UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE FILOLOGÍA

FACULTAD DE INFORMÁTICA

MÁSTER UNIVERSITARIO EN LETRAS DIGITALES:
ESTUDIOS AVANZADOS EN TEXTUALIDADES ELECTRÓNICAS

TRABAJO DE FIN DE MÁSTER

CURSO ACADÉMICO: 2021-2022

Chatbot de predicción y recomendación de títulos universitarios con IA

AUTOR

Moreno Sánchez, Salvador

TUTORES

Vázquez-Poletti, José Luis

Pacios Izquierdo, David

Departamento de Arquitectura de Computadores y Automática

Convocatoria: septiembre 2022

Calificación: 9

Agradecimientos

Durante los últimos meses, tiempo que he dedicado para el desarrollo de este proyecto, se han visto implicadas muchas personas de las que no quisiera olvidarme, ya que este trabajo también es fruto de ellas.

En primer lugar, a toda la gente que se ha prestado a probar el asistente conversacional aquí desarrollado y que se han preocupado de difundirlo: Juanjo, Antonio, Iago, Laura, Fon, Andrea, Eder, Christelle, Iván, Inés y Lucía. También, a todas aquellas que me han aportado una argumentación crítica del funcionamiento del mismo en pos de su mejora: Antonio, Marta, Dani y Adriana.

En segundo lugar, a todos aquellos estudiantes de Bachillerato que han accedido a conversar con el *chatbot*: Dani, Pol, Sara, etc. Pero, en especial, a Alejandro, Adrián y Lucía, a los que tuve la suerte de dar clase como profesor y que, ahora, me han enseñado a mí dándome sus mejores críticas.

En tercer lugar, a mi tutor José Luis Vázquez-Poletti, de quien he escuchado sus consejos de joven con experiencia y de quien salió la idea de bautizar al asistente conversacional como Cloudia.

En cuarto lugar, y con mención especial, a David Pacios Izquierdo, persona con la que contacté después de que saliera encantado de su seminario de LaTeX y que, sin conocerme, aceptó inmediatamente la propuesta de iniciar este Trabajo de Fin de Máster y confió en mí desde el minuto uno. Espero que dicha confianza haya merecido la pena. También, agradecer su cercanía y su modo de ver la enseñanza universitaria. Transmite una pasión y un amor hacia la construcción del conocimiento que disparan las ganas de seguir trabajando. No me queda nada más que decir que mil gracias, David.

En quinto lugar, a Claudia Núñez, por aportarme ideas, críticas y por probar el proyecto una y otra vez durante el proceso de desarrollo.

No querría dejar de escribir esta memoria sin mencionar el lugar donde me he criado. A mi pueblo: Cúllar Vega.

Por último, acabar con el agradecimiento eterno a mis padres, Carmen y Salva, y a mi hermano, Ángel, por haberme educado, guiado y servido de ejemplo para ser lo que soy. Por todo su sacrificio, gracias.

Abstracto

Los altos índices de fracaso escolar en nuestro sistema educativo se repiten cada año y apuntan a la necesidad de replantear nuestra atención y metodología, lo cual puede suponer, en muchos casos, el requerimiento de más personal especializado en las instituciones educativas. De esta forma, existe un gran número de estudiantes que se encuentran desatendidos o confundidos por la escasez de información recibida. Sin embargo, resulta ser un problema que engloba a multitud de etapas educativas, cada una con su propio contexto generacional. Así es que en este estudio nos centramos en aquellos alumnos y alumnas de Bachillerato, quienes están ante una de las decisiones más importantes, la cual puede definir su futura vida laboral: la entrada a un grado universitario. De esta forma, el presente estudio busca desarrollar una herramienta que de forma automatizada dé respuesta a actividades de orientación académica y vocacional, por lo que se propone un *chatbot* de predicción y recomendación de grados universitarios sujeto a Inteligencia Artificial. Como resultado del proyecto, se expone el recorrido completo de una proposición de análisis, desarrollo y prueba real de un asistente conversacional implementado en Telegram y desplegado desde la nube, en el que para su construcción se afrontan tareas de análisis de datos, procesamiento del lenguaje natural y aprendizaje automático. Todo ello de forma nativa y siguiendo la filosofía de *software* libre. Al finalizar todo el proceso, se concluye que el *chatbot* puede resultar un útil instrumento de agitación y exploración de las inquietudes personales de los estudiantes, siendo un posible complemento de iniciación al proceso orientativo en las instituciones educativas, así como una herramienta que puede favorecer el análisis de datos sobre las pretensiones de los estudiantes con el fin de que los centros educativos y las universidades puedan adaptar su atención y oferta educativa.

Palabras clave: python, telegram, bot, chatbot, IA, orientación académica, orientación vocacional

Abstract

The high rates of school failure in our educational system are repeated every year and point to the need to rethink our attention and methodology, which may mean, in many cases, the need for more specialised staff in educational institutions. In this way, there is a large number of students who find themselves neglected or confused by the scarcity of information received. However, it is a problem that encompasses a multitude of educational stages, each with its own generational context. Thus, in this study, we focus on those students in the Baccalaureate, who are facing one of the most important decisions, which may define their future working life: the entrance to a university degree. In this way, this study seeks to develop a tool that automatically responds to academic and vocational guidance activities, so we propose a chatbot for the prediction and recommendation of university degrees subject to Artificial Intelligence. As a result of the project, a complete analysis, development and real test of a conversational assistant implemented in Telegram and deployed from the cloud is presented, in which data analysis, natural language processing and machine learning tasks are tackled for its construction. All of this is done natively and following the open source philosophy. At the end of the whole process, it is concluded that the chatbot can be a useful tool for agitation and exploration of students' personal concerns, being a possible complement to initiate the guidance process in educational institutions, as well as a tool that can favour the analysis of data on students' expectations so that educational centres and universities can adapt their attention and educational offer.

Keywords: python, telegram, bot, chatbot, AI, academic guidance, vocational guidance

Índice general

Capítulo 1

Introducción

Cuando recorremos el camino académico que nos propone el sistema educativo, nos acompañan una serie de elementos que van marcando y, a su vez, conformando parte de nuestra forma de ser. Tu compañero o compañera de batallas, las situaciones que te dejaron en evidencia o ese docente que recordarás de por vida. Sin embargo, hay factores que, como fantasmas, nos atormentan y nos persiguen a lo largo de nuestro viaje y que atacan nuestra mente. A lo largo de nuestro estudio nos centraremos en uno de los muchos que pueden rondar la cabeza de múltiples estudiantes, pero que puede que sea el más común de ellos: el fracaso escolar.

¿Cuándo se considera que el alumnado ha fracasado? En el *Diccionario de la lengua española* se define *fracaso* como «malogro, resultado adverso de una empresa o negocio» o como «suceso lastimoso, inopinado y funesto». Así, suspender un examen significa fracaso, repetir de curso significa fracaso y no trabajar en aquello que estudiaste también significa fracaso. Aunque pueda parecer desmesurado e incluso dañino enunciar esto a día de hoy; a nivel terminológico se ajusta a la realidad. Sin embargo, debemos ser conscientes de la necesidad de separar el correcto uso de la lengua con aquello que puede llegar a producir emocionalmente en una persona, «sobre todo por la frustración que supone para el individuo y por el desajuste de personalidad que lleva consigo, pudiendo ser un factor de depresiones infantiles y juveniles» (Ramos 1989).

Fracaso escolar se trata de un concepto de común empleo en Ciencias de la Educación, pero que ha cruzado el ámbito científico para asentarse dentro del uso social cotidiano, representando la raíz de esta problemática situación que exponemos. De esta forma, y volviendo a la pregunta que hemos planteado, la disociación del uso de *fracaso escolar* entre ámbito científico y ámbito social es crucial para encontrar la respuesta, siendo una necesidad el esfuerzo por cambiar la percepción y significado de su uso popular.

De esta dicotomía nace mi preocupación y atención por uno de los procesos de gran influencia en la búsqueda y el devenir del rendimiento académico del alumnado: la orientación educativa.

De tal manera, el objetivo principal de mi investigación es el de contribuir a disminuir los efectos emocionales que puedan tener los fantasmas del fracaso en el alumnado, dirigiendo el foco en los años de transición entre la educación secundaria y universitaria, momento caracterizado por la presión que supone la determinación de una calificación con el futuro

de cada discente y la decisión de elegir el supuesto camino correcto.

Así, se ha propuesto crear un asistente conversacional[1] que aporte una orientación académica y vocacional basada en la predicción, por un lado, del futuro éxito en las ramas de Ciencias y Humanidades; y, por otro, de grados universitarios que mejor encajen con el perfil del estudiante. Todo ello a través de tres modelos sustentados por Inteligencia Artificial alimentados por *corpus* de rendimiento académico de múltiples alumnos y alumnas y por intenciones relacionadas con cada una de las carreras universitarias ofertadas por la Universidad de Granada, dando lugar a la muestra de unos resultados que se atienen a los datos aportados por el usuario durante la interacción comunicativa con el asistente conversacional. Como canal de comunicación, se ha querido que sea de uso natural y común a la franja de edad a la que se dirige nuestro estudio, por lo que se ha decidido que esté integrado en la aplicación de mensajería Telegram. Además, el asistente conversacional generará un informe personalizado con los resultados del proceso y poseerá la opción de mostrar en el chat algunas recomendaciones para navegar y aprovechar de manera óptima los planes de estudio y las guías docentes de cada grado con el fin de promover el descubrimiento de nuevas vías de estudio y promover una búsqueda adecuada que informe de manera óptima.

La elaboración de nuestro producto ha seguido un proceso compuesto cronológicamente por la investigación de la fundamentación teórica en torno al lenguaje humano, los desarrollos en Inteligencia Artificial, a los asistentes conversacionales en el mundo educativo y a la orientación educativa y vocacional; por la elaboración de dos métodos de predicción en los que intervienen el análisis de datos, el procesamiento del lenguaje natural y el uso de algoritmos de aprendizaje profundo; por la implementación en Telegram; y por una fase de pruebas con público real y análisis de las mismas.

Por último, señalar que la intención de todo el trabajo que aquí se ha realizado sirva de inspiración o ayuda a otros estudiantes o investigadores en la tarea de mejorar y complementar la labor de orientación que se fomenta en las instituciones educativas. Así, el código de desarrollo será publicado en GitHub siguiendo la filosofía de *software* libre bajo la licencia de uso MIT.

[1]Enlace al proyecto completo albergado en GitHub: https://github.com/salvaMsanchez/CloudiaBot

Capítulo 2

Planteamiento del problema

El fracaso escolar en España es uno de los principales problemas que afectan a nuestro sistema educativo. «En el año 2020 en España el abandono temprano de la educación-formación alcanza la cifra de 20,2 % para los hombres y 11,6 % para las mujeres»[1], siendo el primer país de la UE con mayor tasa de fracaso en hombres y el cuarto en mujeres; todo esto en 2020. Aunque los números han descendido respecto a 2017, en el que el porcentaje era de 21,8 % para hombres y 14,5 % para mujeres, es evidente que esto representa un problema estructural que hay que paliar desde las primeras etapas de formación del alumnado y desde diferentes campos: pedagogía, psicología, etc.

No obstante, y ante la amplitud y los diferentes frentes abiertos que posee este problema a combatir, nos queremos centrar en un momento crucial de la vida de cualquier estudiante: la elección de su camino universitario después de completar sus estudios de Bachillerato. Por ello, debemos atender a los datos que se desprenden del abandono universitario existente en nuestro sistema.

En marzo de 2022, la profesora María Fernández Mellizo-Soto, en colaboración con el Ministerio de Universidades, elaboró un *Análisis del abandono de los estudiantes de Grado en las universidades presenciales en España*[2], en el que, como punto de partida se establece qué es el abandono universitario: «porcentaje de estudiantes de nuevo ingreso en un curso que tras haberse matriculado al menos en primer curso no lo hace durante dos cursos seguidos ni titula en cuatro cursos». A lo largo del estudio, el cual se basa en los estudiantes de la cohorte de ingreso de 2015-2016, se expone que «el 11 % de los estudiantes abandonó el Grado, el 6 % de los mismos tras el primer año de matrícula. Se comprueba, por tanto, que el principal riesgo de abandonar se da al inicio de los estudios». Además, se extraen gran cantidad de proposiciones en función de multitud de variables como que «la variable que más impacto tiene en la probabilidad de abandono de los estudios de Grado es el rendimiento del estudiante en el primer año» o que «cuanto

[1] Instituto Nacional de Estadística (2021, 07, 21). «Abandono temprano de la educación-formación» [Online]. Available: https://www.ine.es/ss/Satellite?L=es_ES&c=INESeccion_C&cid=1259925480602&p=%5C&pagename=ProductosYServicios%2FPYSLayout&param1=PYSDetalle&param3=1259924822888

[2] María Fernández-Mellizo (2022, 03). «Análisis del abandono de los estudiantes de Grado en las universidades presenciales en España» [Online]. Available: https://www.universidades.gob.es/stfls/universidades/ministerio/ficheros/Informe_Abandono_Universitario_completo_MFMS.pdf

mayor es la nota de admisión a la universidad del estudiante, menor su probabilidad de abandonar los estudios».

Esto no acaba aquí, ya que, por otro lado, y según los *Datos y cifras del Sistema Universitario Español*[3] publicado por el Ministerio de Universidades en 2020-2021, se expone que en el cohorte de nuevo ingreso de 2016-2017 el abandono del estudio en el primer año fue de 22,4 % en las universidades públicas y de 19,1 % en las privadas. Por otro lado, un 9,2 % del alumnado se cambió de estudios en el primer año en la pública y un 6,2 % en la privada. Respecto a las universidades no presenciales, un 45,2 % del alumnado abandona en el primer año y un 9,1 % se cambia de estudios.

Y es en este punto donde comienza nuestro estudio, con el reconocimiento de una verdad: el elevado porcentaje de abandono o cambio de estudios que se produce solamente en el primer año de carrera universitaria.

Por tanto, tenemos varios problemas a los que dar solución: ¿cómo intentar reducir estos porcentajes? ¿cómo orientar al alumnado de forma adecuada para que elija un camino determinado?

Para comenzar a dar forma a nuestro plan de acción, debemos escoger, en primer lugar, el público objetivo al que dirigir nuestro foco de estudio: estudiantes que estén cerca de elegir su futuro académico después de la educación secundaria, es decir, alumnado tanto de primer como de segundo año de Bachillerato.

La relevancia social de la cuestión no es baladí, así como su estudio científico. De esta forma, y como primera toma de contacto con dichos alumnos y alumnas, resulta relevante conocer su opinión. Para ello, acudiremos a Twitter, donde se han extraído 5000 tuits con la librería *Twint* de Python en relación a la búsqueda «elegir carrera»[4] y, tras un cribado de la información, hemos obtenido la siguiente nube de palabras (figura 2.1):

Figura 2.1: Nube de palabras al extraer tuits sobre la elección de carrera universitaria

Difícil, *futuro*, *decisión*, *miedo*, *trabajo* son los términos con más apariciones, lo cual evidencia que a nuestro objetivo le resulta un proceso decisivo para su futuro laboral y cuya responsabilidad y presión sobre escoger la mejor elección le produce inquietud. Esto

[3]Ministerio de Universidades (2021). «Datos y cifras del Sistema Universitario Español» [Online]. Available: https://www.universidades.gob.es/stfls/universidades/Estadisticas/ficheros/Datos_y_Cifras_2020-21.pdf

[4]Jupyter Notebook donde se ha desarrollado: https://github.com/salvaMsanchez/CloudiaBot/blob/master/wordcloud/extraccion_analisis_twint.ipynb

se refleja en tuits como: «el miedo que le tengo a no elegir la carrera correcta», «miedo de que no pueda elegir la carrera que me voy a estudiar, no se como hacerlo y me da miedo de que si me equivoco pueda fallar y no lograr nada», «tengo miedo de no ser lo suficientemente inteligente para la carrera que voy a elegir»; por lo que, como objetivo, debemos contribuir a la labor de evitar o reducir que aparezcan tuits como: «de verdad que la peor decisión que he tomado ha sido elegir esta carrera», «al final elegir esta carrera fue la peor decisión q tome en toda mi vida».

Capítulo 3

Objetivos del trabajo

Intentar reducir los porcentajes de fracaso que venimos exponiendo representa una tarea de una enorme envergadura, cuya labor será el resultado de multitud de esfuerzos a lo largo del tiempo y por distintas instituciones. Sin embargo, nuestro mayor objetivo es contribuir a ella, por lo que, para ello, nos hemos centrado en la orientación del alumnado mientras está cursando Bachillerato.

A su vez, la orientación es un proceso de largo recorrido y en el que se aplican distintos métodos de intervención y se tienen en cuenta diversos factores, a los que no se puede dar respuesta con una sola herramienta de desarrollo.

Por tanto, debemos empezar por marcar una meta realista constituida por pequeños objetivos que ayuden a generar un bien tanto en la orientación vocacional y académica del alumnado como en las instituciones educativas. Así, nuestra meta u objetivo general es la construcción de una herramienta que represente un complemento a utilizar en las labores de orientación, cuyos objetivos particulares son:

- **Predecir el éxito tanto en Matemáticas como en Lengua (representando la rama científica y la rama humanística, respectivamente):** existen muchas variables que entran en juego a la hora de determinar el éxito académico, por lo que podemos transmitir al estudiante una valoración que le haga reflexionar sobre sus posibilidades y un dato más al que poder atender.
- **Recomendar grados universitarios:** empujar al descubrimiento de otras carreras atendiendo a detalles de su personalidad, gustos, etc. que los alumnos y alumnas no han podido tener en cuenta representa una de las funciones de nuestra herramienta.
- **Informar al alumnado sobre la variedad de la oferta universitaria:** como veíamos en la figura 2.1, los términos *difícil* y *miedo* poseían una gran cantidad de apariciones. Una de las posibles causas puede ser la falta de información que posee el alumnado sobre los distintos grados universitarios, así como de sus planes de estudio y sus guías docentes, de los que puede extraer información muy valiosa para encaminar su decisión.
- **Recopilar información no sensible sobre las preferencias académicas del alumnado de cara a su futuro, ya sea dentro o fuera del ámbito univer-**

sitario: la relevancia de esto recae en la idea de poder ayudar a las instituciones educativas (institutos, universidades, etc.) a obtener datos que apliquen a la mejora de sus prácticas educativas, de la personalización de la enseñanza o de la oferta de sus programas.

- **Elaborar una aplicación que permita su uso en cualquier dispositivo y a la mano de cualquier persona**.

Por otro lado, debemos señalar un último objetivo que también se quiere alcanzar con la realización de este trabajo:

- **Compartir el proceso de realización del proyecto, así como la liberación del código de desarrollo:** el registro escrito del desarrollo y la liberación del código poseen la pretensión de abrir la puerta a otros estudiantes e investigadores a explorar nuevas formas de comunicación, aplicación tecnológica y evaluación tanto en el ámbito educativo como su extrapolación a otros campos. Además, todo esto permite la optimización y mejora de este proyecto con la ayuda de toda persona que se preste desde cualquier parte del mundo.

Capítulo 4

Fundamento teórico y estado de la cuestión

4.1. Inteligencia artificial

El ser humano siempre ha soñado, jugado y experimentado con darle vida a objetos carentes de ella. En el siglo I, Herón de Alejandría compone *Automata*, donde nos habla de diversos mecanismos que, gracias al uso de la gravedad, del agua, etc. son capaces de reproducir movimientos similares a los que podemos ver reflejados en la naturaleza: el más famoso es la eolípila, el cual hacia mover una esfera situada a cierta distancia sobre un caldero de agua caliente al que estaba conectada por dos tubos y que, gracias al vapor, giraba. En el siglo XVI, nos encontramos con uno de los ingenieros más ilustres de la corte de Carlos V, Juanelo Turriano, quien, además de ser conocido por su obra de llevar el agua del Tajo al Alcázar de Toledo, se erige como el posible autor de muchos autómatas de la época, siendo el «Hombre de Palo» el que más repercusión tuvo en los textos y del que se dice que poseía la función de pedir limosna y ser capaz de hacer una reverencia. En 1883, de la mano del florentino Carlo Collodi, que influenciado por algunas de las hipótesis y ensoñaciones de la alquimia por dar vida a lo inerte, nace el texto *Las aventuras de Pinocho*, obra magna de la literatura universal y que narra las pericias de una marioneta de madera que obtiene vida propia. Los ecos de la obra del escritor italiano llegaron a la mente del neoyorquino Stanley Kubrick, quien, basándose en el texto de ciencia ficción *Los superjuguetes no duran todo el verano* de Brian Aldiss, ideó y trabajó para llevar al cine *A.I. Inteligencia Artificial*; sin embargo, Stanley Kubrick vio imposible recrear los efectos futuristas que requería el guion y, al final, fue dirigida por Steven Spielberg en 2001, dos años después de la muerte de Kubrick.

Como vemos, jugar a ser una especie de dios no es cosa del presente, sino que ha sido una constante en la imaginación y recreación humanas. No obstante, no es hasta 1950, con *Maquinaria de computación e inteligencia* de Alan Turing, cuando se comienza a hablar de máquinas inteligentes. Desde este instante, el crecimiento fue exponencial, desde la introducción de las técnicas de aprendizaje profundo por John Hopefield y David Rumelhart en la década de los 80 para dar capacidad al ordenador a aprender a través de la experiencia, pasando por la derrota de Gary Kasparov por Deep Blue de IBM en

1997, hasta la victoria al póker de Libratus contra jugadores profesionales en 2017.

Esto nos lleva a preguntarnos: ¿qué es la Inteligencia Artificial? En palabras de uno de sus padres, el científico estadounidense Marvin Minsky (Minsky 1988), se trata del «estudio de cómo programar computadoras que tengan la facultad de hacer aquello que la mente humana puede realizar». Por otra parte, Lasse Rouhiainen (Rouhiainen 2018) nos la define como «la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano». Dichas definiciones coinciden en que el producto final debe actuar como lo haría un humano, acercándonos cada vez más a los sueños que siempre ha tenido nuestra especie a lo largo de la historia. Sin embargo, y a colación de este tema, debemos tener en cuenta aquello descrito como «valle inquietante» por el profesor experto en robótica Masahiro Mori, cuya hipótesis expone que cuando un sistema con inteligencia se asemeja en exceso a nuestra propia especie, causa rechazo. Todo esto nos lleva a las también interpretaciones erróneas que realizan los usuarios no especializados sobre el concepto de «Inteligencia Artificial», por lo que Sebastian Thrun (Thrun 2017), experto en IA, recomienda el uso de «ciencia de datos» con el fin de que resulte una expresión menos intimidatoria para que no infunda respeto o cree confusión en la sociedad.

4.2. El lenguaje humano y la lingüística

Aproximadamente en el año 360 a. C., el filósofo griego Platón escribe *Crátilo*, un diálogo sobre el significado, esencia y origen de las palabras que enfrenta verbalmente a Hermógenes, quien piensa que la relación entre significante y significado es arbitraria, y a Crátilo, que apuesta por una conexión propia y natural de cada una de las grafías y sonidos con el nombre del vocablo en cuestión. Todo ello marcado por las intervenciones de Sócrates, que va mostrando su parecer de las dos hipótesis. El final es ambiguo y no nos lanza ninguna conclusión contundente. Es por esto que la obra despertara tanto interés a lo largo de los siglos y que su análisis haya llegado hasta nuestros días en el campo de la lingüística. Sin embargo, queremos centrar la mirada en un breve fragmento de la obra:

> Sóc.: ¿Qué hacemos cuando nombramos con el nombre en calidad de instrumento (organon)? [...] ¿Acaso, en realidad, no nos enseñamos algo recíprocamente y distinguimos las cosas tal como son? [...] Entonces el nombre es un cierto instrumento para enseñar y distinguir la esencia, como la lanzadera lo es del tejido (Platón 1983).

Sócrates expone que las palabras son el instrumento que posee el ser humano para representar y diferenciar la multitud de elementos que componen nuestra realidad con el propósito de transmitírselo a los demás. Y es a partir del concepto de *organum*, instrumento o herramienta que nos permite comunicarnos con otra persona, desde donde Karl Bühler expone sus tres funciones del lenguaje en *Teoría del lenguaje*: función representativa, expresiva y apelativa (Bühler 1967).

Más tarde, Roman Jakobson añadió tres funciones más a las señaladas por el lingüista austriaco: fática, poética y metalingüística; las cuales pueden aparecer mezcladas o directamente no presenciarse en la comunicación. Además, señala que seis son las variables que intervienen en el acto comunicativo: emisor, receptor, mensaje, canal, referente y

código.

Por otro lado, el lingüista estadounidense Noam Chomsky (Chomsky 1989) nos habla del lenguaje como «una especie de estructura latente en la mente humana, que se desarrolla y fija por exposición a una experiencia lingüística específica», por lo que el lenguaje es asociado a un carácter cognitivo que es desarrollado por la necesidad del ser humano a relacionarse y a tener experiencias con los de su misma especie, las cuales marcarán su forma de expresarse verbalmente. Así, y según (Chomsky 1979), el lenguaje humano comienza su desarrollo en el momento en el que nacemos y gracias a los sonidos que recibimos, gracias a la experiencia, nuestro cerebro acoge un modo particular de lenguaje.

Teniendo en cuenta estos breves apuntes, podemos señalar que el lenguaje se compone se un conjunto de signos que sirven como instrumento para la comunicación entre un grupo de seres vivos, siendo el del humano el de mayor complejidad acorde con el desarrollo del órgano que se encarga de procesar y organizar la información: el cerebro.

4.3. Procesamiento del lenguaje natural

La Inteligencia Artificial y la lingüística se unen para conformar las bases que asientan el procesamiento del lenguaje natural, un campo interdisciplinario que comenzó a dar sus primeros pasos en la década de los sesenta con ELIZA, un simulador de comprensión automática del lenguaje que conversaba con el usuario como si de un psicoterapeuta se tratara, aunque este no realizara ningún análisis de la estructura lingüística porque se basaba en el reconocimiento de patrones en la propia entrada para dar una respuesta de un conjunto posible con el fin de parecer lo más natural posible.

En la década de los noventa, el análisis de los datos lingüísticos empezó a tomar relevancia, pero, sobre todo, la entrada de los métodos estadísticos en el procesamiento del lenguaje natural cambió el rumbo de la investigación e implementación de nuevos modelos conversacionales gracias a tres fenómenos (Liddy 2001):

- La disponibilidad de extensos córpora textuales que pueden ser procesados por el ordenador.
- Los avances en *hardware*, con ordenadores dotados de más memoria, mayor velocidad de procesamiento y mayor capacidad de almacenamiento.
- La llegada de internet, lo cual favorece no sólo la diseminación del conocimiento especializado sino también la accesibilidad de los recursos lingüísticos.

Actualmente, los corpus representan la fuente de datos más importante y el material de trabajo fundamental para el tratamiento estadístico. La presencia de la ingeniería es mucho mayor que la de la lingüística y eso lo demuestra el grueso de investigaciones que se realizan en torno al procesamiento del lenguaje natural, las cuales no se fundamentan en lingüística, sino en estadística y en la teoría de probabilidades.

¿A qué se debe que no se atienda lo que se debería al campo de la lingüística? (Wintner 2009) presentó los posibles tres factores que han propiciado esta situación, que aquí enunciamos brevemente: se suele argumentar que los sistemas fundamentados en teorías lingüísticas no satisfacen las necesidades del mundo real; el PLN es un campo de investi-

gación de naturaleza aplicada, por lo cual sus objetivos se orientan en definitiva hacia la construcción de aplicaciones informáticas, lo cual no da pie a investigaciones a largo plazo como ocurre en la lingüística teórica; las teorías lingüísticas se han vuelto tan «oscuras, barrocas y egocéntricas» que resultan muy poco atractivas para los informáticos. De esta forma, «se pretende conseguir la mayor efectividad posible incluso a expensas de una clara fundamentación lingüística teórica» (Llisterri & Moure 1996).

Ante esto, el procesamiento del lenguaje natural investiga el uso de computadoras para procesar o entender el lenguaje natural (LN) con el propósito de realizar tareas útiles (Deng & Liu 2018) y crear modelos computacionales del lenguaje con un elevado grado de detalle con el fin de conseguir programas informáticos que ejecuten una serie de órdenes que nos permitan realizar un adecuado tratamiento del lenguaje (Gelbukh 2010).

4.4. Chatbots

Los *chatbots* son servicios de *software* a los que se accede mediante una conversación en lenguaje natural (*Model-driven chatbot development*) y que, además, poseen la habilidad de interactuar con personas utilizando interfaces basadas en el lenguaje (Allison 2012).

Anteriormente, hemos nombrado al simulador de comprensión automático ELIZA, el cual evolucionó en un *chatbot* con personalidad llamado PARRY desarrollado en 1972 (Colby et al. 1971). En 1995 destacó ALICE, considerado el «ordenador más humano» (Wallace 2009) y que se basaba en un algoritmo de concordancia de patrones en Lenguaje de Marcado de Inteligencia Artificial (AIML) (Marietto et al. 2013). Dicho lenguaje marcó las investigaciones y los desarrollos de la primera década del siglo XXI. Actualmente, la dirección del desarrollo es diferente y se basa en algoritmos de tratamiento probabilístico del lenguaje que permiten una deducción de la respuesta en función a la entrada. Esto permite un desarrollo en lenguajes mucho más eficientes respecto a AIML y que nos permiten la constitución de productos como Microsoft Cortana, Apple Siri, Amazon Alexa o IBM Watson, además de su democratización entre la ciudadanía de un producto de alto desarrollo tecnológico.

Según (Carayannopoulos 2018), el éxito que han tenido los *bots* conversacionales en los últimos años se debe a dos razones principales: el uso extendido de programas de mensajería instantánea y el modelo basado en aplicaciones.

Los *chatbots* deben tener los siguientes componentes fundamentales para que pueda haber una conversación[1]:

- **Inteligencia artificial conversacional**: la fuente básica de los *chatbots*, gracias a la cual se produce toda la gestión y el procesamiento del lenguaje natural. Los primeros *chatbots* se centraban en la interpretación y el reconocimiento de patrones y normas. Los más avanzados implementan procesos de aprendizaje profundo para analizar la entrada dada por el humano, aprender de las conversaciones y generar una respuesta lo más adecuada posible.

[1]Blanca Nieves (2018, 05, 03). «IA Conversacional: definición y conceptos básicos» [Online]. Available: https://planetachatbot.com/ia-conversacional-conceptos-basicos-y-definicion/

- **Experiencia de usuario (UX)**: permite establecer una conversación natural, inteligente y coherente.
- **Interfaz de usuario (IU)**: mediante la cual el usuario puede ver o escuchar las conversaciones con el *chatbot*.
- **Diseño conversacional**: permite dotar de lógica humana una interacción artificial.

El proceso de interacción entre la persona y el *chatbot* se puede producir de diferentes maneras según la interfaz comunicativa. Podemos distinguir tres grandes tipos de *chatbots*[2]:

- **Basados en cajas de texto (*chatterboxes*):** la interacción se produce mediante entradas y salidas de texto o de voz. Con el procesamiento de lenguaje natural se puede convertir el texto escrito en texto oral y viceversa, lo que abre las posibilidades comunicativas de la interacción entre persona y *chatbots*[3].
- **Asistentes virtuales personificados (*embodied conversational agents*):** la interfaz se representa con la figura de un cuerpo, o de una cara en forma de avatar, que interactúa con el usuario y que puede contener audio, texto y otros recursos de representación audiovisual y multimedia (Allison 2012).
- **Físicos:** un tercer tipo son los que se presentan como un robot físico, que puede tener forma humanoide o no.

4.4.1. Chatbots en educación

Un *chatbot* educativo, según su tipología, puede poseer o no la intención de involucrarse en el proceso de enseñanza-aprendizaje del estudiante. Por un lado, tendríamos aquellos que se encargan de ejercer tutorización a lo largo del proceso de aprendizaje del alumnado (motivar, hacer reflexionar al usuario, ajustar su ritmo, etc.) y aquellos que se basan en presentar un problema o ejercicio según los contenidos para que el alumnado responda y sea evaluado. Por otro lado, y atendiendo a la no posesión de intencionalidad educativa, encontraríamos los que tratan funciones de administración y orientación del estudiante. La interacción con el *chatbot* en un ámbito educativo se hace muy interesante en el momento en el que le facilitas una determinada información que pueda resultar de difícil acceso para el estudiante en un intervalo de tiempo muy breve, con una disponibilidad permanente y ejecutable en cualquier situación y contexto, por lo que es idóneo para su implementación en los cada vez más extendidos aprendizajes ubicuos.

Para profundizar en ello, veamos una serie de ejemplos asociados a diversas tareas educativas (Garcia Brustenga et al. 2018):

- **Administrativas y de gestión para favorecer la productividad personal:** el *chatbot* diseñado en la Cornell University de Estados Unidos, CourseQ, se encarga de transmitir información tanto al profesorado como al alumnado en relación a

[2]Daniel Cerdas Méndez (2017, 07, 01). «Historia de la Inteligencia artificial relacionada con los Chatbots» [Online]. Available: https://planetachatbot.com/historia-inteligencia-artificial-relacionada-con-chatbots/

[3]Donald Clark (2018, 04). «The fallacy of "robot" teachers» [Online]. Available: http://donaldclarkplanb.blogspot.com/2018/04/the-fallacy-of-robot-teachers.html

fechas de entrega, eventos, horarios, etc. Sus respuestas son extraídas directamente de textos introducidos por el profesorado.

- **Resolución de preguntas frecuentes:** Genie es un *chatbot* construido por la Deakin University en Australia que emplea la tecnología Watson IBM para resolver dudas que poseen los estudiantes en el campus universitario.
- **Acompañamiento al estudiante:** elaborado en el Georgia Institute of Technology, Jill Watson es un *chatbot* que ayuda al alumnado con sus tareas y responde preguntas particulares sobre diversas asignaturas.
- **Motivación:** Differ es un *chatbot* que se usa en la BI Norwegian Business School con el propósito de crear un espacio donde los estudiantes se sientan libres a la hora de preguntar sin la presión ni el juicio de nadie a través del agrupamiento de alumnos y alumnas en la misma situación académica, personal, etc. Está enfocado en esa parte del alumnado que necesita una atención más personalizada para conseguir hacerlo partícipe del proceso de aprendizaje.
- **Práctica de habilidades, destrezas específicas y simulaciones:** la aplicación Duolingo, posee un *chatbot* con el que puedes mejorar tu competencia conversacional en otro idioma en torno a distintas situaciones dialogales.
- **Estrategias de reflexión y metacognitivas:** Replika es una Inteligencia Artificial que, a través de la pantalla, te ofrece un avatar virtual personalizable con el que poder combatir la soledad y cuya intención es la de cubrir las necesidades emocionales del usuario a través de una conversación en la que podemos ver a nuestro avatar expresarse tanto desde lo gestual, lo oral y lo textual preocupándose por tu estado emocional y transmitiendo su preocupación por ti.
- **Evaluación del aprendizaje de los estudiantes:** *The Guardian of History (Introducing a challenging teachable agent)* es un programa que ejecuta un juego en el que, a través de una máquina del tiempo, debes pasar por diferentes escenarios de la historia de la humanidad para aprender y, más tarde, enseñárselo a Timy, un joven elfo que tiene que prepararse para heredar el puesto de guardián del Castillo del Tiempo. De esta forma, el estudiante aprende y es evaluado con las actividades que se proponen para transmitirle el conocimiento aprendido a Timy. A pesar de no ser un sistema que incorpora Inteligencia Artificial, es importante que resaltemos la posibilidad de conversación guiada que ofrece un juego para promover el aprendizaje.

Por otro lado, podemos destacar otros como[4]:

- **LMS invisible:** Otto, de LearningPool, se trata de un *chatbot* que cumple muchas de las funciones que realiza un sistema de gestión del aprendizaje. Como usuario, y a través de preguntas, puedes obtener información sobre los ejercicios, el aprendizaje dado en clase, el contacto de los profesores, enlaces a recursos complementarios, etc. De esta forma, el contenido de un LMS, el cual es fijo, se convierte en un recurso dinámico y que te proporciona aquello que necesitas en ese momento, representando una herramienta complemetaria a las plataformas de aprendizaje de gran utilidad.

[4]Donald Clark (2017, 12, 17). «10 uses for Chatbots in learning (with examples)» [Online]. Available: https://donaldclarkplanb.blogspot.com/2017/12/10-uses-for-chatbots-in-learning-with.html

- **Bot mentor:** se encarga de la orientación del alumnado para que sean capaces de resolver sus problemas por sí mismos. Un ejemplo sería AutoMentor de Roger Schank, el cual, además de responder preguntas frecuentes, aporta consejos para fomentar el pensamiento crítico de los alumnos y alumnas. La intención del autor es que esta herramienta evolucione a una que sea capaz de analizar la ausencia de pensamiento crítico del alumnado para darle una retroalimentación adecuada para mejorar su habilidad crítica. La investigación de este tipo de *bots* supone una línea de desarrollo que posee un largo recorrido.

Llegados a este punto es importante resaltar que los asistentes conversacionales en el campo de la Educación están lejos, muy lejos, de suplir la labor que realiza el personal educativo, por lo que podemos olvidarnos de una completa automatización de la enseñanza. Sin embargo, representan una opción cada vez más interesante para potenciar los procesos de enseñanza-aprendizaje. La creación de *bots* encargados de una tarea específica, tal y como hemos visto en los ejemplos anteriores, son los que están dando resultados. Aun así, conseguir una buena eficacia es una labor complicada y cuya construcción requiere de mucho tiempo, procesos de mejora y conocimientos técnicos.

4.5. Orientación vocacional

Para iniciar el camino de la conformación del pensamiento acerca de la orientación vocacional es de obligación citar una de las obras de gran éxito editorial a principios del siglo XVI en España, *Examen de ingenios para las ciencias* del médico Huarte de San Juan, quien en el Proemio se dirige al rey Felipe II:

> Que el carpintero no hiciese obra tocante al oficio de labrador, ni el tejedor del arquitecto, ni el jurisperito curase, ni el médico abogase, sino que cada uno ejercitase sólo aquel arte para la cual tenía talento natural, y dejase las demás. [...] Y porque no errase en elegir la que a su natural estaba mejor, había de haber Diputados en la República, hombres de gran prudencia y saber, que en la tierna edad descubriesen a cada uno su ingenio (de San Juan 1846).

Dicha preocupación no sólo se producía en España, sino que desde Francia los vientos soplaban de forma similar y la sensación se reflejaba en las palabras de Montaigne en sus *Ensayos* por la dificultad «de forzar las tendencias o propensiones naturales. De donde resulta que por no haber elegido bien, trabajase sin fruto, empleando un tiempo inútil en destinar a los niños precisamente para aquello que no han de servir» (De Montaigne 1997).

Etimológicamente, el vocablo *vocación* procede del latín *vocatio, vocationis* que significa «llamado», «invitación», por lo que observamos que se asocia a aquello que nace y proviene de una llamada exterior y que nos invita a dedicar nuestro tiempo a esa tarea laboral.

Sin embargo, no es de extrañar que la orientación vocacional comenzara a tenerse en cuenta en relación a las necesidades e intereses socioeconómicos de una sociedad industrializada, por lo que a principios del siglo XX nació en Munich la primera oficina que se encargaba de orientar profesionalmente con el objetivo de catalogar a las personas como

aptos o no para desempeñar una determinada labor con el propósito de evitar riesgos y pérdidas a la fábrica con trabajadores que no servían (García Villegas et al. 1965).

No es hasta 1951, con el nacimiento de la Asociación Internacional de Orientación Profesional, cuando se comienza a tener en cuenta el plano individual de cada uno, aunque siempre con la mirada puesta en los factores económicos de la industria. Dicha asociación se encargó de convocar congresos, seminarios, editar un boletín donde mostraba el estado de la orientación profesional en cada país, etc. Con ella, la consciencia sobre la orientación vocacional es mayor y van naciendo metodologías distintas en cada país para su tratamiento, lo cual, por lo general, está bajo control del Ministerio de Trabajo y, en algunos otros países, se colabora con el Ministerio de Educación. Es en este momento cuando en EEUU se empieza a ofrecer *Counselling*, en el que un psicólogo se encarga de evaluar al alumnado a través de exámenes de intereses, aptitudes y motivación (García Villegas et al. 1965).

Actualmente, y generalizando, tal vez debido a la sobrecarga de trabajo que tienen los orientadores, las escasas actividades de orientación vocacional desarrolladas en los colegios tienen esta perspectiva: se pretende que el asunto quede resuelto mediante un test y bajo la pregunta: ¿Qué quieres ser de mayor? (Grañeras & Parras 2009)

Rascovan (Rascovan 2014) elaboró una clasificación de las distintas intervenciones en orientación vocacional que se pueden aplicar:

- **Intervención psicológica:** se enfoca en el comportamiento de la persona con su entorno social, en el propio deseo o en la ausencia del mismo, etc. En general, se ocupa de estudiar la dimensión individual. Para ello, se aplican una serie de evaluaciones para obtener resultados relacionados con los intereses, aficiones, etc. con el objetivo de arrojar cierta luz a la identificación de su vocación.
- **Intervención pedagógica:** se centra en dar a conocer las salidas profesionales y grados universitarios existentes, además de una contextualización de las posibles elecciones en relación a las condiciones sociales y económicas de entorno del individuo.
- **Intervención Sociológica:** se refiere a la llamada «orientación laboral» y se aplica a aquellas personas que no han conseguido desarrollar las competencias normativas para superar las posibilidades académicas que oferta el sistema educativo actual.

Leyendo la visión de diversos expertos de este campo de investigación (véase (Mendoza 2000), (Donoso & Figuera 2007), (Grañeras & Parras 2009) u (Ojea Rúa 2014)), se observa que hay un asentamiento de la enumeración de los factores que afectan al proceso de elección de un grado universitario:

- **Individuales:** referidos al propio conocimiento de sí mismo, tanto personal como competencial; la motivación ante el estudio actual y futuro; el autoconcepto; la autoestima; etc.
- **Sociales:** aquellos que dirigen la mirada a las facilidades sociales aportadas por parte de la administración y de la propia comunidad donde reside el estudiante, y, si estas no suceden, a aquellos factores que fomenten la exclusión social del discente.
- **Estructurales:** los relacionados con las posibles salidas laborales de los estudios a elegir y la probabilidad de entrar al mercado laboral en la zona donde reside o si se

debe optar por la emigración a otra provincia, comunidad o país.

Es interesante tener muy en cuenta lo que propone (Rascovan 2014) en relación a la pregunta que según (Grañeras & Parras 2009) se sucede continuamente en los procesos de orientación vocacional en los centros educativos: ¿qué quieres ser de mayor? Su propuesta se basa en cambiarla por: ¿qué tengo que elegir ahora? o ¿qué es lo mejor para mí ahora? De esta forma, el foco apuntará a los factores tanto individuales como estructurales, obligando al alumnado a entrar en una reflexión de mayor complejidad y madurez consigo mismo y con su entorno.

Por último, y para una aplicación adecuada de la orientación vocacional previa al ingreso en la vida universitaria, debemos tener en cuenta aquellos factores que pueden explicar el abandono universitario, los cuales pueden clasificarse, siguiendo a la profesora María Fernández-Mellizo[5], en:

- **Factores individuales**: se pueden dividir en factores demográficos, socioeconómicos y académicos (según la clasificación de (Garcia 2014)).
  - **Demográficos:** los relacionados con la edad, el género, etc.
  - **Socioeconómicos**: los relacionados con la situación social, económica y cultural del alumno o alumna.
  - **Académicos**: los que tienen que ver con la experiencia educativa previa y con las expectativas académicas.
- **Factores de interacción del estudiante con el ambiente de la universidad**, los cuales pueden ser de dos tipos:
  - **Integración académica:** se trata del rendimiento académico que posee el alumnado en sus primeros pasos en la universidad.
  - **Integración social:** relacionado con el nivel de participación y de compromiso con las actividades propuestas en la universidad.
- **Factores institucionales:** toman en consideración el tipo de universidad (pública o privada), sus recursos, calidad de la enseñanza y sus estudios ofertados, etc.

4.6. Perspectiva teórica

Las Humanidades Digitales se definen como «una serie de principios como la interdisciplinariedad y la construcción de modelos, valores como el acceso libre y el código abierto, y prácticas como la minería de datos y la colaboración» (Rojas Castro 2013), representando «la capacidad de manipular la información textual, cuantitativa, gráfica, geográfica, audiovisual o multimedia para la investigación humanística»(Gayol & Melo Flórez 2017).

Por tanto, podemos enunciar que la perspectiva teórica en la que se inserta este trabajo se alimenta de los conceptos, métodos y prácticas de las Humanidades Digitales, cuya

[5]María Fernández-Mellizo (2022, 03). «Análisis del abandono de los estudiantes de Grado en las universidades presenciales en España» [Online]. Available: https://www.universidades.gob.es/stfls/universidades/ministerio/ficheros/Informe_Abandono_Universitario_completo_MFMS.pdf

justificación viene dada gracias a la confluencia que hay entre las labores en lingüística e informática, que desembocan en el campo del procesamiento del lenguaje natural (PLN); a la apuesta por herramientas y entornos de desarrollo de filosofía de *software* libre y la liberación del código informático, la cual fomenta el libre acceso y expone un modelo sobre el que los futuros estudiantes e investigadores puedan consultar, partir y trabajar; y al ético tratamiento de los datos y su debida exposición para contribuir a la investigación humanística y, además, al bien social. Por último, se deben sumar los conocimientos relacionados con la situación pasada y actual del contexto educativo, la orientación académica y la psicología para aunarlos a un proyecto caracterizado por su interdisciplinariedad.

De esta forma, se pretende crear un «diálogo horizontal entre la informática y las ciencias humanas que permita construir nuevas interpretaciones que no podrían ser elaboradas de otra manera»(Gayol & Melo Flórez 2017).

Capítulo 5

Metodología

5.1. Entorno de desarrollo

Para el desarrollo de nuestro *chatbot* se ha requerido de:

- **Anaconda:** se trata de una distribución de los lenguajes Python y R que se caracteriza por ser de código abierto y por su extenso uso en el campo de la ciencia de datos. Su instalación incluye alrededor de 250 paquetes, el acceso a más de 7.500 paquetes de código abierto a través de PyPi e, incluso, Anaconda Navigator, una interfaz como alternativa a la línea de comandos.
    - **Jupyter Notebook:** aplicación de *software* alojada en un servidor web que nos permite crear documentos que admiten código, texto Markdown, gráficos, etc. Pertenece al *Project Jupyter*, una comunidad cuyo objetivo es «desarrollar software de código abierto, estándares abiertos y servicios para la informática interactiva en docenas de lenguajes de programación»[1]. En ella, los datos pueden ser cargados, transformados y modificados a lo largo de nuestra hoja de trabajo, además de poder testar nuestro código. Es la herramienta que usaremos para desarrollar los tres modelos predictivos que se exponen en la memoria.
- **Python:** creado por el informático holandés Guido van Rossum, se trata de un lenguaje de alto nivel de programación interpretado y que, a pesar de la versatilidad respecto a su empleo debido a su carácter multiparadigma, es muy utilizado en el campo de la ciencia de datos y, concretamente, en el aprendizaje automático. Su nacimiento a principios de los años noventa está relacionado con el código abierto, ya que se trató de un proyecto de *software* libre. Además, su licencia, la *Python Software Foundation License*, es de código abierto. Todo esto contribuye a que posea una gran comunidad detrás que realiza grandes contribuciones al avance del lenguaje y al desarrollo de librerías destinadas a la realización de multitud de tareas. Su comunidad en torno al *Machine Learning* es extensa y, según una encuesta de

[1] Project Jupyter (2022) [Online]. Available: https://jupyter.org/

2021 realizada por Stack Overflow[2], el 27 % de todos los desarrolladores Python usan dicho lenguaje para la ciencia de datos.

- **Librerías Python:**
  - ***TensorFlow*:** se trata de una librería de código abierto para el aprendizaje automático con una numerosa comunidad, lo cual hace que posea un sitio web[3] donde aprender, foros y blogs de ayuda, etc. Nos permite la creación y entrenamiento de modelos sin que veamos afectada nuestra velocidad o rendimiento. Actualmente, podemos encontrar casos de éxito en empresas como Airbnb, PayPal o CocaCola, que han empleado *TensorFlow* para mejorar la experiencia de sus huéspedes, detectar fraudes y reconocer comprobantes de compra, respectivamente.
  - ***Keras*:** consiste en una API de código abierto programada en Python con el objetivo principal de elaborar redes neuronales y cuyo rendimiento se ve potenciado al combinarse con *TensorFlow*. *Keras* nos permite diseñar nuestra propia red ajustando la cantidad de capas a emplear, escogiendo diversas funciones, etc., también nos brinda la posibilidad de utilizar Redes Neuronales ya preentrenadas. Está diseñada para que su empleo resulte de fácil manejo para el usuario y, tal y como declaran en su web, «*Keras is an API designed for human beings, not machines*»[4].
  - ***NumPy*:** biblioteca de Python utilizada para trabajar con matrices y álgebra lineal, sobre todo. Fue creado por Travis Oliphant en 2005 y se trata de un proyecto de código abierto, cuyo rendimiento es más veloz que las listas tradicionales de Python.
  - ***Sklearn*:** es una de las librerías más importantes para la ciencia de datos. Posee funciones para vectorizar o separar los datos de entrada en *train* (datos de entrenamiento) y *test* (datos de prueba), así como funciones para obtener métricas que indican la precisión de los modelos.
  - ***Pandas*:** librería de Python de código abierto dedicada a tareas de análisis, limpieza, exploración y manipulación de grandes cantidades de datos, por lo que nos permite sacar conclusiones estadísticas. Está construida sobre las bases de la librería NumPy.
  - ***NLTK*:** se trata de una de las herramientas más utilizadas para el procesamiento del lenguaje natural. Ofrece más de cincuenta corpus idiomáticos, así como funciones para clasificar, tokenizar, derivar regresivamente, analizar sitácticamente, etc. texto.
  - ***BeautifulSoup*:** su empleo se basa en el análisis de documentos HTML, por lo que su uso más extendido es dedicado a la extracción de la información albergada en la web, proceso comúnmente conocido como *web scraping*.

[2]Stack Overflow (2021). Developer Survey [Online]. Available: https://insights.stackoverflow.com/survey/2021#overview

[3]TensorFlow (2022) [Online]. Available: https://www.tensorflow.org/?hl=es-419

[4]Keras (2022) [Online]. Available: https://keras.io/

- ***Matplotlib* y *Seaborn*:** usadas para la visualización de datos. Nos permiten generar gráficos de todo tipo, los cuales pueden ser exportados en distintos formatos.
- ***Pickle*:** esta librería permite guardar y cargar el contenido de una variable de Python en un fichero. Es útil para entrenar un modelo de *Machine Learning* y guardarlo en un fichero para después cargarlo nuevamente cuando, en nuestro caso concreto, el *bot* reciba un mensaje en lugar de crear el mismo modelo desde cero cada vez.
- ***Os*:** para acceder a la información proporcionada por el entorno de trabajo que empleemos (sistema operativo) cuya principal función es la de manipular directorios para poder leer y escribir archivos.

- **LaTeX:** desarrollado por Leslie Lamport a principios de los ochenta, quien a través de comandos TEXconsiguió elaborar un sistema de composición de textos de gran calidad a nivel tipográfico. Su uso no es WYSIWYG, ya que se basa en una serie de instrucciones que se encargarán de dar formato, permitiendo al usuario centrarse únicamente en la realización del contenido. Su filosofía de *software* libre ha dado pie a una gran comunidad que aporta nuevas funcionalidades y, sobre todo, plantillas para todo tipo de usos.
  - ***pylatex*:** para introducir LaTeXen nuestro proyecto, emplearemos la librería Python llamada *pylatex*. La combinación entre estos dos lenguajes es ideal para las pretensiones de este proyecto, ya que podremos crear un informe sin salirnos del lenguaje nativo de uso y, al mismo tiempo, utilizando la alta calidad y precisión de LaTeX.
- **Telegram:** aplicación gratuita de mensajería instantánea desarrollada por los hermanos Pavel y Nikolái Durov en 2013, cuya parte cliente es libre y de código abierto. Entre sus ventajas destacan su capacidad de consumir menos recursos informáticos (RAM/GPU) y batería; transmitir mayor seguridad y privacidad; la posibilidad de integrar bots de todo tipo (conversión de texto a voz, ejecutar tareas, moderación de grupos, etc.); su versatilidad y disponibilidad para todas las plataformas. Por otra parte, Telegram representa un medio al que están totalmente habituados los jóvenes y cuya interacción se corresponde con los códigos lingüísticos que manejan en su día a día y que, por ende, les resultará de mayor comodidad e informalidad ante un proceso que carácter académico.
  - ***pyTelegramBotAPI:*** librería que se usará para conectar nuestro código con nuestro *bot*. Se trata de una de las librerías más empleadas y con más documentación en relación al lenguaje Python.
- **Google Colab:** se trata de un servicio en la nube que nos aporta acceso gratuito a GPU, lo cual se erige como una muy buena opción para toda persona que no tenga los recursos requeridos a nivel de *hardware*. Por otro lado, es un entorno donde no hay que realizar complejas configuraciones. Además, ya viene con las librerías y los paquetes de mayor uso instalados. El almacenamiento está conectado con Google Drive, lo que nos permite trabajar en cualquier dispositivo asociado a nuestra cuenta. También, admite la conexión con GitHub y maquetar nuestro cuaderno con el lenguaje de marcado ligero Markdown, cosa de gran utilidad para

ofrecer un cuaderno intuitivo y más fácil de comprender para cualquier usuario. Es la herramienta que se emplea para desplegar nuestro asistente conversacional y hacerlo accesible a cualquiera.

Como se observa, con todas estas herramientas que se emplearán, se persigue cumplir con la filosofía que posee este proyecto: la de *software* libre. Se espera conseguir un proyecto basado en la construcción del conocimiento a través de la libertad de código, así como poder ayudar a próximos estudiantes, investigadores o cualquier persona que se preste con la descripción de todo el proceso de creación de este proyecto a lo largo de la presente memoria. Para que todos y todas podamos tener acceso al conocimiento.

5.2. Desarrollo de los modelos de predicción

5.2.1. Predicción de la probabilidad de éxito en Matemáticas y Lengua

Se ha elaborado un modelo predictivo basado en un conjunto de datos encontrado en Kaggle[5], cuyo título es *Student Performance Data Set*[6]. Se trata de un *dataset* basado en el rendimiento de estudiantes de educación secundaria de dos escuelas portuguesas. Recoge variables (32 en total) que pueden llegar a afectar al rendimiento académico en las asignaturas de Matemáticas y Lengua, las cuales van desde características demográficas (sexo, residencia rural o urbana, nivel académico de los padres, etc.), pasando por atributos sociales (actividades extraescolares, relaciones amorosas, etc.) hasta las calificaciones de los estudiantes en el primer y segundo trimestre de cada asignatura.

Además, sobre este conjunto de datos poseemos un interesante estudio realizado por los propios autores (Cortez & Silva 2008), donde los modelos predictivos se sometieron a prueba bajo métodos de minería de datos como Árboles de Decisión (DT), Bosques Aleatorios (RF), Redes Neuronales (NN) y Máquinas de Vectores de Apoyo (SVM) codificados con el lenguaje de programación R y cuyas conclusiones nos servirán de ayuda en la construcción de nuestro propio modelo como, por ejemplo, que debemos realizar las predicciones atendiendo, sobre todo, a los atributos «G1» y «G2» (identificadores dados a las calificaciones del primer y segundo trimestre, respectivamente), ya que los resultados no poseían tanta claridad si se obviaba alguno de estos; o que el entrenamiento de la Red Neuronal conseguía buenos resultados con la parametrización de las épocas (*epochs*) ajustada a 100.

Dicho esto, en nuestro proyecto exclusivamente nos centraremos en construir un modelo con Redes Neuronales en Python, basándonos, en gran parte, en un algoritmo expuesto

[5]Kaggle se trata de la plataforma web que acoge a la comunidad de analistas de datos más importante. En su sitio (https://www.kaggle.com/) ya nos invitan a usar sus más de 50.000 conjuntos de datos públicos y sus 400.000 cuadernos de programación públicos. Además, podemos programar directamente en la web, aprender diversos lenguajes y librerías con cursos que ofrecen, participar en competiciones, etc.

[6]P. Cortez and A. Silva (2014, 11, 27). «Student Performance Data Set» [Online]. Available: https://archive.ics.uci.edu/ml/datasets/student+performance

en un artículo de la famosa web de enseñanza FreeCodeCamp[7], en el que se propone una predicción de si el precio de una casa está por debajo o por encima de la media atendiendo a una serie de parámetros. Se ha escogido este artículo para construir nuestra propia Red Neuronal por tres motivos: el tratamiento pedagógico y didáctico que posee; el uso de redes neuronales, las cuales nos van a permitir crear modelos de predicción basados en un entrenamiento inicial a partir de nuestros específicos datos; y la similitud de los resultados del artículo con aquello que queremos obtener con nuestro modelo: predicciones que vayan del 0 al 1 y que luego podamos extrapolar a porcentajes de éxito.

La elaboración de nuestros dos propios modelos predictivos se dividió en seis pasos: visionado del estado de los datos, limpieza de los datos, procesamiento de los datos, construcción de la Red Neuronal, visualización de la pérdida y precisión del modelo y regularización del modelo.

Visionado del estado de los datos

Importamos las librerías necesarias, en este caso acudiremos a *Pandas*, *Seaborn* y *Matplotlib* para la manipulación y visualización del conjunto de datos. Almacenamos nuestros datos, que están en formato *CSV*, en un *dataframe* de *Pandas*, el cual nos refleja la gran cantidad de información que poseemos y que vamos a tener que cribar y limpiar.

No obstante, antes de continuar es sugerente visualizar gráficamente la última variable del *dataframe* («G3»: la calificación del tercer trimestre), cuyo valor es el que queremos predecir al final del proceso, cosa relevante a la hora de conocer cómo puede actuar nuestra predicción final.

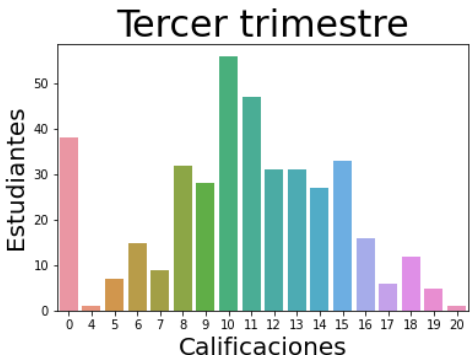


Figura 5.1: Matemáticas

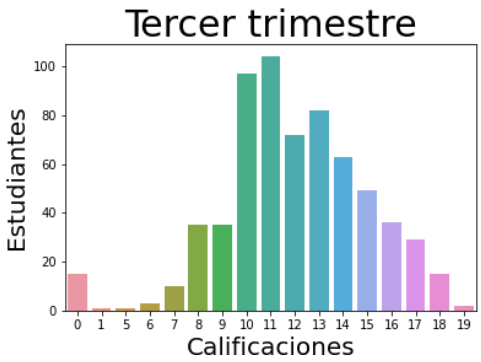


Figura 5.2: Lengua

Antes de nada, debemos tener en cuenta que las calificaciones van desde el 0 hasta el 20, siendo el 10 el límite entre el aprobado y el suspenso. Por tanto, podemos observar que existen variaciones entre los dos conjuntos de datos que vamos a tratar. Se observa en la figura 5.1 que hay un número elevado de suspensos en Matemáticas, mientras que en la figura 5.2 vemos que en Lengua hay un alto porcentaje de aprobados. Estos datos ya nos dan indicios de cómo podrán ser los resultados de nuestra predicción final, donde la probabilidad de éxito en Matemáticas tenderá a ser menor que la de Lengua.

[7]Joseph Lee Wei En (2019, 04, 04). «How to build your first Neural Network to predict house prices with Keras» [Online]. Available: https://www.freecodecamp.org/news/how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/

Limpieza de los datos

Una vez que nos hemos asegurado que no hay ningún valor nulo en nuestro conjunto de datos, es el momento de decidir qué variables consideramos que no van a afectar en la futura predicción de la calificación de los estudiantes. De los 32 atributos que componen nuestro conjunto de datos, se han escogido 12 (5.1): «sex» (sexo), «address» (tipo de domicilio del estudiante), «famsize» (tamaño de la familia), «Medu» (educación de la madre), «Fedu» (educación del padre), «studytime» (tiempo de estudio), «famsup» (apoyo académico en casa), «paid» (apoyo académico fuera de casa), «activities» (actividades extraescolares), «internet» (acceso a internet en casa), «G1» (calificación primer trimestre), «G2» (calificación segundo trimestre), «G3» (calificación tercer trimestre).

	sex	address	famsize	Medu	Fedu	studytime	famsup	paid	activities	internet	G1	G2	G3
0	F	U	GT3	4	4	2	no	no	no	no	5	6	6
1	F	U	GT3	1	1	2	yes	no	no	yes	5	5	6
2	F	U	LE3	1	1	2	no	yes	no	yes	7	8	10
3	F	U	GT3	4	2	3	yes	yes	yes	yes	15	14	15
4	F	U	GT3	3	3	2	yes	yes	no	no	6	10	10

Cuadro 5.1: Primeras 5 columnas del *dataframe* de la asignatura de Matemáticas

La elección de dichas variables se basa en su relación con los factores individuales y sociales que señalábamos en la sección 4.5 (Pág. 14) de la fundamentación teórica como el tipo de domicilio, el tamaño de la familia, la educación de los padres y el acceso a internet; en las recomendaciones que se realizan en el estudio del propio *dataset* como la importancia de tener en cuenta las ayudas académicas tanto fuera como dentro de casa, acudir a actividades extraescolares y la necesidad de incluir las calificaciones trimestrales; y en la evidencia científica sobre los factores que contribuyen al rendimiento académico dada en (Ruiz-Esteban et al. 2018), (Muñoz et al. 2016) y (Gómez-Sánchez et al. 2011), cuyos resultados exponen que el sexo, el nivel socioeconómico y la autoestima son significativos.

La construcción de un cuaderno en *Jupyter*, tal y como estamos haciendo, nos permitirá modificar las variables que se podrían atender para obtener la un modelo predictivo u otro, por lo que estas podrían no ser definitivas en futuros trabajos de investigación, quedando a juicio del creador y de otra argumentación alternativa.

Procesamiento de los datos

Ahora es el momento de acomodar los datos que tenemos a los requerimientos futuros de la Red Neuronal que vamos a elaborar para el aprendizaje automático. Para ello, debemos realizar dos pasos: cambiar los datos en formato de cadena de texto (*string*) por entradas numéricas enteras (tipo *int*) y, luego, convertir estos datos a matrices (*arrays*), los cuales sí serán los que procese nuestra Red Neuronal.

Respecto al primer paso, vamos a mapear los valores que se describan como «yes» y «no» de nuestro *dataframe* por 1 y 0, respectivamente. Esto realiza cambios en las variables «famsup», «paid», «activities» y «internet». Por otro lado, mapeamos los valores «F» y «M» en la variable «sex», «U» y «R» en «address», y «GT3» y «LE3» en «famsize»; todas estas también sustituidas por 1 y 0, respectivamente.

Por último, las calificaciones trimestrales («G1», «G2», «G3») en nuestro conjunto de datos, como ya hemos nombrado, van desde 0 a 20, por lo que las mapearemos para que respondan a 0 si están por debajo de 10 (suspenso) y 1 si están por encima (aprobado). Así, cuando realicemos una predicción, obtendremos un número decimal entre 0 y 1.

De este modo, nuestro *dataframe* quedaría de la siguiente forma (5.2):

	sex	address	famsize	Medu	Fedu	studytime	famsup	paid	activities	internet	G1	G2	G3
0	1	1	1	4	4	2	0	0	0	0	0	0	0
1	1	1	1	1	1	2	1	0	0	1	0	0	0
2	1	1	0	1	1	2	0	1	0	1	0	0	1
3	1	1	1	4	2	3	1	1	1	1	1	1	1
4	1	1	1	3	3	2	1	1	0	0	0	1	1

Cuadro 5.2: Primeras 5 columnas del *dataframe* de la asignatura de Matemáticas después de la limpieza

Pasando al segundo paso, solo tendremos que ejecutar sobre nuestro *dataframe* el método *values* (*df.values*) para obtener nuestro *array*.

Ahora es el momento de dividir nuestro conjunto de datos en aquellas variables que son usadas para predecir (las doce primeras columnas, cuyos datos son almacenados en la variable «X») y lo que deseamos predecir (última columna, almacenada en la variable «Y»).

Aunque es un proceso que se realiza para convertir los datos de entrada de la Red Neuronal en valores similares entre sí cuando se poseen números muy dispares, cosa que no es nuestro caso, representa una buena práctica utilizar la función *MinMaxScaler* para que todos los valores se encuentren entre 0 y 1. Así, facilitaremos el trabajo a nuestra Red Neuronal.

Para terminar el procesamiento, dividimos nuestros datos en un conjunto de entrenamiento, un conjunto de validación y un conjunto de prueba. Para ello, usaremos la función *train_test_split*, importada desde la librería *sklearn*. Sin embargo, esta función sólo nos ayuda a dividir nuestro conjunto de datos en dos (los referidos al conjunto de entrenamiento para «X» e «Y»). Dado que también queremos obtener un conjunto de validación y un conjunto de prueba, podemos utilizar la misma función para realizar la división de nuevo.

Si imprimimos en consola lo obtenido, observamos que el conjunto de entrenamiento tiene 276 datos (75 % del *dataset*), mientras que el conjunto de validación posee 59 datos (15 % del *dataset*) y el conjunto de prueba 60 datos (15 % del *dataset*). Las variables «X» tienen 12 entradas y las variables «Y» sólo tienen una entrada para predecir, lo cual nos indica que está todo acorde con lo deseado.

Construcción de la Red Neuronal

Una vez que los datos son limpiados y procesados, tenemos todo listo para configurar la arquitectura de nuestra Red Neuronal.

Para ello, emplearemos la librería *Keras*, de la que importaremos dos funciones: *Sequential*, que nos permitirá conformar un modelo secuencial, es decir, un modelo donde tenemos que especificar las características de cada capa, las cuales se sucederán una tras otra

desde la capa de entrada, pasando por las capas ocultas hasta llegar a la capa de salida; y *Dense*, donde definiremos como primer parámetro la cantidad de neuronas que tendrá cada capa.

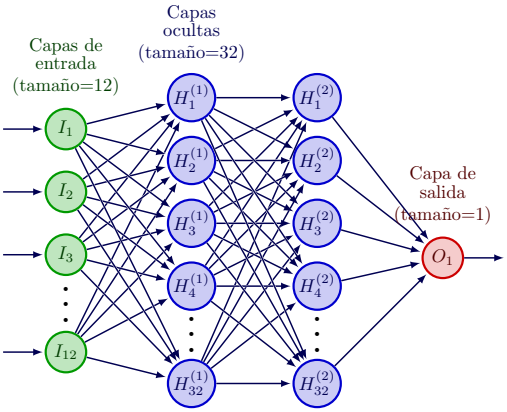


Figura 5.3: Red Neuronal para Matemáticas

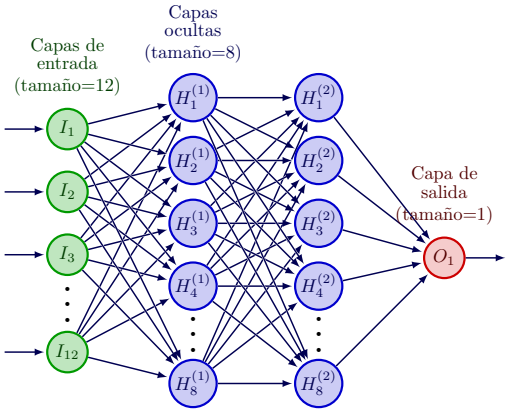


Figura 5.4: Red Neuronal para Lengua

Como vemos en la figura 5.3, nuestra Red Neuronal para la predicción en Matemáticas empleará treinta y dos neuronas en las capas ocultas, mientras que la destinada a Lengua (figura 5.4) tendrá ocho neuronas. Por otra parte, las capas de entrada recibirán un tamaño de doce, coincidente con el número de variables que vamos a introducir al modelo para conseguir la predicción de un valor concreto, lo cual se evidencia en que la capa de salida solo posee una neurona. La elección de diferentes cantidades de neuronas en cada modelo se debe a un proceso de pruebas al que fueron sometidos cada uno y en los que los resultados finales eran los más adecuados en relación a la precisión y a la menor pérdida posible (Código 5.1).

```
model = Sequential([
    Dense(32, activation='relu', input_shape=(12,)),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid'),
])
```

Código 5.1: Red Neuronal para el modelo predictivo de la asignatura de Matemáticas

Ya tendríamos nuestro modelo construido; sin embargo, necesitamos configurar tres parámetros antes de empezar nuestro entrenamiento: algoritmo de optimización (*optimizer*), función de pérdida (*loss*), métricas a almacenar (*metrics*) (Código 5.2).

```
model.compile(optimizer='sgd',
              loss='binary_crossentropy',
              metrics=['accuracy'])

hist = model.fit(X_train, Y_train,
          batch_size=32, epochs=100,
          validation_data=(X_val, Y_val))
```

Código 5.2: Configuración del entrenamiento de la Red Neuronal para la asignatura de Matemáticas

En nuestro caso hemos utilizado *sgd* como algoritmo de optimización; *binary_crossentropy* como función de pérdida, debido a que es la que se corresponde si queremos obtener una salida entre 0 y 1; y *accuracy* como métrica ya que su almacenamiento nos puede ser útil en el caso de que deseemos realizar un seguimiento gráfico de la precisión y poder compararla con la pérdida.

Como último paso, vamos a ajustar los parámetros del modelo a los datos que poseemos, por lo que tenemos que especificar sobre nuestro modelo y dentro de la función *fit* los datos de entrenamiento (*X_train* y *Y_train*); el tamaño del lote que empleará la función de optimización, habiendo escogido el mismo que el de neuronas empleadas en la arquitectura de nuestra Red Neuronal; el tiempo de entrenamiento (*epochs*), cuyo número ha sido elegido por la recomendación que realizan los autores del propio *dataset* en su estudio (Cortez & Silva 2008), tal y como comentamos anteriormente; y los datos de validación (*X_val* y *Y_val*). Todo ello almacenado bajo la variable *hist*[8].

Para evaluar nuestros dos modelos, simplemente debemos acudir a la función *evaluate* e introducirle los datos de prueba (*X_test* y *Y_test*) y obtendremos las métricas de pérdida y de precisión, expuestas en 5.3.

	Modelo para Matemáticas	Modelo para Lengua
Loss	0.2223	0.3198
Accuracy	0.8999	0.8979

Cuadro 5.3: Métricas de pérdida y de precisión: predicción éxito académico

5.2.2. Predicción de grados universitarios

Para este caso, se ha elaborado un modelo predictivo en el que el usuario introduce una entrada como cadena de texto y esta es procesada y enlazada con una de las etiquetas de un documento *JSON* que almacena intenciones, a través del peso probabilístico que posea con cada una de ellas.

Se trata de un modelo que surge de la idea inicial de realizar un asistente conversacional basado en el procesamiento del lenguaje natural del usuario para darle una respuesta acorde con la intención de su mensaje, por lo que mi investigación me llevó, en primer lugar, al artículo «Python Chatbot Project – Learn to build your first chatbot using NLTK Keras»[9], el cual hizo que comenzará a realizar mis primeras pruebas en la conformación de un *chatbot*; sin embargo, su rendimiento era óptimo para entradas en inglés, mientras que en español los resultados no eran del todo satisfactorios. Por ello, seguí con mi investigación hasta que encontré el blog «techwithtim»[10], donde se exponía un código

[8]La variable *hist* contendrá toda la información sobre nuestro modelo. Así, y aunque en la presente memoria no se detalla y desarrolla el proceso, podemos emplearla para la visualización del rendimiento del modelo en pos de ver si existe, por ejemplo, lo que se denomina *overfitting* y poder modificar y regularizar la Red Neuronal

[9]Python Chatbot Project – Learn to build your first chatbot using NLTK Keras [Online]. Available: https://data-flair.training/blogs/python-chatbot-project/. Cabe resaltar la cantidad de proyectos y conocimientos que comparte la plataforma web DataFlair, donde también podemos encontrar cursos gratuitos

[10]Tim Ruscica. Python AI Chat Bot Tutorial [Online]. Available: https://www.techwithtim.net/tutorials/ai-chatbot/part-1/

parecido al anterior artículo, pero con nuevas funcionalidades, como el ajuste en la parametrización para filtrar cuando la entrada no es enrutada a ninguna intención, por lo que el asistente responde con un mensaje de no entendimiento. Por último, llegué al artículo llamado «Contextual Chatbots with Tensorflow»[11], en el que la información sobre la construcción de un asistente conversacional con un código muy similar a los nombrados previamente alcanza un nivel de mayor profundidad, ya que introduce el concepto del almacenaje de la contextualización a lo largo de la conversación. Aun así, todo seguía respondiendo a un modelo basado en un PLN en inglés, por lo que mi objetivo fue el de adaptar el procesamiento al reconocimiento de entradas en castellano.

Para ello, los dos procesos que hay que acomodar son la *tokenización* y la derivación regresiva:

- ***Tokenización*:** se trata de la división del texto a tratar en palabras o, incluso, grupos de palabras (dependiendo de los intereses de cada estudio), siendo denominados como *tokens*. Para ello, necesitamos importar *word_tokenize* de *nltk.tokenize* para luego poder emplear la *tokenización* sobre un texto con *nltk.word_tokenize(texto, 'spanish')*.
- **Derivación regresiva (*word stemming*):** en el campo del procesamiento del lenguaje natural se refiere a la búsqueda de la raíz a través de la eliminación de los afijos de la palabra tratada. Este proceso nos permite optimizar la agrupación y clasificación de textos, así como, en nuestro caso, encontrar similitudes entre entradas según la aparición de las raíces resultantes, pudiendo obtener los pesos probabilísticos de similitud gracias a la frecuencia conseguida. A nivel teórico, se basa en el algoritmo de Porter y, a nivel de *software*, se emplea *Snowball*[12], el cual da soporte a 12 idiomas en total, siendo el castellano, el euskera y el catalán tres de ellos (Código 5.3).

```
from nltk import SnowballStemmer
# 'spanish' (castellano), 'catalan' (catalán) y 'basque' (euskera)
spanish_stemmer = SnowballStemmer('spanish')
```

Código 5.3: Adaptación de la derivación regresiva al castellano

Una vez realizado todo ello, se consiguió un asistente que pudiera responder de manera satisfactoria. Sin embargo, no era de gran utilidad para resolver los problemas que hemos planteado al inicio de esta memoria, por lo que fue necesario adaptar el código para dar respuesta a nuestras necesidades. De esta forma, se comenzó a remodelar el código planteado dando lugar a un modelo de predicción de grados universitarios en función a una entrada de texto.

Respecto a nuestro *JSON*, en el que se encuentran almacenadas las intenciones, poseemos un diccionario de diccionarios en el que cada intención tiene una etiqueta (*tag*) cuyo nombre está relacionado con un grupo de grados universitarios que mantienen relación curricular entre sí y una lista de palabras o frases vinculadas a este y que pueden aparecer en la entrada de nuestros usuarios (*patterns*) (Código 5.4).

[11] gk_ (2017, 05, 07). «Contextual Chatbots with Tensorflow» [Online]. Available: https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077

[12] Snowball (2022) [Online]. Available: https://snowballstem.org/

```
{'intents': [
        {'tag': 'bellas_artes',
        'patterns': ['plástica', 'imaginación', 'arte', 'crear', 'dibujar',
        'escultura', 'pintura']
}]}
```

Código 5.4: Ejemplo del archivo *JSON* con una etiqueta y sus patrones asociados

Debemos destacar que la elaboración de unas intenciones coherentes y adecuadas con cada grupo de grados universitarios es de gran importancia, ya que son la base que determinará la futura predicción. En nuestro caso, las palabras y frases que se han asociado a las carreras se han elegido a la vista de las guías docentes de los mismos y de la subjetiva opinión del autor, además de que la cantidad total de las mismas es baja, por lo que sería necesario agregar muchas más. Representaría uno de los puntos a mejorar de cara al futuro.

Importamos nuestro archivo *JSON* y, a través de un bucle, *tokenizamos* las frases y eliminamos signos de puntuación. Luego, almacenamos en distintas variables las etiquetas y las frases contenidas en ellas. Seguidamente, cada *token* es sometido a la derivación regresiva y almacenado en una lista, además de ser convertido a minúscula y proseguir con la eliminación de las raíces repetidas en la lista. Aunque no se ha realizado, se podría limpiar dicha lista de raíces que pueden resultar inoperantes tales como «a», «e», «la» y «lo», ejemplos que se encuentran en la lista de nuestro modelo.

Lamentablemente, estos datos no sirven como entrada para la Red Neuronal que vamos elaborar, ya que, y al igual que ocurría con el modelo previamente descrito, necesitamos una entrada numérica. Para ello, vamos a recurrir a lo que se denomina «bolsa de palabras», en el que recorremos cada frase listada en cada intención y se comprueba si cada uno de los *tokens* derivados regresivamente se encuentran en ellas, codificando con el número 1 si aparecen y con 0 si no aparecen. Así, conseguiremos una lista de listas donde se formatea la presencia de cada *token* en cada frase y que almacenaremos en la variable *train_x*.

Por otro lado, también debemos formatear la salida, la cual se introducirá en la Red Neuronal y que almacenaremos en la variable *train_y*. Se trata de una lista de listas que codifica cada una de las intenciones, donde la posición del número uno entre ceros determinará qué intención es.

Por último, convertiremos nuestras dos variables obtenidas a *arrays* con la librería *Numpy*.

Antes de comenzar a construir nuestra Red Neuronal, es importante limpiar las operaciones realizadas por defecto por *TensorFlow* y establecer el gráfico global. Destaco este punto porque tuve una serie de problemas para solucionar el error de código que obtenía, ya que *tensorflow.reset_default_graph()* no es compatible a partir de la versión 2 de *TensorFlow*. Para solucionarlo debemos emplear *tensorflow.compat.v1.reset_default_graph()*[13].

En este caso, nuestra Red Neuronal estará compuesta por dos capas ocultas de 8 neuronas, en la que, para su entrenamiento de 2000 épocas, se introducirá la codificación de la «bolsa de palabras» (*train_x*), la cual será procesada y comparada con el *array* de las

[13]Enlace a la documentación de *TensorFlow* donde se expone: https://www.tensorflow.org/api_docs/python/tf/compat/v1/reset_default_graph

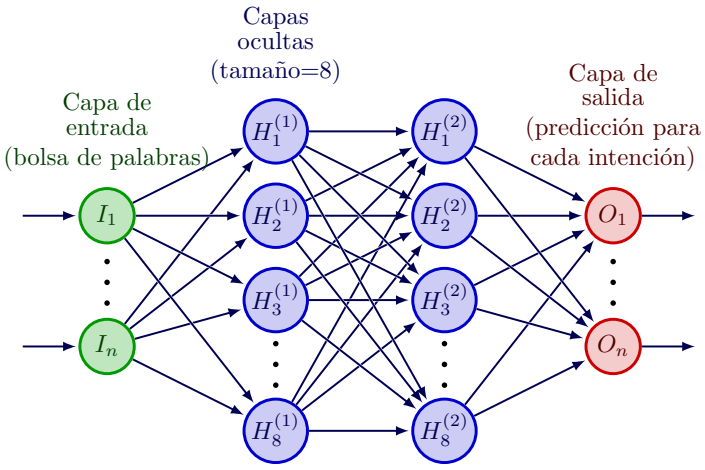


Figura 5.5: Red Neuronal para el modelo predictivo de grados universitarios

intenciones contenidas en el archivo *JSON* (*train_y*) para dar una salida codificada numéricamente, pero que contiene la probabilidad entre 0 y 1 de cada una de las intenciones dependiendo de la entrada dada. Además, aunque la arquitectura de la Red Neuronal es idéntica a la descrita previamente en el modelo predictivo anterior (Código 5.1), aquí se propone un código alternativo para su composición (Código 5.5).

```
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(train_y[0]), activation='softmax')
net = tflearn.regression(net)
```

Código 5.5: Red Neuronal para el modelo predictivo de grados universitarios

Al igual que especificamos con la anterior Red Neuronal, se debe señalar que ni la cantidad de capas ocultas ni las épocas de entrenamiento son definitivas y es de buen uso el probar con multitud de combinaciones para conseguir el que consideremos como adecuado. Gran parte del aprendizaje automático es prueba y error.

En nuestro caso, la pérdida y la precisión del modelo obtenido se refleja en 5.4.

	Modelo para grados universitarios
Loss	0.2044
Accuracy	0.8363

Cuadro 5.4: Métricas de pérdida y de precisión: predicción grados universitarios

Como último paso en la generación de nuestro modelo, guardaremos en un *pickle* todas las variables que hemos necesitado para alimentar a nuestra Red Neuronal, ya que próximamente tendremos que hacer uso de ellas cuando queramos invocar a nuestro modelo.

5.3. Desarrollo de la aplicación en Telegram

La implementación en Telegram supuso realizar una profunda investigación de su uso, de la que destaco los siguientes puntos:

- **BotFather[14]:** como primer paso, debemos crear nuestro bot en la aplicación para que sea registrado como un usuario más. Para ello, acudimos a BotFather[15], el bot oficial de Telegram que se encarga de dicha función y donde conseguiremos nuestro *token* de autenticación, el cual será la llave que conecte nuestro código con nuestro *bot* en la aplicación. Además, BotFather nos permite cambiar la foto de perfil de nuestro *bot*, su descripción inicial, etc.
- **API de Telegram:** en la web de Telegram se ofrecen distintas librerías clasificadas según el lenguaje de programación a emplear[16]. Para Python se ofrecen ocho librerías como alternativa. En nuestro caso, usaremos *pyTelegramBotAPI*, la segunda más empleada por la comunidad. Dicha elección está guiada por considerar subjetivamente que la implementación se realiza de forma más intuitiva y estructurada que en las demás. Sin embargo, cualquiera de ellas puede resultar una óptima opción.
- **Práctica:** es importante dedicar tiempo a probar la gran mayoría de funcionalidades que nos ofrece la API para ver qué se adapta mejor a las necesidades de nuestro proyecto. Con ese fin, *pyTelegramBotAPI* posee una extensa y pedagógica documentación[17] con multitud de ejemplos.

5.3.1. Estructura básica de funcionamiento de un bot en Telegram

Como vemos en el Código 5.6, para dar vida a un *bot* en Telegram necesitamos pocas líneas de código. En primer lugar, importamos el nombre de la librería *pyTelegramBotAPI* para manejar la API. En segundo lugar, creamos la variable que dará nombre al *bot* a lo largo del programa, donde incluiremos el *token* de autenticación dado por BotFather y en formato *string*. En tercer lugar, definimos el grueso del desarrollo, es decir, los comandos a los que responderá el *bot* y los caminos que tomará según transcurra la conversación. En el Código 5.6, se observa que se han definido dos comandos, los cuales se activarán al introducir \start o \empezar[18], recibiendo un «¡Hola!» por parte de nuestro bot y, seguidamente, un «¿Cómo estás?», al que debemos responder, ya que estamos en la obligación de guardar una entrada en la variable *msg*, la cual será empleada en la siguiente función, donde si respondemos con «Bien», obtenemos la respuesta «¡Me alegro!», mientras que si

[14]Acceso a la documentación de la API de Telegram: https://core.telegram.org/api

[15]Para acceder necesitamos tener la aplicación de mensajería Telegram descargada en nuestro dispositivo y escribir BotFather en el buscador de la propia *app* o pulsar en el siguiente enlace: https://t.me/botfather

[16]Enlace web a la lista de librerías que dan acceso a la API de Telegram dependiendo del lenguaje de programación empleado: https://core.telegram.org/bots/samples

[17]Enlace web a la documentación de la librería de Python que permite la conexión con la API de Telegram:https://github.com/eternnoir/pyTelegramBotAPI#message-handlers

[18]Formato identificativo de los comandos en Telegram. Además, señalar que es recomendable que el primer comando reciba el nombre '*start*', ya que es el predeterminado para dar inicio a un *bot* de Telegram y, así, evitaríamos confusiones.

respondemos «Mal», recibimos «¿Cómo puedo ayudarte para que estés mejor?»; o \adios si queremos que nos responda «¡Adiós!». En cuarto lugar, es fundamental introducir el método *infinity_polling()* al *bot* como última línea del código, cuya función es la de activar nuestro asistente conversacional y estar atento a que se produzca una interacción en la aplicación.

```
import telebot # para manejar la API

# creamos el bot
bot = telebot.TeleBot(TOKEN)

@bot.message_handler(commands=["start", "empezar"])
def start(message):
  bot.send_message(message.chat.id, "¡Hola!")
  msg = bot.send_message(message.chat.id, "¿Cómo estás?")
  bot.register_next_step_handler(msg, nombre)

def start(message):
  if message.text == "Bien":
    bot.send_message(message.chat.id, "¡Me alegro!")
  elif message.text == "Mal":
    bot.send_message(message.chat.id, "¿Cómo puedo ayudarte para que estés
     mejor?")

@bot.message_handler(commands=["adios"])
def start(message):
  bot.send_message(message.chat.id, "¡Adiós!")

bot.infinity_polling()
```

Código 5.6: Ejemplo de la estructura básica de un bot de Telegram con la librería *pyTelegramBotAPI*

5.3.2. Clasificación y módulos: funciones de nuestro asistente conversacional

Antes de comenzar con la descripción de lo que hemos denominado módulos (funciones que realiza el bot), es necesario explicitar las consideraciones previas que se tuvieron antes de comenzar y que luego se plasmaron en el desarrollo. Para ello, se ha seguido la taxonomía que expone Jesús Martín, diseñador de asistentes conversacionales, en su blog[19] para clasificar asistentes conversacionales:

- **Libertad conversacional:** en su mayoría, se ha optado por una interacción basada en comandos o botones, apostando por facilitar la navegación al usuario, aunque también se ofrece la opción de introducir la respuesta dada en los botones a través de teclado. Sin embargo, a lo largo de la conversación hay ocasiones en las que se exige responder con una entrada escrita, la cual será procesada y se dará una respuesta positiva o negativa acorde con el contexto. De esta forma, las posibilidades de interacción se multiplican y se adaptan a lo que el usuario prefiera.

[19]Jesús Martín (2022, 06, 05). «Cómo clasificar y diferenciar los distintos tipos de interfaces conversacionales» [Online]. Available: https://jesusmartin.eu/clasificar-tipos-interfaces-conversacionales/

- **Forma de interacción:** se emplea texto para la comunicación con el asistente, pudiendo ser a través de todos los dispositivos compatibles con la aplicación de Telegram. Por tanto, atendiendo a esto, podemos definir a nuestro asistente como *chatbot*.
- **Dominio de conocimiento:** se trata de un asistente especialista, ya que su dominio de conocimiento se acota a la predicción, recomendación y exposición de los planes de estudio de los distintos grados universitarios.
- **Propiedad de la plataforma:** la construcción a nivel de código es nativa y posee una licencia MIT acorde con la filosofía de *software* libre.
- **Iniciativa:** se trata de un asistente proactivo. La conversación es iniciada por el *chatbot* y es dirigida hasta conseguir el objetivo que se persigue.
- **Objetivo:** transaccional, es decir, el asistente se encarga de ejecutar una acción concreta seguida por un guion.
- **Profundidad de la conversación:** es de turno único. La interacción es sencilla y se compone de preguntas y respuestas, evitando que se genere confusión en el usuario y que la conversación sea difusa.

Módulo 1: Bienvenida y preguntas sobre sus intenciones académicas

Este primer módulo comienza con un saludo y con una serie de mensajes que describen brevemente las funciones del *chatbot* y aquello que conseguirá el usuario como fruto de la conversación. Por otro lado, se prosigue con dos preguntas que recogen el nombre y el pronombre del usuario. Dichas cuestiones son imprescindibles al inicio de la conversación para atender a uno de los factores (adaptabilidad) que conforman la dimensión humana de los asistentes conversacionales según (Zierau et al. 2020). Se ha querido hacer hincapié en la personalización de la conversación según el nombre y, sobre todo, el género elegido por el usuario, adaptando los morfemas flexivos a las exigencias del mismo con el fin de evitar la exclusión y crear una conexión al inicio para favorecer la confianza del usuario para con nuestro *chatbot*, además de representar una muestra de respeto hacia su identidad de género (Sobrien 2020)[20].

El siguiente paso es la realización de una serie de preguntas relacionadas con la visión de futuro del usuario referente a las intenciones que posee después de acabar su etapa académica en Bachillerato: si va a ir a la universidad, qué grado tiene como primera opción, qué otras opciones tiene, etc. En la Figura 5.6 se observa cómo se desarrolla la conversación justo cuando empezamos a hablar con nuestro asistente conversacional, donde se pueden ver (en la parte inferior) los botones que se ofrecen al usuario para dar respuesta a la mayoría de preguntas, además de tener la posibilidad de escribir con el teclado.

Por último, destacar que al inicio del módulo se creará un archivo en formato *CSV* para ir almacenando las respuestas del usuario, ya que estas representan una valiosa información para el propio instituto de enseñanza para utilizarlas con el fin de orientar de una u otra

[20]También se han tenido en cuenta los siguientes artículos disponibles en línea en: https://tinyurl.com/leng-inclusivo-2022-tfm y https://www.psyciencia.com/lenguaje-inclusivo-investigaciones/

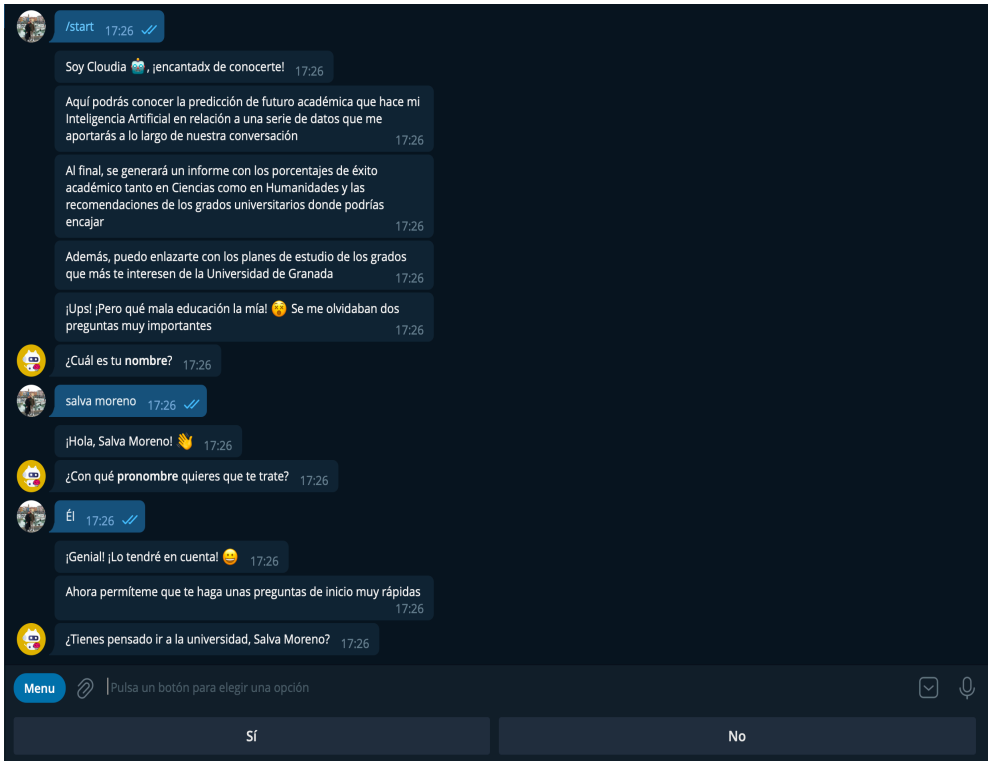


Figura 5.6: Conversación producida con el *chatbot* al comienzo de la interacción comunicativa

forma o adaptar los procesos de enseñanza-aprendizaje; e, incluso, para las instituciones universitarias, debido a que pueden conocer una estimación de la demanda de los grados universitarios.

En el Apéndice A.2 se explicita el flujo conversacional de este módulo.

Módulo 2: Predicción de éxito académico en Matemáticas y Lengua

El segundo módulo se basa en la recopilación de las respuestas necesarias a las variables que se determinaron en el primer modelo predictivo descrito en la Sección 5.2.1 (pág. 23) y que, además, se esquematizan en el Apéndice A.3, dedicado al flujo conversacional de esta parte del intercambio comunicativo.

En la Figura 5.8, podemos observar el desarrollo de las primeras preguntas que se realizan en el segundo módulo. Además, es reseñable el uso que se hace de diversos *stickers* a lo largo de la conversación, los cuales han sido elegidos según la situación comunicativa del momento y configurados para que no sean siempre iguales. Esto último para favorecer la amenidad y dinamicidad del asistente conversacional.

A nivel terminológico, es destacable el empleo de *sexo biológico* en pos de evitar la confusión que pueda existir entre los términos *sexo* y *género*, usados incluso de manera indiscriminada en las redacciones científicas, además de seguir el mismo camino y la misma filosofía descritos cuando se habló de la personalización de la conversación según el pronombre del usuario en el anterior módulo (Sección 5.3.2, pág. 32). De esta forma, basándonos en (Chafetz 2006) y (Hird 2000), se ha optado por el uso de *sexo biológico*. Por otro lado, y marcándonos al respeto hacia el usuario, se ha considerado que al preguntar

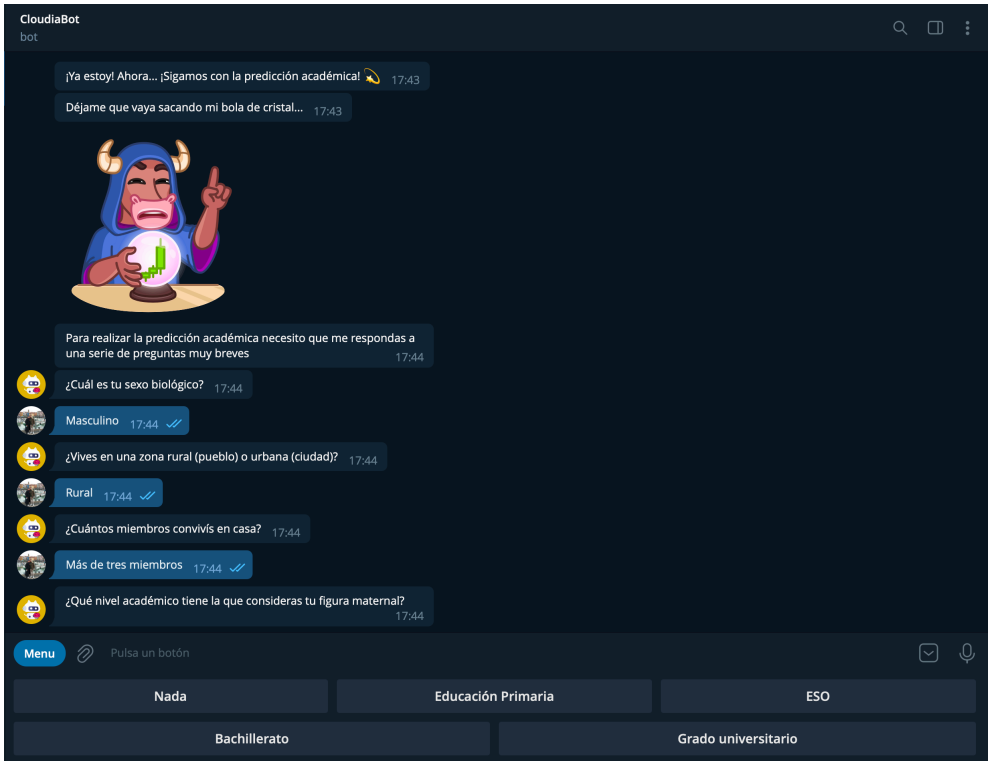


Figura 5.7: Conversación producida con el *chatbot* al inicio del segundo módulo

sobre la educación de sus padres se exprese como: ¿Qué nivel académico tiene la que consideras tu figura maternal? y ¿qué nivel académico tiene la que consideras tu figura paternal? Todo esto para intentar atender a todos los tipos de familia que se puedan dar en los usuarios.

A nivel de código, es importante señalar la implementación de los dos modelos predictivos que habíamos construido en la Sección 5.2.1 (pág. 21) para que pueda actuar una vez que hayamos recopilado la información necesaria. Sin embargo, antes de ello se creó un diccionario al inicio del módulo para almacenar todas las respuestas del usuario, las cuales son listadas en el orden que tenían las variables en el *dataframe* usado para el entrenamiento de la Red Neuronal del modelo para, luego, trasladarlos a un *array* (formato de entrada que necesita nuestro modelo para realizar la predicción). Una vez hecho esto, se importa el modelo con el método *models.load_model* proporcionado por *Keras*, se aplica, se accede al valor (*float*) dado como resultado de la predicción y se multiplica por cien para obtener el valor como porcentaje. Además, los decimales del mismo se redondearon y limitaron a cuatro. Los resultados, tanto para Matemáticas como para Lengua, se almacenan en otro diccionario con el fin de comunicárselos al usuario al final de la interacción (Código 5.3.2).

```
# entrada como array
lengua = np.array([[sexo, direccion, tamano_familia, edu_materno,
    edu_paterno, tiempo_estudio_lengua, apoyo_familia_lengua,
    academia_lengua, extraescolares, internet, primer_trimestre_lengua,
    segundo_trimestre_lengua]])
# carga del modelo
model_lengua = tf.keras.models.load_model('model.school_prediction_len')
```

```
# acceso al valor numérico de la predicción
resultado_prediccion_lengua = model_lengua.predict(lengua)[0][0]# acceder
    al float del numpy.array
# redondeo y guardado en diccionario
prediccion_ciencias_letras["lengua"] = round(resultado_prediccion_lengua *
    100, 4)
```

Código 5.7: Entrada y ejecución de los datos sobre el modelo y guardado de la predicción

Por último, se agradece y se avisa al usuario de que el proceso se ha completado, por lo que ya está preparado para proseguir al siguiente módulo.

Módulo 3: Predicción de grados universitarios

Como primer paso del tercer módulo, vamos a cargar tanto nuestro modelo creado en la Sección 5.2.2 (26) como los datos que se emplearon en la conformación del mismo, los cuales vamos a importar del *pickle* que ya creamos [enlace a código del pickle - quizá es buena idea añadirlo antes]. Además de esto, se importará el archivo *JSON* que contiene las intenciones de nuestros grados universitarios (Código 5.8).

```
# restaurar toda nuestra estructura de datos
data = pickle.load(open("./net_dnn/training_data", "rb")) # rb = lectura
    binaria
words = data["words"]
etiquetas = data["classes"]
train_x = data["train_x"]
train_y = data["train_y"]
# importar nuestro archivo de intenciones
import json
with open("./net_dnn/intenciones.json") as json_data:
  intents = json.load(json_data)
```

Código 5.8: Importación de los datos binarios y del *JSON* con las intenciones

Seguidamente, se construirá de nuevo la Red Neuronal que ya ilustramos en la figura 5.5, se definirá la variable dedicada al modelo y se cargará sobre ella el modelo resultante ya guardado a través del método *load()* (Código 5.9).

```
# definición del modelo
model = tflearn.DNN(net, tensorboard_dir='./net_dnn/tflearn_logs')
# cargar modelo guardado
model.load("./net_dnn/model.tflearn_prediction")
```

Código 5.9: Definición y carga del modelo gaurdado

Se trata de un módulo cuyo flujo conversacional es breve (ver Apéndice A.4), pero, no obstante, contiene un gran importancia en nuestro desarrollo, ya que se encarga de cerrar la predicción y de exponer los resultados al usuario, por lo que posee un tratamiento y procesamiento profundo de la información.

Como observamos en la Figura **??**, al usuario se le requiere que escriba un mensaje donde hable sobre sus asignaturas favoritas, qué se le da bien, qué le gusta, etc. y se le ofrece un ejemplo con el que se puede guiar. Además, se le ofrece si quiere añadir algo más. En caso afirmativo, esta entrada será unida a la anterior para procesarlas como un único texto.

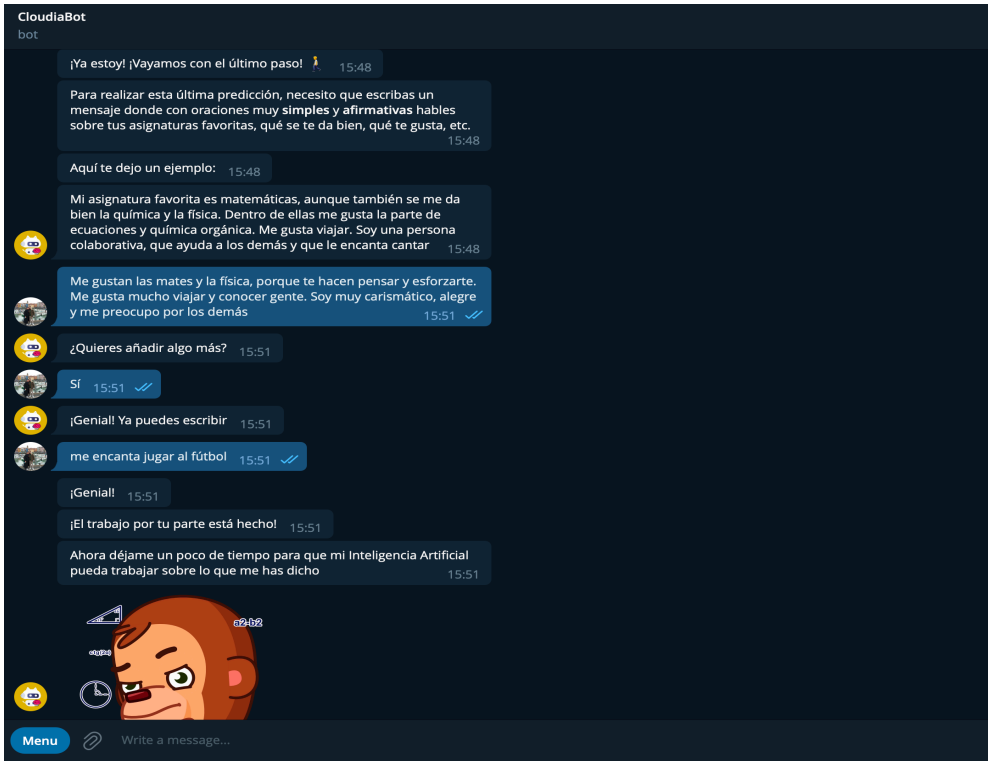


Figura 5.8: Conversación producida con el *chatbot* en el tercer módulo

Una vez que ha introducido la entrada de texto, comienza el procesamiento de la misma, cuyo proceso lo podemos dividir en 3 pasos:

- **Tratamiento de la entrada de texto para dividirla según dos signos ortográficos de unión:** la coma («,»), el punto («.») y un nexo coordinante copulativo: la i griega («y»).
- **Clasificación de los fragmentos divididos.** Este paso se puede subdividir en otros cuatro:
  - **Aplicación del modelo** a cada una de las frases extraídas previamente de la entrada de texto del usuario.
  - **Obtención de porcentajes:** paso del valor numérico entre 0 y 1 que nos da la salida del modelo a valor porcentual.
  - **Limitación de las salidas del modelo a 7:** solo se tendrán en cuenta las siete primeras y más probables intenciones detectadas.
  - **Clasificación de los grados en dos listas:** mayor probabilidad y menor probabilidad[21], así como la separación en otras dos listas de los porcentajes

[21]Se ha considerado que el valor de cribado de los grados de menor probabilidad sea establecido en un 2 % debido al comportamiento dado en las numerosas pruebas de predicción que se han realizado,

siguiendo mismo el orden.

- **Creación de un nuevo archivo *JSON* y extracción de su información:** como ya se indicó, estamos realizando la predicción sobre grupos de grados universitarios, lo que significa que necesitamos saber qué carreras están incluidas en esos grupos para transmitírselas al usuario. Así, fue necesario crear un archivo *JSON* que contuviera como etiqueta el identificador de grupo empleado en *intenciones.json*, una lista con el nombre real de los grados universitarios que se corresponden con cada intención y, además, una lista con el enlace a la web de cada uno de los planes de estudio. De esta forma, este paso nos permite individualizar los nombres de cada uno de los grados y sus correspondientes porcentajes y enlaces web dependiendo de las intenciones resultantes, todo ello tanto para los de mayor como menor probabilidad.
- **Limpieza de aquellos grados que se repiten:** se comparan las dos listas de grados de mayor y menor probabilidad y se eliminan los que coincidan en la de menor probabilidad para evitar repeticiones innecesarias y que alargarían nuestro informe final.

Tras todo este tratamiento y limpieza de la información, obtenemos 5 listas para cada grupo probabilístico: nombres de los grados universitarios individualizados y sus porcentajes; nombres de los grados universitarios agrupados por similitud en el currículo y sus porcentajes; y los enlaces web de cada grado. Todos ellos almacenados en listas y relacionados por su mismo índice entre ellas, es decir, todas las listas poseen la misma longitud. De esta forma, poseemos variedad en los formatos de salida de los grupos de grados y, además, de cada grado individualmente. Esta flexibilidad nos permite la adaptación de la visualización de los resultados según los requerimientos que vayamos a tener actualmente o a futuro.

Por último, debemos construir y generar nuestro informe final, el cual será enviado al usuario para que pueda poseer los resultados. Para ello, hemos realizado dos pasos, los cuales se asocian a dos funciones:

1) **Creación de un gráfico como imagen:** a la función *crear_imagen* se le pasan los nombres de los grados universitarios agrupados por similitud en el currículo y los porcentajes asociados a estos. Así, con la ayuda de la importación *pyplot* de la librería *matplotlib*, realizamos un gráfico circular que reflejará los grupos de grados con mayor probabilidad y que, seguidamente, exportaremos a formato *PNG*.

2) **Escritura del informe:** a la función *latex* se le envían la imagen del gráfico circular; los nombres de los grados universitarios individualizados y sus respectivos porcentajes y enlaces web, tanto para los de mayor como los de menor probabilidad; los resultados del modelo predictivo del segundo módulo; y el nombre que ha introducido el usuario al inicio de la conversación.

   Con la intención de componer un informe que muestre la información de forma estructurada y organizada, se ha elaborado un encabezado compuesto por el imagotipo del asistente conversacional en la esquina izquierda y el nombre y la función general del *chatbot* en la derecha. A continuación del encabezado, se explicita el

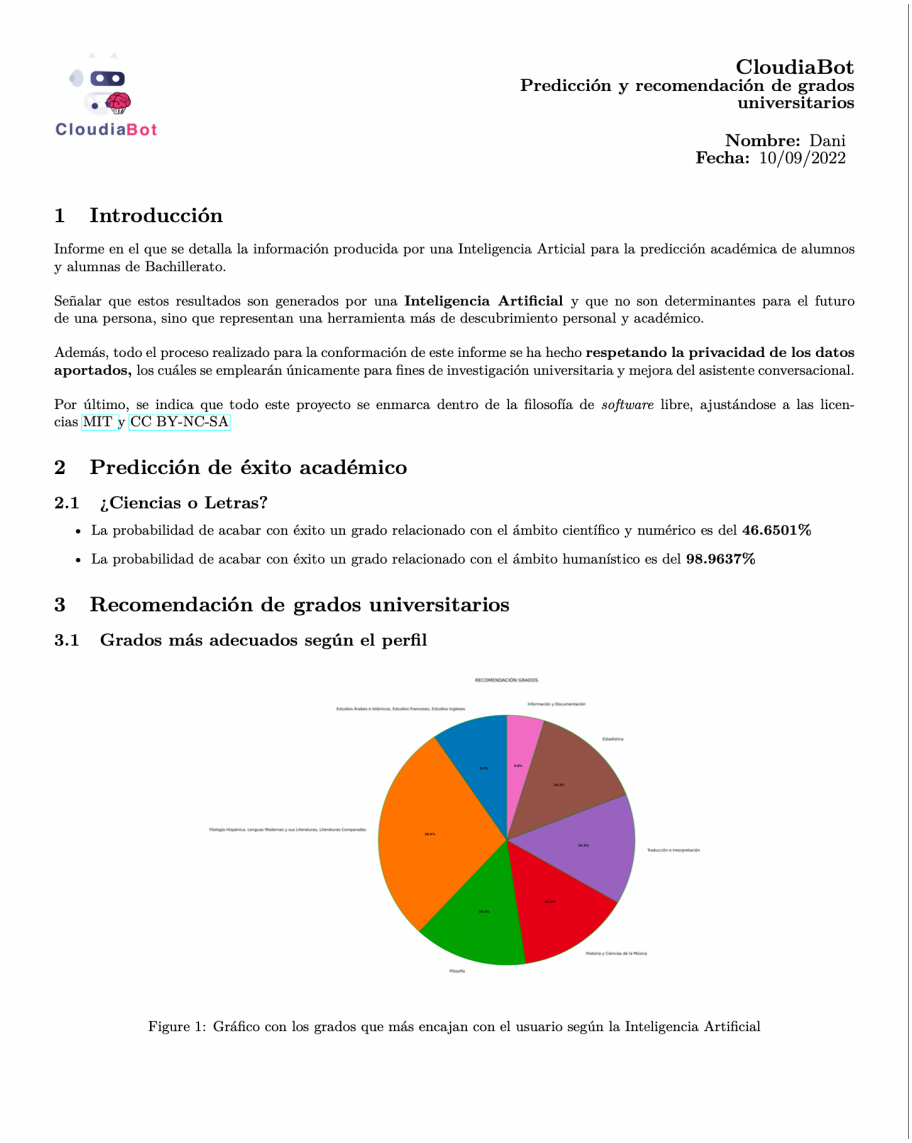
siendo el 2 % el valor que representaba el límite más significativo asociado a la menor probabilidad en la predicción.

nombre del usuario y la fecha de realización, tal y como vemos en la Figura 5.9, para luego seguir con el desarrollo de la información, la cual se compone de las siguientes secciones:

I. **Introducción:** en ella se aporta que los resultados son generados por una Inteligencia Artificial, que los datos recogidos son para fines académicos y de investigación y se indica la filosofía que sigue el proyecto.

II. **Predicción de éxito académico:** se señalan los porcentajes de éxito dados por el segundo módulo tanto para la rama científica como para la humanística.

III. **Recomendación de grados universitarios:** se expone el gráfico circular con los grados con mayor probabilidad, además de una lista con ellos de forma individual y su enlace web. Por último, también se muestra una lista con los grados que han resultado tener un grado de probabilidad menor, pero que pueden representar una opción interesante para el estudiante.



CloudiaBot

CloudiaBot
Predicción y recomendación de grados universitarios

Nombre: Dani
Fecha: 10/09/2022

1 Introducción

Informe en el que se detalla la información producida por una Inteligencia Articial para la predicción académica de alumnos y alumnas de Bachillerato.

Señalar que estos resultados son generados por una **Inteligencia Artificial** y que no son determinantes para el futuro de una persona, sino que representan una herramienta más de descubrimiento personal y académico.

Además, todo el proceso realizado para la conformación de este informe se ha hecho **respetando la privacidad de los datos aportados,** los cuáles se emplearán únicamente para fines de investigación universitaria y mejora del asistente conversacional.

Por último, se indica que todo este proyecto se enmarca dentro de la filosofía de *software* libre, ajustándose a las licencias MIT y CC BY-NC-SA

2 Predicción de éxito académico

2.1 ¿Ciencias o Letras?

- La probabilidad de acabar con éxito un grado relacionado con el ámbito científico y numérico es del **46.6501%**
- La probabilidad de acabar con éxito un grado relacionado con el ámbito humanístico es del **98.9637%**

3 Recomendación de grados universitarios

3.1 Grados más adecuados según el perfil

Figure 1: Gráfico con los grados que más encajan con el usuario según la Inteligencia Artificial

Figura 5.9: Primera página del informe generado con LaTeX

Una vez hecho esto, el usuario es avisado de que el proceso de elaboración del informe

final ha terminado y este se le envía, dando por acabado el proceso de recopilación de datos y de predicción.

Antes de terminar con este módulo, se produce la despedida con el usuario, se le aconseja sobre futuros pasos y se le agradece su participación. Luego, se da pie a que consulte el cuarto y último módulo, además de adjuntarle un enlace web a un formulario de evaluación sobre el asistente y la experiencia conversacional, el cual se adjunta en el Apéndice C.1.

Módulo 4: Planes de estudio

En el cuarto módulo se ha aprovechado una de las funcionalidades de la API de Telegram, la cual nos permite crear botones en línea, es decir, botones en el propio chat. Así, el objetivo es crear un menú, tal y como vemos en la Figura 5.10, donde el usuario podrá navegar por los distintos grados universitarios según la rama que haya elegido previamente a través de una pregunta y dirigirse, gracias a unos botones asociados al número de lista de cada carrera, a la página web del plan de estudios de la misma (véase el Apéndice A.5 para ver el flujo conversacional).

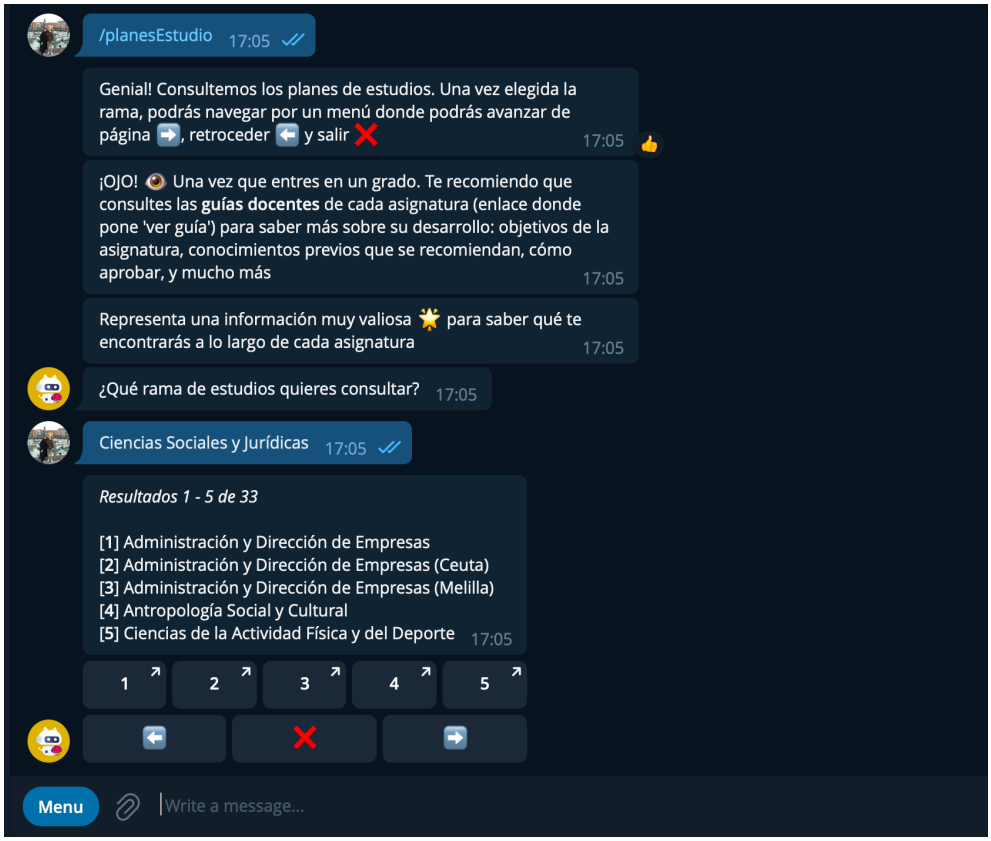


Figura 5.10: Conversación dada en el cuarto módulo que desemboca en el menú

De cara al usuario, esta parte no entraña gran elaboración; sin embargo, se están realizando varios procesos que permiten que el código tenga capacidad de recordar en qué lugar se encuentra para poder retroceder o avanzar en el menú, cuántas entradas tiene en total para mostrar y las que está mostrando, etc.

Pero, ¿de dónde sale la información que mostramos en el menú? Para ello, hemos realizado un proceso de *web scraping*, en el que con las librerías *requests* y *BeautifulSoap* se ha accedido a la página web de la Universidad de Granada[22] que alberga los grados que oferta divididos por ramas de conocimiento. De esta forma, hemos podido almacenar los primeros en listas y los segundos en variables individuales para luego asociar cada rama con sus grados en un único diccionario

Por otro lado, y para el desarrollo del menú en Telegram, se ha iniciado el método *message_handler* asociado al comando \planesEstudio y se han definido varias funciones:

- **'start':** para la bienvenida al módulo.
- **'rama_estudios':** donde se pregunta al usuario qué rama de conocimiento quiere consultar.
- **'scraping':** del diccionario se escoge la rama elegida por el usuario y se crea una lista de listas donde se introduce el nombre del grado previamente tratado para eliminar «Grado en» como primer elemento y el enlace a la web como segundo elemento.
- **'mostrar_pagina':** dicha lista es procesada por esta función para, a través de un bucle, crear un menú listado con los grados correspondientes limitado a 5 salidas, lo cual nos obliga a crear otros dos botones de navegación para avanzar y retroceder con el fin de poder consultar todos los grados. Una vez que creamos los botones con el método *InlineKeyboardButton*, es necesario incluir un argumento llamado *callback_data*, que acepta como entrada un *string*, el cual, una vez que se pulse dicho botón, será procesado y gestionado por la función *callback_query_handler*, que debemos definir antes, y cuyo argumento *func* debe ser una función *lambda* contenedora del *booleano True*[23]. De esta forma, si se pulsa en el botón asociado a «cerrar», el menú o mensaje se eliminará y desaparecerá del chat; y si, por ejemplo, se escoge el botón de «avanzar» en el menú, aparecerán las siguientes 5 salidas. Además, en un *pickle* almacenaremos en qué página del menú se encuentra el usuario para poder avisar cuando se encuentra en la primera página o en la última o, incluso, para emplearlo de forma ininterrumpida si el usuario vuelve a entrar en la conversación para consultar los grados universitarios.
- **'retorno':** pregunta al usuario si quiere volver a consultar los planes de estudio.
- **'despedida':** función dedicada a poner fin al módulo en el caso de que el usuario no quiera seguir consultando los planes de estudio.

5.4. Despliegue de la aplicación en Google Colab

Como ya hemos mencionado en en la Sección 5.1 (pág. 18), Google Colab representa una óptima opción para trabajar y desplegar algoritmos que consumen muchos recursos a nivel de *hardware*, cosa que se ajusta con la filosofía del proyecto: democratizar el conocimiento

[22]Se ha escogido la oferta de la Universidad de Granada debido a que los estudiantes de Bachillerato que han accedido a probar nuestro asistente conversacional son de la provincia de Granada. En la Sección 6 (pág. 43) se detallan las pruebas realizadas.

[23]Funciones anónimas que no se han definido previamente y que solo pueden contener una expresión

y, también, el acceso a las herramientas empleadas. Además, los cuadernos que podemos elaborar en Google Colab son muy similares a aquellos que podemos desarrollar en *Jupyter Notebook*, por lo que significa un entorno que se hace familiar para cualquier desarrollador; sin embargo, para todo aquel que no esté en contacto con el mundo del código, puede resultar una instancia poco intuitiva y difícil de comprender en un primer contacto. Para evitarlo, se ha optado por elaborar un cuaderno que, además de ofrecer la posibilidad de ver el código de desarrollo, muestre un diseño más intuitivo y que se asemeje a cualquier web a la que estamos acostumbrados visitar en nuestro día a día como usuarios. Para ello, se ha utilizado la capacidad de implementar tanto código HTML[24] como Markdown[25] en el cuaderno con el fin de constituir una interfaz, tal y como se observa en la Figura 5.11:

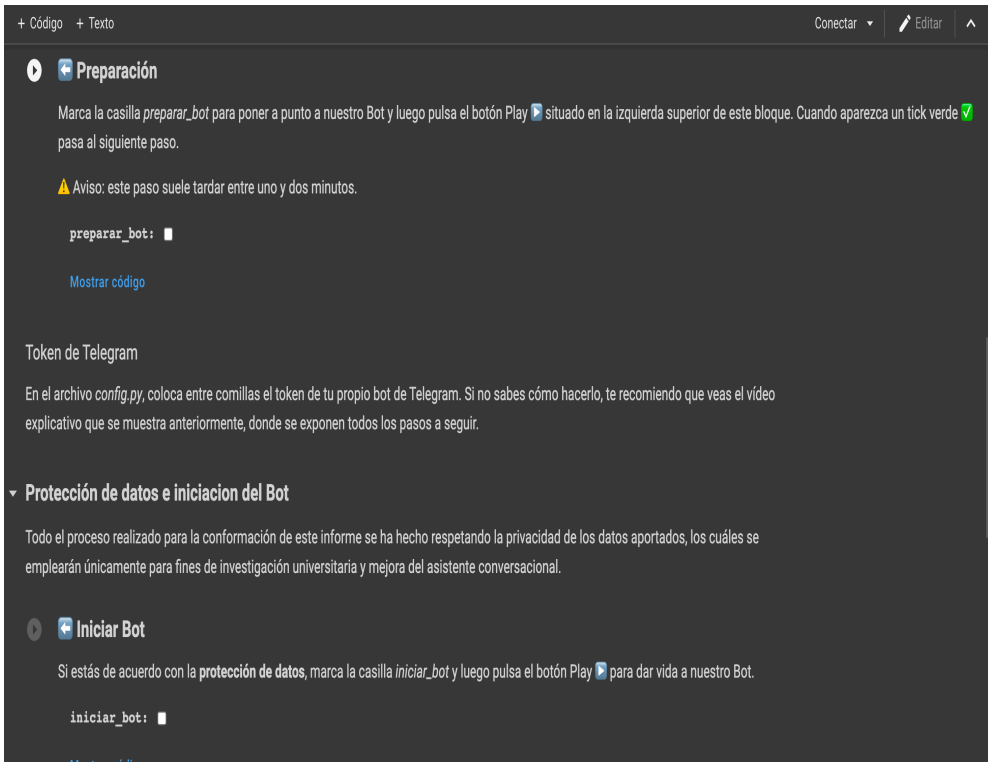


Figura 5.11: Interfaz de despliegue en cuaderno de Google Colab

El cuaderno se ha dividido en tres partes fundamentales, cuyo final es la iniciación de nuestro asistente conversacional:

a). **Vídeo explicativo:** debido a las dificultades que entrañaba el cuaderno para los usuarios que no están familiarizados, se ha realizado un vídeo[26] donde se muestra todo el recorrido necesario para conseguir activar el *bot* y comenzar la conversación en Telegram.

[24]En la documentación de la librería IPython podemos encontrar numerosas funcionalidades para otorgar interactividad a nuestros cuadernos Jupyter. Disponible en línea: https://ipython.readthedocs.io/en/stable/index.html

[25]Se recomienda visitar el siguiente enlace donde se muestran ejemplos de cómo crear formularios en nuestros cuadernos de Google Colab y dar la posibilidad al usuario tanto de ver el código como de poseer una intuitiva interfaz: https://colab.research.google.com/notebooks/forms.ipynb

[26]El vídeo ha sido grabado con la aplicación libre y de código abierto Open Broadcaster Software y subido a la plataforma YouTube en modo oculto.

b). **Preparación del entorno:** clonación del repositorio de GitHub donde se alberga todo el proyecto e instalación en la nube de *pyTelegramBotAPI*, *tensorflow*, *tflearn*, *pylatex* y lo necesario para compilar con LaTeX(*latexmk*, *texlive-latex-extra* y *pdflatex*). La duración de este proceso se completa en uno o dos minutos.

c). **Iniciación del *bot*:** aquí se señala que los datos que se pueden aportar solo se emplearán únicamente para fines de investigación universitaria y mejora del asistente conversacional y, a continuación, se ejecuta el *script bot.py* del proyecto para iniciar todo el programa.

d). **Acceso al *bot* en Telegram:** por último, se indica cómo dirigirse a Telegram para conversar con el asistente conversacional.

Capítulo 6

Presentación de la investigación y análisis de los resultados

Con el objetivo de corroborar su funcionamiento y extraer cuáles son las impresiones de los usuarios, nuestro asistente conversacional ha recorrido dos fases:

1) **Participantes que han superado Bachillerato y su etapa universitaria:** fase para comprobar que todo funcionaba correctamente y que nos ha permitido ver las impresiones de los usuarios que ya han superado el proceso de elección de una carrera universitaria siendo alumnado de Bachillerato, por lo que representaba una visión muy interesante que analizar. El número de participantes ha sido de 17 personas, mientras que 15 de ellas han realizado el formulario de evaluación de la experiencia[1], del que podemos destacar las siguientes conclusiones:

    - **Agradable y fluido:** como vemos en la Figura 6.1, los participantes se han sentido cómodos durante la conversación, cosa que nos indica que el lenguaje empleado es adecuado al contexto social y a la situación comunicativa. Además, destacaron la fluidez de los mensajes, lo cual hacía dinámica la conversación y evitaba entrar en esperas innecesarias y aburridas.

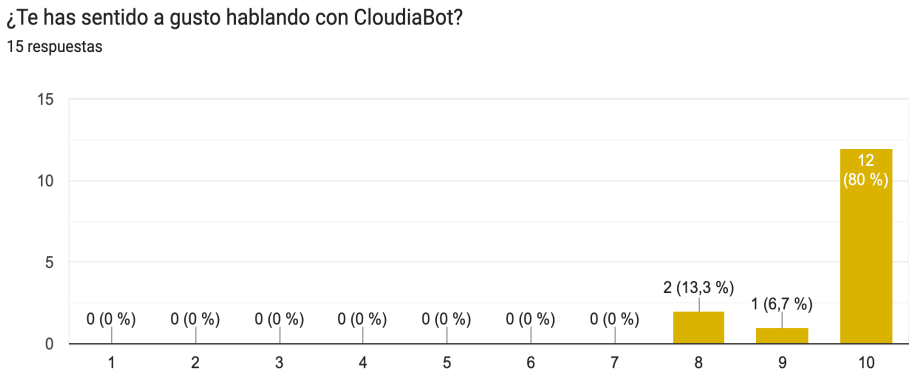


Figura 6.1: Primera pregunta del formulario de la primera fase de pruebas

[1]Este formulario fue adaptado para esta tipología de usuarios y no es exactamente igual que el que se entregará a los estudiantes de Bachillerato (Apéndice C.1)

- **Predicciones que contentan:** en la Figura 6.2, un 60 % calificó con la máxima puntuación que las recomendaciones finales fueron de ayuda; sin embargo, y con respecto al resto de preguntas, hay un 40 % que no está completamente de acuerdo con ellas, incluso en la quinta pregunta del formulario un usuario expresó: «Mejor predicción»; por lo que, todo esto, representa una señal más de la mejora que hay que realizar sobre los modelos de predicción. Además, es un indicador muy relevante debido a la fiabilidad de unos participantes que ya conocen qué ha sido de su futuro más allá de la universidad.

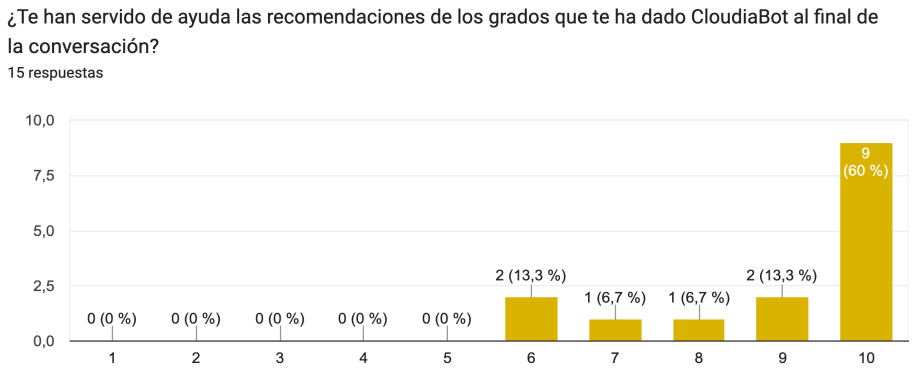


Figura 6.2: Segunda pregunta del formulario de la primera fase de pruebas

- **Gran porcentaje de éxito en el proceso de ejecución del programa:** observamos en la Figura 6.3 que un 86,7 % de los usuarios han completado la conversación sin ningún tipo de problema, frente a dos usuarios que sufrieron un mismo problema a lo largo del primer módulo, el cual estaba relacionado con el guardado en el archivo *CSV* de las respuestas, dando un *key error*, el cual normalmente viene producido por la ausencia de una entrada. Sin embargo, y como otros usuarios no tuvieron este mismo problema, resulta complicado saber qué ocurrió. Aún así, se realizó un repaso del código y se introdujeron los siguientes cambios, los cuales tienen que ver con la optimización del sistema de lectura y guardado del diccionario al archivo *CSV*:

  - A través de la librería *os*, se rastrea si el archivo *CSV* de guardado de datos está creado para evitar acumulación de nuevas cabeceras en el documento.
  - Eliminación de los datos del diccionario de Python que acumula las entradas de los usuarios con el objetivo de que no se guarden en el *CSV* los datos actuales y los anteriores, ya que desencadenaba que el *CSV* final contuviera mucha información repetida.

  Por otro lado, se ha considerado que el error haya podido venir por la coincidencia de dos o más usuarios dentro de los flujos de conversación, aunque esta opción se presenta como la menos probable, se consideraría buena práctica la creación de dos diccionarios donde guardar la información del primer módulo con sus respectivos archivos *CSV*.

- **Herramienta de utilidad:** en la Figura 6.4 vemos que existe una opinión unánime respecto a si los participantes hubieran requerido nuestro asistente conversacional durante su etapa en Bachillerato para tener en cuenta las

Figura 6.3: Cuarta pregunta del formulario de la primera fase de pruebas

predicciones y recomendaciones con el objetivo de descubrir las opciones universitarias ofertadas.

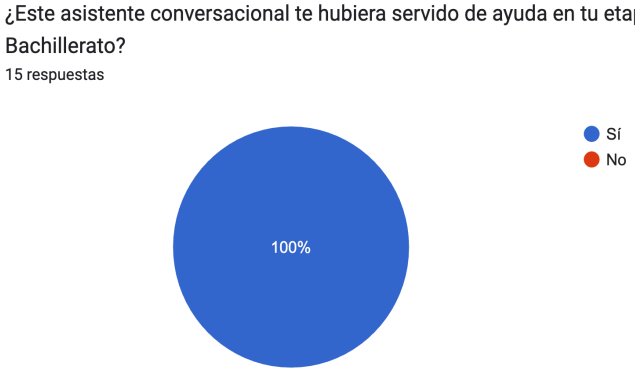


Figura 6.4: Última pregunta del formulario de la primera fase de pruebas

Debemos apuntar que esta fase se ha realizado de forma telemática, siendo iniciada la aplicación desde un único dispositivo por un administrador con el fin de facilitar la tarea a los participantes y evitar solapamientos en el despliegue.

Sin embargo, esta modalidad implicaba resolver un problema: la desactivación del programa pasado un límite de tiempo de inactividad para con el cuaderno de Google Colab. Esta cuestión se resolvió automatizando la pulsación de un botón del propio cuaderno interviniendo en la consola del navegador, introduciendo código JavaScript en dos pasos: función que apunta al botón situado en la barra de navegación con el identificador *connect*, que al ser llamada ejecuta el *click* (Código 6.1); y definición de la variable que llamará a la función y que la hará actuar en un intervalo de 3 minutos (Código 6.2). Realizado esto, nuestro asistente conversacional estaría activo durante todo el tiempo necesario, simulando la experiencia de estar albergado en un servidor de forma continua.

```
1    function ConnectButton(){
2    console.log('Connect pushed');
3    document.querySelector('#top-toolbar > colab-connect-button').
    shadowRoot.querySelector('#connect').click()}
4
```

Código 6.1: Primer paso

```
var connect = setInterval(ConnectButton, 180000)

```

Código 6.2: Segundo paso

Durante el desarrollo de la misma, se aprovechó para realizar una serie de pruebas relacionadas con la activación de nuestro asistente conversacional desde Google Colab, ya que se nos pueden presentar varios escenarios[2]:

- **Inicio de la aplicación desde un punto y que sea operable en la misma zona geográfica:** la prueba se realizó con participantes compartiendo la misma red wifi y situados en la misma ciudad. Esta fue superada con éxito.
- **Inicio de la aplicación desde un punto y que sea operable en otra zona geográfica:** la prueba se efectuó con usuarios pertenecientes a las ciudades de A Coruña y Granada, mientras que la aplicación fue activada en Madrid. El asistente conversacional no se vio afectado y no hubo problemas, por lo que la prueba fue superada con éxito.
- **Inicio simultáneo de la aplicación desde dos dispositivos diferentes conectados a la misma red:** la prueba no fue superada y no se puede realizar esta acción, ya que obtenemos en consola un error[3] que nos avisa de que no podemos invocar la API de Telegram con el mismo *token*.
- **Inicio simultáneo de la aplicación desde dos dispositivos diferentes conectados a diferentes redes:** ocurre lo mismo que en el caso anterior, no podemos tener el mismo *bot* de Telegram activado desde dos consolas.

2) **Participantes de Bachillerato:** entramos en la fase donde testeamos nuestro *chatbot* con alumnos y alumnas que se encuentran en el primer o segundo curso de Bachillerato. En este caso, el número total de participantes ha sido de 13 personas, siendo 6 las que han realizado el formulario de evaluación de la experiencia[4], obteniendo unos resultados muy similares a los de la fase anterior y de los que señalamos lo más relevante:

- **Sentimiento de comodidad a la hora de conversar:** como se observa en la Figura 6.5, el 83,3 % ha indicado con la máxima calificación que ha estado a gusto con nuestro *bot*. Con esto, podemos enunciar que el lenguaje y el modo de expresión adaptado para el asistente conversacional es adecuado y amigable. Además, volvemos a destacar una de las opiniones que se repiten en las dos pruebas y que tiene que ver con la rapidez y dinamicidad que posee el *chatbot* a la hora de interactuar contigo. En relación a ello, uno de los usuarios, en la quinta pregunta del formulario reflejó lo siguiente: «Lo bueno es que enseña muchos grados y contacta muy rápido conmigo».
- **Predicciones:** en la Figura 6.6 se examina que los participantes poseen una buena opinión sobre las predicciones que han obtenido, incluso mejores que las

[2]Todo esto teniendo en cuenta que se usa el mismo *API Token* de Telegram

[3]Error: telebot.apihelper.ApiTelegramException: A request to the Telegram API was unsuccessful. Error code: 409. Description: Conflict: terminated by other getUpdates request; make sure that only one bot instance is running

[4]Apéndice C.2 para consultar formulario entregado

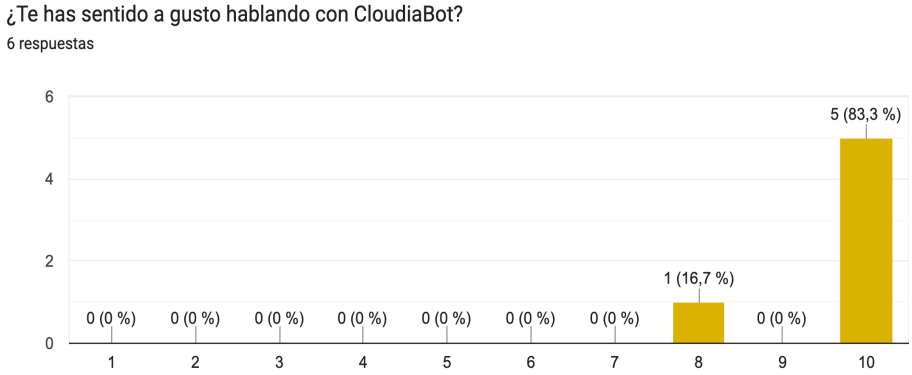


Figura 6.5: Primera pregunta del formulario de la segunda fase de pruebas

recogidas en la fase anterior. Así, podemos sacar la conclusión de que, aunque los modelos predictivos tengan mucho margen de mejora, la variabilidad de los resultados arrojados por la Inteligencia Artificial puede resultar de agrado para aquellas personas que se encuentran indecisas o que necesitan descubrir más opciones.

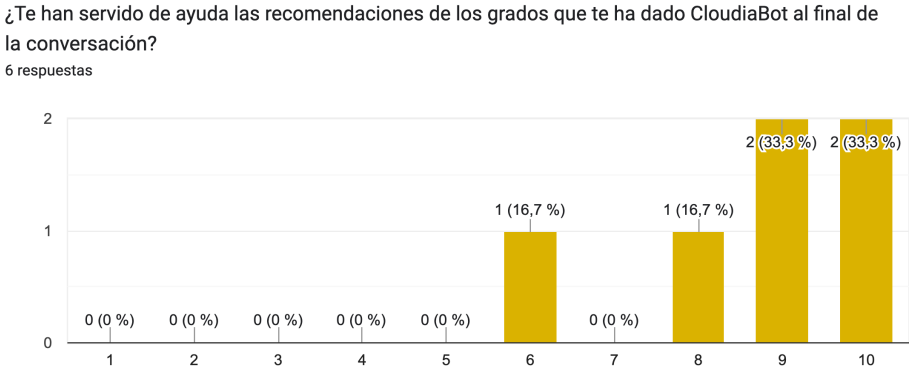


Figura 6.6: Segunda pregunta del formulario de la primera fase de pruebas

- **Desarrollo de la fase sin ningún problema:** en la Figura 6.7 vemos que los encuestados han reflejado que no han sufrido ningún fallo durante la ejecución del programa. Esto nos apunta que, posiblemente, la optimización del código realizada después de la primera fase haya servido como mejora, señalándonos la importancia de una evaluación en distintas fases de prueba de cualquier herramienta tecnológica en pos de ofrecer la mejor experiencia a nuestro público objetivo.

6.1. Trabajos a futuro

6.1.1. A nivel proyecto

- **Aportaciones de los usuarios participantes en las pruebas del asistente conversacional:** la quinta pregunta de cada formulario presentado tanto en la primera como en la segunda fase de pruebas versa sobre qué tema les sería interesante

Figura 6.7: Cuarta pregunta del formulario de la segunda fase de pruebas

añadir a la conversación para que CloudiaBot pueda informar o ayudar al alumnado de Bachillerato y las respuestas más relevantes fueron las siguientes:

- **«La conversación es bastante completa y los resultados son claros y pueden ayudar al alumno a tener una idea sobre a qué grado universitario acceder, por esa razón añadiría más preguntas sobre más asignaturas y gustos del alumno para tener una información aún más precisa del alumno y su perfil académico»:** aunque en este trabajo se ha apostado por un diagnóstico dado en un breve espacio de tiempo de conversación, la capacidad de este proyecto de desarrollarse más allá es inmensa, pudiendo, tal y como requiere este usuario, realizar más preguntas y ahondar en aspectos académicos y personales. Además, el recorrido de este proyecto puede no poseer límites, ya que se pueden crear diversos canales de comunicación que vayan creando un recorrido de descubrimiento para el alumnado (orientación académica y vocacional a través de una narrativa transmedia).
- **«Para la gente que no sabe qué estudiar, estaría bien empezar por sus sueños o aspiraciones»:** podría ser adecuado crear un nuevo flujo para aquellos que no saben si van a ir a la universidad e indagar sobre las posibilidades y vías que existen más allá del camino universitario.
- **«Rama preferida»:** ante este respuesta se me ocurre que si en la predicción de éxito académico en relación a las Ciencias y las Humanidades existe una gran diferencia entre los porcentajes, cribar la recomendación de grados universitarios en función al porcentaje mayor obtenido. Por ejemplo, si se consigue un porcentaje del 90 % en Ciencias y un 20 % en Humanidades, tiene sentido que, en este caso tan pronunciado, solo se muestren grados universitarios de Ciencias en la recomendación.
- **«Quizás, aunque igual es difícil, saber dónde vive la persona y si estaría dispuesta a desplazarse y cuánto. Para llevarlo a todas partes porque creo que podría ser de muchísima ayuda para todos los estudiantes de bachiller»:** es relevante la adaptación de las recomendaciones de grados teniendo en cuenta la universidad de destino y a un nivel nacional.

- **Optimización del código** para reducir el consumo de recursos, por lo que sería necesaria una revisión de todos los pasos realizados.

- **Unificación de la terminología de las variables empleadas,** ya que no se ha seguido un mismo criterio y existen unas en inglés y otras en español.
- **Informe generado con LaTeX**: hay varios puntos a mejorar, ya que la idea a futuro sería conseguir el diseño de un informe clínico:
  - Gráfico a generar en el propio código de producción de LaTeXpara ahorrar la importación de una imagen previa conformada con *matplotlib*.
  - Mejorar la intuitividad de los enlaces mostrados en el informe, siendo necesario introducir un color a la fuente para indicar que son *links*.

6.1.2. A nivel de investigación

- **Catalán y euskera:** la adaptación al castellano dada a lo largo de este proyecto es totalmente extensible a las lenguas catalana y vasca, por lo que se abre la puerta a la reutilización del código con el fin de aplicarlo a dichas lenguas.
- **Análisis de sentimiento:** tratamiento de oraciones con sentimiento negativo en la entrada del usuario en el tercer módulo, ya que si, por ejemplo, se introduce «odio la lengua», esta frase nos llevará a la recomendación del grupo de grados de filología. De esta forma, una posible solución sería analizar la negatividad de las oraciones y clasificarla con la opción contraria dada por el modelo, obteniendo, así, la antítesis de la predicción.
- **Mejora del procesamiento de la entrada en el tercer módulo:**
  - Como ya hemos mencionado, el texto introducido por el usuario es separado por unos límites: el punto «.», la coma «,» y la i griega «y». No obstante, esto provoca que expresiones como «Además,» o «Sin embargo,» sean analizadas y clasificadas por el modelo, siendo trozos sin significado para nuestro estudio. Habría que eliminar aquellas frases que vayan antes de «,» y que no tengan peso semántico.
  - Añadir como divisores de las frases los nexos más importantes y comunes como los correspondientes a las oraciones copulativas y adversativas como mínimo, además de signos ortográficos como el punto y coma («;») y los dos puntos («:»).
- **Mejora de las frases relacionadas a cada grupo de grados universitarios del archivo *intenciones.json*:** en este proceso es relevante realizar una serie de acciones como analizar aquellas entradas que introduce el usuario (guardadas en *entradas_usuarios.json*) y clasificar todo aquello que encaje con cada grado; y hacer una encuesta a la comunidad universitaria perteneciente a cada grupo de carreras universitarias que se reflejan en las intenciones para saber la opinión de aquellas habilidades, gustos, etc. generales que encajan con sus respectivos estudiantes con el objetivo de considerar el máximo número de variables para realizar el entrenamiento del modelo predictivo.
- **Accesibilidad:** se trata de un punto muy importante y en el que nuestro proyecto tiene mucho que mejorar. Sería idóneo adaptar nuestro *bot* para que responda tanto a texto como a audio y que, además, dé las respuestas en estos dos formatos, siendo,

así, tanto un *chatbot* como un *voicebot*. Para ello, si el usuario envía su entrada a través de un audio, este deberá ser pasado a texto para luego ser procesado y dar una respuesta. Dicha respuesta, como hemos dicho, deberá estar programada para darla en texto y en audio, cosa que es totalmente posible con la API de Telegram que venimos empleando. Por todo ello, sería muy interesante ver cómo se puede plantear esta interacción y qué efectos tendría en el usuario, además de la vía de investigación que supondría el estudiar la accesibilidad de los asistentes conversacionales en todo tipo de ámbitos.

Capítulo 7

Conclusiones

La opinión de los usuarios que han probado nuestro asistente conversacional ha sido satisfactoria y ha señalado que se han cumplido con la mayoría de objetivos marcados al inicio de la investigación cuando se propuso la conformación de un asistente conversacional: predecir, recomendar e informar a los estudiantes. Sin embargo, y como ya hemos detallado en la Sección 6.1 (pág. 47), quedan muchos puntos en los que mejorar, de los que destacaría, sobre todo, la necesidad de aumentar la efectividad de los modelos predictivos expuestos realizando cambios en la parametrización detallada en esta memoria e, incluso, probando otros algoritmos, y alimentando los mismos con datos mucho más completos y enriquecidos, asunto que daría para otro trabajo de investigación. Así, el paso más inmediato comprendería la optimización de dichos modelos y su prueba con una muestra de usuarios mayor.

Como vemos, se ha alcanzado la creación de una herramienta que, aunque tenga mucho que progresar y recorrer un camino de perfeccionamiento, se ajusta a la definición de Inteligencia Artificial de (Rouhiainen 2018) que exponíamos en la Sección 4.1 (pág. 9), ya que usamos algoritmos que aprenden de los datos para tomar decisiones. Todo esto sumado a un entorno que, una vez desarrollado, es rápido (cosa que agradecen los usuarios que conversan con el *chatbot*), intuitivo y accesible a cualquier persona.

Además, la experiencia que me ha aportado la realización de diversas pruebas para el testeo del proyecto con usuarios reales me ha llevado a la conclusión de que, en pos de obtener el mayor rendimiento de nuestra herramienta, un uso presencial de la misma es más adecuado que el realizado aquí, ya que se pueden solucionar problemas de funcionamiento al instante, realizar recomendaciones de uso, etc. De esta forma, pienso que el contexto idóneo es aquel en el que un orientador educativo activa y monitoriza el asistente conversacional mientras que los estudiantes conversan con el *chatbot* al mismo tiempo. Aún así, como hemos comprobado, un uso telemático es totalmente posible.

También, debemos señalar que la alternativa propuesta a un despliegue en la nube ha dado un resultado óptimo, proponiendo otro tipo de instancia donde iniciar programas, independientemente de los recursos y sistema operativo del usuario, apuntando hacia el democrático acceso del conocimiento. Además, nos permite una compartición fácil y amigable para aquellos que no tengan avanzados saberes o se estén iniciando en el aprendizaje.

Por otra parte, se ha evidenciado el guardado de datos no sensibles, del que solo quedaría realizar un tratamiento de su información para lograr conformar un informe que la muestre de forma visual al profesorado, personal universitario, etc. con el propósito de que les ayude a tomar las mejores decisiones a la hora de adaptar su currículo, oferta...

Los asistentes conversacionales tienen mucho que decir en el ámbito educativo. No como sustitutos, sino como complementos de apoyo a las funciones que conforman los procesos educativos producidos en cada uno de los centros. No son la solución a nada, sino la ayuda que pretende aportar su grano de arena a las prácticas a realizar sobre el crecimiento humano de cada alumno y alumna. Sin embargo, y como hemos visto a lo largo de esta memoria, la construcción de un *chatbot* entraña conocimientos técnicos que complican su acceso entre el personal educativo, por lo que estamos en la necesidad de abrir las vías de investigación que rodean a la utilidad de los asistentes conversacionales en Educación: Inteligencia Artificial, experiencia de usuario, interfaz de usuario, diseño conversacional y la convergencia de todas en herramientas de administración, resolución de preguntas, motivación, evaluación, etc. De esta forma, podremos conocer los efectos educativos y sociales de su implementación, y, sobre todo, su viabilidad económica.

Índice de figuras

Índice de cuadros

Índice de códigos

Bibliografía

Allard, J. & Atalla, N. (2009), *Propagation of Sound in Porous Media*, second edn, John Wiley and Sons, Chichester, pp. 72–89.

Allison, D. (2012), 'Chatbots in the library: is it time?', *Library Hi Tech* .

Bühler, K. (1967), *Teoría del lenguaje*, Selecta de Revista de Occidente, pp. 207–210.

Carayannopoulos, S. (2018), 'Using chatbots to aid transition', *The International Journal of Information and Learning Technology* .

Chafetz, J. S. (2006), *Handbook of the Sociology of Gender*, Springer Science & Business Media.

Chomsky, N. (1979), *Reflexiones sobre el lenguaje*, Ariel.

Chomsky, N. (1989), *El conocimiento del lenguaje*, Alianza.

Colby, K. M., Weber, S. & Hilf, F. D. (1971), 'Artificial paranoia', *Artificial Intelligence* **2**(1), 1–25.

Cortez, P. & Silva, A. M. G. (2008), 'Using data mining to predict secondary school student performance'.

De Montaigne, M. (1997), *Ensayos escogidos*, Vol. 9, Unam, pp. 64–109.

de San Juan, J. H. (1846), *Examen de ingenios para las ciencias...*, Imp. de Ramón Campuzano.

Deng, L. & Liu, Y. (2018), *Deep learning in natural language processing*, Springer.

Donoso, T. & Figuera, M. P. (2007), 'Niveles de diagnóstico en los procesos de inserción y orientación profesional', *Electronic Journal of Research in Education Psychology* **5**(11), 103–124.

Garcia, A. M. (2014), 'Rendimiento académico y abandono universitario modelos, resultados y alcances de la producción académica en la argentina'.

Garcia Brustenga, G., Fuertes Alpiste, M. & Molas Castells, N. (2018), 'Briefing paper: los chatbots en educación'.

García Villegas, P. et al. (1965), 'El movimiento internacional de orientación profesional', *Revista de educación* .

Gayol, V. & Melo Flórez, J. (2017), 'Presente y perspectivas de las humanidades digitales en américa latina', *Mélanges de la Casa de Velázquez* **47**, 281–284.

Gelbukh, A. (2010), 'Procesamiento de lenguaje natural y sus aplicaciones', *Komputer Sapiens* **1**, 6–11.

Gómez-Sánchez, D., Martínez-López, E. I. & Oviedo-Marín, R. (2011), 'Factores que influyen en el rendimiento académico del estudiante universitario', *Tecnociencia Chihuahua* **5**(2), 90–97.

Grañeras, M. & Parras, A. (2009), 'Orientación educativa: fundamentos teóricos, modelos institucionales y nuevas perspectivas. cide. ministerio de educación. gobierno de españa. secretaría general técnica'.

Hird, M. J. (2000), 'Gender's nature: Intersexuality, transsexualism and the 'sex'/'gender'binary', *Feminist theory* **1**(3), 347–364.

Liddy, E. D. (2001), 'Natural language processing'.

Llisterri, J. & Moure, T. (1996), Lenguaje y nuevas tecnologías. el campo de la lingüística computacional, *in* 'Avances en lingüística aplicada', Universidade de Santiago de Compostela, pp. 147–228.

Marietto, M. d. G. B., de Aguiar, R. V., Barbosa, G. d. O., Botelho, W. T., Pimentel, E., França, R. d. S. & da Silva, V. L. (2013), 'Artificial intelligence markup language: a brief tutorial', *arXiv preprint arXiv:1307.3091* .

Mendoza, M. T. M. (2000), *Eleccion de carrera profesional: Visiones, promesas y desafios.*, Cambridge University Press.

Minsky, M. (1988), *Society of mind*, Simon and Schuster.

Muñoz, A. S., Serrano, R. M. & Urbieta, C. T. (2016), 'La autoestima infantil, la edad, el sexo y el nivel socioeconómico como predictores del rendimiento académico', *Revista de investigación en educación* **14**(1), 33–66.

Ojea Rúa, M. (2014), 'Programa de orientación vocacional construye', *Programa de orientación vocacional Construye* pp. 1–192.

Platón, J. L. C. (1983), *Crátilo (Introducción, traducción y notas de J. L. Calvo)*, Gredos, pp. 388b–c.

Ramos, I. (1989), 'El fracaso como desencadenante de fracasos en la escuela', *Estudios sobre las depresiones* **36**, 81.

Rascovan, S. (2014), *Orientación vocacional: Una perspectiva crítica*, Paidós.

Rojas Castro, A. (2013), 'Las humanidades digitales: principios, valores y prácticas'.

Rouhiainen, L. (2018), 'Inteligencia artificial', *Madrid: Alienta Editorial* .

Ruiz-Esteban, C., Méndez, I. & Herrero, Á. D. (2018), 'Evolución de las metas académicas en función del sexo y la edad y su influencia en el rendimiento académico en adolescentes murcianos', *Educatio Siglo XXI* **36**(3 Nov-Feb1), 319–332.

Sobrien, E. M. (2020), 'Explorando lo no-binario: Un proyecto sobre el lenguaje inlcusivo, los pronombres de género, y el género no-binario en español'.

Thrun, S. (2017), 'Artificial intelligence-q&a with sebastian thrun', *Udacity, YouTube* **13**.

Wallace, R. S. (2009), The anatomy of alice, *in* 'Parsing the turing test', Springer, pp. 181–210.

Wintner, S. (2009), 'Last words: what science underlies natural language engineering?', *Computational Linguistics* **35**(4), 641–644.

Zierau, N., Wambsganss, T., Janson, A., Schöbel, S. & Leimeister, J. M. (2020), The anatomy of user experience with conversational agents: A taxonomy and propositions of service clues, *in* 'International Conference on Information Systems (ICIS).-Hyderabad, India'.

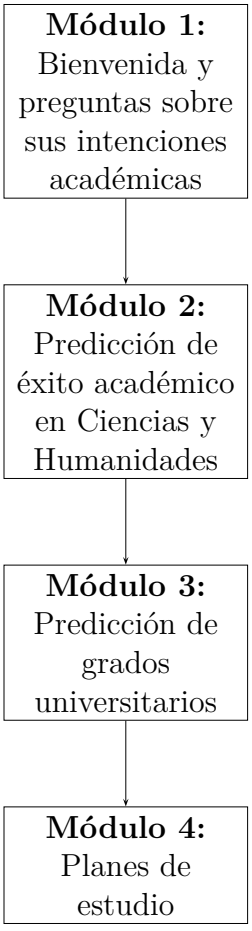
Apéndices

Apéndice A

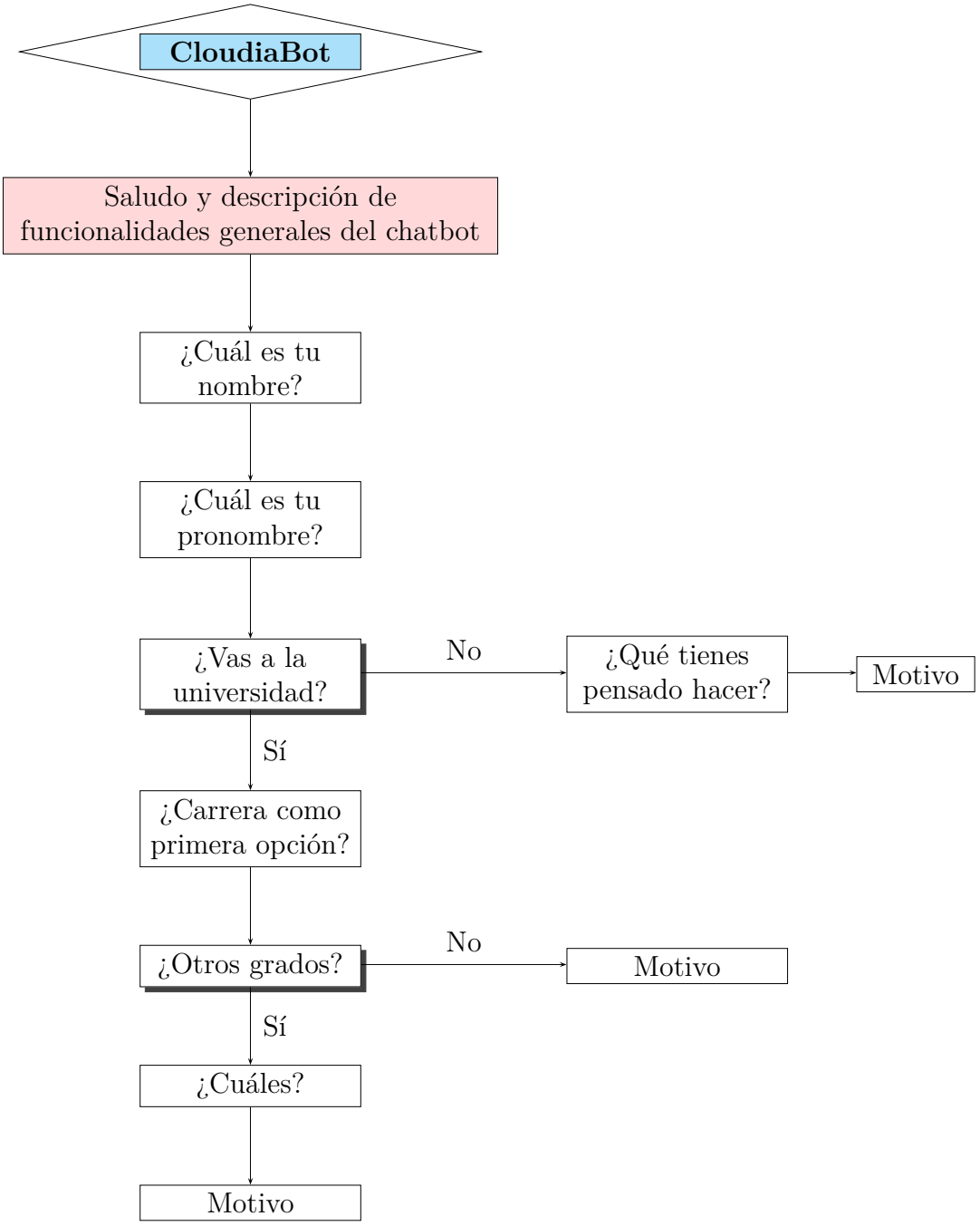
Flujos conversacionales

A.1. Estructura general

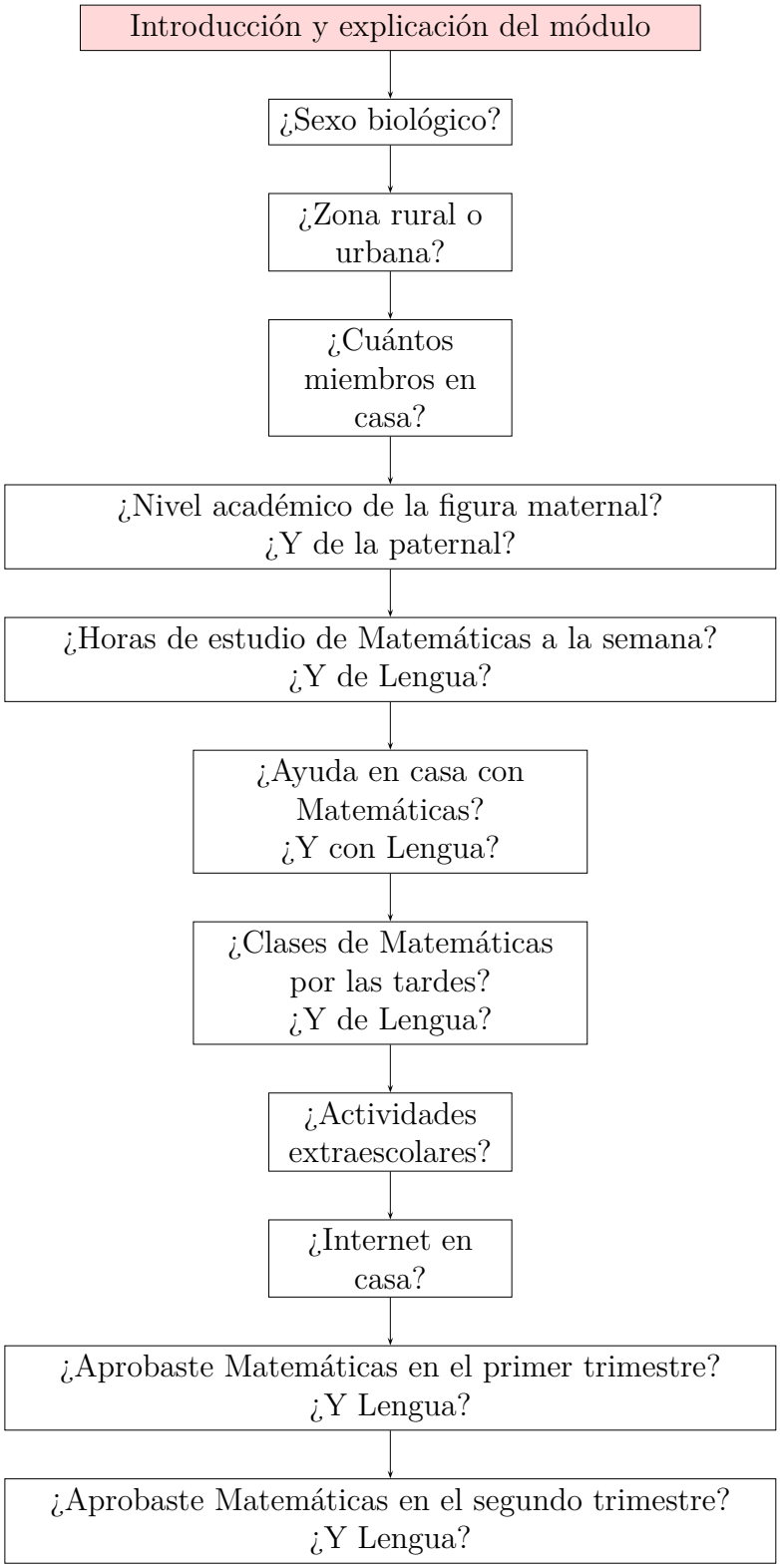
Como se observa, la dirección del flujo conversacional es lineal y no recursiva con lo que respecta a sus cuatro módulos.



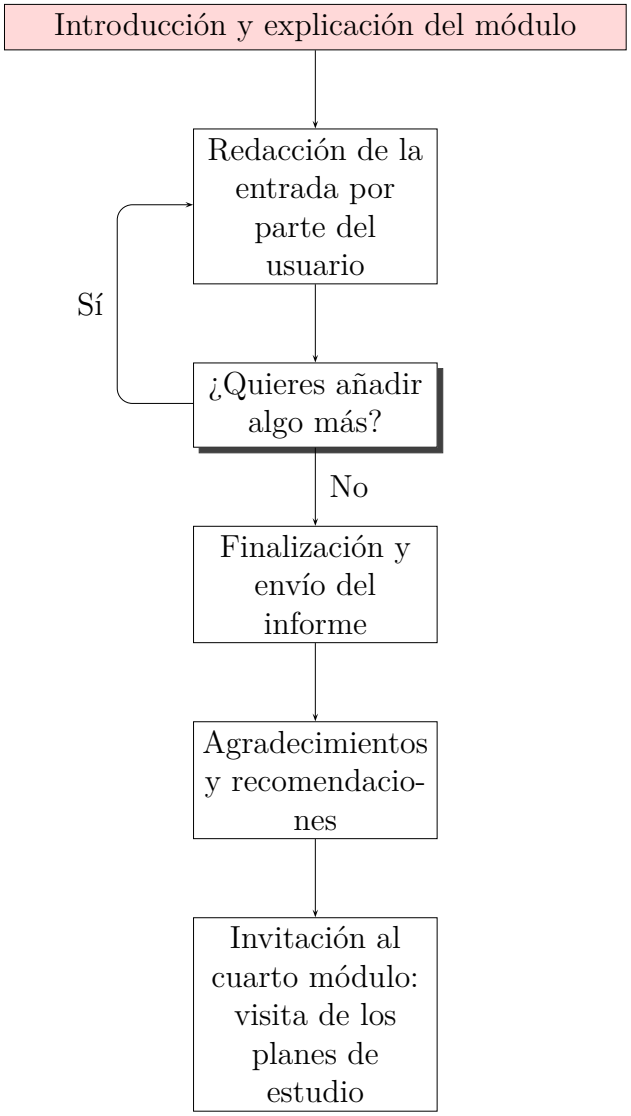
A.2. Módulo 1: Bienvenida y preguntas sobre sus intenciones académicas



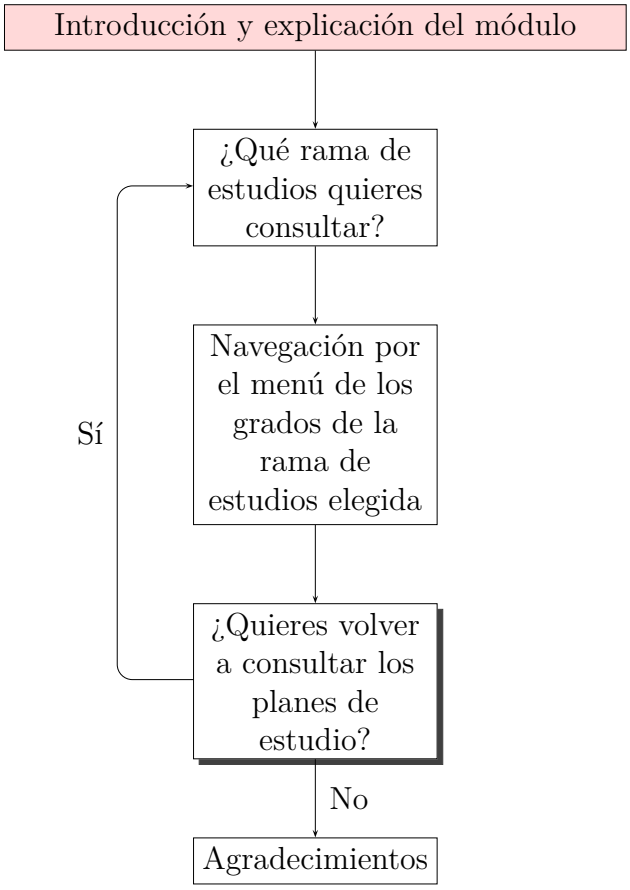
A.3. Módulo 2: Predicción de éxito académico en Ciencias y Humanidades



A.4. Módulo 3: Predicción de grados universitarios



A.5. Módulo 4: Planes de estudio



Apéndice B

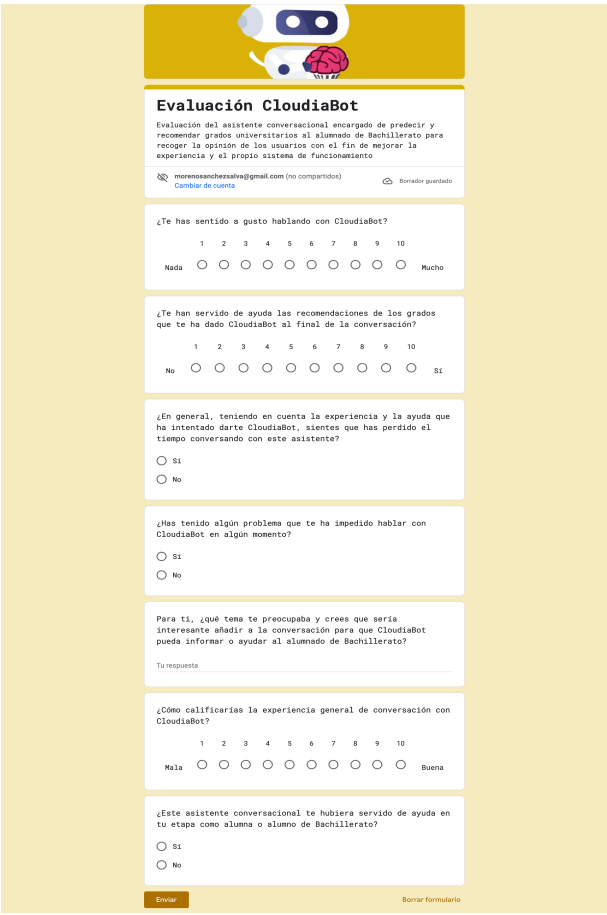
Árbol de directorios del proyecto

- cloudiaBot ..... GitHub: https://github.com/salvaMsanchez/CloudiaBot
  - [model.school_prediction_mat] ..... Predicción éxito Ciencias
    - assets
    - variables
      - variables.data-00000-of-00001
      - [variables.index]
    - [saved_model.pb]
    - [keras_metadata.pb]
  - [model.school_prediction_len] ..... Predicción éxito Humanidades
    - assets
    - variables
      - [variables.data-00000-of-00001]
      - [variables.index]
    - [saved_model.pb]
    - [keras_metadata.pb]
  - [net_dnn] ..... Predicción grados universitarios
    - [tflearn_logs]
      - TQ4IK6
        - events.out.tfevents.1658074919.989012d543cf
    - [model.tflearn_prediction.meta]
    - [model.tflearn_prediction.index]
    - checkpoint
    - [training_data]
    - [model.tflearn_prediction.data-00000-of-00001]
    - intenciones.json
    - grados.json
  - bot.py ..... Script de iniciación del programa
  - config.py ..... Script que alberga el token de Telegram
  - helper.py ..... Script de funciones generales

Apéndice C

Formularios de evaluación

C.1. Formulario entregado a participantes postuniversitarios



Evaluación CloudiaBot

Evaluación del asistente conversacional encargado de predecir y recomendar grados universitarios al alumnado de Bachillerato para recoger la opinión de los usuarios con el fin de mejorar la experiencia y el propio sistema de funcionamiento

morenosanchezsalva@gmail.com (no compartidos)
Cambiar de cuenta

Borrador guardado

¿Te has sentido a gusto hablando con CloudiaBot?

1 2 3 4 5 6 7 8 9 10

Nada ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mucho

¿Te han servido de ayuda las recomendaciones de los grados que te ha dado CloudiaBot al final de la conversación?

1 2 3 4 5 6 7 8 9 10

No ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Sí

¿En general, teniendo en cuenta la experiencia y la ayuda que ha intentado darte CloudiaBot, sientes que has perdido el tiempo conversando con este asistente?

○ Sí
○ No

¿Has tenido algún problema que te ha impedido hablar con CloudiaBot en algún momento?

○ Sí
○ No

Para tí, ¿qué tema te preocupaba y crees que sería interesante añadir a la conversación para que CloudiaBot pueda informar o ayudar al alumnado de Bachillerato?

Tu respuesta

¿Cómo calificarías la experiencia general de conversación con CloudiaBot?

1 2 3 4 5 6 7 8 9 10

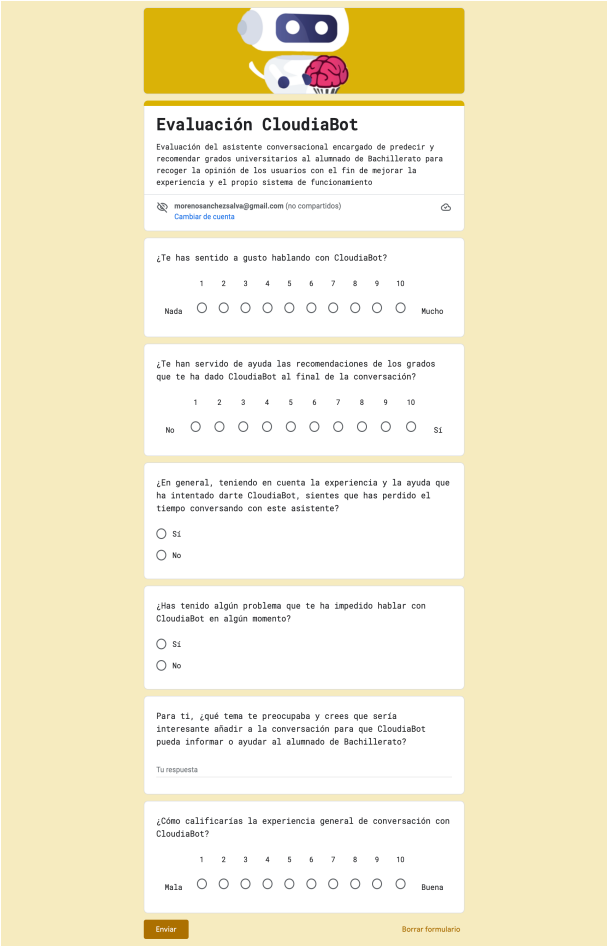
Mala ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Buena

¿Este asistente conversacional te hubiera servido de ayuda en tu etapa como alumna o alumno de Bachillerato?

○ Sí
○ No

Enviar Borrar formulario

C.2. Formulario entregado al alumnado de Bachillerato



Evaluación CloudiaBot

Evaluación del asistente conversacional encargado de predecir y recomendar grados universitarios al alumnado de Bachillerato para recoger la opinión de los usuarios con el fin de mejorar la experiencia y el propio sistema de funcionamiento

morenosanchezsalva@gmail.com (no compartidos)
Cambiar de cuenta

¿Te has sentido a gusto hablando con CloudiaBot?

1 2 3 4 5 6 7 8 9 10

Nada Mucho

¿Te han servido de ayuda las recomendaciones de los grados que te ha dado CloudiaBot al final de la conversación?

1 2 3 4 5 6 7 8 9 10

No Sí

¿En general, teniendo en cuenta la experiencia y la ayuda que ha intentado darte CloudiaBot, sientes que has perdido el tiempo conversando con este asistente?

Sí
No

¿Has tenido algún problema que te ha impedido hablar con CloudiaBot en algún momento?

Sí
No

Para ti, ¿qué tema te preocupaba y crees que sería interesante añadir a la conversación para que CloudiaBot pueda informar o ayudar al alumnado de Bachillerato?

Tu respuesta

¿Cómo calificarías la experiencia general de conversación con CloudiaBot?

1 2 3 4 5 6 7 8 9 10

Mala Buena

Enviar Borrar formulario