente** capaz de procesar lenguaje natural.
- Desarrollo de una **Arquitectura RAG** robusta para basar las respuestas en documentos reales (desde S3).
- Despliegue de un **Motor LLM en SageMaker**.
- Uso de un **Backend Serverless** (Lambda y API Gateway) para eficiencia y escalabilidad.

Arquitectura General y Componentes

La solución opera con una interfaz web estática desplegada en AWS Amplify, conectada a un backend serverless que coordina las interacciones con el motor LLM y la base de datos de documentos RAG.

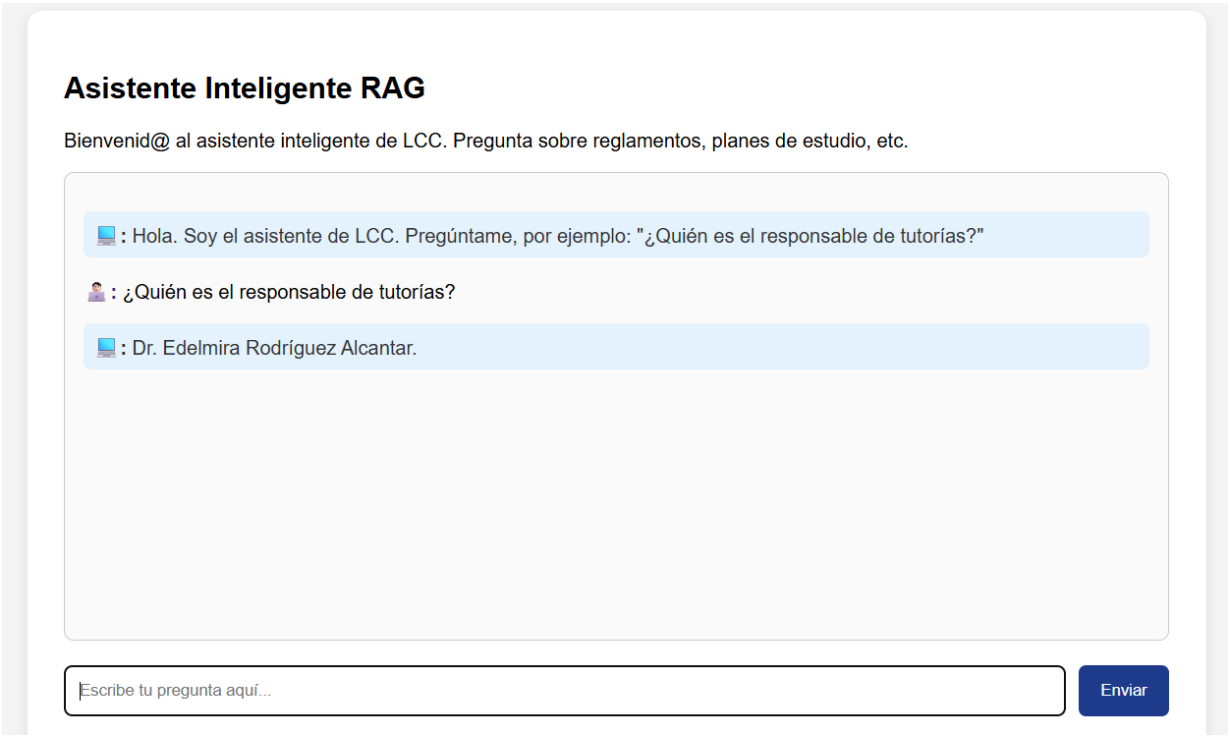

Figure 1: Esquema conceptual del ChatBot y Arquitectura.

Tecnologías AWS Utilizadas: Amplify, Cognito, S3, Lambda, API Gateway y SageMaker.

Figure 2: Stack de Servicios AWS.

1 Detalles del Sistema y Funcionalidades

1.1 Flujo de Trabajo y Backend Serverless

El flujo central de la aplicación es gestionado por AWS Lambda.

1. **Motor de RAG y LLM:** La arquitectura RAG recupera y procesa los documentos (almacenados en S3) para generar una respuesta basada en datos reales, conectándose al modelo de lenguaje en SageMaker.
2. **Backend Serverless:** AWS Lambda recibe la solicitud del usuario, ejecuta la lógica del RAG, se conecta al endpoint del LLM en SageMaker y gestiona la respuesta JSON al frontend.

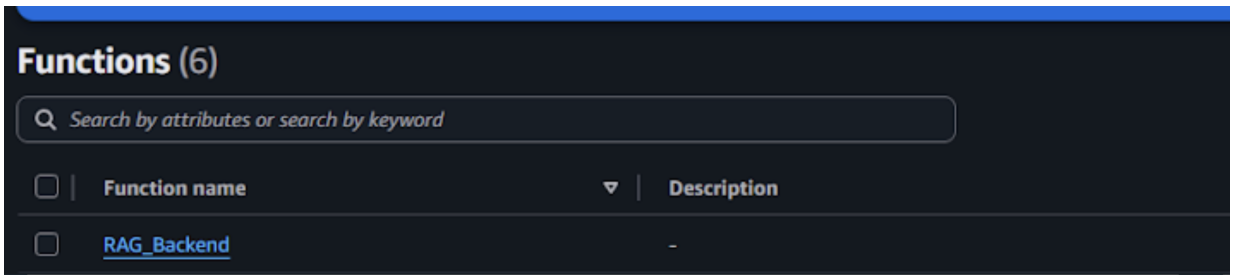

Figure 3: Flujo de la Función Lambda.

3. **Endpoint de Modelo:** El modelo de lenguaje está alojado en un endpoint de Amazon SageMaker, asegurando un acceso dedicado y escalable para la inferencia.

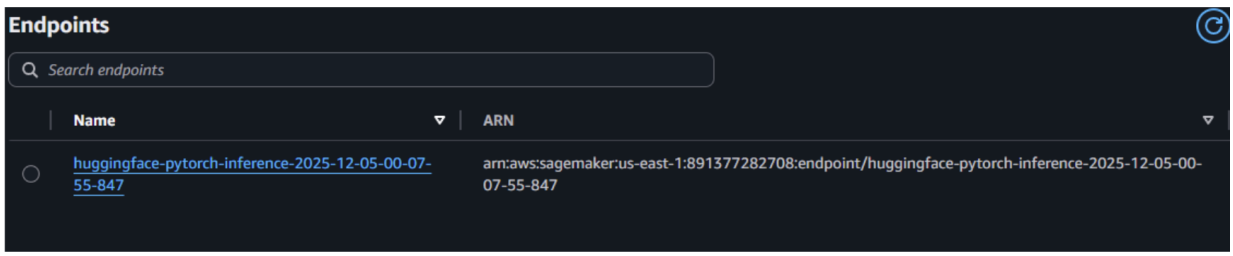

Figure 4: Endpoint del Motor LLM en SageMaker.

1.2 Exposición y Seguridad

5. **API Gateway:** Actúa como la interfaz segura y expuesta del backend, manejando la seguridad y siendo el *trigger* directo para la función Lambda.

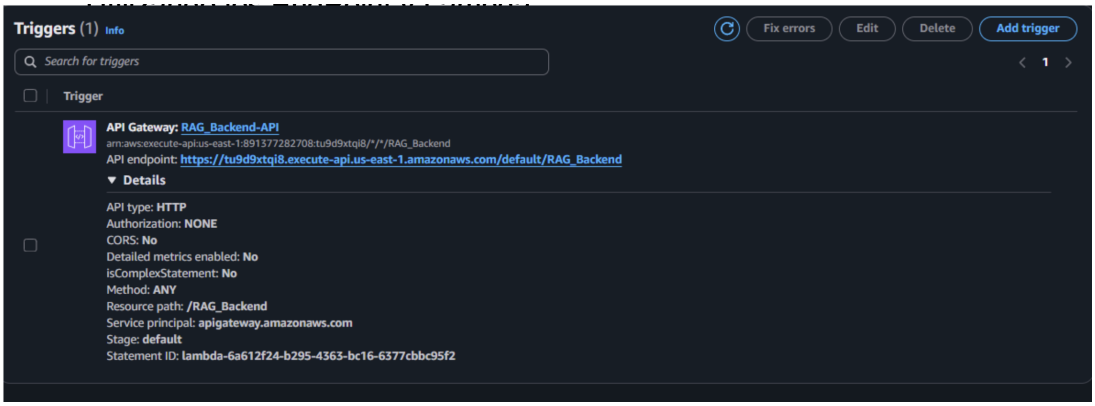

Figure 5: Configuración del Trigger en API Gateway.

6. **Hosting y Despliegue (Amplify):** La página web estática se despliega mediante AWS Amplify, facilitando la integración directa y la implementación por ZIP upload.
7. **Seguridad y Gestión de Usuarios (Cognito):** La seguridad se reforzó con la gestión de usuarios vía Amazon Cognito, junto con la aplicación de roles y un firewall básico de Amplify.

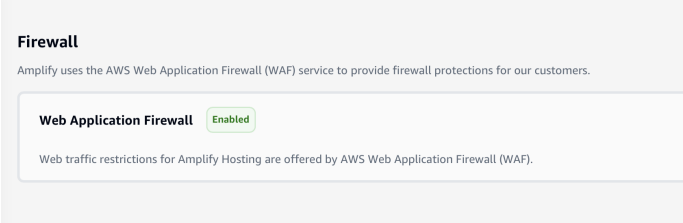

Figure 6: Configuración del Firewall y Roles de AWS Amplify.

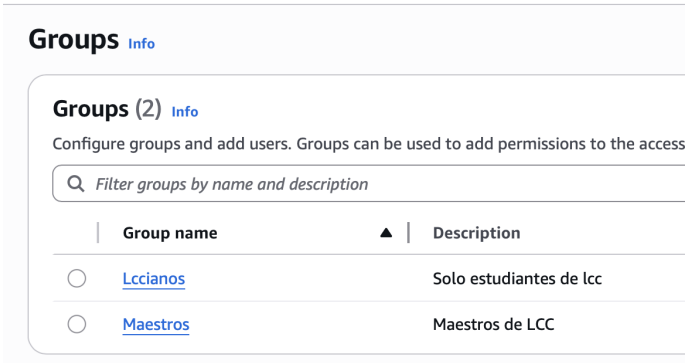

Figure 7: Gestión de Usuarios y Grupos en Cognito.