

Diseño e implementación de un Bot para atender consultas sobre Capstone design y trabajos de grado en los programas de pregrado de la E3T

Juan Nicolas Mora Bernal y Davidson Duvan Leal Mogollón

Trabajo de Grado para Optar al Título de Ingeniero Electrónico

Director

Homero Ortega Boada

Doctor en ciencias de la ingeniería, radio comunicaciones

Codirector (Externo)

Nelson Enrique Trillos León

Ingeniero Electrónico

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Ingeniería Electrónica

Bucaramanga

2023

Dedicatoria

A Dios que me ha permitido estar donde estoy y me ha dado mucha fortaleza y talentos.

A mi madre Liliana Mora, que siempre ha estado allí para mí, me ha formado y me ha apoyado en cada etapa de mi vida.

A mis abuelos Hernando Mora y Lilia de Mora quienes son mi familia más cercana y me han apoyado siempre para poder estar donde estoy hoy en día.

Además, quisiera agradecer a mi familia, incluyendo a mis tíos, tías y primos, quienes han sido una fuente constante de apoyo y motivación en mi vida.

A los profesores y personas que han contribuido en mi formación, no solo en la parte académica, si no también que han compartido sus experiencias personales, lo que tiene mucho valor y ha contribuido en mi formación personal.

A mis amigos, que tanto fuera de la carrera han sido un apoyo fundamental y también dentro de la carrera he conocido increíbles personas con las cuales pude compartir y en muchas ocasiones ayudarnos en nuestra formación profesional.

Juan Nicolas Mora Bernal

Dedicatoria

Inicialmente a Dios, quien ha sido mi guía en todo momento, brindándome entendimiento, sabiduría y fortaleza para continuar en el camino hacia mis metas. Agradezco por su presencia constante en el desarrollo de mi vida y carrera profesional, y por haberme sostenido en cada uno de los desafíos que he enfrentado.

A mis abuelos, Isaías Mogollón Blanco y Aydee Rojas Gualdrón, quienes han sido un ejemplo de amor, apoyo y guía en mi vida. Gracias por haberme orientado y ayudado en todo momento, siendo un ejemplo transparente del valor de la familia ante las dificultades. Su presencia en mi vida ha sido fundamental para alcanzar este logro, y estoy profundamente agradecido por ello.

A mi madre, Ana Milena Mogollón Rojas, quien ha sido un ejemplo invaluable de perseverancia y determinación en mi vida. Gracias por enseñarme que los sacrificios traen recompensas y por haberme inculcado la importancia de luchar por lo que quiero. Agradezco infinitamente por haberme brindado la oportunidad de estudiar y por haber creído en mí y en mis capacidades. Su amor, dedicación y paciencia han sido fundamentales en mi camino hacia este logro, y estaré eternamente agradecido por ello.

A los profesores, que hicieron parte de mi formación académica y personal, por sus valiosas enseñanzas que me ayudaron a ser lo que soy hoy en día. Agradezco infinitamente su dedicación y compromiso en mi proceso de aprendizaje, por haberme brindado su sabiduría y experiencia para ayudarme a crecer como persona y como profesional.

A mis amigos y colegas: Carlos Rueda, Javier Izquierdo, Diego Jaimes, Fabian Ruiz, Arley Domínguez, Elizabeth Pérez, Anny Godoy y Heyder Lizarazo, por haber sido una parte

fundamental de mi formación como Ingeniero Electrónico. Su amistad, apoyo y ánimo han sido fundamentales en el desarrollo de este proyecto de vida. Siempre recordaré las experiencias compartidas y los momentos difíciles superados.

A mi compañero de proyecto, Juan Nicolas Mora por la continua dedicación, apoyo y perseverancia para la realización de este trabajo de grado.

Davidson Duvan Leal Mogollón

Agradecimientos

A la Universidad Industrial de Santander, por desarrollar y potenciar continuamente nuestra formación moral, ética y como profesionales de Ingeniería Electrónica.

Al Grupo de Investigación RadioGis, que nos brindó la oportunidad de realizar este proyecto de grado y nos guio en la consolidación de nuestros conocimientos a través del proceso investigativo desarrollado. Su apoyo y compromiso han sido fundamentales para alcanzar este logro académico.

Al Doctor Homero Ortega Boada, por compartir sus valiosos conocimientos, acompañamiento, apoyo y compromiso en la ejecución del presente trabajo de grado.

Al Ingeniero Nelson Enrique Trillos León, por la constante dedicación y esfuerzo brindado para el desarrollo y culminación exitosa del presente trabajo de grado.

A los estudiantes Juan Isidro, Omar Vargas, Oscar Olejua, Camilo Santos y Daniel Espinoza por estar dispuestos a ayudarnos en las pruebas, ya que su retroalimentación fue fundamental para la culminación del proyecto.

Tabla de contenido

Introducción	17
1. Objetivos.....	19
1.1 Objetivo General.....	19
1.2 Objetivos Específicos.....	19
2. Marco de referencia	20
2.1 Inteligencia artificial	20
2.1.1 Machine Learning	20
2.1.2 Aprendizaje supervisado.....	21
2.1.3 Aprendizaje no supervisado.....	21
2.1.4 Aprendizaje por refuerzo	21
2.1.5 Redes neuronales artificiales.....	22
2.1.6 Deep learning.....	22
2.1.7 Procesamiento de lenguaje natural (NLP)	22
2.1.8 Medidas de rendimiento.....	22
2.2 Chatbot.....	24
2.2.1 Conceptos generales de los chatbots en Dialogflow CX	24
2.3 Google Cloud Platform.....	26
2.4 Aplicaciones web	27
2.4.1 Apps Script.....	27
3. Marco de referencia	27
3.1 Chatbot como una herramienta de referencia virtual en una biblioteca.....	28

3.2 Artículo que compara el desempeño de diferentes plataformas para chatbots	28
3.3 Discusión sobre el uso de chatbots en la educación en la actualidad	29
3.4 Trabajo realizado en el grupo de investigación RadioGis	29
4. Requerimientos y restricciones de diseño	29
5. Selección de la plataforma	30
5.1 Dialogflow CX.....	31
5.1.1 Características	31
5.1.2 Costos.....	32
5.2 Amazon Lex.....	33
5.2.1 Características	33
5.2.2 Costos.....	33
5.3 IBM Watson Assistant	34
5.3.1 Características	34
5.3.2 Costos.....	35
5.4 Microsoft Azure Bot Service	35
5.4.1 Características	35
5.4.2 Costos.....	35
5.5 Criterio de selección	36
6. Recopilación de preguntas	37
7. Casos de uso.....	37
7.1 Rol estudiante.....	37
7.2 Rol desarrollador.....	38
8. Diseño	38

8.1 Diseño de la arquitectura del chatbot.....	38
8.1.1 Selección de la plataforma Dialogflow CX	39
8.1.2 Definición de las paginas	39
8.1.3 Definición de los intents	40
8.1.4 Definición de las routes	40
8.1.5 Definición del flujo conversación.....	40
8.1.6 Diseño del método para actualizar el chatbot	42
8.1.7 Integración con otras herramientas	42
8.2 Diseño de la interfaz del usuario.....	43
8.3 Diseño de la base de conocimiento.....	43
8.3.1 Estructuración de la información.....	43
8.3.2 Carga de la información en el chatbot	44
8.4 Diseño de las pruebas.....	44
8.5 Diseño de la arquitectura de la solución	45
9. Implementación.....	47
9.1 Implementación en Dialogflow CX	47
9.1.1 Creación y configuración de un intent	48
9.1.2 Configuración de una Route	49
9.1.3 Resumen del funcionamiento.....	51
9.2 Programa para la carga de datos y actualización del chatbot.....	52
9.2.1 Resumen del funcionamiento del código.....	52
9.2.2 Explicación del código.....	53
9.2.3 Instrucciones para ejecutar el código.....	58

9.3 Script para guardar las preguntas de los usuarios	58
9.4 Integración del chatbot en la página web.....	59
10. Análisis de resultados	60
10.1 Descripción de la prueba.....	61
10.2 Resultados obtenidos	62
11. Trabajos futuros	64
12. Conclusiones.....	64
13. Recomendaciones	66
Referencias Bibliográficas	67
Apéndices	70

Lista de Tablas

	Pág.
Tabla 1. Precios aproximados para la plataforma Dialogflow CX	32
Tabla 2. Ejemplo de precios para la plataforma Amazon Lex.....	33
Tabla 3. Ejemplo del formato de las pruebas.....	61
Tabla 4. Ejemplo de resultados para cada clase.....	62
Tabla 5. Métricas de rendimiento promedio	63

Lista de Figuras

	Pág.
Figura 1. Flujo conversacional planteado para el proyecto	41
Figura 2. Diagrama de arquitectura de solución	45
Figura 3. Configuración de un intent	48
Figura 4. Asociar un intent a una route.....	49
Figura 5. Configuración de una route	50
Figura 6. Asociar una transition a una route	51
Figura 7. Instalación de las librerías	53
Figura 8. Subir los archivos credenciales	53
Figura 9. Eliminar lo que este en la página “actualizar”	54
Figura 10. Leer y guardar la información del documento	55
Figura 11. Eliminar y crear nuevos intents	56
Figura 12. Creación de routes en la página “actualizar”	57
Figura 13. Chatbot implementado en la página web.....	59

Lista de Apéndices

**Ver apéndices adjuntos y pueden ser consultados en la base de datos de la
Biblioteca UIS**

Apéndice A. Lista de preguntas y respuestas recopiladas

Apéndice B. Documento de Google sheets para actualizar el chatbot

Apéndice C. Código para actualizar el chatbot a partir de un documento de Google sheets

Apéndice D. Código del script para guardar las preguntas de los usuarios

Apéndice E. Pruebas de rendimiento

Glosario

Capston Design: se refiere a una experiencia mayor de diseño en ingeniería, que se realiza en los trabajos de grado del Programa de Ingeniería Electrónica en la E3T.

Colab: es una plataforma de Google que permite programar y ejecutar código en lenguaje Python desde el navegador, además nos da la posibilidad de aumentar nuestro poder de cómputo a partir de máquinas virtuales. (Google, 2023)

E3T: es como se le conoce a la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

FAQ: es la abreviatura que se usa para preguntas frecuentes en el idioma inglés.

HTML: es un lenguaje que se utiliza para crear páginas web, su estructura tiene una serie de etiquetas que se utilizan para crear las diferentes secciones de una página web, y finalmente el navegador web interpreta estas etiquetas para mostrarle el contenido de la página web al usuario. (World Wide Web Consortium, 2023)

JavaScript: es un lenguaje de programación muy conocido que se usa comúnmente en páginas web y en algunos entornos de desarrollo. (Mozilla, 2023)

JSON: es un tipo de formato que permite intercambiar datos de manera sencilla entre sistemas y aplicaciones. Es fácilmente legible y editable para los humanos, al mismo tiempo que las máquinas lo pueden analizar y crear sin dificultad, lo que facilita la comunicación entre sistemas. (JSON, 2023)

POST: El método POST es utilizado para enviar datos al servidor para que sean procesados y almacenados, pudiendo ser utilizados para distintas funciones como la creación de nuevos

recursos, publicación de mensajes, envío de formularios y anotaciones en recursos existentes.

(Internet Engineering Task Force, 2014)

Python: es uno de los lenguajes de programación más utilizados actualmente y es muy común ver que se usa en aplicaciones relacionadas con inteligencia artificial. (Python, 2023)

Resumen

Título: Diseño e implementación de un Bot para atender consultas sobre Capstone Design y trabajos de grado en los programas de pregrado de la E3T*

Autor: Juan Nicolas Mora Bernal y Davidson Duvan Leal Mogollón**

Palabras Clave: Chatbot, Agente virtual, Inteligencia artificial, Chatbot de preguntas frecuentes

Descripción: En respuesta a la creciente necesidad de herramientas virtuales eficientes en el ámbito educativo, este proyecto presenta una solución innovadora: la implementación de chatbots en la educación. Con el objetivo de mejorar la experiencia educativa de estudiantes y profesores, automatizando el proceso para responder preguntas repetitivas sobre una asignatura. Por lo cual, se desarrolló e implemento un chatbot que fue entrenado con unas preguntas que fueron recopiladas durante dos semestres distintos. Se realizo utilizando la plataforma de Dialogflow CX, y además se implementó un código en el lenguaje de Python que permite que las preguntas y respuestas del chatbot se puedan actualizar a partir de la información contenida en un documento de Google sheets, lo que facilita al docente su utilización sin necesidad de tener conocimientos avanzados en tecnología, además se realizó una aplicación web para recopilar las preguntas que hacen los usuarios, esto se hace con el propósito de agregar una funcionalidad de mejora continua al chatbot implementado, lo que permitirá su evolución y mejoramiento constante a lo largo del tiempo. La implementación realizada fue probada por estudiantes de últimos semestres, los estudiantes que probaron el chatbot expresaron una alta satisfacción con su funcionamiento y manifestaron que les pareció una herramienta útil para su proceso formativo, por lo que se espera que tenga un impacto positivo en el desarrollo de la asignatura Trabajo de Grado I y permita mejorar su experiencia en el curso, además de que se genere interés por este tipo de herramientas tecnológicas que están en auge y proveen de muchas oportunidades para el futuro.

* Trabajo de Grado de investigación

** Facultad de ingeniería Fisicomecánicas. Escuela de ingenierías eléctrica, electrónica y de telecomunicaciones. Ingeniería electrónica. Director: Homero Ortega Boada. Doctor en ciencias de la ingeniería, radio comunicaciones. Codirector: Nelson Enrique Trillos León. Ingeniero electrónico.

Abstract

Title: The Design and Implementation of a Bot for Handling Inquiries on Capstone Design and Undergraduate Theses in the E3T Programs*

Author(s): Juan Nicolas Mora Bernal and Davidson Duvan Leal Mogollón**

Key Words: Chatbot, virtual agent, artificial intelligence, FAQ chatbot

Description: In response to the growing need for efficient virtual tools in the educational field, this project presents an innovative solution: the implementation of chatbots in education. With the aim of improving the educational experience of students and teachers, automating the process of answering repetitive subject-related questions. To achieve this, a chatbot was developed and implemented, trained with questions collected over two different semesters. The chatbot was built using the Dialogflow CX platform and a Python code was implemented to update the chatbot's questions and answers based on information contained in a Google Sheets document. This facilitates the teacher's use of the chatbot without the need for advanced technology skills. Additionally, a web application was created to collect user questions with the purpose of adding a continuous improvement functionality to the implemented chatbot. This will allow for its constant evolution and improvement over time. The implementation was tested by senior students, who expressed high satisfaction with its performance and found it to be a useful tool for their educational process. It is expected to have a positive impact on the development of the Thesis I course and generate interest in these types of technological tools that provide many opportunities for the future.

* Research degree work

**College of Physicomechanical Engineering. School of Electrical, Electronics and Telecommunications Engineering. Electronic Engineering Program. Director: Homero Ortega Boada. PhD in Engineering Sciences, Radio Communications. Codirector: Nelson Enrique Trillos León. Electronic Engineer.

Introducción

La pandemia cambió la forma en que aprendemos, enseñamos y nos comunicamos. La educación en línea se convirtió en la norma, y la demanda de herramientas virtuales aumentó significativamente. Sin embargo, muchas de estas herramientas no ofrecen una experiencia de usuario satisfactoria y limitan el potencial de la tecnología. ¿Qué pasaría si pudieras tener una herramienta que permitiera una comunicación eficiente y mejorara la experiencia educativa? Los Chatbots son una solución que está revolucionando el mundo de los negocios y de internet, y ahora también puede ser parte de la educación.

Este proyecto busca apoyar a los estudiantes y profesores comprometidos en la actividad de Capstone Design, una estrategia de acreditación ABET que promueve el desarrollo de habilidades clave como el trabajo en equipo, la resolución de problemas y la comunicación en proyectos de diseño e ingeniería a gran escala. Sin embargo, la complejidad del proceso de Capstone Design puede generar dificultades para los estudiantes, quienes requieren una orientación constante, y debido a que solo hay un docente para la asignatura de Trabajo de Grado I, este no puede dar respuesta a todos de manera rápida y se ha venido notando que muchas de las preguntas son repetitivas. Para mejorar la comunicación entre estudiantes y docentes, se propone la implementación de un Chatbot en la asignatura de Trabajo de Grado I, que permita a los estudiantes resolver sus dudas en cualquier momento y fomentar su participación en el curso. Además, el Chatbot puede aumentar la eficiencia y liberar tiempo para los docentes, quienes podrán enfocarse en tareas de mayor valor agregado.

Con este proyecto, se espera contribuir a mejorar la experiencia educativa de los estudiantes y generar un uso más eficiente del recurso humano. Asimismo, se busca fortalecer la

investigación en el campo de la educación y mejorar la formación de los futuros profesionales de la E3T.

1. Objetivos

1.1 Objetivo General

Diseñar e implementar un BOT, en forma de chat de texto que oriente a los estudiantes cuando tengan dudas respecto a las diferentes etapas del Proyecto de Trabajo de Grado y/o Capstone Design.

1.2 Objetivos Específicos

- Seleccionar la plataforma a usar. Esto basado en un estudio de las plataformas que se está utilizando actualmente para bots en el grupo de investigación RadioGis Research Group, así como otras posibilidades.
- Diseñar el flujo conversacional del Bot, basado en las bases de datos de preguntas existentes que han sido recopiladas.
- Implementar el Bot en la página web o plataforma seleccionada.
- Realizar pruebas de funcionamiento del Bot y analizar su funcionamiento.
- Explicar un método para que en un futuro se puedan editar las conversaciones del Bot.

2. Marco de referencia

En este capítulo se presentarán algunos conceptos básicos esenciales para comprender el desarrollo del proyecto. Abordaremos temas relacionados con la inteligencia artificial, incluyendo los conceptos comúnmente utilizados en la creación de un chatbot o asistente virtual en la plataforma seleccionada, así como otros conceptos importantes que son relevantes en el desarrollo del proyecto. El objetivo es proporcionar una base sólida para el resto del trabajo y facilitar la comprensión de los temas abordados.

2.1 Inteligencia artificial

La inteligencia artificial (IA) se refiere a la capacidad de las máquinas para imitar las capacidades humanas en áreas como el razonamiento, el aprendizaje y la toma de decisiones. Es el estudio y diseño de agentes inteligentes que pueden aprender, razonar y resolver problemas de manera autónoma. (Russell & Norvig, 1994)

2.1.1 Machine Learning

El Machine Learning es un subcampo de la inteligencia artificial, este concepto nace a raíz de que las maquinas pueden aprender e interpretar a partir de un conjunto de datos, esto lo pueden hacer sin necesidad de programar explícitamente a la máquina para realizar una tarea específica. (Biswas, 2018)

Existen diferentes tipos de sistemas de Machine learning, a continuación, revisaremos los tres tipos de aprendizaje automático: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

2.1.2 Aprendizaje supervisado

El aprendizaje supervisado es un método de aprendizaje automático donde el sistema busca aprender un modelo capaz de realizar predicciones precisas a partir de un conjunto de datos de entrenamiento previamente etiquetados. En otras palabras, el modelo se entrena utilizando datos con una etiqueta predefinida que indica lo que se espera de cada dato. La finalidad es que el modelo sea capaz de predecir la etiqueta correcta para nuevos datos no vistos previamente. (Vahid Mirjalili, 2020)

2.1.3 Aprendizaje no supervisado

El aprendizaje no supervisado es una técnica de aprendizaje automático en la que se utilizan datos de entrenamiento que no están etiquetados. A diferencia del aprendizaje supervisado, el sistema no tiene un objetivo específico al que llegar y por lo tanto no cuenta con un maestro. (Geron, 2017) En su lugar, el objetivo es descubrir patrones, estructuras y relaciones entre los datos, agrupándolos en diferentes categorías o segmentos. Esto puede ayudar a los investigadores a obtener una mejor comprensión de los datos y a tomar decisiones basadas en patrones y tendencias descubiertas en ellos. (Vahid Mirjalili, 2020)

2.1.4 Aprendizaje por refuerzo

El aprendizaje por refuerzo es un enfoque de aprendizaje automático en el que un sistema interactúa con su entorno para aprender a tomar decisiones que maximicen una recompensa a medida que pasa el tiempo. A través de la retroalimentación, que se conoce como recompensa, el sistema aprende a adaptarse al entorno y mejorar su rendimiento. Este aprendizaje puede mejorar a partir de prueba y error, y es utilizado en aplicaciones que requieran una toma de decisiones en función del entorno. (Vahid Mirjalili, 2020)

2.1.5 Redes neuronales artificiales

Las redes neuronales artificiales se inspiran en la estructura y funcionamiento de las neuronas que se encuentran en el cerebro humano. Su objetivo es imitar este comportamiento neuronal para procesar datos y tomar decisiones. (Biswas, 2018)

2.1.6 Deep learning

Cuando hablamos de redes neuronales artificiales, nos referimos a que estas pueden tener muchas capas de neuronas y formar redes muy complejas. A medida que aumenta la complejidad de estas redes, podemos comenzar a hablar de Deep learning, que es un tipo de aprendizaje que busca entrenar redes neuronales artificiales en la resolución de problemas complejos. (Biswas, 2018)

2.1.7 Procesamiento de lenguaje natural (NLP)

El procesamiento de lenguaje natural busca que las maquinas entiendan el lenguaje que utilizamos los seres humanos, por lo que busca que las maquinas además de entender también puedan interpretar y generar lenguaje. Este se puede utilizar para muchas aplicaciones, especialmente para análisis, clasificación y generación de texto. (Arumugam & Shanmugamani, 2018)

2.1.8 Medidas de rendimiento

Al enfrentar un problema de clasificación, es importante evaluar el rendimiento del modelo en términos de las predicciones correctas con respecto al total de predicciones realizadas. En proyectos de este tipo, se utiliza comúnmente una matriz de confusión multiclase para tener una visión general del rendimiento del modelo, ya que se puede ver cada pregunta como una clase. En el caso de este proyecto, se utiliza dicha matriz para evaluar el desempeño del chatbot al responder preguntas de los usuarios, lo que permite determinar si el chatbot se equivoca o no en una prueba

realizada con algunos usuarios. A partir de esta matriz de confusión, se pueden calcular varias métricas de precisión, tales como la exactitud, precisión, *Recall* y *F1-score*. Más adelante en este trabajo, se profundizará sobre el uso de estas métricas y su interpretación en el contexto del proyecto. (Pandey et al., 2020)

2.1.8.1 Exactitud. Es una medida que indica la proporción de predicciones correctas, es decir, la cantidad de verdaderos positivos (TP) y verdaderos negativos (TN) obtenidos en relación con todas las predicciones realizadas. (Lopez Cubides, 2022)

$$Exactitud = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Donde TP hace referencia a los verdaderos positivos, TN a los verdaderos negativos, FP a los falsos positivos y FN a los falsos negativos.

2.1.8.2 Precisión. La precisión se define como la proporción de muestras predichas correctamente para una clase en particular (verdaderos positivos) con respecto al total de muestras predichas para esa clase, incluyendo tanto verdaderos positivos como falsos positivos. (Pandey et al., 2020)

$$Precisión = \frac{TP}{TP+FP} \quad (2)$$

2.1.8.3 Recall. Es una medida que indica la relación entre los verdaderos positivos y el número total de muestras que realmente pertenecen a esa clase. En otras palabras, el Recall mide la capacidad del modelo para identificar correctamente todas las muestras que corresponden a una clase en particular, incluyendo los verdaderos positivos y los falsos negativos. Un valor alto de Recall indica que el modelo clasifica correctamente la mayoría de las muestras de esa clase, minimizando así los errores de clasificación en otras clases. (Pandey et al., 2020)

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

2.1.8.4 F1-score. Es una medida que combina tanto la precisión como el Recall y busca un balance entre ellas. Es una media armónica ponderada entre ambos y por lo general se escoge el modelo que tenga mayor esta medida. (Pandey et al., 2020)

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Ahora pasaremos a revisar conceptos relacionados con los chatbots, un tipo de aplicación de inteligencia artificial que permite la comunicación automatizada con los usuarios.

2.2 Chatbot

Los Chatbot, como su nombre lo indica son unos bots o robots que se comunican por medio de un chat con el usuario, usualmente se comunican por medio de texto o de voz, son agentes conversacionales inteligentes que se pueden utilizar para manejar tareas que sean repetitivas y que se quieran automatizar, por ello es por lo que son muy usados hoy en día en muchas áreas, especialmente en el servicio al cliente o atención al usuario para hacer compras, resolver dudas o realizar tareas. (Peyton & Unnikrishnan, 2023)

2.2.1 Conceptos generales de los chatbots en Dialogflow CX

A continuación, se tratan conceptos importantes para que el lector pueda entender el diseño y la implementación del proyecto que se realizó en la plataforma escogida más adelante.

2.2.1.1 Agents. Un agente en Dialogflow CX es un agente virtual o como vimos anteriormente un chatbot que tiene conversaciones con los usuarios finales. Este módulo está diseñado para entender el lenguaje humano y está preparado para manejar las conversaciones configuradas previamente. (Google Cloud, 2022a)

2.2.1.2 Flows. Los flujos o flows en Dialogflow CX representan los diferentes caminos que puede tomar una conversación con el usuario final. Cada flujo está diseñado para guiar al usuario a través de una secuencia de pasos específicos para lograr un objetivo en particular. Los

flujos se organizan en árboles jerárquicos, y cada nodo en el árbol representa una página o punto de decisión en la conversación. De esta manera, los flujos permiten crear conversaciones complejas y estructuradas que se ajusten a las necesidades específicas del usuario y del agente virtual. (Google Cloud, 2022a)

2.2.1.3 Pages. Las pages o páginas en Dialogflow CX son estados en un flujo de conversación que permiten que el agente virtual ejecute diferentes acciones y realice transiciones a otras páginas dentro del mismo flujo. Cada página tiene su propio conjunto de acciones y puede contener diferentes respuestas para los usuarios finales. En resumen, las páginas en Dialogflow CX funcionan como los estados en una máquina de estados y permiten que el flujo de conversación entre el usuario y el agente se divida en diferentes secciones, cada una con acciones específicas y transiciones a otras páginas o flujos. En otras palabras, el usuario navega a través del flujo de conversación a medida que va pasando por las diferentes páginas del mismo. (Google Cloud, 2022a)

2.2.1.4 Routes. En Dialogflow CX, las rutas (o "Routes" en inglés) son un componente clave para definir el flujo de una conversación en una página. Las rutas se configuran dentro de una página específica y se les asigna una condición o *intent* para que el flujo de la conversación ejecute las acciones correspondientes. Entre estas acciones se encuentran la generación de respuestas mediante el *fulfillment*, el uso de *webhooks* y la transición a otras páginas, entre otras posibles acciones. En resumen, las rutas son el lugar donde se definen las acciones que se ejecutan en respuesta a las solicitudes de los usuarios finales durante una conversación. (Google Cloud, 2022a)

2.2.1.5 Intents. En el contexto de Dialogflow CX, las intenciones o intents se refieren a lo que el usuario busca o desea. Es decir, cuando un usuario realiza una pregunta, el agente debe

asociar esa pregunta a un intent existente para poder dar una respuesta adecuada. Los intents se crean a partir de frases de entrenamiento o *training phrases*, que son sinónimos o formas similares de realizar una misma pregunta. Al crear un intent, se busca agregar suficientes frases de entrenamiento para que el agente pueda relacionar la pregunta del usuario con el intent correspondiente y proporcionarle la respuesta correcta. (Google Cloud, 2022a)

2.2.1.6 Webhook. El webhook es un servicio proporcionado por Dialogflow CX que permite a los desarrolladores extraer información de las conversaciones o sesiones con los usuarios y enviarla a un servicio web externo para su procesamiento. Esta información puede ser validada o utilizada para realizar acciones específicas. En resumen, el webhook es una herramienta importante que ayuda a los agentes virtuales a interactuar con servicios externos y mejorar la capacidad de procesamiento y respuesta de la aplicación. (Google Cloud, 2022a)

2.2.1.7 Integraciones. Las integraciones en Dialogflow CX se refieren al apartado que permite conectar el agente virtual con otras plataformas. Esta funcionalidad permite al agente ser agregado a diferentes plataformas, como redes sociales, aplicaciones de mensajería, y otros servicios conocidos. Es importante destacar la integración para páginas web, ya que es una de las más utilizadas. De esta manera, se puede integrar el agente virtual en una página web para que los usuarios puedan interactuar con él directamente desde la página. (Google Cloud, 2022a)

2.3 Google Cloud Platform

Google Cloud Platform (GCP) es una plataforma que ofrece servicios de infraestructura en la nube y herramientas para desarrollar y administrar aplicaciones. En el marco del proyecto, se utiliza GCP para habilitar y acceder a la Interfaz de Programación de Aplicaciones (API) de Google Sheets y Dialogflow, lo que permite la integración y la interacción entre el agente de Dialogflow y otras plataformas. Es importante tener conocimientos básicos de GCP para poder

crear credenciales y permisos necesarios para acceder y llamar a las API desde otros entornos de desarrollo. (Google Cloud, 2023)

2.4 Aplicaciones web

Se denomina Aplicación Web (webApp) a una aplicación que se ejecuta dentro de un navegador web y corre en la nube, permitiendo que los usuarios accedan a ella desde cualquier dispositivo con acceso a internet. (Google, 2022)

2.4.1 Apps Script

La plataforma de Apps Script permite crear programas por medio de una secuencia de comandos. Si una secuencia de comandos contiene una función doGet(e) o doPost(e) y muestra un objeto HtmlOutput de HTML o un objeto de servicio de contenido TextOutput, puede ser publicada como una aplicación web. Donde el método GET se utiliza para solicitar datos de un recurso específico, mientras que el método POST se utiliza para enviar datos al servidor para su procesamiento. (Google, 2022)

3. Marco de referencia

La presente sección del trabajo tiene como objetivo presentar un marco de referencia que sirva de base para el desarrollo del proyecto. En este capítulo se expondrán diversos proyectos en los que se han utilizado chatbots como herramienta de interacción con los usuarios, con el fin de analizar las diferentes soluciones que han sido implementadas, las herramientas que han sido utilizadas, teniendo en cuenta que se busca un enfoque de chatbot educativo. Asimismo, se revisarán las investigaciones más recientes en el ámbito de los chatbots, con el fin de conocer las tendencias actuales en este campo. Por último, se hablará del trabajo previo realizado en el grupo

RadioGis, que sirvió de inspiración para el desarrollo del presente proyecto. Todo lo anterior permitirá establecer un marco de referencia sólido y actualizado para el proyecto en cuestión.

3.1 Chatbot como una herramienta de referencia virtual en una biblioteca

En la biblioteca de la Universidad Estatal de San José en Estados Unidos, se desarrolló un chatbot con un enfoque educativo para ayudar a los estudiantes en la búsqueda de información y recursos de la biblioteca. El chatbot se implementó en la página web de la biblioteca y se eligió la plataforma de desarrollo de Dialogflow debido a su capacidad para integrarse con múltiples plataformas y páginas web. El objetivo del chatbot era proporcionar a los estudiantes herramientas adicionales durante la pandemia. Después de un año de uso, el chatbot demostró resultados positivos al ser utilizado por muchos estudiantes, lo que contribuyó a disminuir la cantidad de preguntas que llegaban a los bibliotecarios y permitió que estos se enfocaran en problemas más complejos. (Rodriguez & Mune, 2022)

3.2 Artículo que compara el desempeño de diferentes plataformas para chatbots

En este artículo se llevó a cabo una comparación del desempeño de diferentes plataformas utilizadas en el desarrollo de chatbots, incluyendo a Dialogflow, IBM Watson Assistant, Microsoft Bot Framework y RASA. Se comparó su rendimiento con el modelo de lenguaje de procesamiento de texto BERT para responder preguntas frecuentes de los estudiantes, lo que hace que este enfoque no solo sea una comparación, sino también educativo. Se concluye que BERT supera a todas las demás plataformas, aunque estas últimas aún son una opción viable para el desarrollo de chatbots siempre y cuando las preguntas no sean extremadamente complejas. Además, se destaca que la plataforma de Dialogflow es fácil de usar y proporciona buenos resultados. (Peyton & Unnikrishnan, 2023)

Sin embargo, la selección de una plataforma dependerá de las necesidades específicas del proyecto y de los recursos disponibles. (Abdellatif et al., 2022)

3.3 Discusión sobre el uso de chatbots en la educación en la actualidad

Este artículo proporciona una visión clara del estado actual y los desafíos que enfrentan los chatbots en el ámbito educativo a través de un análisis exhaustivo de la literatura existente. Los autores identifican tres categorías principales en las que se utilizan los chatbots: apoyo a la enseñanza, evaluación y comunicación. Entre los beneficios se destacan la retroalimentación personalizada y rápida, la promoción de la participación del estudiante y la reducción de la carga de trabajo de los docentes. Sin embargo, también se identifican desafíos importantes como la falta de inteligencia emocional y la privacidad y seguridad de los datos. (Wollny et al., 2021)

En conclusión, se señala que el futuro de los chatbots en la educación es prometedor, pero aún existen desafíos importantes que deben ser abordados para aprovechar todo su potencial. Además, se destaca que los chatbots en otros campos, como los negocios, son más investigados en comparación con los chatbots educativos. (Wollny et al., 2021)

3.4 Trabajo realizado en el grupo de investigación RadioGis

El grupo de investigación RadioGis de la Universidad Industrial de Santander desarrolló un chatbot educativo con el objetivo de fomentar la curiosidad de los niños que viven cerca del Páramo de Santurbán, un ecosistema protegido. El chatbot permitía a los niños aprender sobre el Páramo y las nuevas tecnologías, como los chatbots. (Lopez Cubides, 2022) Este trabajo inspiró a la universidad a seguir investigando en el área de los chatbots y llevarlo a un enfoque diferente, resolviendo dudas de los estudiantes para mejorar los procesos.

4. Requerimientos y restricciones de diseño

En este capítulo, se describen los requerimientos y restricciones que fueron tomados en cuenta desde el inicio del proyecto, así como aquellos que surgieron durante la fase de planificación. Es fundamental tener en cuenta estos aspectos, ya que pueden limitar o permitir diferentes caminos para la solución del problema planteado en el proyecto. Asimismo, cumplir con estos requerimientos y restricciones es un objetivo importante para el éxito del proyecto.

- El chatbot debe ser de texto.
- Debe entrenarse a partir de las preguntas que han venido formulando los estudiantes mientras cursan la asignatura de trabajo de grado en los semestres anteriores.
- Debe tener la capacidad de interpretar preguntas, aunque se formulen con expresiones diferentes.
- Debe brindarle al profesor la posibilidad de incrementar o editar las preguntas que puede responder el chatbot de una manera sencilla, a medida que nuevas incógnitas van apareciendo en el tiempo.
- Debe apoyarse en los chatbots que se han desarrollado en el grupo RadioGis de la E3T.

5. Selección de la plataforma

Un elemento clave en el proyecto es encontrar una plataforma para diseñar e implementar el chatbot que nos ayude a alcanzar los objetivos planteados. En este capítulo se realizó un análisis de algunas plataformas utilizadas actualmente y revisadas en el grupo de investigación, en el que analizaremos las principales características, como el diseño de su interfaz de trabajo, la posibilidad

de agregar funcionalidades adicionales, su capacidad de integración con otras plataformas y su costo, entre otras.

Después de investigar las plataformas revisadas en el grupo de investigación y algunas de las utilizadas actualmente, se seleccionaron las siguientes para su análisis: Dialogflow CX, Amazon Lex, IBM Watson Assistant y Microsoft Azure Bot Service.

5.1 Dialogflow CX

Dialogflow CX es una plataforma de Google que permite diseñar y desarrollar agentes de conversación utilizando máquinas de estado. Esta herramienta tiene como objetivo mejorar la experiencia del usuario final y facilitar el flujo de trabajo de desarrollo, ya que proporciona un buen control sobre la conversación (Google Cloud, 2022a).

5.1.1 Características

- La documentación es bastante intuitiva y se puede revisar mediante una guía que pone a nuestra disposición Google.
- Permite entradas y salidas en texto y en audio.
- Está enfocada para flujos conversacionales que puedan llegar a ser complejos y da a los desarrolladores una interfaz gráfica interactiva que permite editar de forma más sencilla los parámetros que componen la construcción del agente, dando así mucho margen para la personalización y el control de las respuestas.
- Permite integrar los agentes en páginas web o diferentes canales de mensajería.
- Es una plataforma ideal para trabajar en proyectos tanto a pequeña escala, como a gran escala.

5.1.2 Costos

Dialogflow CX proporciona un crédito de 600 dólares que se pueden gastar durante un año la primera vez que accedemos al servicio (Google Cloud, 2022a). Luego de esto, las solicitudes tendrán costo y las podremos ver en la tabla 1 que se muestra a continuación.

Tabla 1

Precios aproximados para la plataforma Dialogflow CX

Concepto	Total (Pesos colombianos / interacción)
Texto (incluidas todas las solicitudes DetectIntent, StreamingDetectIntent, FulfillIntent, AnalyzeContent y StreamingAnalyzeContent que no contienen audio)	\$34.51
Entrada y salida de audio (reconocimiento de voz, conversión de texto a voz, STT, síntesis de voz, conversión de texto a voz, TTS, telefonía)	\$295.84
Solicitudes durante el diseño Por ejemplo, llamadas para crear o actualizar un agente. También se incluyen las solicitudes de tiempo de ejecución de audio y texto enviadas por el simulador de la consola de Dialogflow.	\$0
Otras solicitudes durante las sesiones Por ejemplo, para configurar u obtener entidades de la sesión.	\$0

Nota. Esta tabla muestra el costo por cada iteración para la plataforma Dialogflow CX. Adaptado de (Google Cloud, 2022b).

5.2 Amazon Lex

La plataforma de Amazon Lex ofrece a los desarrolladores la posibilidad de crear agentes virtuales de voz o texto que integran inteligencia artificial, permitiéndoles interactuar con los usuarios de manera natural y personalizada. Estos agentes automatizan tareas como la atención al cliente o el procesamiento de pedidos, lo que les brinda a los desarrolladores la oportunidad de centrarse en el diseño y programación de la lógica de negocio sin preocuparse por el hardware y la infraestructura subyacentes (Amazon, 2022a).

5.2.1 Características

- En su página web, Amazon Lex ofrece documentación y tutoriales detallados para ayudarte a comenzar a utilizar el servicio de manera efectiva.
- Permite integrar el chatbot fácilmente con las aplicaciones de Amazon Web Services y de terceros.
- Amazon Lex está diseñada para ayudarte a crear chatbots y aplicaciones de voz que puedan ejecutar y automatizar tareas de manera eficiente.

5.2.2 Costos

Con Amazon Lex, puedes obtener 10000 solicitudes de texto gratuitas cada mes durante un año (Amazon, 2022a). Después de este período, se aplicará un costo por iteración por cada solicitud adicional de texto. Puedes ver un ejemplo de los precios en la tabla 2 a continuación.

Tabla 2

Ejemplo de precios para la plataforma Amazon Lex

Solicitudes de entrada	Costo por solicitud	Número de solicitudes	Total
8,000 solicitudes de voz	0.004 USD	8,000 solicitudes	32.00 USD
2,000 solicitudes de texto	0.00075 USD	2,000 solicitudes	1.50 USD
Cargos totales de Amazon Lex para el mes			33.50 USD

Nota: Esta tabla muestra un ejemplo del costo en dólares (USD) por un número de solicitudes. La información fue adaptada de: (Amazon, 2022b).

5.3 IBM Watson Assistant

La empresa de tecnología y consultoría IBM también nos ofrece una plataforma para crear chatbots o asistentes, que permite utilizar una interfaz gráfica simple, en la cual se puede diseñar el flujo conversacional y proporciona herramientas de inteligencia artificial y procesamiento del lenguaje natural para cumplir con este objetivo (IBM, 2022).

5.3.1 Características

- La documentación está disponible en su página web principal, donde se cuenta con videos, documentos y tutoriales.
- Permite entradas y salidas en texto y en audio.
- No se necesita ser experto en un lenguaje de programación, aunque es compatible con Python y JavaScript.
- Permite integrar el chatbot en las redes sociales o como widget en las páginas web.
- Puede utilizar bases de datos grandes de las que el chatbot pueda extraer respuestas a las preguntas.
- Está orientado para satisfacer las necesidades de los clientes de una empresa.

5.3.2 Costos

La plataforma Watson Assistant cuenta con un plan gratuito que ofrece la oportunidad de implementar chatbots y utilizar un número determinado de llamadas a la API. Sin embargo, es importante tener en cuenta que hay un límite en cuanto a la cantidad de llamadas permitidas. Para superar esta limitación, se recomienda considerar la adquisición de un plan Premium, aunque esto puede conllevar un costo elevado, debido a que la empresa se enfoca en atender a empresas. Para obtener más información sobre los costos y los planes Premium, se sugiere entrar en contacto con el departamento de ventas de la empresa (IBM, 2022).

5.4 Microsoft Azure Bot Service

Microsoft también pone a nuestra disposición una plataforma para crear experiencias conversacionales en un entorno de desarrollo integrado para la creación de bots, además de utilizar la integración con Power Virtual Agents, una plataforma que le permite a los desarrolladores crear chatbots sin necesidad de tener conocimiento en código de programación, ya que cuenta con una interfaz gráfica colaborativa de bajo código (Microsoft, 2022a).

5.4.1 Características

- Cuenta con una documentación muy completa.
- No se necesita ser experto en un lenguaje de programación para desarrollar los chatbots.
- Permite integrar el chatbot fácilmente en las aplicaciones de Microsoft.
- Cuenta con múltiples certificaciones y promete seguridad y privacidad de los datos.

5.4.2 Costos

Según Azure Bot Service, se ofrece una prueba gratuita que permite hacer uso de 200 dólares durante el primer mes de uso. Además, para los meses subsiguientes, solo se cobra si se

supera la cantidad de 10000 mensajes gratuitos cada mes. En caso de superar este límite, el valor a pagar será de 0.5 dólares por cada 1000 mensajes adicionales (Microsoft, 2022b).

5.5 Criterio de selección

Watson Assistant es una plataforma enfocada en trabajar con grandes empresas, pero a pesar de tener buenas herramientas para buscar información en una base de datos, sus costos son elevados y no es una opción viable para nuestro proyecto debido a que nuestro banco de preguntas es pequeño.

Azure Bot Service se enfoca en trabajar con empresas, ya que se utiliza en aplicaciones que requieren integrar el servicio de llamadas en los agentes. Se puede integrar con aplicaciones de Microsoft, como por ejemplo Skype, entre otras. Además, cuenta con un sistema de protección para el tratamiento seguro de datos. Sin embargo, como el proyecto tiene un enfoque educativo, no necesitamos este tipo de herramientas, por lo que esta opción no es viable.

Amazon Lex busca facilitar la automatización de acciones mediante el uso de asistentes virtuales. Además, las empresas pueden ofrecer servicios más ágiles gracias a los bots, ya que integra muchos de los servicios de Amazon. Sin embargo, para el proyecto en cuestión, buscamos algo que esté más enfocado en la parte conversacional y tenga mayor compatibilidad al integrarse en páginas web. Por lo tanto, aunque sea la opción más económica, no la consideramos la mejor.

Por último, Dialogflow CX es una plataforma que da gran importancia a mejorar la experiencia del usuario. Esto les brinda a los desarrolladores varias opciones para perfeccionar las conversaciones. Aunque no es la opción más económica, va enfocada en lo que buscamos y tiene buena compatibilidad al integrarse en páginas web, además en el grupo de investigación RadioGis se ha actualizado en anteriores proyectos y ha tenido buenos resultados.

Por lo tanto, se ha determinado que Dialogflow CX es la plataforma seleccionada debido a que tiene el enfoque que buscamos y ha sido utilizada con éxito en proyectos de chatbots educativos en el grupo de investigación RadioGis.

6. Recopilación de preguntas

Antes de comenzar con el diseño del chatbot, es importante recopilar preguntas y respuestas para que el chatbot pueda ser entrenado. Para esto, se realizó una recopilación de preguntas formuladas por los estudiantes en diferentes grupos de WhatsApp, los cuales estaban destinados para la asignatura "Trabajo de Grado I". Además, se consideró un documento de preguntas frecuentes elaborado por el docente de la asignatura. En un esfuerzo adicional para mejorar la experiencia del usuario, se agregaron algunos diálogos que hacen la conversación más amena. La lista de preguntas y respuestas recopiladas se incluye en el Apéndice A.

7. Casos de uso

El principal objetivo de este proyecto es el desarrollo de un sistema de chat conversacional para estudiantes, el cual pueda ser actualizado en cualquier momento. Para lograr esto, se plantean dos casos de uso: el rol de estudiante y el rol de desarrollador.

7.1 Rol estudiante

El rol de estudiante hace referencia a los usuarios principales del chatbot, quienes son los estudiantes que cursan la asignatura de trabajo de grado 1. Estos estudiantes podrán tener una experiencia conversacional con el chatbot y hacerle preguntas sobre la asignatura.

7.2 Rol desarrollador

Por otro lado, el rol de desarrollador está enfocado en profesores o estudiantes que tengan permiso para actualizar el chatbot. Estos usuarios tendrán acceso a las credenciales necesarias para editar y agregar preguntas y respuestas al chatbot, utilizando el método que se explica en el capítulo 12. De esta manera, se busca crear una herramienta constantemente actualizada que pueda evolucionar en el futuro.

8. Diseño

En este capítulo se presentará el diseño del chatbot, cuyo objetivo es proporcionar a los estudiantes que cursan la asignatura de Trabajo de Grado 1 una herramienta conversacional que les permita obtener información relevante sobre la asignatura. Es fundamental establecer un diseño adecuado para el chatbot, definiendo su enfoque y creando un flujo conversacional coherente que se ajuste a las necesidades de los usuarios y les brinde una experiencia satisfactoria. En este sentido, se expondrán las decisiones tomadas y se explicará el funcionamiento buscado para el chatbot, así como el diseño de una herramienta para que el profesor o una persona autorizada pueda actualizar el chatbot sin tener conocimientos avanzados en la plataforma utilizada.

8.1 Diseño de la arquitectura del chatbot

Para diseñar el chatbot, se tuvo en cuenta que se buscaba no solo un bot conversacional, sino también una herramienta que permitiera su actualización fácil y rápida. Por esta razón, se decidió optar por un chatbot básico, con un flujo conversacional sencillo y enfocado en la resolución de preguntas frecuentes de los estudiantes.

En esta sección, se explicará detalladamente cómo se planteó la arquitectura del chatbot y cómo se desarrolló el flujo conversacional para cumplir con los objetivos del proyecto. Además, se presentarán las decisiones tomadas y las razones detrás de ellas para brindar una comprensión completa del proceso de diseño.

8.1.1 Selección de la plataforma Dialogflow CX

En la selección de la plataforma para la implementación del chatbot, se optó por Dialogflow CX, tal como se mencionó en una etapa previa del proyecto. Esta plataforma es una herramienta muy útil que permite crear chatbots de manera sencilla, sin requerir conocimientos avanzados de programación. La interfaz gráfica que ofrece es intuitiva y se adapta muy bien a los objetivos del proyecto.

En Dialogflow CX, la lógica para construir los chatbots se basa en la creación de un flujo conversacional a través de páginas, las cuales permiten controlar toda la conversación con el usuario. En la siguiente sección, se explicará en detalle este concepto, ya que es fundamental para entender el diseño del flujo conversacional del chatbot.

8.1.2 Definición de las paginas

En Dialogflow CX, las páginas son una representación de un estado específico en el flujo conversacional del chatbot. Cada página tiene un conjunto de intents asociados que definen cómo el chatbot responderá en esa página en particular. Por ejemplo, la página inicial del chatbot podría tener un intent de saludo que permita al chatbot saludar al usuario y presentarse. En otra página posterior, el chatbot podría tener un intent específico para responder preguntas sobre un tema específico.

Las páginas están conectadas mediante rutas que definen el flujo conversacional del chatbot. Dependiendo de la página en la que se encuentre el usuario, el comportamiento del chatbot

cambiará y se adaptará a la página correspondiente. El flujo conversacional del chatbot dependerá de cómo se hayan establecido las conexiones entre las diferentes páginas y cómo se hayan definido los intents asociados a cada página.

8.1.3 Definición de los intents

En el proyecto, los intents se definen como las preguntas preestablecidas que el chatbot puede responder. Son cruciales para que la inteligencia artificial comprenda la pregunta del usuario y la relacione con el intent más adecuado. Cada intent debe contar con frases de entrenamiento para capacitar al chatbot y garantizar su efectividad. Cuando se identifica el intent, el chatbot proporciona una respuesta apropiada y relevante al usuario.

8.1.4 Definición de las routes

Las rutas son elementos esenciales dentro de cada página del chatbot, ya que permiten la transición de una página a otra en función de distintos criterios, como la detección de un intent específico o una condición establecida previamente. Asimismo, las rutas también posibilitan que el chatbot proporcione una respuesta al usuario una vez que se cumpla el intent o la condición establecidos, lo que se conoce como fulfillment. En resumen, las rutas son el medio por el cual el chatbot dirige la conversación hacia el objetivo deseado y provee las respuestas necesarias para resolver la necesidad del usuario.

8.1.5 Definición del flujo conversacional

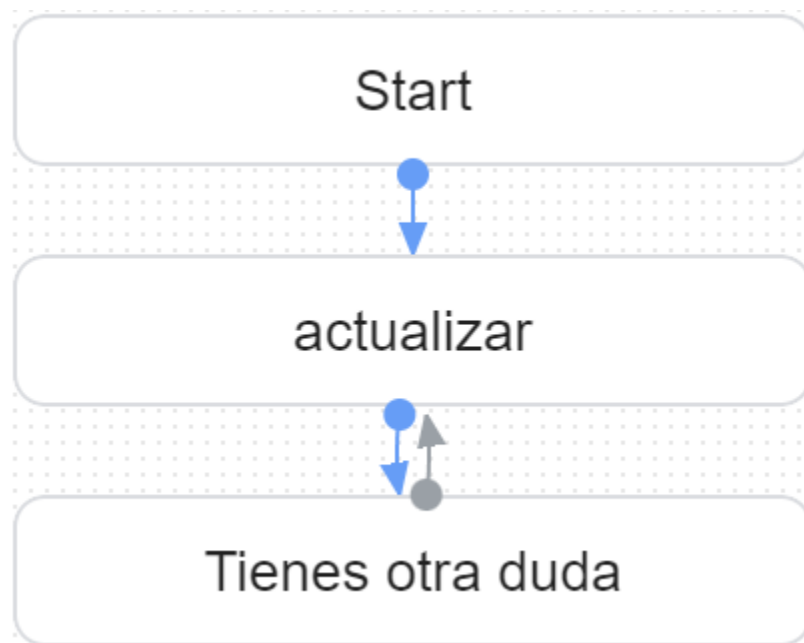
Para la definición del flujo conversacional, se buscó un enfoque sencillo pero efectivo que permitiera la actualización del chatbot en el futuro. Como resultado, se estableció un flujo con tres páginas principales: "Start", "Actualizar" y "Tienes otra duda". En la página "Start", el chatbot saluda al usuario y se presenta, por lo cual esta página solo consta de un intent, que se encargara de detectar saludos. En la página "Actualizar", es donde estarán todos los intents que sean definidos

para identificar preguntas, por lo cual esta página es la más importante del flujo, ya que aquí nos damos cuenta de que será donde debemos editar las preguntas y respuestas del chatbot, por lo tanto, esta página es la que maneja el mayor número de intents por lo que el usuario podrá en esta página hacer preguntas y obtener respuestas, lo que constituye el núcleo de la conversación. En la página "Tienes otra duda", el usuario puede solicitar más información o finalizar la conversación, por lo que en esta página solo habrá un intent que identifica si el usuario tiene intención de hacer otra pregunta. Esto permite brindar una mejor experiencia al usuario y asegurarse de que se le hayan resuelto todas sus dudas.

En la figura 1 se muestra la representación gráfica del flujo conversacional en la plataforma de Dialogflow CX:

Figura 1

Flujo conversacional planteado para el proyecto



Nota. La figura muestra el flujo conversacional elegido para el proyecto, este diseño simple pero funcional del flujo conversacional es una importante decisión que se tomó en el proyecto para

garantizar la actualización y mejora continua del chatbot, dando una mejor experiencia al usuario y asegurándose de que se le hayan resuelto todas las dudas. Captura de pantalla tomada de la plataforma Dialogflow CX.

8.1.6 Diseño del método para actualizar el chatbot

El diseño de un método eficiente para actualizar las preguntas y respuestas del chatbot fue uno de los principales objetivos del proyecto. Se buscó crear un método que permitiera al profesor o a una persona con los permisos correspondientes, modificar las preguntas y respuestas sin necesidad de tener conocimientos especializados en la plataforma de desarrollo del chatbot. Para lograrlo, se planteó la creación de una base de datos de preguntas y respuestas en un documento de Google Sheets. Posteriormente, se construyó un algoritmo en Python que se puede ejecutar desde Google Colab. Para ejecutar el algoritmo, es necesario tener acceso a los archivos de credenciales de Dialogflow CX y Google Sheets.

El funcionamiento del programa se resume en lo siguiente: al ejecutar el código, se eliminan todas las preguntas y respuestas que se encuentran en la página "Actualizar" del chatbot. Luego, el programa lee la información contenida en el documento de Google Sheets y crea las preguntas y respuestas correspondientes. De esta manera, se asegura que el chatbot tenga la información más actualizada en todo momento, permitiendo una mejor experiencia para el usuario y evitando la obsolescencia de la información proporcionada. En el capítulo de implementación, se explicará con más detalle el código utilizado en este método.

8.1.7 Integración con otras herramientas

Una vez finalizada la construcción del chatbot, se procedió a implementarlo en una página web. La página web se diseñó con características simples para permitir que los usuarios puedan

acceder al chatbot de manera práctica y sencilla. La implementación de la página web se explicará en detalle en el siguiente capítulo de implementación.

En la página "actualizar", se habilitó el webhook para enviar información de la conversación cada vez que el usuario realice una pregunta. Con el fin de analizar cómo los usuarios hacen preguntas para mejorar el chatbot en el futuro, se creó un programa en Apps Script de Google que guarda las preguntas en un documento de Google Sheets. Esta integración con otras herramientas es fundamental para evaluar el rendimiento del chatbot y hacer mejoras en base a los patrones de conversación que se identifiquen.

8.2 Diseño de la interfaz del usuario

En cuanto a la interfaz del usuario, se decidió utilizar la solución proporcionada por Dialogflow CX para la integración en páginas web, conocida como "Dialogflow Messenger". Dicha solución ofrece una interfaz básica que permite a los usuarios interactuar con el chatbot de forma sencilla y directa. Aunque se consideró la posibilidad de personalizar la interfaz, finalmente se optó por mantenerla en su versión original para asegurar su compatibilidad y facilidad de uso en diferentes plataformas y dispositivos.

8.3 Diseño de la base de conocimiento

Las preguntas y respuestas recopiladas con las que se realizó el chatbot, se pueden ver en el apéndice A.

8.3.1 Estructuración de la información

Para la estructuración de la información necesaria para el chatbot, es importante tener en cuenta que además de las preguntas y respuestas que se pueden encontrar en el apéndice A, se deben crear frases de entrenamiento para cada pregunta. Estas frases son fundamentales para entrenar al chatbot y mejorar su desempeño. Por lo cual, la información necesaria para la

estructuración de la información incluye la pregunta, la respuesta y un número adecuado de frases de entrenamiento.

8.3.2 Carga de la información en el chatbot

Una vez que se ha estructurado la información necesaria para el chatbot, se procede a cargarla en la plataforma de Dialogflow CX. La información se encuentra almacenada en un documento de Google Sheets y se puede ver en el apéndice B, este consta de tres columnas: pregunta, respuesta y frases de entrenamiento. Para agregar nuevas frases de entrenamiento, el usuario con permisos para actualizar el chatbot debe modificar directamente el documento de Google Sheets.

Para cargar la información en el chatbot, se utiliza un programa en Python que procesa los datos y los carga en la plataforma de Dialogflow CX. Este proceso garantiza una actualización ágil y sencilla de la información del chatbot en caso de cambios o actualizaciones en la base de conocimiento. Adicionalmente se tomó la decisión de implementar el programa utilizando Google Colab, esto con el fin de que se pueda ejecutar el código desde cualquier navegador web siempre y cuando se tenga los archivos de credenciales, que son dos archivos necesarios para ejecutar el código, ya que permiten realizar el proceso de autenticación, permitiendo así que se puedan hacer llamadas a la API, tanto de Dialogflow CX como de Google sheets, por lo cual la persona que quiera ejecutar el código solo necesita tener esos dos archivos descargados en su computador, ya que el código le pedirá subirlos.

8.4 Diseño de las pruebas

Para el diseño de las pruebas del chatbot, se tuvo en cuenta la investigación que sirvió de inspiración para este trabajo de grado (Lopez Cubides, 2022). Por esta razón, se decidió utilizar una matriz de confusión para cada una de las clases, las cuales en este proyecto representan las

distintas preguntas o intents. De esta manera, se pueden obtener medidas de rendimiento como la exactitud, precisión, Recall y F1-score para cada clase en particular. Además, esta matriz permite identificar las preguntas específicas en las que el chatbot presenta fallos, lo que la convierte en una excelente opción para evaluar el funcionamiento del chatbot.

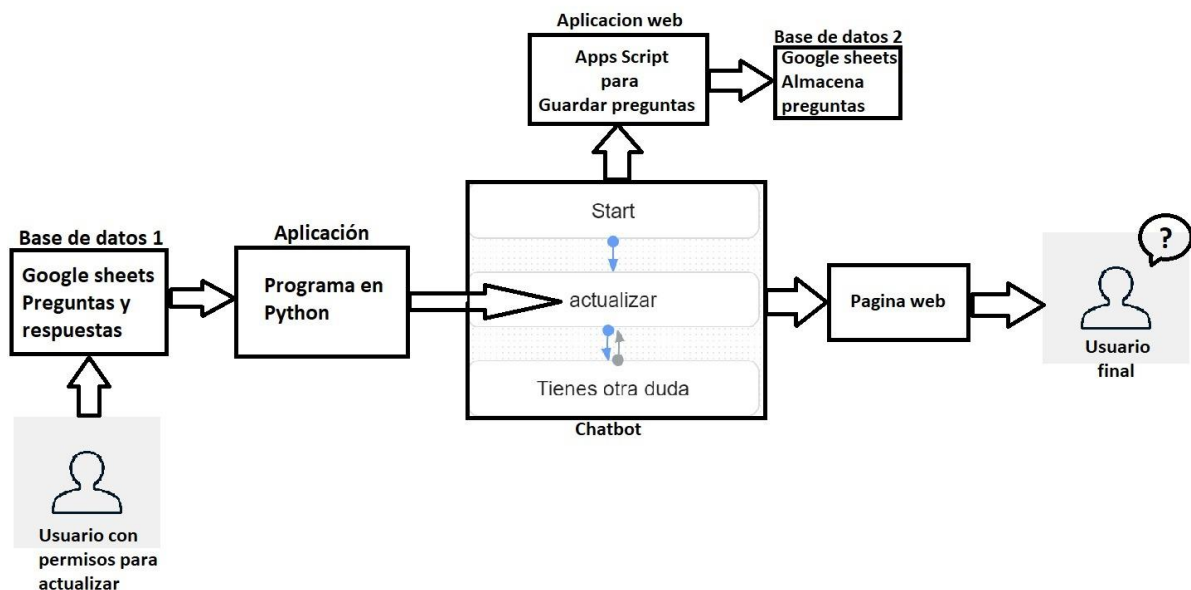
Para llevar a cabo la evaluación del chatbot, se contó con la participación de seis estudiantes de últimos semestres, quienes estuvieron dispuestos a realizar un total de 66 iteraciones cada uno. De esta forma, fue posible evaluar el desempeño del chatbot y analizar los resultados obtenidos.

8.5 Diseño de la arquitectura de la solución

El diseño de la arquitectura de solución es una fase crítica en cualquier proyecto de desarrollo de software. La creación de un diagrama que ilustre la integración de las diferentes bases de datos y aplicaciones, así como la interacción con los usuarios, permite tener una visión clara y completa del proyecto. En este sentido, la figura 2 muestra la arquitectura de la solución final del presente proyecto, proporcionando una representación visual de gran utilidad para entender el diseño completo y la dirección de la información.

Figura 2

Diagrama de arquitectura de solución



Nota. La figura muestra el diagrama de arquitectura de la solución final, que describe todo el funcionamiento del proyecto. Imagen creada por el autor para representar la arquitectura de la solución

En la figura 2 se muestra un diagrama que resume el diseño completo del proyecto. Para explicar el proceso, partimos de un usuario con permisos para actualizar el chatbot, quien modificará la base de datos 1 que contiene las preguntas y respuestas. Luego, una aplicación programada en Python leerá esta base de datos y llevará las preguntas y respuestas al chatbot. Específicamente, el programa interactúa con la página "actualizar", creando automáticamente preguntas y respuestas en la plataforma, junto con las configuraciones necesarias para cada una de ellas. Esto ahorra tiempo y esfuerzo, ya que no se necesita crear manualmente cada pregunta y respuesta en la plataforma, lo que sería un proceso muy largo y requiere conocimientos específicos.

Una vez que el chatbot cuenta con las preguntas y respuestas configuradas, ya no es necesario utilizar la base de datos 1 ni el programa en Python. El chatbot tendrá en su propia base de datos las preguntas y respuestas configuradas y estará disponible para ser utilizado por los

usuarios finales a través de una página web. Además, el chatbot enviará información a la aplicación web cuando los usuarios finales interactúan con él, lo que permite guardar las preguntas de los usuarios en la base de datos 2 y llevar un seguimiento de cómo los usuarios realizan sus preguntas. Esto permite una mejora continua del chatbot.

En resumen, este proceso garantiza un chatbot funcional y el ciclo se repetirá cada vez que el usuario con permisos para actualizar quiera hacer un cambio en la base de datos 1.

9. Implementación

En este capítulo, se detalla la implementación del chatbot en el proyecto, centrándose en los aspectos más importantes de cada fase. En primer lugar, se describe la configuración de la plataforma de Dialogflow CX, que fue la base para la creación del chatbot. Luego, se revisa en detalle el programa en Python utilizado para cargar los datos necesarios en el chatbot. También se revisará el método para guardar las preguntas que hacen los usuarios. Por último, se examina la integración del chatbot en la página web creada para el proyecto. El objetivo de este capítulo es brindar al lector una comprensión completa del proceso de implementación del chatbot en el proyecto.

9.1 Implementación en Dialogflow CX

En esta sección, se presentará al lector el proceso de implementación de las principales configuraciones en la plataforma de Dialogflow CX. Se describirán los pasos necesarios para configurar el chatbot y se dará un resumen del funcionamiento del flujo conversacional.

9.1.1 Creación y configuración de un intent

Para crear un intent en la plataforma de Dialogflow CX, debemos ir al menú de la izquierda llamado “Manage” y luego ir a la sección de “Intents”, luego podremos crear uno seleccionando la opción de “Create”, posteriormente nos aparecerá la configuración del intent como se visualiza en la figura 3.

Figura 3

Configuración de un intent

← Intent Save Cancel T Agent settings Test Agent

Intents represent something your users want to do during a conversation with your agent (for example, schedule an appointment). [Learn more](#)

Display name *

Labels ?

Description
Describe what task this intent does. For example: This intent triggers when user asks a payment question.

Training phrases
When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

Type a training phrase and press 'Enter' Add

↑ 🗑️

Nota. La figura muestra el menú de configuración de un intent. Captura de pantalla tomada de la plataforma Dialogflow CX.

Como se mencionó en el capítulo anterior, los intents son como las preguntas que el chatbot puede asociar con la base de datos. Para configurar un intent, solo son necesarios dos elementos: el "Display name", que es el nombre de la pregunta en la base de datos, y las “Training phrases” o frases de entrenamiento, que son las formas en que los usuarios pueden referirse a esa pregunta.

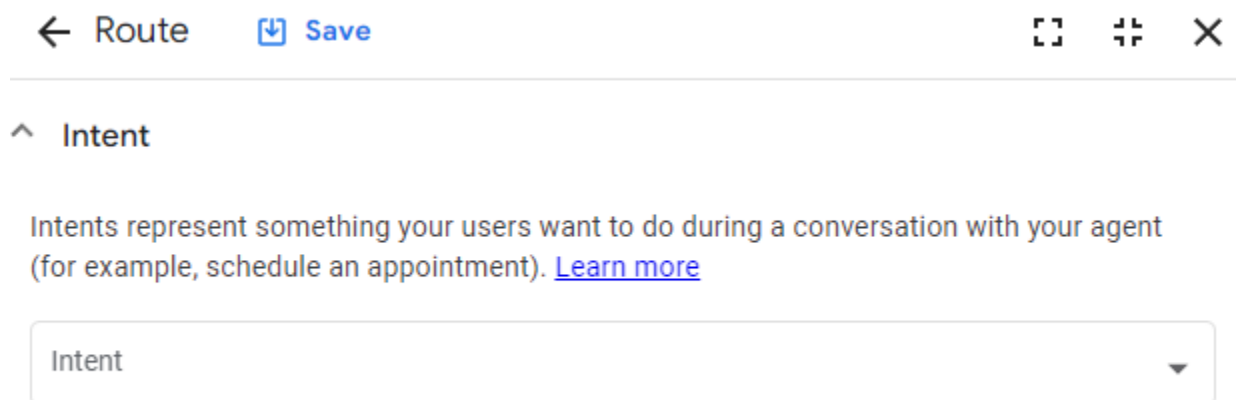
La configuración de los intents es crucial para el rendimiento del chatbot, ya que depende en gran medida de las frases de entrenamiento establecidas para cada pregunta.

9.1.2 Configuración de una Route

Para configurar una route en la plataforma de Dialogflow CX, es necesario crear una página dentro del flujo en el que se está trabajando. Una vez dentro de la página, se puede agregar una nueva route a través del menú que se presenta en la pantalla. El menú incluye varias opciones como se evidencia en la figura 5, que permiten establecer las condiciones necesarias para que la route sea activada, así como las acciones que se deben realizar después de que se active.

Figura 4

Asociar un intent a una route



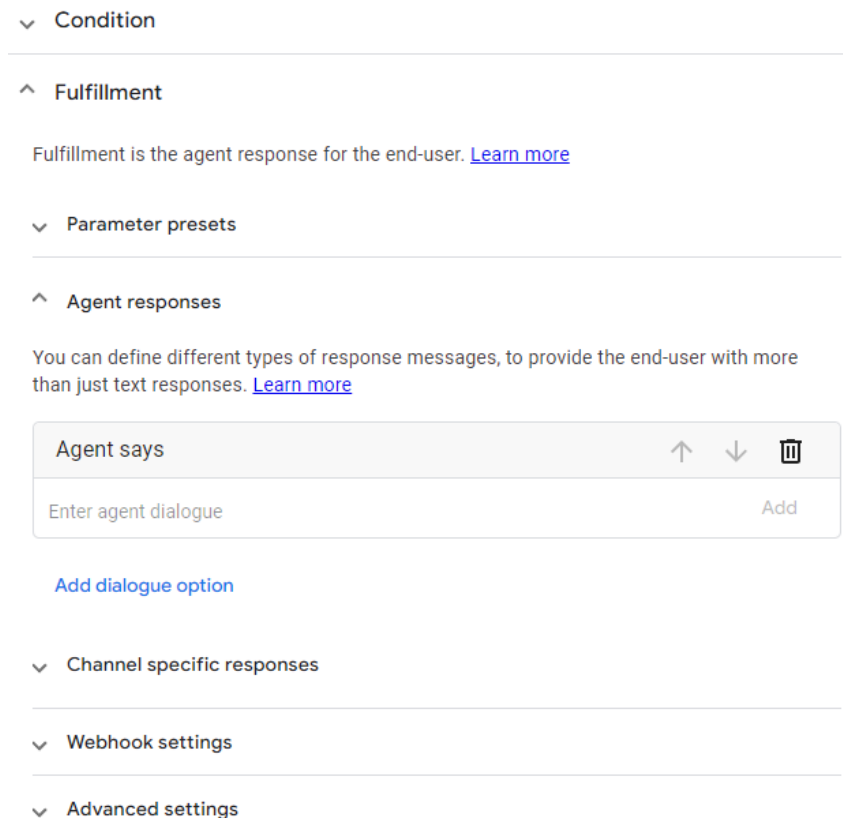
Nota. La figura muestra el menú de configuración de una route, donde la primera opción que se puede configurar es para asociar un intent a la route. Captura de pantalla tomada de la plataforma Dialogflow CX.

Para asociar un intent a una route, como se evidencia en la figura 4, debemos seleccionar la opción 'Intent' dentro de la sección de configuración de la route y elegir el intent correspondiente. Una vez asociado el intent, el flujo conversacional se dirigirá automáticamente hacia la route

configurada cuando el usuario escriba una pregunta asociada a ese intent y realizará lo que este configurado en esa route.

Figura 5

Configuración de una route



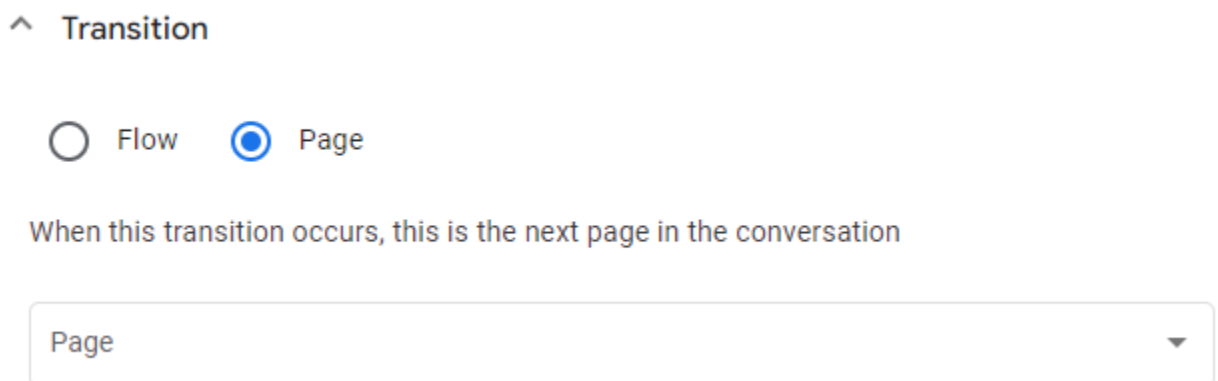
Nota. La figura muestra otra parte del menú de configuración de una route, donde se puede ver que tenemos la opción de agregar una condición, un fulfillment y configuraciones adicionales como lo son el webhook. Captura de pantalla tomada de la plataforma Dialogflow CX.

Además, una vez que el flujo de la conversación ingresa en la route, podemos agregar una condición, que comúnmente se utiliza para solicitar información adicional al usuario. Sin embargo, en este proyecto en particular no se utiliza esta opción. A continuación, encontramos la sección de fulfillment, que es la respuesta que el agente proporciona al usuario. Podemos hacer una analogía

entre el intent y la pregunta del usuario, y el fulfillment como la respuesta del chatbot para esa pregunta. Por último, podemos activar la opción de webhook para enviar la pregunta a un webservice y recopilar las preguntas que hacen los usuarios para mejorar continuamente el chatbot.

Figura 6

Asociar una transition a una route



Nota. La figura muestra la parte final del menú de configuración de una route, donde podemos agregar una transition a la route. Captura de pantalla tomada de la plataforma Dialogflow CX.

Para finalizar la configuración de una route, es importante seleccionar la ruta o transición a seguir, como se muestra en la figura 6, después de que la acción de la route haya sido ejecutada. En nuestro proyecto, en lugar de cambiar de flujo, queremos cambiar de página. Por lo tanto, para elegir la siguiente página a la que se dirigirá el flujo conversacional después de que el chatbot responda a la pregunta, simplemente abrimos el menú desplegable y seleccionamos la página correspondiente.

9.1.3 Resumen del funcionamiento

En el capítulo anterior se detalló que el flujo conversacional del chatbot consta de tres páginas. En la primera página, llamada "Start", se crea una route con un intent asociado que permita

al chatbot reconocer los saludos del usuario. Luego, el chatbot saludará y, mediante una transición, llevará al usuario a la siguiente página llamada "actualizar". En esta página, el chatbot esperará la pregunta del usuario y, si es capaz de entenderla, proporcionará una respuesta. De lo contrario, el chatbot esperará hasta que el usuario haga una pregunta que pueda asociar con las preguntas conocidas. Después de proporcionar la respuesta, el chatbot realizará la transición a la última página llamada "Tienes otra duda", donde le indicará al usuario si está satisfecho con la respuesta obtenida. Si el usuario no está satisfecho o desea hacer otra pregunta, deberá escribir algo para activar la transición hacia la página "actualizar", donde podrá intentar hacer otra pregunta. Así, el flujo conversacional permanecerá en las páginas "actualizar" y "Tienes otra duda" hasta que el usuario decida finalizar la conversación.

9.2 Programa para la carga de datos y actualización del chatbot

En esta sección, se explicará el funcionamiento del programa escrito en Python, que ha sido implementado en el entorno de Google Colab para permitir la actualización de las preguntas y respuestas que el chatbot pueda responder en la página "actualizar". Cabe destacar que, para llevar a cabo este código, es necesario contar con conocimientos en "Google Cloud Platform", ya que es una parte fundamental para la autenticación y conexión de las diferentes plataformas, lo que permite realizar interacciones con la API de manera segura y efectiva. (Google Cloud, 2023)

9.2.1 Resumen del funcionamiento del código

El objetivo del código es eliminar todo el contenido dentro de la página "actualizar". Para ello, se borra la página existente y se crea una nueva vacía con el mismo nombre. Posteriormente, se eliminan todos los intents que pertenecen a dicha página. Luego, se extrae la información del documento de Google Sheets y se utilizan las preguntas y frases de entrenamiento para crear nuevos intents. Por último, se crean las nuevas rutas a partir de los nuevos intents generados y se

configuran con la respuesta adecuada, la transición a la página "Tienes otra duda" y se activa y selecciona el webhook.

9.2.2 Explicación del código

Para entender el código de una mejor manera, este se dividió en seis partes y a continuación revisaremos cada una, además el lector puede encontrar el código completo en el apéndice C.

La primera parte es la figura 7, y es la instalación de las librerías, donde necesitamos librerías para autenticar y poder realizar llamadas a la API, como también se necesita la librería de Dialogflow CX.

Figura 7

Instalación de las librerías

```
!pip install google-auth google-api-python-client  
!pip install --upgrade google-cloud-dialogflow-cx
```

Nota. La figura muestra las librerías utilizadas. Captura de pantalla tomada del código implementado en Colab.

Luego para que se pueda ejecutar el código de la figura 8, se necesita tener los dos archivos de credenciales (estos archivos fueron generados desde "Google Cloud Platform"), y el usuario que tenga permisos de actualizar el chatbot, deberá tener los dos archivos descargados en su propio computador, uno para Dialogflow CX y otro con las credenciales de Google sheets, ya que la segunda parte del código, se le pide al usuario que suba estos dos archivos.

Figura 8

Subir los archivos credenciales

```

from google.colab import files
print("Por favor suba el archivo de credenciales de google sheets")
uploaded = files.upload()
print("Por favor suba el archivo de credenciales de dialogflow")
uploaded = files.upload()

```

Nota. La figura muestra la parte del código que le pide al usuario subir los dos archivos de credenciales. Captura de pantalla tomada del código implementado en Colab.

Luego en la siguiente sección del código, se utiliza el archivo de credenciales de Dialogflow CX para autenticarnos y permitir la interacción con la API desde el programa. En la figura 9, el objetivo es crear una nueva página llamada "actualizar", que esté vacía y reemplazar la página anterior. Como resultado, se eliminarán todas las rutas previamente definidas en esa página, y se tendrá una página vacía.

Figura 9

Eliminar lo que este en la página “actualizar”

```

#Este codigo borra todas las routes que esten en la pagina "actualizar"
from google.oauth2 import service_account
from google.cloud.dialogflowcx_v3.services.pages import PagesClient
from google.cloud.dialogflowcx_v3.types.page import Page

# Configuración de Dialogflow CX
DIALOGFLOW_CX_CREDENTIALS = "/content/proyecto-final-379916-a58d7bc22b78.json" # Se usa el archivo de credenciales subido anteriormente
DIALOGFLOW_CX_PROJECT_ID = 'proyecto-final-379916' # Nombre del proyecto
DIALOGFLOW_CX_LOCATION = 'us-central1' # Ubicacion del proyecto
DIALOGFLOW_CX_AGENT_ID = '2542e633-3551-4551-9862-c1c95acc71b7' # ID del agente
DIALOGFLOW_CX_FLOW_ID = '00000000-0000-0000-0000-000000000000' # ID del flujo que contiene la página
DIALOGFLOW_CX_PAGE_ID = 'f603290a-5278-4775-b40e-9b710db103d6' # ID de la página "actualizar" a la que se quiere acceder

# Autenticación con Dialogflow CX
credentials = service_account.Credentials.from_service_account_file(DIALOGFLOW_CX_CREDENTIALS)
pages_client = PagesClient(credentials=credentials, client_options={"api_endpoint": "us-central1-dialogflow.googleapis.com"})

# Construir el nombre de la página a partir de los valores proporcionados
page_name = f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}/flows/{DIALOGFLOW_CX_FLOW_ID}/pages/{DIALOGFLOW_CX_PAGE_ID}"

# Obtener la página actual
page = pages_client.get_page(name=page_name)

# Crear una nueva página con el mismo nombre y displayName pero sin rutas
new_page = Page(name=page.name, display_name=page.display_name)

# Actualizar la página en Dialogflow CX para borrar todas las rutas existentes
updated_page = pages_client.update_page(page=new_page)

```

Nota. La figura muestra la parte del código que elimina el contenido de la página “actualizar”.

Captura de pantalla tomada del código implementado en Colab.

A continuación, en la figura 10, se procede a utilizar el archivo de credenciales de Google Sheets para autenticarse y definir el documento de Google Sheets y la página donde se guarda la información para actualizar el chatbot. Esto permitirá leer y guardar toda la información que se encuentra en el documento.

Figura 10

Leer y guardar la información del documento

```
#Autenticación de google sheets y obtención de datos del documento
from google.oauth2 import service_account
from google.cloud.dialogflowcx_v3.services.pages import PagesClient
from google.cloud.dialogflowcx_v3.types.page import Page, Form
from google.cloud.dialogflowcx_v3.types.page import TransitionRoute
from google.cloud.dialogflowcx_v3.types import Fulfillment
from googleapiclient.discovery import build

# Configuración de Dialogflow CX
GOOGLE_SHEETS_CREDENTIALS = "/content/proyecto-final-379916-505ec0fea11e.json"
DIALOGFLOW_CX_PROJECT_ID = 'proyecto-final-379916'
DIALOGFLOW_CX_LOCATION = "us-central1"
DIALOGFLOW_CX_AGENT_ID = '2542e633-3551-4551-9862-c1c95acc71b7'
DIALOGFLOW_CX_FLOW_ID = '00000000-0000-0000-0000-000000000000' # ID del flujo que contiene la página
DIALOGFLOW_CX_PAGE_ID = 'f603290a-5278-4775-b40e-9b710db103d6' # ID de la página a la que se quiere acceder
PAGE_TRANSICION = "354634ba-24f9-4879-9227-93e72a9e4269" #ID de la pagina "Tienes otra duda"

# Configuración de Google Sheets
SPREADSHEET_ID = '10dR1VZivZuroMsMY-1j2V2ecTxGr8ZDoP_qvs9jFs1Q'
RANGE_NAME = 'Hoja 1!A2:C'

# Autenticación con Dialogflow CX y Google Sheets
credentials = service_account.Credentials.from_service_account_file(GOOGLE_SHEETS_CREDENTIALS)
pages_client = PagesClient(credentials=credentials, client_options={"api_endpoint": "us-central1-dialogflow.googleapis.com"})
sheets_service = build('sheets', 'v4', credentials=credentials)

# Obtener los valores de la hoja de cálculo
result = sheets_service.spreadsheets().values().get(spreadsheetId=SPREADSHEET_ID,
                                                    range=RANGE_NAME).execute()
values = result.get('values', [])
```

Nota. La figura muestra la parte del código que lee y guarda la información del documento de Google sheets. Captura de pantalla tomada del código implementado en Colab.

En la figura 11 se realiza el siguiente paso consiste en eliminar todos los intents creados previamente, excepto aquellos que se encuentran en otra página que no sea "actualizar". Luego de borrarlos, se procede a crear nuevos intents a partir de la información leída del documento en la sección anterior. Para esta parte, se utilizarán únicamente la primera columna del documento, que

corresponde al nombre del intent, y la tercera columna, que contiene las frases de entrenamiento. Cabe aclarar que se pueden agregar tantas frases de entrenamiento como se requieran, separadas por un guion medio ("-") en la misma celda de la tercera columna. Por último, se han agregado retrasos en la eliminación y creación de intents para evitar sobrepasar el límite de llamadas a la API en un minuto. El tiempo de ejecución de esta parte del código es de aproximadamente nueve minutos, para eliminar y crear 68 intents.

Figura 11

Eliminar y crear nuevos intents

```
#Eliminar todos los intents y luego creamos un intent por cada fila que exista en el google sheets
#tiempo de ejecucion aproximada de 9 minutos para 68 intents
from google.cloud.dialogflowcx_v3.services.intents import IntentsClient
from google.cloud.dialogflowcx_v3.types.intent import Intent
from google.cloud import dialogflowcx_v3 as dialogflowcx
import time

endpoint = 'us-central1-dialogflow.googleapis.com:443'
intents_client = IntentsClient(credentials=credentials, client_options={'api_endpoint': endpoint})

# Eliminar todos los intents excepto dos específicos y el intent de fallback
intents_to_keep = ["Default Welcome Intent", "sin condición para volver a actualizar"]
intents = intents_client.list_intents(parent=f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}")
for intent in intents:
    if not intent.is_fallback and intent.display_name not in intents_to_keep:
        intents_client.delete_intent(name=intent.name)
        time.sleep(1.5) # Agregar una demora entre las llamadas a la API para evitar exceder el límite de llamadas que se pueden hacer a la API

# Crear intents a partir de las filas del documento de Google Sheets
for row in values:
    display_name = row[0]
    training_phrases = row[2].split("-") if len(row) > 2 and row[2].strip() else [] # Verificar contenido de la celda y separar las frases por guiones "-"

    # Crear un nuevo intent con el nombre y las frases de entrenamiento especificadas
    intent = dialogflowcx.Intent(display_name=display_name)
    if training_phrases:
        intent.training_phrases.extend([dialogflowcx.Intent.TrainingPhrase(parts=[dialogflowcx.Intent.TrainingPhrase.Part(text=phrase)], repeat_count=1) for phrase in training_phrases])
    intents_client.create_intent(parent=f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}", intent=intent)

# Agregar una demora entre las llamadas a la API para evitar exceder el límite de llamadas que se pueden hacer a la API
time.sleep(5)
```

Nota. La figura muestra la parte del código que elimina los intents existentes en la página “actualizar”, y crea unos nuevos a partir de la información que este en el documento de Google sheets. Captura de pantalla tomada del código implementado en Colab.

Finalmente, en la figura 12, encontramos la última sección del código se encarga de crear las rutas en la página "actualizar". Para hacerlo, el programa creará una ruta para cada intent previamente creado y le asignará su respectiva respuesta o "fulfillment" a cada una de las preguntas. Además, activará y seleccionará el webhook llamado "Enviar preguntas", que permitirá

enviar las preguntas de los usuarios a un script externo con el fin de guardarlas en un documento de Google Sheets. Esto nos permitirá analizar cómo los usuarios hacen preguntas y así poder mejorar el chatbot en el futuro. Por último, a todas las rutas se le agrega la transición a la siguiente página llamada "Tienes otra duda", completando así la configuración de la route.

Figura 12

Creación de routes en la página “actualizar”

```
#Por ultimo creamos una route en la pagina "actualizar" para cada intent que se creo anteriormente
from google.cloud.dialogflowcx_v3.services.pages import PagesClient
from google.cloud.dialogflowcx_v3.types import page

WEBHOOK_ID = "44c8f82d-0b0a-489f-8b28-87e92f4e72a4" #ID del webhook para mandar informacion y guardar las preguntas que hacen los usuarios

pages_client = PagesClient(credentials=credentials, client_options={'api_endpoint': endpoint})

page_name = f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}/flows/{DIALOGFLOW_CX_FLOW_ID}/pages/{DIALOGFLOW_CX_PAGE_ID}"

# Obtener la página
my_page = pages_client.get_page(name=page_name)

# Obtener una lista de todos los intents en el agente
intents = intents_client.list_intents(parent=f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}")

# Crear un diccionario para mapear los nombres de los intents a sus IDs
intent_ids = {intent.display_name: intent.name.split('/')[-1] for intent in intents}

# Crear una ruta para cada fila del documento de Google Sheets
for row in values:
    display_name = row[0]
    fulfillment_message = row[1]

    # Obtener el ID y el nombre del recurso del intent existente
    intent_id = intent_ids[display_name]
    intent_name = f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}/intents/{intent_id}"

    # Crear una nueva ruta con el nombre y el mensaje de cumplimiento especificados
    route = page.TransitionRoute(intent=intent_name)
    route.trigger_fulfillment.messages.append(dialogflowcx.ResponseMessage(text=dialogflowcx.ResponseMessage.Text(text=[fulfillment_message])))

    # Seleccionar el webhook que se utilizará para esta ruta
    route.trigger_fulfillment.webhook = f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}/webhooks/{WEBHOOK_ID}"

    # Especificar la etiqueta de cumplimiento para el webhook
    route.trigger_fulfillment.tag = "Enviar pregunta del usuario"

    # Agregar la transición de página a otra página específica
    route.target_page = f"projects/{DIALOGFLOW_CX_PROJECT_ID}/locations/{DIALOGFLOW_CX_LOCATION}/agents/{DIALOGFLOW_CX_AGENT_ID}/flows/{DIALOGFLOW_CX_FLOW_ID}/pages/{PAGE_TRANSICION}"

    # Agregar la ruta a la página
    my_page.transition_routes.append(route)

# Actualizar la página con las nuevas rutas
request = page.UpdatePageRequest(page=my_page)
pages_client.update_page(request)
```

Nota. La figura muestra la parte del código que crea las routes en la página “actualizar”, a partir de los intents creados anteriormente. Captura de pantalla tomada del código implementado en Colab.

9.2.3 Instrucciones para ejecutar el código

Para ejecutar el código, es necesario tener acceso a los archivos de credenciales de Dialogflow y Google Sheets. Estos archivos deben estar descargados en su propio ordenador y se debe tener acceso al enlace de Colab donde se encuentra el código.

Una vez en el enlace, para ejecutar cada sección del código se debe hacer clic en el botón "Ejecutar celda" ubicado en la parte superior izquierda de cada sección de código. Es importante esperar a que la confirmación de que la ejecución de la sección ha sido completada aparezca antes de continuar con la siguiente sección.

Es importante recordar que, en la segunda sección del código, se pedirá al usuario que cargue los archivos de credenciales de Dialogflow y Google Sheets antes de continuar. Si todas las secciones se ejecutan correctamente y no se obtienen errores, el chatbot quedará actualizado.

9.3 Script para guardar las preguntas de los usuarios

Para hacer seguimiento de las preguntas de los usuarios, se ha creado un código en Apps Script de Google. Este código tiene como función principal recibir la información que se envía a través del webhook desde Dialogflow CX, filtrar los datos relevantes y guardarlos en un documento de Google Sheets. El envío de información ocurre en el momento en que el usuario hace una pregunta al chatbot en la página "Actualizar" y este la asocia con uno de los intents, lo que significa que el chatbot ha comprendido la pregunta y se ha entrado en una de las rutas definidas.

Cuando el chatbot genera una respuesta para el usuario, se envía información hacia el script a través de una solicitud POST en formato JSON mediante el webhook activado en cada una de las rutas. El código recibe esta información a través de la función `doPost` y la guarda en la variable "e", que luego se convierte en un objeto de JavaScript para filtrar los datos de interés.

Por último, se extraen los datos importantes del objeto, como la pregunta del usuario, la pregunta asociada al intent y la respuesta proporcionada por el chatbot. Estos datos se almacenan en un documento de Google Sheets para su posterior monitoreo. De esta forma, se puede comprender mejor cómo los usuarios hacen preguntas y mejorar el chatbot en el futuro. El código se puede revisar en el apéndice D.

9.4 Integración del chatbot en la página web

Finalmente, se creó una página web utilizando la plataforma Netlify. Esta plataforma permite crear páginas web de forma gratuita y accesible a cualquier persona. Para crear la página web, se utilizó una plantilla en formato HTML, descargada de una de las muchas páginas web que ofrecen plantillas gratuitas. La plantilla elegida fue una página web sencilla, que se editó para agregarle un título y las instrucciones de uso del chatbot. Como el enfoque principal del proyecto no estaba orientado a la creación de la página web, se optó por una plantilla ya hecha para ahorrar tiempo y recursos.

Para integrar el chatbot en la página web previamente creada, se accedió al proyecto de Dialogflow CX y se fue al menú izquierdo en la sección "Manage". Luego, se seleccionó la opción "integrations" y en la sección "Text Based", se eligió la opción "Dialogflow Messenger" para integrar el chatbot con la página web. Se habilitó la herramienta y se obtuvo un código, el cual se copió y pegó en el archivo HTML principal de la página web. Con esto, se logró que el widget del chatbot apareciera en la página web y funcionara correctamente. El resultado de la implementación se puede ver en la figura 13.

Figura 13

Chatbot implementado en la página web



Nota. La figura muestra el chatbot funcional, que fue implementado en la página web.

10. Análisis de resultados

En este capítulo se detallará el proceso de pruebas y se presentarán los resultados obtenidos. Como se mencionó en el capítulo anterior, se seleccionaron seis estudiantes que habían cursado recientemente la materia para evaluar el desempeño del chatbot en preguntas relacionadas con la misma. El objetivo de esta prueba fue obtener retroalimentación sobre el funcionamiento del chatbot y conocer las opiniones de los estudiantes acerca de si una herramienta de este tipo hubiera sido beneficiosa durante su estudio de la materia. En este capítulo se analizarán los resultados obtenidos y se discutirán las principales conclusiones derivadas de la prueba.

10.1 Descripción de la prueba

Debido a que el chatbot está enfocado en responder solo lo relacionado a las temáticas de la asignatura trabajo de grado I, se planteó una prueba en la cual cada uno de los estudiantes deberá hacerle todas las preguntas a las que el chatbot está entrenado para resolver. A cada estudiante se le indicaba la temática de cada pregunta, lo que servía como ejemplo para que los estudiantes pudieran hacer cada una de las preguntas, solo que debían tratar de escribirla tal y como ellos lo harían. Para llevar a cabo la prueba, se diseñó una tabla en la que los estudiantes debían registrar sus preguntas y las respuestas proporcionadas por el chatbot.

Tabla 3

Ejemplo del formato de las pruebas

	Pregunta a realizar	Intento	Pregunta realizada por el usuario	Respuesta del chatbot	Intento asociado	TP	TN	FP	FN
1	Pregunta sobre ID o sigla del proyecto	2							

Nota. La tabla muestra el formato en que los estudiantes registran el resultado de las pruebas.

Adaptado del Google sheets apéndice E.

En la Tabla 3 se puede observar que a cada estudiante se le indicaba la temática sobre la cual debía realizar su pregunta al chatbot. El estudiante debía redactar su propia pregunta y llenar las casillas correspondientes a "pregunta realizada" y "respuesta del chatbot". De esta forma, se podía evaluar si el chatbot respondía correctamente o no a cada pregunta, y clasificar las respuestas en verdaderos positivos (TP) cuando el chatbot respondía correctamente, falsos positivos (FP) cuando el chatbot respondía incorrectamente, verdaderos negativos (TN) cuando el chatbot no

respondía porque no estaba entrenado para esa pregunta y falsos negativos (FN) cuando el chatbot no respondía correctamente a una pregunta que debería ser capaz de responder.

10.2 Resultados obtenidos

Todas las pruebas realizadas se pueden ver en el apéndice E, una vez que todos los datos fueron clasificados se realizó una tabla para calcular las medidas de rendimiento de cada clase. En la figura 13 podemos ver un ejemplo de los resultados para las clases 1 y 2, en el caso de la clase 1, vemos que el chatbot se equivocó una sola vez, esto se reflejó en las medidas del Recall de 83.33% y el F1-score de 90.91%, de esto podemos concluir que una buena medida para evaluar el rendimiento del chatbot es el Recall, debido a que su porcentaje es acorde al porcentaje de acierto.

Tabla 4

Ejemplo de resultados para cada clase

Pregunta o clase	TP	TN	FP	FN	Precisión	Recall	F1-score
1	5	0	0	1	100%	83.33%	90.91%
2	6	0	0	0	100%	100%	100%

Nota. La tabla muestra un resumen de las pruebas y el cálculo de las métricas de rendimiento para cada clase. Adaptada del Google sheets para verla completa visite el apéndice E y diríjase a la página llamada “Matriz de confusión”.

Empleando las fórmulas (2), (3) y (4) se encontró la precisión, el Recall y el F1-score respectivamente para cada clase, finalmente se encontraron los valores promedio para toda la prueba, donde se encontró la exactitud con ayuda de la formula (1), la precisión, el Recall y el F1-score.

En la Tabla 5 podemos observar las medidas promedio obtenidas, lo que nos da una visión más clara del rendimiento total del chatbot.

Tabla 5*Métricas de rendimiento promedio*

Métrica	Valor Promedio
Exactitud	87.12%
Precisión	98.87%
Recall	88.03%
F1-Score	91.74%

Nota: Esta tabla muestra las métricas de rendimiento promedio de todas las clases.

En conclusión, de los resultados obtenidos en la tabla 3, llama la atención el valor de la precisión, ya que con 98.87% supera bastante a las otras medidas, esto se debe principalmente a la naturaleza y diseño de las pruebas realizadas, ya que la mayoría de los resultados de clasificación era un TP o un FN, y si miramos la fórmula (2) nos damos cuenta de que la precisión no depende de los FN y por ello es por lo que da tan alto ese valor. En cuanto a las demás medidas que fueron exactitud 87.12%, Recall 88.03% y F1-score de 91.74%, también podemos decir que dan valores bastante altos, pero era lo que se esperaba, ya que solo se estaba evaluando el funcionamiento del chatbot con respecto a las preguntas que estaba entrenado. Por lo tanto, se esperaba obtener valores altos en las métricas evaluadas. En general, los resultados sugieren que el chatbot funciona adecuadamente para responder preguntas relacionadas con la temática de la asignatura trabajo de grado I. Sin embargo, se recomienda seguir explorando y mejorando su capacidad para responder preguntas más complejas y fuera del alcance de su entrenamiento actual.

11. Trabajos futuros

A partir de los resultados obtenidos en este proyecto, se abren nuevas oportunidades para investigar en el campo de las tecnologías educativas. Se sugiere que futuras investigaciones se centren en la exploración de herramientas basadas en inteligencia artificial, como las tecnologías GPT, para abordar desafíos similares a los planteados en este proyecto. Sería interesante comparar el desempeño del chatbot implementado en este proyecto con soluciones basadas en tecnologías GPT, que pueden leer documentos y responder preguntas en base a ellos. Al enfocarse en estas áreas, se pueden encontrar nuevas soluciones para mejorar la eficacia de las tecnologías educativas y brindar una mejor experiencia de aprendizaje para los usuarios. Además sería interesante comparar el desempeño del chatbot implementado en este proyecto, con soluciones como GPT, que pueden leer un documento y en base a esto responder preguntas.

Es relevante destacar que actualmente existen soluciones basadas en código abierto que podrían ser de gran utilidad para abordar problemáticas similares a las planteadas en este proyecto. El uso de estas soluciones podría brindar la posibilidad de desarrollar herramientas gratuitas en el futuro, ya que uno de los inconvenientes que presentan otras soluciones es su costo. Sería interesante investigar y evaluar estas opciones de software libre para determinar su potencial aplicabilidad en futuros proyectos.

12. Conclusiones

Después de haber implementado el chatbot y analizado los resultados obtenidos durante las pruebas, se puede concluir que el chatbot demostró un rendimiento satisfactorio para las

preguntas para las cuales fue entrenado, tal y como se evidencia en la Tabla 5. Las medidas de rendimiento obtenidas, incluyendo una exactitud del 87.12%, una precisión del 98.87%, un Recall del 88.03% y un F1-score del 91.74%, indican un correcto funcionamiento para la mayoría de las preguntas. Además, los estudiantes que participaron en las pruebas expresaron una retroalimentación positiva sobre la capacidad del chatbot para responder preguntas.

Cabe destacar que gracias a haber seleccionado la plataforma Dialogflow CX, a lo largo del proyecto se encontró que esta nos permitió contar con ventajas que no conocíamos al inicio cuando la escogimos, ya que para la implementación del método que puede editar las conversaciones o preguntas y respuestas del Bot, que era el objetivo clave del proyecto, al contar con dos plataformas que eran de Google, como lo son Dialogflow CX y Google sheets, esto facilitó la realización del programa en Python que permite interactuar con ambas, gracias a que ambas API se pueden habilitar y configurar desde la plataforma de Google Cloud Platform.

La implementación del código en Python y la interacción con la API de Dialogflow CX condicionaron el diseño del flujo conversacional del Bot. Debido a esta limitación, se realizó una recopilación de preguntas existentes y se organizó en categorías para facilitar la interacción del usuario con el Bot. El diseño del flujo conversacional fue cuidadosamente planeado para lograr un equilibrio entre la complejidad del flujo y la complejidad del código de Python.

La implementación fue exitosa y el chatbot se puede acceder fácilmente a través de la página web. Esta integración permite que los usuarios puedan interactuar con el chatbot de manera práctica y sencilla, lo que facilita la experiencia de usuario y mejora la accesibilidad del chatbot.

La implementación del chatbot funcional y la aplicación web para recopilar las preguntas de los usuarios son productos valiosos que se entregan en este proyecto. Además, el código de Python proporcionado para permitir la edición de las preguntas y respuestas del chatbot, garantiza

que la herramienta no se quedará obsoleta y se mantendrá actualizada. Estas soluciones muestran un gran potencial para mejorar la calidad del aprendizaje en la asignatura Trabajo de Grado I y pueden ser utilizadas por los estudiantes para resolver sus dudas en cualquier momento. Además, la exploración de temas de investigación como Google Cloud Platform y la comunicación entre servicios de aplicaciones web son una clara indicación del potencial de expansión y mejora de la herramienta en el futuro. En resumen, estos productos entregados demuestran el éxito del proyecto y su capacidad para hacer una contribución valiosa a la educación y la investigación en el área de la Ingeniería Electrónica.

13. Recomendaciones

Se espera que con el tiempo se puedan agregar más preguntas y respuestas al chatbot, así como frases de entrenamiento con ayuda de las preguntas que se vayan recopilando gracias a la aplicación web que se implementó. Además, se espera poder incorporar el chatbot en una página o plataforma donde esté toda la información relacionada con el trabajo de grado, para que sea más fácil para los estudiantes encontrarlo y se pueda tener un mejor control de quién puede o no usar el chatbot.

También se busca que este tipo de proyectos promueva el desarrollo de nuevas tecnologías en la escuela y que se continúe con la investigación en este campo, ya que el uso de estas tecnologías puede traer muchos beneficios tanto en el ámbito educativo como empresarial.

Referencias Bibliográficas

- Abdellatif, A., Badran, K., Costa, D. E., & Shihab, E. (2022). A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering. *IEEE Transactions on Software Engineering*, 48(8), 3087-3102. 10.1109/TSE.2021.3078384
- Amazon. (2022a). *Amazon Lex* . Retrieved 2022, from <https://aws.amazon.com/es/lex/?nc=sn&loc=1>
- Amazon. (2022b). Amazon Lex - Precios. <https://aws.amazon.com/es/lex/pricing/>
- Arumugam, R., & Shanmugamani, R. (2018). *Hands-On Natural Language Processing with Python*. Packt Publishing.
- Biswas, M. (2018). AI and Bot Basics. *Beginning AI Bot Frameworks* (). Apress L. P.
- Geron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. Reilly Media, Incorporated.
- Google. (2022). *Aplicación web*. <https://developers.google.com/apps-script/guides/web?hl=en-419>
- Google. (2023). *Colab*. <https://colab.research.google.com/>
- Google Cloud. (2022a). *Dialogflow CX documentation*. <https://cloud.google.com/dialogflow/cx/docs/basics?hl=en>

Google Cloud. (2022b). Precios de Dialogflow CX.

<https://cloud.google.com/dialogflow/pricing?hl=es-419#cx-trial>

Google Cloud. (2023). API de Google Cloud.

<https://cloud.google.com/apis/docs/overview?hl=es-419>

IBM. (2022). *IBM Watson Assistant - Virtual Agent*. Retrieved Feb 4, 2023, from

<https://www.ibm.com/products/watson-assistant>

Internet Engineering Task Force. (2014). *Método POST*. Retrieved 2023, from

<https://datatracker.ietf.org/doc/html/rfc7231#section-4.3.3>

JSON. (2023). *Notación de Objetos de JavaScript*. <https://www.json.org/json-en.html>

Lopez Cubides, J. A. (2022). *Diseño, desarrollo y entrenamiento de un Bot para una plataforma transmedia del páramo de Santurbán*. Universidad Industrial de Santander.

Microsoft. (2022a). *Azure Bot Service Conversational AI Application | Microsoft Azure*.

Retrieved 2022, from <https://azure.microsoft.com/en-us/products/bot-services>

Microsoft. (2022b). *Precios de Azure Bot Services | Microsoft Azure*. Retrieved Feb 4, 2023,

from <https://azure.microsoft.com/es-es/pricing/details/bot-services/>

Mozilla. (2023). JavaScript. <https://developer.mozilla.org/es/docs/Web/JavaScript>

Pandey, Y. N., Rastogi, A., Kainkaryam, S., Bhattacharya, S., & Saputelli, L. (2020). *Machine Learning in the Oil and Gas Industry*. Apress L. P.

- Peyton, K., & Unnikrishnan, S. (2023). A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs. *Results in Engineering*, 17, 100856. 10.1016/j.rineng.2022.100856
- Python. (2023). Python documentation. <https://www.python.org/about/>
- Rodriguez, S., & Mune, C. (2022). Uncoding library chatbots: deploying a new virtual reference tool at the San Jose State University library. *Reference Services Review*, 50(3), 392-405. 10.1108/RSR-05-2022-0020
- Russell, S. J., & Norvig, P. (1994). Inteligencia artificial : un enfoque moderno. *Inteligencia artificial: un enfoque moderno (2a. ed.)* (). Pearson Educación.
- Vahid Mirjalili, S. R. (2020). *Python Machine Learning* Marcombo. Retrieved from <https://www.perlego.com/book/2152522/python-machine-learning-pdf>
- Wollny, S., Schneider, J., Di Mitri, D., Weidlich, J., Rittberger, M., & Drachsler, H. (2021). Are we there yet? - A systematic literature review on chatbots in education. *Frontiers in Artificial Intelligence*, 4(654924), 18-4:654924
- World Wide Web Consortium. (2023). *HTML*. <https://html.spec.whatwg.org/multipage/>

Apéndices

Apéndice A. Lista de preguntas y respuestas recopiladas

Apéndice B. Documento de Google sheets para actualizar el chatbot

Apéndice C. Código para actualizar el chatbot a partir de un documento de Google sheets

Apéndice D. Código del script para guardar las preguntas de los usuarios

Apéndice E. Pruebas de rendimiento

Los apéndices están adjuntos y puede visualizarlos en la base de datos de la biblioteca UIS