



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

ChatScriptor

Interfaz gráfica para la gestión
de chatbots en la plataforma
de Dialogflow.



Presentado por Claudia Landeira Viñuela
en la Universidad de Burgos — 4 de julio
de 2023

Tutor: Dr. Raúl Marticorena Sánchez



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Raúl Marticorena Sánchez, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que la alumna Dña. Claudia Landeira Viñuela, con DNI 71052355L, ha realizado el Trabajo Final de Grado en Ingeniería Informática titulado “Interfaz gráfica para la gestión de chatbots en la plataforma Dialogflow”.

Y que dicho trabajo ha sido realizado por la alumna bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 4 de julio de 2023

Vº. Bº. del Tutor:

D. Raúl Marticorena Sánchez

Resumen

Dialogflow es una plataforma desarrollada por Google que permite desarrollar y administrar chatbots o asistentes virtuales utilizando técnicas como el procesamiento del lenguaje natural (PNL) que facilitan las interacciones entre aplicación y persona.

La interfaz que nos encontramos en su versión básica y gratuita está, actualmente, muy limitada.

Durante este proyecto, se analiza y evalúa esta herramienta, permitiendo identificar aquellos detalles que se han considerado como necesidades actuales a la hora de crear un chatbot. Dada la relevancia de este tipo de productos, es importante que posea elementos que hagan que la producción de chatbot esté al alcance de todos.

Es por esto, que este trabajo se propone desarrollar una interfaz gráfica que mejore la interacción con respecto a la oficial de Google, así como añadir servicios que permitan optimizar y clarificar la experiencia del usuario.

Descriptores

Dialogflow; gestión de chatbots; Google Cloud; aplicación web; aprendizaje automático; Inteligencia Artificial (IA); Procesamiento del Lenguaje Natural (PNL); Interacciones Hombre-Máquina (IHM).

Abstract

Dialogflow is a platform developed by Google that allows the development and management of chatbots or virtual assistants using techniques such as natural language processing (NLP) that facilitate interactions between application and person.

The interface that we find in its basic and free version is currently very limited.

During this project, this tool is analysed and evaluated, making it possible to identify those details that have been considered as current needs when creating a chatbot. Given the relevance of this type of product, it is important that it has elements that make chatbot production accessible to everyone.

For this reason, this work aims to develop a graphical interface that improves the interaction with respect to the official Google one, as well as adding services that allow optimising and clarifying the user experience.

Keywords

DialogFlow; chatbots management; Google Cloud; web application; machine learning; Artificial Intelligent (AI); Natural Language Processing (NLP); Human-Computer Interaction (HCI).

Índice general

Índice general	iii
Índice de figuras	v
Introducción	1
1.1. Estructura de la memoria	2
1.2. Estructura de los anexos	3
1.3. Materiales	3
Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	5
2.3. Objetivos personales	6
Conceptos teóricos	7
3.1. Introducción a chatbots	7
3.2. Dialogflow	13
Técnicas y herramientas	19
4.1. Técnicas	19
4.2. Herramientas	21
Aspectos relevantes del desarrollo del proyecto	27
5.1. Inicio del proyecto	27
5.2. Metodologías	28
5.3. Formación	28
5.4. Desarrollo del proyecto	28

5.5. Problemas encontrados	30
5.6. Testing	38
5.7. Documentación	38
5.8. Uso en producción	39
Trabajos relacionados	41
Conclusiones y Líneas de trabajo futuras	43
7.1. Conclusiones	43
7.2. Líneas de Trabajo Futuras	44
Bibliografía	47

Índice de figuras

3.1. Esquema introductorio para chatbots [35]	8
3.2. Búsqueda desde Google de Dialogflow [12]	14
3.3. Primera pantalla después de crear un agente	15
3.4. Creación de entidad	15
3.5. Frases de entrenamiento	16
3.6. Respuestas	16
3.7. Interfaz de configuración del agente	17
4.1. Patrón de diseño MVP [18]	20
4.2. Patrón de diseño Adaptador [1]	21
4.3. Interfaz de Trello [16]	22
4.4. Interfaz para el uso de Markdown a través de Pycharm Professional	23
4.5. Interfaz para el uso de L ^A T _E X a través de Overleaf	24
5.1. Logo ChatScriptor	27
5.2. Petición <i>GET</i> del agente	31
5.3. Petición <i>GET</i> de los intents	31
5.4. Petición <i>GET</i> de un intent	32
5.5. Petición <i>GET</i> de las entidades	32
5.6. Petición <i>GET</i> de una entidad	33
5.7. Crear proyecto en la consola de Google Cloud	33
5.8. Panel principal del nuevo proyecto	34
5.9. Acceso a las APIs y servicios	34
5.10. Buscar y habilitar la API de Dialogflow	35
5.11. Credenciales para el proyecto	35
5.12. Estructura de directorios de un chatbot ejemplo	36

Introducción

Los chatbots son herramientas software que nos permiten hacer consultas y obtener una respuesta de forma rápida. Se desarrollan mediante inteligencia artificial, ya que algunos de ellos, son capaces de interpretar imágenes o sonidos. Los más sencillos, actúan siguiendo una serie de guiones que le son establecidos por su creador.

Actualmente, este tipo de interacciones hombre-máquina, se utilizan con fines comerciales, industriales y educativos. Permiten tener acceso a la información sin tener que realizar interacciones directas y poder resolver cualquier pregunta las 24 horas del día, los 365 días del año. Esto proporciona un gran número de ventajas, tanto para el usuario que crea el chatbot como para quién hace uso del mismo.

Existen gran variedad de empresas que permiten la creación, mantenimiento y seguimiento de los chatbots tanto de manera gratuita como de pago. Algunas de las más conocidas y usadas son: Microsoft Bot Framework, Amazon Lex, IBM Watson Assistant, Salesforce Einstein Bots, SAP Conversational AI y Google Dialogflow.

Con la explosión de las inteligencias artificiales actuales, se está desarrollando aún más software que permiten crear chatbots. Debido a las necesidades actuales, muchos de los usuarios de este tipo de herramientas no tienen conocimientos de programación, con lo que es importante tener opciones que estén basadas en programación *low code*¹.

Nos encontramos con varios tipos de chatbots, entre ellos están los asistentes que utilizan algoritmos de generación de lenguaje natural basados

¹La programación *low code* hace referencia a la capacidad de desarrollar software reduciendo el uso de programación tradicional [30].

en modelos avanzados como las redes neuronales recurrentes (RNR). Estos permiten crear respuestas y mantener conversaciones fluidas, ya que tienen la capacidad de originar sus propias respuestas usando los conceptos relevantes de la discusión, adaptándose al usuario. Aunque no todo son ventajas, ya que sus respuestas pueden ser incompletas o erróneas, debido, comúnmente, a la falta de entrenamiento de estos modelos.

Por otro lado, existen los chatbots basados en guiones, patrones o reglas predefinidas. Estos chatbots utilizan estas pautas para conversar con el usuario, dándole las respuestas según estas instrucciones, haciendo responsable al creador de añadir suficientes oraciones a las diferentes entradas. Es por esto, que poseen una flexibilidad bastante baja así como poca capacidad de interpretación de las conversaciones, pero al revés que los anteriores, las respuestas serán las correctas siempre y cuando coincidan con los patrones. Estos chatbots se usan en campos estructurados que requieren una mayor precisión de respuesta, como puede ser un asistente de preguntas frecuentes, con información general.

Este trabajo se centra en Google Dialogflow, la plataforma que nos ofrece Google mediante Google Cloud. Los productos que se pueden crear con esta aplicación son del segundo tipo, es decir, chatbots basados en guiones. El motivo de esta elección es la capacidad que tiene para expandirse y desarrollarse, ya que al no poseer una compleja técnica de creación de chatbots, permite añadir funcionalidades y mejoras. Además de que es un software gratuito en su versión básica.

A lo largo de este proyecto, se desarrolla una aplicación web basada en la oficial, realizando un previo análisis para encontrar puntos a mejorar, que permitan al usuario tener una mejor experiencia de uso.

1.1. Estructura de la memoria

La memoria posee la siguiente estructura:

- **Introducción:** descripción del problema y cuál es la solución elegida, estructura de la memoria, de los anexos y los materiales adjuntados.
- **Objetivos del proyecto:** explicación de forma precisa y concisa de los objetivos que se persiguen con la realización del proyecto.
- **Conceptos teóricos:** descripción de los conceptos teóricos estudiados y aplicados a la solución del proyecto.
- **Técnicas y herramientas:** descripción de las técnicas y herramientas usadas tanto para la gestión como para el desarrollo del proyecto.

- **Aspectos relevantes del desarrollo:** exposición de los puntos destacables que han surgido durante la realización del proyecto.
- **Trabajos relacionados:** relación del proyecto actual con otros de similares características.
- **Conclusiones y líneas de trabajo futuras:** conclusiones del trabajo y las diferentes posibles mejoras y ampliaciones del proyecto.

1.2. Estructura de los anexos

Los anexos poseen la siguiente estructura:

- **Plan del proyecto software:** planificación y viabilidad del proyecto.
- **Especificación de requisitos del software:** fase de análisis, describiendo los requisitos y objetivos del software.
- **Especificación de diseño:** descripción de diseño de datos, diseño procedimental y diseño arquitectónico.
- **Manual del programador:** documentación relacionada con la estructura, instalación, ejecución, etc.
- **Manual del usuario:** guía de usuario.

1.3. Materiales

Materiales adjuntados junto con la memoria:

- Página web desplegada: <https://chatscriptor.azurewebsites.net/>
- Repositorio del proyecto: <https://github.com/clv1003/Chat-Scriptor>
- Chatbots (Agentes) de ejemplo:
https://universidaddeburgos-my.sharepoint.com/:f:/g/personal/clv1003_alu_ubu_es/EpCn10HnKBRDi2I4tRlaYTwBvYpzJZDTckz0qWBcacVkzw?e=OSJIFG

Objetivos del proyecto

A continuación, se detallan los objetivos que han motivado la realización del proyecto:

2.1. Objetivos generales

- Desarrollar una aplicación web que permita realizar el mantenimiento y la gestión de chatbots Dialogflow.
- Mejorar la interfaz gráfica desarrollada por Google para su aplicación web Dialogflow.
- Aportar nuevas funcionalidades a la aplicación desarrollada.

2.2. Objetivos técnicos

- Desarrollar una aplicación Flask que realice de manera clara las funciones ya implementadas en la versión original y permita realizar otras nuevas.
- Emplear un *framework* para diseño de aplicaciones web como Bootstrap.
- Aplicar una estructura software en 3 capas como la arquitectura MVP (*Model-View-Presenter*) para el desarrollo de la aplicación.
- Utilizar Git como software de control de versiones junto con el uso de la herramienta GitHub.
- Aplicar metodologías ágiles como la metodología Scrum en el desarrollo del proyecto.

2.3. Objetivos personales

- Poner en práctica la mayor cantidad de conocimientos adquiridos durante el periodo académico desarrollado en la Universidad de Burgos.
- Conocer el procedimiento del desarrollo completo de un proyecto software.
- Adquirir conocimientos sobre la aplicación de la inteligencia artificial sobre campos como el comercio, la industria o la educación.
- Profundizar en el desarrollo software y diseño de interfaces.

Conceptos teóricos

En este apartado se explicarán de manera teórica todos los conceptos y detalles necesarios para el completo entendimiento del proyecto.

En primera instancia, se incluyen conceptos sobre chatbots, su funcionamiento y cuáles son las consecuencias de su uso en la vida cotidiana. En el segundo apartado, se desarrollan las alternativas que existen en el mercado. Por último, se analizará en profundidad la página de Dialogflow en su estado actual.

3.1. Introducción a chatbots

Tal y como se ha descrito en la introducción de esta memoria, los chatbots son herramientas software que permiten realizar una serie de interacciones con respuestas inmediatas.

En esta sección, se hablará de los diferentes tipos de tecnologías que utilizan los agentes virtuales actuales, así como sus tipos, cuáles de estas son las que utilizan y cómo influye su uso en la sociedad.

Como primera idea, debemos saber que los chatbots trabajan con una combinación de diferentes técnicas como son la inteligencia artificial, los modelos de interacción Hombre-Máquina [3] y el “Natural Language Processing” (NLP) [29] o “Procesamiento del Lenguaje Natural” (PLN), que permiten que los chatbots puedan realizar sus actividades objetivo.

A continuación se muestra una figura esquemática (ver Figura 3.1) extraída del artículo “*An Overview of Machine Learning in Chatbots*” [35], que muestra una vista general de un chatbot.

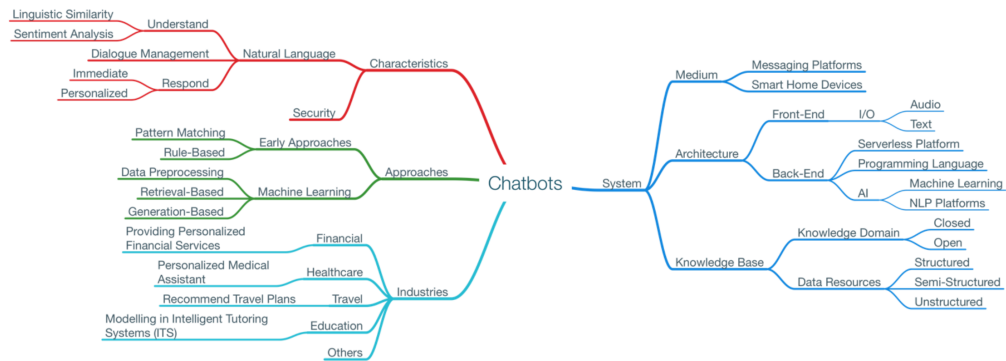


Figura 3.1: Esquema introductorio para chatbots [35]

Tecnologías detrás del funcionamiento de los chatbots

Hablemos de las tecnologías que intervienen en el funcionamiento de estos productos software.

A lo largo de los últimos años, se ha visto como este tipo de elementos se han desarrollado y se han introducido en nuestras vidas sin saber exactamente cómo funcionan.

En primer lugar, una de las herramientas más conocidas y que interviene en gran medida en este proceso, es la **inteligencia artificial**.

Siguiendo su definición: “*Disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico*” [25], obtenida del diccionario de la *Real Academia Española* en su edición 23, se puede establecer como la lógica detrás de la generación de respuestas coherentes a las entradas del usuario al mantener una conversación con un chatbot.

El campo de la inteligencia artificial es inmenso, con lo que debemos buscar aquellas áreas que se encarguen de procesos concretos para la realización de funciones como son el procesamiento del lenguaje o el aprendizaje automático.

Aprendizaje automático (*Machine Learning*)

El primer gran grupo que interviene en el funcionamiento de los asistentes virtuales dentro de la inteligencia artificial, es el aprendizaje automático o *machine learning*, definido como los algoritmos computacionales que,

basados en la inteligencia humana y las redes neuronales, es capaz de analizar y obtener la información relevante de un conjunto de datos en busca de patrones, en nuestro caso lingüísticos o auditivos, adquiriendo de ellos conocimientos para mejorar y predecir comportamientos a partir de los que ya se han producido [24]. Viendo esta acepción, es más que obvio la elevada relevancia que tiene y la capacidad de adaptabilidad que otorga a los agentes.

Concretando en nuestro objetivo, esta tecnología permite a los chatbots analizar los mensajes recibidos de los usuarios, clasificar e identificar la información relevante, generar una respuesta y predecir lo siguiente que puede ser preguntado por el usuario.

Natural Language Processing (NLP)

Otro punto muy importante a la hora de tratar con conversaciones humanas en las que es relevante el contexto, es la comprensión de esa situación por parte de nuestra máquina. Para completar este objetivo, se combina el aprendizaje automático junto con los Procesadores de Lenguaje Natural (PLN) o “*Natural Language Processing*” (NLP).

En este caso, suelen intervenir tres herramientas conjuntamente, los procesadores del lenguaje natural y, dentro de su ámbito, el “*Natural Language Understanding*” (NLU) y el “*Natural Language Generation*” (NLG).

El primero de ellos (NLP) [33], es el encargado de establecer las interacciones entre el agente y el usuario. Esto conlleva que el chatbot tenga la necesidad de conocer y comprender el lenguaje, su sintaxis principalmente, permitiendo analizar la conversación y entenderla.

Esto nos deja con un problema, ya que comprender y conocer la sintaxis de un idioma, muchas veces no es lo que se busca en un agente conversacional, si no que también se quiere que la máquina comprenda semánticamente la conversación, permitiendo que las respuestas puedan ser más concretas y fiables. Para esto se utiliza el “*Natural Language Understanding*” (NLU) [5], capaz de interpretar y extraer el contexto.

Teniendo estas dos tecnologías implementadas en nuestros chatbots, faltaría una última función importante, ya que a parte de comprender sintácticamente y semánticamente una conversación, un asistente debe ser capaz de hacer el proceso contrario, generar de manera coherente las respuestas. La herramienta encargada de esto es el “*Natural Language Generation* (NLG)” [32], que le otorga la habilidad de poder generar la

información estructurada y bien redactada a partir de los datos que ya posee y que ya ha analizado (*machine learning*).

Para finalizar, es importante guardar un histórico, tanto de la parte sintáctica como de la semántica, que fomentará que la siguiente interacción pueda ser mejor, permitiendo al agente tener un grupo de conocimientos que ya conoce y otros que no, creando un posible contenido relevante para esa conversación.

Tipos de chatbots

Las tecnologías explicadas anteriormente son una base general para tener conocimientos básicos sobre el funcionamiento de los chatbots, pero no todos hacen uso de todas, incluso hay algunos que no las utilizan.

Para crear una clasificación mucho más clara, se hará uso de una serie de parámetros. Esta diferenciación está basada en la desarrollada en el artículo “*An Overview of Chatbot Technology*” [26].

La primera forma de clasificación se centra en el alcance que tiene un chatbot para obtener el conocimiento.

- Dominio abierto: este hace referencia a los chatbots en los que los conceptos son globales, con un gran acceso y alcance a mucho contenido.
- Dominio cerrado: son asistentes creados para usos concretos y de campos más precisos, donde sus conocimientos están preestablecidos.

La segunda clasificación se basa en el tipo de relación que se establece entre el usuario y la máquina.

- Interpersonal: son aquellos en los que la comunicación es mucho más fría. Son los más usados para los agentes de preguntas frecuentes o de ejecución de comandos, entre otros.
- Intrapersonal: poseen cierta personalidad que los hace más cercanos con el usuario. Se utilizan en aplicaciones de redes sociales como *Discord* o *Telegram*.

La siguiente y última clasificación, es la más representativa, ya que se tiene en cuenta las tecnologías que utiliza.

- Basado en reglas: estos se programan bajo unas directrices que utilizan cuando el usuario emplea una serie de términos específicos. Esto los convierte en los más precisos, pero a la vez, en los menos adaptable e inflexibles.
- Basado en inteligencia artificial: son los asistentes más versátiles, pudiendo analizar las conversaciones y las interacciones para generar una respuesta en tiempo real.

Consecuencias en el mundo real

Conocido el funcionamiento interno que tienen los chatbots, es posible vislumbrar el inmenso número de puertas que nos abre. Este apartado tratará de resumir algunos de los ámbitos más importantes, ya que durante la investigación se han encontrado un exorbitante número de artículos relacionados.

Educación

Uno de estos campos es el de la educación [7]. Visto que los agentes virtuales poseen un gran potencial a la hora de introducirse en las aulas, podrían llegar a ser un elemento clave en la interacción alumno-profesor.

Una de las posibles aplicaciones es el uso de un chatbot abierto las 24 horas del día para preguntas frecuentes, permitiendo al alumno resolver sus dudas rápidamente. Esto aporta una serie de beneficios, también para el profesor, porque podrá ver las estadísticas de aquellos conceptos que han sido más buscados y poder insistir en la explicación de los mismos. Además, si se utiliza un tipo de chatbot mediante inteligencia artificial, se podría tener una máquina capaz de ir añadiendo los conceptos nuevos basándose en la frecuencia de las preguntas.

Siguiendo dentro de esta categoría, otra posible utilidad sería crear chatbots con cuestionarios que permitan a los alumnos profundizar de forma práctica los conocimientos impartidos, así como tenerlos disponibles en cualquier momento y lugar, evolucionando la forma de aprender. No todo son cosas positivas, ya que, como bien está descrito en el artículo de *Sam Cunningham-Nelson* y colaboradores (“*A review of chatbots in education: Practical steps forward*”) [7], estaríamos adentrándonos en la creación de chatbots de una complejidad mucho más elevada, además de tener que adaptar el concepto de aprendizaje. Siendo esta una idea muy interesante, es recomendable usarlo como mecanismo de apoyo y no como algo único a la hora de hablar de educación.

Sanidad

El siguiente sector es el de la sanidad [37]. Al igual que cuando se ha tratado el de educación, no se puede sustituir la tradicional forma de cubrir las necesidades médicas con este tipo de tecnología, pero si que pueden resultar bastante útiles en varios aspectos.

En el artículo “*Chatbot for Health Care and Oncology Applications Using Artificial Intelligence and Machine Learning: Systematic Review*” [37], explican como los agentes virtuales pueden mejorar la experiencia del paciente, pudiendo recibir información verídica a sus dudas o ser intermediario en situaciones delicadas. Además de poder obtener datos informativos de la evolución de las enfermedades y cómo reacciona un paciente a los tratamientos, permitiendo predecir o detectar más rápidamente otros problemas.

Otra posible aplicación es la capacidad de mejorar el estilo de vida de la población, consiguiendo crear chatbots con rutinas de cuidado físico y psicológico, todo ello siguiendo con la facilidad de disponibilidad de esta herramienta.

Comercio electrónico

Otro área que utiliza con gran frecuencia estos asistentes, es el comercio electrónico. Las oportunidades que puede ofrecer un chatbot de cara a un página de venta de productos en línea son muy elevadas.

Podemos encontrarnos agentes encargados de la asistencia técnica, como tiene *Amazon*, o los que realizan comparativas de productos con el único objetivo de facilitar al cliente la toma de decisiones, o encontrarnos otros mucho más complejos, como *SuperAgent* [6], un chatbot especializado en la búsqueda de información que al cliente le puede resultar relevante sobre el producto que está viendo en ese momento, usando todos los datos disponibles públicamente y a gran escala.

Marketing

Por último, el siguiente sector donde más se usan los chatbots es en Marketing [4].

Varias de las posibilidades que ofrecen los agentes se centran en la promoción de productos y en devolver la retroalimentación de los mismos, pero también son claves en el uso de las redes sociales, base fundamental del marketing en los tiempos actuales. Las funciones principales a las que se dedican este tipo de asistentes es a responder mensajes claros y fáciles

de interpretar, generar una nueva interacción cuando un producto ha sido lanzado al mercado, entre otras muchas actividades representativas.

Lo más conocido

Todo lo anterior nos vale como marca representativa en situaciones y campos específicos, donde son claros los objetivos de los usuarios que los utilizan, pero es cierto que actualmente, existen varios chatbots muy reconocidos que ofrecen una innumerable cantidad de recursos.

Entre estos están los siguientes: *Siri* [21], *Alexa* [8], *Google Assistant* [9], *Cortana* [10] y *ChatGPT* [11]. Debido a los mercados de hoy en día, todas las grandes empresas, desarrollan sus propios agentes, pero el objetivo que tienen todos es muy similar, facilitar las actividades cotidianas teniendo esta herramienta disponible en cualquiera de nuestros dispositivos digitales.

3.2. Dialogflow

Tal y como se ha explicado con anterioridad, Dialogflow es una plataforma de Google de creación y gestión de agentes² que basa su funcionamiento en el entendimiento del lenguaje natural.

En este apartado, se analizará la aplicación web de Dialogflow, realizando los correspondientes comentarios y razones por los que se podría mejorar.

Acceso

El acceso a la web oficial es un tanto confuso, ya que si hacemos la consulta en el buscador, el primer enlace, el que suele ser para cualquier web su página principal, nos encontramos, directamente, con la página de compra del producto.

²Término usado por Dialogflow para definir a los chatbots.

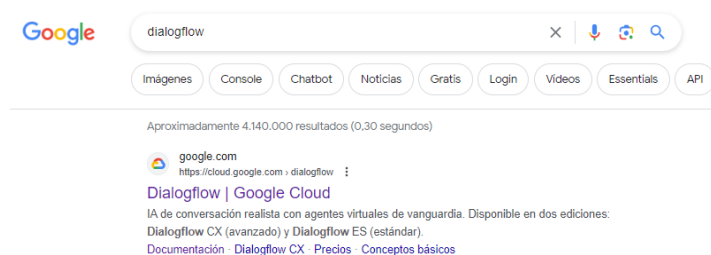


Figura 3.2: Búsqueda desde Google de Dialogflow [12]

No es hasta el apartado de ediciones, encontrado en la parte inferior, que se desvela el acceso a la versión gratuita de Dialogflow.

Esto ya es una desventaja. Un usuario espera hacer una sencilla búsqueda y encontrar el resultado con facilidad.

Interfaz principal

Una vez accedemos a la pantalla principal se nos muestra una interfaz de diseño sencillo. Las divisiones de las distintas partes, al estar en una tonalidad muy parecida, dificultan en leve medida su distinción.

Al iniciar el proceso de creación de un agente³, nos aparece una interfaz donde se nos pide indicar diferentes parámetros iniciales, entre ellos y del cual se habla en el apartado de “*Aspectos relevantes*”, la creación de un proyecto *Google Cloud*, pero no se explica claramente la función de este parámetro, ya que es probable que para el tipo de chatbot que nos permite crear esta versión gratuita, no sea relevante. Además de esto, como cada asistente va asignado a uno de estos proyectos de Google y se tiene una limitación a la hora de crearlos, suma otra desventaja. Añadir que el borrado de un proyecto de este tipo, en caso de necesitar crear un nuevo chatbot, conlleva esperar a que pase un tiempo antes de que este se haya desactivado y podamos generar uno nuevo.

Una vez hemos creado esta parte, llegamos al menú principal del agente, con estética idéntica al de la creación del agente o la del inicio, si no tenemos ningún chatbot creado.

³Término usado por Dialogflow para los chatbots.

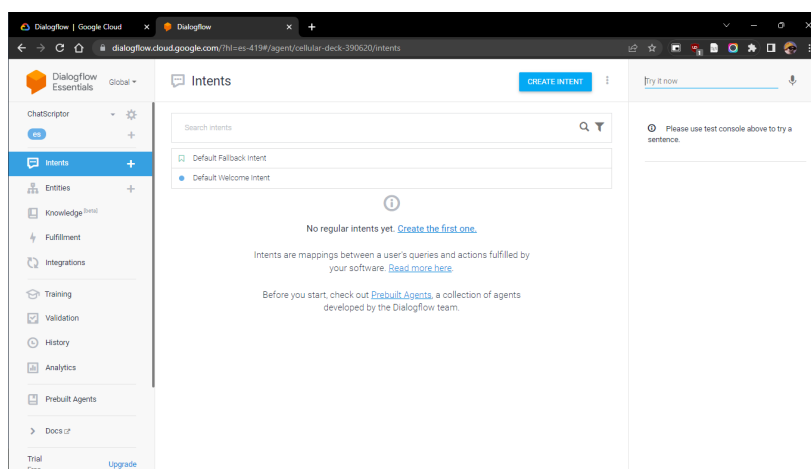


Figura 3.3: Primera pantalla después de crear un agente

En la barra de navegación de la parte izquierda, podemos ir accediendo a las entidades e intents de nuestro chatbot y para acceder a la configuración, a través del engranaje al lado del nombre.

Centrándonos en las partes importantes, veamos la creación de un nuevo intent y una nueva entidad.

- Entidad: este primer elemento permite establecer sinónimos de palabras o acciones claves. Su creación se centra en ir añadiendo filas a una tabla con los diferentes vocablos.

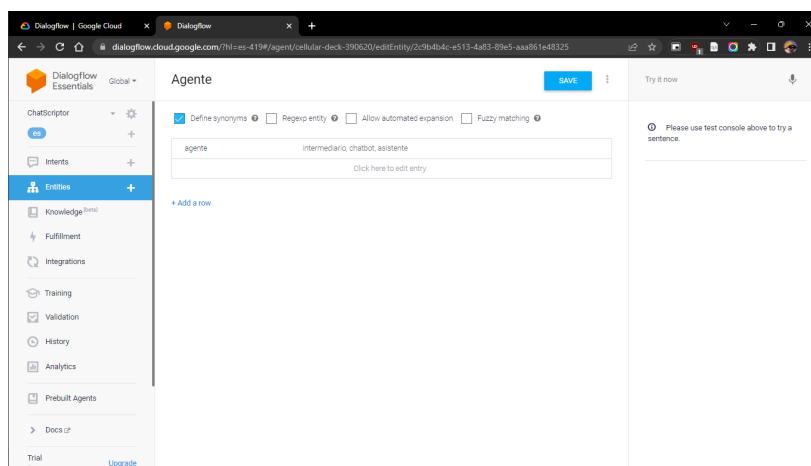


Figura 3.4: Creación de entidad

- Intent: es el lugar donde se establecen la frases de entrenamiento y las normas o reglas que sigue el chatbot para contestar. Su creación se basa en ir añadiendo dichas frases en tablas. Al contrario que en las entidades, crea uno por defecto con saludos iniciales de conversación.

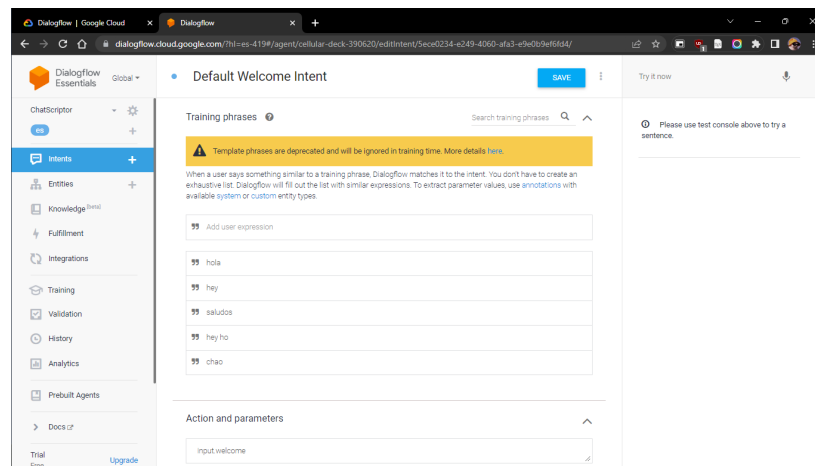


Figura 3.5: Frases de entrenamiento

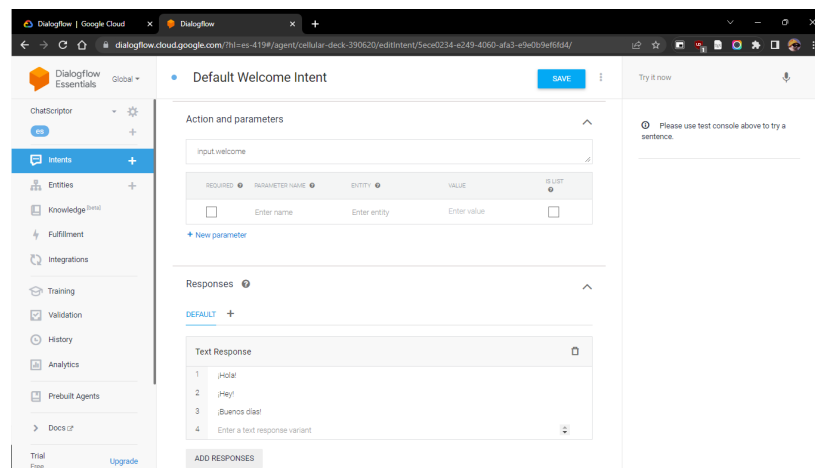


Figura 3.6: Respuestas

La parte negativa de esto, es la similitud de ambas interfaces, ya que esto confunde en que parte estás. Además, la gran cantidad de parámetros que están disponibles están poco explicados, lo que dificulta saber cuál es el efecto que tendrán sobre el agente. Es cierto que se dispone de una extensa

documentación [14], pero esto no es algo que se consulte con frecuencia por los usuarios y sería algo más claro si se añadiera una zona de parámetros más específicos de configuración que fuera ocultada hasta que el usuario quisiera especificarlos.

Siguiendo con la configuración (engranaje) del agente, es donde encontraremos los parámetros para establecer una imagen y descripción al asistente, así como la opción de importar, restaurar o exportar, entre otras. También posee un apartado de “*Language*”, esta funcionalidad está creada con el objetivo de poder tener el agente en varios idiomas, pero no funciona tan bien como debería. Este ha sido uno de los principales objetivos de este proyecto, crear una opción dentro de la aplicación que permita traducir el chatbot generando uno equivalente en el idioma deseado.

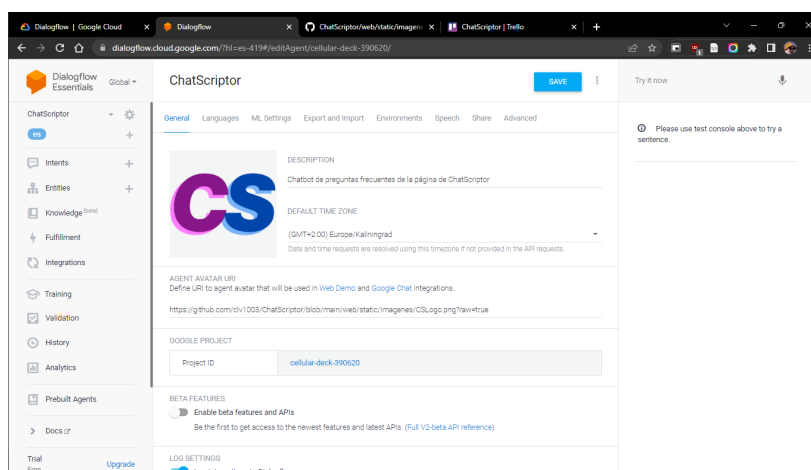


Figura 3.7: Interfaz de configuración del agente

Como último detalle, Dialogflow incluye buscadores tanto en entidades como en *intents*. Cuando se realiza una búsqueda en ellos, estos proporcionan una respuesta sobre los datos que se ven en pantalla. Esto provoca búsquedas algo engorrosas, puesto que si estamos buscando un término concreto que aparezca dentro de varias de las frases o de los sinónimos, no aparecerán y tendremos que ir uno por uno localizando lo que queramos cambiar. Al igual que con el traductor, este ha sido otro de los objetivos del trabajo.

Técnicas y herramientas

4.1. Técnicas

Metodologías

Scrum

Scrum [22] es un tipo de metodología ágil muy recomendada para el desarrollo de productos software. Se trata de trabajar de manera incremental, haciendo que cada una de las iteraciones contenga una parte funcional del desarrollo, estas iteraciones se denominan sprints.

Patrones de diseño

Model-View-Presenter (MVP)

Model-View-Presenter (MVP) [36] es un patrón de diseño que está enfocado en el desarrollo de interfaces de usuario. Esto facilita el diseño de las diferentes partes lógicas permitiendo estructurar el proyecto en módulos.

- Model (M) [36]: se trata de definir una interfaz donde se establece la información que se va a mostrar y cómo actuará dicha interfaz. Es el encargado de tratar con los datos usados en la aplicación.
- View (V) [36]: se trata de la interfaz pasiva que recibirá las ordenes y le transmitirá al presentador para que realice la acción, evitando la interacción directa con el modelo.
- Presenter (P) [36]: se trata del intermediario entre el modelo y la vista, se encarga de recibir las acciones a realizar, obtiene los datos necesarios del modelo y los muestra en la vista.

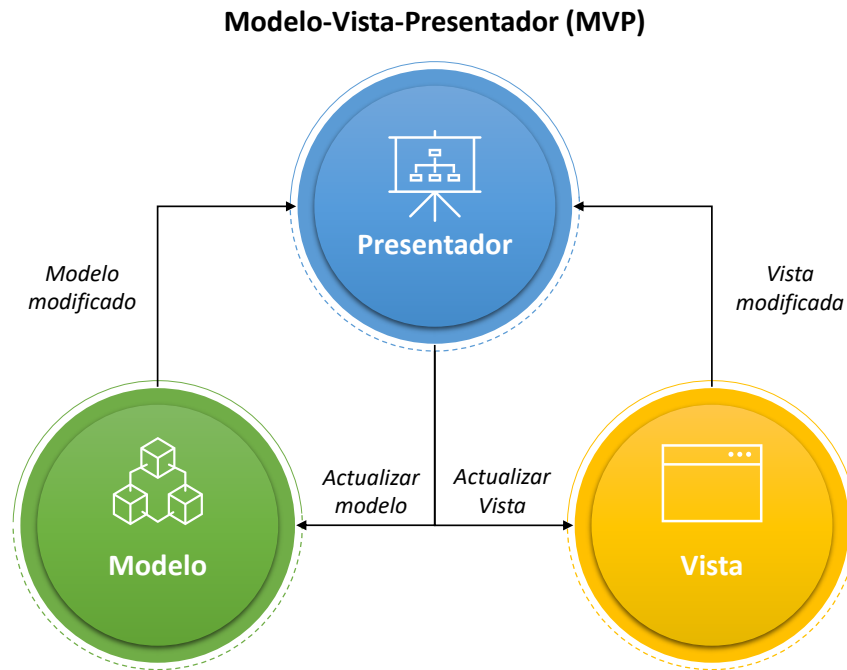


Figura 4.1: Patrón de diseño MVP [18]

Adaptador

El patrón de diseño software *Adapter* o Adaptador [1] es un patrón estructural que permite convertir dos interfaces que, inicialmente no son compatibles, en compatibles, permitiendo trabajar juntas. Esto provoca que podamos usar un intermediario (el adaptador) que sea el encargado de hacer que ambas se entiendan, con esto conseguimos crear clases reutilizables.

- Cliente: será el encargado de usar la interfaz “Adaptador” para interactuar con el “Adaptado”. No sabe de la existencia de “Adaptador”.
- Objetivo: interfaz que espera “Cliente” y que “Adaptador” implementa.
- Adaptado: es el elemento que necesita ser adaptado porque es incompatible por si solo.
- Adaptador: implementa a “Objetivo” y se encarga de hacer que las peticiones de “Cliente” y “Adaptado” se entiendan.

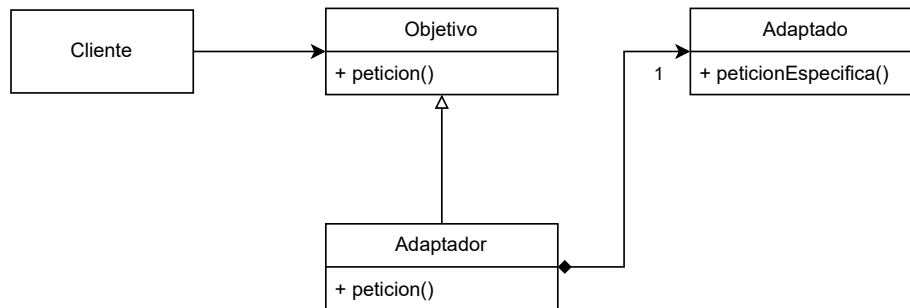


Figura 4.2: Patrón de diseño Adaptador [1]

4.2. Herramientas

Gestión del proyecto

Control de versiones

Para el control de versiones, se ha utilizado Git.

Git es un software gratuito y *open source* que permite llevar un registro de los cambios en un proyecto, archivo, etc. que tengamos en local, incluyendo la capacidad de trabajar junto a varias personas.

Además de esto, permite realizar y gestionar flujos de trabajo, el más conocido es Git-Flow.

Gestor del proyecto

A la hora de gestionar las tareas pendientes, cuáles están en proceso y cuáles están hechas, se ha utilizado el *software* Trello [16], una herramienta muy útil para estos casos y, que en su versión gratuita, ha permitido seguir fácilmente cada uno de los objetivos para cada sprint.

A continuación se muestra una prueba de cómo se ha utilizado:

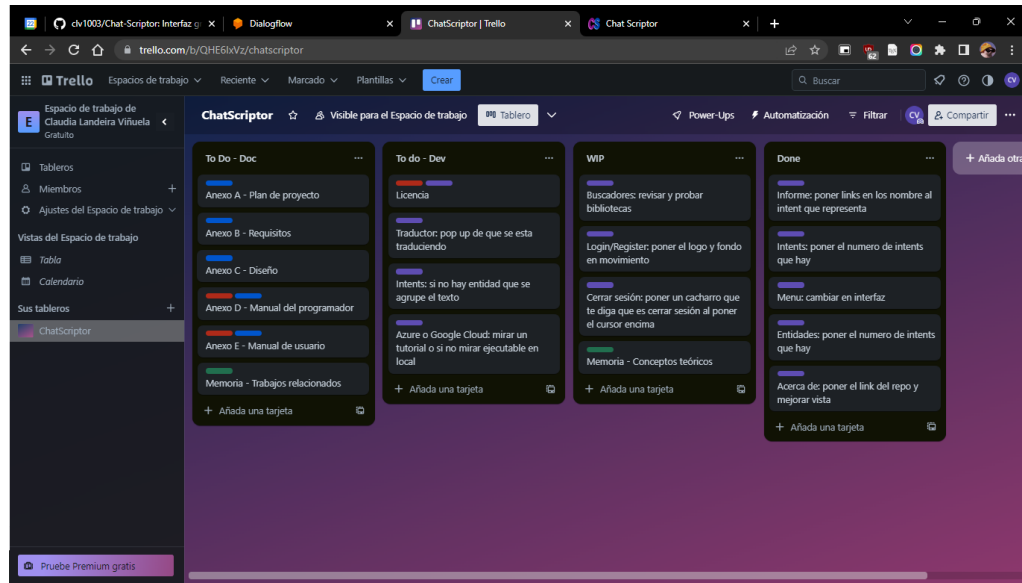


Figura 4.3: Interfaz de Trello [16]

Hosting del repositorio

Para hospedar el repositorio se ha usado GitHub (<https://github.com/>).

Github es la plataforma de hospedaje de repositorios más conocida. Permite usar las funcionalidades de Git, además de otras muchas que la hacen una de las mejores opciones para el desarrollo de proyectos software.

A continuación, se incluye el enlace directo al repositorio de este proyecto: <https://github.com/clv1003/Chat-Scriptor>

Entorno de desarrollo integrado

Python (IDE)

El entorno de desarrollo (IDE) seleccionado para el desarrollo del proyecto ha sido PyCharm Professional.

Esta decisión se tomó debido a la gran cantidad de opciones y funcionalidades que posee la herramienta. Además de tener un sistema propio de descarga de bibliotecas y *plugins* que permiten añadir todo aquello que se necesite sin necesidad de salir del producto.

También existe una versión gratuita con una menor cantidad de opciones, pero debido a la cuenta educativa de la Universidad, este proyecto se ha desarrollado en la versión completa.

Markdown

Los textos escritos con este lenguaje, no se han desarrollado con ningún software añadido, ya que, como se explicó en el apartado anterior, PyCharm Professional da la opción de editar este tipo de archivos y visualizarlos.

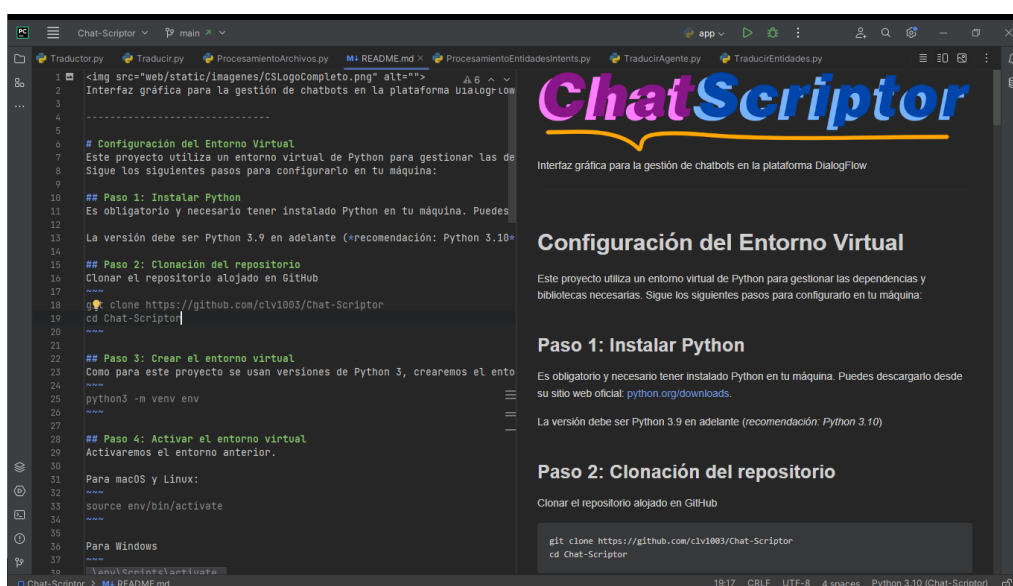


Figura 4.4: Interfaz para el uso de Markdown a través de Pycharm Profesional

L^AT_EX

Para escribir en L^AT_EX se ha optado por la herramienta Overleaf.

Overleaf es un editor gratuito online que permite escribir documentos con L^AT_EX, eligiendo el compilador así como el visor de PDF.

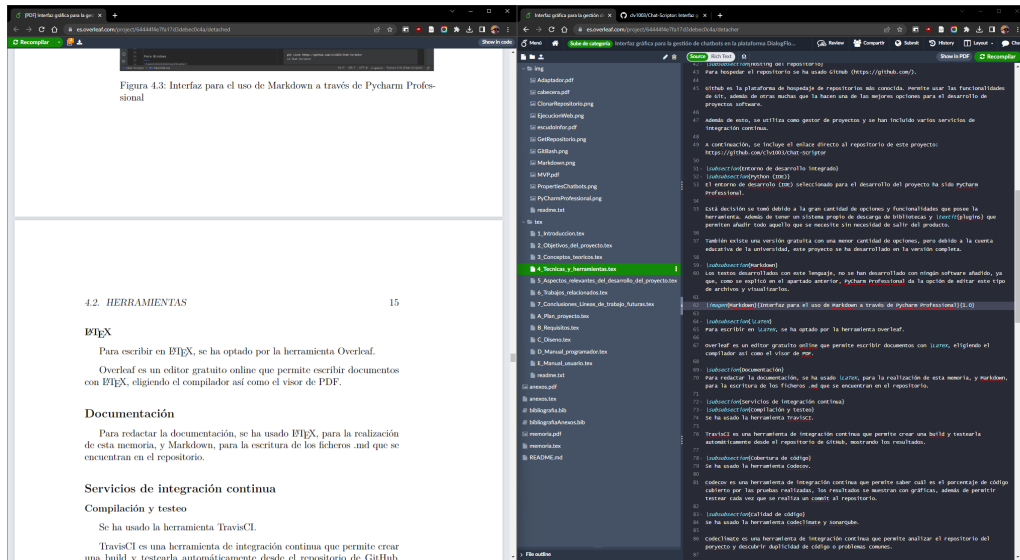


Figura 4.5: Interfaz para el uso de \LaTeX a través de Overleaf

Docker

Docker [28] es una herramienta de contenedores que permite el aislamiento de las estructura de código, facilitando la compilación, ejecución y producción de proyectos software.

Genera imágenes de contenedores que encapsulan todas las funcionalidades y dependencias, dando un elemento muy liviano y capaz de ser ejecutado en cualquier sistema operativo.

Documentación

Para redactar la documentación, se ha usado \LaTeX , para la realización de esta memoria, y Markdown, para la escritura de los ficheros .md que se encuentran en el repositorio.

Librerías

Todas las librerías descritas a continuación, se han incluido en el archivo *requirements.txt* del repositorio permitiendo una fácil instalación de las mismas, además de que, al tener un archivo *Dockerfile* se facilita aún más. Se incluye más información acerca de esto en el “Anexo D”.

- Bootstrap 5.3.0: utilizado para el diseño de interfaz.

- Bootstrap Icons: obtención de los iconos usados en la interfaz.
- os: gestión de directorios y archivos, operaciones de rutas y nombres de archivos, etc.
- Flask: framework web ligero flexible desarrollado en Python.
 - render_template: permite renderizar plantillas HTML.
 - url_for: permite generar URLs dinámicas.
 - redirect: permite redirigir al usuario hacia una nueva URL.
 - request: permite hacer solicitudes HTTP para recibir o enviar información, para más tarde hacer uso de la misma.
 - session: permite almacenar datos en forma de variables de sesión.
- json: funciones para trabajar con datos en formato JSON (codificar o decodificar archivos).
- re: amplia gama de funciones para trabajar con expresiones regulares.
- shutil: operaciones de gestión y manejo de archivos y directorios.
- csv: funciones para trabajar con datos en formato CSV.
- bcrypt: funciones para el cifrado y descifrado de datos, además de su verificación de forma segura.
- zipfile: funciones relacionadas con el tratamiento de ficheros ZIP (compresión y descompresión de archivos).
- transformers, torch, torchvision, sentencepiece, sacremoses: combinación de librerías que permiten usar los modelos de lenguajes de traducción.
- waitress: librería encargada de generar servidores HTTP muy ligeros y fáciles de utilizar incluso en productivo.

Página web

Bootstrap y Bootstrap Icons

Bootstrap [17] es un framework para diseño de interfaces. Contiene una serie de estilos que permiten ser usados directamente sobre las páginas HTML, que junto a Bootstrap Icons y su extensa documentación, agiliza el proceso de desarrollo *front-end*.

Microsoft Azure

Microsoft Azure [13] es un producto de Microsoft centrado en el servicio en la nube que ofrece una gran variedad de funcionalidades de desarrollo, administración y despliegue de aplicaciones web.

Es por esto, que al tener licencias gratuitas de la cuenta de estudiantes y su gran catalogo de servicios, se ha decidido usar para el despliegue de la aplicación web.

Otras herramientas

Draw.io

Draw.io [15] es una herramienta online que permite crear cualquier tipo de diagrama necesario con gran cantidad de personalización, además de ser un software con opciones gratuitas muy completas.

Postman

Postman [20] es una herramienta que realiza peticiones a APIs, permitiendo comprender el comportamiento de las mismas, así como saber si ese tipo de *requests* funcionan con las claves o *tokens* que hayamos asignado a dichas peticiones.

Aspectos relevantes del desarrollo del proyecto

En este apartado, se va a comentar todo lo relacionado con el proyecto y su realización. Desde cómo se ha empezado, las decisiones tomadas, los problemas encontrados, etc.

5.1. Inicio del proyecto

La idea de poder realizar un proyecto de desarrollo software y más concretamente del diseño de interfaces siempre ha sido una de las cosas que más me ha llamado la atención desde que empecé mis estudios.

Esta es la razón principal por la que, en el momento en el que me puse a buscar un tema para el proyecto, este fue el elegido.

Una vez se dio el visto bueno y se me otorgó el tema, comenzó el proceso de desarrollo.



Figura 5.1: Logo ChatScriptor

5.2. Metodologías

A la hora de poder mantener una organización y un buen proceso de desarrollo, tal y como se nos enseñó en algunas de las asignaturas de la carrera, se estableció desde el principio el uso de metodologías ágiles. Se debe tener en cuenta que, al tratarse de un proyecto en el que no interviene un gran equipo, esta metodología se ha tratado de forma flexible.

¿Cómo se ha aplicado?

- Se estableció una fecha objetivo para la finalización del proyecto.
- Se estableció la realización de *sprints* cada dos semanas, aunque una vez se acercaba la fecha de objetivo, se incrementaron haciéndolas cada semana.
- En cada fin de *sprint*, se entregaba una parte incremental funcional del proyecto, se realizaba una reunión de revisión y, en la misma, se establecían las tareas para la siguiente iteración del proyecto.
- Una vez se establecían dichas tareas, se ha realizado un tablero nuevo en *Trello* de tareas a completar, ordenadas con una plantilla de colores que ha determinado su prioridad.

5.3. Formación

Para poder realizar el proyecto, se han utilizado muchos de los conceptos aprendidos a lo largo de mi trayectoria académica en la Universidad de Burgos, pero respecto al desarrollo web, ha habido una gran parte del tiempo invertido en el auto-aprendizaje e investigación personal.

Se han realizado pruebas de proyectos de desarrollos web con la idea de conocer el funcionamiento antes de comenzar con el trabajo final. Además de investigar entre varias comunidades como *StackOverflow* o tutoriales en la plataforma de *Youtube*, para adquirir ciertas habilidades.

5.4. Desarrollo del proyecto

En esta sección, se explicarán los diferentes puntos a completar para conseguir cumplir los diferentes objetivos propuestos para el proyecto. Siendo los problemas generales que se han querido solventar:

1. **La visualización de los datos del agente, las entidades y los *intents*.** Debido a la similitud de las pantallas en la interfaz oficial, se ha querido implementar una visualización de los datos de tal forma, que el usuario pueda tener toda la información lo más limpia posible.
2. **En el caso de querer buscar una entidad o *intent* para ver o cambiar su información, se debe ir accediendo a cada una individualmente.** Es por esto, que se ha querido desarrollar una serie de buscadores que permitan obtener la información rápidamente, ofreciendo una función que antes no existía.
3. **Obtener un informe representativo de los *intents*.** Esto permitiría buscar y poder obtener las frases de entrenamiento a golpe de vista, permitiendo al usuario obtener fácilmente una vista de las diferentes opciones del chatbot.

Se comenzó interactuando con la página como si de un usuario que quiere crear chatbots para uso personal se tratase. Con esto, se obtuvieron los primeros contactos e impresiones para realizar una exhaustiva investigación, así como la lectura de las documentaciones.

El primer paso en el desarrollo, fue crear una interfaz que fuera capaz de mostrar la información de los chatbots a través del tratamiento de sus archivos *json* obtenidos de la exportación de la página oficial. Con esto, se pudo planificar la estructura inicial de la página, así como la eliminación de toda la información irrelevante para el usuario.

Una vez, este prototipo se desarrolló y se le incluyó la capacidad de visualización, se agregaron funcionalidades y se eliminaron ciertas opciones a causa de las restricciones presentadas por Google ante la creación de identificadores propios, desarrollado más adelante en el apartado de “*Aspectos relevantes del proyecto*”.

Cuando las funciones básicas estaban ya cubiertas, se trabajó en la implementación de los buscadores, estableciendo que, dependiendo de la zona en la que se encontrara el usuario, el buscador mostraría las coincidencias dentro de este. Haciendo que si buscásemos un término en la pantalla inicial de selección de chatbots, buscaría entre todos los chatbots y, si estuviéramos en el agente de uno concreto, nos mostraría solo las coincidencias dentro del agente. Se pensó que esta sería una buena forma para mejorar la capacidad de la interfaz original para que el usuario pudiera buscar, ver y editar la información que deseara.

Al finalizar la implementación de las opciones principales, se llevó a cabo la inclusión de un traductor que posibilitara la traducción de los chatbots. Se pensó que sería una funcionalidad interesante que permitiría que los chatbots fueran multilingüaje, convirtiéndolos en un producto de gran adaptabilidad a las diferentes situaciones donde fueran a ser usados.

Una vez se implementó el traductor basado en modelos de NPL (*Natural Processing Language*) de “*Hugging Face*” [19], se realizaron un gran número de pruebas, intentando mejorar todas las funcionalidades e incluso incrementar su eficiencia.

Cuando se obtuvo un resultado final funcional, se realizó el despliegue mediante los servicios de *Microsoft Azure*, pudiendo tener ChatScriptor en productivo: <https://chatscriptor.azurewebsites.net/>

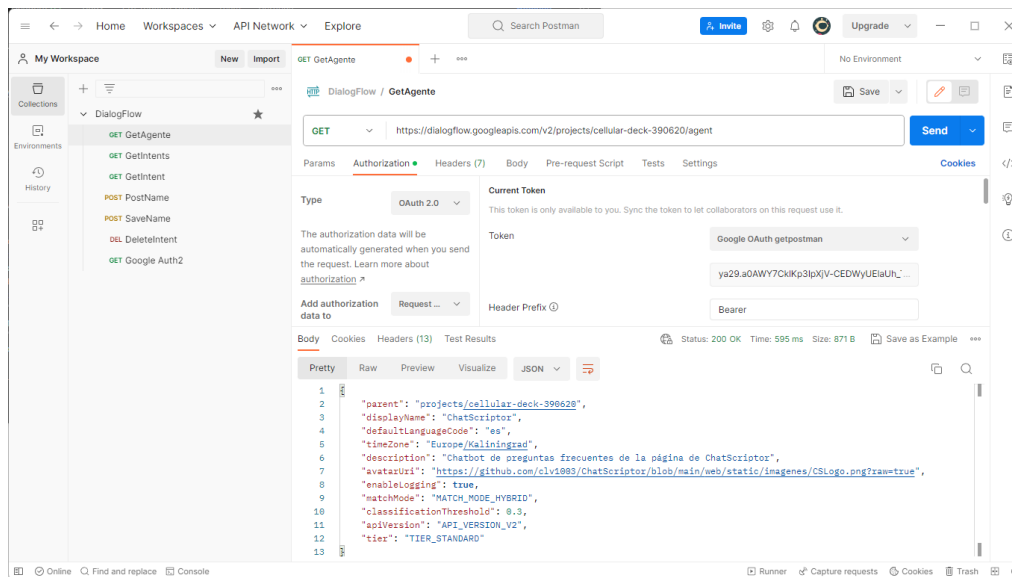
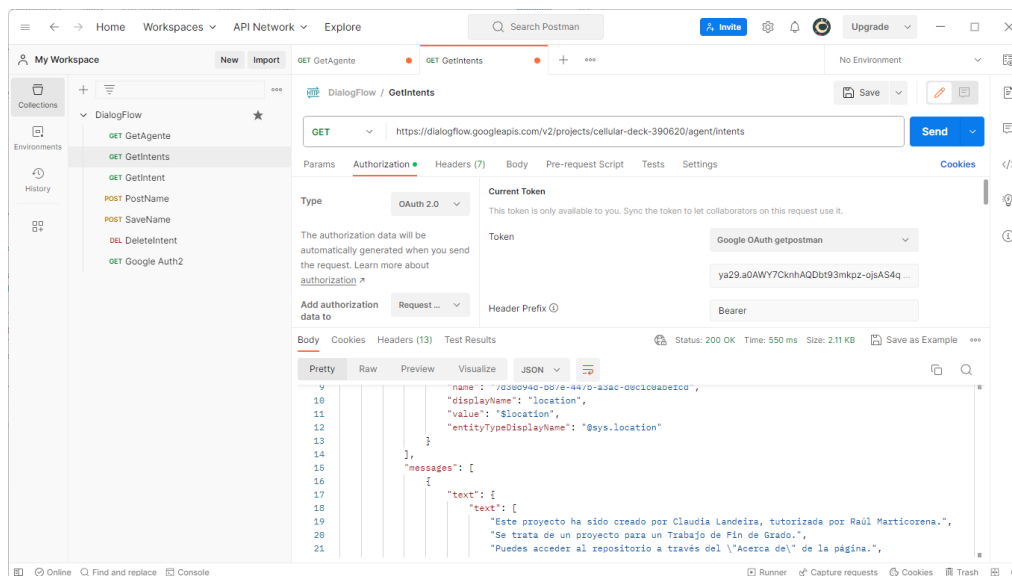
5.5. Problemas encontrados

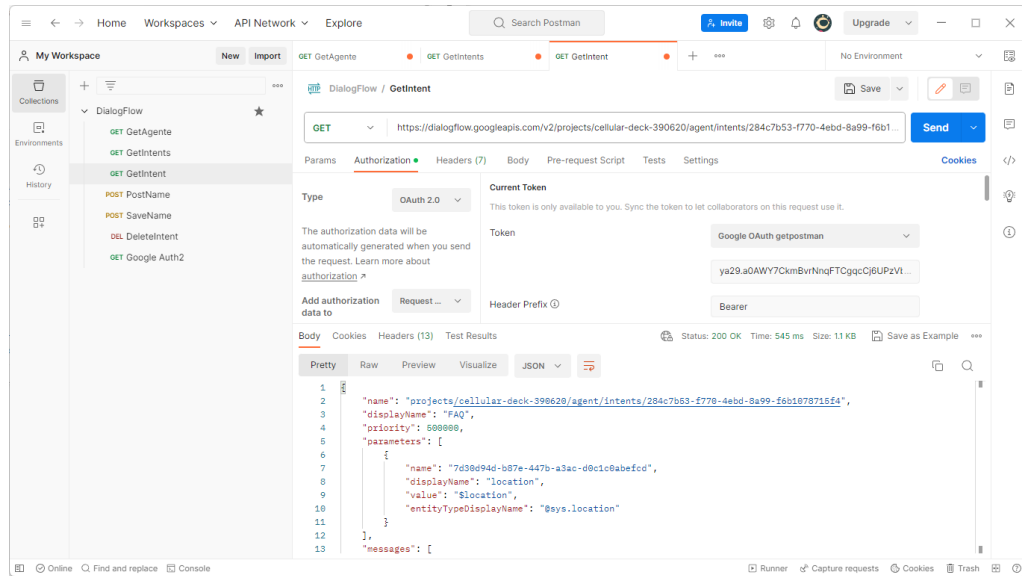
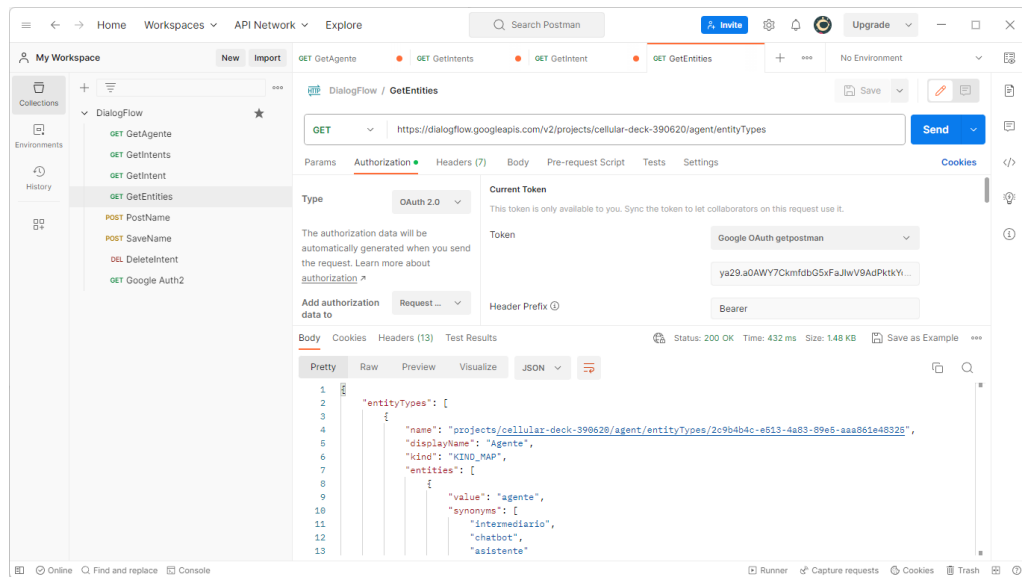
Como es habitual en el desarrollo de un proyecto, no todo lo que se planea se puede llevar a cabo.

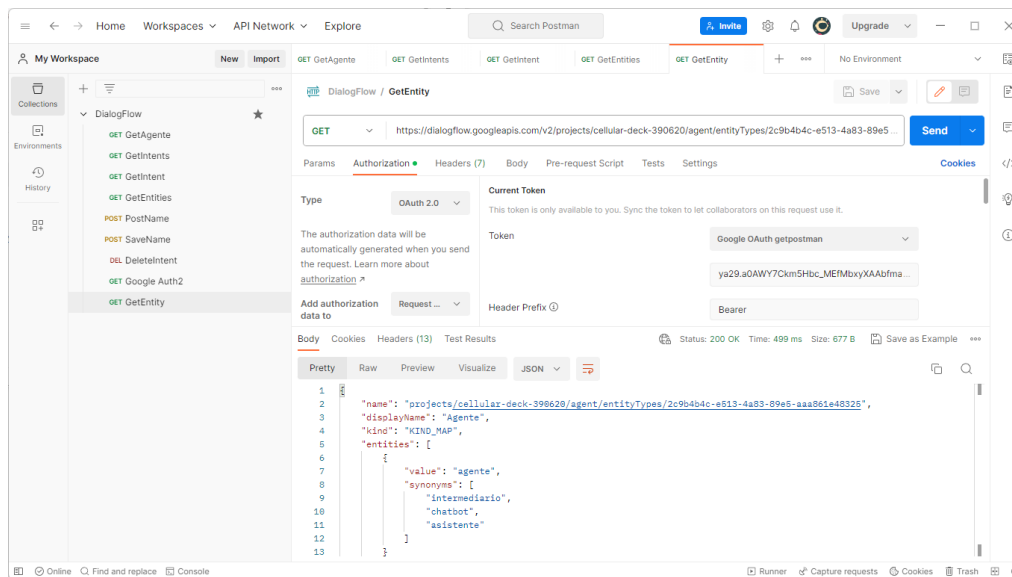
Interacciones con la API de Dialogflow

Una de las primeras ideas que se barajaron para desarrollar la web fue el uso de la API de Dialogflow para realizar las peticiones de información de los chatbots. Para esto, se quiso establecer un inicio de sesión con la cuenta de Google asociada a la cuenta de Dialogflow que permitiera el acceso a los chatbots de ese usuario. El primer problema fue este mismo inicio de sesión.

Se probó si esto era posible. Se realizaron peticiones a la API, haciendo uso de *Postman* [20]. Los resultados fueron satisfactorios para chatbots de prueba, para los cuales se obtuvo su “token” y su autenticador *OAuth2* de forma manual. A continuación, se muestran unas capturas (ver figuras 5.2, 5.3, 5.4, 5.5 y 5.6) de ejemplo:

Figura 5.2: Petición *GET* del agenteFigura 5.3: Petición *GET* de los intents

Figura 5.4: Petición *GET* de un intentFigura 5.5: Petición *GET* de las entidades

Figura 5.6: Petición *GET* de una entidad

El siguiente paso fue la implementación en la propia aplicación web. Para conseguir acceso a estas interacciones vía API de forma automática, se investigaron las diferentes documentaciones que ofrece Google junto con otras encontradas de personas que en sus proyectos usaban esta API. El proceso consiste en lo siguiente:

- Acceder a la página de la consola de Google Cloud, uno de los múltiples servicios que ofrece Google, y seguir los pasos para crear un proyecto.

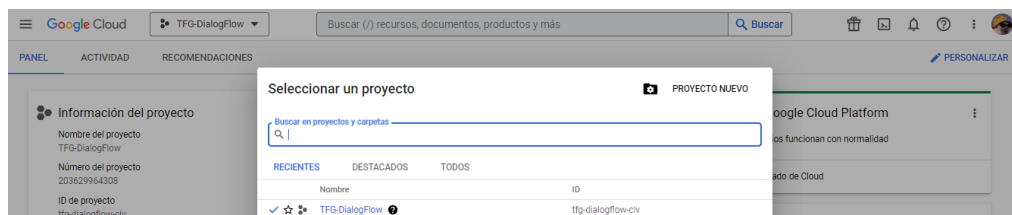


Figura 5.7: Crear proyecto en la consola de Google Cloud

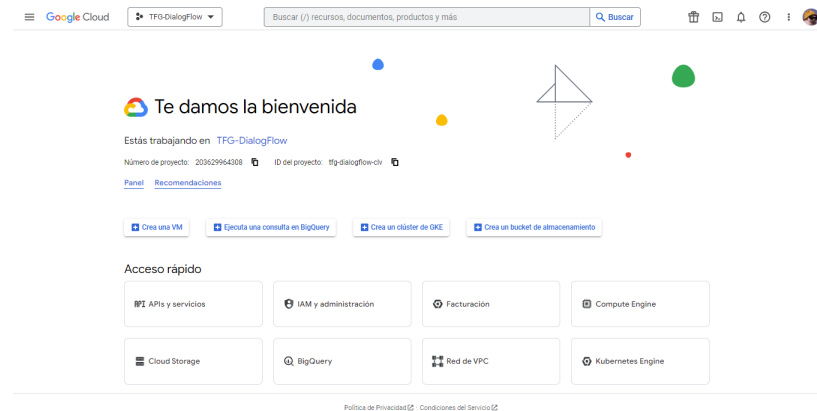


Figura 5.8: Panel principal del nuevo proyecto

- Añadir las APIs que se quieran usar y habilitarlas. En este caso, la API de Dialogflow.

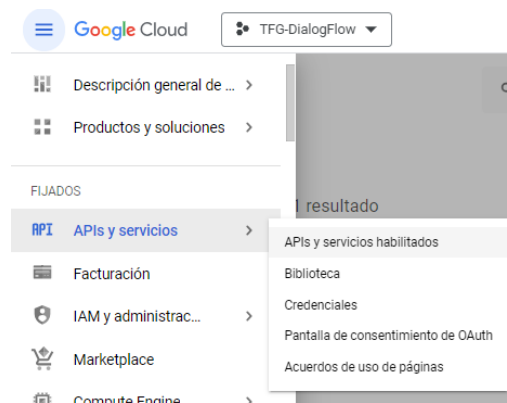


Figura 5.9: Acceso a las APIs y servicios

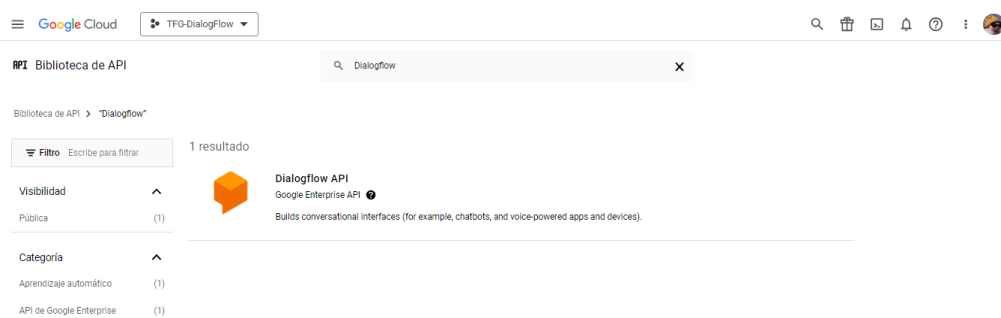


Figura 5.10: Buscar y habilitar la API de Dialogflow

- Crear las credenciales de acceso que se introducen en el código del proyecto.

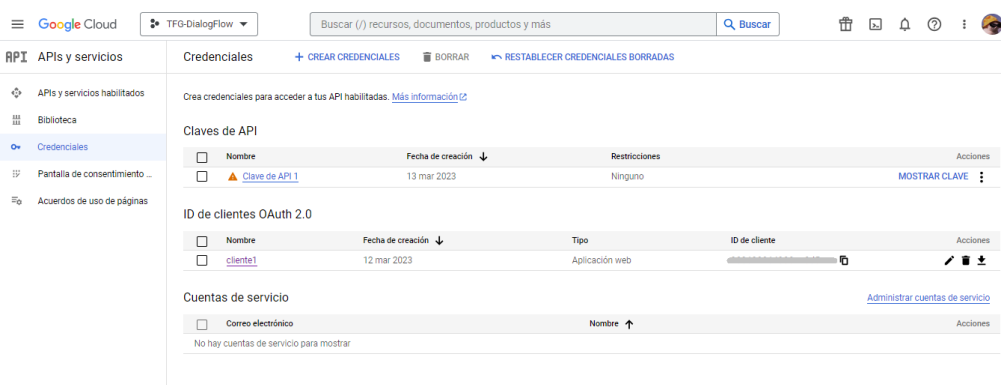


Figura 5.11: Credenciales para el proyecto

Con el proceso anterior, se consiguió la obtención de las credenciales.

Otro problema que se encontró al terminarlo, fue el añadir el acceso a través del inicio de sesión de Google, ya que muchas partes de la documentación tenían versiones obsoletas o se estaban sustituyendo, produciéndose un problema de compatibilidad. Aunque después de buscar y probar muchas combinaciones, se encontró la versión correcta.

Una vez se pudo acceder, se realizaron pruebas con la misma cuenta asignada a la creación del proyecto en la consola de Google Cloud, esto provocó que se evitara lo que más tarde hizo que no se pudiera implementar el uso de la API.

El problema fue el siguiente: las opciones que ofrece Google para el uso de sus APIs van ligadas a un proyecto al que solo tiene acceso la cuenta que ha creado el proyecto. Visto de otra manera, si se usara un usuario diferente para iniciar sesión a la que está establecida en el proyecto, no se podría acceder.

Se buscaron varias alternativas para solucionar esto, como crear una parte en la guía de usuario que explica que como crear su proyecto con sus credenciales y modificar el código de tal forma que introduciendo dichas credenciales en la interfaz, pudieras acceder. Se llegó a la conclusión de que esto no era práctico ni mejoraría la experiencia de usuario. Además, la API está diseñada con el objetivo de que puedas personalizar un chatbot completamente, configurando campos que la web oficial no permite.

La solución elegida fue desarrollar una versión a través de la exportación de los chatbots desde Dialogflow. A partir de aquí, se trabajó con los ficheros JSON.

Estructura y archivos *JSON*

El siguiente problema encontrado, fue la estructuración de los directorios y ficheros *JSON*. Que presentan la siguiente estructura genérica:

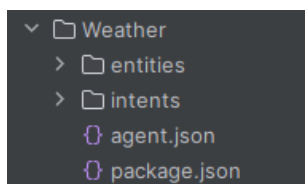


Figura 5.12: Estructura de directorios de un chatbot ejemplo

Dialogflow te muestra los datos en archivos *JSON*, donde en varias ocasiones se pueden encontrar listas con diccionarios que poseen más listas anidadas, formando archivos difíciles de tratar. Añadido a esto, dentro de los directorios que contienen entidades e *intents* (*entities* e *intents*), nos encontramos con otra estructuración compleja que divide los archivos de cada entidad y cada *intent* en dos partes que contienen información diferente.

Este análisis, es también la solución, ya que se obtuvieron los puntos relevantes y la forma de unificar los archivos, determinando la información que se ha terminado mostrando por pantalla.

Generación de identificadores

El siguiente problema descubierto, está relacionado con la adición de información en aquellos zonas que necesitan la generación de identificadores.

Como posible solución, se intentó implementar un método de generación de identificadores o *IDs* con la misma estructura con la que los creo el propio Dialogflow. A pesar de que dichos IDs no coincidían con los originados por la aplicación oficial, al intentar importar el chatbot con las partes nuevas en la aplicación oficial, nos indica que algo falla al cargar el agente. Por esta razón, la capacidad de añadir la información se ha reducido a aquellos lugares que no dependan de identificadores.

Traductor

Respecto al traductor, han surgidos diversas complicaciones. Se ha tratado de implementar mediante APIs de diferentes traductores, siendo el principal problema el tamaño de los chatbots (presentan una gran variabilidad) y las limitaciones de los propios traductores (en cuanto a opciones y número de palabras). Por lo que se tuvieron que buscar alternativas.

La solución encontrada ha sido el uso de modelos entrenados desarrollados en “*Hugging Face*” [19] (<https://huggingface.co/>).

Estos modelos permiten realizar la traducción a un idioma, teniendo en local los archivos del idioma. Esto nos da la ventaja de que la traducción no depende de algo exterior al proyecto, pero al mismo tiempo, genera que aumente el tamaño del proyecto, lo que genera otro tipo de problemas.

Despliegue de la aplicación web

El problema que genera el uso de los modelos de traducción es el aumento de tamaño del proyecto, tal y como se ha explicado anteriormente. Esta es la razón por la cual se ha tenido que extender la búsqueda del servidor de despliegue a aquellos que posean un límite amplio de capacidad de almacenamiento.

Durante la investigación, se barajaron varias posibilidades como los servicios del propio Google, el despliegue en GitHub (que está limitado a aplicaciones estáticas), Heroku, entre otros muchos. De entre estas opciones apareció la idea de usar la cuenta educativa en los servicios de Azure [13], esperando conseguir alguna alternativa, ya que sus servidores en la versión gratuita están muy limitados, pero al final esta resultó ser la solución.

Debido a dicha cuenta educativa, *Microsoft Azure* ofrece 100\$ para la realización de pruebas en todos sus servicios. Con esto, se pudo elegir un plan que permitiera tener el almacenamiento suficiente.

Opciones de importación y restauración en Dialogflow

Una vez se realizan todos los cambios pertinentes del agente en ChatScriptor y queremos volverlo a importar en Dialogflow, se ha descubierto que tanto la opción de importación como de restauración ofrecidas no tiene un funcionamiento pulido.

Varios de los problemas son la falta de carga de los archivos *JSON*, es decir, algunas partes de los mismos no las interpreta y no las cambia. Algunos ejemplos de esto son la carga de la imagen del agente, así como las respuestas de los *intents*, punto muy relevante para el chatbot.

5.6. Testing

A la hora de hacer pruebas sobre la aplicación, surgieron una serie de cuestiones a tener en cuenta.

Al no poder utilizar la API de Dialogflow para realizar las peticiones, se tuvo que cambiar la forma de pensar de todo el funcionamiento de la aplicación web y con ello la forma de probarla. Añadir a esto, que tampoco era posible añadir información en todos aquellos lugares donde había un identificador. Por consiguiente, ha sido necesario tener unos chatbots de prueba creados desde Dialogflow para poder añadirlos y hacer las pruebas.

Todo esto, ha llevado a que el proceso de pruebas de cada una de las funciones y pantallas se haya realizado de forma manual. Estas pruebas se realizaron por pares (auto-test y por parte del tutor del proyecto), con chatbots tanto de ejemplo que permite crear Dialogflow, como por chatbots creados para uso personal.

Todas las pruebas, desarrolladas en el “*Anexo D. Manual del programador*”, se han realizado tanto en desarrollo (a través de un servidor local) y a través del servidor en producción.

5.7. Documentación

Desde la primera reunión, se ha establecido que la memoria se haría con la plantilla en *L^AT_EX*, utilizando el editor online *Overleaf*.

Para la toma de notas de las reuniones y otros apuntes que se incluyen en esta memoria, se utilizó *Microsoft OneNote*.

5.8. Uso en producción

A la hora de poner este proyecto en producción, se han analizado muchas de las opciones más conocidas para hospedar aplicaciones web. Muchas de ellas proporcionan servicios gratuitos muy básicos, que para páginas web estáticas o para pequeños proyectos son interesantes.

El problema que se ha encontrado, explicado en apartados anteriores, ha sido el almacenamiento. Los modelos de traducción usados ocupan un amplio lugar en disco, por lo que es necesario que el servidor donde se despliegue la aplicación tenga la posibilidad de almacenar el traductor y todos los chatbots con los que trabaja cada usuario que, a pesar de ser archivos de pequeño tamaño, si nos ponemos en la situación de tener numerosos usuarios con varios chatbots cada uno, podría llegar a provocar problemas de capacidad. Para reducir el tamaño, se tomo la decisión de hacer el despliegue utilizando contenedores Docker.

La solución que se ha encontrado a los problemas de almacenamiento de este tipo, a largo plazo, consiste en establecer fechas de caducidad para los chatbots que lleven un determinado tiempo sin ser usados, con previo aviso al dueño de los mismos. Esta idea no será desarrollada a fechas de entrega, pero formará parte de las “*Líneas de trabajo futuras*”.

La página resultante se encuentra en la siguiente dirección: <https://chatscriptor.azurewebsites.net/>

Añadir, que Azure limita los *timeouts* de las peticiones del servidor a cuatro minutos no modificables [2] (obligan a adquirir un plan de pago más elevado), por lo que de forma productiva está reducido su uso a chatbots de pequeño tamaño.

Trabajos relacionados

En este apartado, se espera encontrar proyectos y trabajos semejantes al realizado. Aunque tras llevar a cabo una exhaustiva búsqueda, no se ha encontrado ningún trabajo similar.

Por lo que el propósito de las siguientes divisiones es el de mostrar trabajos sobre los que influye el desarrollo de esta mejora de interfaz y de funcionalidades con respecto a Dialogflow.

Se han encontrado diferentes trabajos que utilizan la API Dialogflow, junto con otras funcionalidades, con el fin de crear agentes con funcionamientos concretos. Algunos de los cuales se describen a continuación.

El primero de ellos, es “*Desarrollo de un Chatbot con Dialogflow en el Marco de las Ciudades Inteligentes*” realizado por Sergio Francisco Iáñez González [27].

Este escrito, muestra la aplicación de un agente conversacional creado con Dialogflow en el ámbito de las ciudades inteligentes, así como sus funcionalidades dentro de la sociedad actual y su interacción con la misma sin necesidad de pasar por las grandes curvas de aprendizaje que, en algunas situaciones, representa el uso de este tipo de herramientas.

El siguiente trabajo encontrado fue el titulado “*Interfaz conversacional en Dialogflow para recomendación de películas*” realizado por Carlos Magán López [31].

En este caso, se trata de la elaboración de un asistente capaz de realizar recomendaciones de películas basándose en el uso de algoritmos de recomendación, como filtros colaborativos, utilizando APIs y otras técnicas de *web*

*scraping*⁴ que, unidos a la interfaz conversacional, permiten dar al usuario una serie de elementos cinematográficos adaptados a sus gustos.

Otra publicación encontrada fue “*Diseño y desarrollo de un asistente interactivo transaccional para la reserva de hoteles*” realizado por Beatriz Soro Vegas [34].

Esta memoria, tal y como viene representado en el título, tiene como objetivo desarrollar un agente que permita realizar reservas de alojamientos a los usuarios, añadiendo las funcionalidades de diferentes procesamientos de pagos, haciendo uso de diversas APIs con esas aplicaciones.

La cuestión final es saber en qué se relacionan estos trabajos con este proyecto. La realidad es que al ser agentes creados desde la plataforma de Dialogflow, existe la posibilidad de exportarlos e importarlos en nuestra aplicación, permitiendo utilizar y configurar estos asistentes virtuales, traducirlos o disponer de una mayor claridad a la hora de buscar información que se quiera editar.

Tal y como se explicó en el previo análisis de la aplicación web de Dialogflow, dentro del apartado de *Conceptos teóricos*, concretamente en el punto 3.2 *Dialogflow*, sus buscadores son muy poco prácticos y, para este caso de desarrollos, que poseen una gran cantidad de entidades e *intents*, ir comprobando en cuál de ellos se encuentra el término que quiero modificar, es algo increíblemente negativo.

Además de que la adición de lenguajes en el chatbot puede llevar a tener traducciones semánticas erróneas ya que se queda un poco justa, mientras que si traducimos directamente el chatbot completo con ChatScriptor, debido al uso de tecnologías como el *Natural Language Processing* (NLP), las frases son traducidas siguiendo la intención de dicha oración, interpretándola y buscando la mejor forma de expresarla.

⁴El *web scraping* [23] es una forma de extracción de información de páginas web mediante software.

Conclusiones y Líneas de trabajo futuras

En este punto de la memoria, se tratan las conclusiones obtenidas de la realización de este proyecto, junto con las posibles opciones a añadir que no se han implementado a fecha de la presentación de este proyecto y que quedan abiertas a continuidad.

7.1. Conclusiones

Las conclusiones obtenidas del desarrollo del proyecto son las siguientes:

- El objetivo principal del proyecto, crear una interfaz de Dialogflow más amigable que la que ofrece Google, se ha cumplido satisfactoriamente. Los usuarios que quieran gestionar los chatbots creados con Dialogflow, podrán hacerlo de forma eficiente y sencilla.
- Al usar un lenguaje de programación tan extendido como es Python, ha permitido tener una gran diversidad de opciones, tanto en su desarrollo, como a la hora de elegir entre infinidad de librerías que le otorgan una gran flexibilidad al proyecto.
- El desarrollo de este proyecto ha permitido utilizar la mayor parte de los conocimientos estudiados en la carrera, además de incrementar los relacionados con desarrollo software, diseño de interfaces o procesadores del lenguaje, entre otros.
- Gracias a la investigación realizada sobre los chatbots, se ha podido comprender la inteligencia artificial y los procesadores de lenguaje, así

como determinar la forma en la que ambos campos interactúan para ofrecer a los usuarios este tipo de tecnologías que facilitan muchas tareas.

- El uso de metodologías ágiles durante todo el proyecto, ha permitido mantener una organización en todas las tareas a completar, estableciendo objetivos en periodos cortos de tiempo, haciendo mucho más eficiente, flexible y sencillo cumplirlos.
- Durante todo este trabajo, se han usado varias metodologías y herramientas que han fomentado la calidad y el correcto funcionamiento del proyecto. Añadir, que algunas de ellas han llegado a crear sobrecarga, tanto funcional, como de usabilidad de las mismas, pero es cierto, que todas ellas han permitido que el proyecto haya salido adelante.

Como conclusión personal, este proyecto ha cumplido con las expectativas y los objetivos personales que se habían establecido inicialmente. No ha sido un camino sencillo, pero ha llevado a obtener un resultado digno del esfuerzo entregado.

7.2. Líneas de Trabajo Futuras

En cada proyecto que se crea y más hablando de un desarrollo software, quedan muchos frentes abiertos por los que se podría continuar, ya que siempre es posible añadir nuevas funcionalidades, mejoras o extensiones. A continuación, se muestran algunas de estas ideas:

- Migración de ChatScriptor a otra plataformas como aplicación de escritorio o aplicación móvil, entre otras.
- Añadir más modelos de traducción estudiando mecanismos para que no se sobrecargue la aplicación.
- Poder obtener un nuevo servicio que pueda almacenar mucha más información y permita que sea mucho más eficiente que el actual.
- Para facilitar el almacenamiento de información, desarrollar una funcionalidad que permita añadir una “fecha de caducidad” para que cuando un chatbot lleve almacenado mucho tiempo y no se haya usado, este se elimine, enviando una copia del chatbot al correo del propietario e informándole de la eliminación del mismo del servidor.

- Teniendo los datos de los números de frases de entrenamiento, entidades y sus respectivas respuestas, añadir un informe estadístico con los datos de todos ellos.
- Añadir filtros de ordenamiento en las páginas de selección de chatbots, entidades e intents que faciliten la organización de los mismos.
- Realizar, mediante *web scraping*, la obtención del historial desde Dialogflow, permitiendo realizar un informe estadístico de los usos de los agentes.

Bibliografía

- [1] Jesús Alonso Abad. Tema 2 - adaptador. Diseño y mantenimiento del software. Universidad de Burgos.
- [2] Microsoft Azure. Application performance faqs - azure | microsoft learn. <https://learn.microsoft.com/en-us/troubleshoot/azure/app-service/web-apps-performance-faqs#why-does-my-request-time-out-after-230-seconds>.
- [3] Himanshu Bansal and Rizwan Khan. A review paper on human computer interaction. *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, 8(4):53, 2018.
- [4] Ayşe BARIŞ. A new business marketing tool: chatbot. *GSI Journals Serie B: Advancements in Business and Economics*, 3(1):31–46, 2020.
- [5] Madeleine Bates. Models of natural language understanding. *Proceedings of the National Academy of Sciences*, 92(22):9977–9982, 1995.
- [6] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, system demonstrations*, pages 97–102, 2017.
- [7] Sam Cunningham-Nelson, Wageeh Boles, Luke Trouton, and Emily Margerison. A review of chatbots in education: practical steps forward. In *30th annual conference for the australasian association for engineering education (AAEE 2019): educators becoming agents of change: innovate, integrate, motivate*, pages 299–306. Engineers Australia, 2019.

- [8] Autor desconocido. Amazon alexa official site: What is alexa? <https://developer.amazon.com/es-ES/alexa>.
- [9] Autor desconocido. Asistente de google: tu google personal. https://assistant.google.com/intl/es_es/.
- [10] Autor desconocido. Ayuda y aprendizaje de cortana. <https://support.microsoft.com/es-es/cortana>.
- [11] Autor desconocido. Chatgpt. <https://openai.com/chatgpt>.
- [12] Autor desconocido. Dialogflow | google cloud. <https://cloud.google.com/dialogflow?hl=es-419>.
- [13] Autor desconocido. Documentación de azure | microsoft learn. <https://learn.microsoft.com/es-es/azure/?product=popular>.
- [14] Autor desconocido. Documentación de dialogflow | google cloud. <https://cloud.google.com/dialogflow/docs?hl=es-419>.
- [15] Autor desconocido. draw.io. <https://www.drawio.com/>.
- [16] Autor desconocido. Gestiona los proyectos de tu equipo desde cualquier lugar | trello. <https://trello.com/es>.
- [17] Autor desconocido. Get started with bootstrap · bootstrap v5.3. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
- [18] Autor desconocido. Model view presenter (mvp) powerpoint template - ppt slides. <https://www.sketchbubble.com/en/presentation-model-view-presenter.html>.
- [19] Autor desconocido. Models - hugging face. <https://huggingface.co/models>.
- [20] Autor desconocido. Postman api platform | sign up for free. <https://www.postman.com/>.
- [21] Autor desconocido. Siri - apple (es). <https://www.apple.com/es/siri/>.
- [22] Autor desconocido. What is scrum? | scrum.org. <https://www.scrum.org/learning-series/what-is-scrum>.

- [23] Autor desconocido. ¿qué es el web scraping? cómo extraer legalmente el contenido de la web. <https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/#:~:text=El%20web%20scraping%20se%20refiere,precios%20de%20varias%20tiendas%20online>.
- [24] Issam El Naqa and Martin J Murphy. *What is machine learning?* Springer, 2015.
- [25] Real Academia Española. Real academia española: Diccionario de la lengua española, 23.^a ed. <https://dle.rae.es/inteligencia#2DxmhCT>.
- [26] Shafquat Hussain, Omid Ameri Sianaki, and Nedat Ababneh. A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*, pages 946–956. Springer, 2019.
- [27] Sergio Francisco Iañez González. Desarrollo de un chatbot con dialogflow en el marco de las ciudades inteligentes. B.S. thesis, 2018.
- [28] Docker Inc. Docker overview | docker documentation. <https://docs.docker.com/get-started/overview/>.
- [29] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744, 2023.
- [30] Fred Lores. ¿qué es una plataforma low-code? <https://www.velneo.com/blog/que-es-plataforma-lowcode>.
- [31] Carlos Magán López. Interfaz conversacional en dialogflow para recomendación de películas. B.S. thesis, 2019.
- [32] David D McDonald. Natural language generation. *Handbook of natural language processing*, 2:121–144, 2010.
- [33] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.
- [34] Beatriz Soro Vegas et al. Diseño y desarrollo de un asistente interactivo transaccional para la reserva de hoteles. 2020.

- [35] Prissadang Suta, Xi Lan, Biting Wu, Pornchai Mongkolnam, and Jonathan H Chan. An overview of machine learning in chatbots. *International Journal of Mechanical Engineering and Robotics Research*, 9(4):502–510, 2020.
- [36] Anshul vyas. Model view presenter. <https://anshul-vyas380.medium.com/model-view-presenter-b7ece803203c>.
- [37] Lu Xu, Leslie Sanders, Kay Li, James CL Chow, et al. Chatbot for health care and oncology applications using artificial intelligence and machine learning: Systematic review. *JMIR cancer*, 7(4):e27850, 2021.