



Proyecto integrador (Gpo 10)

Dra. Grettel Barceló Alonso

Dr. Luis Eduardo Falcón Morales

Mtra. Verónica Sandra Guzmán de Valle

Proyecto: Como la GenIA está redefiniendo el diseño: Partir de lo abstracto para llegar a lo concreto

Alumnos:

Ian André Ortega Suárez-A01021190

Enrique Ricardo García Hernández-A01315428

Proyecto en colaboración con el Dr. Ricardo A. Ramirez-Mendoza, Profesor del área de Mecatrónica e Ingeniería Mecánica, con base a su reporte técnico “Engineering the Present: How Generative AI Models Are Redefining Design”

Avance 5 Modelo Final

Domingo 2 de Junio de 2024

Introducción.	1
Objetivo del proyecto.	1
Etapas del proyecto	2
1. Ensamble	2
2. Selección	3
3. Gráficos	4
4. Conclusiones	10
5. Referencias bibliográficas	11

Introducción.

En el ámbito de la Inteligencia Artificial (IA) y el Procesamiento de Lenguaje Natural (PLN), los chatbots han emergido como herramientas esenciales para mejorar la interacción entre humanos y sistemas informáticos. El desarrollo de chatbots inteligentes y conversacionales ha evolucionado gracias a la integración de modelos de lenguaje generativos (GenIA) y plataformas de IA avanzadas. Este proyecto se centra en el diseño y la implementación de un chatbot utilizando herramientas de Google Cloud Platform, como Vertex AI, Dialogflow, y Google Chat API. El objetivo es crear un sistema robusto y eficiente que pueda comprender y responder de manera coherente a las consultas de los usuarios, mejorando así la experiencia del usuario y la eficiencia operativa.

Objetivo del proyecto.

1. **Desarrollar un chatbot inteligente** utilizando Vertex AI, Dialogflow y Google Chat API.
2. **Optimizar la comprensión del lenguaje natural** mediante el uso de modelos de lenguaje generativo.
3. **Evaluar el rendimiento del chatbot** a través de métricas específicas de PLN.
4. **Proponer mejoras y futuras expansiones** basadas en los resultados obtenidos.

Etapas del proyecto

1. Ensamble

En esta etapa, el objetivo es integrar todos los componentes desarrollados previamente para asegurar que el chatbot funcione de manera cohesiva y eficiente. El ensamblaje implica la configuración final y la puesta en marcha del sistema completo.

Pasos en la etapa de Ensamble:

2. Integración de componentes

- a. Conexión del modelo de lenguaje generativo (GenIA) con Dialogflow.
 - Utilizar Vertex AI para desplegar el modelo de lenguaje generativo.
 - Configurar Dialogflow para que pueda enviar las consultas de los usuarios al modelo de Vertex AI y recibir las respuestas generadas.
- b. Configuración de la API de Google Chat.
 - Implementar la API de Google Chat para conectar el chatbot con Google Chat.
 - Configurar los endpoints necesarios para recibir mensajes entrantes y enviar respuestas desde el chatbot.
- c. Desarrollo de Cloud Functions.
 - Crear funciones en Google Cloud Functions que manejen la lógica de negocio, como el procesamiento de entradas, la gestión de sesiones de usuario, y la llamada a Vertex AI para generar respuestas dadas las interacciones del usuario que realiza la consulta.

3. Pruebas de integración.

- a. Pruebas de conectividad.
 - Verificar que todos los componentes se comunican correctamente entre sí.
 - Asegurarse de que las consultas enviadas desde Google Chat lleguen a Dialogflow y sean procesadas adecuadamente por Vertex AI.
- b. Pruebas funcionales.
 - Realizar pruebas exhaustivas de la interacción entre los componentes para garantizar que las respuestas del chatbot sean coherentes y relevantes.
 - Identificar y corregir cualquier problema de comunicación o procesamiento que surja durante las pruebas.

4. Ajustes finales

- a. Optimización del modelo.
 - Ajustar los parámetros del modelo generativo en Vertex AI para mejorar el rendimiento basado en los resultados de las pruebas.
- b. Optimización de la lógica de Dialogflow
 - Revisar y ajustar las intenciones y entidades configuradas en Dialogflow para asegurar una comprensión precisa de las consultas de los usuarios.
- c. Ajuste de desempeño
 - Implementar mejoras en el rendimiento del sistema, como optimizaciones en las Cloud Functions y ajustes en la infraestructura de Google Cloud para garantizar una respuesta rápida y eficiente.

2. Selección

Pasos en la etapa de Ensamble:

a) Evaluación de los modelos.

- i) Comparación de modelos generativos.
 - Probar diferentes versiones del modelo de lenguaje generativo en Vertex AI (e.g., GPT-4, GPT-3) para seleccionar el mejor en términos de precisión y rendimiento.
 - Evaluar el desempeño de cada modelo utilizando un conjunto de datos de validación específico del dominio.
- ii) Evaluación de intenciones y entidades de Dialogflow
 - Evaluar la efectividad de las intenciones y entidades configuradas en Dialogflow mediante pruebas con datos reales y simulados.
 - Ajustar la configuración según los resultados de las pruebas para mejorar la precisión en la comprensión de las consultas de los usuarios.

b) Criterios de selección.

- i) Métricas de evaluación.
 - Definir métricas clave de evaluación como precisión, recall, F1-score, y tiempo de respuesta.
 - Utilizar estas métricas para comparar el desempeño de diferentes configuraciones y modelos.
- ii) Selección de configuración óptima.
 - Seleccionar las configuraciones que maximicen estas métricas para asegurar la mejor experiencia de usuario.
 - Realizar pruebas A/B para comparar el rendimiento de diferentes configuraciones en un entorno controlado.

c) Herramientas y recursos

- i) Decisión sobre recursos adicionales.
 - Evaluar la necesidad de utilizar bases de datos externas, API de terceros, o herramientas adicionales para enriquecer la funcionalidad del chatbot.
 - Seleccionar los recursos que aporten mayor valor al proyecto, asegurando su integración eficiente con el sistema existente.

3. Gráficos

Esta etapa se centra en la visualización de datos y resultados para comunicar de manera clara y efectiva los hallazgos del proyecto. Los gráficos ayudarán a ilustrar el rendimiento del chatbot y los beneficios alcanzados.

a) Visualización de resultados

- i) Gráficas de desempeño del modelo.
 - Crear gráficos de barras y líneas para mostrar métricas de rendimiento como precisión, recall y F1-score.
 - Visualizar el tiempo de respuesta del chatbot bajo diferentes configuraciones y cargas de trabajo.
- ii) Distribución de intenciones y entidades.
 - Generar gráficos e histogramas para mostrar la distribución de las intenciones y entidades reconocidas por el chatbot.
 - Analizar cómo las diferentes intenciones se distribuyen entre las consultas de los usuarios.

b) Análisis comparativo

- i) Comparación de resultados antes y después de optimización
 - Utilizar gráficos de líneas y boxplots para comparar el rendimiento del chatbot antes y después de la optimización de los modelos y configuraciones.
- ii) Visualización de variabilidad.
 - Crear gráficos de dispersión para ilustrar la variabilidad en el rendimiento del chatbot bajo diferentes condiciones.
 - Analizar los puntos fuera de lo común para identificar posibles áreas de mejora.

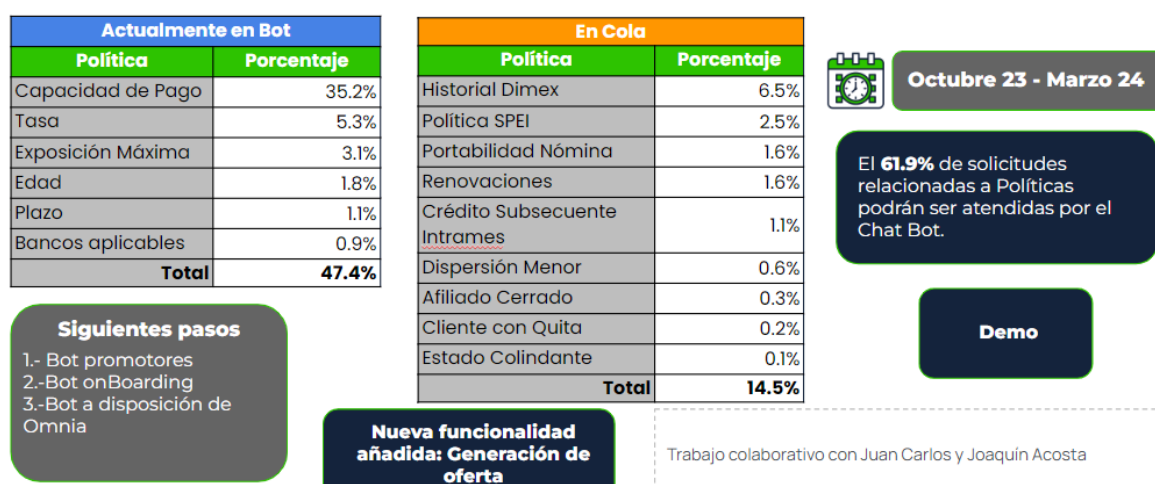
c) Tableros / dashboards interactivos

- i) Desarrollo de tableros en Looker Studio o alguna herramienta de visualización y análisis.
 - Crear un dashboard interactivo utilizando herramientas como Looker Studio para permitir una exploración detallada de los datos.
- ii) Incorporación de gráficos dinámicos

- Incluir gráficos dinámicos que permitan filtrar y segmentar los datos para un análisis más profundo.
 - Facilitar la visualización de tendencias y patrones en el rendimiento del chatbot.
- iii) Ajuste de desempeño
- Implementar mejoras en el rendimiento del sistema, como optimizaciones en las Cloud Functions y ajustes en la infraestructura de Google Cloud para garantizar una respuesta rápida y eficiente.

Chat Bot | Políticas de Créditos

Los JEDI's recibieron un total de **10,960** solicitudes relacionadas a Políticas de Créditos.



Este proyecto atenderá el **61.9%** de las llamadas recibidas al call center, generando una eficiencia en la capacidad instalada de la compañía y que ese conocimiento y habilidades se puedan ocupar en otras actividades de mayor valor.

Flujo del chatbot.

Flujo del Chatbot

Se implementa el flujo que seguirá el chatbot, así como la recopilación de la información necesaria para dar una respuesta de crédito

```
def conversacionChatbot():
    ## Flujo de la conversación
    limite_superior_edad = 76
    limite_inferior_edad = 25
    estatus = 0
    interes_aceptar = 'alternativa'

    while estatus == 0:
        interesa_credito = obtenerInteresaCredito() ## Se pregunta si el usuario quiere un crédito
        if interesa_credito == True:
            edad_cliente = obtenerEdadCliente() ## Se pregunta la edad del cliente
            if ((edad_cliente >= limite_inferior_edad) & (edad_cliente <= limite_superior_edad)):
                estatus_cliente = obtenerEstatusCliente() ## Se obtiene si el cliente es jubilado, pensionado o activo
                if estatus_cliente == "NINGUNA":
                    agradecimientos()
                    estatus = 1
                else:
                    bool_institucion = obtenerInstitucionBool(estatus_cliente) ## Se obtiene si el cliente el jubilado esta afiliado a una institución de convenio
                    if bool_institucion == False:
                        agradecimientos()
                        estatus = 1
                    else:
                        cliente_institucion = obtenerInstitucionCliente(estatus_cliente) ## Se obtiene la institución a la que está afiliado el cliente
                        if cliente_institucion == 'NINGUNA':
                            agradecimientos()
                            estatus = 1
                        else:
                            cliente_banco = obtenerBancoCliente(estatus_cliente, cliente_institucion)
                            if cliente_banco == 'NINGUNA':
                                agradecimientos()
                                estatus = 1
                            else:
                                cliente_nomina = obtenerNominaCliente(cliente_banco, cliente_institucion, estatus_cliente)
                                oferta_cliente = obtenerOfertaCliente(estatus_cliente, cliente_institucion, cliente_banco, cliente_nomina)
                                if oferta_cliente == "SIN OFERTA":
                                    print('Gracias por escribirnos, sin embargo en estos momentos no tenemos una oferta disponible para usted.')
                                    estatus = 1
                                else:
                                    print(oferta_cliente)
                                    while (interes_aceptar == 'alternativa'):
                                        interes_aceptar = obtenerInteresAceptarCredito()
                                    if interes_aceptar == False:
                                        dudas = obtenerDudas()
                                        if dudas == False:
                                            agradecimientos()
                                            estatus = 1
                                        else:
                                            cliente_telefono = obtenerTelefonoCliente()
                                            print('Bot: En unos momentos más nos comunicaremos con usted. Muchas gracias por su paciencia.')
                                            estatus = 1
                                    else:
                                        medio_comunicacion = obtenerMedioComunicacion()
                                        if medio_comunicacion == 'sucursal':
                                            mostrarSucursales()
                                            mostrarDocumentos()
                                            print("\nBot: Gracias por contactarnos. Que tenga un excelente día.")
                                            estatus = 1
                                else:
                                    agradecimientos()
                                    estatus = 1
            else:
                confirma_no_credito = confirmacionNoCredito()
                if confirma_no_credito == True:
                    estatus = 1
```

Conversación del chatbot

```
conversacionChatbot()

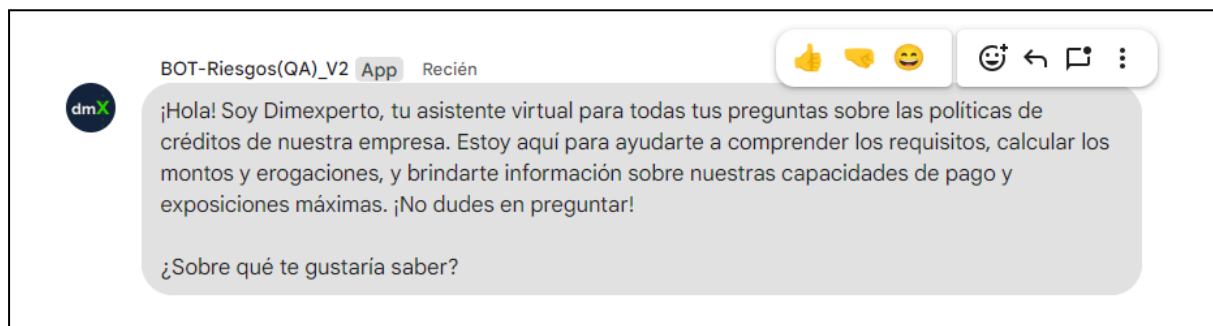
Bot: Hola, ¿estás buscando un crédito?
Tú: sí
Bot: ¡Perfecto! Por favor respuesta a la siguiente serie de preguntas para saber qué es lo que le podemos ofrecer
Bot: ¿Qué edad tiene usted?
Tú: 66
Bot: ¿Me podría decir si es jubilado, pensionado o si aún se mantiene activo en el ámbito laboral?
Tú: jubilado
Bot: ¿Es jubilado de alguna de estas instituciones? IMSS, ISSSTE, CFE, PEMEX
Tú: sí
Bot: ¿En qué institución es usted jubilado? (IMSS, ISSSTE, PEMEX, CFE)
Tú: IMSS
Bot: ¿En qué banco le depositan su jubilación IMSS?
Tú: banorte
Bot: ¿Cuánto le depositan en BANORTE de su jubilación IMSS?
Tú: 10 mil
Bot: ¿Podría ingresar el monto que se le deposita?
Bot: ¿Cuánto le depositan en BANORTE de su jubilación IMSS?
Tú: 10,000
Bot: ¿Podría ingresar el monto que se le deposita?
Bot: ¿Cuánto le depositan en BANORTE de su jubilación IMSS?
Tú: 10,000
Bot: ¿Podría ingresar el monto que se le deposita?
Bot: ¿Cuánto le depositan en BANORTE de su jubilación IMSS?
Tú: 10000
Bot: Buenas noticias, con la información que nos ha proporcionado le podemos ofrecer hasta $25500.0 pagando $2500.0 mensuales durante 36 meses
Bot: La oferta realizada está sujeta a revisión de capacidad de pago, presentando la documentación requerida.
Bot: ¿Le interesa la oferta que le acabo de mostrar?
Tú: sí
Bot: ¡Perfecto!
Bot: ¿Por qué medio le gustaría que lo siguiéramos atendiendo? (Kiosko más cercano, Sucursal más cercana, Descargar la app en su dispositivo móvil, Call Center)
Tú: sucursal
Bot: Muy bien

Bot: Las sucursales más cercanas a ti son:
1. Alameda
2. San Pedro
3. Garza Sada

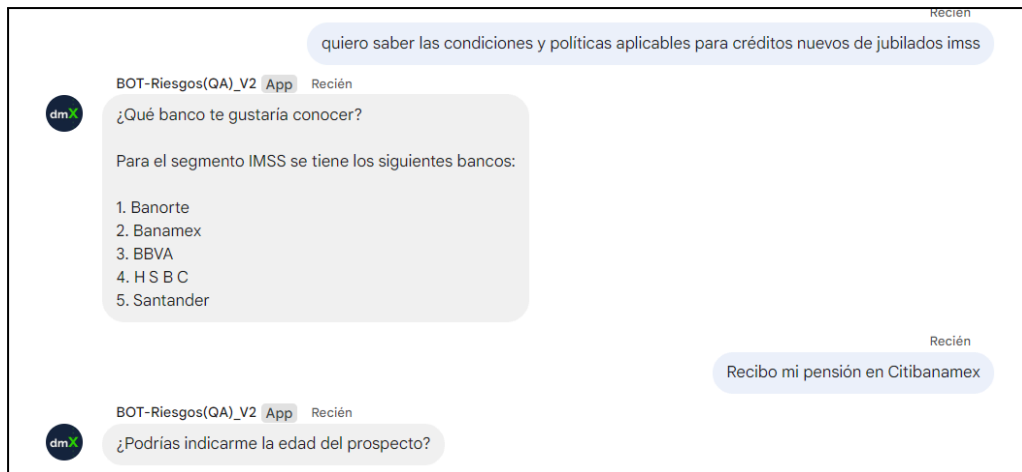
Bot: Vamos a requerir la siguiente documentación:
1. Identificación Oficial
2. Comprobante de ingresos
3. Estados de cuenta
4. Comprobante de domicilio

Bot: Gracias por contactarnos. Que tenga un excelente día.
```

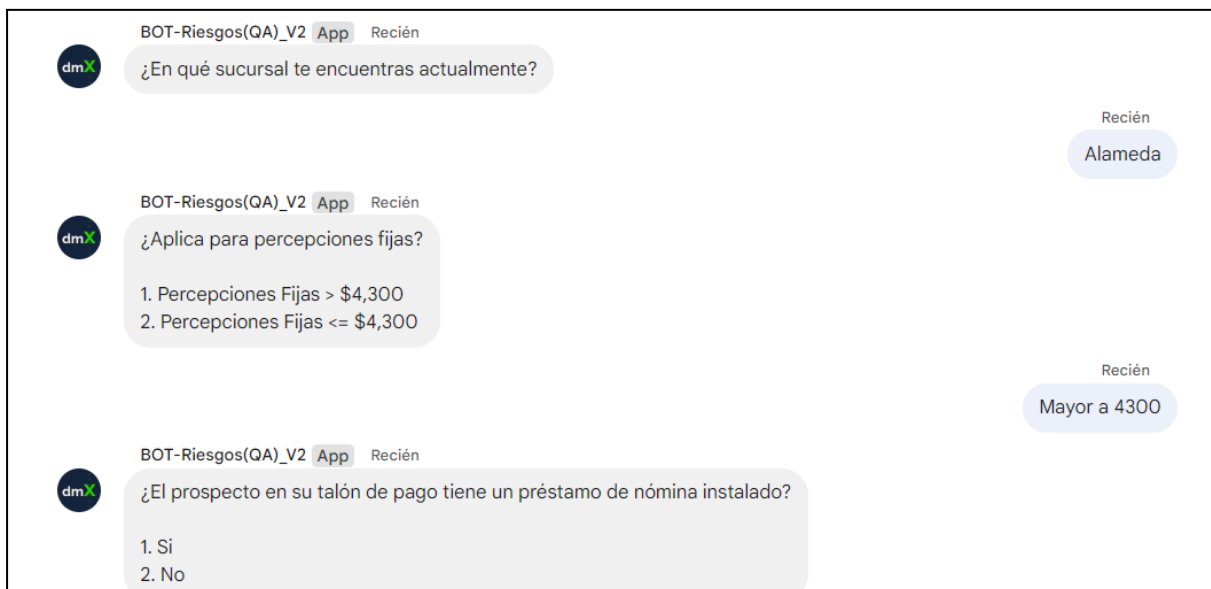
Puesta en producción



Sabe interactuar y homologar las etiquetas de Citibanamex, Citi y Banamex



Además solicita información adicional para perfilar mejor al solicitante



Con base en la información, antes de dar una propuesta pregunta para validar los datos previamente obtenidos

Información

dmX

Por favor valida que los resultados de tu consulta sean correctos:

- Segmento: IMSS
- Producto: CREDITO NUEVO
- Banco: BANAMEX
- Giro: Pensionado
- Edad: 68.0
- Sucursal: ALAMEDA

¿Los datos son correctos?

Recién

Una vez obtenida y validada la información se proponen las condiciones iniciales del crédito

dmX

BOT-Riesgos(QA)_V2

App

2 min

La oferta que tiene DIMEX para un cliente Pensionado de IMSS que solicita un CREDITO NUEVO y recibe su pensión en el banco BANAMEX es de un monto máximo de hasta \$100,000.0 con una capacidad de pago del 80.0% de su ingreso, con opción de tasas de 83.0% a 105.0% y plazos de 12 a 36 meses.

¿Te gustaría conocer la oferta que le podríamos hacer a un prospecto con las características anteriores?

Preguntas referentes al ingreso neto recurrente y otras deducciones.

dmX

BOT-Riesgos(QA)_V2

App

Recién

¡Perfecto! Necesitaré la siguiente información:

¿Me podrías indicar el ingreso neto del prospecto? Ejemplo: 10000

Recién

45000

dmX

BOT-Riesgos(QA)_V2

App

Recién

Por favor ingresa las domiciliaciones con otras financieras con las que cuenta el prospecto. (Ejemplo: 4000)

Recién

2500

dmX

BOT-Riesgos(QA)_V2

App

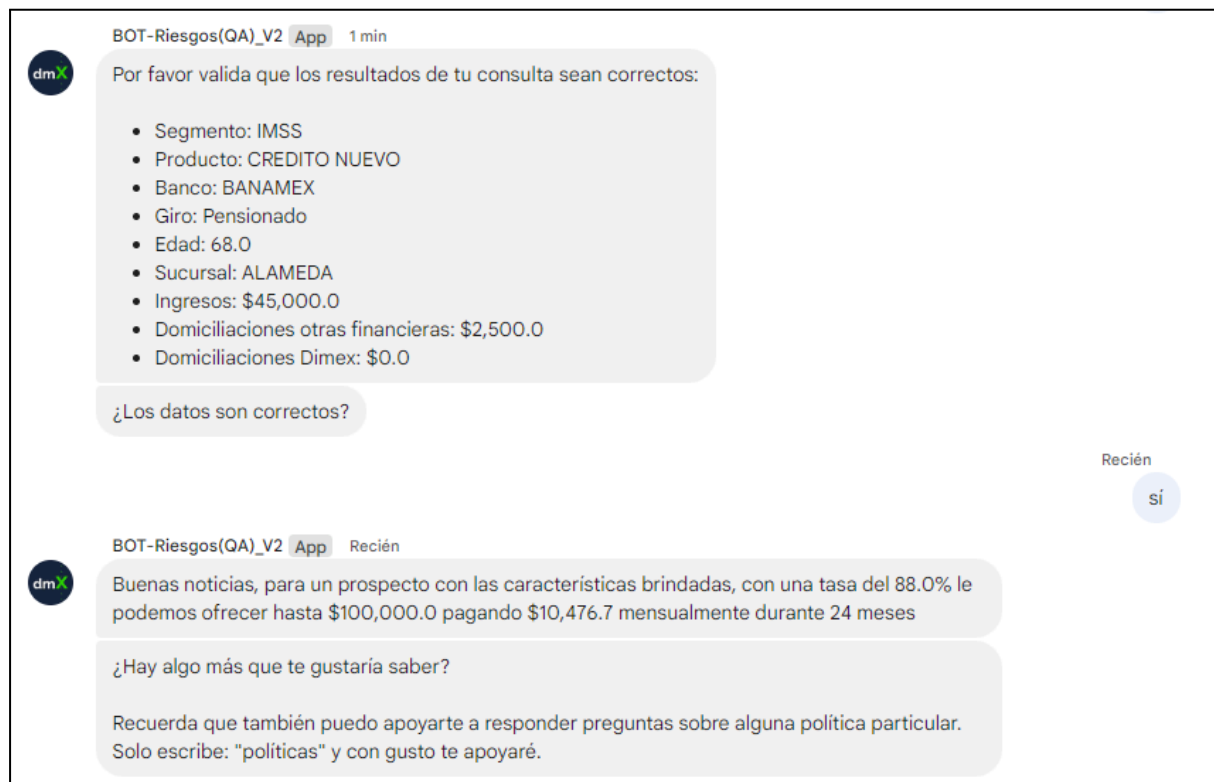
Recién

Por favor ingresa las domiciliaciones de créditos dimex con las que cuenta el prospecto. (Ejemplo: 4000)

Recién

0

Con base en la información proporcionada, da una propuesta inicial (sujeto a validación de la información otorgada por el solicitante)



4. Conclusiones

El desarrollo de un chatbot inteligente mediante herramientas avanzadas de IA, como Google Cloud Platform, Vertex AI, Dialogflow y Google Chat API, ha demostrado ser un proyecto integral y desafiante, pero altamente gratificante. A lo largo de este proceso, se han abordado y superado diversos desafíos técnicos, desde la integración de modelos de lenguaje generativo hasta la configuración de API de comunicación y la optimización del rendimiento del sistema.

El ensamblaje de todos los componentes permitió la creación de un sistema cohesivo que puede comprender y responder de manera efectiva a las consultas de los usuarios, mejorando significativamente la experiencia de usuario. La selección cuidadosa de modelos y configuraciones óptimas, basada en métricas específicas de rendimiento, aseguró la implementación de las mejores prácticas y el aprovechamiento de las tecnologías más adecuadas para este proyecto.

La etapa de visualización de datos mediante gráficos permitió una comprensión profunda del rendimiento del chatbot y facilitó la identificación de áreas de mejora. Estas visualizaciones no solo mostraron el éxito del proyecto en términos de

precisión y eficiencia, sino que también destacaron las oportunidades para futuras optimizaciones y expansiones.

Las lecciones aprendidas durante el desarrollo de este chatbot proporcionan una base sólida para futuros proyectos en el campo del Procesamiento de Lenguaje Natural y la Inteligencia Artificial. Las propuestas de mejora y el plan de mantenimiento y actualización continua asegurarán que el chatbot permanezca relevante y efectivo a lo largo del tiempo.

En resumen, este proyecto no solo cumple con los objetivos planteados, sino que también establece un marco robusto para el desarrollo de chatbots inteligentes en el futuro. Las capacidades mejoradas de interacción y respuesta del chatbot abren nuevas posibilidades para aplicaciones en diversos dominios, desde el servicio al cliente hasta la educación y la asistencia personal. La experiencia adquirida y las mejores prácticas documentadas servirán como una guía valiosa para la comunidad académica y profesional en la implementación de soluciones de IA avanzadas.

5. Referencias bibliográficas

- A. Google Cloud. (2024). Google Cloud Platform documentation. Recuperado de <https://cloud.google.com/docs>
- B. Google Cloud. (2024). Vertex AI documentation. Recuperado de: <https://cloud.google.com/vertex-ai/docs>
- C. Google Cloud. (2024). Dialogflow documentation. Recuperador de: <https://cloud.google.com/dialogflow/docs>
- D. Google Cloud. (2024). Google Chat API documentation. Recuperado de: <https://developers.google.com/chat>
- E. OpenAI. (2024). [GPT-4 API documentation](https://beta.openai.com/docs). Recuperado de: <https://beta.openai.com/docs>
- F. Raj, A., & Rathore, V. (2023). Natural Language Processing with Python. Packt Publishing.