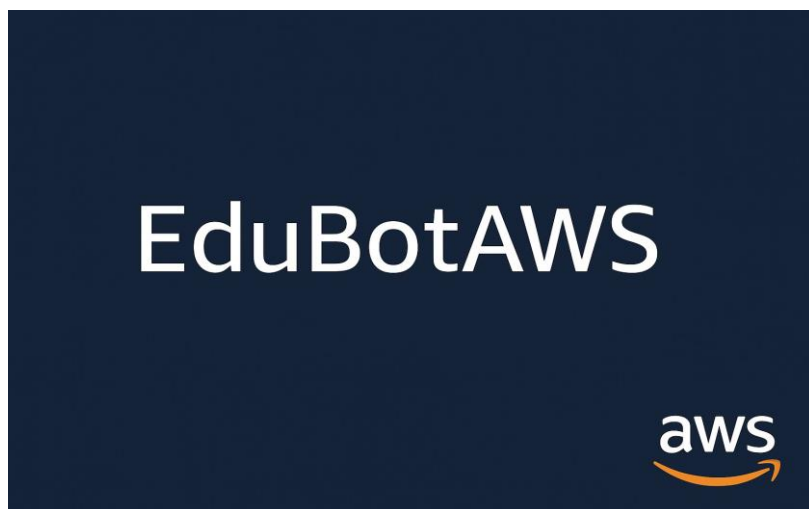


# EduBotAWS

**Asistente Virtual Inteligente para registro de  
Solicitudes de Docentes para niños de edades  
tempranas**



Documento Técnico - Arquitectura y Funcionamiento  
Versión 1.0 | Octubre 2025

Autor: Fredy Alexander Meyer

## Executive Summary

**EduBotAWS** es un asistente virtual inteligente diseñado para automatizar el proceso de registro de solicitudes docentes en educación temprana. Utilizando la suite de servicios en la nube de Amazon Web Services, el sistema combina inteligencia conversacional, procesamiento de lenguaje natural y almacenamiento sin servidor para ofrecer una experiencia eficiente, escalable y humana.

## Project Objective

El objetivo principal del proyecto es permitir que padres o acudientes puedan registrar solicitudes docentes de forma automática mediante un chatbot web. El sistema guía al usuario en un flujo conversacional estructurado, valida las respuestas y almacena los datos en una base de datos segura (DynamoDB). Además, incluye retroalimentación auditiva con Amazon Polly, ofreciendo accesibilidad y una experiencia más empática.

## Architecture Overview

La arquitectura de EduBotAWS se basa completamente en componentes serverless de AWS, asegurando alta disponibilidad y bajo costo operativo. Los principales servicios utilizados son:

- **Amazon Lex V2:** Motor de procesamiento de lenguaje natural (NLU) que identifica intenciones del usuario y gestiona los flujos conversacionales.
- **AWS Lambda:** Función sin servidor que implementa la lógica del negocio, validaciones y conexión con DynamoDB.
- **Amazon DynamoDB:** Base de datos NoSQL para almacenamiento seguro y escalable de solicitudes.
- **Amazon Polly:** Servicio de conversión texto-voz que permite respuestas habladas del bot.
- **Amazon S3:** Almacén y host del frontend web (HTML y JS) para interacción directa con los usuarios.

## Conversation Flow

El flujo conversacional sigue un enfoque guiado: saludo inicial, captura de datos del estudiante, validaciones dinámicas, confirmación de la información y registro exitoso. El bot valida edad (2–12 años), formato de celular (10 dígitos), y límites de días/horas semanales. Finalmente, genera un identificador único de solicitud y lo muestra en pantalla.

## Results and Learnings

El sistema demostró alta precisión en las validaciones de entrada y una experiencia fluida para los usuarios. Su arquitectura serverless garantiza escalabilidad, seguridad y bajo mantenimiento. Entre los aprendizajes más importantes se destaca la importancia de

manejar intents secundarios (como saludos) y de mejorar continuamente los utterances para una comprensión más natural del lenguaje.

## Website URL (Amazon S3)

URL del sitio web S3	<a href="http://solicituddocente.s3-website-us-east-1.amazonaws.com/">http://solicituddocente.s3-website-us-east-1.amazonaws.com/</a>
----------------------	---

## GitHub Repository

EduBotAWS	<a href="https://github.com/alexmeyerm/EduBotAWS">https://github.com/alexmeyerm/EduBotAWS</a>
-----------	---

## Intents and Utterances (samples)

Intent	Utterances (muestras)
SolicitudServicio	Quiero registrar una solicitud de docente
FallbackIntent	Cubre entradas fuera de dominio, redirige con ayuda/ejemplos

## SolicitudServicio Intent

El Intent **SolicitudServicio** se activa cuando un usuario expresa su deseo de registrar una solicitud de docente para su hijo o hija.

Este intent es el responsable de:

1. Recolectar toda la información necesaria (slots) como nombre, número celular, nombre del hijo, edad, escolaridad, días, horas y jornada.
2. Validar cada uno de esos datos mediante la función Lambda conectada (Lambda\_function.py).
3. Confirmar la información antes de guardar el registro en DynamoDB.

En resumen, este intent simula una conversación humana guiada, donde el bot acompaña al usuario paso a paso hasta completar la solicitud.

## Ejemplos de *Utterances*

**“Mi nombre es {NombreContacto} requiero tus servicios de docente.”**

**Función:** activa el intent cuando el usuario se presenta y solicita un docente.

**Slots invocados:** {NombreContacto}.

**Comportamiento:** el bot detecta la intención de registrar una solicitud y continúa pidiendo el número de celular.

**“Te contacto para solicitar servicio de docente para mi hijo {NombreHijo}.”**

**Función:** inicia el flujo de registro cuando el usuario menciona explícitamente el motivo y el nombre del hijo.

**Slots invocados:** {NombreHijo}.

**Ventaja:** la frase ya contextualiza el servicio y facilita una respuesta personalizada (“Perfecto, ¿podrías indicarme tu nombre?”).

**“Mi hijo {NombreHijo} tiene {Edad} años y está en {Escolaridad}.”**

**Función:** proporciona múltiples datos en una sola frase.

**Slots invocados:** {NombreHijo}, {Edad}, {Escolaridad}.

**Uso:** el bot extrae automáticamente estos valores, reduciendo pasos y mejorando la experiencia de usuario.

**“Deseo {DiasSolicitados} días a la semana en la {JornadaSolicitada}.”**

**Función:** el usuario indica la disponibilidad o preferencia de jornada.

**Slots invocados:** {DiasSolicitados}, {JornadaSolicitada}.

**Comportamiento:** el bot valida que los días estén entre 1 y 5, y la jornada sea “Mañana” o “Tarde”.

**“Quiero registrar una solicitud de docente para mi hijo {NombreHijo}.”**

**Función:** una de las frases más genéricas y naturales.

**Slots invocados:** {NombreHijo}.

**Rol:** permite al bot entender la intención sin requerir que el usuario diga “registrar” o “solicitud” literalmente, gracias al aprendizaje de Lex.

## SolicitudServicio Intent Slots

Nombre del slot	Tipo	Regla/Restricción
NombreContacto	AMAZON.FirstName	Texto libre (nombre)
NumCelular	AMAZON.PhoneNumber	Debe tener exactamente 10 dígitos (CO)
NombreHijo	AMAZON.FirstName	Texto libre (nombre)
Edad	AMAZON.AlphaNumeric	Rango 2–12 años (educación temprana)
Escolaridad	AlphaNumeric	Opciones cerradas: Preescolar/Primaria
DiasSolicitados	AlphaNumeric	Rango 1–5 (días por semana)
HorasDiariasSolicitadas	AlphaNumeric	Rango 1–4 (horas/día)
JornadaSolicitada	AlphaNumeric	Opciones: Mañana/Tarde (quick replies)
Confirmacion	AlphaNumeric	SI/NO vía botones; dispara cumplimiento

## Key validations (Lambda code hooks)

- **NumCelular:** Aceptar solo 10 dígitos; si no cumple, re-prompt con mensaje de error.
- **Edad:** 2-12; si no, mensaje educativo y reintento.
- **DiasSolicitados:** 1–5.
- **HorasDiariasSolicitadas:** 1–4.
- **Escolaridad/Jornada:** quick replies (Preescolar/Primaria; Mañana/Tarde).
- **Confirmación:** resumen completo + botones Sí/No.

## Amazon DynamoDB – Modelo de datos

Atributo	Tipo	Descripción
SolicitudPK	S	Partition Key. e.g., Contract#A-12AB34C
SolicitudSK	S	Sort Key. e.g., APPT#A-12AB34C
nombreContacto	S	Nombre del solicitante
celContacto	S	Celular validado 10 dígitos
nombreHijo	S	Nombre del menor
edadHijo	N	Edad 2–12
escolaridadHijo	S	Preescolar   Primaria
diasPorSemana	N	1–5
horasPorDia	N	1–4
jornadaPreferidaContrato	S	Mañana   Tarde

## Lambda Function

Define la función AWS Lambda que sirve como backend serverless para el bot EduBotAWS, implementado en Amazon Lex V2.

Su misión es controlar todo el flujo conversacional, validar la información ingresada, y registrar las solicitudes docentes en DynamoDB.

Contiene un conjunto de utilidades para manejar correctamente el formato de los eventos Lex:

- `get_slot()` y `set_slot_in_event()` → para leer y actualizar valores ingresados.
- `elicit_slot()` → para pedir al usuario un nuevo dato (slot) y mostrar mensajes o botones.
- `delegate()` → indica a Lex que continúe el flujo automáticamente.
- `close_ok()` → cierra el intent con un mensaje final.
- `rc()` → genera *response cards* (tarjetas visuales con botones) como “Sí / No” o “Mañana / Tarde”.

Define un diccionario PROMPT con los textos predefinidos para cada pregunta, como:

- “¿Cuál es tu número de celular?”
- “¿Cuántos años tiene tu hijo?”
- “¿En qué nivel de escolaridad está tu hijo?”
- Estos mensajes se utilizan dinámicamente dependiendo del progreso de la conversación.

Cada campo tiene una función específica de validación:

- `validate_NumCelular` → acepta solo 10 dígitos numéricos.
- `validate_Edad` → asegura que la edad esté entre 2 y 12 años.
- `validate_Escolaridad` → fuerza la selección entre “Preescolar” o “Primaria” con botones.
- `validate_DiasSolicitados` → restringe entre 1 y 5 días.
- `validate_HorasDiariasSolicitadas` → restringe entre 1 y 4 horas.
- `validate_JornadaSolicitada` → solo permite “Mañana” o “Tarde”.

Estos validadores no solo verifican la información, sino que responden con mensajes correctivos y vuelven a preguntar si el usuario ingresa datos inválidos.

El flujo principal está manejado por la función `lambda_handler(event, context)`:

1. Identifica el intent activo.
2. `DialogCodeHook` (fase de conversación):
  - Usa `ask_next_missing_or_validate()` para pedir los slots faltantes o corregir errores.
  - Cuando todos los datos están listos, llama a `pedir_confirmacion()` y muestra un resumen con botones Sí / No.
3. `FulfillmentCodeHook` (fase de registro):
  - Si el usuario confirma, ejecuta `guardar(event)` que crea un ID único y guarda el registro en DynamoDB.
  - Devuelve un mensaje de éxito con un botón “Registrar otra”.
4. Si el usuario responde “no”, el bot limpia la confirmación y permite corregir o reiniciar.

La función `guardar(event)`:

- Genera un ID único tipo A-XXXXXXX.
- Inserta un ítem en DynamoDB con los campos:
- `nombreContacto`, `celContacto`, `nombreHijo`, `edadHijo`,
- `escolaridadHijo`, `numDiasSemanaContrato`, `numHorasDiariasContrato`,
- `jornadaPreferidaContrato`
- Devuelve un mensaje de confirmación con el ID y un botón para crear otra solicitud.

Funciones Adicionales

- Detección del botón “Registrar otra”.
- Control del flujo de conversación aun cuando Lex pierde contexto.
- Mensajes de fallback (“¿En qué puedo ayudarte?”) ante casos inesperados.

## Amazon S3 – Frontend web estático

- Bucket **solicituddocente** con index.html y webapp.js.
- Se activó hosting estático y objetos public-read para servir el sitio.
- UI modo claro/oscuro, render de burbujas, botones manuales (Sí/No/Opciones), impresión del ID de registro al cierre y botón
- Registrar otra que limpia el estado.

## Conclusions

**EduBotAWS** representa una demostración tangible del poder transformador de la inteligencia artificial cuando se combina con arquitecturas serverless en la nube. Su desarrollo demuestra cómo los servicios de Amazon Web Services (AWS) —en particular Lex V2, Lambda, Polly, DynamoDB, Cognito y S3— pueden integrarse de forma armónica para construir soluciones conversacionales eficientes, seguras y escalables orientadas al sector educativo.

Desde una perspectiva técnica, el proyecto valida la capacidad de AWS para ofrecer un ecosistema robusto donde la automatización, la validación de datos y la experiencia conversacional se integran en un flujo coherente y sin fricciones. La arquitectura empleada no solo optimiza el procesamiento de solicitudes en tiempo real, sino que garantiza alta disponibilidad, bajo costo operativo y mínima intervención manual, atributos esenciales en entornos educativos de gran demanda.

En el ámbito funcional, EduBotAWS evidencia cómo la inteligencia artificial puede humanizar la interacción entre docentes, padres y sistemas administrativos, reduciendo tiempos, errores y barreras de comunicación. El asistente logra interpretar lenguaje natural, guiar al usuario mediante validaciones contextuales, y registrar información de forma estructurada y confiable.

Desde una perspectiva social, el proyecto subraya el impacto positivo de aplicar tecnologías conversacionales en contextos de educación temprana, donde la accesibilidad, la empatía y la inmediatez son factores críticos. EduBotAWS no solo automatiza un proceso, sino que acerca la tecnología al ciudadano, democratizando el acceso a servicios educativos mediante interfaces naturales y comprensibles.

Finalmente, este caso de uso consolida una base sólida para futuras ampliaciones con capacidades cognitivas más avanzadas como análisis de sentimiento con Amazon Comprehend, predicción de demanda con SageMaker, o traducción automática con Translate, orientando el camino hacia ecosistemas educativos inteligentes, inclusivos y sostenibles.

En síntesis, **EduBotAWS** no es solo un asistente virtual, sino un modelo de innovación educativa apoyado en IA y cloud computing, que demuestra cómo la tecnología puede potenciar la eficiencia operativa y, al mismo tiempo, generar valor humano y social.