

## **Etapas 1 - Proyecto final**

### **Integrantes:**

- Cadavid Calle Daniel
- Herrera Villa Mateo
- Gomez Jimenez Jose Manuel
- Restrepo Tangarife Camilo

### **1. Descripción del problema**

- En el sector empresarial, la atención al cliente juega un papel importante clave, al momento de percibir una marca o un servicio por parte de los clientes. Cada uno de los usuarios al momento de hacer una consulta, espera interacciones rápidas y personalizadas sobre los productos o servicios deseados. Muchas veces en el área de atención al cliente, los empleados presentan una sobrecarga de consultas básicas y repetitivas consumiendo tiempo valioso que podrían dedicar a resolver problemas más complejos dentro de la organización. En la era digital, donde los usuarios esperan respuestas rápidas y personalizadas, implementar un chatbot puede ser de gran ayuda para intervenir de manera eficaz la automatización de respuestas a preguntas frecuentes (FAQs), reduciendo el tiempo de espera y generando más tiempo a los empleados para concentrarse en tareas más importantes haciendo el equipo de más productivo y eficiente. Si una empresa no puede manejar grandes volúmenes de consultas de manera eficiente, los clientes pueden sentirse frustrados y dejar de usar los servicios.

Al implementar un chatbot, se puede responder de forma instantánea a cualquier hora del día y de la noche sin importar la zona horaria donde opere la empresa, obteniendo respuestas rápidas que siempre estarán disponibles logrando mejorar la satisfacción de los clientes, mantener los bajos costos operativos, asegurar un servicio más sostenible a largo plazo, también, como el chatbot se basa en servicios de nube bajo demanda (AWS Lambda), se paga solo por el uso real, sin necesidad de invertir en infraestructura costosa; Si el número de consultas crece, la solución en la nube puede adaptarse fácilmente. No es necesario hacer grandes cambios para atender más usuarios, los servicios de Amazon Lex y DynamoDB permiten manejar este crecimiento sin problema, además, usando DynamoDB, podemos almacenar información de los usuarios de forma segura, asegurando la privacidad y protección de los datos.

### **2. Investigación de herramientas cloud**

#### **1. Amazon DynamoDB – Almacenamiento de Datos NoSQL**

- **Descripción:**

DynamoDB es una base de datos NoSQL de gran disponibilidad y también rendimiento, es ideal para guardar datos claves y de gran valor.

- **Uso en el proyecto:**
  - Almacena las preguntas frecuentes, el historial de interacciones del chatbot y cualquier configuración específica designada por el cliente.
  - Da la posibilidad para que el chatbot ingrese rápidamente a todos los datos en tiempo real.
- **Free Tier:**
  - 25 GB de almacenamiento y 25 unidades de lectura y escritura por segundo gratis, solo durante el primer año.
  - o Adecuado para manejar las necesidades iniciales del chatbot sin costo adicional.
  - o Apropiado para administrar las necesidades iniciales del chatbot sin un costo adicional.

## **2. AWS Lambda – Lógica del Chatbot (Serverless)**

- **Descripción:**
  - AWS Lambda posibilita correr funciones bajo una demanda sin la necesidad de controlar servidores. Entonces en cada ocasión que al chatbot le entre una solicitud, una función Lambda procesa la lógica de la respuesta.
- **Uso en el proyecto:**
  - Activa funciones para verificar datos diligenciados por el usuario, como por ejemplo: consultar los horarios disponibles.
  - Enlaza el chatbot con la base de datos DynamoDB para rescatar o guardar información en tiempo real.
- **Free Tier:**
  - 1 millón de peticiones mensuales y 400,000 GB-segundos de tiempo de ejecución gratis, durante el primer año.
  - Esto da la seguridad de que el backend del chatbot se corra sin costos siempre y cuando no exceda el límite.

## **3. Amazon Lex – Procesamiento de Lenguaje Natural (NLP)**

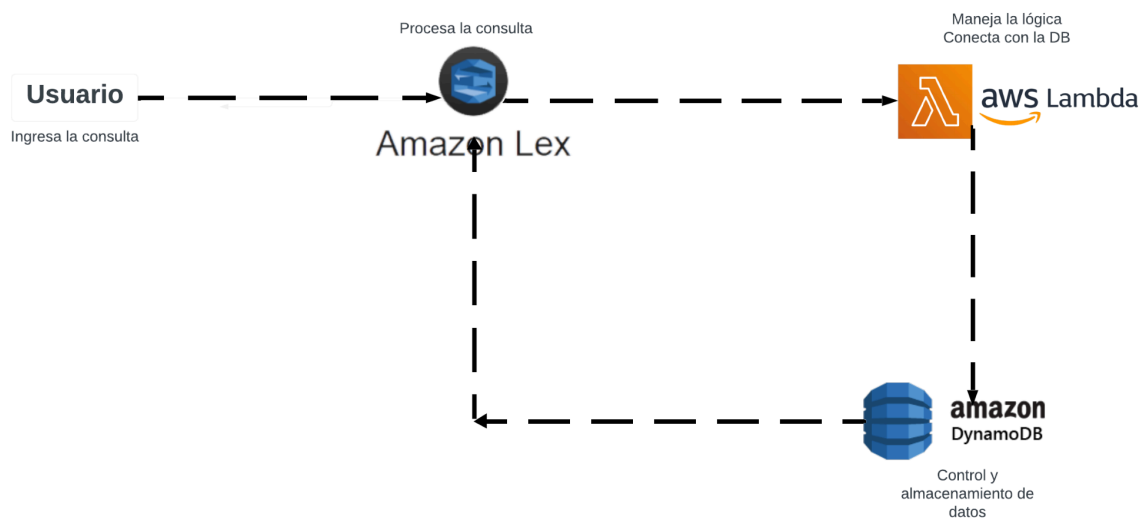
- **Descripción:**
  - Amazon Lex da la posibilidad para hacer chatbots y asistentes virtuales con las capacidades de lenguaje natural, es algo similar a un Alexa.
- **Uso en el proyecto:**
  - Esto puede procesar cualquier tipo de consultas de los usuarios además convierte texto y voz en intentos para así saber las acciones solicitadas.
  - Define cosas como: “consulta de horarios” y otros como los datos que el usuario debe dar adicionalmente.
- **Free Tier:**
  - Puede generar 10,000 textos o 5,000 audios procesados totalmente gratis por mes, solo durante el primer año.
  - Es fundamental para hacer un chatbot que puede dar respuestas a preguntas frecuentes o también para consultas rápidas sin un costo inicial.

## **3. Documento de Diseño del Proyecto**

La arquitectura del chatbot está basada en servicios de AWS, utilizando componentes serverless para minimizar costos y la complejidad de la infraestructura:

### Servicios utilizados:

- **Amazon Lex:** Procesamiento del lenguaje natural (NLP) que gestiona la interacción del usuario.
- **AWS Lambda:** Funciones serverless. Se usarán para procesar las intenciones y realizar operaciones.
- **Amazon DynamoDB:** Base de datos NoSQL, que almacenará la información relevante para el chatbot.



### Flujo de la aplicación:

1. **Input del Usuario:** El usuario ingresa una pregunta o un comando a través del chatbot.
2. **Procesamiento en Amazon Lex:**
  - **Reconocimiento de la intención:** Amazon Lex analiza la entrada del usuario y la asocia con una intención predefinida (por ejemplo, "Consultar eventos").
  - **Recogida de slots:** Si la intención requiere información adicional (como una fecha o categoría), Lex pedirá esa información al usuario.
3. **Invocación de AWS Lambda:**
  - Una vez que la intención ha sido identificada, Amazon Lex invoca una función de AWS Lambda para manejar la lógica necesaria. Por ejemplo, si el usuario pregunta por eventos, Lambda consultará DynamoDB.
4. **Interacción con DynamoDB:**

- **Lectura/escritura de datos:** Lambda se conecta con DynamoDB para leer o escribir datos. Por ejemplo, puede recuperar la lista de eventos o guardar información del usuario.

#### 5. Respuesta al Usuario:

- Lambda devuelve la respuesta a Amazon Lex, que la comunica al usuario en texto o en voz, según la interacción.

**Lambda** será el motor principal del procesamiento lógico, mientras que **DynamoDB** se encargará del almacenamiento de datos. De este modo, el procesamiento se mantiene eficiente y sin necesidad de infraestructura dedicada.

#### 4. Cronograma y Presupuesto

Semana	Fechas	Tarea	Descripción	Responsable
<b>Semana 1</b>	21/09 - 27/09	Planteamiento del problema	Identificar el problema que el chatbot resolverá, su importancia, y cómo la solución beneficiará al usuario. Justificar la relevancia del proyecto en un contexto específico (salud, educación, etc.).	Daniel Cadavid Calle
<b>Semana 2</b>	28/09 - 04/10	Investigación de herramientas cloud	Investigar herramientas AWS Free Tier que se usarán en el proyecto, como Amazon Lex para lenguaje natural, AWS Lambda para la lógica del chatbot, y DynamoDB para almacenar datos.	Jose Manuel Gomez Jimenez
<b>Semana 3</b>	05/10 - 11/10	Diseño de arquitectura y flujo de trabajo	Diseñar la arquitectura de alto nivel, describiendo cómo se conectarán los servicios en la nube (Amazon Lex, AWS Lambda, DynamoDB). Crear un diagrama visual de la arquitectura.	Mateo Herrera Villa

<b>Semana 4</b>	12/10 - 14/10	Documento de diseño del proyecto y presupuesto	Redactar el documento de diseño del proyecto, describiendo los servicios que se utilizarán, el flujo de trabajo en la nube, y cómo se gestionarán los datos. Incluir el cronograma y el presupuesto basado en los límites de AWS Free Tier.	Camilo Restrepo Tangarife
<b>Semana 5</b>	15/10	Entrega etapa 1 - Configuración de Amazon Lex	Presentar la descripción del problema, investigación de herramientas cloud, diseño de arquitectura, cronograma y presupuesto.	Todo el equipo
<b>Semana 5</b>	16/10 - 22/10	Configuración de Amazon Lex	Implementar la primera versión del chatbot en Amazon Lex. Definir los intents (intenciones del usuario), los slots (variables), y configurar los diálogos básicos para la interacción.	Camilo Restrepo Tangarife
<b>Semana 6</b>	23/10 - 29/10	Configuración de AWS Lambda	Desarrollar las funciones en AWS Lambda que gestionen la lógica de negocio del chatbot. Implementar la conexión entre Lex y Lambda para procesar las consultas del usuario y realizar acciones específicas.	Mateo Herrera Villa
<b>Semana 7</b>	30/10 - 05/11	Integración de DynamoDB	Configurar una base de datos en Amazon DynamoDB	Daniel Cadavid Calle

			para almacenar datos relacionados con el usuario (consultas previas, información relevante). Asegurar que la interacción entre Lambda y DynamoDB esté funcionando correctamente	
<b>Semana 8</b>	06/11 - 11/11	Pruebas iniciales del chatbot	Probar las interacciones entre Lex, Lambda y DynamoDB. Realizar ajustes si es necesario.	Todo el equipo
<b>Semana 9</b>	12/11 - 18/11	Preparación de la demo	Realizar pruebas funcionales finales, optimizar el flujo de trabajo y preparar la demo funcional para la presentación.	Todo el equipo

## Presupuesto

### 1. Amazon Lex

- Costo bajo Free Tier: 10,000 interacciones de texto gratuitas por mes.
- Uso Estimado: Para mantenernos dentro de este límite, podemos configurar el chatbot para que reciba no más de 10,000 interacciones de texto al mes, lo que equivale a aproximadamente 333 interacciones diarias.

### 2. AWS Lambda

- Costo bajo Free Tier: 1 millón de solicitudes gratuitas y 400,000 GB-segundos de computación por mes.
- Uso estimado: Asumiendo que el chatbot maneja una cantidad moderada de solicitudes, el uso de Lambda puede mantenerse por debajo del millón de solicitudes mensuales.

### 3. Amazon DynamoDB

- Costo bajo Free Tier: 25 GB de almacenamiento, 200 millones de lecturas y 200 millones de escrituras por mes.
- Uso Estimado: Diseñaremos el esquema de la base de datos para que el uso de lecturas y escrituras no supere este límite, lo que es factible si se optimiza el acceso a los datos.

### 4. Amazon CloudWatch

- Costo bajo Free Tier: 10 métricas y 5 GB de logs gratuitos por mes.
- Uso Estimado: Configuraremos el monitoreo para no superar el límite de métricas y optimizaremos el almacenamiento de logs.

### **Total estimado de costos**

- Mensual: Dentro de los límites del Free Tier, se puede mantener un costo de \$0 si se controlan adecuadamente las interacciones y el uso de servicios.

### **Estrategias para evitar sobre costos**

#### **1. Control de Interacciones:**

- Limitar el uso del chatbot a 10,000 interacciones de texto al mes, implementando restricciones en el número de usuarios o el tiempo de acceso al servicio.

#### **2. Monitoreo y Alertas:**

- Utilizar CloudWatch para establecer alertas en el uso de servicios, lo que permitirá supervisar el uso en tiempo real y recibir notificaciones si se acercan a los límites.

#### **3. Optimización del Código:**

- Diseñar el código de Lambda para manejar múltiples solicitudes de manera eficiente, minimizando el tiempo de ejecución y evitando solicitudes innecesarias.

#### **4. Optimización de DynamoDB:**

- Configurar lecturas/escrituras y almacenamiento de manera eficiente, realizando operaciones en lote y limitando el acceso a la base de datos solo cuando sea necesario.

#### **5. Revisiones Mensuales:**

- Revisar mensualmente el uso de servicios para identificar patrones y ajustar el acceso al chatbot, asegurando que no se excedan los límites.

### **Permisos necesarios:**

- 1. Amazon Lex:** Crear y gestionar el chatbot (intents, diálogos).
- 2. AWS Lambda:** Crear y ejecutar funciones serverless.
- 3. Amazon DynamoDB:** Crear y gestionar tablas para almacenamiento de datos.
- 4. IAM Roles:** Crear roles que permitan la interacción entre los servicios (Lex, Lambda, DynamoDB).
- 5. Acceso a AWS Console:** Monitorear y gestionar los servicios.

### **Recursos necesarios:**

- 1. Amazon Lex:** Chatbot con 5 intents y capacidad para 10 sesiones simultáneas.
- 2. AWS Lambda:** Tres funciones Lambda con un tiempo de ejecución máximo de 5 segundos por solicitud.
- 3. Amazon DynamoDB:** Dos tablas con capacidad de 25 GB y 25 solicitudes por segundo.
- 4. CloudWatch:** Monitoreo de logs y rendimiento de los servicios.