



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO Y CONTRUCCION DE UN CHATBOT BASADO EN MACHINE
LEARNING: CASO DE ESTUDIO ASISTENTE VIRTUAL PARA
MAESTRIAS UTMACH

GANAN FLORES RONALD FERNANDO
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO Y CONTRUCCION DE UN CHATBOT BASADO EN
MACHINE LEARNING: CASO DE ESTUDIO ASISTENTE
VIRTUAL PARA MAESTRIAS UTMACH

GANAN FLORES RONALD FERNANDO
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DISEÑO Y CONTRUCCION DE UN CHATBOT BASADO EN MACHINE
LEARNING: CASO DE ESTUDIO ASISTENTE VIRTUAL PARA MAESTRIAS
UTMACH

GANAN FLORES RONALD FERNANDO
INGENIERO DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 27 DE ABRIL DE 2021

MACHALA
2021

ronald

INFORME DE ORIGINALIDAD

7%

INDICE DE SIMILITUD

6%

FUENTES DE INTERNET

1%

PUBLICACIONES

1%

TRABAJOS DEL
ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

1%

★ www.dspace.uce.edu.ec

Fuente de Internet

Excluir citas

Activo

Excluir coincidencias < 15 words

Excluir bibliografía

Activo

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, GANAN FLORES RONALD FERNANDO, en calidad de autor del siguiente trabajo escrito titulado DISEÑO Y CONTRUCCION DE UN CHATBOT BASADO EN MACHINE LEARNING: CASO DE ESTUDIO ASISTENTE VIRTUAL PARA MAESTRIAS UTMACH, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 27 de abril de 2021



GANAN FLORES RONALD FERNANDO
0705860088

UNIVERSITAS
MAGISTRORUM
ET SCHOLARIUM

DEDICATORIA

Este trabajo está dedicado principalmente a Dios por ser el centro de mi vida, por darme la sabiduría y fortaleza para seguir siempre adelante con mis estudios.

A mis padres por ser mi soporte, por su apoyo incondicional, por su comprensión y por su confianza, la misma que ha sido valiosa a lo largo de este camino y sobre todo por permitir que nunca me rinda e incentivarme a terminar mi carrera universitaria.

Ganán Flores Ronald Fernando

AGRADECIMIENTO

El agradecimiento es un sentimiento que no se puede comparar con nada, por este motivo le estoy eternamente agradecido a Dios por ser el motor principal en mi vida y guiarme siempre.

A mis padres por ser el pilar fundamental de mi vida y acompañarme a lo largo de mi proceso de formación profesional.

A mis hermanas y amigos que estuvieron presentes, motivando en los buenos y malos momentos que se presentaron en mi vida universitaria ayudándome a cumplir con este objetivo propuesto.

Y por último a mi tutor el Ing. Wilmer Rivas Asanza por su valiosa guía y ayuda en la elaboración del presente proyecto de investigación.

Ganán Flores Ronald Fernando

RESUMEN

En la actualidad el desarrollo de la inteligencia artificial ha evolucionado de manera constante debido a los grandes avances tecnológicos que se han presentado gracias al apoyo de las grandes empresas tecnológicas, las cuales han desarrollado los diferentes tipos de aprendizajes que tiene la inteligencia artificial los cuales permiten desarrollar nuevos conocimientos o tomar decisiones sin la supervisión de un ser humano, todos estos avances ya forman parte de nuestras vidas y las podemos ver a diario en un teléfono inteligente, videojuegos, computadoras o asistentes personales como es el caso de Siri y en años posteriores se plantea que la inteligencia artificial podrá realizar las mismas tareas que un ser humano.

Para las empresas la atención al cliente es lo más importante, es por este principal motivo que la mayoría de empresas están optando por el uso de los chatbots, como son asistentes virtuales, agentes de ventas, atención al cliente, etc., los cuales están basados en inteligencia artificial y aprendizaje automático esto quiere decir que aprenden constantemente, su principal funcionalidad es que siempre está disponible y brinda una atención personalizada al usuario como si estuviera interactuando con un ser humano.

En el presente proyecto se desarrollará un asistente virtual que mostrara la información sobre las maestrías que oferta la Universidad Técnica de Machala, el cual estará basado en inteligencia artificial y machine learning, utilizando la plataforma Mindmeld que permite el análisis, comprensión y procesamiento del lenguaje natural, mediante la separación de palabras en las conversaciones, se creara una base del conocimiento la cual estará conformada por dominios e intenciones, en donde se almacenara la información para que aprenda automáticamente el asistente virtual y pueda cumplir con la petición que le hace el usuario de forma correcta, también permite la generación del lenguaje natural, la cual ayuda a convertir el lenguaje de maquina a texto para que el usuario pueda comprender y para el desarrollo de la página web que permitirá la interacción entre el usuario y el asistente virtual, se usaran Frameworks como son React y Flask mediante el lenguaje de programación Python.

Para poder cumplir con la finalidad del proyecto planteado se desarrolló una arquitectura la cual permitirá el procesamiento de la información etapa por etapa

estructurando la entrada de los datos basándose en el procesamiento del lenguaje natural y la salida de información la cual se basa en la generación el lenguaje y diálogos personalizados.

Una vez concluido con éxito el desarrollo y la ejecución del chatbot se procedió a realizar la evaluación de calidad para poder realizar esta evaluación nos basamos en un matriz de confusión, la cual estará conformada por los dominios, base del conocimiento e intenciones, al finalizar las pruebas de evaluación los resultados fueron muy prometedores llegando a cumplir un 95% de exactitud al momento de interactuar con el usuario en las tres pruebas que se realizó y la posibilidad de que llegue a un 100% es altísima debido a que tiene un aprendizaje supervisado excelente y mientras tenga mayor interacción con el usuario más alto será su grado de aprendizaje.

Palabras clave: Inteligencia artificial, Chatbot, Aprendizaje automático, Procesamiento de lenguaje natural, Mindmeld.

ABSTRACT

At present the development of artificial intelligence has evolved constantly due to the great technological advances that have been presented thanks to the support of large technology companies, which have developed the different types of learning that artificial intelligence has which allow develop new knowledge or make decisions without the supervision of a human being, all these advances are already part of our lives and we can see them daily on a smartphone, video games, computers or personal assistants as in the case of Siri and in later years they suggests that artificial intelligence can perform the same tasks as a human being.

For companies, customer service is the most important thing, it is for this main reason that most companies are opting for the use of chatbots, such as virtual assistants, sales agents, customer service, etc., which are Based on artificial intelligence and machine learning, this means that they are constantly learning, its main function is that it is always available and provides a personalized one to the user as if he were interacting with a human being.

In this project, a virtual assistant will be developed that will show the information about the master's degrees offered by the Technical University of Machala, which will be based on artificial intelligence and machine learning, using the Mindmeld platform that allows the analysis, understanding and processing of natural language Through the separation of words in the conversations, a knowledge base will be created which will be made up of domains and intentions, where the information is stored so that the virtual assistant learns automatically and can comply with the request made by the user in a correct, it also allows the generation of natural language, which helps to convert the machine language to text so that the user can understand and for the development of the web page that will allow the interaction between the user and the virtual assistant, Frameworks will be used such as React and Flask using the Python programming language.

In order to fulfill the purpose of the proposed project, an architecture was developed which will allow the processing of information stage by stage, structuring the input of maintenance data in natural language processing and the

output of information which is based on the generation custom language and dialogs.

Once the development and execution of the chatbot had been successfully concluded, the quality evaluation was carried out in order to carry out this evaluation, we are based on a confusion matrix, which will be made up of the domains, knowledge base and intentions, at the end of the evaluation tests the results were very promising reaching 95% accuracy when interacting with the user in the three tests that were carried out and the possibility of reaching 100% is very high due to the fact that it has excellent supervised learning and the greater the interaction with the user, the higher the degree of learning.

Keywords: Artificial Intelligence, Chatbot, Machine Learning, Natural Language Processing, Mindmeld.

Contenido

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN	3
ABSTRACT	5
INDICE DE FIGURAS	9
INDICE DE TABLAS	10
INTRODUCCION	11
1. CAPITULO I: DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS	13
1.1 Descripción del contexto y hechos de interés	13
1.2 Establecimientos de requerimientos	13
1.3 Justificación de requerimiento a satisfacer.....	14
2. CAPITULO II: DESARROLLO DEL PROTOTIPO	14
2.1 Definición del prototipo tecnológico	14
2.2 Fundamentación teórica del proyecto	14
2.2.1. Entorno del trabajo	14
2.1.1.1. Ubuntu	14
2.1.1.2. MindMeld.....	15
2.1.1.3. Librerías y Frameworks	19
SPACY	16
NLKT	16
REACT	16
FLASK	16
2.2.2. Inteligencia Artificial.....	17
2.2.2.1 Categorías de la Inteligencia Artificial.....	18
2.2.3. Machine Learning.....	19
2.2.3.1. Tipos de algoritmos de Machine Learning	20
Aprendizaje supervisado	20
Aprendizaje no supervisado.....	21
2.2.4. Procesamiento de lenguaje natural (PLN).....	21
2.2.4.1. Niveles del procesamiento de lenguaje natural.....	22
2.2.4.2. Arquitectura de un sistema de PLN.....	24
2.2.5. Chatbot.....	24
2.2.5.1. Historia del Chatbot	25

2.2.5.2. Características del Chatbot	28
2.2.5.3. Arquitectura de un Chatbot	29
2.2.5.4. Clasificación de un Chatbot.....	29
2.2.5.5. Funcionamiento del Chatbot.....	30
2.3 Objetivos del prototipo	31
2.3.1. Objetivo General	31
2.3.2. Objetivos específicos.....	31
2.4. Diseño del prototipo	32
2.4.1. Herramientas de desarrollo	32
2.4.2. Diseño arquitectónico del chatbot.....	33
2.4.3. Desarrollo del chatbot.....	34
Instalación de Visual Studio	34
Instalación de Python.....	34
Instalación de Docker	35
Instalación de Elasticsearch	37
Instalación de Mindmeld.....	38
Codificación del chatbot.....	38
2.5. Ejecución del prototipo	44
3. CAPITULO III: EVALUACIÓN DEL PROTOTIPO	50
3.1. Plan de evaluación	50
3.1.1. Métrica de clasificación.....	50
3.2. Resultados de la evaluación	51
Resultado prueba 1	51
Resultado prueba 2	53
Resultado prueba 2	54
3.3. Conclusiones.....	56
3.4. Recomendaciones	57
4. Bibliografía	58
5. Anexos.....	62

INDICE DE FIGURAS

Ilustración 1: Arquitectura de MindMeld.....	15
Ilustración 2: Etapas de la Inteligencia Artificial	17
Ilustración 3: Campos del aprendizaje automático y sus ejemplos	20
Ilustración 4: Aplicaciones del PLN	22
Ilustración 5: Arquitectura de un sistema de PLN	24
Ilustración 6: Arquitectura de ChatterBott.....	26
Ilustración 7: Flujo de proceso de ChatterBot.....	26
Ilustración 8: Chatbot Eliza	27
Ilustración 9: Cronología de los Chatbots.....	28
Ilustración 10: Arquitectura de un chatbot	29
Ilustración 11: Arquitectura del chatbot a desarrollar	33
Ilustración 12: Ejecución de Visual Studio.....	34
Ilustración 13: Instalación de paquetes para Python	35
Ilustración 14: Instalación de Python mediante consola de ubuntu.....	35
Ilustración 15: Instalación de paquetes para Docker	35
Ilustración 16: Instalación de Docker mediante consola de ubuntu	36
Ilustración 17: Ejecución de Docker	36
Ilustración 18: Instalación de llave pública para Elasticsearch	37
Ilustración 19: Instalación de Elasticsearch	37
Ilustración 20: Creación del entorno virtual para Mindmeld.....	38
Ilustración 21: Instalación de Mindmeld mediante terminal de visual studio	38
Ilustración 22: Librerías de mindmeld	38
Ilustración 23: Back End y Front End del chatbot	39
Ilustración 24: Base del conocimiento del chatbot	39
Ilustración 25: Dominios del chatbot.....	41
Ilustración 26: Intenciones del chatbot.....	41
Ilustración 27: Iniciar librería Elasticsearch	44
Ilustración 28: Ejecución de Docker	44
Ilustración 29: Ejecución del archivo webex_bot_server	45
Ilustración 30: Ejecución del chatbot.....	45
Ilustración 31: Ventana principal del chatbot	46
Ilustración 32: Saludo del chatbot.....	46
Ilustración 33: Interacción del usuario con el chatbot sobre las maestrías que oferta la UTMACH.....	47
Ilustración 34: Tipos de maestrías que ofrece la UTMACH.....	47
Ilustración 35: Inscripciones de la maestría	48
Ilustración 36: Modalidades de la maestría.....	48
Ilustración 37: Requisitos de la maestría.....	49
Ilustración 38: Malla curricular de la maestría	49
Ilustración 39: Búsqueda de maestría por área.....	62
Ilustración 40: Error 1 del Chatbot.....	62
Ilustración 41: Error N° 2 del chatbot	63

INDICE DE TABLAS

Tabla 1: Tipos de niveles del Lenguaje Natural	24
Tabla 2: Matriz de confusión.....	50
Tabla 3: Matriz de confusión N° 1.....	51
Tabla 4: Predicción del chatbot matriz N° 1	52
Tabla 5: Predicción del aprendizaje matriz N° 1.....	52
Tabla 6:Resultado final de la matriz N° 1.....	52
Tabla 7: Matriz de confusión N° 2.....	53
Tabla 8: Predicción del chatbot matriz N° 2.....	53
Tabla 9: Predicción del aprendizaje matriz N° 2.....	53
Tabla 10: Resultados finales matriz N° 2.....	54
Tabla 11: Matriz de confusión N° 3.....	54
Tabla 12: Predicción del chatbot matriz N° 3.....	55
Tabla 13: Predicción del aprendizaje matriz N° 3	55
Tabla 14: Resultados finales de la matriz N° 3.....	55

INTRODUCCION

En la actualidad la inteligencia artificial está revolucionando la forma de comunicación con el mundo, debido a su gran capacidad de aprendizaje a través de métodos como el Machine Learning o en el uso para la interconexión de dispositivos conocido como Internet of Things y también en el procesamiento del lenguaje natural [1], desde el año 1950 cuando el matemático Alan Turing descubre su origen y propone lo que ahora conocemos como el Test de Turing, el cual plantea la siguiente hipótesis, una computadora puede pensar por sí misma o una computadora puede simular que habla con un ser humano sin que él se dé cuenta y en la actualidad aún no se logra resolver [2].

El uso de los chatbots en la actualidad es muy común y se han utilizado en diferentes disciplinas desde atención al cliente, asesor de ventas o ayudantes que pueden mantener la conversación de cualquier tema en específico emulando un dialogo que un usuario mantendría en la vida real con otra persona.

En el presente proyecto de titulación se desarrollara un chatbot que será capaz de dar atención al cliente sin la necesidad de la intervención humana es decir iniciara una conversación siempre que el usuario responda además se expondrán conceptos claves para la creación de un chatbot, su modo de funcionamiento, las técnicas de procesamiento del lenguaje natural que emplea, así como también las ventajas que supone la implementación de chatbots en organizaciones o instituciones que brinden atención al cliente.

El presente trabajo de titulación se encuentra estructurado de la siguiente manera:

Capítulo 1: en este capítulo se identifica y plantea el problema a investigar, los requerimientos del prototipo, la justificación de la investigación y los respectivos solución planteada para el desarrollo de la investigación.

Capítulo 2: en este capítulo se presenta los antecedentes investigativos referentes a la investigación, también contiene la fundamentación teórica que sirve de soporte a la investigación para el desarrollo de la aplicación a implementar, así como los respectivos objetivos a llevar a cabo durante el desarrollo de la investigación.

Capítulo 3: se establecen los procesos de implementación, desarrollo y evaluación de los resultados de la aplicación implementada, así como también se detallan las conclusiones y recomendaciones según los objetivos descritos y acorde a la solución propuesta.

1. CAPITULO I: DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1 Descripción del contexto y hechos de interés

El gran desarrollo que ha tenido en los últimos años la inteligencia artificial ha permitido un crecimiento de los sistemas de procesamiento del lenguaje natural debido al gran auge que ha tenido el Machine Learning se pueden desarrollar sistemas que aprendan automáticamente [3], y esto ha impulsado el fortalecimiento y desarrollo de los chatbots que ya no son únicamente aplicaciones de mensajería instantánea si no que en la actualidad se crean para ayuda, soporte y venta a los usuarios.

La Universidad Técnica de Machala es una Institución pública de Educación Superior, se encuentra ubicada en la vía panamericana km 5 ½ vía a Pasaje, cuenta con muy buenas instalaciones y docentes fuertemente preparados, en la actualidad se encuentra ofertando maestrías y se vio la necesidad de tener un chatbot en una institución de educación superior que facilitaría la información al usuario sobre cada maestría que se esté ofertando sin la intervención humana.

Los chatbot han aumentado su popularidad en la actualidad encontrándose en diversas instituciones públicas o privadas, en Ecuador el chatbot ya es utilizado por instituciones de tercer nivel, entre ellas la Universidad Casa Grande [4] y la Universidad Técnica Particular de Loja [5].

1.2 Establecimientos de requerimientos

Mindmeld es una plataforma que nos permite la creación de agentes conversacionales o chatbots, utiliza algoritmos basado en aprendizaje supervisado y procesamiento del lenguaje natural.

Este proyecto está diseñado por etapas las cuales se detallarán a continuación:

Recolección de la información en este punto se realizará un estudio completo y exhaustivo sobre el desarrollo de un agente conversacional o chatbot así también como el aprendizaje automático que se empleará para su creación.

Se realizará una base de conocimiento la cual procesara el lenguaje natural mediante las diferentes librerías que se detallaran en el presente informe permitiendo que el chatbot pueda entender lo que le escribe el usuario y poder

responderle de forma inmediata y correcta de acuerdo a las necesidades del usuario.

1.3 Justificación de requerimiento a satisfacer

Desde la aparición de ELIZA en el año de 1966 siendo el primer chatbot de la historia capaz de simular una conversación real gracias a la inteligencia artificial hoy en día existen diferentes agentes conversacionales que crean ambientes de diálogos reales como son Cortana, Alexa, Siri que utilizan herramientas automatizadas que son capaces de dar respuestas inteligentes a las peticiones del usuario logrando satisfacer sus necesidades.

Este proyecto será desarrollado con la finalidad de brindar al usuario una atención al cliente que le permita informarse sobre las maestrías que se ofrecen en la UTMACH solventando las dudas de los usuarios mediante una interacción usuario-maquina en tiempo real creando una conversación agradable y amena.

2. CAPITULO II: DESARROLLO DEL PROTOTIPO

2.1 Definición del prototipo tecnológico

Al hablar de la inteligencia artificial, nos auto direccionamos al concepto de introducir ciertas herramientas informáticas que simulen la inteligencia humana, durante todo este proceso de introducir maquinas inteligentes ha permitido recrear modelos o alusiones para el cerebro humano.

En la actualidad el avance de esta rama tecnológica está en constante desarrollo e innovación, debido a que universidades, empresas y el sector privado están dedicándose a trabajar conjuntamente como es el caso de Google y Apple que han logrado implementar servicios que son basados en inteligencia artificial como son reconocimiento facial y asistentes inteligentes dentro de las diferentes empresas [6].

2.2 Fundamentación teórica del proyecto

2.2.1. Entorno del trabajo

2.1.1.1. Ubuntu

De acuerdo con [7], ubuntu es un sistema operativo basado en Linux que fue creado por Mark Shuttleworth en el año 2004 en su primera versión de llamo

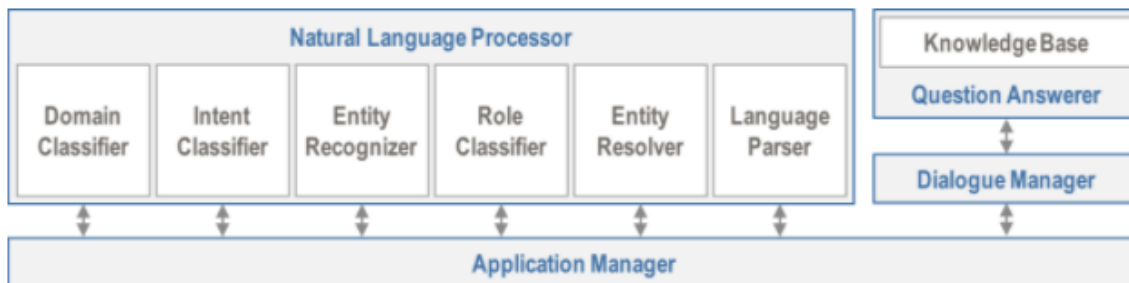
“Warty Warthong”, Es una distribución de Debian y sus principales características son [8]:

- Facilidad de manejo
- Actualizaciones frecuentes
- Libertad y uso de distribución
- Facilidad de instalación del sistema
- Búsqueda e instalación de programas robusta y fácil al basarse en paquetes.

2.1.1.2. MindMeld

Es una plataforma de la inteligencia artificial que permite crear aplicaciones conversacionales basado en aprendizaje automático a través de Python que tiene los algoritmos y utilidades necesarios para la finalidad de esta plataforma, optimizando la creación de asistentes virtuales que permitan una comprensión profunda de un caso en particular brindando una experiencia de conversación muy versátil [9].

Ilustración 1: Arquitectura de MindMeld



Fuente: basado en [1]

Las principales funciones de MindMeld son [1]:

- Procesamiento de Lenguaje Natural
- Dialogo versátil
- Crea base de conocimientos personalizados
- Respuesta avanzada a preguntas
- Análisis de datos a gran escala

Según [10], las principales plataformas que utilizan inteligencia artificial incluyen una API de predicción de la nube de Google como son MindMeld, Tensorflow,

Premonition y muchas más, permitiéndole a estas plataformas aprender a reconocer, identificar e interpretar bases de conjuntos y conocimientos de datos permitiendo crear una aplicación conversacional de vanguardia.

SPACY

Es una librería de Python utilizada para el desarrollo de aplicaciones con procesamiento de lenguaje natural mediante modelos pre entrenados con datos propios permitiendo analizar un texto y extraer la información del mismo, así como sus predicciones del modelo sobre su significado del contexto [11].

NLKT

NLKT Es una plataforma que funciona con datos del lenguaje humano para el análisis de texto fue desarrollada por Edward Loper y se lo implementa en Python como un kit de herramientas del procesamiento del lenguaje natural que permite controlar y gestionar el contenido a partir del reconocimiento del texto organizando las oraciones y dividiendo las palabras para su extracción semántica y de significado [12].

REACT

React es una librería que pertenece a JavaScript y se centra principalmente en el desarrollo de interfaces de forma sencilla para aplicaciones web, fue desarrollada por Jordan Walke, en sus inicios estaba a cargo de Facebook, pero ahora es de código abierto [13].

FLASK

Flask es un “micro” Framework basado en WSGI y Jinja2 fue desarrollado en Python y permite el desarrollo de páginas web de forma ágil y sencilla manteniendo el núcleo simple pero extensible [14].

ELASTICSEARCH

Es un conjunto de herramientas gratuitas y abiertas para el análisis distribuido y la visualización de todos los tipos de datos, incluidos estructurados, no estructurados, textuales, numéricos y geoespaciales además puede almacenar e indexar de manera efectiva los datos con alta confidencialidad y eficiencia [15].

De acuerdo con [16], Elasticsearch nos proporciona una api de java la cual nos permite ordenar una serie de datos por: números dinámicos, pares (clave – valor), además nos da un motor de búsqueda parecido al de Google el cual permite las búsquedas sean rápidas y precisas así como la recuperación de datos.

2.2.2. Inteligencia Artificial

Intentar definir la inteligencia artificial aun es incierto debido a que el interés hacia ella va crece y crece año tras año porque el desarrollo tecnológico no se detiene, tratando siempre de que haga algo nuevo ganando efectividad en su proceso de aprendizaje para demostrar que tienen la capacidad de adquirir conocimientos humanos e incluso nos puedan superar [17].

Podemos identificar dos métodos que nos ayudara al aprendizaje de este artefacto, aprendizaje con datos supervisados e interacciones correctas e incorrectas entre el humano y la máquina, la segunda es el aprendizaje automático, que su principio es dar experiencia a la máquina para ser capaz de auto educarse de forma independiente, por medio de una serie de normas lógicas iniciales [18].

Ilustración 2: Etapas de la Inteligencia Artificial



Fuente: Autor

Por lo tanto, cabe recalcar que este campo de investigación se encuentra aún en desarrollo, pese a que su historia cuenta el avance en diferentes etapas; sin embargo, varios objetivos aún son considerados inalcanzables motivando a realizar mejores investigaciones, con mayor disciplina y que posean riqueza en su dominio aplicativo y fundamenten la teoría.

2.2.2.1 Categorías de la Inteligencia Artificial

De acuerdo con [19] y [20] básicamente existen 4 enfoques como inicio para el desarrollo de la inteligencia artificial a saber y son:

- **Pensar Humanamente:** se basa principalmente en enfocarse en los procesos mentales y neuronales del ser humano para después poder implementarlos con un programa mediante una computadora, con la finalidad de que la inteligencia artificial no solo resuelva problemas o tome decisiones si no que intente razonar como un humano, para poder cumplir con esta funcionalidad primero tenemos que entender cómo funciona nuestra mente, pero a la actualidad no se tiene la total certeza de cómo funciona los procesos mentales del ser humano y este motivo lo convierte en la principal debilidad del acercamiento a Pensar Humanamente, ya que no podemos programar algo que no conocemos desde la raíz.
- **Actuar Humanamente:** este enfoque se basa en programar una computadora para que pueda actuar como un ser humano mediante algoritmos óptimos los cuales emulan los procesos mentales de un ser humano, tal como lo realizó Alan Turing en la Prueba de Turing en el año de 1950 que fue calificado por expertos en la materia como realista ya que simular una conducta inteligente y hacer que parezca como tal no era fácil.
- **Pensar Relacionalmente:** sigue las leyes del pensamiento lógico y racional utilizando las leyes de la lógica propuestas desde Aristóteles con sus silogismos: “Todos los hombres son mortales. Sócrates es un hombre, luego Sócrates es mortal”.

A través del tiempo los científicos procedieron a la creación de base de datos con relaciones lógicas que permitirían modelar todas las iteración posibles pero se presentaron 2 obstáculos que aún no han sido resueltos: el primero de ellos es que no se puede tener la completa certeza de todas las interrelaciones lógicas del mundo, y el segundo obstáculo es que existe una gran diferencia entre resolver un problema “en principio” y resolverlo de manera “práctica o real” que al final de cuentas es lo que buscamos.

- **Actuar Relacionalmente:** en la actualidad este enfoque es el que más aproximado está en la realidad y es posible de ser implementado a través de las computadoras.

Su mayor exponente es un agente que “es algo que hace algo” y se define como un programa que funciona automáticamente basado en los estímulos que percibe en su entorno, se adapta al cambio, tiene tiempos prolongados y persigue sus objetivos.

Un agente puede actuar de 2 maneras: la primera es que puede ser a través de leyes lógicas persiguiendo una respuesta correcta y la segunda es que si no tiene un resultado 100% seguro buscara el mejor resultado mediante un análisis de probabilidades.

2.1.1.3. Librerías y Frameworks

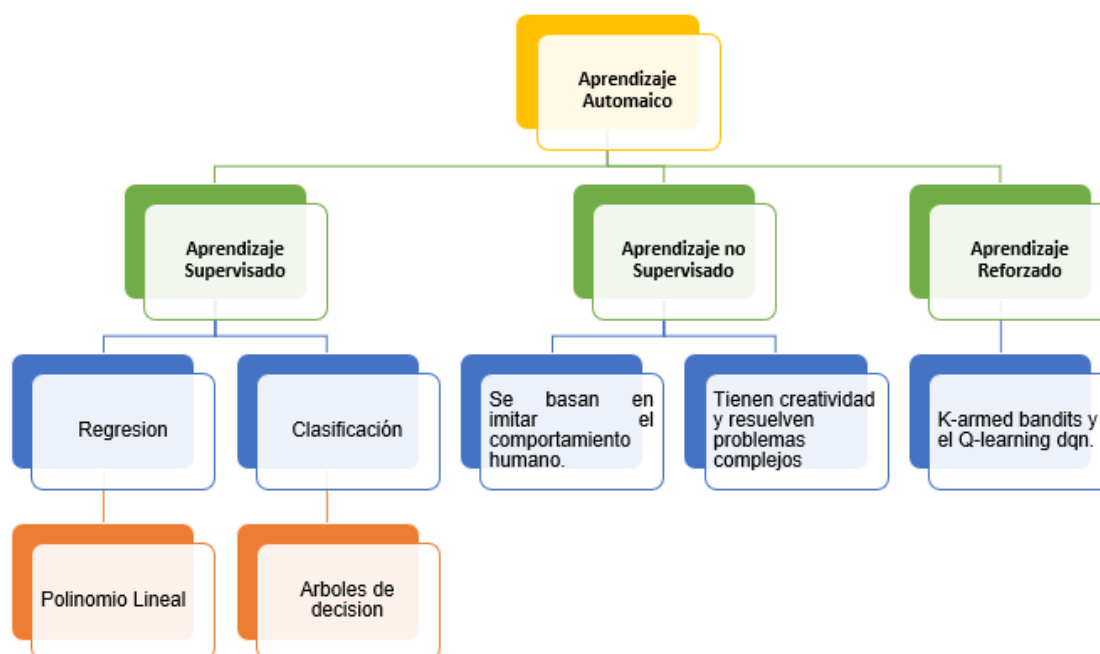
2.2.3. Machine Learning

De acuerdo con [20] el aprendizaje automático o también conocido como Machine Learning, es un campo de la informática que evolucionó a partir del estudio del reconocimiento de patrones y la teoría del aprendizaje computacional mediante la inteligencia artificial sobre un conjunto de datos a través de procedimientos que operan mediante la construcción de un modelo a partir de entradas de ejemplo para realizar predicciones o elecciones basadas en datos en lugar.

Se basa en emular la característica humana como dice [21], basándose en experiencias las cuales se desarrollaran mediante entrenamientos (conjunto de datos) que le permitirán predecir con gran precisión un estado para cada situación que se presente.

Desde el punto de vista de [22], el aprendizaje automático está basado en algoritmos que fueron desarrollados en los últimos 30 años permitiendo tener grandes avances tecnológicos sobre la predicción de datos, reconocimiento de imágenes, traducción de textos y muchos más debido al gran procesamiento de datos en CPU muy veloces.

Ilustración 3: Campos del aprendizaje automático y sus ejemplos



Fuente: Autor

2.2.3.1. Tipos de algoritmos de Machine Learning

El machine Learning tiene una amplia gama de aplicaciones las cuales están basadas en el aprendizaje de máquina el cual se establece en dos pilares principales llamados aprendizaje supervisado y aprendizaje no supervisado, aunque también se encuentra el aprendizaje reforzado como se muestra en la ilustración 3.

Aprendizaje supervisado

El aprendizaje supervisado su principal objetivo es aprender mediante un algoritmo donde se le dan los datos desde la entrada hasta la salida cuyos valores son correctos, mientras mayor sea el conjunto de datos el algoritmo más aprenderá sobre el tema, una vez que finaliza su entrenamiento se le brindan nuevos datos y en base a su experiencia pasada predecirá el resultado [23], dependiendo del tipo de predicción se pueden emplear algoritmos regresivos (busca la aprobación más cercana a un valor continuo) o de clasificación (permite identificar dos o más clases mediante una serie de variables) pero para que estos procesos sean exitosos tiene que tener tiquetes fundamentales para el procesamiento de datos [24].

Aprendizaje no supervisado

El aprendizaje no supervisado se diferencia del supervisado ya que no tiene un conocimiento a priori debido a que se entrena mediante un gran conjunto de datos que no están etiquetados ni relacionados entre sí, la idea principal de este aprendizaje es que pueda encontrar por si solo patrones para luego relacionarlos y agruparlos [25].

2.2.4. Procesamiento de lenguaje natural (PLN)

De acuerdo con [26], el procesamiento del lenguaje natural (PNL) dentro de la inteligencia artificial es el área del conocimiento y está basada en técnicas computacionales las cuales permiten investigar las formas de relacionar maquinas con personas mediante un análisis automático y la representación del lenguaje humano, para poder tener un enfoque completo del texto el PLN debe realizar procesos de recuperación de la información, desambiguación de la palabra y similitud del texto [27].

Según [28], la principalidad funcionalidad del lenguaje natural es facilitar la comunicación entre el humano y la máquina, definiendo un conjunto de reglas o símbolos para poder comprender y generar una respuesta para el usuario si la necesidad de que el usuario sepa el lenguaje máquina.

Según [29] y [30] mediante el uso del PLN se puede realizar aplicaciones como son:

- **Traducción Automática de textos (TA)**

Es la tarea de transformar una oración o frase de un lenguaje a un nuevo lenguaje es una tarea complicada ya que los lenguajes tienen estructuras morfológicas complejas y para esto se necesita utilizar redes neuronales. es un área de la lingüística computacional que investiga el uso de un sistema capaz de traducir texto o habla de un lenguaje natural a otro.

- **Sistemas conversacionales**

La iteración entre máquinas y humanos a través de la voz es una de las aplicaciones más exitosas y se usa en la mayoría de artefactos virtuales ya que llevan un pequeño ordenador con el que se pueden comunicar.

- **Recuperación y extracción de la información**

La extracción de la información se basa principalmente en clasificar, resumir y relacionar como pasa en los buscadores de internet que fueron la primera aplicación masiva del lenguaje natural en el mundo del tic.

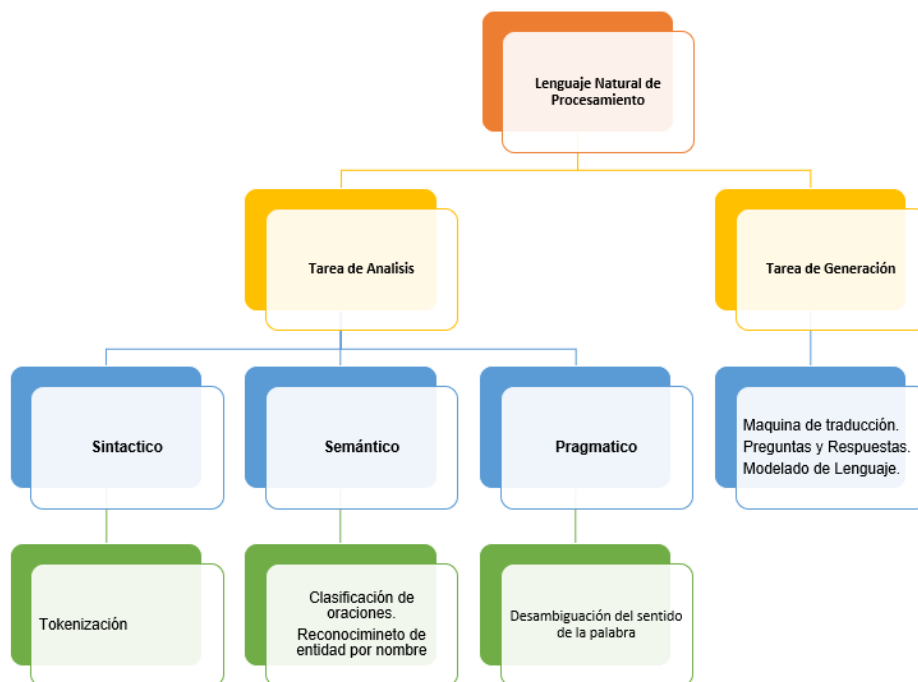
- **Etiquetado morfológico, sintáctico y semántico**

Mediante este etiquetado se procesa el contenido que usan los etiquetadores que analizan los mensajes de diferentes niveles, estos sirven para anotar las formas verbales o la asignación de las etiquetas de sujeto y objeto directo en la oración.

- **Respuestas automáticas y preguntas**

Su principal función es producir una respuesta a partir de una pregunta y son muy utilizadas en el área comercial ya que sirven como sistemas para chatbots o asistentes como Google Assistant o Apple Siri.

Ilustración 4: Aplicaciones del PLN



Fuente: basada en [29] y [30]

2.2.4.1. Niveles del procesamiento de lenguaje natural

Un sistema PLN tiene diferentes niveles para análisis que se encuentran interrelacionados y son muy importantes para la correcta interpretación de la estructura del lenguaje. A continuación, se detalla cada uno de los niveles según [31]:

Nivel Fonológico: se basa en los sonidos del lenguaje humano, es decir analiza como las palabras se relacionan con los sonidos que representan.

Nivel Morfológico: este nivel se enfoca en la composición y derivación de palabras a través de secuencias de caracteres las cuales se transformarán en secuencia de morfemas, permitiéndonos saber el significado de cada morfema, además permite clasificar las palabras como son: verbos, adverbios, sustantivos, entre otros.

Nivel Léxico: identifica el vocabulario de un idioma de forma individual utilizando técnicas de asignación de etiquetas a cada palabra del texto para categorizar la palabra y así poder asignarle su función en el texto.

Nivel Sintáctico: se enfoca en ordenar y relacionar las palabras dentro de una oración y cuál es la función que cumplen mediante un análisis de la estructura gramatical, a través de un parser que tiene como finalidad obtener la estructura y representarla en forma de árbol o red mediante las redes de transición.

Nivel Semántico: se basa en el significado de las palabras y en como estos significados se unen para dar significado a una oración y al significado independiente del contexto mediante técnicas de desambiguación semántica para comprender las palabras que tengan diversos significados.

Nivel de Discurso o Contextual: se basa en interpretar el texto de forma que comprenda no solo las palabras si no un conjunto de oraciones interrelacionadas en función del contexto, permitiéndonos identificar el significado de las palabras polisémicas, utiliza el reconocimiento de la estructura del texto y la resolución anafórica.

Nivel Pragmático: analiza como las oraciones se usan en distintas situaciones, tomando distintos significados dependiendo de la situación, utiliza módulos de inferencia y bases de conocimientos como herramientas de interpretación de las intenciones y objetivos de un texto.

Tabla 1: Tipos de niveles del Lenguaje Natural

Niveles del PLN	Ámbito de Actuación	Herramientas que utiliza
Fonético	Sonidos	Corpus de aprendizaje Modelos acústicos
Léxico	Formas	Etiquetado POS Léxico computacional
Morfológico	Formas	Etiquetado POS Léxico computacional
Sintáctico	Estructuras	Bases de datos sintácticas Arboles de decisión
Semántico	Significados	Bases de datos semánticas
Discurso	Comunicación	Bases de datos semánticas Ontologías
Pragmático	Comunicación	Bases de datos semánticas Ontologías

Fuente: Autor

2.2.4.2. Arquitectura de un sistema de PLN

A través de los diferentes niveles de lenguaje natural la computadora está capacitada para interpretar y analizar las oraciones que le sean proporcionadas, en el siguiente grafico se observa el funcionamiento de un sistema PLN:

Ilustración 5: Arquitectura de un sistema de PLN



Fuente: basada en [19]

2.2.5. Chatbot

De acuerdo con [31], un Chatbot es un software basado en mensajes estructurados para mantener una conversación mediante el lenguaje natural con un humano, tienen la capacidad de diferenciar entre palabras y emoticones para

poder cumplir esta función el software ha sido especialmente diseñado, entrenado y programado para que interprete la conversación.

Como expresa [32] un Chatbot no solo se encuentra ligado a mensajes de texto debido a que usa técnicas de procesamiento de lenguaje natural, los chatbots son aplicables en muchos campos como son: educación, viajes, web, servicio al cliente y compras ya que son capaces de tener conversaciones con razonamientos basados en diálogos con el ser humano.

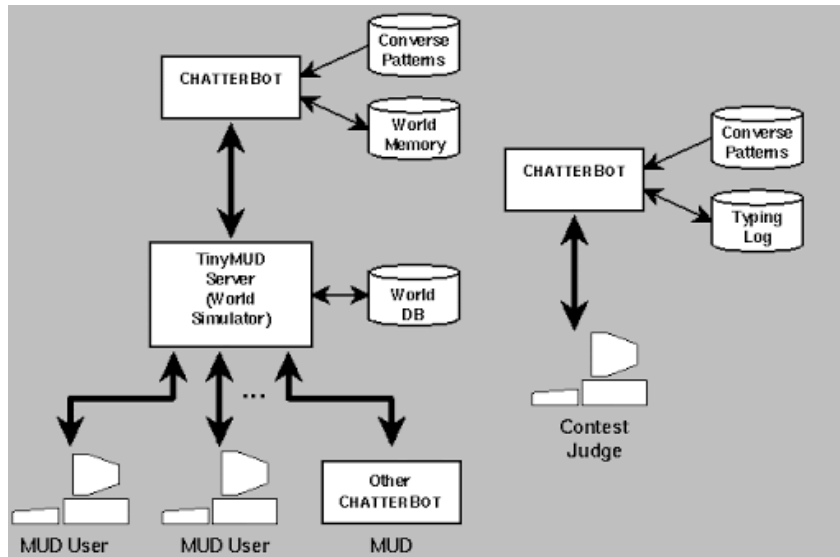
Según [33], un Chatbot a simple vista parece una sistema automatizado de mensajería normal que dialoga con el usuario pero esta aplicación se encuentran conformada por 3 partes como son: la capa de aplicación, la base de datos y una Api (aplicación de interfaz de programación) que trabajan en segundo plano que llevan los registros de las conversaciones y ayudan a comprender el dialogo con el usuario, un chatbot se lo puede utilizar en cualquier plataforma web, móvil y aplicación de escritorio, funciona y responde a todas las personas que están interactuando con el no importa la hora ni el día [34].

2.2.5.1. Historia del Chatbot

La palabra "Chatbot" está formada por los términos chatter proveniente del inglés que significa "charla" y bot que es la abreviación de "robot" esto quiere decir que es un programa informático que simula una conversación mediante un modelo de la inteligencia artificial que permite la interacción entre un humano y una computadora [35].

El termino chatbot tiene más de 60 años desde la primera vez que se lo planteó tal como lo propuso Alan Turing en su Test o también conocido "como el juego de la imitación" el cual se basaba en que una maquina podía ser indistinguible de un ser humano a través de una conversación textual [36], para este procedimiento se utilizó el procesamiento del lenguaje natural que permitió la evolución de los agentes conversacionales o chatbots tomando en cuenta este procedimiento se creó el primer chatbot en el año de 1966 denominado Eliza [37], y fue desarrollado por Joseph Weizenbaum mediante lenguaje C y se basaba en patrones los cuales simulaban al comportamiento humano, en el año de 1989 se basaron en Eliza para la creación de ChatterBot también llamado TYNMUD [38].

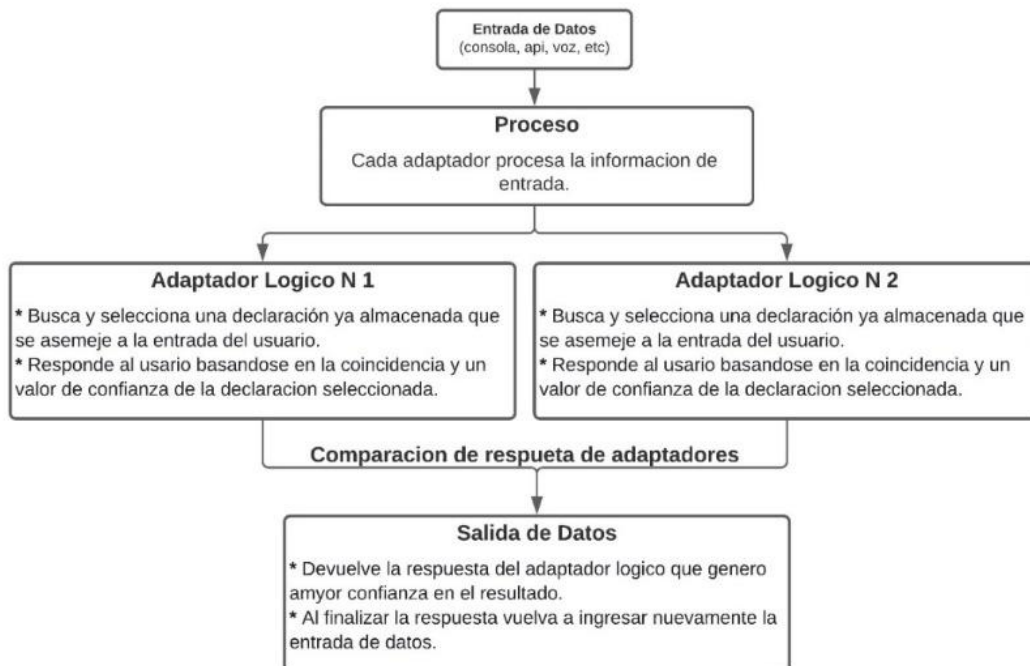
Ilustración 6: Arquitectura de ChatterBott



Fuente: [38]

En la actualidad ChatterBot es una biblioteca de Python diseñada para la generación de respuestas automáticas mediante chatbots en la interacción usuario-maquina, para el aprendizaje automático utiliza algoritmos que le permiten desarrollar diferentes tipos de respuestas, permitiendo a los desarrolladores la creación de chatbots para los diferentes escenarios en el mundo laboral [38].

Ilustración 7: Flujo de proceso de ChatterBot



Fuente: Autor

De acuerdo con [39] el Dr. Richard Wallace en el año de 1995 crea la primera versión de A.L.I.C.E (Entidad informática de Internet lingüística artificial), el cual estaba basado en lógica matemática mediante SETL pero no tuvo mucho éxito, debido a este gran fracaso surge una nueva tecnología para el desarrollo de los chatbots como es AIML(Lenguaje de Marcado de Inteligencia Artificial) que tiene gran usabilidad, facilidad de aprendizaje y ejecución para el desarrollo de los chatbots [40], y para la última versión de A.L.I.C.E se utiliza como estándar AIML orientado a XML(Lenguaje de marcado extensible) y con esta versión obtuvo el premio Loebner Prize en el año 2004 pero debido a las limitaciones que aún posee AIML no pudo pasar la prueba de Turing [41].

Ilustración 8: Chatbot Eliza

A screenshot of a chatbot conversation. The text is displayed in a white box with a thin black border. The conversation consists of several lines of text, alternating between 'Human:' and 'Eliza:'. At the bottom, there is a grey input field with the label 'Chat:' and a white cursor. The text of the conversation is as follows:

Human: Men are all alike.
Eliza: In what way?
Human: They're always bugging us about something specific or other.
Eliza: Can you think of a specific example?
Human: Well, my boyfriend made me come here.
Eliza: Your boyfriend made you come here?
Human: He says I'm depressed much of the time.
Eliza: I am sorry to hear you are depressed.
Human: It's true. I'm unhappy.
Eliza: Do you think coming here will help you not to be unhappy?
Human: I need some help, that much seems certain.
Eliza: What would it mean to you if you got some help?
Human: Perhaps I could learn to get along with my mother.
Eliza: Tell me more about your family.
...

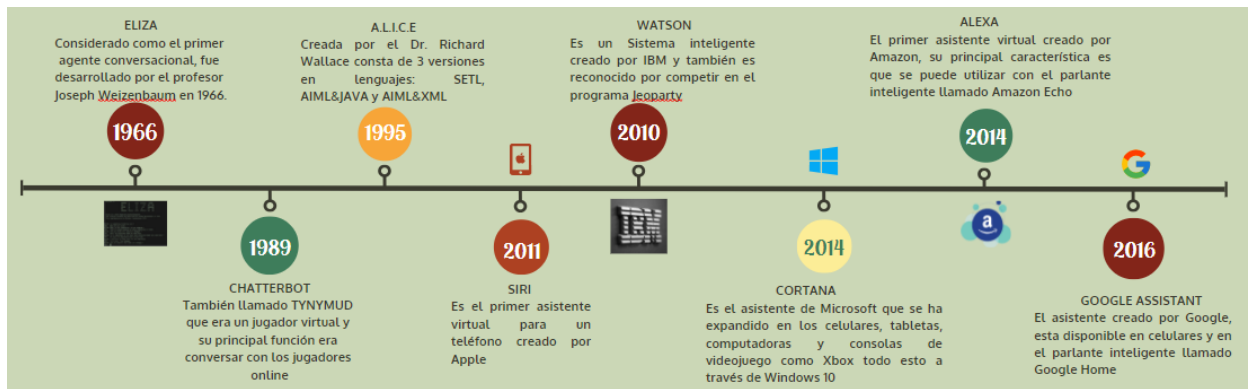
Fuente: [41]

Con el avance de la inteligencia artificial se impulsó al desarrollo de nuevas tecnologías para la creación de chatbots es así que en el año 2014 se desarrolló un chatbot llamado Eugene Gootsman que se hacía pasar por un niño de 13 años logrando aprobar el test de Turing por primera vez en la historia convenciendo al 33% de los jurados de que era un ser humano [42].

En la actualidad las grandes empresas han desarrollado sus propios chatbots que tienen un gran diseño y funcionalidad como son: Apple (Siri), Google (Google Assistant), Microsoft (Cortana), IBM (Watson), Amazon (Alexa) anuncian

innovaciones para el desarrollo de chatbots que son capaces de interactuar con el usuario a partir de entradas de texto y voz [43].

Ilustración 9: Cronología de los Chatbots



Fuente: Autor

2.2.5.2. Características del Chatbot

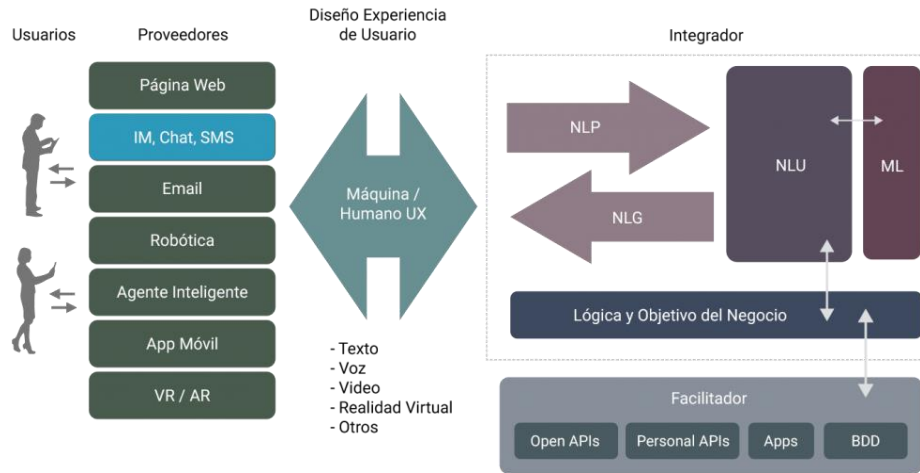
Según [44] los chatbots tienen características que le permiten realizar diversas funciones entre las más importantes tenemos:

- **Accesibilidad:** está caracterizada por su fácil manejo y ayuda en la información para responder adecuadamente detectando las intenciones en un tiempo oportuno para su respuesta además debe estar disponible siempre y cuando el usuario lo requiera.
- **Funcionalidad:** tiene que ser muy preciso en la interpretación de la entrada lingüística mediante el procesamiento del lenguaje natural para que la salida sea correcta.
- **Rendimiento y Robustez:** al hablar de estas características nos referimos a que un chatbot debe ser capaz de mantener la conversación hasta el final del proceso y evitar que la tarea no se frustre en su proceso.
- **Humanidad:** a través de esta característica nos referimos a que el chatbot deber tener la capacidad de interactuar de forma convincente y mantener una interacción con el usuario de forma natural siempre tomando en cuenta el contexto de la conversación.
- **Ética y comportamiento:** es fundamental que el chatbot deba proteger y respetar la privacidad del usuario, así como su cultura y principios éticos.

2.2.5.3. Arquitectura de un Chatbot

La arquitectura de un chatbot está conformada por proveedores, diseño de experiencia de usuario y el integrador como se observa en la ilustración 10.

Ilustración 10: Arquitectura de un chatbot



Fuente: [45]

De acuerdo con [45] la funcionalidad de cada capa de la arquitectura del chatbot sería la siguiente:

Proveedores: es la interfaz que permite al usuario conectarse con el chatbot.

Diseño de experiencia de usuario: esta capa es la que se encuentra en medio y permite la comunicación entre el usuario y está conformada por 2 capas que son fundamentales y dependen entre si ya que sin una de ellas no sería posible tener una buena interacción entre el usuario y el chatbot.

Integrador: es la pieza fundamental del chatbot debido a que contiene todas las herramientas del procesamiento de lenguaje natural, el aprendizaje automático y es la parte de la inteligencia artificial del chatbot.

2.2.5.4. Clasificación de un Chatbot

La clasificación de los chatbots puede variar de acuerdo a como se desarrollen a continuación, detallaremos las tecnologías y el campo de aplicación que son utilizados en el desarrollo de los chatbots.

2.2.5.4.1. Chatbots según su tecnología:

De acuerdo con [46] los chatbots según su tecnología son:

Simples: son chatbots sencillos y su funcionamiento se basa en la coincidencia de patrones básicos.

Complejos: estos chatbots utilizan técnicas de inteligencia artificial para poder procesar la información permitiendo tener una conversación coherente con el usuario.

2.2.5.4.2. Chatbots según su campo de aplicación:

Según [35], los chatbots según su tecnología son:

Sociales: están enfocados a tener una buena relación con el usuario y hacer que la conversación fluya de forma normal como si estuviera conversando con otra persona son muy utilizados en redes sociales y atención al cliente.

De Utilidades: tiene un objetivo en específico y su éxito se mide por el cumplimiento del objetivo.

Asistentes: están incluidos dentro de los sistemas operativos como es el caso de Cortana o Google Assistant tienen una gran personalidad al momento de interactuar y pueden cumplir las peticiones de los usuarios.

2.2.5.5. Funcionamiento del Chatbot

El funcionamiento del chatbot se inicia cuando recibe una petición del usuario a través del aplicativo que utiliza de entrada como puede ser la voz o el texto, y la respuesta que recibirá el usuario sería una predeterminada que ya aprendió con las interacciones cabe recalcar que puede generar respuesta más inteligentes conforme se lo utilice [47].

2.3 Objetivos del prototipo

2.3.1. Objetivo General

- Desarrollar un chatbot basado en inteligencia artificial con técnicas de machine Learning para la gestión de asistente virtual en maestrías de la UTMACH.

2.3.2. Objetivos específicos

- Investigar las tecnologías para el procesamiento del lenguaje natural (NLP) con machine Learning utilizadas en la creación de chatbot.
- Analizar las características y funcionamientos de los chatbots.
- Diseñar una arquitectura para el desarrollo del chatbot.
- Desarrollar una aplicación web para la comunicación entre el usuario y el chatbot.

2.4. Diseño del prototipo

2.4.1. Herramientas de desarrollo

Python

Es un lenguaje de programación orientado a objetos (POO), está conformado por instrucciones las cuales se deben convertir en operaciones ejecutables para ser interpretadas [48]. Con Python se puede diseñar, configurar y ejecutar scripts, mediante herramientas y librerías que soporta python además es un lenguaje multiplataforma que se puede compilar en los distintos sistemas operativos existentes (Linux, Unix, Windows, Mac, OS) [49].

Docker

Docker es una herramienta diseñada para facilitar la creación, implementación y ejecución de aplicaciones a través del uso de contenedores los cuales permiten empaquetar la aplicación con todas las partes que la componen, como bibliotecas y otras dependencias enviando todo a un solo paquete [50], esto garantiza que el software siempre se ejecutará igual, independientemente de su entorno, en la actualidad Docker está siendo muy utilizado debido a que reduce el uso de los recursos del sistema mejorando el procesamiento y memoria del equipo.

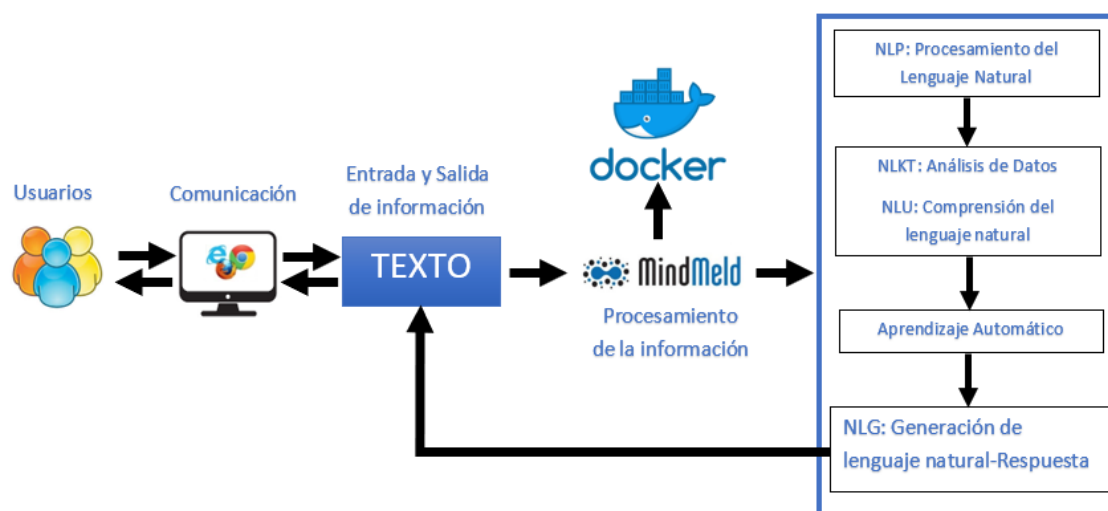
Visual Studio

Es un entorno de desarrollo integrado para sistema diferentes sistemas operativos, es compatible con los diferentes lenguajes de programación como son: Python, Visual C++, C#, Java, PHP y muchos más, permite crear aplicaciones móviles, web y muchas más [51].

2.4.2. Diseño arquitectónico del chatbot

Para el diseño del chatbot creamos su propia arquitectura como se muestra a continuación.

Ilustración 11: Arquitectura del chatbot a desarrollar



Fuente: Autor

Como se muestra en la ilustración 11 nuestra arquitectura está compuesta por:

Usuario: es la persona encargada de interactuar con el chatbot o agente conversacional.

Comunicación: es el medio en el cual el usuario podrá visualizar el chatbot.

Entrada de la información: el usuario ingresa solo texto para la comunicación con el chatbot debido a que el chatbot solo reconoce texto.

Procesamiento de la información: para el procesamiento de la información se utilizó mindmeld el cual permitirá el procesamiento del lenguaje natural, el análisis de la información, la comprensión de la información (NLU) que permite estructurar las entradas para que la maquina pueda entender por último la generación del lenguaje natural (NLG) que es donde las maquinas escriben el lenguaje y el proceso del NLG los convierte en datos estructurados en texto para el usuario.

Docker: es el complemento de mindmeld le permite iniciar todos sus servicios para el aprendizaje del chatbot.

Salida de la información: una vez que los datos son estructurados se procesan para que a su salida se presente un mensaje de texto para el usuario.

2.4.3. Desarrollo del chatbot

Para el desarrollo del chatbot se debe realizar lo siguiente:

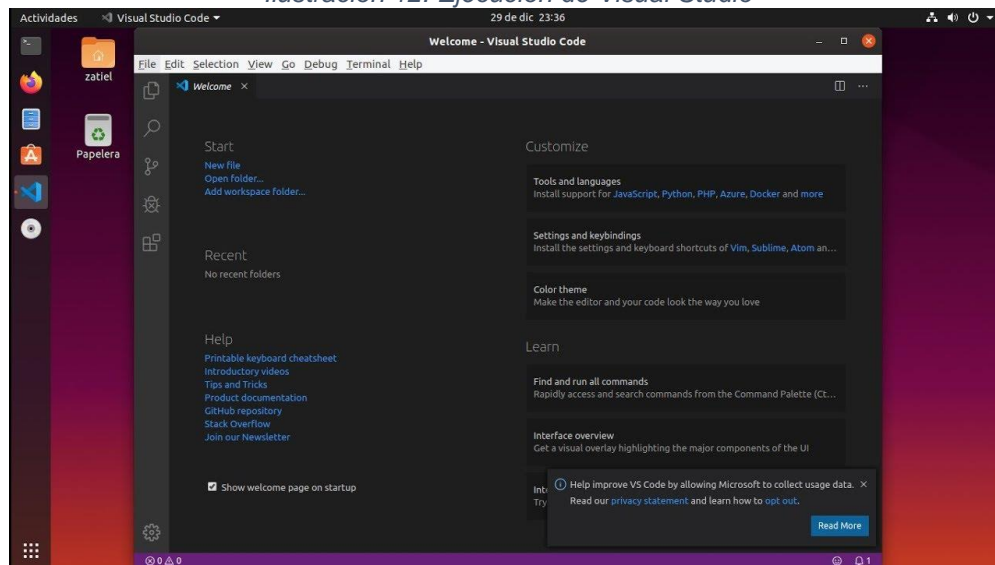
Instalación de herramientas

A continuación, se detallará la instalación de cada uno de los requisitos necesarios para la creación de la aplicación chatbot.

Instalación de Visual Studio

- A. Descargamos el instalador desde la página de visual estudio, <https://code.visualstudio.com/Download>.
- B. Descomprimos el archivo con la siguiente línea de comando: **sudo unzip VSCode-linux-x.zip -d /opt/vscode**.
- C. Después de descomprimir el programa procedemos a instalarlo con la siguiente línea de comando: **sudo apt install code**.
- D. Una vez instalado procedemos a la ejecución del programa.

Ilustración 12: Ejecución de Visual Studio

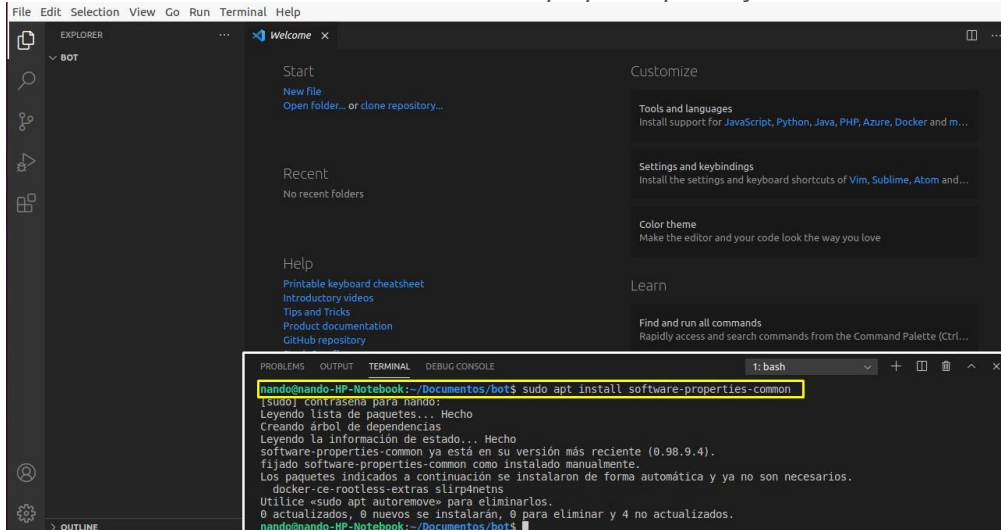


Fuente: Autor

Instalación de Python

Para realizar la instalación de python ingresamos a nuestro entorno de trabajo Visual Studio y a continuación ejecutamos **sudo apt install software-properties-common** mediante esta línea podremos administrar los paquetes que provienen de proveedores de software independiente.

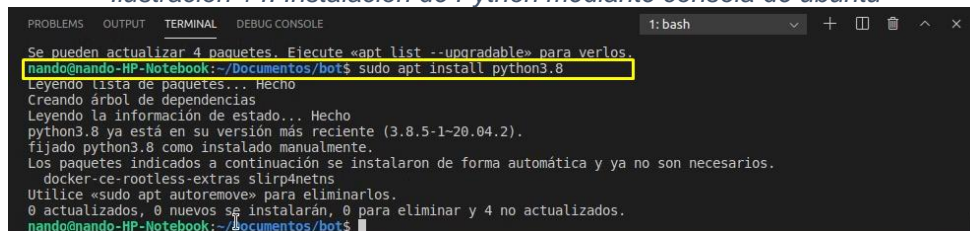
Ilustración 13: Instalación de paquetes para Python



Fuente: Autor

Una vez realizado el primer paso a continuación se procederá a la instalación de python en su versión 3.8.

Ilustración 14: Instalación de Python mediante consola de ubuntu

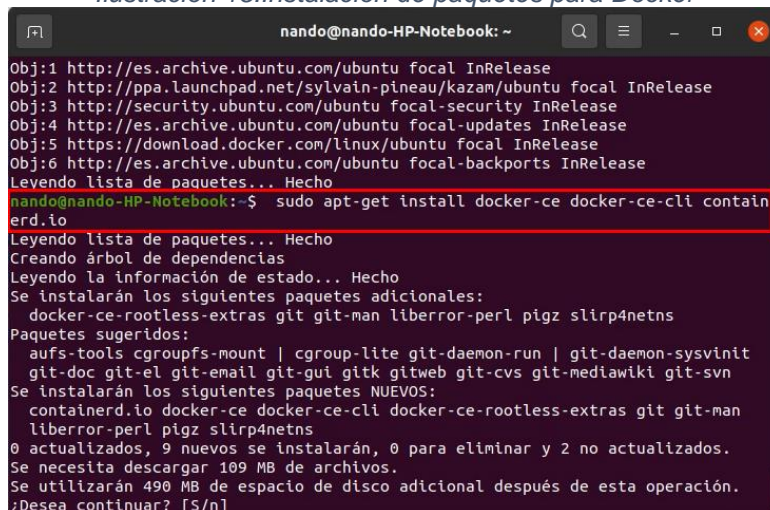


Fuente: Autor

Instalación de Docker

Antes de proceder a la instalación de Docker primero se instala todos los paquetes que pertenecen a la aplicación como se muestra a continuación.

Ilustración 15: Instalación de paquetes para Docker



Fuente: Autor

Después de haber instalado todos los paquetes correspondientes se procederá a la instalación de Docker.

Ilustración 16: Instalación de Docker mediante consola de ubuntu

```
nando@nando-HP-Notebook:~$ sudo apt-get install docker-ce=5:19.03.13~3-0~ubuntu-focal docker-ce-cli=5:19.03.13~3-0~ubuntu-focal containerd.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
containerd.io ya está en su versión más reciente (1.4.4-1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  docker-ce-rootless-extras slirp4netns
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  aufs-tools cgroupfs-mount
Se instalarán los siguientes paquetes NUEVOS:
  aufs-tools cgroupfs-mount
Se DESACTUALIZARÁN los siguientes paquetes:
  docker-ce docker-ce-cli
0 actualizados, 2 nuevos se instalarán, 2 desactualizados, 0 para eliminar y 2 no actualizados.
Se necesita descargar 66,8 MB de archivos.
Se liberarán 515 kB después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 aufs-tools amd64 1:4.14+20190211-1ubuntu1 [104 kB]
0% [1 aufs-tools 9.517 B/104 kB 9%] [Conectado a download.docker.com (13.227.29
```

Fuente: Autor

Ejecución de Docker

Una vez instalada la aplicación procederemos a ejecutar para comprobar que Docker está instalado correctamente.

Ilustración 17: Ejecución de Docker

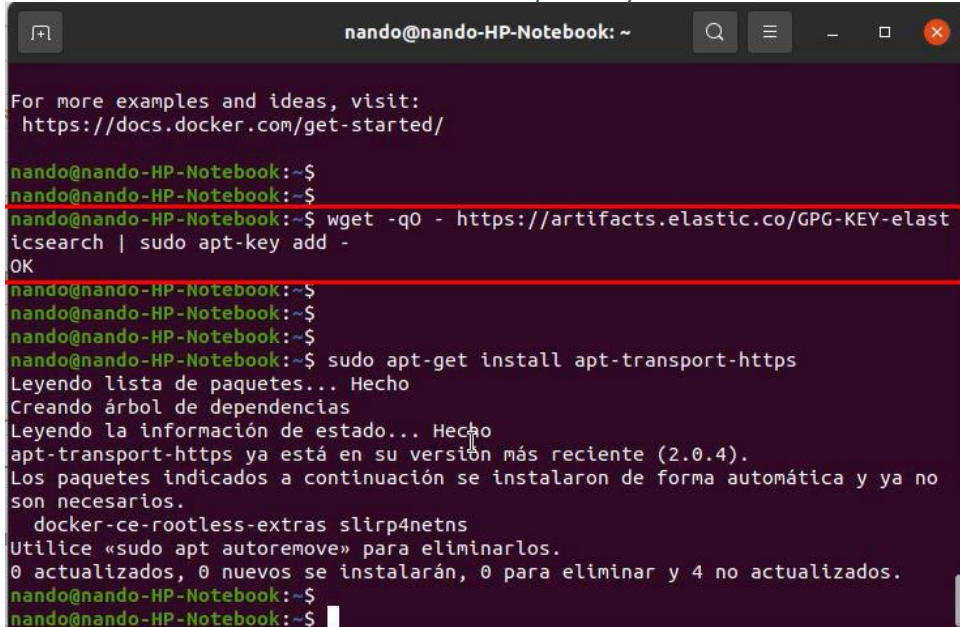
```
nando@nando-HP-Notebook:~$ sudo apt-get install docker-ce=5:19.03.13~3-0~ubuntu-focal docker-ce-cli=5:19.03.13~3-0~ubuntu-focal containerd.io
Preparando para desempaquetar .../docker-ce-cli_5%3a19.03.13~3-0~ubuntu-focal_amd64.deb ...
Desempaquetando docker-ce-cli (5:19.03.13~3-0~ubuntu-focal) sobre (5:20.10.5~3-0~ubuntu-focal) ...
dpkg: atención: desactualizando docker-ce de 5:20.10.5~3-0~ubuntu-focal a 5:19.03.13~3-0~ubuntu-focal
Preparando para desempaquetar .../docker-ce_5%3a19.03.13~3-0~ubuntu-focal_amd64.deb ...
Desempaquetando docker-ce (5:19.03.13~3-0~ubuntu-focal) sobre (5:20.10.5~3-0~ubuntu-focal) ...
Configurando aufs-tools (1:4.14+20190211-1ubuntu1) ...
Configurando docker-ce-cli (5:19.03.13~3-0~ubuntu-focal) ...
Configurando cgroupfs-mount (1.4) ...
Configurando docker-ce (5:19.03.13~3-0~ubuntu-focal) ...
Procesando disparadores para systemd (245.4-4ubuntu3.5) ...
Procesando disparadores para man-db (2.9.1-1) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9.2) ...
nando@nando-HP-Notebook:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:308866a43596e83578c7dfa15e27a73011bdd402185a84c5cd7f32a88b501a24
Status: Downloaded newer image for hello-world:latest
```

Fuente: Autor

Instalación de Elasticsearch

Para poder instalar la librería Elasticsearch primero necesitamos instalar la clave pública del certificado para poder utilizar los paquetes de su repositorio.

Ilustración 18: Instalación de llave pública para Elasticsearch

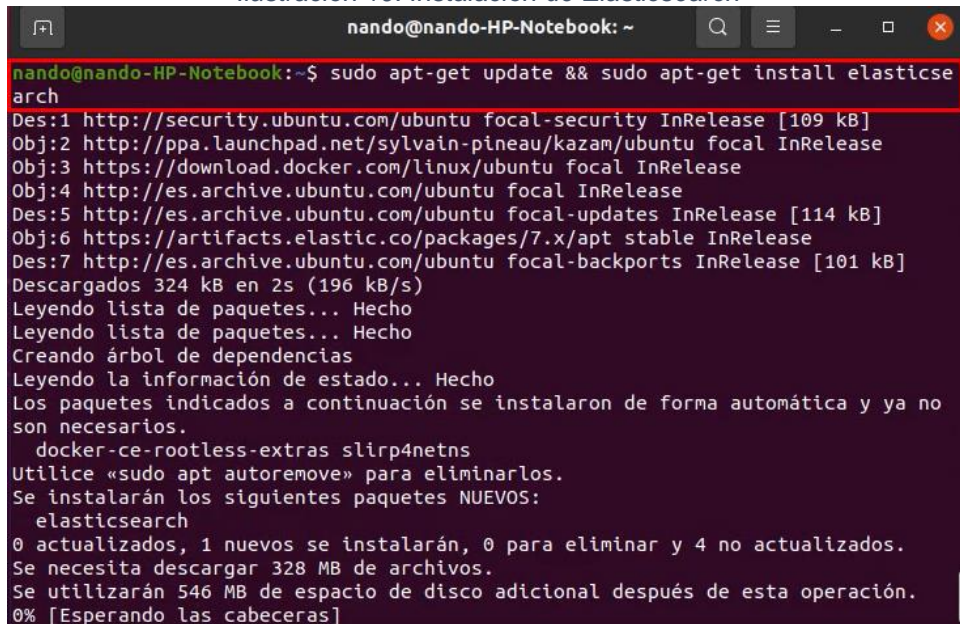


```
nando@nando-HP-Notebook: ~  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
  
nando@nando-HP-Notebook:~$  
nando@nando-HP-Notebook:~$  
nando@nando-HP-Notebook:~$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -  
OK  
nando@nando-HP-Notebook:~$  
nando@nando-HP-Notebook:~$  
nando@nando-HP-Notebook:~$  
nando@nando-HP-Notebook:~$ sudo apt-get install apt-transport-https  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
apt-transport-https ya está en su versión más reciente (2.0.4).  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
  docker-ce-rootless-extras slirp4netns  
Utilice «sudo apt autoremove» para eliminarlos.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 4 no actualizados.  
nando@nando-HP-Notebook:~$  
nando@nando-HP-Notebook:~$
```

Fuente: Autor

Una vez que se instalaron los paquetes necesarios procedemos a la instalación de la librería Elasticsearch.

Ilustración 19: Instalación de Elasticsearch



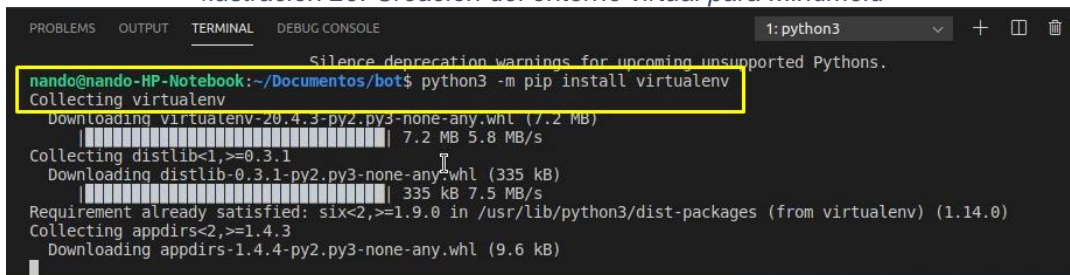
```
nando@nando-HP-Notebook: ~  
nando@nando-HP-Notebook:~$ sudo apt-get update && sudo apt-get install elasticse  
arch  
Des:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]  
Obj:2 http://ppa.launchpad.net/sylvain-pineau/kazam/ubuntu focal InRelease  
Obj:3 https://download.docker.com/linux/ubuntu focal InRelease  
Obj:4 http://es.archive.ubuntu.com/ubuntu focal InRelease  
Des:5 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Obj:6 https://artifacts.elastic.co/packages/7.x/apt stable InRelease  
Des:7 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]  
Descargados 324 kB en 2s (196 kB/s)  
Leyendo lista de paquetes... Hecho  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
  docker-ce-rootless-extras slirp4netns  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes NUEVOS:  
  elasticsearch  
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 4 no actualizados.  
Se necesita descargar 328 MB de archivos.  
Se utilizarán 546 MB de espacio de disco adicional después de esta operación.  
0% [Esperando las cabeceras]
```

Fuente: Autor

Instalación de Mindmeld

Primero iniciar Visual Studio, y abrir una terminal para crear un entorno aislado para luego proceder a la instalación de mindmeld.

Ilustración 20: Creación del entorno virtual para Mindmeld

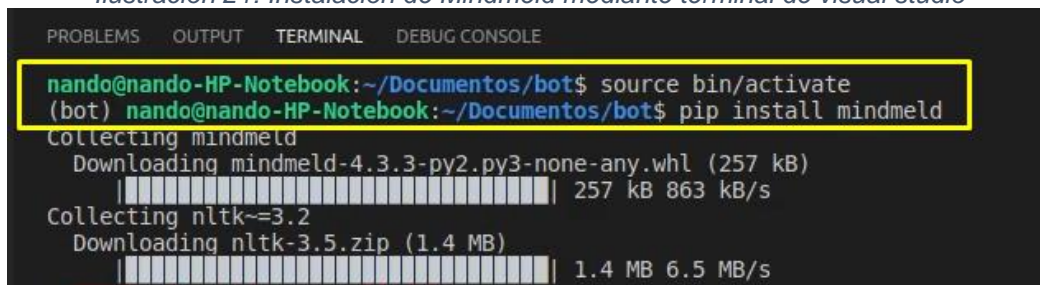


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: python3
Silence deprecation warnings for upcoming unsupported Pythons.
nando@nando-HP-Notebook:~/Documentos/bot$ python3 -m pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.4.3-py2.py3-none-any.whl (7.2 MB)
    |#####| 7.2 MB 5.8 MB/s
Collecting distlib<1,>=0.3.1
  Downloading distlib-0.3.1-py2.py3-none-any.whl (335 kB)
    |#####| 335 kB 7.5 MB/s
Requirement already satisfied: six<2,>=1.9.0 in /usr/lib/python3/dist-packages (from virtualenv) (1.14.0)
Collecting appdirs<2,>=1.4.3
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
```

Fuente: Autor

Una vez creado el entorno virtual procedemos a su activación y posteriormente procedemos a la instalación de mindmeld.

Ilustración 21: Instalación de Mindmeld mediante terminal de visual studio

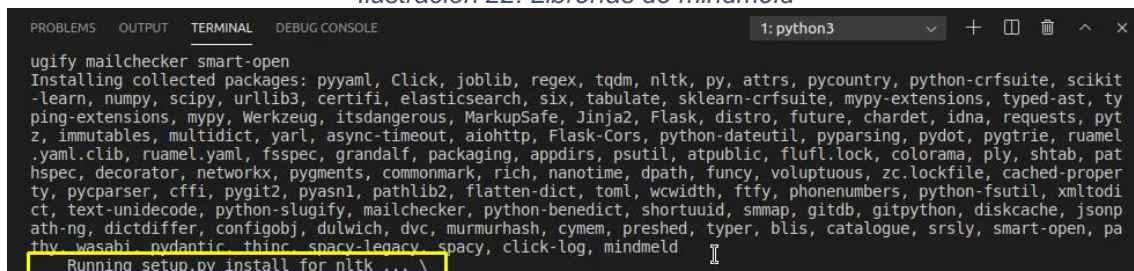


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
nando@nando-HP-Notebook:~/Documentos/bot$ source bin/activate
(bot) nando@nando-HP-Notebook:~/Documentos/bot$ pip install mindmeld
Collecting mindmeld
  Downloading mindmeld-4.3.3-py2.py3-none-any.whl (257 kB)
    |#####| 257 kB 863 kB/s
Collecting nltk~=3.2
  Downloading nltk-3.5.zip (1.4 MB)
    |#####| 1.4 MB 6.5 MB/s
```

Fuente: Autor

Como se puede observar en la ilustración 20, al instalar mindmeld se instalan por defecto todas sus dependencias inclusive las del procesamiento de lenguaje natural como se muestra a continuación.

Ilustración 22: Librerías de mindmeld



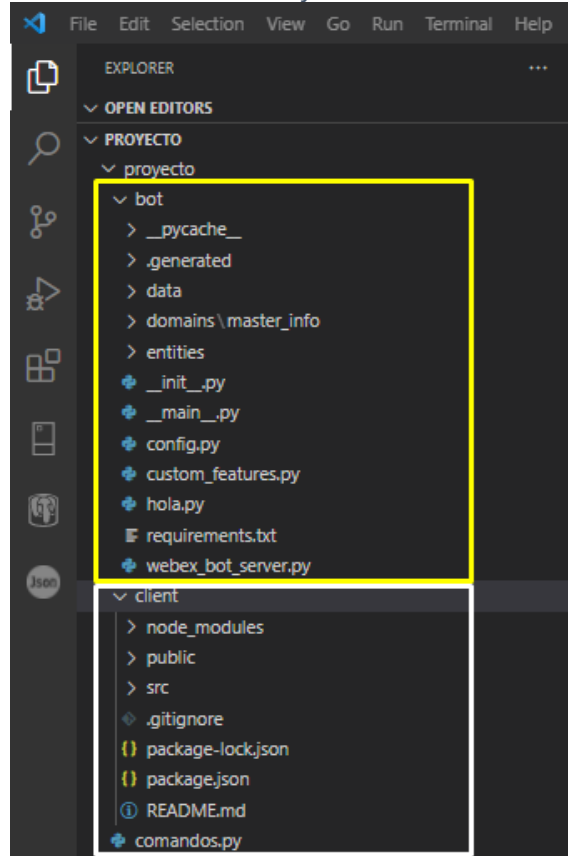
```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: python3
ugify mailchecker smart-open
Installing collected packages: pyyaml, Click, joblib, regex, tqdm, nltk, py, attrs, pycountry, python-crfsuite, scikit-learn, numpy, scipy, urllib3, certifi, elasticsearch, six, tabulate, sklearn-crfsuite, mpyy-extensions, typed-ast, ty, ping-extensions, mpyy, Werkzeug, itsdangerous, MarkupSafe, Jinja2, Flask, distro, future, chardet, idna, requests, pytz, immutables, multidict, yarl, async-timeout, aiohttp, Flask-Cors, python-dateutil, pyparsing, pydot, pygtrie, ruamel.yaml.clib, ruamel.yaml, fsspec, grandalf, packaging, appdirs, psutil, atpublic, fluf.lock, colorama, ply, shtab, pat, hspec, decorator, networkx, pygments, commonmark, rich, nanotime, dpath, funcy, voluptuous, zc.lockfile, cached-proper, ty, pycparser, cffi, pygit2, pyasn1, pathlib2, flatten-dict, toml, wcwidth, ftfy, phonenumbers, python-fsutil, xmltodict, text-unidecode, python-slugify, mailchecker, python-benedict, shortuuid, smmap, gitdb, gitpython, diskcache, jsonpath-ng, dictdiffer, configobj, dulwich, dvc, murmurhash, cymem, preshed, typer, blis, catalogue, srsly, smart-open, pathy, wasabi, pydantic, thinc, snacy-legacy, spacy, click-log, mindmeld
Running setup.py install for nltk ... \
```

Fuente: Autor

Codificación del chatbot

A continuación, se especificará el desarrollo del chatbot el cual constara de 2 partes que son el **bot** que será el Back End donde el chatbot aprenderá y el cliente que será el Front End la interfaz gráfica del chatbot de donde se comunicara con el usuario.

Ilustración 23: Back End y Front End del chatbot



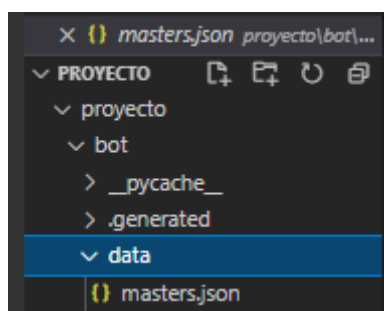
Fuente: Autor

Codificación del Back End del Chatbot

La carpeta bot está compuesta por la base del conociendo, los dominios y las intenciones.

A continuación, se mostrará la base del conocimiento.

Ilustración 24: Base del conocimiento del chatbot



Fuente: Autor

En el archivo **masters.json** que se encuentra dentro de la carpeta data que es el base del conocimiento guarda toda la información referente a las maestrías de la UTMACH y es la respuesta que dará el chatbot cuando el usuario pregunte por las maestrías.

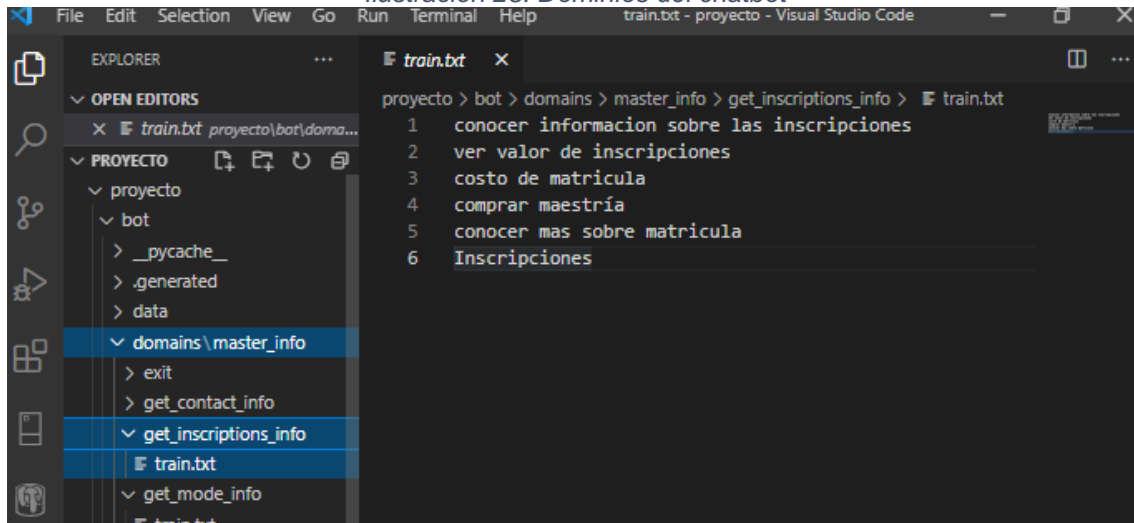
```

[
  {
    "id": "1",
    "name": "Maestría en Software",
    "titulo": {
      "nombre": "Magister en Software",
      "mencion": ""
    },
    "costo": {
      "valor": "$ 6.000 (sin incluir admisión)",
      "arancel": "$5.500",
      "matricula": "$500",
      "admisión": "$100"
    },
    "requisitos": [
      "Copia certificada del título de tercer nivel",
      "Certificado de inglés A1",
      "Referencias laboral y académica"
    ],
    "modalidad": "En Línea",
    "triptico": "https://posgrado.utmachala.edu.ec/posgrado/wp-content/uploads/2020/09/triptico-redisenado-SOFTWARE.pdf",
    "area": "Informatica"
  },
  {
    "id": "2",
    "name": "Maestría en Ingeniería Civil",
    "titulo": {
      "nombre": "Magister en Ingeniería Civil",
      "mencion": "mención Vialidad"
    },
    "costo": {
      "valor": "$ 9.346 (sin incluir admisión)",
      "arancel": "$8.546",
      "matricula": "$800",
      "admisión": "$150"
    },
    "requisitos": [
      "Fotocopia a color certificada del título de tercer nivel",
      "Certificado de inglés A1",
      "Solicitud de ingreso"
    ],
    "modalidad": "Híbrida",
    "triptico": "https://posgrado.utmachala.edu.ec/posgrado/wp-content/uploads/2021/01/TRIPTICO_MAESTRIA_CIVIL_LADO1.pdf",
    "area": "Ingenieria civil" },
]

```

Los dominios son considerados como el tema principal de la conversación y el chatbot aprenderá de estos dominios mientras más se utilice.

Ilustración 25: Dominios del chatbot

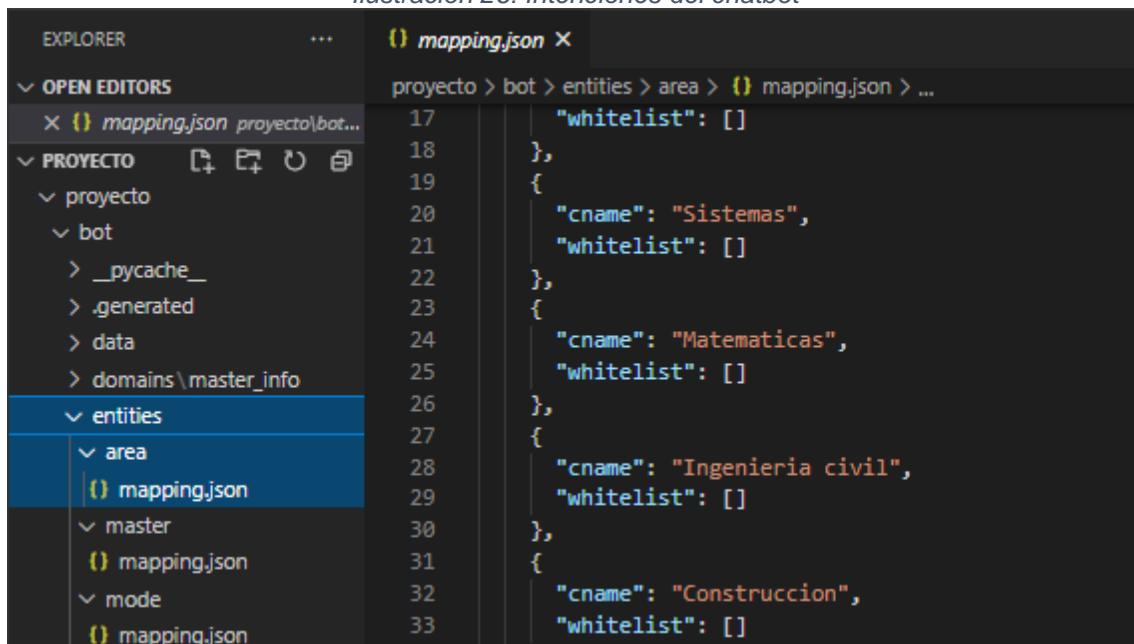


Autor: Autor

Como se puede observar en la ilustración 25 los dominios (temas principales de conversación) están guardados por carpetas y cada carpeta tiene un archivo de entrenamiento llamado train.txt como se aprecia en la imagen donde el tema principal son las inscripciones.

Y las intenciones que serían las acciones que tienen que ver con el dominio.

Ilustración 26: Intenciones del chatbot



Fuente: Autor

Como se aprecia en la ilustración 26 las intenciones son acciones que estarían ligadas a la base del conocimiento en este caso tenemos la carpeta área y sus acciones serían una rama ligada a la maestría.

En cuanto a la parte lógica del chatbot la encontramos en el archivo init.py como se detalla a continuación.

En la primera parte del código se basa en la respuesta que dará el chatbot cuando no tiene conocimiento de lo que el usuario pregunta.

```
"""This module contains a template MindMeld application"""
from mindmeld import Application

app = Application(__name__)

__all__ = ['app']

@app.handle(default=True)
@app.handle(intent='unsupported')
def default(request, responder):
    """This is a default handler."""
    print("unsupported")
    replies = ["No he encontrado lo que buscabas. Puedo ayudarte si empiezas por decirme "
               "el área de la maestría que te gustaria conocer"]
    responder.reply(replies)
```

A continuación, se mostrará el proceso de buscar una maestría.

La función **def set_master** se conecta con la base del conocimiento haciendo todo el procesamiento del lenguaje natural y si encuentra la petición que realizo el usuario le responde adecuadamente caso contrario presentara el mensaje que se expuso en la primera parte.

```
@app.handle(intent='set_master')
def set_master(request, responder):
    print("set_master")
    selected_master = request.frame.get('master')

    master_entity = next(
        (e for e in request.entities if e['type'] == 'master'), None)
    print(master_entity)

    if master_entity:
        if len(master_entity['value']) > 0 and master_entity['value'][0]['score'] > 40:
```

```

        selected_master = _kb_fetch(
            'masters', master_entity['value'][0]['id'])
        responder.frame['master'] = selected_master
        responder.slots['master_name'] = selected_master['name']
        responder.reply(
            "La UTMACH ofrece la {master_name}, que información desea
s conocer.")
        responder.reply("Inscripciones")
        responder.reply("Modalidades")
        # responder.reply("Horarios")
        responder.reply("Requerimientos")
        responder.reply("Malla curricular")
        # responder.reply("Información de contacto")
        responder.listen()
    else:
        responder.slots['master_name'] = master_entity['text']
        responder.reply(
            "Al parecer no contamos con {master_name}. Puedes probar
ingresando el área de estudio")
        responder.listen()
    return

```

Para finalizar la parte del back end tenemos el archivo webex_bot_server.py que es donde se realizará la importación de mindmeld para el procesamiento del lenguaje natural y también permitirá la comunicación con el front end.

```

# -*- coding: utf-8 -*-
import os
from flask import Flask, request, make_response, jsonify
from mindmeld.components import NaturalLanguageProcessor
from mindmeld.components.dialogue import Conversation
from mindmeld import configure_logs
from flask_cors import CORS
import json

app = Flask(__name__)
CORS(app)
configure_logs()
nlp = NaturalLanguageProcessor('.')
nlp.build()

conv = Conversation(nlp=nlp, app_path='.')
@app.route('/message', methods=['POST'])
def login():
    if request.method == 'POST':
        # try:
            data = json.loads(request.data)

```



```

message = data.get("message", None)
print(message)
res = conv.say(message)
r = {"response": res}
response = app.response_class(response=json.dumps(r),
                               status=200,
                               mimetype='application/json')

print(response)
return response
# except:
# print('Error')
# return 'No te he entendido, puedes intentar de otra forma'
# export FLASK_APP=webex_bot_server

if __name__ == '__main__':
    app.run(debug=True)

port_number = 5000
print('Running server on port {}'.format(port_number))

```

2.5. Ejecución del prototipo

Para poder ejecutar el proyecto primero necesitamos iniciar las librerías juntamente con Docker.

Primero iniciamos Elasticsearch

Ilustración 27: Iniciar librería Elasticsearch

```

nando@nando-HP-Notebook: ~
nando@nando-HP-Notebook:~$ sudo service elasticsearch start
[sudo] contraseña para nando:

```

Fuente: Autor

A continuación, inicializamos Docker para el procesamiento de la información

Ilustración 28: Ejecución de Docker

```

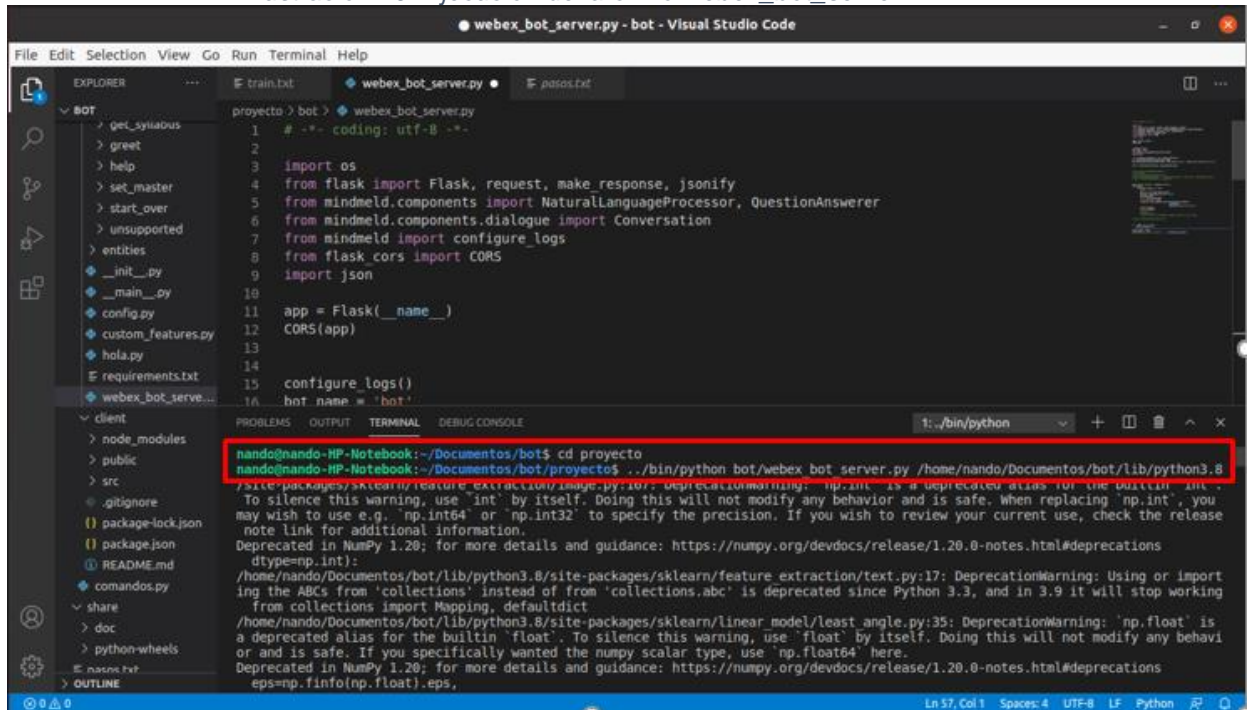
nando@nando-HP-Notebook: ~
nando@nando-HP-Notebook:~$ sudo service elasticsearch start
[sudo] contraseña para nando:
nando@nando-HP-Notebook:~$ docker pull mindmeldworkbench/duckling:master && dock
er run -p 0.0.0.0:7151:7151 mindmeldworkbench/duckling:master -ti -d
master: Pulling from mindmeldworkbench/duckling
Digest: sha256:fa2904e76e5026708dd75093679071e7c8e864f561040e71034b50a46d0cfc43
Status: Image is up to date for mindmeldworkbench/duckling:master
docker.io/mindmeldworkbench/duckling:master
Listening on http://0.0.0.0:7151

```

Fuente: Autor

Abrimos nuestro entorno de trabajo visual studio abrimos una nueva terminal y ejecutamos el archivo **webex_bot_server.py** el cual nos permitirá la comunicación entre el usuario y el chatbot.

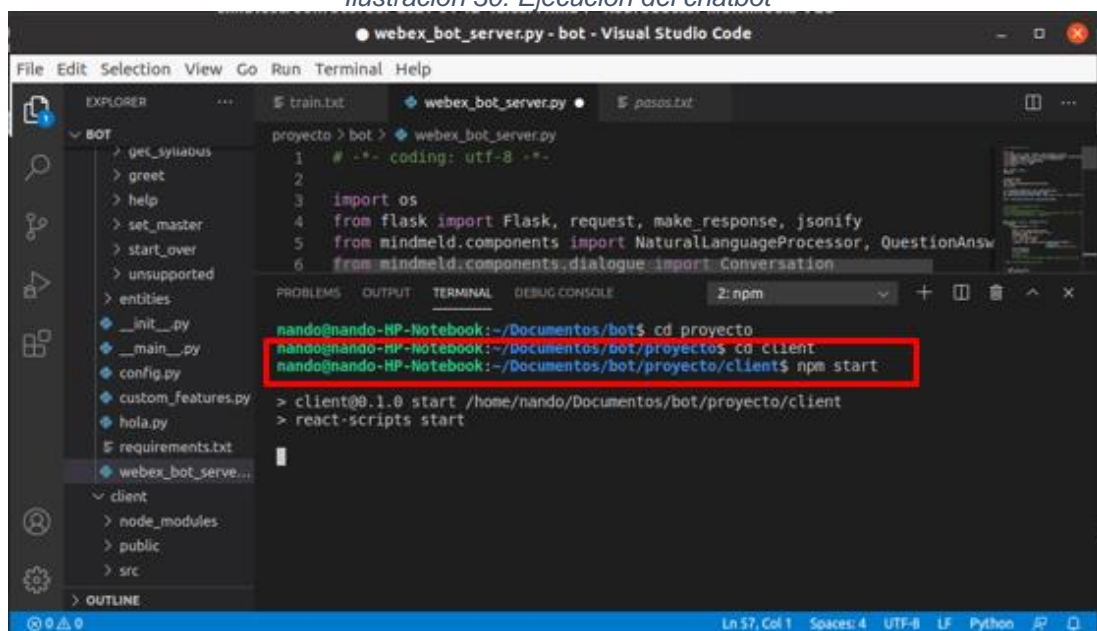
Ilustración 29: Ejecución del archivo *webex_bot_server*



Fuente: Autor

Después de haber realizado los pasos anteriores procedemos a abrir una nueva terminal donde se ejecutará la línea **npm start** para que el chatbot se ejecute como se muestra en la siguiente imagen.

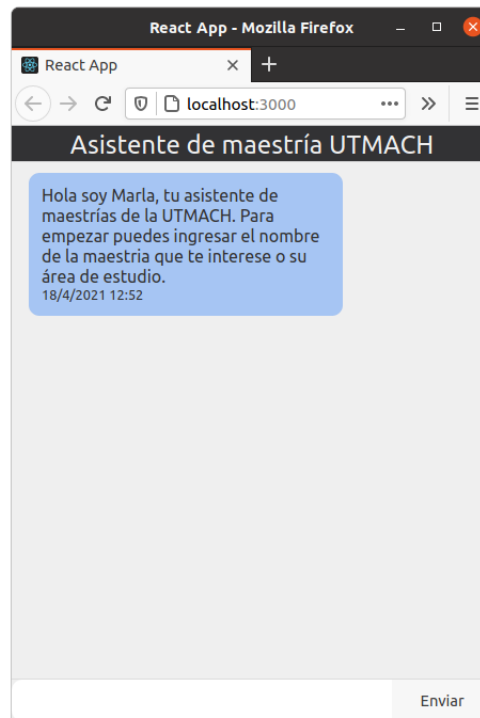
Ilustración 30: Ejecución del chatbot



Fuente: Autor

A continuación, nuestro chatbot llamado “Marla” se abrirá en el navegador Mozilla Firefox.

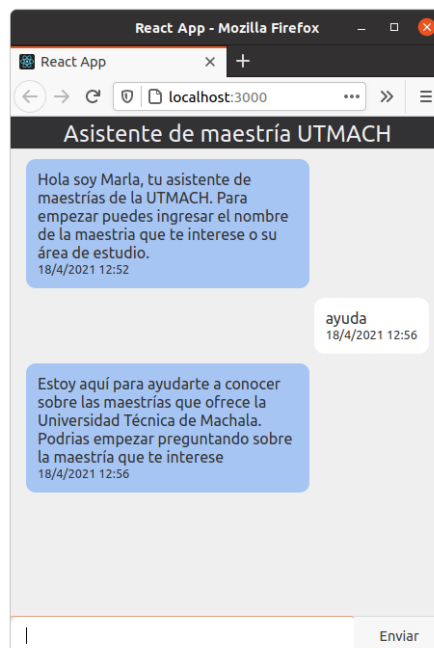
Ilustración 31: Ventana principal del chatbot



Fuente: Autor

Para comenzar a interactuar con Marla debemos escribirle tal como se muestra en la ilustración 32.

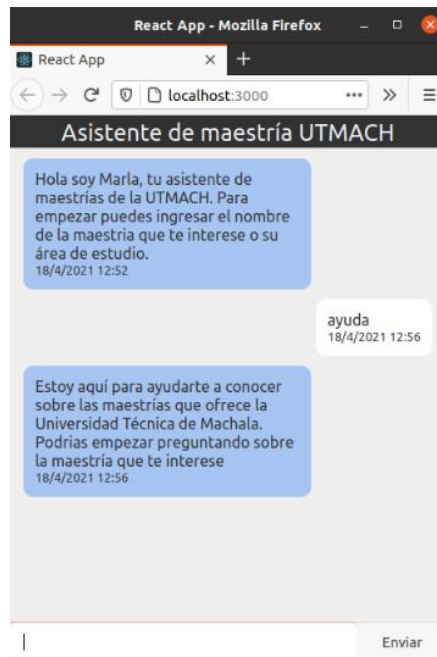
Ilustración 32: Saludo del chatbot



Fuente: Autor

La interacción con Marla continuará y le pediremos información sobre maestrías como se mostrará en la siguiente imagen:

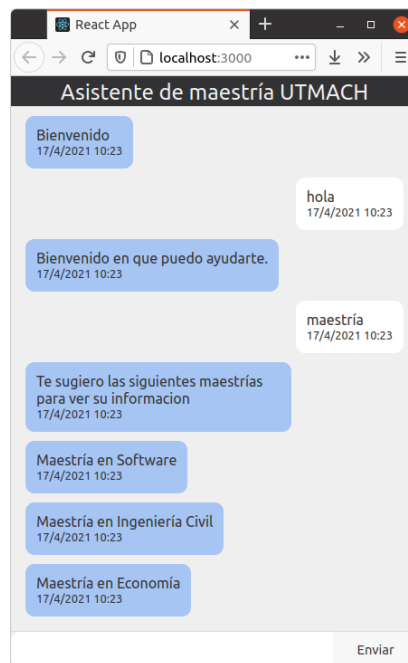
Ilustración 33: Interacción del usuario con el chatbot sobre las maestrías que oferta la UTMACH



Fuente: Autor

Después que Marla nos mostró las maestrías que oferta la UTMACH le pediremos información sobre la maestría en software.

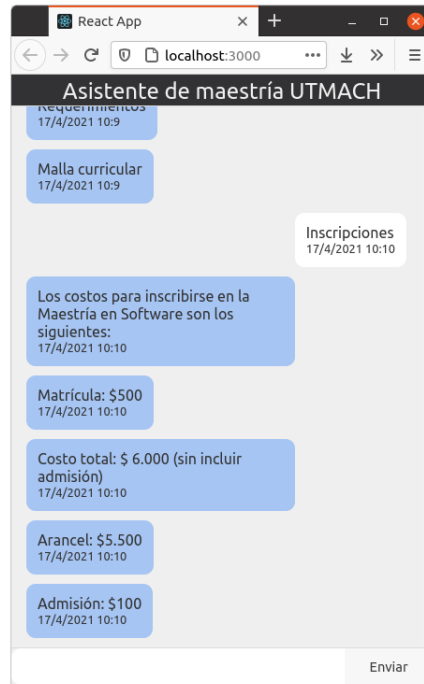
Ilustración 34: Tipos de maestrías que ofrece la UTMACH



Fuente: Autor

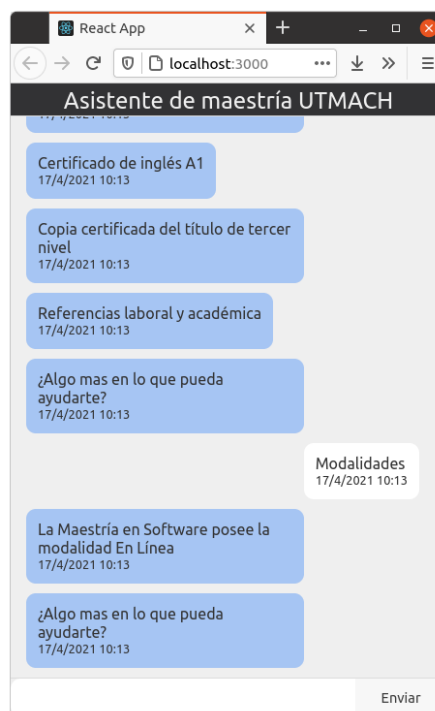
A continuación, seguiremos interactuando con Marla y le pediremos información sobre cada una de las opciones que nos ofrece la maestría seleccionada como se mostrará a continuación mediante las ilustraciones y al finalizar la interacción Marla se despedirá.

Ilustración 35: Inscripciones de la maestría



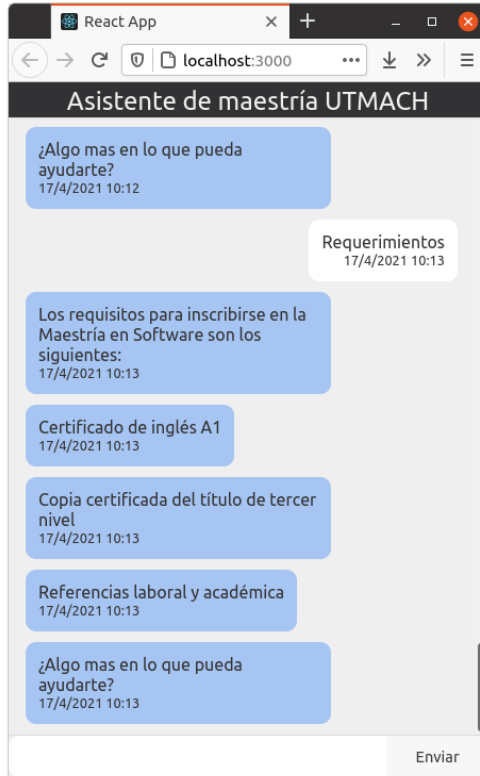
Fuente: Autor

Ilustración 36: Modalidades de la maestría



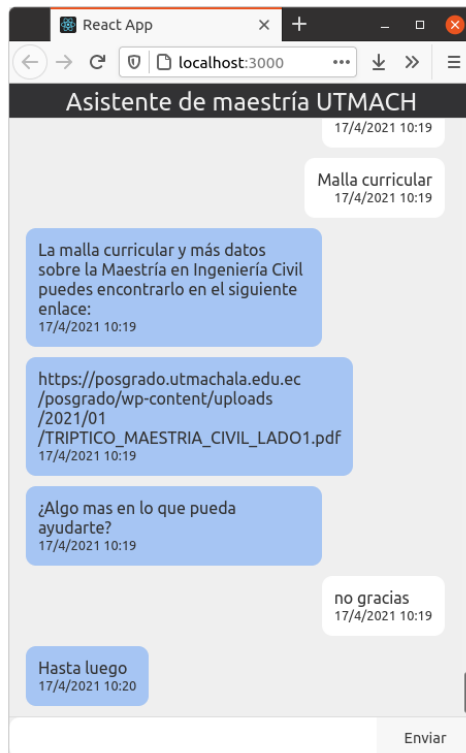
Fuente: Autor

Ilustración 37: Requisitos de la maestría



Fuente: Autor

Ilustración 38: Malla curricular de la maestría



Fuente: Autor

3. CAPITULO III: EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

Para evaluar los resultados del chatbot se utilizaron métricas de clasificación, los parámetros a evaluar fueron los dominios y las intenciones.

3.1.1. Métrica de clasificación

Se utilizará el concepto de la matriz de confusión, en donde los resultados Verdaderos Positivos son la cantidad de dominios e intenciones detectadas de forma correcta, los Falsos Positivos son el total de dominios e intenciones no detectados, que no se encontraron el procesamiento de la información mediante mindmeld [52].

Tabla 2: Matriz de confusión

		Predicción del Chatbot	
		Dominio e Intención Presente	Dominio e Intención no Presente
Situación Real	Dominio e Intención Presente	Verdadero Positivo	Falso Negativo
	Dominio e Intención no Presente	Falso Positivo	Verdadero Negativo

Fuente: [43]

Mediante esta teoría se pueden calcular las métricas de calidad para detectar si la información que nos presenta el chatbot es la correcta.

Precisión

La precisión se calculará por cada intención y al finalizar se promedian los resultados generales. El verdadero positivo es la detección correcta de la intención. El falso negativo corresponde a que nos dará un resultado diferente al que esperábamos (una de las otras intenciones). Falso positivo corresponde a otras intenciones a continuación, se mostrará la fórmula para calcular la precisión.

$$\text{Precisión} = \frac{\text{Verdadero Positivo}}{\text{Verdadero Positivo} + \text{Falso Positivo}}$$

Sensibilidad (Recall)

La sensibilidad se calculará por cada intención y al finalizar se promedian los resultados generales a continuación, se mostrará la fórmula para calcular la sensibilidad.

$$\text{Sensibilidad} = \frac{\text{Verdadero Positivo}}{\text{Verdadero Positivo} + \text{Falso Negativo}}$$

3.2. Resultados de la evaluación

Para la matriz de confusión tomamos en cuenta:

- **Base de conocimiento**
- **Dominios**
- **Intenciones**
- **Saludo y despedida.**

En base a estos 4 ítems procedemos a realizar las pruebas.

Resultado prueba 1

Tabla 3: Matriz de confusión N° 1

saludo y despedida	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	9	1	
	negativo	1	29	

base de conocimiento	PREDICCIÓN			Total
actual		positivo	Negativo	40
	positivo	5	5	
	negativo	1	29	

dominios	PREDICCIÓN			Total
actual		positivo	Negativo	40
	positivo	6	4	
	negativo	2	28	

intenciones	PREDICCIÓN			Total
actual		positivo	Negativo	40
	positivo	8	2	
	negativo	2	28	

Fuente: Autor

Tabla 4: Predicción del chatbot matriz N° 1

B		Predicción del chatbot					total
		S y D	. Con	Dominios	Intenciones	ND	
Test	saludo y despedida	9	0	0	0	1	10
	base de conocimiento	1	5	2	0	2	10
	dominios	0	1	6	2	1	10
	intenciones	0	0	0	8	2	10
	total	10	6	8	10	6	40

Fuente: Autor

Tabla 5: Predicción del aprendizaje matriz N° 1

		Predicción del aprendizaje (%)					total
		S y D	B. Con	Dominios	Intenciones	ND	
Test	saludo y despedida	90	0	0	0	10	100
	base de conocimiento	10	50	20	0	20	100
	dominios	0	10	60	20	10	100
	intenciones	0	0	0	80	20	100
	total	100	60	80	100	60	

Fuente: Autor

Tabla 6: Resultado final de la matriz N° 1

	TOTAL EXPRESIONES	ACIERTOS	PRECISION	ESPECIFICIDAD	RECALL
saludo y despedida	10	9	0,90	0,97	0,90
base de conocimiento	10	5	0,83	0,97	0,50
dominios	10	6	0,75	0,93	0,60
intenciones	10	8	0,80	0,93	0,80

Fuente: Autor

En la primera matriz de aprendizaje después de analizar los 4 ítems propuestos para la evaluación nos damos cuenta que el número de aciertos en base al conocimiento es muy poco en comparación con saludo y despedida, aunque a nivel general la precisión y sensibilidad del aprendizaje es de un 76%.

Resultado prueba 2

Tabla 7: Matriz de confusión N° 2

saludo y despedida	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	10	0	
	negativo	0	30	

base de conocimiento	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	7	3	
	negativo	0	30	

dominios	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	8	2	
	negativo	1	29	

intenciones	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	9	1	
	negativo	0	30	

Fuente: Autor

Tabla 8: Predicción del chatbot matriz N° 2

		Predicción del chatbot					total
		S y D	B. Con	Dominios	Intenciones	ND	
Test	saludo y despedida	10	0	0	0	0	10
	base de conocimiento	1	7	1	0	1	10
	dominios	1	0	8	0	1	10
	intenciones	0	0	0	9	1	10
	total	12	7	9	9	3	40

Fuente: Autor

Tabla 9: Predicción del aprendizaje matriz N° 2

		Predicción del aprendizaje (%)					total
		S y D	B. Con	Dominios	Intenciones	ND	
Test	saludo y despedida	100	0	0	0	0	100
	base de conocimiento	10	70	10	0	10	100
	dominios	10	0	80	0	10	100
	intenciones	0	0	0	90	10	100
	total	120	70	90	90	30	

Fuente: Autor

Tabla 10: Resultados finales matriz N° 2

	TOTAL EXPRESIONES	ACIERTOS	PRECISIÓN	ESPECIFICIDAD	RECALL
saludo y despedida	10	9	1,00	1,00	1,00
base de conocimiento	10	7	1,00	1,00	0,70
dominios	10	8	0,89	0,97	0,80
intenciones	10	9	1,00	1,00	0,90

Fuente: Autor

En la segunda matriz de aprendizaje después de analizar los 4 ítems propuestos para la evaluación nos damos cuenta que el número de aciertos en base al conocimiento es muy poco en comparación con saludo y despedida, aunque a nivel general la precisión y sensibilidad del aprendizaje es de un 91%.

Resultado prueba 3

Tabla 11: Matriz de confusión N° 3

saludo y despedida	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	10	0	
	negativo	0	30	

base de conocimiento	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	10	0	
	negativo	0	30	

dominios	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	9	0	
	negativo	1	30	

intenciones	PREDICCIÓN			Total
actual		positivo	negativo	40
	positivo	9	0	
	negativo	1	30	

Tabla 12: Predicción del chatbot matriz N° 3

		Predicción del chatbot					total
		S y D	B. Con	Dominios	Intenciones	ND	
Test	saludo y despedida	10	0	0	0	0	10
	base de conocimiento	0	10	0	0	0	10
	dominios	0	0	9	1	1	10
	intenciones	0	0	0	9	0	10
	total	10	10	9	10	1	40

Fuente: Autor

Tabla 13: Predicción del aprendizaje matriz N° 3

		Predicción del modelo (%)					total
		S y D	B. Con	Dominios	Intenciones	ND	
Test	saludo y despedida	100	0	0	0	0	100
	base de conocimiento	0	100	0	0	0	100
	dominios	0	0	90	10	10	110
	intenciones	0	0	0	100	0	90
	total	100	100	90	100	10	

Fuente: Autor

Tabla 14: Resultados finales de la matriz N° 3

	TOTAL, EXPRESIONES	ACIERTOS	PRECISION	ESPECIFICIDAD	RECALL
saludo y despedida	10	10	1,00	1,00	1,00
base de conocimiento	10	10	1,00	1,00	1,00
dominios	10	9	0,90	0,97	1,00
intenciones	10	9	0,90	0,97	1,00

Fuente: Autor

Al finalizar las pruebas de evaluación con la matriz de confusión N° 3 nos damos cuenta que el aprendizaje del chatbot fue aumentando conforme se lo iba utilizando en la última prueba el resultado de la predicción a las peticiones del usuario fueron de un 95% en precisión y sensibilidad del aprendizaje utilizado.

3.3. Conclusiones

- El desarrollo del presente proyecto de titulación se realizó con éxito mediante el uso de herramientas tecnológicas de programación como fue Python y React por su lenguaje de alto nivel que permitió tener flexibilidad y escalabilidad en el desarrollo del chatbot además se usaron APIS para el procesamiento del lenguaje natural, mediante el aprendizaje automático y Docker dando como resultado final el asistente virtual para la UTMACH llamado “Marla” que facilitará información a los usuarios sobre las maestrías que ofrece la universidad.
- Se utilizó la plataforma Mindmeld para el procesamiento del lenguaje natural y la generación del lenguaje permitiendo al chatbot comunicarse con el usuario de forma automática.
- Se hizo uso de los framework React y Flask para el desarrollo de la aplicación web con la finalidad de crear una interfaz amigable del chatbot para el usuario.
- El análisis de la información que se recopiló para el desarrollo de este proyecto fue basado en fuentes confiables permitiendo crear un documento fiable y seguro.
- Mientras más se utilice el asistente virtual su aprendizaje automático permitirá que vaya aprendiendo de cada conversación y así poder responder de forma correcta y acertada la petición del usuario.

3.4. Recomendaciones

- Utilizar un framework que crea conveniente para el proyecto y que cumpla con las necesidades planteadas por el usuario.
- Instalar las librerías necesarias y con la versión correcta caso contrario puede crear un problema de compatibilidad el cual puede ocasionar que el proyecto no se termine de forma correcta.
- Interactuar el mayor tiempo con el chatbot para que el procesamiento del lenguaje natural genere más contenido para la base de conocimiento.

4. Bibliografía

- [1] C. González García, E. Núñez-Valdez, V. García-Díaz, C. Pelayo G-Bustelo, and J. M. Cueva-Lovelle, "A Review of Artificial Intelligence in the Internet of Things," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 5, no. 4, p. 9, 2019, doi: 10.9781/ijimai.2018.03.004.
- [2] V. Gómez and J. Andrés, "Problemas bioéticos emergentes de la inteligencia artificial," *Divers. Perspect. en Psicol.*, vol. 12, no. 1, pp. 137–147, 2016, [Online]. Available: <https://www.redalyc.org/articulo.oa?id=67945904010>.
- [3] S. J. Russell and P. Norvig, *Artificial Intelligence A Modern Approach Fourth Edition*, Person Edu. United States of America: Person, 2020.
- [4] Universidad Casa Grande, "Universidad Casa Grande - Hacer para ser." <https://www.casagrande.edu.ec/> (accessed Apr. 21, 2021).
- [5] Universidad Técnica Particular de Loja., "UTPL | Decide ser más," 2020. <https://www.utpl.edu.ec/> (accessed Apr. 21, 2021).
- [6] Elsevier Connect, "Asistentes virtuales de voz: una 'revolución silenciosa' en el ámbito de la salud," Aug. 11, 2020.
- [7] Canonical, "Enterprise Open Source and Linux | Ubuntu," 2016. <https://ubuntu.com/> (accessed Apr. 21, 2021).
- [8] IONOS, "Ubuntu: un sistema para todos basado en Linux - IONOS." Accessed: Apr. 21, 2021. [Online]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/ubuntu-un-sistema-para-todos-basado-en-linux/>.
- [9] Cisco System, "Introduction to Conversational Applications — The Conversational AI Playbook 4.3.4rc5 documentation," 2019. https://www.mindmeld.com/docs/intro/introduction_to_conversational_applications.html (accessed Apr. 21, 2021).
- [10] L. J. Catania and E. Nicolitz, "Artificial Intelligence and Its Applications in Vision and Eye Care," *Adv. Ophthalmol. Optom.*, vol. 3, no. 1, pp. 21–38, Aug. 2018, doi: 10.1016/j.yaoo.2018.04.001.
- [11] SpaCy, "spaCy · Industrial-strength Natural Language Processing in Python," *Official website*, 2020. <https://spacy.io/> (accessed Apr. 21, 2021).
- [12] E. Lopes, E. Klein, and S. Bird, "Natural Language Toolkit (NLTK)," pp. 1–11, 2019, Accessed: Apr. 21, 2021. [Online]. Available: <http://www.nltk.org/>.
- [13] React, "React – Una biblioteca de JavaScript para construir interfaces de usuario," 2019. <https://es.reactjs.org/> (accessed Apr. 21, 2021).
- [14] A. Ronacher, "Welcome to Flask — Flask Documentation (1.1.x)," *Pallets Projects*, 2020. <https://flask.palletsprojects.com/en/1.1.x/> (accessed Apr. 21, 2021).
- [15] elastic, "Elasticsearch: El motor de búsqueda y analítica distribuido oficial | Elastic." <https://www.elastic.co/es/elasticsearch/> (accessed Apr. 21, 2021).
- [16] D. Chen *et al.*, "Real-Time or Near Real-Time Persisting Daily Healthcare Data Into HDFS and Elasticsearch Index Inside a Big Data Platform," *IEEE Trans. Ind. Informatics*, vol. 13, no. 2, pp. 595–606, Apr. 2017, doi: 10.1109/TII.2016.2645606.
- [17] M. D. Gallego Calabria, "Inteligencia artificial: ¿por qué no puede ser?," *Neurol. Argentina*, vol. 12, no. 1, pp. 1–3, Jan. 2020, doi: 10.1016/j.neuarg.2020.01.001.

- [18] J. F. Avila-Tomás, M. A. Mayer-Pujadas, and V. J. Quesada-Varela, "La inteligencia artificial y sus aplicaciones en medicina I: introducción antecedentes a la IA y robótica," *Atención Primaria*, vol. 52, no. 10, pp. 778–784, Dec. 2020, doi: 10.1016/j.aprim.2020.04.013.
- [19] A. Dúran Pincheira, "Diseño e implementación de un asesor virtual con interfaz web basado en un sistema de gestión de conocimientos y autoaprendizaje," ESPE UNIVERSIDAD DE LAS FUERZAS ARMADAS, 2015.
- [20] R. A. Manjarrés-Betancur and M. M. Echeverri-Torres, "Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural," *Rev. Politécnica*, vol. 16, no. 31, pp. 85–95, May 2020, doi: 10.33571/rpolitec.v16n31a7.
- [21] M. Lu and F. Li, "Survey on lie group machine learning," *Big Data Min. Anal.*, vol. 3, no. 4, pp. 235–258, Dec. 2020, doi: 10.26599/BDMA.2020.9020011.
- [22] G. Mcgraw, R. Bonett, H. Figueroa, and V. Shepardson, "Security Engineering for Machine Learning," *Computer (Long. Beach. Calif.)*, vol. 52, no. 8, pp. 54–57, Aug. 2019, doi: 10.1109/MC.2019.2909955.
- [23] M. Gramajo, L. Ballejos, and M. Ale, "Seizing Requirements Engineering Issues through Supervised Learning Techniques," *IEEE Lat. Am. Trans.*, vol. 18, no. 07, pp. 1164–1184, Jul. 2020, doi: 10.1109/TLA.2020.9099757.
- [24] Y.-F. Li, L.-Z. Guo, and Z.-H. Zhou, "Towards Safe Weakly Supervised Learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2019, doi: 10.1109/TPAMI.2019.2922396.
- [25] Y. Xu, Y. Zhou, P. Sekula, and L. Ding, "Machine learning in construction: From shallow to deep learning," *Dev. Built Environ.*, vol. 6, p. 100045, May 2021, doi: 10.1016/j.dibe.2021.100045.
- [26] J. Camacho, S. Moreno, F. Suarez-Obando, J. Carlos Puyana, and C. Gomez-Restrepo, "El procesamiento de lenguaje natural y su relación con la investigación en salud mental," *Rev. Colomb. Psiquiatr.*, vol. 42, no. 2, pp. 227–233, Jun. 2013, doi: 10.1016/S0034-7450(13)70011-8.
- [27] G. Tarantino, S. Monica, and F. Bergenti, "A probabilistic matrix factorization algorithm for approximation of sparse matrices in natural language processing," *ICT Express*, vol. 4, no. 2, pp. 87–90, Jun. 2018, doi: 10.1016/j.ict.2018.04.005.
- [28] A. Agarwal, C. Baechle, R. Behara, and X. Zhu, "A Natural Language Processing Framework for Assessing Hospital Readmissions for Patients With COPD," *IEEE J. Biomed. Heal. Informatics*, vol. 22, no. 2, pp. 588–596, Mar. 2018, doi: 10.1109/JBHI.2017.2684121.
- [29] L. Machado Redrovan, "Desarrollo de un chatbot web como asistente de ventas," 2019.
- [30] N. Alvarenga Alcaraz and P. Alvarenga Alcaraz, "Web application for Analysis and Automatic Translation Guarani - Spanish and Spanish - Guarani," *Rev. Científica la UCSA*, vol. 7, no. 2, pp. 41–69, Aug. 2020, doi: 10.18004/ucsa/2409-8752/2020.007.02.041.
- [31] B. Srivastava, F. Rossi, S. Usmani, and M. Bernagozzi, "Personalized Chatbot Trustworthiness Ratings," *IEEE Trans. Technol. Soc.*, vol. 1, no. 4, pp. 184–192, Dec. 2020, doi: 10.1109/TTS.2020.3023919.
- [32] M. Arain, M. A. Memon, S. Bhatti, and M. Arain, "Feasibility of Chatbot for Mehran UET Examination Department," *Rev. Inf. Eng. Appl.*, vol. 6, no. 2, pp. 17–28, 2019, doi:

10.18488/journal.79.2019.62.17.28.

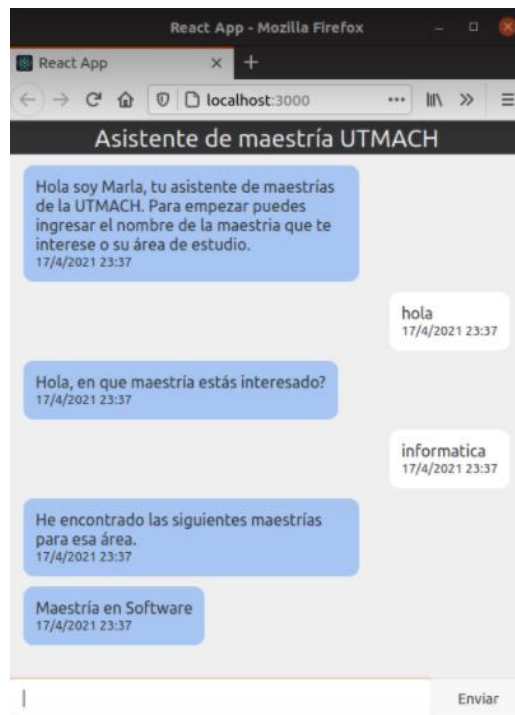
- [33] N. Akma, M. Hafiz, A. Zainal, M. Fairuz, and Z. Adnan, "Review of Chatbots Design Techniques," vol. 181, no. 8, pp. 7–10, Aug. , doi: 10.5120/ijca2018917606.
- [34] M. Nuruzzaman and O. K. Hussain, "IntelliBot: A Dialogue-based chatbot for the insurance industry," *Knowledge-Based Syst.*, vol. 196, p. 105810, May 2020, doi: 10.1016/j.knosys.2020.105810.
- [35] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Mach. Learn. with Appl.*, vol. 2, p. 100006, Dec. 2020, doi: 10.1016/j.mlwa.2020.100006.
- [36] R. DALE, "The return of the chatbots," *Nat. Lang. Eng.*, vol. 22, no. 5, pp. 811–817, Sep. 2016, doi: 10.1017/S1351324916000243.
- [37] M. T. ZEMČÍK, "A Brief History of Chatbots," *DEStech Trans. Comput. Sci. Eng.*, no. aicae, Oct. 2019, doi: 10.12783/dtcse/aicae2019/31439.
- [38] G. Cox, "About Chatterbot," 2020. <https://chatterbot.readthedocs.io/en/stable/> (accessed Apr. 21, 2021).
- [39] G. P. Piedra-Illescas and D. M. Cordero-Guzmán, "Chatbot como herramienta de disminución de la intervención humana en la resolución de incidencias de ofimática en la Coordinación Zonal 7 – Salud," *Dominio las Ciencias*, vol. 5, no. 2, pp. 376–409, 2019, doi: <http://dx.doi.org/10.23857/dom.cien.pocaip.2019.5.2.abril.376-388>.
- [40] S. Arsovski, H. Osipyan, M. I. Oladele, and A. D. Cheek, "Automatic knowledge extraction of any Chatbot from conversation," *Expert Syst. Appl.*, vol. 137, pp. 343–348, Dec. 2019, doi: 10.1016/j.eswa.2019.07.014.
- [41] H. Shum, X. He, and D. Li, "From Eliza to Xiaolce: challenges and opportunities with social chatbots," *Front. Inf. Technol. Electron. Eng.*, vol. 19, no. 1, pp. 10–26, Jan. 2018, doi: 10.1631/FITEE.1700826.
- [42] M. Alfonseca, "¿Basta la prueba de Turing para definir la 'inteligencia artificial'?", *Sci. Fides*, vol. 2, no. 2, p. 129, Nov. 2014, doi: 10.12775/SetF.2014.018.
- [43] M. Romero, C. Casadevante, and H. Montoro, "Cómo construir un psicólogo-chatbot," *Papeles del Psicólogo - Psychol. Pap.*, vol. 41, no. 1, 2020, doi: 10.23923/pap.psicol2020.2920.
- [44] L. S. G. Piccolo, S. Roberts, A. Iosif, and H. Alani, "Designing Chatbots for Crises: A Case Study Contrasting Potential and Reality," Jul. 2018, doi: 10.14236/ewic/HCI2018.56.
- [45] P. Cornejo, "Arquitectura de un Chatbot. Al comenzar a desarrollar un Chatbot es... | by Pat Cornejo | Medium," Apr. 03, 2018. <https://medium.com/@patcornejo/arquitectura-de-un-chatbot-cb2d1c5f86c7> (accessed Apr. 21, 2021).
- [46] A. D. Escobar Macías, "Análisis del uso del procesamiento del lenguaje natural y su aplicación en sistemas conversacionales," UNIVERSIDAD DE GUAYAQUIL, 2019.
- [47] G. Bonales, N. Pradilla, and E. Martínez, "Chatbot como herramienta comunicativa durante la crisis sanitaria COVID-19 en España," *ComHumanitas Rev. científica Comun.*, vol. 11, no. 3, pp. 1–22, Jan. 2021, doi: 10.31207/rch.v11i3.270.
- [48] F. Yang and J. E. Menard, "PyISOLVER—A Fast Python OOP Implementation of LRDFIT

Model,” *IEEE Trans. Plasma Sci.*, vol. 48, no. 6, pp. 1793–1798, Jun. 2020, doi: 10.1109/TPS.2019.2958001.

- [49] J. M. Monteiro, J. McGibbon, and R. Caballero, “symp1 (v. 0.4.0) and climt (v. 0.15.3) – towards a flexible framework for building model hierarchies in Python,” *Geosci. Model Dev.*, vol. 11, no. 9, pp. 3781–3794, Sep. 2018, doi: 10.5194/gmd-11-3781-2018.
- [50] A. Zerouali, T. Mens, and C. De Roover, “On the usage of JavaScript, Python and Ruby packages in Docker Hub images,” *Sci. Comput. Program.*, vol. 207, p. 102653, Jul. 2021, doi: 10.1016/j.scico.2021.102653.
- [51] Microsoft, “IDE de Visual Studio 2019: software de programación para Windows,” 2021. <https://visualstudio.microsoft.com/es/vs/> (accessed Apr. 21, 2021).
- [52] P. Meyer, “En tecnología conversacional, la Precisión y el Recall no son (en verdad) tus mejores amigos | by Patrick Meyer | Planeta Chatbot : todo sobre los Chat bots, Voice apps e Inteligencia Artificial.” <https://planetachatbot.com/tecnologia-conversacional-precision-y-recall-4b8961b8f580> (accessed Apr. 21, 2021).

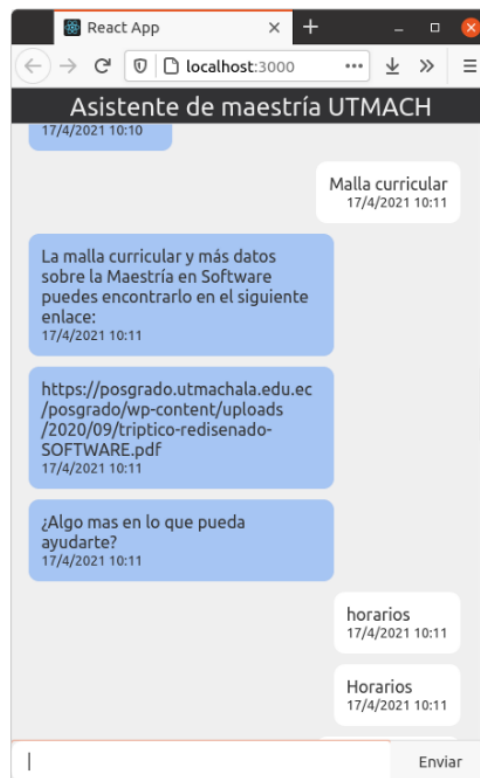
5. Anexos

Ilustración 39: Búsqueda de maestría por área



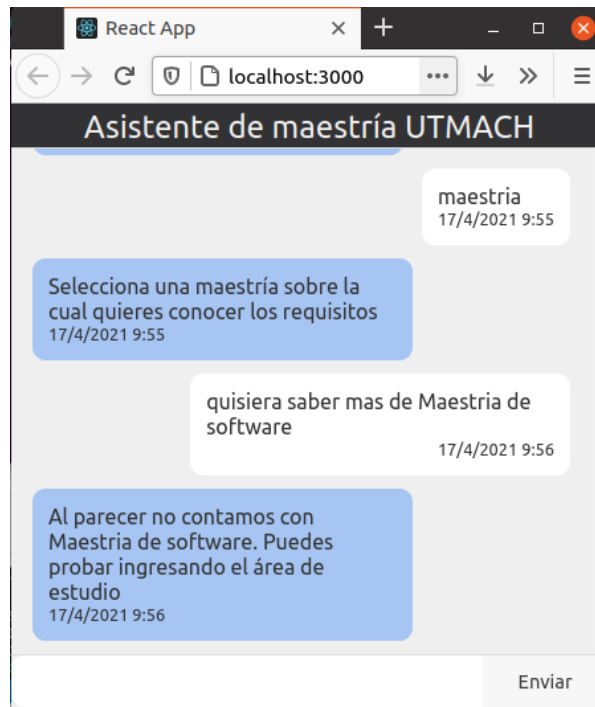
Fuente: Autor

Ilustración 40: Error 1 del Chatbot



Fuente: Autor

Ilustración 41: Error N° 2 del chatbot



Fuente: Autor