



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Ingeniería en Sistemas

Framework para el despliegue de temáticas de asistentes virtuales aplicado a la
Carrera de Ingeniería en Sistemas.

AUTOR:

Francisco Carmelino Agreda Sánchez

Jorge Luis Auquilla Villamagua

DIRECTOR:

Ing. Roberth Gustavo Figueroa Díaz M.Sc.

Loja - Ecuador

2022

CERTIFICACIÓN DE TESIS

Loja, .. de ... de 20... **Esta fecha puede ser diferente a la entrega en biblioteca**

Dr, Dra, Ing., Lic, Med, Odontólogo/a, Psicólogo/a **Nombres completos del/la Director/a.**
Esp., Mg. Sc. o PhD.

DIRECTOR/A DE TESIS/ TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración de tesis de grado titulado:
"Nombre de la tesis" de autoría del/la estudiante _____, previa a la obtención
del título de _____, una vez que el trabajo cumple con todos los requisitos exigidos por la
Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva
sustentación y defensa.

Nombre profesor

DIRECTOR/A DE TESIS

Autoría

Yo, _____, declaro ser autor/a del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido de la misma. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi tesis en el Repositorio Digital Institucional – Biblioteca Virtual.

Cédula de Identidad:

Fecha: **actualizada al mes y año de entrega en la biblioteca.**

Correo electrónico: correo_institucional@unl.edu.ec

Teléfono o Celular:

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR (A) PARA LA CONSULTA DE PRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DE TEXTO COMPLETO

Yo _____ declaro ser autor(a) de la tesis titulada _____ como requisito para optar el título de _____ autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RI, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los _____ días del _____ (mes) _____ del _____ (año en letras) _____

Firma: _____

Autor: _____

Cédula: _____

Dirección: _____ Correo electrónico: _____

Teléfono: _____ Celular: _____

DATOS COPLEMENTARIOS

Director de Tesis: _____

Tribunal de Grado: _____

Dedicatoria

Redacción sin sangría y justificado.

Nombres y apellidos del autor/a

Agradecimiento

Redacción sin sangría y justificado.

Nombres y apellidos del autor/a

ÍNDICE DE CONTENIDOS

Título	1
Resumen	2
2.1. Abstract	3
Introducción	4
Marco Teórico	6
4.1. Framework	6
4.1.1. Clasificación de los frameworks	6
4.1.2. Desarrollo de un framework	7
4.1.3. Ventajas de un framework	8
4.2. Chatbots o agentes conversacionales	8
4.2.1. Arquitectura de un chatbot	9
4.2.2. Fases de un Chatbot	10
4.3. Lingüística	10
4.4. Redes Neuronales Artificiales	11
4.5. Redes bayesianas	11
4.6. Procesamiento de Lenguaje Natural	12
4.7. Comprensión del lenguaje natural	13
4.8. Machine Learning	13
4.9. Frameworks de Chatbot	14
4.9.1. BERT	14
4.9.2. Microsoft Bot Framework	16
4.9.3. RASA	17
4.9.4. Wit AI	18
4.9.5. Dialog Flow	20
4.9.6. IBM Watson	20
4.9.7. Amazon Lex	22
4.9.8. Pandorabots	22
4.9.9. BotPress	24
4.9.10. Botkit	25
4.9.11. ChatterBot	25
4.10 Métodos	26
4.10.1. Método científico	26
4.10.2. Revisión Sistemática de Literatura	27
4.11. Trabajos relacionados	28

Metodología.	29
5.1. Área de estudio	30
5.2. Procedimiento	30
5.2.1. Fase 1: Estudiar las tecnologías para la construcción del framework de asistentes virtuales: Se realizó con el fin de obtener un estudio preliminar de todos los aspectos que constituyen el desarrollo e implementación de un framework de asistentes virtuales.	31
5.2.1.1. Desarrollo de estado de Arte.	31
5.2.1.2. Revisión de los trabajos relacionados.	31
5.2.1.3. Clasificación de la información.	32
5.2.1.4. Documentación de cada aspecto investigativo.	33
5.2.2. Fase 2: Construir el framework de despliegue de asistentes virtuales en función de las tecnologías vigentes seleccionadas: Se realizó con el fin de obtener una arquitectura, estructura y producto bien desarrollado para el despliegue de asistentes virtuales.	33
5.2.2.1 Establecimiento del Corpus lingüístico del framework.	33
5.2.2.2 Codificación del framework.	33
5.2.2.3 Documentación técnica de cada aspecto desarrollado en las distintas etapas de la construcción del framework.	34
5.2.3. Fase 3: Validar el framework mediante el despliegue de información académica relacionada a las temáticas dentro de la CIS: Se realizó con el fin de validar la correcta funcionalidad del framework a partir de tematicas relacionadas a la carrera de la CIS.	34
5.2.3.1. Identificación y recopilación de la fuente de información de las tematicas de la Carrera de CIS.	34
5.2.3.2. Evaluación del framework a partir de la fuente de información	34
5.2.3.3. Corrección y validación del framework.	34
5.2.3.4 Documentación de los resultados obtenidos de la validación del framework.	34
5.3. Materiales	34
5.3.1. Recursos de Hardware y Software	34
5.3.2. Participantes	35
5.3.3 Métodos	36
Resultados.	37
6.1. Primera fase: Estudiar las tecnologías para la construcción del framework de asistentes virtuales.	37
6.1.1. Desarrollo de estado del Arte.	38
6.1.1.1. Revisión bibliográfica y literaria de autores en el ámbito de (Machine Learning) y procesamiento de lenguaje natural.	38
6.1.1.2. Revisión sistemática de autores en el ámbito de (Machine Learning) y procesamiento de lenguaje natural.	39

6.1.2. Revisión de trabajos relacionados.	42
6.1.2.1. Revisión bibliográfica y literaria de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales.	42
6.1.2.2. Revisión sistemática de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales.	43
6.1.3. Clasificación de la información.	47
6.1.3.1. Comparación de las mejores tecnologías para el desarrollo de un framework de asistentes virtuales.	48
6.1.3.2. Selección de la tecnología para el desarrollo del framework.	51
6.1.4. Documentación de cada aspecto investigativo.	51
6.2. Segunda fase: Construir el framework de despliegue de asistentes virtuales en función de las tecnologías vigentes seleccionadas.	52
6.2.1. Establecimiento del Corpus lingüístico del framework.	52
6.2.1.1. Bert-base-multilingual-cased.	52
6.2.2. Codificación del framework.	54
6.2.2.1. Desarrollar el componente inteligente del Framework.	54
6.2.2.2. Desarrollar el backend del framework.	56
6.2.3. Documentación técnica de cada aspecto desarrollado en las distintas etapas de la construcción del framework.	62
6.3. Tercera fase: Validar el framework mediante el despliegue de información académica relacionada a las temáticas dentro de las CIS.	62
6.3.1. Identificación y recopilación de la fuente de información de las temáticas de la Carrera de CIS.	62
6.3.2. Evaluación del framework a partir de la fuente de información .	66
6.3.3. Corrección y validación del framework.	66
6.3.4. Documentación de los resultados obtenidos de la validación del framework.	66
Discusión.	66
7.1. Objetivo 1: Estudiar las tecnologías para la construcción del framework de asistentes virtuales.	66
7.2. Objetivo 2: Construir el framework de despliegue de asistentes virtuales en función de las tecnologías vigentes seleccionadas.	69
7.3. Objetivo 3: Validar el framework mediante el despliegue de información académica relacionada a las temáticas dentro de la CIS.	70
Conclusiones.	70
Recomendaciones	72
Bibliografía.	73
Anexos.	79
11.1 Anexo 1: Revisión Sistemática de Literatura.	79
11.2 Anexo 2: Revisión Sistemática de Literatura.	80

ÍNDICE DE FIGURAS

Figura 1. Ilustración de la arquitectura básica de un chatbot	9
Figura 2. Modelo Bert	15
Figura 3. Arquitectura del modelo Bert	15
Figura 4. Arquitectura Microsoft Bot Framework	17
Figura 5. Arquitectura de Rasa Framework	18
Figura 6. Representación de entidades e intenciones	19
Figura 7. Diagrama de Arquitectura de IBM Watson	21
Figura 8. Proceso de conversión de voz a texto Amazon Lex	22
Figura 9. Etiquetas AIML Pandorabots	23
Figura 10. Proceso de Información Botpress	25
Figura 11. Pasos del Método Científico	27
Figura 12. Fases de la RSL	28
Figura 13. Arquitectura General del Framework de despliegue de asistentes virtuales	51
Figura 14. Combinaciones de Hyperparameters	54
Figura 15. Configuración de la arquitectura DIET	55
Figura 16. Incrustación del hiper parámetro bert-base-multilingual-cased	55
Figura 17. Archivos creados al iniciar un proyecto de Rasa	56
Figura 18. Código de entrenamiento en Rasa	57
Figura 19. Código de las intenciones y entidades en Rasa	58
Figura 20. Código de las historias en Rasa	59

ÍNDICE DE TABLAS

Tabla 1. Recursos de Hardware y Software utilizados	
35	
Tabla 2. Participantes	36
Tabla 3. Resultados de los estudios seleccionados a partir de la pregunta planteada	38
Tabla 4. Criterios PICOC utilizados en la investigación	39
Tabla 5. Criterios de Inclusión y Exclusión	40
Tabla 6. Cadenas de búsqueda de cada base de datos	40
Tabla 7. Resultados del proceso de selección de estudios primarios	
Tabla 8. Artículos Evaluados	
Tabla 9. Tipo PLN por artículo	
Tabla 10. Herramientas por artículo	
Tabla 11. Resultados de los estudios seleccionados a partir de la pregunta planteada	
Tabla 12. Criterios PICOC utilizados en la investigación	
Tabla 13. Criterios de Inclusión y Exclusión	
Tabla 14. Cadenas de búsqueda de cada base de datos	
Tabla 15. Resultados del proceso de selección de estudios primarios.	
Tabla 16. Artículos Evaluados	
Tabla 17. Frameworks por artículo	
Tabla 18. Tecnologías por artículo	
Tabla 19. Arquitectura por artículo	
Tabla 20. Comparación de Frameworks para la construcción de Chatbots	48
Tabla 21. Conjunto de datos finales de intenciones para la fase de entrenamiento NLU	
Tabla 22. Análisis de las respuestas de la primera tarea de Revisión bibliográfica/literaria y Sistemática de Literatura.	

Tabla 23. Análisis de las respuestas de la segunda tarea de Revisión bibliográfica/literaria y Sistemática de Literatura.

ÍNDICE DE ANEXOS

ANEXOS

Anexo 1. Título del anexo.....	15
Anexo 1. Título del anexo.....	18

1. Título

Framework para el despliegue de temáticas de asistentes virtuales aplicado a la Carrera de Ingeniería en Sistemas.

Framework for the deployment of Chatbots themes applied to Degree in Systems Engineering.

2. Resumen

La Universidad Nacional Loja cuenta con un gran crecimiento en el ámbito de la creación de asistentes virtuales por parte de su comunidad estudiantil, sin embargo, el mismo colectivo universitario en especial los desarrolladores de estas áreas son los que contemplan que la

complejidad de desarrollo y procesamiento de lenguaje natural son los principales problemas que se enfrentan debido a la ausencia de un entorno de trabajo que de las facilidades de desarrollo de estos sistemas, por lo tanto, se muestra la oportunidad de exponer mediante un enfoque en inteligencia Artificial (IA) el potencial para sustentar la falta de organización en herramientas tecnológicas para el despliegue de diversidad temáticas de asistentes virtuales, además de lograr un impacto y contribución importante en el área de conocimiento, a través de la incursión de adoptar un modelo de framework.

El presente trabajo de titulación está orientado a: “Desarrollar un framework de procesamiento para el despliegue de temáticas de asistentes virtuales aplicado a la Carrera de Ingeniería en Sistemas”, tal que, el resultado esperado de los diferentes procesos desarrollados, sea una modelo base de arquitectura de framework que utilice tecnologías de procesamiento de lenguaje natural (PNL) e IA, para resaltar la importancia de estas áreas de conocimiento.

La investigación y conceptualizaciones englobados al desarrollo del framework así como los subprocesos adyacentes en la revisión de literatura son abordados y ejecutados mediante la Método de Revisión Sistemática de Literatura (RSL), para la fase de construcción e implementación se contempló emplear el método científico, conjuntamente con las distintas herramientas como RASA X, BERT y Transformers para el desarrollo de la arquitectura base del modelo de framework de despliegue de asistentes virtuales.

El principal aporte del Trabajo de Titulación TT es el desarrollo de un framework de procesamiento de asistentes virtuales que abarquen la reusabilidad de código, ahorro de tiempo e incentivo a continuar en esta línea de investigación. Dentro de las líneas futuras se destaca la contribución de desarrolladores para robustecer la arquitectura para un mayor alcance, así como la aplicación de lecturas de voz en el ingreso de consultas al asistente virtual.

2.1. Abstract

The National University of Loja has a great growth in the field of the creation of virtual assistants by its student community, however, the same university collective, especially the developers of these areas are those who contemplate that the complexity of development

and processing of natural language are the main problems faced due to the absence of a working environment that provides the facilities for the development of these systems, Therefore, it is shown the opportunity to expose through a focus on Artificial Intelligence (AI) the potential to support the lack of organisation in technological tools for the deployment of thematic diversity of virtual assistants, in addition to achieving a significant impact and contribution in the area of knowledge, through the incursion of adopting a framework model.

This degree project is aimed at: "Developing a processing framework for the deployment of virtual assistant themes applied to the Systems Engineering Degree", such that the expected result of the different processes developed is a framework architecture base model that uses natural language processing (NLP) and AI technologies to highlight the importance of these areas of knowledge.

The research and conceptualisations involved in the development of the framework as well as the adjacent sub-processes in the literature review are approached and executed using the Systematic Literature Review (SLR) method. For the construction and implementation phase, the scientific method was used, together with the various tools such as RASA X, BERT and Transformers for the development of the base architecture of the framework model for the deployment of virtual assistants.

The main contribution of the TT Degree Project is the development of a framework for processing virtual assistants that includes code reusability, time saving and an incentive to continue in this line of research. Future lines of research include the contribution of developers to strengthen the architecture for a wider scope, as well as the application of voice readings in the input of queries to the virtual assistant.

3. Introducción

El presente trabajo de titulación (TT) se enmarca en la temática central Framework para el despliegue de temáticas de asistentes virtuales aplicado a la Carrera de Ingeniería en Sistemas, la incorporación del desarrollo de sistemas de procesamiento de lenguaje natural (PNL) [72], junto a la creciente popularidad de los chatbots como asistentes virtuales ha

llevado a muchas organizaciones a lanzar frameworks para diseñar dichos asistentes virtuales. Sin embargo, estos marcos suelen dar lugar a asistentes virtuales inflexibles y difíciles de mantener [71]. Según [73] un marco de trabajo es una estructura real o conceptual destinada a apoyar o guiar la construcción de algo que amplía la estructura en algo útil y proporciona un modelo extensible para un tipo específico de aplicaciones. Mientras que, el procesamiento de lenguaje natural es el campo de conocimiento de la Inteligencia Artificial que se ocupa de investigar la manera de comunicar las máquinas con las personas mediante el uso de lenguas naturales, como el español, el inglés o el chino [74]. Motivo para proveer a la comunidad de estudiantil de un marco de trabajo que permita el procesamiento y despliegue de temáticas de asistentes virtuales, que sea capaz de aportar un valor añadido al ámbito técnico - académico, así como en la ciencia informática, por lo que se hace necesario proporcionar modelos al estado de la técnica que permitan el desarrollo de asistentes virtuales.

Se abarcan distintas conceptualizaciones de los métodos, técnicas, tecnologías que engloban alrededor del desarrollo de Frameworks, describiendo diferentes herramientas acordes a trabajos relacionados, que de un aporte significativo en la elaboración de una arquitectura base. Con todo lo antes mencionado se logró cumplir con los objetivos que contemplan “ Estudiar las tecnologías para la construcción del framework de asistentes virtuales “, así como especificar un modelo framework para posteriormente evaluar el modelo de framework en temáticas de la carrera de ingeniería en sistemas.

Cabe mencionar que el desarrollo del framework es la reusabilidad de código, que aporta un ahorro de tiempo importante en la implementación de experimentos, y que incentiva a continuar esta línea de investigación. Con la utilización del framework, se decrementa la cantidad de código que el desarrollador tiene que escribir, testear y depurar [75]. Esto también puede bajar los costos de mantenimiento y acelerar la puesta en producción de una aplicación o la puesta en ejecución de experimentos por parte de un investigador en el dominio [76]. Otro factor relevante es el desarrollo de esta investigación para motivar el progreso de este tipo de herramientas y que puede presentar una gran mejora para la carrera de CIS, el resultado podría animar a los estudiantes a construir nuevos asistentes virtuales que estimulen la posibilidad de crear un entorno virtual capaz de satisfacer a los

usuarios en sus consultas, poder producir y compartir nuevos conocimientos a los que muy probablemente no se podría haber tenido acceso.

El documento desarrollado dispone de secciones de Revisión literatura como de Revisión Sistemática de literatura con el preámbulo de los conceptos de Chatbot, frameworks y procesamiento de lenguaje natural (PNL), donde se describen las herramientas, arquitectura, modelos, tecnologías y trabajos relacionados con estas principales temáticas; sección de Materiales y Métodos donde se describen el contexto de la metodología tratada, los recursos empleados, los participantes involucrados; seguido de la sección de Resultados generado de acuerdo a los objetivos planteados; sección de Discusión se aborda el análisis de los datos concretados en los resultados de acuerdo a la investigación, desarrollo y pruebas del modelo creado; sección de Conclusiones donde se presentan los puntos concretos basados en la experimentación y análisis de datos con relación a los objetivos del TT; sección de Recomendaciones donde se proyecta la descripción de los puntos finales de acuerdo a la experiencia desarrollada; sección de bibliografía donde se manifiesta la información bibliográfica encontrada en el transcurso de desarrollo del TT, I igual que la información complementaria en Anexos

4. Marco Teórico

En esta sección se realiza la recopilación de información bibliográfica importante que apoya el desarrollo del TT, así como los conceptos que permiten una mejor comprensión del tema. Se inicia con una breve introducción a las definiciones generales del Framework, se hace hincapié en los Chatbots o agentes conversacionales, también se destaca la lingüística y los algoritmos de extracción y clasificación. Se han revisado documentos obtenidos de fuentes fiables sobre marcos de trabajo para chatbots, todo ello con el objetivo de comparar diferentes marcos de trabajo para chatbots e identificar características que ayudan al

programador a desarrollar chatbots de forma más rápida en función del número de características proporcionadas por el marco de trabajo.

4.1. Framework

Un framework se define como un esquema de reutilización del software conformado por componentes y sus relaciones, por ejemplo: abstracción de clases , objetos o componentes, además de proveer distintos componentes de conexión a base de datos [1]. “Un framework integra una funcionalidad avanzada a un lenguaje de programación, también automatiza muchos de los patrones de programación para dirigirlos a un determinado objetivo, proporcionando una estructura al código, lo mejora y lo hace más entendible y sostenible, y permite que la aplicación se divida en capas [2]”. Diseñar un aplicación basada en un framework implementa una instancia del mismo, es decir, rellena los huecos que faltan al framework para ser una aplicación concreta, el mismo soporta la estructura general de cualquier aplicación del dominio al que se adscribe, a modo de esqueleto, y son esos huecos los que aportan la flexibilidad requerida para ajustar una aplicación a intereses concretos [3].

4.1.1. Clasificación de los frameworks

Los frameworks pueden brindar distintas funcionalidades, los cuales pueden ser de bajo nivel o de alto nivel, como aquellos que se encargan de proporcionar un área de trabajo para satisfacer las necesidades del usuario.

- Frameworks de caja blanca o frameworks dirigidos por la arquitectura

El framework de caja blanca toma las características de la orientación a objetos, tales como la ligadura dinámica o la herencia, estos definen interfaces que pueden integrarse mediante composición de objetos. La dificultad del uso de este tipo de frameworks radica en el profundo conocimiento y comprensión de clases a extender [53].

- Frameworks de caja negra o frameworks basados en datos

El framework de caja negra se estructura utilizando una composición de objetos y delegación de herencias, resaltan la relaciones dinámicas entre objetos en lugar de la relaciones estáticas entre clases. El usuario del framework no tiene que conocer a

profundidad los detalles del mismo, sino sólo conocer cómo usar y combinar los objetos existentes [4].

- **Frameworks de caja gris**

El framework de caja gris es una combinación de los dos anteriores, permiten extensión mediante herencias y la ligadura dinámica, además de permite ocultar información innecesaria del framework al usuario [4].

4.1.2. Desarrollo de un framework

El desarrollo y ciclo de vida del framework es un proceso iterativo, no secuencial, el cual se apoya en las metodologías de ingeniería de software, estas constan en tres etapas repetitivas [5], pues el dominio del problema a menudo es cambiante, por lo tanto las etapas son una orientación para comprobar la evolución y el desarrollo de un framework.

- **Etapas 1: Análisis**

Esta primera etapa procura realizar un análisis del dominio de la aplicación, requisitos y estudio de la viabilidad del proyecto.

- **Etapas 2: Diseño**

Esta segunda etapa define las abstracciones genéricas dentro del dominio, que conforman el conjunto de abstracciones del framework.

- **Etapas 3: Instanciación**

Esta tercera etapa supone una instancia del framework dependiendo si es de caja blanca (se realizará a través de nuevas clases, que hereden de las abstracciones de las ranuras del sistema) o caja negra (se realizará a través de scripts de configuración).

4.1.3. Ventajas de un framework

El uso de un framework en el desarrollo de un proyecto siempre ofrece importantes ventajas, no solo en facilitar la creación de aplicaciones, sino en el mantenimiento del código y la ampliación del mismo.

- **Patrones de diseño**

Una de las principales ventajas que ofrece es el uso de los patrones de diseño, estos proveen un esquema para refinar componentes de un sistema de software y la forma

como se relacionan entre sí, también definen la estructura de comunicación de los componentes en un contexto particular. Tienen un nivel de abstracción menor en comparación con los patrones arquitectónicos, lo que los hace independientes del lenguaje o paradigma de programación. Los patrones de diseño son soluciones bien documentadas por lo cual son usados para dar solución a nuevos problemas similares [6].

- **Estructura predefinida de la aplicación**

El programador no necesita plantearse una estructura global, ya que esta es proporcionada por el propio framework.

- **Código altamente testeado**

Todo el código que conforma el framework está probado y garantiza un funcionamiento.

4.2. Chatbots o agentes conversacionales

Un chatbot o agente conversacional es un sistema de software que puede interactuar con un usuario mediante el uso del lenguaje natural. Un chatbot no está sujeto no solo a mensajes de texto, si no a una gama de contenido multimedia que permite una mejor interacción con el usuario. Para una mejor interpretación del lenguaje natural un chatbot, en la actualidad, utiliza técnicas de procesamiento del lenguaje natural [7].

Mediante el uso de computación cognitiva, a través del uso de inteligencia artificial, un chatbot entiende lo que el usuario está intentando decir y responde con un mensaje coherente, relevante y directo relacionado con la tarea o petición que el usuario está solicitando [8].

Además, los agentes para mejorar las respuestas al usuario se pueden conectar con aplicaciones externas que responden a las peticiones que el bot haga. Las características propias que contiene un chatbot lo convierten en una especie de sistema experto, que basado en el conocimiento que contiene simula un diálogo inteligente con el usuario [9].

4.2.1. Arquitectura de un chatbot

La arquitectura básica de un chatbot se divide en tres componentes [10], como se observa en la Figura 1.

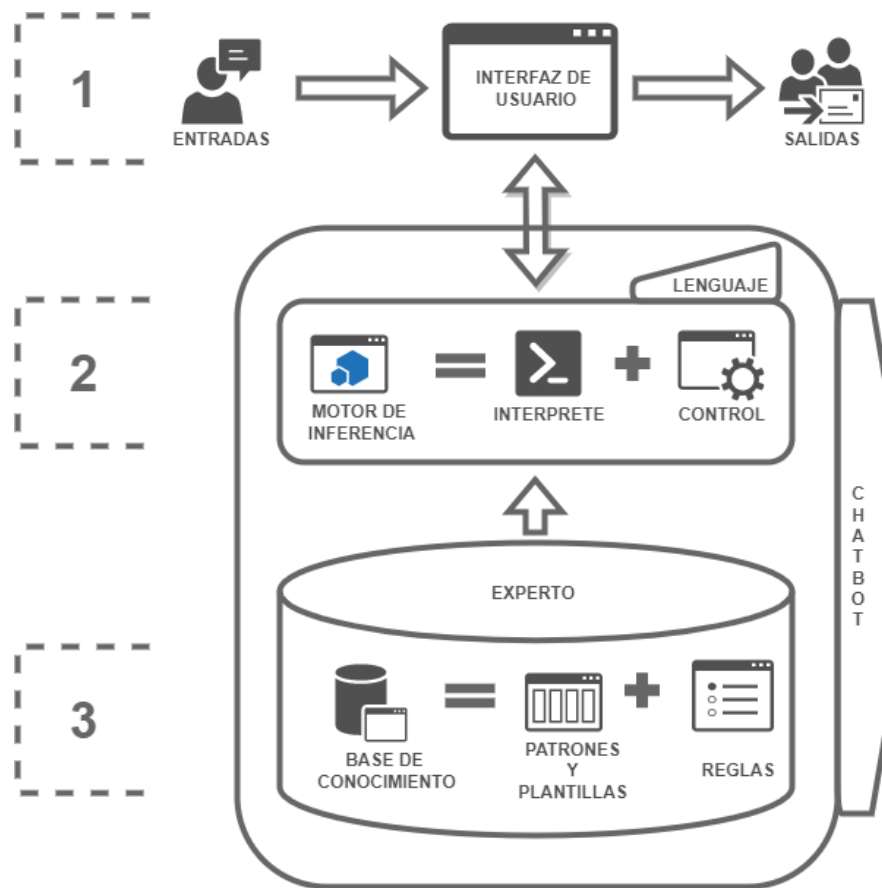


Figura 1. Ilustración de la arquitectura básica de un chatbot.

1. Interfaz de Usuario: es el medio por el cual el usuario envía las entradas de información hacia el chatbot.
2. Motor de inferencia: analiza la información, establece los objetivos y obtiene la respuesta de acuerdo con la base de conocimiento, enviándola de vuelta a la interfaz de usuario. Un motor de inferencia utiliza dos elementos: los datos (hechos o evidencias) y el conocimiento (conjunto de reglas almacenado en la base de conocimiento) para obtener nuevas conclusiones o hechos.
3. La Base de conocimiento: contiene todo el conocimiento del experto humano, en función de plantillas, patrones y reglas.

4.2.2. Fases de un Chatbot

Para el desarrollo de un chatbot o asistente virtual en particular, se utiliza múltiples fases [11] que son las siguientes:

- **Fase de Pre-Procesamiento:** Aplicar una función de preprocesamiento al texto de entrada para estandarizar la entrada de acuerdo con los requisitos del sistema. Dependiendo de las palabras clave utilizadas en el texto, se reconoce un contexto apropiado.
- **Fase de procesamiento y extracción:** Procesar el texto de entrada para extraer las palabras clave requeridas donde la información solicitada por un usuario es entendida y proporcionada (respuesta) desde la base de conocimiento.
- **Fase de respuesta:** Si la respuesta está disponible, la respuesta se brinda a través de un módulo en particular y se envía al usuario.

4.3. Lingüística

La lingüística se realiza en base a una estructura estratificada, la cual comienza con unidades de lenguaje básicas, las que a su vez aumentan su complejidad lingüística a medida que se asciende a los niveles superiores de una estructura estratificada, donde cada capa corresponde a un nivel lingüístico distinto [12]. De este modo, los diferentes sonidos utilizados en el lenguaje se pueden encasillar en el nivel Fonológico. El estudio de los sistemas escritos se pueden asociar a un nivel que contempla la Ortografía. La Morfología, se encarga de la estructura de las palabras donde se considera la forma de estas y las inflexiones que puedan presentar.

La Sintaxis describe el ordenamiento de las palabras y cómo pueden ser combinadas para formar estructuras más complejas como frases y sentencias. La Semántica se encarga de analizar el significado de las palabras individuales y de éstas en unidades más complejas como frases y sentencias, recibiendo el nombre de Semántica Composicional. El nivel Pragmático se encarga de analizar cómo las palabras y frases se relacionan en un contexto. Y por último, el nivel de discurso corresponde a la forma en que las personas y las cosas son presentadas en forma de temas y subsecuentemente nombradas en declaraciones [13].

Por otro lado, en base a la literatura existente, se tiene que los niveles pueden variar ya que su inclusión dependerá del grado de profundidad en aspectos lingüísticos que el autor haya decidido incorporar en su obra, así como también, limitar la incorporación de sólo aquellas capas que tienen relación con el alcance de los contenidos tratados. Además, pueden existir algunas sutiles diferencias en los nombres asignados a éstos, pero el sentido de fondo es el mismo [14], [15].

4.4. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) se basan en el funcionamiento del cerebro humano en el cual la neurona es el elemento fundamental [16]. “Las RNA intentan imitar el comportamiento del cerebro humano, que se caracteriza por aprender a través de la experiencia y por abstraer conocimientos genéricos a partir de un conjunto de datos [17]”.

Mediante la abstracción del funcionamiento de las estructuras neuronales biológicas del sistema nervioso central es cómo se crearon las RNA [18]. En las cuales se plantean una serie de entradas (capa de entrada) que se encuentran conectadas con otras una capa interior denominada capa oculta. En función del diseño de la RNA puede darse el caso de que se encuentren más capas ocultas. La o las capas ocultas al final deben proveer una salida la cual se genera en la capa de salida.

4.5. Redes bayesianas

Las redes bayesianas (RB) son una representación gráfica de dependencias para razonamiento probabilístico, en la cual los nodos representan variables aleatorias y los arcos representan relaciones de dependencia directa entre las variables. En un RB, las relaciones de independencia condicional que se muestran en un grafo corresponden a relaciones de independencia en la distribución de probabilidad. Dichas independencias simplifican la representación del conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades). Una red bayesiana representa en forma gráfica las dependencias e independencias entre variables aleatorias, en particular las independencias condición [19].

Según [20] “define a una red bayesiana como un conjunto de variables codificadas probabilísticamente, una estructura gráfica que conecta esas variables en términos de

relaciones de independencia condicional susceptibles de ser modificadas con base en evidencias por medio del teorema de Bayes". Sin embargo, la descripción más frecuente de una red bayesiana se basa en dos elementos, una dimensión cualitativa y otra cuantitativa, los fundamentos teóricos de estos elementos se fundamentan en dos grandes pilares, uno la teoría de grafos y la teoría de la probabilidad [21].

4.6. Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (PLN) consiste en una serie de métodos y procedimientos computacionales cuyo fin principal es tomar como entrada el lenguaje humano, el cual puede haber sido voz, texto o imágenes, para que puedan ser leídos y procesados por un ordenador o cualquier dispositivo digital que tenga las funciones de un procesador de computadora, y entonces cumplir una función en particular. El PLN siempre ha estado unido a la relación humano-computador, ya sea porque necesitan "hablarse" o "entenderse" entre ambos [22]. Se trata de una disciplina de la informática siempre en evolución y que se vuelve cada vez más compleja debido a las mayores exigencias del sector.

El PLN comenzó en la década de 1950 como la intersección de la inteligencia artificial y la lingüística [23]. Desde entonces diversas teorías han ayudado a la evolución del PLN, como la notación Backus-Naur Form (BNF) y herramientas como analizadores léxico y sintáctico. El estado del arte revela que el estudio del PLN está involucrado con la conjunción de técnicas como: autómatas de estado finito y de árbol, gramática libre del contexto y etiquetado de parte del discurso [24], por mencionar algunos. Una de las áreas donde el PLN está colaborando y siendo más útil es en la ciencia, medicina y biomedicina.

Un estudio del término PLN en los localizadores de bases de datos científicas permite encontrar numerosos artículos publicados en revistas no relacionadas con la informática, lo que da fe de la enorme participación de las PLN en estos campos. Se pueden encontrar otros campos de aplicación, por ejemplo, en los sistemas de información geográfica [25].

4.7. Comprensión del lenguaje natural

La Comprensión del Lenguaje Natural (NLU) o sus siglas en inglés "Natural Language Understanding" son el núcleo de cualquier tarea de PNL. Es una técnica para implementar

interfaces de usuario naturales como un chatbot. NLU tiene como objetivo extraer el contexto y los significados de las entradas del usuario del lenguaje natural, que pueden no estar estructuradas y responder adecuadamente de acuerdo con la intención del usuario. Identifica la intención del usuario y extrae entidades específicas del dominio. Más específicamente, la intención representa un mapeo entre lo que dice un usuario y qué acción debe tomar el chatbot. Las acciones corresponden a los pasos que tomará el chatbot cuando las entradas del usuario activen intenciones específicas y pueden tener parámetros para especificar información detallada al respecto. La detección de intenciones generalmente se formula como una clasificación de oraciones en la que se predicen etiquetas de intención únicas o múltiples para cada oración [26].

4.8. Machine Learning

El Machine Learning (ML) es una subárea de la Inteligencia Artificial IA y se diferencia de esta en que tiene un mayor acceso a datos y los algoritmos de ML aprenden de estos. El ML es un área de rápida evolución, el cual su principal actividad es el diseño y análisis de algoritmos que le permitan a las computadoras aprender [27]. Por su parte, [28] también concuerdan en el hecho de que ML se basa en emplear algoritmos para analizar grandes conjuntos de datos y además agregan que el potencial del ML es que puede realizar analítica predictiva mucho más rápido que cualquier humano, permitiendo trabajar de forma más eficaz.

Los algoritmos de ML se diferencian de otros paradigmas de programación tradicionales como la programación funcional o la programación orientada a objetos en que estos últimos están más orientados a tener una interacción con el usuario, o a leer datos de entrada de un determinado puerto periférico para realizar una serie de acciones que se suceden. Mientras que en el ML no se contempla tener una interacción de ningún tipo, por el contrario, mientras que en los paradigmas tradicionales se busca obtener resultados, en este modelo se basa en los resultados para generar un modelo de predicción o clasificación, que luego puede predecir resultados a partir de nuevos datos de entrenamiento [29].

Generalmente, un algoritmo de ML se compone de las fases de recogida de datos, preprocesamiento, aprendizaje (modelado) y evaluación, donde el entrenamiento se realiza proporcionando datos que pueden estar etiquetados o no, lo que determina en primer lugar

el tipo de aprendizaje utilizado. (supervisado, no supervisado, etc.), así como los clasificadores a aplicar. Durante la fase de prueba, se trabaja con datos que el clasificador no había "visto" para generar predicciones sobre estos datos a partir del modelo construido en la fase anterior. Algunas de las aplicaciones típicas del ML son: el reconocimiento de imágenes, la minería de datos o la robótica [30].

4.9. Frameworks de Chatbot

Los frameworks de chatbot ayudan a crear, conectar, publicar y gestionar chatbots inteligentes e interactivos para ofrecer la mejor experiencia de usuario. Ayudan a crear fácilmente nuevos chatbots. Los marcos de chatbot que se identificaron son: Bert, Microsoft Bot Framework, Rasa, Wit AI, Dialogflow, IBM Watson, Amazon Lex , Pandorabots, Botpress, Botkit, ChatterBot. Todas estas plataformas se basan en algoritmos de aprendizaje automático. La principal operación que realizan los frameworks es identificar la intención y la entidad. Una intención representa una coincidencia entre lo que dice un usuario y la acción que debe realizar el chatbot. Representa una parte de la conversación. Una entidad, por otro lado, es una herramienta para extraer valores de parámetros de la entrada del lenguaje natural. Cualquier palabra clave importante que se quiera obtener de la petición de un usuario se vincularía a una entidad. Por ejemplo, "¿Reservar un vuelo a París?" Es la entrada de la intención de solicitar reservar un vuelo, mientras que -París" es un individuo de una posible entidad llamada Ciudad que recoge los nombres de cualquier ciudad del mundo.

4.9.1. BERT

BERT (Bidirectional Encoder Representations from Transformers) es un modelo de procesamiento de lenguaje natural propuesto por investigadores de Google Research en 2018. Este modelo alcanza una precisión de vanguardia en muchas tareas de PNL y NLU, BERT lee toda la secuencia de palabras de manera bidireccional, lo que le permite al modelo aprender el contexto de una palabra en función de todo su entorno, tanto a la izquierda, como a la derecha de la misma. Bert tiene dos pasos principales: el entrenamiento previo o pre-training y fine tuning, como se muestra en la Figura 2. Durante el entrenamiento previo, el modelo se entrena con datos sin etiquetar en diferentes tareas. Para el fine tuning, BERT se inicializa primero con los parámetros entrenamiento previo, y estos parámetros se ajustan con datos etiquetados de tareas posteriores. En cada tarea posterior se crean modelos

separados, pero siempre se inicializa con los valores del entrenamiento previo [31].

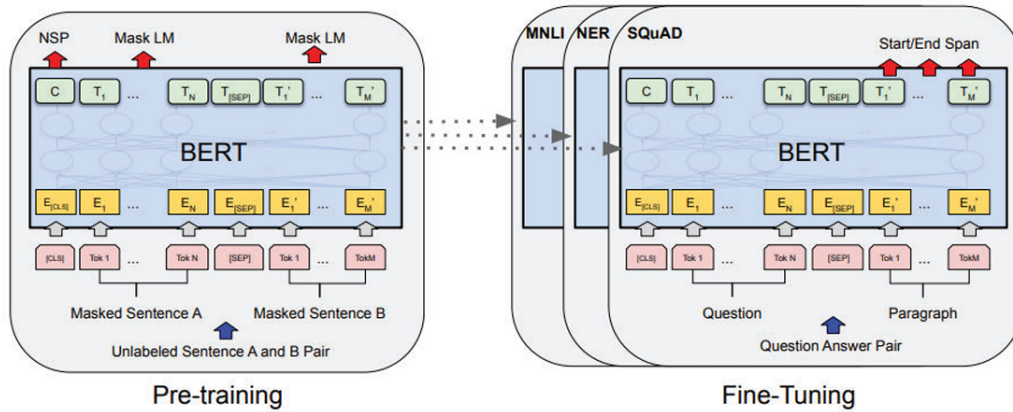


Figura 2. Modelo Bert.

BERT se presenta en dos tamaños BERT BASE y BERT LARGE. El modelo BASE se utiliza para medir el rendimiento de la arquitectura en comparación con otras arquitecturas y el modelo LARGE produce resultados de última generación que se recogen en el documento de investigación. A medida que aumenta el número de capas BERT, también aumenta el número de parámetros (pesos) y el número de cabezas de atención. BERT BASE tiene un total de 12 cabezas de atención (permite que cada token se centre en otros tokens) y 110 millones de parámetros. Mientras que BERT LARGE tiene 16 cabezas de atención con 340 millones de parámetros. BERT BASE tiene 768 capas ocultas mientras que el BERT LARGE tiene 1024 capas ocultas [31].

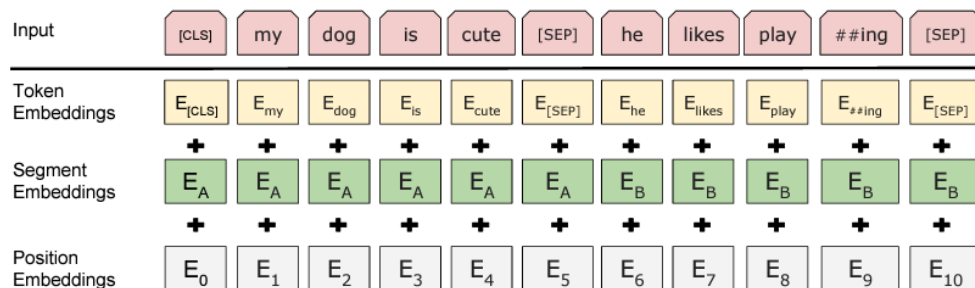


Figura 3. Arquitectura del modelo Bert.

4.9.2. Microsoft Bot Framework

Microsoft Bot Framework es una plataforma que ofrece una serie de plantillas para la creación de bots. Estas plantillas van desde un servicio básico de recopilación de datos (FormBot) hasta el bot de comprensión del lenguaje más avanzado que aprovecha el paquete NLU de Microsoft para impulsar experiencias de conversación inteligentes. Dicho esto, Microsoft Bot Framework ciertamente no es para la creación de bots ocasionales, ya que requiere experiencia en programación para implementarse de manera efectiva [32].

Microsoft Bot Framework tiene dos partes fundamentales que lo convierten en una gran herramienta para construir chatbots, son:

- **El SDK de Bot Framework:** El SDK de Bot Framework es el conjunto de librerías de lenguaje (Microsoft.Bot.Builder, Connector, Configuration, etc.) que nos ayudarán a construir el chatbot con un mínimo esfuerzo. Los SDKs están disponibles en las plataformas .NET y Node.js. El SDK. El portal del marco de bots es el panel de administración para registrar y gestionar los chatbots. El portal nos permite configurar el chatbot para que funcione con múltiples canales o aplicaciones de mensajería [32].
- **El Portal de Bot Framework:** El portal del marco del bot admite actualmente los siguientes canales: Facebook, GroupMe, Kik, LINE, Microsoft Teams, Skype, Skype for Business, Slack, Telegram, Twilio, WeChat, Email, Web Chat y Cortana. La parte más interesante del marco es que soporta construir bots a través de múltiples canales usando el mismo código base. Lo que esto significa es que cuando un mensaje llega desde Facebook, se ve exactamente igual que cuando viene de slack. De esta manera, no tenemos que tener en cuenta tantas diferentes variaciones de datos, lo que hace nuestra vida mucho mucho más fácil [32].

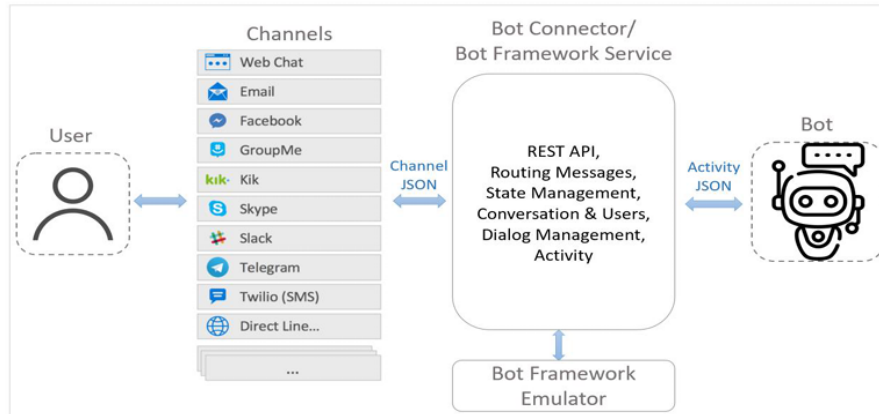


Figura 4. Arquitectura Microsoft Bot Framework.

4.9.3. RASA

Basado en Python, RASA es una pila de tecnología NLP de código abierto que ayuda a los desarrolladores a crear chatbots altamente inteligentes utilizando modelos probabilísticos en lugar del enfoque habitual basado en diagramas de flujo. En lugar de depender de estructuras lógicas si/entonces, RASA implementa modelos de aprendizaje automático no sólo para reconocer identidades e intenciones, sino también para determinar el flujo de la conversación. Esto convierte a RASA en el sueño de los desarrolladores de chatbots cuando crean experiencias de conversación altamente sofisticadas que evolucionan continuamente sin necesidad de intervención manual [33].

Desafortunadamente, la misma riqueza y complejidad lo hacen inadecuado para usuarios no técnicos o para organizaciones que buscan construir bots simples basados en reglas. RASA NLU se puede utilizar para crear bots basados en voz y texto y se integra con la mayoría de los canales de mensajería a través de API.

El framework de RASA se compone de dos partes:

- **RASA NLU:** Esta parte se encarga del lenguaje natural, para ello se nutre de los idiomas disponibles en la librería spaCy. Su misión es realizar el procesamiento del mensaje y posteriormente transformarlo en datos con estructura concreta.

- **RASA Core:** Esta parte se encarga de la gestión del diálogo, se conecta con RASA NLU. El componente decide qué acciones tomar en cada momento haciendo uso de un modelo de machine learning creado con tensorflow y keras.

La Figura 5 Proporciona una descripción general de la arquitectura de código abierto de Rasa [32]. Los dos componentes principales son la comprensión del lenguaje natural (NLU) y la gestión del diálogo. NLU es la parte que maneja la clasificación de intenciones, la extracción de entidades y la recuperación de respuestas. Se muestra a continuación como la canalización de NLU porque procesa las expresiones del usuario mediante un modelo de NLU generado por la canalización entrenada. El componente de gestión de diálogo decide la siguiente acción en una conversación en función del contexto. Esto se muestra cómo las políticas de diálogo en el diagrama [33].

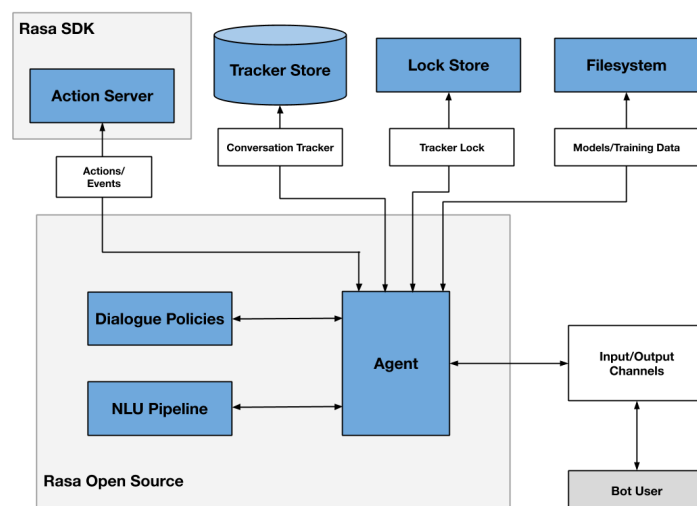


Figura 5. Arquitectura de Rasa.

4.9.4. Wit AI

Wit.ai es una plataforma gratuita de IA conversacional de código abierto respaldada por Facebook. Dispone de funciones NLU integradas para comprender intenciones, extracción de entidades, acciones, contexto, y ofrece soporte para más de 137 idiomas. Los desarrolladores pueden desarrollar e implementar chatbots en muchos diferentes canales. Wit.ai proporciona un SDK en Nodejs, Python, Ruby y HTTP API. El uso de Knowledge Graphs y bases de datos se integra con el uso de APIs [34].

Permite extraer información como la ubicación, la hora, la fecha, el tiempo y también permite crear intenciones propias. Analiza la entrada del usuario en datos estructurados y también ayuda en la conversación enviando el texto. Sin embargo, para realizar acciones sobre los datos estructurados, la salida de wit.ai debe ser enviada a la aplicación o al servidor cuya respuesta wit.ai devuelve a la plataforma de conversación como messenger o Kik. Wit.ai tiene una falta de ingresos estáticos como en el resto de plataformas obliga a la herramienta a ofrecer un servicio más modesto, mucho más lento y con peores resultados, quedando en segundo plano cuando se está buscando un resultado profesional y comercializable [35].

El modelo de aprendizaje automático del procesamiento de lenguaje natural se basa en ejemplos, es decir que se alimenta la base de conocimiento mediante el ingreso de frases por parte del usuario. Una vez ingresadas las reglas, se las puede identificar mediante intenciones entidades dependiendo del contexto que se maneje. Wit.ai tiene un gran mecanismo para entender las entidades del lenguaje, permitiendo asignar un rol a las entidades, para ayudar en el procesamiento ya sea de texto o de voz en el lado del servidor [36]. En la Figura 6 se representa una intención “reserva de restaurante” y las entidades que la acompañan “número de personas”, “lugar”, “día”.

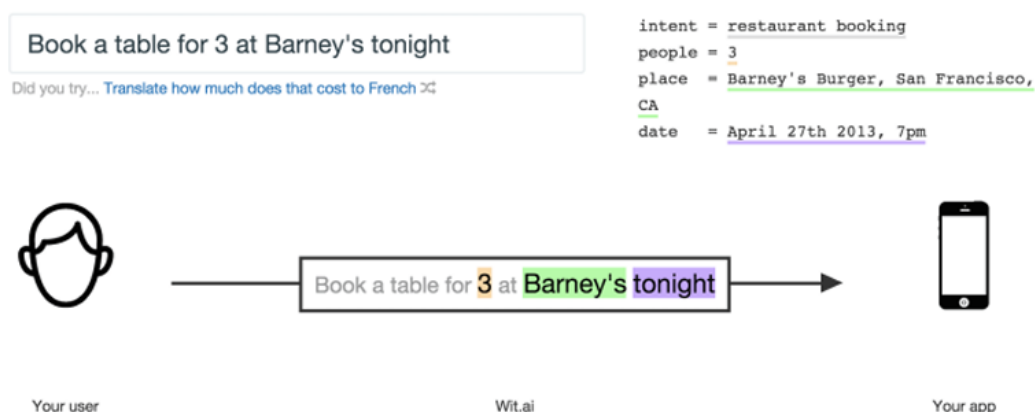


Figura 6. Representación de entidades e intenciones.

4.9.5. Dialog Flow

Dialog Flow es una tecnología que permite a los usuarios pertenecientes a Google,

anteriormente llamada Api.ai interactuar a través de interfaces conversacionales basadas en voz y texto , potenciados por inteligencia artificial, con productos como puedan ser Google Assistant o Amazon Alexa entre otros tantos. Esta herramienta proporciona la posibilidad de crear agentes conversacionales dedicados a obtener información de los usuarios, para después interpretarla, comprenderla, y proporcionar una respuesta previamente programada. Con este intercambio de información es como se perfecciona la conversación [37].

Dialogflow facilita su integración en multitud de plataformas, actualmente 16, entre las que destacan: Twitter, Slack, Facebook Messenger, Amazon Alexa, etc. Permite a sus usuarios acceder a sus interfaces a través de cualquier dispositivo inteligente como smartphones o smartwatches, por lo que pone a disposición del desarrollador diferentes SDKs. más que suficientes [38] .

La arquitectura de Dialog Flow consta de un módulo de comprensión del lenguaje natural que comprende los matices del lenguaje humano, llegando a manejar conversaciones con los usuarios. Dialog Flow traduce el texto o el audio del usuario final durante una conversación a datos estructurados que sus aplicaciones y servicios pueden entender. Estos agentes son entrenados para manejar los escenarios de conversación esperados, y su entrenamiento no necesita ser demasiado explícito.

4.9.6. IBM Watson

IBM Watson es una plataforma de Inteligencia Artificial para la creación de conversaciones con los clientes a través de respuestas rápidas, directas y precisas en cualquier aplicación. IBM Watson utiliza intenciones, entidades y el seguimiento de diálogos para centrarse en casos complejos y no en respuestas redundantes cuando conversa con el usuario. Además, es compatible con servicios de traducciones, reconocimiento y digitalización de voz y se puede implementar en la nube o de manera local.

La plataforma ofrece un conjunto de características avanzadas que incluyen el descubrimiento de datos guiados y aprovecha los flujos de diálogo dinámicos junto con un sistema de preguntas y respuestas para impulsar la interacción del usuario. IBM Watson

también ofrece enrutamiento de habilidades y opciones de personalización de experiencias.

Se basa en una red neuronal que utiliza un marco de procesamiento para comprender y aprender las señales de conversación. La plataforma también ofrece kits de desarrollo de software (SDK) que permiten construir e integrar chatbots utilizando Java, Python, Node, iOS y Unity [39].

El sistema operativo optimizado que diseña para la analítica compleja, se hace posible por la integración de procesadores paralelos de POWER7 y el software de DeepQa el cual estructura al sistema de forma que reciba una pregunta y tenga una determinada respuesta.

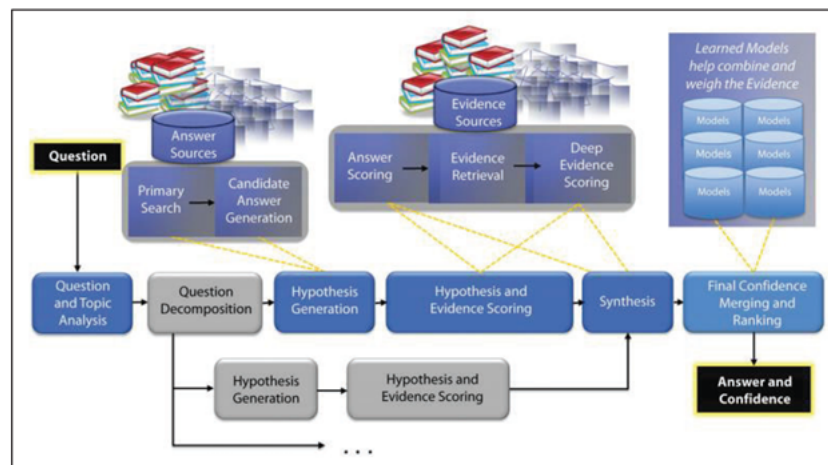


Figura 7. Diagrama de Arquitectura de IBM Watson.

Para realizar la búsqueda de respuestas a una pregunta determinada, la tecnología de IBM Watson realiza cuatro pasos [39], que se pueden evidenciar en la Figura 7.

Los pasos realizados por IBM Watson son [39] :

- Análisis de la pregunta.
- Generación de hipótesis.
- Puntuación de la hipótesis a través de la evidencia.
- Unión final y clasificación.

4.9.7. Amazon Lex

Amazon Lex es un servicio de Amazon Web Services (AWS) para el desarrollo de interfaces conversacionales llamadas chatbots para cualquier aplicación que utilice voz y texto. Está diseñado para permitir al usuario construir una interfaz de chat dentro de la consola de Amazon Lex tiene muchas funcionalidades y características a las que tiene que someterse antes de que una conversación pueda tener lugar. Hace uso de tecnologías de aprendizaje profundo como el Reconocimiento Automático del Habla (ASR) para convertir el habla en texto, y la NLU ayuda al bot a entender y averiguar las intenciones del usuario. Estas tecnologías trabajan juntas para crear un sistema de comprensión del habla y el lenguaje que ayuda a garantizar que las aplicaciones construidas sean dinámicas y lo más flexibles posible [40].

El reconocimiento automático del habla (también conocido como reconocimiento del habla o "speech-to-text") es el mecanismo por el cual los algoritmos incorporados en un sistema o una máquina se utilizan para traducir las señales de voz en su correspondiente secuencia de palabras u otras entidades lingüísticas [41].

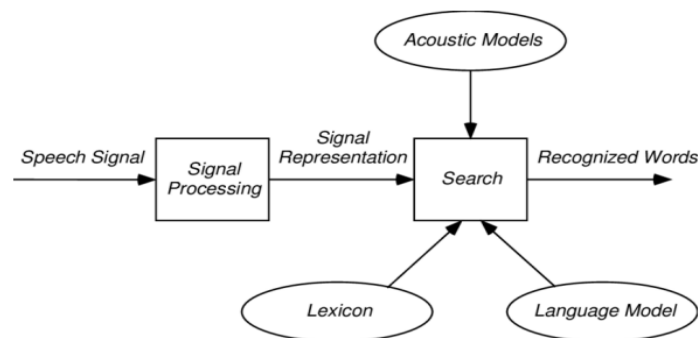


Figura 8. Proceso de conversión de voz a texto Amazon Lex.

4.9.8. Pandorabots

Pandorabots es un servicio web en línea para construir y desplegar chatbots. La plataforma Pandorabots soporta el lenguaje de script estándar abierto, Artificial Intelligence Markup Language (AIML) para actuar como base de conocimiento y parcialmente el comportamiento

del chatbot.

Pandorabots no incluye ningún algoritmo de aprendizaje automático para procesar las entradas humanas. El desarrollador debe utilizar AIML para definir los conjuntos de datos para las reglas de comunicación entre un usuario y el chatbot .

AIML utiliza la comunicación basada en patrones, y busca y combina la entrada del usuario con los patrones del chatbot en las etiquetas. Las etiquetas AIML son un pequeño bloque de construcción de conocimiento que contiene la IA del chatbot. Las etiquetas más importantes son <category>, <pattern> y <template> . La categoría contiene reglas para asignar la entrada del usuario almacenada en la etiqueta de patrón con una respuesta almacenada en la etiqueta de plantilla. La siguiente Figura 9 describe cómo una pregunta del usuario y una respuesta del chatbot podrían almacenarse en AIML [42].

```
<category>  
    <pattern>Hello, how are you doing </pattern>  
    <template>Hi, I'm doing fine, thank you for asking.</template>  
</category>
```

Figura 9. Etiquetas AIML Pandorabots.

La plataforma ofrece SDK para la programación en Java, Ruby, PHP y Javascript y puede integrarse con muchos servicios y canales de terceros que incluyen plataformas de voz y mensajería como Messenger, Kik, Telegram y Twitch. El marco de Pandorabots es gratuito para su uso en cualquier proyecto, pero tiene una cantidad limitada de solicitudes a 1000 mensajes por mes. Algunos requisitos previos para el desarrollo con Pandorabots son un navegador web y una cuenta de correo electrónico registrada en el sitio web de Pandorabots [42].

4.9.9. BotPress

Botpress es un framework de código abierto escrito en Javascript. Cuenta con módulos integrados como QnA, Analytics, Flow editor y NLU para construir un chatbot conversacional

para cualquier sistema. Un chatbot de Botpress puede ser construido, desplegado y gestionado tanto localmente como en cualquier servidor, con el uso de los conectores incorporados o a través de la API pública para la plataforma preferida.

La plataforma combina el código básico y la infraestructura necesarios para implementar y operar un chatbot, y ofrece una plataforma completa y fácil de usar que incluye todas las herramientas necesarias para capturar, implementar y administrar chatbots de nivel de producción [43]. La plataforma de Botpress incluye:

- Tareas integradas de procesamiento de lenguaje natural, como clasificación de intenciones , revisión ortográfica, extracción de entidades y muchas otras.
- Un estudio de conversación visual para diseñar conversaciones y flujos de trabajo de varios turnos.
- Un emulador y un depurador para simular conversaciones y depurar los chatbot.
- Compatibilidad con canales de mensajería populares como Slack, Telegram, Microsoft Teams, Facebook Messenger y un chat web integrado.
- Un SDK y un editor de código para ampliar las capacidades.

Botpress ofrece soporte multilingüe para usuarios con licencia Pro. Esto significa que las personas que hablan diferentes idiomas pueden usar el mismo chatbot empleando herramientas de traducción integradas.

Un chatbot desarrollado con el uso del marco Botpress recibirá mensajes de un usuario a través de un canal, luego procesa los mensajes para que el chatbot los entienda y los traduzca a datos estructurados. A continuación, el chatbot decide qué acción realizar y genera una respuesta al usuario a través del módulo Dialog Manager [44]. La Figura 10 describe cómo el chatbot de Botpress procesa las entradas del usuario.

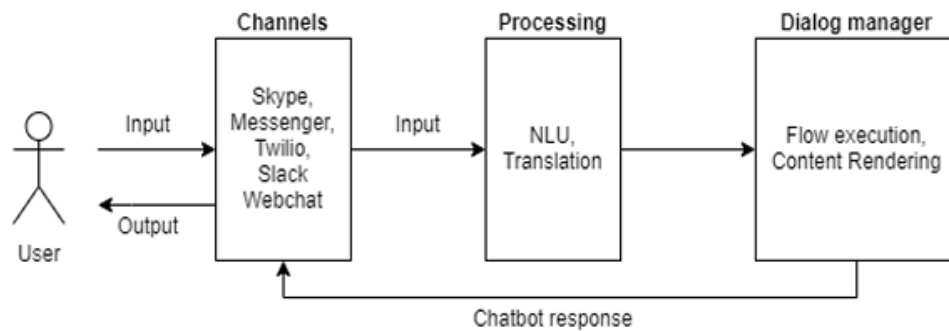


Figura 10. Proceso de Información Botpress.

4.9.10. Botkit

Botkit es una librería node.js de código abierto para el desarrollo de chatbots por howdy.ai (recientemente adquirida por Microsoft). Ofrece ciertas funcionalidades que pueden ser utilizadas para desarrollar chatbots basados en reglas escritas en JavaScript. No está bien documentado qué chatbots se han desarrollado con Botkit. Sin embargo, hasta ahora se han desarrollado alrededor de 10000 chatbots utilizando Botkit.

Está diseñado para ayudar a la gente ocupada a crear rápida y fácilmente bots para sus necesidades, sin tener que profundizar en el código interno, proporcionando una interfaz unificada para enviar y recibir mensajes. Inicialmente destinado a Slack, ahora tiene una funcionalidad ampliada para admitir la conexión a varias plataformas de mensajería. El marco tiene flujos de trabajo intuitivos organizados de manera clara y concisa, está bien documentado y proporciona una gran cantidad de ejemplos de chatbots en vivo para que los explore, por lo que es realmente fácil comenzar y usar además es importante tener en cuenta que el marco no tiene capacidades de NLP, pero se puede resolver conectando servicios de NLP existentes o personalizados a través de middlewares [45].

4.9.11. ChatterBot

ChatterBot es una biblioteca de Python que facilita la generación de respuestas automatizadas a las entradas de los usuarios. ChatterBot utiliza una selección de algoritmos de aprendizaje para generar diferentes tipos de respuestas. Esto permite a los desarrolladores crear fácilmente chatbots y automatizar las conversaciones con los usuarios.

El diseño neutral de ChatterBot permite entrenarlo para que hable cualquier idioma. Además, la naturaleza de aprendizaje automático de ChatterBot permite a un agente ampliar su propio conocimiento de las posibles respuestas cuando interactúa con humanos y otras fuentes de datos de inteligencia.

ChatterBot está diseñado para facilitar la creación de software que pueda iniciar una conversación. Un ejemplo no entrenado de ChatterBot comienza sin saber cómo comunicarse. Cada vez que un usuario escribe una frase, la biblioteca almacena el texto que ha escrito y el que responde. A medida que ChatterBot recibe más información, aumenta el número de respuestas que puede responder y la precisión de cada respuesta con respecto a la frase grabada. El programa selecciona la respuesta más parecida encontrando la configuración conocida más cercana que corresponde a la entrada, y luego seleccionando una respuesta de la selección de soluciones conocidas para esa configuración [46].

ChatterBot utiliza algoritmos de clasificación bayesiana ingenua para determinar si una expresión de entrada cumple un conjunto de criterios específicos que garantizan la generación de una respuesta de ese modificador lógico. La clasificación bayesiana ingenua es una técnica de clasificación basada en el teorema de Bayes con una presunción de independencia entre los predictores. En resumen, el clasificador Naive Bayes asume que la presencia de una característica particular en una clase no está relacionada con la presencia de otra característica. Por ejemplo, una fruta puede considerarse una manzana si es roja, redonda y de unos 5 cm de diámetro. Aunque estos caracteres dependen unos de otros o de la existencia de otros caracteres, todas estas propiedades contribuyen de forma independiente a la probabilidad de que esta fruta sea una manzana y, por tanto, se dice que es "Naive".

4.10 Métodos

4.10.1. Método científico

El concepto del método surgió porque grandes filósofos griegos como Aristóteles, Platón o Sócrates notaron la necesidad de seguir un conjunto de axiomas o reglas para lograr el objetivo previsto, que en contadas ocasiones dan sabiduría [65], este método consta de

varios pasos, que engloban todo el trabajo de investigativo [66], para el presente TT se tomaron en cuenta cuatro de ellas como se muestra en la Figura 11.

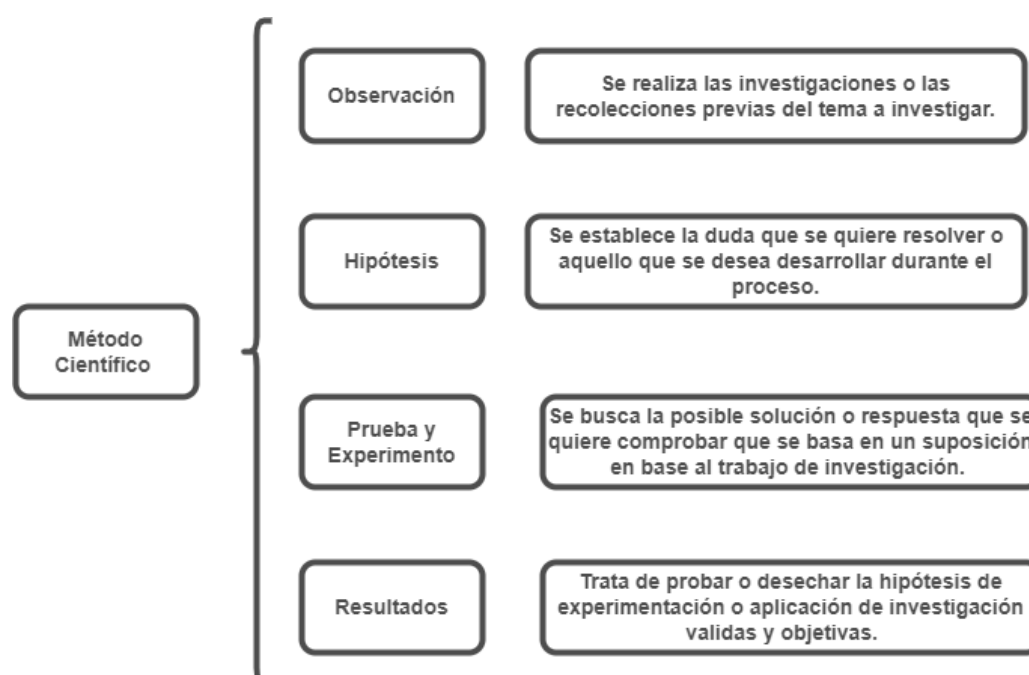


Figura 11. Pasos del Método Científico.

4.10.2. Revisión Sistemática de Literatura

La revisión sistemática de literatura (RSL) se define como estudios secundarios, integrativos, observacionales y retrospectivos el cual se combina en conjunto con los estudios que examinan el mismo objetivo. La revisión sistemática, se presenta en dos formas: "cuantitativa o metanálisis" y "cualitativo o overview", pero sus diferencias radican en el uso de métodos estadísticos que permiten la combinación y el análisis cuantitativo de los resultados obtenidos en cada estudio [67]. Dentro de los trabajos relacionados a la RSL se encuentran en dos tipos: aquellos que proponen pautas metodológicas para RSL y aquellos que han llevado a cabo estudios retrospectivos terciarios de RSL. Entre los primeros se destaca el trabajo de Kitchenham, donde recomiendan pautas para realizar revisiones sistemáticas de la literatura sobre desarrollo de software para guiar a los investigadores en la evaluación e interpretación de todas las publicaciones de investigación disponibles para una pregunta de investigación en particular. Estas directrices proporcionan un marco definido para el desarrollo de la RSL y sus subfases, que consideran la metodología de investigación, los resultados y finalmente, la agregación de evidencia [68].

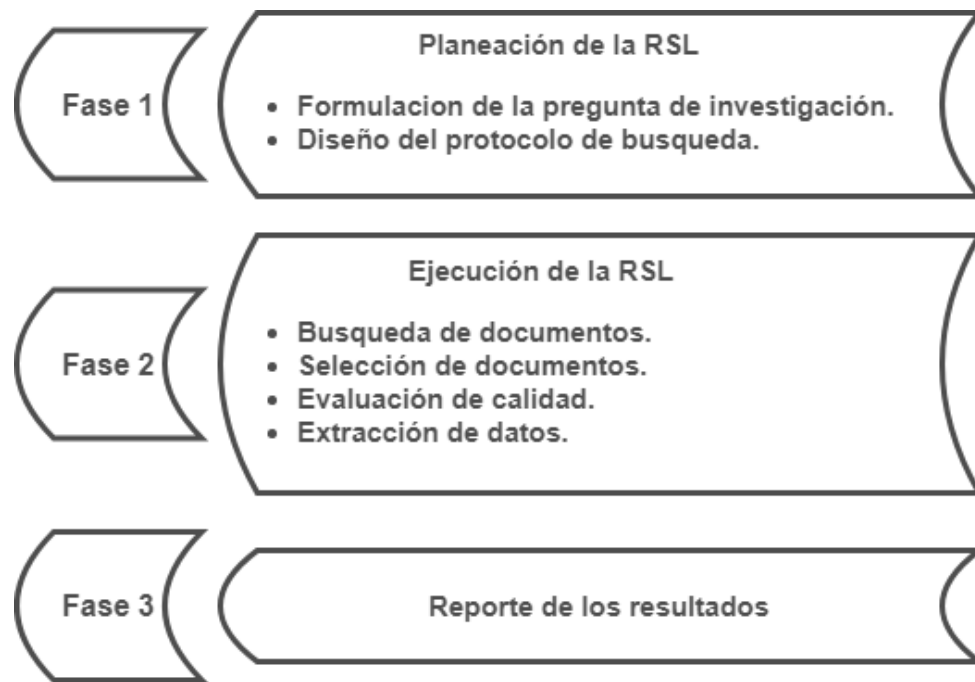


Figura 12. Fases de la RSL.

4.11. Trabajos relacionados

Proposed Chatbot Framework for COVID-19

En este artículo, presenta un sistema de chatbot inteligente que puede comunicarse con las personas y proporcionarles respuestas sobre la COVID-19. Para abordar la popular función de respuesta a preguntas, el modelo utilizó el modelo de lenguaje BERT de Google previamente entrenado. Además del BERT, añadieron dos fases arquitectónicas para la tarea de respuesta a preguntas. La primera fase es una técnica de clasificación de texto que emplea el transformador BERT para clasificar la entrada de texto en varias categorías basadas en el significado de las propias palabras. La aplicación real del modelo BERT, así como el dominio de consulta para las respuestas, es el segundo paso. El sistema propuesto se ha entrenado y probado en SQuAD V2.0 de la Universidad de Stanford, un conocido conjunto de datos de respuesta a preguntas [47].

A Conceptual Framework for Engineering Chatbots

Este artículo esboza un marco conceptual de alto nivel para crear chatbots flexibles basados en abstracciones orientadas a los agentes: objetivos, planes y compromisos de la IA y los sistemas multiagente. Las abstracciones permiten el diseño formal y la verificación en tiempo de ejecución y van acompañados de metodologías prácticas de diseño. El framework

propuesto consta de cinco componentes con responsabilidades asociadas: administrador de diálogo, motores de inferencia, base de conocimiento, planificador e interfaz de servicios externos, este trabajo describe un modelo más genérico para diseñar chatbots en todos los dominios [48].

Conversational Question Answering Over Knowledge Base using Chat-Bot Framework

En este trabajo, proponen un Asistente de Respuesta Robusto (RADA) que utiliza el marco del chatbot para superar este problema. Consiste en un conjunto de modelos de reconocimiento de entidades, predicción de entidades, respuesta a preguntas y sistema de diálogo. Llevaron a cabo experimentos cuantitativos, incluyendo comparaciones con el estado del arte en el conjunto de datos Web-Question [49].

Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural.

El siguiente proyecto, plantea el desarrollo de un framework orientado a objetos para Sistemas de Generación de Lenguaje Natural (GLN), donde revisan las arquitecturas utilizadas hasta la fecha, se revisa una propuesta de arquitectura genérica, y se describe el tipo de solución que se aspira a conseguir. A partir de esa información muestran cómo la utilización de metodologías típicas de Ingeniería de Software, como la Orientación a Objeto o los Patrones de Diseño, resuelve algunos de los problemas más acuciantes de los desarrolladores de aplicaciones de GLN [50].

5. Metodología.

Para el desarrollo de la presente sección y de acuerdo a lo establecido en el Reglamento de Régimen Académico de Educación Superior de Ecuador, en el artículo 21, se estipula que un Trabajo de Titulación (TT) deberá constar de una propuesta innovadora que comprenda de una investigación exploratoria y diagnóstica, además, acorde al artículo 72 del mencionado reglamento, la investigación a nivel de grado se desarrollará en el campo formativo de creación, adaptación e innovación tecnológica mediante el dominio de técnicas investigativas de carácter exploratorio y descriptivo [51]. La investigación exploratoria permitió realizar un enfoque de teoría fundamentada, buscar sobre las tecnologías vigentes para el desarrollo de

frameworks y disponer de una idea más clara de cómo construir el framework de despliegue de chatbots. En cambio, la investigación descriptiva permitió analizar el objeto de estudio, para describir su funcionamiento, estructuración y las limitaciones que tiene para resolver un determinado problema [52].

Dentro de la presente investigación, se considera el uso del método científico como base para la exploración de los distintos frameworks de chatbot en base a las tecnologías actuales. Primero se realizó un estudio preliminar del trabajo, cuyo objetivo fue obtener las tecnologías y herramientas como lo son: BERT y RASA, que permitieron establecer la arquitectura del framework de despliegue de chatbots. Segundo se realizó el diseño y codificación de la arquitectura del framework con herramienta de RASA y BERT, obtenidos en los resultados de la investigación preliminar. Finalmente se realizaron las pruebas de framework a partir de un ejemplo de gestión de matrículas de la carrera de Ingeniería en Sistemas/ Computación; (véase en la sección Metodología, apartado 5.2) se explica el proceso que se cumplió para el desarrollo del TT; además, se detallan los materiales, métodos y participantes que interactuaron en el desarrollo del TT (véase en la sección Metodología, apartado 5.3).

5.1. Área de estudio

El presente proyecto de trabajo de titulación fue realizado en la Facultad de la Energía, los Recursos Naturales no Renovables de la Universidad Nacional de Loja específicamente en la carrera de Ingeniería en Sistemas/Computación, lugar de experimentación y recolección de datos.

5.2. Procedimiento

En el presente apartado se narra el proceso que se siguió para desarrollo del objetivo general del presente proyecto de trabajo de titulación, detallado a continuación:

5.2.1. Fase 1: Estudiar las tecnologías para la construcción del framework de asistentes virtuales: Se realizó con el fin de obtener un estudio preliminar de todos los aspectos que constituyen el desarrollo e implementación de un framework de asistentes virtuales.

5.2.1.1. Desarrollo de estado de Arte.

Para el cumplimiento de la primera tarea comprendida por el desarrollo del estado de arte se definieron dos subtarear, las cuales comprenden de la revisión de los autores en el ámbito del machine learning, tomando en cuenta dos revisiones: la primera bibliográfica y literaria y la segunda mediante la RSL, esto con el fin de abarcar de una forma correcta todo el estudio previo a la problemática a resolver del TT.

- Revisión bibliográfica y literaria de autores en el ámbito de (Machine Learning) y procesamiento de lenguaje natural: El cumplimiento de la primera subteara se lo realizó mediante una revisión bibliográfica y literaria utilizando el método científico el cual permitió obtener las posibles tecnologías para el desarrollo de frameworks de despliegue de asistentes virtuales que hacen uso diferentes autores en el PLN y Machine Learning, para más detalle (véase en la sección Resultados, apartado 6.1.1.1.).
- Revisión sistemática de autores en el ámbito de (Machine Learning) y procesamiento de lenguaje natural: El cumplimiento de la segunda subteara se lo realizó mediante una RSL utilizando la metodología de Barbara Kitchenham, teniendo en cuenta las fases de: planeación, ejecución y reporte, el cual permitió obtener resultados más certeros de las tecnologías para el desarrollo de frameworks de despliegue de asistentes virtuales que hacen uso diferentes autores en el PLN y Machine Learning, para más detalle (véase en la sección Resultados, apartado 6.1.1.2.).

5.2.1.2. Revisión de los trabajos relacionados.

Para el cumplimiento de la segunda tarea comprendida por la revisión de los trabajos relacionados se definieron dos subtarear, las cuales comprenden de la revisión de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales, tomando en cuenta dos revisiones: la primera bibliográfica y literaria y la segunda

mediante la RSL, esto con el fin de abarcar de una forma correcta todo el estudio previo a la problemática a resolver del TT.

- Revisión bibliográfica y literaria de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales: El cumplimiento de la primera subtarea se lo realizó mediante una revisión bibliográfica y literaria utilizando el método científico y a partir de las posibles tecnologías encontradas en la primer tarea, permitió conocer obtener dar más contexto a las tecnologías y elementos técnicos que constituyen el desarrollo de un frameworks de asistentes virtuales, para más detalle (véase en la sección Resultados, apartado 6.1.2.1.).
- Revisión sistemática de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales: El cumplimiento de la segunda subtarea se lo realizó mediante una RSL utilizando la metodología de Barbara Kitchenham, teniendo en cuenta las fases de: planeación, ejecución y reporte, partiendo de las posibles tecnologías encontradas en la primera tarea, permitió obtener contextos más certeros de las tecnologías y elementos técnicos que constituyen el desarrollo de un frameworks de asistentes virtuales, para más detalle (véase en la sección Resultados, apartado 6.1.2.2.).

5.2.1.3. Clasificación de la información.

Para el cumplimiento de la tercera tarea comprendida por la clasificación de la información se definieron dos subtarear: la primera comprendió de la comparación de todas las tecnologías recopiladas de la primer tarea y la segunda comprendió de la selección de la mejor tecnología a partir de una comparación de todos los frameworks obtenidos de la segunda tarea, esto con el fin de abarcar de una forma correcta todo el estudio previo a la problemática a resolver del TT.

- Comparación de las mejores tecnologías para el desarrollo de un framework de asistentes virtuales: El cumplimiento de la primera subtarea se realizó mediante una tabla comparativa que abarcan los marcos de frameworks de chatbot, permitiendo obtener una visión más específica del contenido y características necesarias e

importantes de cada framework para su posterior selección, para más detalle (véase en la sección Resultados, apartado 6.1.3.1.).

- Selección de la tecnología para el desarrollo del framework: El cumplimiento de segunda subtarea comprendió de la selección de la tecnología a partir de la comparación realizada en la primera subtarea, esto con el fin de obtener una arquitectura ideal para el framework de despliegue de asistentes virtuales, para más detalle (véase en la sección Resultados, apartado 6.1.3.2.).

5.2.1.4. Documentación de cada aspecto investigativo.

5.2.2. Fase 2: Construir el framework de despliegue de asistentes virtuales en función de las tecnologías vigentes seleccionadas: Se realizó con el fin de obtener una arquitectura, estructura y producto bien desarrollado para el despliegue de asistentes virtuales.

5.2.2.1 Establecimiento del Corpus lingüístico del framework.

Para el cumplimiento de la primera tarea comprendida por el establecimiento del Corpus lingüístico del framework, se seleccionó como base un corpus lingüístico obtenidos de la plataforma de código abierto huggingface.co, con el fin de obtener un modelo multilingües previamente entrenado con predicción de oración, para más detalle (véase en la sección Resultados, apartado 6.2.1.).

5.2.2.2 Codificación del framework.

Para el cumplimiento de la segunda tarea comprendida por la codificación del framework se definieron dos subtarear: la primera comprendió por el desarrollo del componente inteligente que abarca la parte del backend del framework y la segunda comprendió por el desarrollo del backend con la parte del componente inteligente y el corpus lingüístico, esto con el fin de tener abarcar de forma correcta toda la codificación del framework.

- Desarrollar el componente inteligente del framework: El cumplimiento de la primera subtarea se realizó mediante una tabla comparativa que abarcan los marcos de frameworks de chatbot, permitiendo obtener una visión más específica del contenido

y características necesarias e importantes de cada framework para su posterior selección, para más detalle (véase en la sección Resultados, apartado 6.1.3.1.).

- Selección de la tecnología para el desarrollo del framework: El cumplimiento de segunda subtarea comprendió de la selección de la tecnología a partir de la comparación realizada en la primera subtarea, esto con el fin de obtener una arquitectura ideal para el framework de despliegue de asistentes virtuales, para más detalle (véase en la sección Resultados, apartado 6.1.3.2.).

5.2.2.3 Documentación técnica de cada aspecto desarrollado en las distintas etapas de la construcción del framework.

5.2.3. Fase 3: Validar el framework mediante el despliegue de información académica relacionada a las temáticas dentro de la CIS: Se realizó con el fin de validar la correcta funcionalidad del framework apartir de tematicas relacionadas a la carrera de la CIS.

5.2.3.1. Identificación y recopilación de la fuente de información de las tematicas de la Carrera de CIS.

5.2.3.2. Evaluación del framework a partir de la fuente de información

5.2.3.3. Corrección y validación del framework.

5.2.3.4 Documentación de los resultados obtenidos de la validación del framework.

5.3. Materiales

Para solventar a la pregunta de investigación y objetivos planteados en el Anteproyecto se usaron los siguientes materiales:

5.3.1. Recursos de Hardware y Software

Los recursos de Hardware y Software utilizados en el transcurso del desarrollo del presente TT se muestran a más detalle en la Tabla 1.

Tabla 1. Recursos de Hardware y Software utilizados

HARDWARE	
Equipo	Descripción del uso
Laptop Asus NotebookSKU Core i7 modelo G75Vs	Permitió realizar las diferentes investigaciones referentes al TT, así como las reuniones con el director de tesis, además, fue base fundamental para el desarrollo del framework
Laptop DELL Core i7 modelo inspiron 15 Series 15000	Permitió realizar las diferentes investigaciones referentes al TT, así como las reuniones con el director de tesis, además, fue base fundamental para el desarrollo del framework.
SOFTWARE	
Visual Studio Code	Utilizado para la codificación del Framework
Paquete Office	Utilización para la redacción del documento del TT.
Mendeley Desktop	Utilizado para la gestión bibliográfica del TT.
Google Drive	Utilización para la administración de los documentos en la nube.
Git Hub	Utilizado para el control de versiones en el desarrollo del framework.
Bert	Es un marco de aprendizaje automático de código abierto para el procesamiento del lenguaje natural.
Rasa	Marco de aprendizaje automático de código abierto para crear asistentes de inteligencia artificial y chatbots

5.3.2. Participantes

El presente TT enfocado en la línea de investigación de línea de investigación, el desarrollo del framework fue realizado en la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja del periodo académico Abril - Septiembre 2022, por los participantes mostrados en la Tabla 2.

Tabla 2. Participantes

Rol	Participante	Funciones
Estudiantes	Francisco Carmelino Agreda Sánchez y Jorge Luis Auquilla Villamagua	Estudiantes autores del presente TT. Sus actividades iniciaron desde el planteamiento del tema del TT, hasta el desarrollo y finalización de los objetivos planteados en el presente TT.
Director del TT	Ing. Roberth Gustavo Figueroa Díaz M.Sc.	Como tutor académico y director del TT, quién supervisó los avances académicos y técnicos desarrollados por los autores del presente trabajo.

5.3.3 Métodos

Para solventar a la pregunta de investigación y objetivos planteados en el Anteproyecto se usaron los siguientes métodos:

- **Método Científico**

Este método se empleó para la recolección de la información relacionada al TT, a través de investigación de documentos provenientes de fuentes confiables, todo esto sirvió como base para el desarrollo del presente TT, desde su planteamiento en el Anteproyecto, hasta la culminación del mismo. Esto se puede evidenciar a través del desarrollo de las tareas planteadas por cada objetivo (véase en la sección

Metodología, apartado 5.2) y los resultados obtenidos de cada objetivo (véase sección Resultados, apartado 6.1.), en conjunto con la respectiva discusión (véase sección Discusión) y sus conclusiones (véase sección Conclusiones).

- **Revisión Sistemática de Literatura**

Este método se empleó para el cumplimiento del objetivo uno, específicamente de las tareas uno y dos del presente TT, véase Anexo 11.1 y Anexo 11.2, tomando como base la metodología de Barbara Kitchenham, [62][63][64] como un medio para identificar y analizar toda la investigación disponible relevante respecto a una interrogante de investigación particular, su importancia radica en la objetividad de los resultados que se obtiene con el proceso sistemático (véase en la sección Resultados, apartado 6.1.).

- **Estudio de casos**

Este método de estudios de casos sirvió para la exploración de casos reales y a su vez poder tener un conocimiento más extenso acerca de la problemática de investigación planteada en el anteproyecto y poder dar una solución a la misma. Además permitirá delimitar el caso de estudio y obtener las tecnologías necesarias para el diseño de la arquitectura, desarrollo e implementación de la solución a la problemática (véase en la sección Resultados, apartado 6.1.1. y 6.1.2.) y (véase en la sección Marco Teórico, apartado 4.11.).

6. Resultados.

En esta sección se presenta la evidencia empírica obtenida, enfocados a cumplir con los objetivos, tanto general como específicos del trabajo de titulación “Framework para el despliegue de temáticas de asistentes virtuales aplicado a la Carrera de Ingeniería en Sistemas”, este proceso fue realizado en tres fases, con sus respectivas tareas, las cuales fueron ejecutadas de manera ordenada y se detallan a continuación.

6.1. Primera fase: Estudiar las tecnologías para la construcción del framework de asistentes virtuales.

Para el cumplimiento de la primera fase se desarrollaron las siguientes tareas.

6.1.1. Desarrollo de estado del Arte.

Para llevar a cabo el cumplimiento de la primera tarea el desarrollo del estado del arte se tomaron en cuenta dos subtareas, que consistieron en la revisión bibliográfica y literaria, y revisión sistemática de literatura, cuyos resultados se detallan a continuación.

6.1.1.1. Revisión bibliográfica y literaria de autores en el ámbito de (Machine Learning) y procesamiento de lenguaje natural.

Para la realización de la primera subtarea se elaboró una revisión bibliográfica y literaria con el fin de identificar y analizar los autores en el ámbito del Machine Learning y procesamiento de lenguaje natural (PLN). Para el desarrollo de la revisión literaria fue tomado en cuenta el método científico y a partir de la temática central denominada “Autores en el ámbito del Machine Learning y procesamiento de lenguaje natural” se planteó una pregunta de investigación: ¿Casos de éxito de trabajos de desarrollo frameworks de chatbots?, además se han tomado 3 criterios de inclusión y exclusión. Los estudios se obtuvieron de las bases de datos IEEE Xplore, Science@Direct y ACM Digital Library (ver Tabla 3).

Tabla 3. Resultados de los estudios seleccionados a partir de la pregunta planteada.

Preguntas Planteada	Estudios Seleccionados	Revisión Literaria
¿Casos de éxito de trabajos de desarrollo frameworks de chatbots?	[49]	Conversational Question Answering Over Knowledge Base using Chat-Bot Framework. En este trabajo, proponen un Asistente de Respuesta Robusto (RADA) que utiliza el marco del chatbot, que consiste en un conjunto de modelos de reconocimiento de entidades, predicción de entidades, respuesta a preguntas y sistema de diálogo. Llevaron a cabo experimentos cuantitativos, incluyendo comparaciones con el estado del arte en el conjunto de datos Web-Question.
	[50]	Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural. El siguiente proyecto, plantea el desarrollo de un framework orientado a objetos para Sistemas de Generación de Lenguaje Natural (GLN), donde

		revisan las arquitecturas utilizadas hasta la fecha, revisan una propuesta de arquitectura genérica, y se describe el tipo de solución que aspiran conseguir. A partir de esa información muestran cómo la utilización de metodologías típicas de Ingeniería de Software, como la Orientación a Objeto o los Patrones de Diseño, resuelve algunos de los problemas más acuciantes de los desarrolladores de aplicaciones de GLN.
--	--	--

6.1.1.2. Revisión sistemática de autores en el ámbito de (Machine Learning) y procesamiento de lenguaje natural.

Para la realización de la segunda subtarea, se elaboró una revisión sistemática de literatura (RSL) con el fin de identificar y analizar los autores en el ámbito del Machine Learning y procesamiento de lenguaje natural (PLN).

- Planificación de la RSL

Para el desarrollo de la RSL se empleó la metodología propuesta de Bárbara Kitchenham, para orientar la RSL, tomando en cuenta la metodología empleada y a partir de la temática central denominada “Autores en el ámbito del Machine Learning y procesamiento de lenguaje natural” se aplicó el método PICOC que consiste en definir las variables: Población, Intervención, Comparación, Resultado, Contexto, en la Tabla 4 se describen las variables usadas para el planteamiento de la pregunta.

Tabla 4. Criterios PICOC utilizados en la investigación.

Criterio	Descripción
Población	Autores en Machine Learning en Procesamiento de Lenguaje Natural
Intervención	Machine Learning y Procesamiento de Lenguaje Natural
Comparación	Conocimiento general, Tipos
Resultados	Chatbots, Herramientas
Contexto	Tipos y herramientas usadas en el desarrollo de chatbots

A Partir de los resultados obtenidos de la realización del método PICOC se plantean tres preguntas de investigación clasificadas en preguntas para el mapeo sistemático y la revisión sistemática: ¿Casos de éxito de herramientas para el desarrollo de chatbots? ¿Tipos de procesamiento de lenguaje natural implicados? ¿Existen métodos o estándares?, además la selección de los artículos se realizó acorde a un rango máximo de 5 años desde su publicación de acuerdo a los criterios de inclusión (ver tabla 5). Los estudios se obtuvieron de las bases de datos IEEE Xplore, Science@Direct y ACM Digital Library (para la recolección de esta información se usó cadenas de búsqueda y motores de búsqueda) (ver Tabla 6).

Tabla 5. Criterios de Inclusión y Exclusión.

Criterios de Inclusión	Criterios de Exclusión
<ul style="list-style-type: none"> • Documento con rango de publicación no mayor a 5 años de antigüedad. • Documento en Español o Inglés. • Documento relacionado al Machine Learning, PLN o Chatbots. 	<ul style="list-style-type: none"> • Documento con rango de publicación mayor a 5 años de antigüedad. • Documento en otro idioma. • Documento sin relación al Machine Learning, PLN o Chatbots.

Tabla 6. Cadenas de búsqueda de cada base de datos.

Fuente Bibliográfica	Cadena de búsqueda
IEEE Library	("Autores" OR "PLN" OR "Procesamiento de lenguaje Natural") AND ("Machine Learning" OR "Aprendizaje Automático") AND ("Tipos" OR "clasificación") AND ("Chatbots" OR "Asistentes virtuales, Agentes virtuales" OR "Herramientas" OR "Tecnologías")
Science@Direct	("Autores" OR "PLN" OR "Procesamiento de lenguaje Natural") AND ("Machine Learning" OR "Aprendizaje Automático") AND ("Tipos" OR "clasificación") AND ("Chatbots" OR "Asistentes virtuales, Agentes virtuales" OR "Herramientas" OR "Tecnologías")

ACM Digital Library	("Autores" OR "PLN" OR "Procesamiento de lenguaje Natural") AND ("Machine Learning" OR "Aprendizaje Automático") AND ("Tipos" OR "clasificación") AND ("Chatbots" OR "Asistentes virtuales, Agentes virtuales" OR "Herramientas" OR "Tecnologías")
---------------------	--

- Ejecución de la RSL

Los resultados de los estudios primarios obtenidos de la RSL a partir de la ejecución de las cadenas de búsquedas en cada base de datos fueron 72 artículos, los cuales fueron revisados y analizados en su título y resumen (en la Tabla 7 se detalla el proceso de selección de los estudios, de los cuales se seleccionó 20 documentos para evaluarlos a través de los criterios de calidad).

Tabla 7. Resultados del proceso de selección de estudios primarios.

Base de Datos	Artículos				
	Encontrados	Duplicados	Revisados	Eliminados	Seleccionados
IEEE Xplore	27	0	27	21	6
Science@ Direct	25	0	25	12	13
ACM	20	0	20	19	1
Total	72	0	72	52	20

Después de haber aplicado los criterios de inclusión y exclusión, se presentan los 2 artículos seleccionados donde se detalla la información por: el título, el año de publicación y su fuente de publicación (ver Tabla 8).

Tabla 8. Artículos Evaluados

Título	Año/ Revista/Congreso
A Conceptual Framework for Engineering Chatbots [47]	2018 IEEE Internet Computing
A Proposed Chatbot Framework for	2021 IEEE Internet Computing

De esta manera al obtener los dos estudios mostrados en la Tabla 8, y los criterios de selección de estudios establecen la pauta de extracción de información relevante. Por cada uno se sintetizó y clasificó a cada artículo para tener una visión clara de las preguntas que se establecieron para evaluar la calidad de los artículos preseleccionados (Ver Anexo 1)como se describe en la Tabla 9 y Tabla 10.

Tabla 9. Tipo PLN por artículo.

Tipo PLN	Artículo
PLN	[47]
NPL	[48]
NSP	[48]

Tabla 10. Herramientas por artículo.

Herramientas	Artículo
BERT	[47][48]

- Reporte de los resultados.

El reporte obtenido tras la ejecución de la RSL se detalla en el Anexo 1.

6.1.2. Revisión de trabajos relacionados.

Para llevar a cabo el cumplimiento de la segunda tarea la revisión de trabajos relacionados se tomaron en cuenta dos subtareas, que consisten en la revisión bibliográfica y literaria, y revisión sistemática de literatura, cuyos resultados se detallan a continuación.

6.1.2.1. Revisión bibliográfica y literaria de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales.

Para la realización de la primera subtarea de la revisión de trabajos relacionados se elaboró una revisión bibliográfica y literaria con el fin de identificar, analizar y clasificar los artículos

publicados sobre las tecnologías, métodos y elementos técnicos implicados en el desarrollo de un framework de asistentes virtuales. Para el desarrollo de la revisión literaria fue tomado en cuenta el método científico y a partir de la temática central denominada “Tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales” se planteó una pregunta de investigación: ¿Tecnologías usadas para el desarrollo de frameworks de despliegue de chatbots?, además se han tomado 3 criterios de inclusión y exclusión. Los estudios se obtuvieron de las bases de datos IEEE Xplore, Science@Direct y ACM Digital Library (ver Tabla 11).

Tabla 11. Resultados de los estudios seleccionados a partir de la pregunta planteada.

Preguntas Planteada	Estudios Seleccionados	Frameworks de Chatbot
¿Tecnologías usadas para el desarrollo de frameworks de despliegue de chatbots?	[31]	BERT
	[32]	Microsoft Bot Framework
	[33]	RASA
	[34][35][36]	Wit AI
	[37][38]	Dialog Flow
	[39]	IBM Watson
	[40]	Amazon Lex
	[42]	Pandarabots
	[43]	BotPress
	[45]	Botkit
	[46]	ChatterBot

6.1.2.2. Revisión sistemática de las tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales.

Para la realización de la segunda subtarea se elaboró una revisión sistemática de literatura (RSL) con el fin de identificar, analizar y clasificar los artículos publicados sobre las

tecnologías, métodos y elementos técnicos implicados en el desarrollo de un framework de asistentes virtuales.

- Planificación de la RSL

Para el desarrollo de la RSL se empleó la metodología propuesta de Bárbara Kitchenham, para orientar la RSL, tomando en cuenta la metodología empleada y a partir de la temática central denominada “Tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales” se aplicó el método PICOC que consiste en definir las variables: Población, Intervención, Comparación, Resultado, Contexto, en la Tabla 12 se describen las variables usadas para el planteamiento de la pregunta.

Tabla 12. Criterios PICOC utilizados en la investigación.

Criterio	Descripción
Población	Frameworks, Chatbots
Intervención	Frameworks, Tecnologías, Elementos Tecnicos
Comparación	Conocimiento general, Tecnologías
Resultados	Tecnologías, estructuras, Métodos
Contexto	Frameworks, tecnologías y métodos usados en el desarrollo de Chatbots.

A partir de los resultados obtenidos de la realización del método PICOC se plantean cuatro preguntas de investigación clasificadas en preguntas para el mapeo sistemático y la revisión sistemática: ¿Tecnologías usadas en el desarrollo de chatbots? ¿Frameworks usados para construir Chatbots? ¿Metodología para desarrollar Frameworks? ¿Herramientas para el desarrollo de frameworks?, además la selección de los artículos se realizó acorde a un rango máximo de 4 años desde su publicación de acuerdo a los criterios de inclusión (ver tabla 13). Los estudios se obtuvieron de las bases de datos IEEE Xplore, Science@Direct y ACM Digital Library (para la recolección de esta información se usó cadenas de búsqueda y motores de búsqueda) (ver Tabla 14).

Tabla 13. Criterios de Inclusión y Exclusión.

Criterios de Inclusión	Criterios de Exclusión
<ul style="list-style-type: none"> • Documento con rango de publicación no mayor a 4 años de antigüedad. • Documento en Español o Inglés. • Documento relacionado con Frameworks de Chatbots. 	<ul style="list-style-type: none"> • Documento con rango de publicación mayor a 4 años de antigüedad. • Documento en otro idioma. • Documento sin relación con Frameworks de Chatbots.

Tabla 14. Cadenas de búsqueda de cada base de datos

Fuente Bibliográfica	Cadena de búsqueda
IEEE Library	("Frameworks" OR "Marco de Trabajo") AND ("Chatbots" OR "Asistentes Virtuales") AND ("Elementos Técnicos" OR "Tecnologías") AND ("Tecnologías") AND ("Arquitectura" OR "Estructura" OR "Metodologías" OR "Métodos")
Science@Direct	("Frameworks" OR "Marco de Trabajo") AND ("Chatbots" OR "Asistentes Virtuales") AND ("Elementos Técnicos" OR "Tecnologías") AND ("Tecnologías") AND ("Arquitectura" OR "Estructura" OR "Metodologías" OR "Métodos")
ACM Digital Library	("Chatbots" OR "Frameworks") AND ("Tecnologías") AND ("Arquitectura" OR "Metodologías" OR "Métodos")

- Ejecución de la RSL

Los resultados de los estudios primarios obtenidos de la RSL a partir de la ejecución de las cadenas de búsquedas en cada base de datos fueron 125 artículos, los cuales fueron revisados y analizados en su título y resumen (en la Tabla 15 se detalla el proceso de selección de los estudios, de los cuales se seleccionó 20 documentos para evaluarlos a través de los criterios de calidad).

Tabla 15. Resultados del proceso de selección de estudios primarios.

Base de Datos	Artículos				
	Encontrados	Duplicados	Revisados	Eliminados	Seleccionados
IEEE Xplore	50	0	50	35	15
Science@ Direct	25	0	25	20	5
ACM	50	0	50	50	0
Total	125	0	125	105	20

Después de haber aplicado los criterios de inclusión y exclusión, se presentan los 8 artículos seleccionados donde se detalla la información por: el título, el año de publicación y su fuente de publicación (ver Tabla 16).

Tabla 16. Artículos Evaluados

Título	Año/ Revista/Congreso
College Enquiry Chatbot using Rasa Framework [54]	2021
Implementation of Chatbot on University Website Using RASA Framework [55]	2021
An Intelligent Arabic Chatbot System Proposed Framework [56]	2021
Designing a Transactional Smart Assistant in Indonesian using Rasa Framework [57]	2021
Bilingual Chatbot For Covid-19 Detection Based on Symptoms Using Rasa NLU [58]	2022
Design and Implementation of Chatbot Framework For Network Security Cameras[59]	2019
Chatbot development using Java tools and libraries [60]	2020
Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States[61]	2019 Procedia Computer Science

De esta manera al obtener los dos estudios mostrados en la Tabla 16, y los criterios de selección de estudios establecen la pauta de extracción de información relevante.

Por cada uno se sintetizó y clasificó a cada artículo para tener una visión clara de las preguntas que se establecieron para evaluar la calidad de los artículos preseleccionados (Ver Anexo 2)como se describe en la Tabla 17, Tabla 18 y Tabla 19.

Tabla 17. Frameworks por artículo.

Frameworks	Artículo
RASA Framework	[54][55][57][58]
ELIZA	[56]
ALICE	[56]

Tabla 18. Tecnologías por artículo.

Herramientas	Artículo
Rasa	[54][57]
Entity Transformer (DIET)	[55]
JAVA	[56][60]
Plataforma Facebook Messenger	[59]
AIML	[61]

Tabla 19. Arquitectura por artículo.

Arquitectura	Artículo
Verdadero	[55][57][58][61]
Falso	[54][56][59][60]

- Reporte de los resultados.

El reporte obtenido tras la ejecución de la RSL se detalla en el Anexo 2.

6.1.3. Clasificación de la información.

Para llevar a cabo el cumplimiento de la tercera tarea clasificación de la información, se tomaron en cuenta dos subtarefas, que consistieron en la comparación de las tecnologías

encontradas en la primera y segunda tarea, y a partir de esta comparación se seleccionó la tecnología que dio paso a la arquitectura de desarrollo del framework.

6.1.3.1. Comparación de las mejores tecnologías para el desarrollo de un framework de asistentes virtuales.

Para la realización de la tarea se comparan los marcos de chatbot que se presentan en la sección 4.9. del Marco Teórico. Los marcos de chatbot que se analizan tienen como característica común el procesamiento de palabras. Están pensados para detectar la intención del usuario e identificar la entidad a partir de la petición realizada por el usuario. La funcionalidad principal de todos los marcos es la detección de la intención y el reconocimiento de la entidad. En la Tabla 20, se comparan los marcos en función de las siguientes funciones adicionales que ofrecen a los usuarios.

- **Integración:** Tipo de integración con el que cuenta la herramienta.
- **Lenguajes de programación:** Lenguajes que puede soportar la herramienta .
- **Licencia:** Es el precio por el uso de los marcos de trabajo.
- **Type:** Indica si el marco puede procesar el habla y el texto para la conversación.
- **Técnica Implicada:** Tipo de procesamiento del lenguaje con el que cuenta la herramienta.
- **Plataforma:**Entornos y aplicaciones en los que se puede integrar la herramienta.
- **Idiomas:** Indica el número de idiomas que admite el marco de trabajo.
- **Tipo Base de Datos:** Indica la compatibilidad con las bases de datos o si la herramienta puede utilizarse en la nube o no.

En la Tabla 20, se comparan diferentes marcos de chatbot basándose en los criterios mencionados anteriormente. Estos criterios especifican las funcionalidades que ofrece cada marco. Las funcionalidades básicas que ofrecen todos los marcos son el procesamiento de palabras, la detección de intenciones y el reconocimiento de entidades. Aparte de éstas, los marcos ofrecen otras características útiles para facilitar el desarrollo del chatbot. Los criterios discutidos aquí son las características que ayudan al desarrollador a desarrollar los chatbots más rápidamente debido al número de características proporcionadas por el marco.

Tabla 20. Comparación de Frameworks para la construcción de Chatbots

ChatBot Name	BERT	Microsoft Bot Framework	Rasa	Wit AI	Dialogflow
Integración	Librería	Bot Builder SDK Bot Connector Developer Portal Bot Directory SDK API REST	SDK, API REST	Node.js client Python client Ruby client On other platforms: HTTP API	SDK, API REST
Lenguajes de programación	Python	C#, Java, NodeJs ,JavaScript o Python	Python , Node.js, PHP	Python, Ruby,NodeJs	NodeJS
Licencia	Open Source	Free Trial + Paid	Open Source	Free with certain limits	Free Trial
Type	Voice, Text	Voice, Text	Voice, Text	Voice, Text	Voice, Text
Técnica Implicada	NPL ,NSP	NLU	NLP,NLU	NLP,NLU	NLP,NLU
Plataforma	Websites	App ,Websites	Website, telegram, Facebook, Whatsapp	Facebook, Alexa	Websites, Facebook, Alexa, Google Assistant
Idiomas	Más de 102 idiomas	Más de 30 idiomas	Multilingüe	Más de 137 idiomas	Más de 20 idiomas
Tipo Base de Datos	—	Azure SQL	SQL, MongoDB, Redis, DynamoDB	Nube	Firebase (Nube)

ChatBot Name	IBM Watson Conversation Service	Amazon Lex	Pandorabots	Botpress	Botkit	ChatterBot
Integración	SDK, API REST	SDK	API ,SDK	SDK	Librería	Librería

Lenguajes de programación	Java, C++	Java, .Net, Ruby	Java, Ruby, Go, PHP, Python, Node. js	JavaScript, Python , C#	JavaScript servidores de Node	Node js, Python
Licencia	Free trial + Limitado	Paid	Paid	Open Source	Free , No open Source	Open Source
Type	Voice, Text	Voice, Text	Voice, Text	Voice, Text	Text	Voice, Text
Técnica Implicada	NLP,NLU	NPL	NLU , AWS Lambda	NLU	No soporte NLU pero puede usarlas integrando otras herramientas	NPL
Plataforma	Websites, Facebook, SMS, Slack, WhatsApp	SMS	Websites, Facebook, SMS, Slack, eCommerce	Slack Telegram Microsoft Teams Facebook Messenger	Facebook Slack Microsoft Teams	Facebook, WhatsApp
Idiomas	Más de 13 idiomas	Más de 71 idiomas	Multilingüe	Más de 12 idiomas	Multilingüe	Multilingüe
Tipo Base de Datos	Db2, PostgreSQL (Nube)	Db2, PostgreSQL (Nube)	Google Cloud	SQLite (predeterminado) PostgreSQL (alternativa)	Nube	SQLm MongoDB

6.1.3.2. Selección de la tecnología para el desarrollo del framework.

Para la selección de la tecnología se tomó en cuenta las dos tareas anteriores y sus respectivas subtarefas, todo la revisión literaria y sistemática conlleva a un planteamiento de una arquitectura general ver Figura 13, con las tecnologías a usarse en el desarrollo del framework para el despliegue de asistentes virtuales.

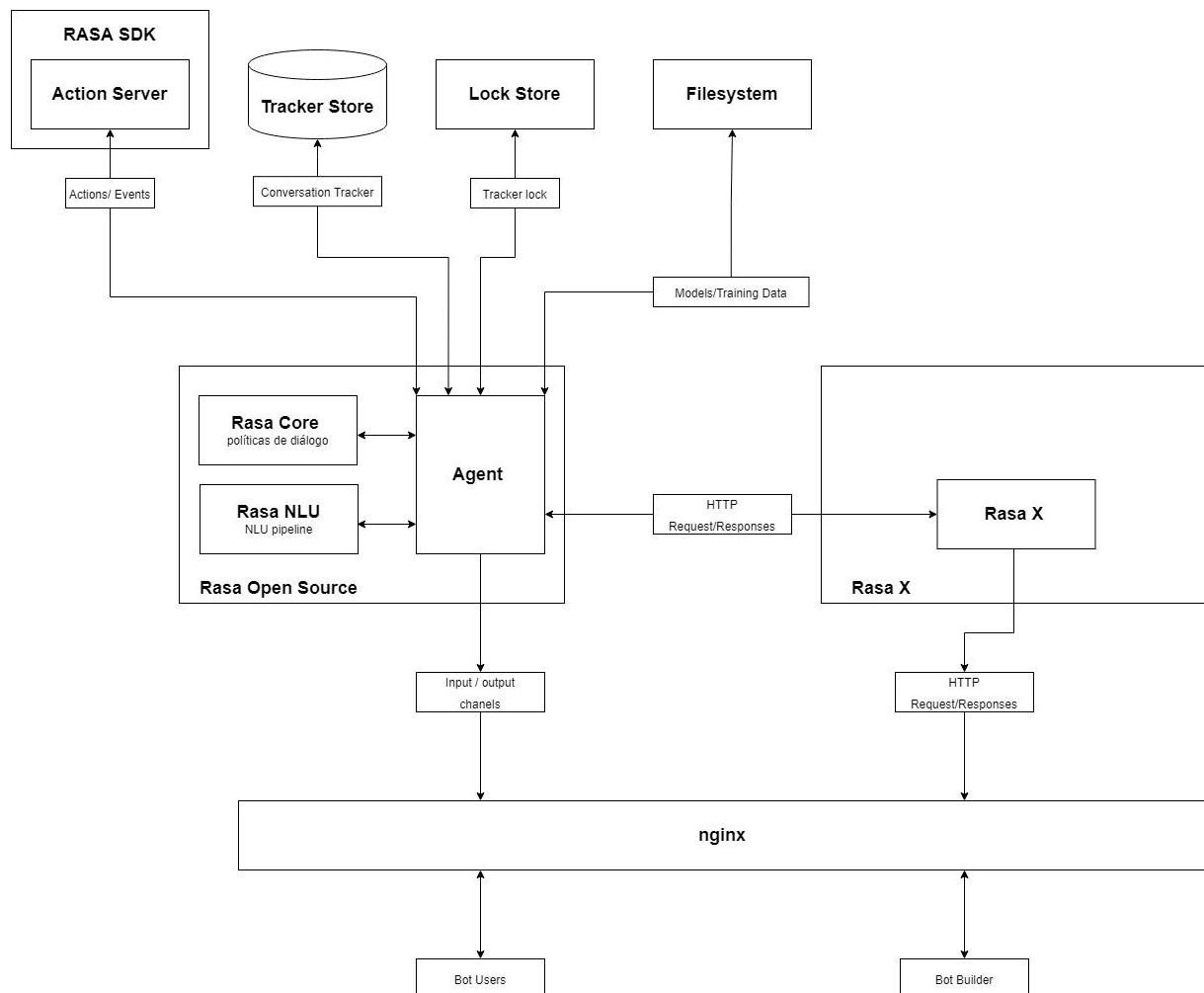


Figura 13. Arquitectura General del Framework de despliegue de asistentes virtuales.

6.1.4. Documentación de cada aspecto investigativo.

6.2. Segunda fase: Construir el framework de despliegue de asistentes virtuales en función de las tecnologías vigentes seleccionadas.

Para el cumplimiento de la primera fase se desarrollaron las siguientes tareas.

6.2.1. Establecimiento del Corpus lingüístico del framework.

Para llevar a cabo el cumplimiento de la primera tarea del establecimiento del Corpus lingüístico se seleccionó el modelo pre-entrenado bert-base-multilingual-cased, el cual se detalla a continuación.

6.2.1.1. Bert-base-multilingual-cased.

Para establecimiento del corpus lingüístico que abarca el framework se tomó el modelo de transformers bert-base-multilingual-cased, siendo este un gran corpus de datos multilingües de manera auto supervisada y entrenado previamente sólo con textos sin procesar, dispone de dos objetivos principales para el desarrollo del framework, siendo estos.

- **Modelo de lenguaje enmascarado (MLM):** tomando una oración, el modelo enmascara aleatoriamente el 15% de las palabras en la entrada, luego se ejecuta la oración enmascarada completa a través del modelo y tiene que predecir las palabras enmascaradas. Esto es diferente de las redes neuronales recurrentes tradicionales (RNN) que generalmente ven las palabras una tras otra, o de los modelos autorregresivos como GPT que enmascaran internamente los tokens futuros. Permite que el modelo aprenda una representación bidireccional de la oración.
- **Predicción de la siguiente oración:** el modelo concatena dos oraciones enmascaradas como entradas durante el pre entrenamiento. A veces corresponden a oraciones que

estaban una al lado de la otra en el texto original, a veces no. Luego, el modelo tiene que predecir si las dos oraciones se estaban siguiendo o no.

De esta forma, el modelo aprende una representación interna de los idiomas en el conjunto de entrenamiento que luego se puede usar para extraer funciones útiles para tareas posteriores: si se tiene un conjunto de datos de oraciones etiquetadas, por ejemplo, puede entrenar un clasificador estándar usando las funciones producidas por el modelo BERT como entradas.

Procedimiento de entrenamiento

preprocesamiento

Los textos están en minúsculas y tokenizados usando WordPiece y un tamaño de vocabulario compartido de 110,000. Los idiomas con una Wikipedia más grande están sub muestreados y los que tienen menos recursos están sobre muestreados. Para idiomas como el chino, el kanji japonés y el hanja coreano que no tienen espacio, se agrega un bloque CJK Unicode alrededor de cada carácter.

Las entradas del modelo son entonces de la forma:

[CLS] Sentence A [SEP] Sentence B [SEP]

Con probabilidad 0.5, la oración A y la oración B corresponden a dos oraciones consecutivas en el corpus original y en los demás casos, es otra oración aleatoria en el corpus. Tener en cuenta que lo que se considera una oración aquí es un tramo de texto consecutivo, generalmente más largo que una sola oración. La única restricción es que el resultado con las dos "oraciones" tiene una longitud combinada de menos de 512 tokens.

Los detalles del procedimiento de enmascaramiento para cada oración son los siguientes:

- El 15% de los tokens están enmascarados.
- En el 80% de los casos, los tokens enmascarados se reemplazan por [MASK].
- En el 10% de los casos, las fichas enmascaradas se reemplazan por una ficha aleatoria (diferente) a la que reemplazan.
- En el 10 % de los casos restantes, los tokens enmascarados se dejan como están.

6.2.2. Codificación del framework.

Para llevar a cabo el cumplimiento de la segunda tarea de la codificación del framework se tomaron en cuenta dos subtareas, que consistieron en el desarrollo del componente inteligente del framework y el desarrollo del backend del framework, cuyos resultados se detallan a continuación.

6.2.2.1. Desarrollar el componente inteligente del Framework.

Para el desarrollo del componente inteligente se tomaron en cuenta las plataformas de código abierto como Rasa y HuggingFace, BERT y otras arquitecturas de transformadores. Por otro lado, los científicos de datos de Rasa han desarrollado un clasificador especial basado en transformadores llamado Transformador de entidades de doble intención o clasificador DIET que está hecho a medida para la tarea de extraer entidades y clasificar intenciones simultáneamente.

El plan para la inteligencia: config.yml

En esta sección se prueban todas las combinaciones posibles de hiper parámetros ver Figura 14, y se elige la configuración que proporcione las métricas más altas, a esto de lo llama búsqueda en cuadrícula ya que las incrustaciones de palabras pre entrenadas de BERT por sí solas no proporcionaron buenos resultados (ver Figura 15). Las incrustaciones BERT pre entrenadas multilingües que usamos fueron entrenadas en texto en mayúsculas en los 104 idiomas principales con las Wikipedias más grandes", según la documentación de HuggingFace.

Sin embargo, el modelo fue entrenado en funciones personalizadas usando DIET y luego combinado con incrustaciones pre entrenadas de BERT en una capa de avance (ver Figura 14), esta opción es ideal para proyectos en español con pocos datos de entrenamiento. El

modelo combinado se desempeñó mejor que el modelo que usó DIET sin incorporaciones pre entrenadas de BERT (diet_without_BERT). Esto significa que para los chatbots que no hablan inglés con una cantidad moderada de datos de entrenamiento.

	precision	recall	f1-score	config
0	0.6545	0.6304	0.6036	SklearnIntentClassifier
0	0.5518	0.5562	0.5234	diet_BERT_only
0	0.8126	0.7824	0.7810	diet_without_BERT
0	0.8193	0.7842	0.7845	diet_BERT_combined

Figura 14. Combinaciones de Hyperparameters

En la Figura 15 se muestra la configuración del archivo config que se utilizó.

```
language: es

pipeline:
  - name: HFTransformersNLP
    model_weights: bert-base-multilingual-cased
    model_name: bert
    cache_dir: null
  - name: LanguageModelTokenizer
  - name: LanguageModelFeaturizer
  - name: RegexFeaturizer
  - name: DIETClassifier
    constrain_similarities: true
    random_seed: 42
    use_masked_language_model: true
    epochs: 100
    number_of_transformer_layers: 4
    transformer_size: 256
    drop_rate: 0.2
    weight_sparsity: 0.7
    batch_size:
      - 64
      - 256
    embedding_dimension: 30
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
    constrain_similarities: true
    model_confidence: linear_norm
```

```

policies:
  - name: MemoizationPolicy
    max_history: 5
  - name: TEDPolicy
    max_history: 5
    epochs: 100
  - name: FormPolicy
  - name: FallbackPolicy
    nlu_threshold: 0.5
    core_threshold: 0.5
    fallback_action_name: action_default_fallback

```

Figura 15. Configuración de la arquitectura DIET

Para integrar BERT o cualquiera de los otros modelos pre entrenados disponibles en el sitio web de HuggingFace , simplemente se reemplaza el hiper parámetro `model_weights` en la siguiente línea con cualquier incrustación pre entrenada que se desea usar, en este caso se hace uso de `bert-base-multilingual-cased` porque era el mejor modelo disponible para español, ver figura 16.

```

- name: HFTransformersNLP
  model_weights: bert-base-multilingual-cased
  model_name: bert

```

Figura 16. Incrustación del hiper parámetro `bert-base-multilingual-cased`.

6.2.2.2. Desarrollar el backend del framework.

Para el desarrollo del backend del framework se utiliza la tecnología de Rasa, version 2.8.9, el lenguaje de programación utilizado por Rasa es Python, donde la versión utilizada fue la 3.8. La instalación de Rasa se realiza introduciendo la siguiente línea de comandos en la terminal:

pip install rasa ==2.8.9

Para iniciar un proyecto Rasa se ejecuta este comando:

rasa init

A continuación se generan los archivos que se muestran en la Figura 17.

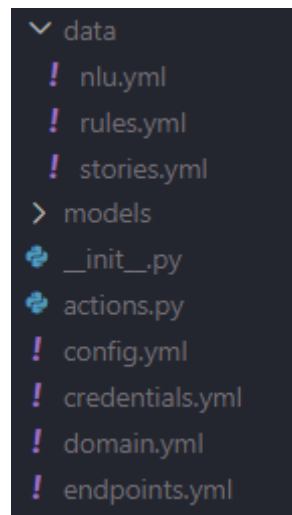


Figura 17. Archivos creados al iniciar un proyecto de Rasa.

Este comando crea todos los archivos necesarios para iniciar un proyecto Rasa y entrena un bot simple con datos de muestra por defecto. El modelo entrenado se guarda en la carpeta models. Los archivos que componen el corpus de entrenamiento son tres: nlu.yml, stories.yml y domain.yml. El formato más utilizado para estos archivos es YAML (Yet Another Markup Language) porque suele ser más fácil de usar, aunque también pueden escribirse en JSON. Las acciones personalizadas se implementan en el archivo de Python llamado actions. A continuación se definen y explican los diferentes archivos que componen el repositorio. Los archivos de referencia presentados no están completos, sirven para visualizar la estructura y el formato que deben tener.

Frases de entrenamiento: nlu.md

Este archivo se compone de los mensajes que el usuario puede escribir. En él debe indicar la intención correspondiente y las entidades que aparecen en la frase. A partir de estas frases de referencia, el bot aprende a identificar intenciones y entidades. Además, hay dos secciones adicionales que son los sinónimos y las expresiones regulares. La primera de ellas permite asociar diferentes entidades con el mismo valor. La segunda mencionada, puede utilizarse para facilitar la extracción de entidades. Por ejemplo, si la entidad tiene una estructura predeterminada (como la señal, cuatro letras seguidas de cuatro dígitos), se utiliza una expresión regular para facilitar la detección de dicha entidad.

```

nlu:
- intent: greet
  examples: |
    - Hola
    - hola
    - buenos días
    - buenas tardes
    - como estas
    - Hola.
    - Vamos.
    - Hola, amigo.
- intent: goodbye
  examples: |
    - buenas por
    - hasta luego
    - buenas noches
    - adiós
    - que tengas un buen día
    - nos vemos
    - adiós, adiós

```

Figura 18. Código de entrenamiento en Rasa.

Dominio: domain.yml

El dominio es el entorno en el que opera el asistente. Incluye las respuestas del bot, el nombre de las acciones personalizadas creadas, los intentos, las entidades y las ubicaciones que el bot necesita conocer. Los nuevos conceptos aquí son las acciones y las ubicaciones personalizadas. Las acciones personalizadas son funciones que pueden ejecutar cualquier código que se desee, por lo que la respuesta no tiene que ser fija y puede ser coherente con el mensaje del usuario. Las ubicaciones son la memoria del bot. Actúan como un almacén de valores clave que se utiliza para almacenar información (entidades) que el usuario ha proporcionado, como el nombre de un token. Normalmente, se quiere que estas ranuras influyan en el diálogo, por lo que son muy utilizadas en las acciones personalizadas. Los diferentes tipos de ubicación son: Texto, Booleano, Categórico, Flotante, Lista y Sin característica.

Por último, en este archivo se puede definir el periodo de inactividad a partir del cual se inicia una nueva sesión de chat. El parámetro `session_expiration_time` define el tiempo de inactividad en minutos. Otro parámetro que se puede declarar en este archivo es `carry_over_slots_to_new_session`. Mediante un valor booleano se determina si se quiere que los slots se olviden entre diferentes sesiones.

```
responses:
  utter_greet:
    - text: ¡Hola! ¿Cómo estás??
  utter_cheer_up:
    - image: https://i.imgur.com/nGF1K8f.jpg
      text: 'Aquí tienes algo para animarte:'
  utter_did_that_help:
    - text: Te ha servido de algo?
  utter_happy:
    - text: Genial, ¡sigue!
  utter_goodbye:
    - text: Bye
  utter_iamabot:
    - text: Soy un bot, impulsado por Rasa.
```

Figura 19. Código de las intenciones y entidades en Rasa.

Historias: `stories.md`

Las historias son simulaciones de una conversación real entre el bot y el usuario. Es la estructura de los datos que se utiliza para entrenar el modelo de gestión del diálogo. Los mensajes del usuario se convierten en las intenciones correspondientes mientras que las respuestas se expresan con el debido nombre de la acción. Se muestra a continuación un ejemplo de una conversación real con el asistente y el formato equivalente que utiliza el bot para entrenarse.

```
stories:
- story: happy path
  steps:
    - intent: greet
    - action: utter_greet
    - intent: mood_great
    - action: utter_happy
```

```

- story: sad path 1
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

```

Figura 20. Código de las historias en Rasa.

Acciones personalizadas: actions.py

Como se ha mencionado anteriormente, si por ejemplo el usuario dice hola, el bot debe responder con un texto de saludo predefinido que se especifica en el archivo de dominio. Sin embargo, la respuesta no siempre es fija, a veces es necesario utilizar la información proporcionada por el usuario para devolver una respuesta de acuerdo con los datos proporcionados. Las acciones personalizadas son respuestas de ayuda que incluyen código personalizado. Este código personalizado puede ir desde una simple respuesta de texto hasta, como es el caso del chatbot desarrollado, conectar con una base de datos externa, extraer los datos deseados y devolver un gráfico. Las funciones personalizadas que se han creado para el bot tienen que ver con el componente FallbackPolicy de la sección de gestión del diálogo, que se encarga de comprobar si se superan o no los umbrales establecidos para la comprensión y gestión del diálogo mediante la predicción, y en caso contrario imprimir la frase definida por defecto. Para configurar esta función, basta con escribir la frase que se quiere devolver por defecto.

Archivos credentials.yml y endpoints.yml

El archivo de credenciales se utiliza para que el asistente esté disponible en plataformas de mensajería como Facebook Messenger, Telegram, etc. Este trabajo no se hizo para estas plataformas, pero hay que tener en cuenta que Rasa permite al bot conectarse a diferentes servicios. Es por ello que dicho archivo no debe ser modificado y debe permanecer como está por defecto. Un endpoint es un extremo de un canal de comunicación. Cuando una API interactúa con otro sistema, ya sea para acceder a recursos o enviar peticiones, los puntos de

contacto de esta comunicación se consideran terminales. En el caso de las API, un terminal puede incluir una URL de un servidor. En este caso, lo único que se ha utilizado y que hay que descomentar es el endpoint de acciones para iniciar el servidor de acciones personalizadas.

Entrenamiento y prueba del modelo

El siguiente paso es el entrenamiento del modelo con las historias y datos del NLU. Para ello, en la última versión (rasa combinada con rasa X) simplemente introduciendo el siguiente comando en directorio del proyecto se entrenan las dos partes del sistema: Rasa NLU y Rasa Core.

rasa train

A continuación también se explica cómo hacer esto en varias partes, y lo que hasta hace poco era la única forma de hacerlo. En algunos casos, se recomienda crear un entorno virtual (una herramienta para proyectos con diferentes dependencias, aislándolas con un entorno virtual), para evitar problemas de compatibilidad de versiones, ya que es fácil tener varias versiones de Python o incluso de Rasa. Al crear el entorno virtual se pueden omitir los siguientes pasos.

Primero, para asegurarte de que tienes Python 3.6, puedes ejecutar:

python3 --version

A continuación, se crea un entorno virtual con los siguientes comandos:

python3 -m venv env

source env/bin/activate

Luego se entrenan los módulos del sistema (NLU y Core) y se crea el servidor local para ejecutar el chatbot. Para ello, se utilizan los siguientes comandos:

make train-nlu

make train-core

make action-server

Finalmente, el último comando se ejecuta en el terminal y el chatbot comienza a ser utilizado.

make cmdline

En este punto se puede comenzar a hablar con el chatbot. Además, se presentaron los conceptos básicos para mejorar y crear nuevos asistentes virtuales.

6.2.3. Documentación técnica de cada aspecto desarrollado en las distintas etapas de la construcción del framework.

6.3. Tercera fase: Validar el framework mediante el despliegue de información académica relacionada a las temáticas dentro de las CIS.

Para el cumplimiento de la tercera fase se desarrollaron las siguientes tareas.

6.3.1. Identificación y recopilación de la fuente de información de las temáticas de la Carrera de CIS.

Para llevar a cabo el cumplimiento de la primera tarea de identificación y recopilación de la fuente de información, se tomaron en cuenta temáticas relacionadas a la carrera de CIS, en este caso del regimen academico de la Universidad Nacional de Loja, especificamente la seccion de matriculas, exceptuando las matriculas de postgrados.

La primera sección aborda el conjunto de datos NLU recolectados para entrenar adecuadamente el pipeline NLU del agente (modelos de aprendizaje automático). Por lo tanto, se recopiló un conjunto de datos de las intenciones que representan los principales casos de uso del chatbot.

Sin embargo, para facilitar la tarea de los voluntarios, se pidió que se anotarán las entidades relacionadas. La recolección del conjunto de datos se llevó a cabo mediante un formulario de Google enviado a una amplia comunidad. En total, 50 voluntarios completaron el formulario, dando ejemplos para cada intención. El formulario no estaba dirigido a la población de usuarios finales, ya que la recopilación de intenciones es independiente de la edad del usuario. El objetivo era crear suficientes ejemplos para que un modelo de aprendizaje automático fuera capaz de generalizar sobre nuevos datos.

El formulario presentaba el chatbot y el sistema de recomendación y explicaba el alcance del trabajo. El formulario se dividía en secciones que se corresponden con las intenciones. El caso de uso o intención y un ejemplo se presentaban al principio de la página y luego se pedía al voluntario que escribiera uno, dos o tres (según la intención) ejemplos de la frase que utilizará para alcanzar el objetivo descrito. Los ejemplos que tenían variabilidad y los que eran más importantes para las acciones principales del chatbot fueron los seleccionados para ser recogidos en el formulario.

De este modo, se recolectaron datos ya divididos por intención y también aumentar el número de ejemplos. Reunir una colección diversa de entidades es importante, ya que se trata de un grupo léxico muy amplio para que el clasificador aprenda. El formulario recogió un total de 45 respuestas. Los datos en bruto se analizaron para eliminar los valores atípicos o las respuestas fuera de lugar. Las palabras mal escritas se mantuvieron para aumentar la solidez del modelo frente a ellos. Además, se anotaron las entidades relacionadas con un total de 45 anotadas en el formato aceptado por Rasa. Tras estos pasos de procesamiento, se añadieron más ejemplos a los extraídos del formulario. Otros intentos menores que no presentan tanta variación como las recogidas en el formulario se añadieron manualmente al archivo de datos NLU.

Además, el análisis de las respuestas del formulario permitió aumentar la percepción de las funciones que deberían estar disponibles en el chatbot. Hubo algunos ejemplos que no se tuvieron en cuenta en el plan inicial del agente. Este análisis es muy importante, ya que permite recoger los detalles utilizados en las solicitudes y que deben ser tratados correctamente por el chatbot.

Los ejemplos incorrectos o fuera de alcance se añadieron a los grupos de intenciones incorrectas o de chats. Teniendo en cuenta esto, el conjunto de datos final para la fase de entrenamiento de NLU contiene 40 muestras, divididas en 23 intenciones y 20 instancias para 40 tipos de entidades diferentes.

Tabla 21. Conjunto de datos finales de intenciones para la fase de entrenamiento NLU.

Intent	Descripción	Número de Ejemplos
afirmativo	El usuario está de acuerdo con algo o da una respuesta positiva respuesta	
concepto_matricula	El usuario pregunta sobre información general de la matrícula	

condicion_estudiante	El usuario pregunta sobre información relacionada a la condición del estudiante	
tipos_de_matriculas	El usuario pregunta sobre los diferentes tipos de matrículas. Se usa botones para desplegar los tipos de matrículas existentes en la universidad	
matricula_ordinaria	El usuario pregunta sobre información de la matrícula ordinaria	
registro_matricula_ordinaria	El usuario pregunta sobre información del proceso de matriculación relacionada con la matrícula ordinaria	
matricula_extraordinaria	El usuario pregunta sobre información de la matrícula extraordinaria	
registro_matricula_extraordinaria	El usuario pregunta sobre información del proceso de matriculación relacionada con la matrícula extraordinaria	
matricula_especial	El usuario pregunta sobre información de la matrícula ordinaria especial	
registro_matricula_especial	El usuario pregunta sobre información del proceso de matriculación relacionada con la matrícula especial	
matricula_internado_rotativo	El usuario pregunta sobre una carrera de la salud que requiera internado rotativo	
anulacion_matricula	El usuario pregunta sobre información general de la anulación de matrícula	
motivo_anulacion_matricula	El usuario pregunta sobre motivos por el cual se pueda anular la matrícula	

como_anular_matricula	El usuario pregunta sobre el proceso para la anulación de la matrícula	
registro_matricula_primer_ciclo	El usuario pregunta sobre el lugar donde se registra la matrícula	
requisitos_matricula_primer_ciclo	El usuario pregunta sobre los requisitos necesarios para la matriculación de primer ciclo	
requisitos_segunda_carrera	El usuario pregunta sobre los requisitos para el registro en una segunda carrera	
requisitos_matricula_segundo_ciclo	El usuario pregunta sobre los requisitos necesarios para la matriculación de segundo ciclo	
requisitos_matricula_estudiantes_extranjeros	El usuario pregunta sobre los requisitos necesarios para la matriculación de los estudiantes extranjeros	
retiro_asignatura	El usuario pregunta información sobre el retiro del estudiante	
requisitos_retiro_asignatura	El usuario pregunta sobre los requisitos necesarios para poder retirarse de una asignatura	
retiro_segunda_tercera	El usuario pregunta sobre los requisitos necesarios para poder retirarse de una segunda o tercera asignatura	
tercera_matricula	El usuario pregunta sobre información de la pérdida de tercera matrícula	

6.3.2. Evaluación del framework a partir de la fuente de información .

6.3.3. Corrección y validación del framework.

6.3.4. Documentación de los resultados obtenidos de la validación del framework.

7. Discusión.

En esta sección se presenta el análisis de los resultados obtenidos de los tres objetivos específicos del trabajo de titulación “Framework para el despliegue de temáticas de asistentes virtuales aplicado a la Carrera de Ingeniería en Sistemas”, se discute cada uno de los resultados argumentando con fuentes bibliográficas previamente analizadas en el transcurso del desarrollo del TT.

7.1. Objetivo 1: Estudiar las tecnologías para la construcción del framework de asistentes virtuales.

Para el cumplimiento del primer objetivo se lo llevó a cabo mediante una Revisión bibliográfica/literaria y conjuntamente con la Revisión Sistemática de Literatura (RSL), permitió contestar las preguntas de investigación planteadas (véase en la sección Resultados, apartado 6.1.1.1., 6.1.1.2., 6.1.2.1. y 6.1.2.2.), al igual que desarrollar bases sólidas de temas con respecto a la Inteligencia Artificial, procesamiento de lenguaje Natural y tecnologías convenientes y actuales que permitan la construcción de frameworks enfocados a los asistentes virtuales.

A partir de la recolección de información, selección de documentos y respectivo análisis de literatura (ver Anexo 1 y 2), se identificó un extenso contexto de las herramientas,

tecnologías, arquitecturas y frameworks que son utilizados en estos últimos 4 años en el desarrollo de frameworks, la Tabla 22 muestra las preguntas planteadas para la primera tarea (Autores en el ámbito de Machine Learning y procesamiento de lenguaje natural) y en la Tabla 23 muestra las preguntas planteadas para la segunda tarea (Tecnologías y elementos técnicos que componen el desarrollo de un framework), así como el análisis de las respuestas elaboradas del producto de investigación realizado.

Tabla 22. Análisis de las respuestas de la primera tarea de Revisión bibliográfica/literaria y Sistemática de Literatura.

Preguntas de investigación	Tipo de Revisión	Análisis de la Respuesta
¿Casos de éxito de trabajos de desarrollo frameworks de chatbots?	Revisión bibliográfica / literaria	<p>Han sido varias las contribuciones exitosas en el desarrollo de frameworks de chatbots, en las que se destacan dos trabajos que muestran marcos de trabajo y modelos, como lo es el trabajo de [49], que aplica un marco de trabajo centrado en el reconocimiento y predicción de entidades, respuestas a preguntas y sistemas de diálogo que proporcionan datos robustos en cuestión del sistema.</p> <p>En cuanto a otro caso de éxito, se tiene el trabajo de [50], con el desarrollo una arquitectura de framework para sistemas de generación de lenguaje natural (GLN), realizan la utilización de metodologías en el desarrollo de software, esto mediante la orientación a objetos y los patrones de diseño permiten resolver problemas en las aplicaciones GLN.</p>
¿Casos de éxito de herramientas para el desarrollo de chatbots?	RSL	Es notable considerar según [69][70] las técnicas de aprendizaje automático pueden definirse como un conjunto de métodos capaces de detectar de forma automática una serie de patrones, bajo esta definición el machine learning lleva existiendo desde hace 50 años, periodo en el que se definieron los métodos estadísticos y se aplicaron al machine learning mediante simples algoritmos.
¿Tipos de procesamiento de lenguaje natural implicados?	RSL	<p>En la elaboración de una Revisión Sistemática de Literatura, evidenciando de esta forma que los tipos de PLN en desarrollo de chatbots van de la mano con machine learning [48].</p> <p>A continuación se pudo evidenciar de manera objetiva que los PLN sí son aplicables en el ámbito del</p>

		desarrollo de chatbots como estructura principal para el desarrollo de frameworks [47].
¿Existen métodos o estándares?	RSL	Según [47][48], todas las entradas que realiza el usuario al chatbot son netamente textuales y no habladas, no obstante también se destaca y diferencia de algunos estudios el chatbot si puede devolver una respuesta hacia el usuario ya sea en forma textual o hablada. Mencionado esto, son varias las herramientas encontradas en los distintos artículos, pero no todas resultan útiles para el objeto de estudio en esta revisión. Se puede apreciar que la herramienta usada con más frecuencia para la interpretación de la entrada textual de usuario es BERT [47][48].

Tabla 23. Análisis de las respuestas de la segunda tarea de Revisión bibliográfica/literaria y Sistemática de Literatura.

Preguntas de investigación	Tipo de Revisión	Análisis de la Respuesta
¿Tecnologías usadas para el desarrollo de frameworks de despliegue de chatbots?	Revisión bibliográfica / literaria	Como se puede evidenciar en [31][32][33][34][35][36][37][38][39][40][42][43][44][45][46] a lo largo de los años la tecnología ha tenido un crecimiento exponencial, naciendo la necesidad de usar un marco desarrollo, porque combinan la funcionalidad y desarrollo, todo esto implementado en un lenguaje de programación único que permite a los desarrolladores crear fácil y rápidamente funcionalidades de diferente índole y de acuerdo a sus necesidades. El framework está diseñado precisamente para evitar estos problemas y generar aplicaciones a partir de la calidad.
¿Tecnologías usadas en el desarrollo de chatbots?	RSL	Se muestran algunos datos relacionados a las tecnologías empleadas en el desarrollo de frameworks de chatbot que han sido desarrollados en cada uno de los estudios primarios, son varias las herramientas encontradas en los distintos artículos, pero no todas resultan útiles para el objeto de estudio en esta revisión.
¿Frameworks usados para construir Chatbots?	RSL	Lo que destaca de BERT es primero por su facilidad de entender que BERT significa Representaciones de codificador bidireccional de transformadores. Cada palabra aquí tiene un significado. Por ahora, la

		conclusión clave de esta línea es que BERT se basa en la arquitectura Transformer [58]. En segundo lugar, BERT es un modelo "profundamente bidireccional". Bidireccional significa que BERT aprende información tanto del lado izquierdo como del lado derecho del contexto de un token durante la fase de entrenamiento. Y finalmente, el aspecto más impresionante de BERT es poder ajustarlo agregando solo un par de capas de salida adicionales para crear modelos de última generación para una variedad de tareas de NLP [59].
¿Metodología para desarrollar Frameworks?	RSL	Se muestran algunos datos relacionados a las arquitecturas empleadas del chatbot que ha sido desarrollado en cada uno de los estudios primarios. La cual se utiliza para realizar un análisis de las tecnologías y métodos aplicados. En lugar de depender de estructuras lógicas si/entonces, RASA implementa modelos de aprendizaje automático no sólo para reconocer identidades e intenciones, sino también para determinar el flujo de la conversación. Esto convierte a RASA en el sueño de los desarrolladores de chatbots cuando crean experiencias de conversación altamente sofisticadas que evolucionan continuamente sin necesidad de intervención manual.
¿Herramientas para el desarrollo de frameworks?	RSL	Se puede apreciar que la herramienta usada con más frecuencia para el desarrollo de frameworks de Chatbots es RASA, seguida de JAVA, Entity Transformer, IML y una no tan aplicada comúnmente la plataforma Facebook Messenger [55][56][57][58][59][60][61].

7.2. Objetivo 2: Construir el framework de despliegue de asistentes virtuales en función de las tecnologías vigentes seleccionadas.

Para el cumplimiento del segundo objetivo se llevó a cabo mediante de desarrollo del framework, consistiendo en el componente inteligente, backen y conjuntamente con el establecimiento del corpus lingüístico, permitió obtener un producto o base de desarrollo de asistentes virtuales (véase en la sección Resultados, apartado 6.2.1. y 6.2.2.).

Con la información que se obtuvo del objetivo 1, se pudo llevar a cabo la selección del framework de trabajo de rasa para la automatización de conversaciones basadas en texto la cual cuenta con una arquitectura de transformador multitarea, liviana y de última generación para NLU: Dual Intent and Entity Transformer (DIET) la cual maneja la clasificación de intenciones y el reconocimiento de entidades juntos. Brindando la capacidad de conectar y usar varias incrustaciones previamente entrenadas como BERT, GloVe, ConveRT, etc.

Posteriormente se hizo la selección del modelo BERT “bert-base- multilingual-cased” el cual está pre-entrenado en los 104 idiomas principales con la Wikipedia más grande utilizando un objetivo de modelado de lenguaje enmascarado (MLM). Este conjunto de datos contiene muchos pesos BERT populares recuperados directamente en el repositorio de modelos de Hugging Face y alojados en Kaggle contando con (104 idiomas, 12 capas, 768 ocultos, 12 cabezas, 110 millones de parámetros).

Posteriormente se diseñó la arquitectura donde se muestran todos los elementos que se utilizaron para la implementación como se puede ver en la Figura 13. Luego se realizó el proceso de codificación donde se utilizaron los lenguajes de programación y los frameworks, Rasa, como marco de trabajo para el desarrollo del chatbot que permite al usuario entrenar el modelo y agregar acciones personalizadas, Python, para la construcción de las intenciones, entidades que constituirán la base de datos del chatbot, Rasa X, que facilita la GUI del chatbot, bert base multilingual cased, para entender el significado del lenguaje ambiguo en el texto utilizando el texto circundante para establecer el contexto en las preguntas que el usuario pueda hacer al chatbot. Por último, se realizaron pruebas de rendimiento, carga y estrés.

7.3. Objetivo 3: Validar el framework mediante el despliegue de información académica relacionada a las temáticas dentro de la CIS.

8. Conclusiones.

Cumplidos los objetivos en el presente trabajo de titulación bajo la premisa Framework para el despliegue de temáticas de asistentes virtuales aplicado a la Carrera de Ingeniería de Sistemas, se establecen las siguientes conclusiones.

- Teniendo como resultados una revisión sistemática de la literatura de los datos relevantes sobre las tecnologías y métodos utilizados en frameworks de chatbot, dicho esto se puede concluir de los estudios considerados como relevantes, se evidencia que la aplicación de distintas tecnologías de frameworks si son aplicables al proceso de desarrollo de chatbots a nivel estándar siendo la herramienta más usada por su capacidad de personalización y su riqueza en desarrollo para crear bots es RASA conjuntamente con interpretación de la entrada del usuario que es BERT.
- Durante la realización de este trabajo se alcanzaron con éxito los objetivos inicialmente planteados. Se exploró el estado actual de la creación de chatbot y se investigaron diferentes principios de diseño para aplicarlos al proyecto. Posteriormente, todos estos conocimientos adquiridos se reflejaron en el diseño e implementación del asistente virtual, implementando todas las funcionalidades previstas inicialmente, se recopilaron las conversaciones mantenidas por los usuarios con el servicio, de manera que se pudo trabajar en base a esta información, para corregir y mejorar el servicio. Gracias a la recopilación de datos, fue posible recoger y responder a nuevas preguntas que no estaban previstas inicialmente, añadiendo una mejora notable a la base de conocimientos del chatbot.
- Las pruebas de rendimiento han demostrado que los requisitos y las funciones implementadas en el chatbot funcionan correctamente, sin embargo, estas pruebas se realizaron en un entorno controlado, por lo que si este chatbot se implementará en un entorno real, se podrían detectar algunos problemas o errores a la hora de corregirlos.

9. Recomendaciones

10. Bibliografía.

- [1] J. Ortega, Dinarle Guevara, María Benavides, “Elementary: Un Framework De Programación Web,” *Télématique*, vol. 15, no. 2, pp. 144–171, 2016, [Online]. Available: <https://www.redalyc.org/pdf/784/78457627004.pdf>

- [2] D. Biancha Gutiérrez, G. Camacho Sánchez, and G. Martínez Villalobos, "Design Framework for the Development Dynamic Web Applications," *Sci. Tech.*, vol. XVI, pp. 178–183, 2010, [Online]. Available: <https://www.redalyc.org/pdf/849/84917316032.pdf>
- [3] R. Johnson and B. Foote, "Designing reusable classes," *J. Object-Oriented Program.*, vol. 1, no. 2, pp. 22–30, 35, 1988.
- [4] P. Mutjé and D. Escuela, "Agradecimientos Resumen," pp. 1–64, 2015.
- [5] J. Diego Cruz Freire Darwin Vinicio Chimbo Chimbo, "Escuela Superior Politécnica De Chimborazo," 2015.
- [6] I. Hernán Fabricio Naranjo Ávalos Mg ii, "LÍNEA DE INVESTIGACIÓN: Desarrollo de Software AUTOR: Gamboa Teneta, Erick Daniel TUTOR: Ing. Naranjo Ávalos, Hernán Fabricio Mg.," 2019.
- [7] V. Hristidis, "Chatbot Technologies and Challenges," 2018 First International Conference on Artificial Intelligence for Industries (AI4I), 2018, pp. 126–126, doi: 10.1109/AI4I.2018.8665692.
- [8] A. Bozzon, "Enterprise Crowd Computing for Human Aided Chatbots," 2018 IEEE/ACM 1st International Workshop on Software Engineering for Cognitive Services (SE4COG), 2018, pp. 29–30.
- [9] J. Rodríguez, H. Merlino, and E. Fernández, "Comportamiento Adaptable de Chatbots Dependiente del Contexto," *Rev. Latinoam. Ing. Softw.*, vol. 2, no. 2, p. 115, 2014, doi: 10.18294/relais.2014.115-136.
- [10] O. Zuñiga, "Implementación de un Chatbot con BotFramework: Caso de estudio, servicios a clientes del área de finanzas de seguros equinoccial.,"
- [11] N. Al-Madi, K. Maria, M. Al-Madi, M. Alia and E. Maria, "An Intelligent Arabic Chatbot System Proposed Framework," 2021 International Conference on Information Technology (ICIT), 2021, doi: 10.1109/ICIT52682.2021.9491699.
- [12] G. Wilcock, *Introduction to linguistic annotation and text analytics*, vol. 2, no. 1. 2009. doi: 10.2200/S00194ED1V01Y200905HLT003.
- [13] J. Listerri, "HLT Central; Language Technology World; Listerri y Martí," pp. 9–71, 1997, [Online]. Available: http://liceu.uab.es/~joaquim/publicacions/TecnoLing_Lynx02.pdf
- [14] Christopher D. Manning, "Speech and Language Processing: An introduction to natural language processing," *SPEECH Lang. Process. An Introd. to Nat. Lang. Process. Comput. Linguist. Speech Recognit.*, pp. 1–18, 2021, [Online]. Available: <http://www.cs.colorado.edu/~martin/slp.html>
- [15] J. Vilares Ferro, "Aplicaciones del procesamiento del lenguaje natural en la recuperación de información en español," *Proces. del Leng. Nat.*, vol. 36, no. 2006, pp. 57–58, 2006, [Online]. Available: https://rua.ua.es/dspace/bitstream/10045/1258/1/PLN_36_06.pdf
- [16] I. González Carrasco, "Análisis, optimización y evaluación de modelos de redes de neuronas artificiales para la clasificación y predicción de impactos de alta

- velocidad sobre distintos materiales,” Feb. 2010, Accessed: Mar. 20, 2022. [Online]. Available: <https://e-archivo.uc3m.es/handle/10016/7471>
- [17] R. F. López and J. M. F. Fernández, *Las Redes Neuronales Artificiales*, Primer Edi. España: Netbiblo, 2008. [Online]. Available: <https://books.google.es/books?id=X0uLwi1Ap4QC>
- [18] M. Arbib, “The Handbook of Brain Theory and Neural Networks,” Editor. Advis. Board, vol. Second Edi, 2003.
- [19] L. Sucar, “Introduction to bayesian networks and influence diagrams,” *Decis. Theory Model. Appl. Artif. Intell. Concepts Solut.*, pp. 9–32, 2011, doi: 10.4018/978-1-60960-165-2.ch002.
- [20] E. Horvitz, D. Hovel, and C. Kadie, “MSBNx: A Component-Centric Toolkit for Modeling and Inference with Bayesian Networks,” 2001. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/msbnx-a-component-centric-toolkit-for-modeling-and-inference-with-bayesian-networks/>
- [21] J. Puga, J. García, L. De, F. Sánchez, I. De, and F. Solana, “Las redes bayesianas como herramientas de modelado en psicología *,” *An. Psicol.*, vol. 23, pp. 307–316, 2007.
- [22] X. Yue Xiaoguang, G. Di Guangzhi, Y. Yu Yueyun, W. Wang, and H. Shi, “Analysis of the combination of natural language processing and search engine technology,” *Procedia Eng.*, vol. 29, pp. 1636–1639, 2012, doi: 10.1016/j.proeng.2012.01.186.
- [23] L. Jiang, C. Li Chaoqun, S. Wang, and L. Zhang, “Deep feature weighting for naive Bayes and its application to text classification,” *Eng. Appl. Artif. Intell.*, vol. 52, pp. 26–39, 2016, doi: <https://doi.org/10.1016/j.engappai.2016.02.002>.
- [24] A. Maletti, “Survey: Finite-state technology in natural language processing,” *Theor. Comput. Sci.*, vol. 679, pp. 2–17, 2017, doi: <https://doi.org/10.1016/j.tcs.2016.05.030>.
- [25] D. Calì, A. Condorelli, S. Papa, M. Rata, and L. Zagarella, “Improving intelligence through use of Natural Language Processing. A comparison between NLP interfaces and traditional visual GIS interfaces,” *Procedia Comput. Sci.*, vol. 5, pp. 920–925, 2011, doi: <https://doi.org/10.1016/j.procs.2011.07.128>.
- [26] S. Jung, “Semantic vector learning for natural language understanding,” *Comput. Speech Lang.*, vol. 56, pp. 130–145, 2019, doi: <https://doi.org/10.1016/j.csl.2018.12.008>.
- [27] A. Nathan and A. Scobell, *Machine Learning Reinforced: Foundations, Algorithms, and Applications*, vol. 91, no. 5. 2012.
- [28] P. Langley and J. G. Carbonell, *Approaches to machine learning*, vol. 35, no. 5. 1984. doi: 10.1002/asi.4630350509.
- [29] H. Rios, “Desarrollo de un Framework para la identificación del Nivel de Complejidad de Texto para el Entrenamiento de Chatbots basado en Machine Learning Tesis para obtener el título profesional de Ingeniero Informático,” 2022.

- [30] A. González-Marcos Fernando Alba-Elías, "Machine Learning En La Industria: El Caso De La Siderurgia," *Econ. Ind.*, pp. 55–63, 2017, [Online]. Available: <http://www.minetad.gob.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/405/GONZALEZ MARCOS Y ALBA ELÍAS.pdf>
- [31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- [32] Vegesna, A.; Jain, P.; Porwal, D. Ontology based Chatbot (For E-Commerce Website). *Int. J. Comput. Appl.* 2018, 179, 51–55.
- [33] Mantha, Mady. Introducing DIET: state-of-the-art architecture that outperforms fine-tuning BERT and is 6X faster to train. Rasa Research website.
- [34] Nimavat, K.; Champaneria, T. Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. *Int. J. Sci. Res. Dev.* 2017, 5, 1019–1026
- [35] Handoyo, E.; Arfan, M.; Soetrisno, Y.A.A.; Somantri, M.; Sofwan, A.; Sinuraya, E.W. Ticketing Chatbot Service using Serverless NLP Technology. In *Proceedings of the 2018 5th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2018, Semarang, Indonesia, 27–28 September 2018*; pp. 325–330.
- [36] Qaffas, A.A. Improvement of Chatbots Semantics Using Wit.ai and Word Sequence Kernel: Education Chatbot as a Case Study. *Int. J. Mod. Educ. Comput. Sci.* 2019, 11, 16–22.
- [37] Braun, D.; Hernandez-Mendez, A.; Matthes, F.; Langen, M. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In *Proceedings of the SIGDIAL 2017—18th Annual Meeting of the Special Interest Group on Discourse and Dialogue 2017, Saarbrücken, Germany, 15–17 August 2017*; pp. 174–185.
- [38] Orth, A. Building Chatbots with Dialogflow and Grank.AI in Grank.ai. 2017. Available online: <https://blog.grakn.ai/chatbotsand-grakn-ai-67563c64cfde> (accessed on 10 March 2021).
- [39] D. A. Ferrucci, «Introduction to This is Watson,» *Journal of Research and Debelopment*, vol. 56, 2012
- [40] M. Anusuya and S. Katti, "Speech Recognition by Machine, A Review," *International Journal of Computer Science and Information Security*, vol. 6, 01/13 2010.
- [41] A. Vandana¹, D. J. Rani, V. A. Goud, M. C. K. K. Reddy, and B. V. R. Murthy, "A STUDY OF CHATBOTS THROUGH ARTIFICIAL INTELLIGENCE."
- [42] "AIML Fundamentals · Pandorabots Documentation." [Online]. Available: <https://pandorabots.com/docs/aiml/aiml-fundamentals.html>. [Accessed: 01-Feb-2018].
- [43] "Guides and references for all you need to know about botpress." <https://botpress.com/docs>. Última visita: Abril 2021.

- [44] Botpress - Un Marco de Desarrollo de Chatbot Maker & Bot. (2017). Recuperado 10 de mayo de 2022, a partir de <https://botpress.io/>
- [45] Belfin, R. V., Shobana, A. J., Manilal, M., Mathew, A. A. and Babu, B. (2019). A Graph Based Chatbot for Cancer Patients. 2019 5th International Conference on Advanced Computing Communication Systems. ID:1, pp. 717–721. ISBN:2469-5556
- [46] Liu, Y., Liu, M., Wang, X., Wang, L., Li, J.: Pal: a chatterbot system for answering domain-specific questions. In: ACL (Conference System Demonstrations)
- [47] E. Amer, A. Hazem, O. Farouk, A. Louca, Y. Mohamed, and M. Ashraf, "A proposed chatbot framework for covid-19," in 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC).
- [48] Kalia, Anup , Vukovic, Maja, Pandita, Rahul, and Singh, Munindar P., "A conceptual framework for engineering chatbots," IEEE Internet Computing, vol. 22, no. 6, pp. 54–59, 2018.
- [49] J. Sai Sharath and R. Banafsheh, "Conversational question answering over know-ledge base using chat-bot framework," in 2021 IEEE 15th International Conference on Semantic Computing (ICSC), pp. 84–85, 2021.
- [50] C. García Ibáñez and R. Hervás, "Una Arquitectura Software para el Desarrollo de Aplicaciones de Generación de Lenguaje Natural," Sociedad Española para el Procesamiento del Lenguaje Natural, Procesamiento de Lenguaje Natural, vol. 33, pp. 111–118 ST – Una Arquitectura Software para el De, 2004
- [51] CES, «Reglamento de Régimen Académico,» pp. 1-53, 2019
- [52] M. Genero Bocco, J. A. Cruz Lemus y M. G. Piattini Velthuis, «Métodos de investigación en ingeniería del software,» 2014.
- [53] R. Antonio Soto Pizarro, "Evolución de frameworks de caja blanca hacia caja negra basándose en lenguaje de patrones.," pp. 1–64, 2006.
- [54] S. Meshram, N. Naik, M. VR, T. More and S. Kharche, "College Enquiry Chatbot using Rasa Framework," 2021 Asian Conference on Innovation in Technology (ASIANCON), 2021, pp. 1-8, doi: 10.1109/ASIANCON51346.2021.9544650.
- [55] L. Fauzia, R. B. Hadiprakoso and Girinoto, "Implementation of Chatbot on University Website Using RASA Framework," 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2021, pp. 373-378, doi: 10.1109/ISRITI54043.2021.9702821.
- [56] N. A. Al-Madi, K. A. Maria, M. A. Al-Madi, M. A. Alia and E. A. Maria, "An Intelligent Arabic Chatbot System Proposed Framework," 2021 International Conference on Information Technology (ICIT), 2021, pp. 592-597, doi: 10.1109/ICIT52682.2021.9491699.
- [57] L. W. Narendra and E. R. Setyaningsih, "Designing a Transactional Smart Assistant in Indonesian using Rasa Framework," 2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE), 2021, pp. 1-6, doi: 10.1109/ICEEIE52663.2021.9616946.

- [58] Y. Mellah, T. Bouchentouf, N. Rahmoun and M. Rahmoun, "Bilingual Chatbot For Covid-19 Detection Based on Symptoms Using Rasa NLU," 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2022, pp. 1-5, doi: 10.1109/IRASET52964.2022.9738103.
- [59] T. Van Cuong and T. M. Tan, "Design and Implementation of Chatbot Framework For Network Security Cameras," 2019 International Conference on System Science and Engineering (ICSSE), 2019, pp. 324-328, doi: 10.1109/ICSSE.2019.8823516.
- [60] A. Ilić, A. Ličina and D. Savić, "Chatbot development using Java tools and libraries," 2020 24th International Conference on Information Technology (IT), 2020, pp. 1-4, doi: 10.1109/IT48810.2020.9070294.
- [61] R. Sutoyo, A. Chowanda, A. Kurniati, and R. Wongso, "Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States," *Procedia Comput. Sci.*, vol. 157, pp. 621–628, 2019, doi: <https://doi.org/10.1016/j.procs.2019.08.226>.
- [62] J. D. Velásquez and J. D. Velásquez, "Una Guía Corta para Escribir Revisiones Sistemáticas de Literatura Parte 1," *DYNA*, vol. 81, pp. 9–10, oct 2014.
- [63] J. D. Velásquez and J. D. Velásquez, "Una Guía Corta para Escribir Revisiones Sistemáticas de Literatura Parte 3," *DYNA*, vol. 82, pp. 9–12, feb 2015.
- [64] J. D. Velásquez Henao and J. D. V. Henao, "Una Guía Corta para Escribir Revisiones Sistemáticas de Literatura. Parte 4," *DYNA*, vol. 82, pp. 9–12, may 2015.
- [65] V. Asensi and A. Parra, "El método científico y la nueva filosofía de la ciencia," *An. Doc.*, vol. 5, pp. 9–19, 2002, [Online]. Available: <https://www.redalyc.org/pdf/635/63500001.pdf>
- [66] C. Deiana, D. Granados -María, and F. Sardella, "El método científico," tech. rep., Argentina, 2018.
- [67] M. Centeno-Romero, G. Chacón-Arrieta, J. A. Vega-Aguilar, A. Gonzalez-Torres, and J. Leitón-Jiménez, "Revisión sistemática de literatura," *Rev. Tecnol. en Marcha*, no. 1, pp. 60–69, 2021, doi: 10.18845/tm.v34i2.4982.
- [68] D. Carrizo and C. Moller, "Estructuras metodológicas de revisiones sistemáticas de literatura en Ingeniería de Software: un estudio de mapeo sistemático," *Ingeniare. Rev. Chil. Ing.*, vol. 26, pp. 45–54, 2018, doi: 10.4067/s0718-33052018000500045.
- [69] J. Calvo, M. Guzmán, and D. Ramos, "Machine Learning, una pieza clave en la transformación de los modelos de negocio," *Manag. Solut.*, p. 144, 2018, [Online]. Available: <https://www.managementsolutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf>.
- [70] Faggella, D. (2018). 7 Applications of Machine Learning in Pharma and Medicine. [online] TechEmergence. Available at:

<https://www.techemergence.com/machinelearning-in-pharmamedicine/> [Accessed 16 May 2018].

[71] Rosas, Maria Veronica, "Un framework de ingeniería del lenguaje para el pre-procesado semántico de textos," vol. 16, no. 1, pp. 587-596, 2015.

[72] Kalia, Anup, Vukovic, Maja, Pandita, Rahul, and Sing, Munindar P., "A conceptual framework for engineering chatbots," IEEE Internet Computing, vol. 22, no. 6, pp. 54-59, 2018.

[73] J. Cahn, "CHATBOT : Architecture, Desing, Development," pp. 45-46, 2017.

[74] K. Chowdhary, Natural Language Processing, pp. 603-604. New Delhi: Springer India, 2020.

[75] Marc Solé Fonte, "Implementación de un chatbot mediante una arquitectura serveless," 2019.

[76] Figueroa Sacoto Stanlin, "Universidad Politécnica Salesiana Sede Cuenca" Diseño Y Desarrollo De Un Chatbot Usando Redes Neuronales Recurrentes Y Procesamiento De Lenguaje," 2021.

1. Anexos.

11.1 Anexo 1: Revisión Sistemática de Literatura.

Autores en el ámbito del Machine Learning y procesamiento de lenguaje natural: Revisión Sistemática de Literatura

Agreda-Sánchez Francisco, Auquilla-Villamagua Jorge
Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas
Loja - Ecuador
(francisco.c.agreda, jorge.auquilla)@unl.edu.ec

Resumen — Este artículo describe un estudio sobre el mapeo y revisión sistemática de literatura (RSL) con el fin de identificar y analizar los autores en el ámbito del Machine Learning y procesamiento de lenguaje natural (PLN). Para el desarrollo de la RSL se empleó la metodología propuesta de Bárbara Kitchenham, la misma que permitió seleccionar 2 artículos que presentan información relacionada al objeto de estudio. La selección de los artículos se realizó acorde a un rango máximo de 5 años desde su publicación de acuerdo a los criterios de inclusión; los estudios se obtuvieron de las bases de datos IEEE Xplore, Science@Direct y ACM Digital Library. Para realizar la presentación de los resultados, se elaboraron tablas que muestran datos relevantes sobre las herramientas, estándares y tipos de PLN que utilizan los autores en el ámbito del machine learning y PLN.

Palabras Clave —Machine Learning; Herramientas; Chatbot; PLN; Estándares; Asistente virtual; Aprendizaje Automático.

Abstract — This article describes a study on mapping and systematic literature review (RSL) in order to identify and analyze authors in the field of Machine Learning and Natural Language Processing (PLN). For the development of the RSL, the methodology proposed by Bárbara Kitchenham was used, which allowed the selection of 2 articles that present information related to the object of study. The articles were selected according to a maximum range of 5 years from their publication according to the inclusion criteria; studies were obtained from the IEEE Xplore, Science@Direct, and ACM Digital Library databases. To present the results, tables were prepared that show relevant data on the tools, standards and types of PLN used by the authors in the field of machine learning and PLN.

Keywords —Machine Learning; Tools; Chatbot; PLN; Natural language processing; Standards; Virtual assistant.

I. INTRODUCCIÓN

“El Machine Learning es un campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente” [1]. Se dice que un programa de computadora aprende de la experiencia con respecto a alguna clase de tareas y medida de desempeño, si su desempeño en las tareas, medido por, mejora con la experiencia” [2].

Para explicar cómo el aprendizaje automático es un elemento clave en el desarrollo del mundo se tiene que volver a sus inicios, ya que esta herramienta es un derivado de la inteligencia artificial [3]. La aplicación de este material se remonta en el pasado, más precisamente a 1943, cuando el matemático Walter Pitts y el neurocientífico Warren

McCulloch, quienes presentaron su trabajo de lo que hoy se conoce como inteligencia artificial, debido a la teoría del análisis cerebral que propusieron, dando a conocer al cerebro como un organismo informático en evolución y la creación computadoras que funciona tan bien o mejor que las redes neuronales humanas [4].

Pero no fue hasta los primeros meses de 2006 que el aprendizaje automático tuvo un gran comienzo. Ascendiendo a la vanguardia de la computación y el procesamiento, desde entonces de la mano de grandes empresas IBM y Microsoft, comenzaron a expandirse globalmente. Y así fue en 2008 en Microsoft lanzó una versión beta del programa Azure Machine Learning, una herramienta que brinda servicios para sus usuarios con la posibilidad de almacenar aplicaciones directamente en el centro. El procesamiento de Microsoft permite ejecutar también un mayor nivel de confidencialidad que muchos otros sistemas de la época. Luego, tres años después Industria IBM también revolucionará el aprendizaje automático con el lanzamiento de su computadora Watson [5].

Para hablar sobre el procesamiento del lenguaje natural, se necesita hablar en primer lugar del lenguaje natural y sus dimensiones en el medio social. Lenguaje natural significa lenguaje hablado y escrito. El objetivo es la comunicación entre una o varias personas, es más directo el lenguaje natural para expresar lo que se quiere transmitir propenso a malentendidos cuando el término se usa con sentido. La comunicación es importante en el lenguaje natural porque este proceso consiste en enviar y recibir información [6].

El lenguaje natural es un fenómeno muy complejo, pero generalmente se ha demostrado que la expresión del lenguaje humano obedece a un conjunto de reglas. Todas las expresiones humanas están claramente organizadas: las palabras en una oración están asociadas con descripciones de cosas y acciones que pueden ser complicadas. El objetivo del analizador es precisamente encontrar estas asociaciones entre palabras, llamadas estructuras sintácticas. Un analizador es un programa que toma una oración como entrada e intenta encontrar una estructura de sintaxis que describa la relación entre las palabras de esa oración. El analizador busca la estructura correcta en un conjunto de posibilidades de análisis, este conjunto suele estar definido por la gramática. El modelo de lenguaje en el que se basa el analizador determina los componentes sintácticos de una oración y cómo se relacionan [7].

El procesamiento del lenguaje natural (PNL) es una subdisciplina de la inteligencia artificial y la rama de ingeniería de la lingüística computacional, ahora, el objetivo principal de PLN es construir sistemas y mecanismos que permitan la comunicación entre el hombre y la máquina a través de lenguajes naturales [8].

II. MÉTODO DE INVESTIGACIÓN

En [9] menciona que toda investigación debe estar basada en evidencias que permitan identificar cómo ha sido abordado el objeto de investigación por otros autores. Sin embargo, es necesario buscar y agregar evidencias usando estudios secundarios como son las RSL y estudios de mapeo sistemático [10]. Para esta investigación se utilizó el protocolo definido por [11] y se estructuró el documento en referencia a [12] [13] [14] [15].

A. Preguntas de Investigación

A partir de la temática central denominada “Autores en el ámbito del Machine Learning y procesamiento de lenguaje natural” se planteó tres preguntas de investigación clasificadas en preguntas para el mapeo sistemático (MQ) y la revisión sistemática (RQ):

- RQ1: ¿Casos de éxito de herramientas para el desarrollo de chatbots?
- RQ2: ¿Tipos de procesamiento de lenguaje natural implicados?
- RQ3: ¿Existen métodos o estándares?

B. Proceso de Búsqueda

Se determinaron un conjunto de términos basados en las preguntas de investigación para construir la cadena de búsqueda. Se utilizó el método PICOC propuesto por [16] para definir el ámbito de la RSL:

- Población (P): “Autores en Machine Learning, Autores en Procesamiento de Lenguaje Natural”.
- Intervención (I): “Machine Learning y Procesamiento de Lenguaje Natural”.
- Comparación (C): “Conocimiento general, Tipos”.
- Resultados (O): “Chatbots, Herramientas”.
- Contexto (C): Tipos y herramientas usadas en el desarrollo de chatbots.

C. Definición de los criterios de inclusión y exclusión

Se especificaron 3 criterios de inclusión (IC):

- IC1: Documento relacionado al Machine Learning, PLN o Chatbots..
- IC2: Documento en Español o Inglés.

- IC3: Documento con rango de publicación no mayor a 5 años.

Se especificaron 3 criterios de exclusión (EC):

- EC1: Documento sin relación al Machine Learning, PLN o Chatbots..
- EC2: Documento en otro idioma.
- EC3: Documento con rango de publicación mayor a 5 años.

D. Cadenas de Búsquedas

Se definieron palabras clave a través del método PICOC que permitieron junto con la sinonimia de cada palabra realizar diversas combinaciones, usando operadores lógicos “OR” para los conceptos similares, “AND” para los conceptos complementarios y “NOT” para los términos excluyentes. Las búsquedas aplicadas en las bases de datos seleccionadas fueron las siguientes:

IEEE Library:

- ("Autores" OR "PLN" OR "Procesamiento de lenguaje Natural") AND ("Machine Learning" OR "Aprendizaje Automático") AND ("Tipos" OR "clasificación") AND ("Chatbots" OR "Asistentes virtuales, Agentes virtuales" OR "Herramientas" OR "Tecnologías")

Science@Direct:

- ("Autores" OR "PLN" OR "Procesamiento de lenguaje Natural,") AND ("Machine Learning" OR "Aprendizaje Automático") AND ("Tipos" OR "clasificación") AND ("Chatbots" OR "Asistentes virtuales, Agentes virtuales" OR "Herramientas" OR "Tecnologías")

ACM Digital Library:

- ("Autores" OR "PLN" OR "Procesamiento de lenguaje Natural") AND ("Machine Learning" OR "Aprendizaje Automático") AND ("Tipos" OR "clasificación") AND ("Chatbots" OR "Asistentes virtuales, Agentes virtuales" OR "Herramientas" OR "Tecnologías")

E. Evaluación de Calidad

Definida la cadena de búsqueda es importante realizar la evaluación de calidad de los documentos seleccionados. Cada artículo es evaluado siguiendo los criterios de importancia, picoc, keywords, synonyms, and questions, haciendo uso de la herramienta parsifal que permite realizar revisiones sistemáticas.

Las siguientes preguntas se establecieron para evaluar la calidad de los artículos preseleccionados:

- QA1: ¿El documento detalla alguna herramienta de desarrollo de chatbots?
- QA2: ¿El documento implica algún tipo de procesamiento de lenguaje natural?
- QA3: ¿El autor del documento utiliza algún estándar o modelo de desarrollo de chatbot?

III. RESULTADOS

Los resultados obtenidos de la RSL en los siguientes pasos:

- 1) Se ejecutaron las cadenas de búsqueda en cada base de datos y se obtuvieron 72 artículos.
- 2) Los 72 artículos fueron revisados y analizados en su título y resumen, tomando en consideración los criterios de inclusión y exclusión. Del total se eliminaron 52 artículos (72.22%) que son irrelevantes al objeto de estudio, además se descartaron porque su argumentación referente al Machine Learning es débil y no dan contestación a las preguntas de investigación planteadas. En la Tabla I se detalla el proceso de selección de los estudios, de los cuales se seleccionó 20 documentos para evaluarlos a través de los criterios de calidad.

Tabla I ARTÍCULOS REVISADOS

Base de Datos	Artículos				
	Encontrados	Duplicados	Revisados	Eliminados	Seleccionados
IEEE Xplore	27	0	27	21	6
Science@Direct	25	0	25	12	13
ACM	20	0	20	19	1
Total	72	0	72	52	20

- 3) Cada una de las preguntas planteadas para el desarrollo del control de calidad tiene un puntaje de 1.5 si se califica con “Útil”, 1 si la respuesta es “Medio” y 0 si la respuesta se evaluó con “No útil”. Cada artículo obtuvo un puntaje de 0 a 4.5 puntos. Si un artículo tiene un puntaje igual o superior a 3, será seleccionado para extraer su información. Los artículos que cumplieron con los criterios de calidad son 2.

A. Informe del mapeo sistemático

MQ1 “¿Cuántos estudios se publicaron a lo largo de los años en el área de Machine Learning y PLN?” Como se detalla en la Fig.1. se muestra el año de publicación de cada artículo tomando en cuenta que en el año 2021 se encuentran 16 que representan el valor más alto.

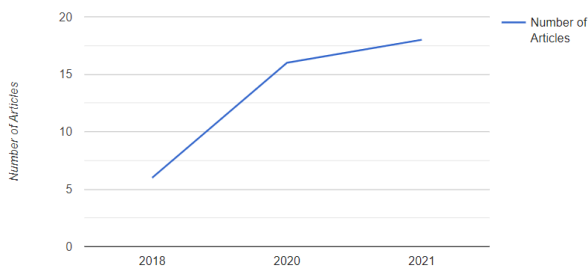


Figure 1. Publicaciones por año

En la Tabla II se presentan los 2 artículos seleccionados donde se detalla la información por: el título, el año de publicación y su fuente de publicación (nombre del congreso o revista) para identificar posibles escenarios de publicación.

Tabla II ARTÍCULOS EVALUADOS

Título	Año / Revista / Congreso
A Conceptual Framework for Engineering Chatbots [17]	2018 IEEE Internet Computing
A Proposed Chatbot Framework for COVID-19 [18]	2021

B. Extracción de la Información

Los criterios de selección de estudios establecen la pauta de extracción de información relevante. Por cada uno se sintetizó y clasificó a cada artículo para tener una visión clara de las preguntas RQ1, RQ2 y RQ3 como se describe en la Tabla III y Tabla IV.

TABLE III TIPO PLN POR ARTÍCULO

Tipo PLN	Artículo
PLN	[17]
NPL	[18]
NSP	[18]

TABLE IV HERRAMIENTAS POR ARTÍCULO

Herramientas	Artículo
BERT	[17][18]

II. DISCUSIÓN

A. Datos históricos sobre Machine Learning y el procesamiento de lenguaje natural

El Machine Learning está experimentando un auge sin precedentes, como en el ámbito empresarial o el ámbito académico, ya que constituyen un pilar fundamental en la transformación, si bien el uso del aprendizaje automático era minoritario, los diversos factores de la época actual han provocado que su uso se intensifique, extendiéndose hacia otros campos [1][3].

Es notable considerar según [2][4] las técnicas de aprendizaje automático pueden definirse como un conjunto de métodos capaces de detectar de forma automática una serie de patrones, bajo esta definición el machine learning lleva existiendo desde hace 50 años, periodo en el que se definieron los métodos estadísticos y se aplicaron al machine learning mediante simples algoritmos.

Un debate que ha perdurado en la comunidad de machine learning es ver si es necesario asumir una estructura para progresar hacia modelos de inteligencia más generales, en sí comparado con el aprendizaje de las personas este es posible porque el cerebro se ha desarrollado con la estructura necesaria para su comprensión, por lo tanto la inteligencia artificial y el machine learning si evolucionarán si se incorporan más estructuras y modelos, dado que en los últimos años están evitando incorporar estructuras previas a modelos, siempre que haya suficientes datos para que el modelo se adapte a la tarea a realizar [5].

B. Tipos de PLN usados por los diferentes autores.

En la elaboración de una Revisión Sistemática de Literatura permitió obtener de 72 estudios de los cuales 2 de ellos fueron seleccionados para el criterio de evaluación según el objeto de estudio, evidenciando de esta forma que los tipos de PLN en desarrollo de chatbots de la mano con machine learning [6][18].

A continuación se pudo evidenciar de manera objetiva que los PLN sí son aplicables en el ámbito del desarrollo de chatbots como estructura principal para el desarrollo de frameworks [7][8][17].

C. Herramientas usadas en el desarrollo de Chatbots

Se muestran algunos datos relacionados a las arquitecturas empleadas del chatbot que ha sido desarrollado en cada uno de los estudios primarios. La cual se utiliza para realizar un análisis de las tecnologías y métodos aplicados.

Según [17][18], todas las entradas que realiza el usuario al chatbot son netamente textuales y no habladas, no obstante también se destaca y diferencia de algunos estudios el chatbot si puede devolver una respuesta hacia el usuario ya sea en forma textual o hablada [6][7]. Mencionado esto, son varias las herramientas encontradas en los distintos artículos, pero no todas resultan útiles para el objeto de estudio en esta revisión. Se puede apreciar que la herramienta usada con más frecuencia para la interpretación de la entrada textual de usuario es BERT [17][18].

Según [19], BERT de Google es uno de esos marcos de PLN, siendo quizás el más influyente en los últimos tiempos. “BERT significa Representaciones de codificador bidireccional de transformadores. Está diseñado para entrenar previamente representaciones bidireccionales profundas a partir de texto sin etiquetas al condicionar conjuntamente el contexto izquierdo y derecho. Como resultado, el modelo BERT pre entrenado se puede ajustar con solo una capa de salida adicional para crear modelos de última generación para una amplia gama de tareas de PLN”.

Lo que destaca de BERT es primero por su facilidad de entender que BERT significa Representaciones de codificador bidireccional de transformadores. Cada palabra aquí tiene un significado. Por ahora, la conclusión clave de esta línea es que BERT se basa en la arquitectura Transformer [20]. En segundo lugar, BERT es un modelo “profundamente bidireccional”. Bidireccional significa que BERT aprende información tanto del lado izquierdo como del lado derecho del contexto de un token durante la fase de entrenamiento. Y finalmente, el aspecto más impresionante de BERT es poder ajustarlo agregando solo un par de capas de salida adicionales para crear modelos de última generación para una variedad de tareas de NLP [21].

III. CONCLUSIONES

En este artículo se elaboró una revisión sistemática de literatura sobre machine learning y PLN en el desarrollo de chatbots, para ello se aplicó la metodología de Barbara Kitchenham. Teniendo como resultados una revisión sistemática de la literatura de los datos relevantes sobre los tipos de PLN aplicados y herramientas utilizados en la implantación de chatbot, cabe mencionar que los estudios relacionados no son a nivel nacional, sino internacional dicho esto se puede concluir de los estudios considerados como relevantes, se evidencia que la aplicación de los PLN y si son aplicables al proceso de desarrollo de chatbots. La herramienta más usada por su capacidad de personalización del chatbot y su interpretación de la entrada del usuario es BERT. No obstante existen otras herramientas, que son de conocimiento general y sencillo de usar. Se pudo evidenciar que para obtener mejores resultados al hablar del aprendizaje mediante chatbots, se puede combinar con algún modelo de aprendizaje abierto.

REFERÊNCIAS BIBLIOGRÁFICA

- [1] Synergic Partners (2018). Una Breve Historia del Machine Learning. [online] Available at: <http://www.synergicpartners.com/una-breve-historia-del-machinelearning/> [Accessed 8 May 2018].
- [2] [1] J. Calvo, M. Guzmán, and D. Ramos, "Machine Learning, una pieza clave en la transformación de los modelos de negocio," *Manag. Solut.*, p. 144, 2018, [Online]. Available: <https://www.managementsolutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf>.
- [3] T. H. E. Sakarya and J. Of, "El Machine Learning a través de los tiempos y los aportes a la humanidad," vol. 7, no. 2, pp. 44–68, 2018.
- [4] Faggella, D. (2018). 7 Applications of Machine Learning in Pharma and Medicine. [online] TechEmergence. Available at: <https://www.techemergence.com/machinelearning-in-pharmamedicine/> [Accessed 16 May 2018].
- [5] Ramon, R. Machine Learning es un área de la inteligencia artificial que busca desarrollar técnicas para que las computadoras aprendan. (2014). [online] Available at: <https://www.youtube.com/watch?v=jBsUx-xiK88>.
- [6] S. R. Correa and Paula Andrea Benavides Cañón, "Procesamiento Del Lenguaje Natural En La Recuperación De Información," *Proces. Del Leng. Nat. En La Recuper. Inf.*, 2007, [Online]. Available: http://eprints.rclis.org/9598/1/PROCESAMIENTO_DEL LENGUAJE NATURAL_EN_LA_RECUPERACION_DE_INFORMACION.pdf.
- [7] JANSSEN, T., «Compositionality and the form of rules in Montague grammar», en *Proceedings of the Second Amsterdam Symposium on Montague Grammar and Related Topics*, Groenendijk, J. and Stokhoff, M. (eds.), 1978.
- [8] J. Lloret Egea, "Procesamiento del Lenguaje Natural (PLN) | Natural Language Processing (NLP)." 2019. doi: 10.13140/RG.2.2.30635.39205.
- [9] Sedelmaier, Y., & Landes, D. (2017). Experiences in teaching and learning requirements engineering on a sound didactical basis. *Annual Conference on Innovation and Technology in Computer Science Education, ITICSE, Part F1286*, 116–121. <https://doi.org/10.1145/3059009.305901>.
- [10] Sabriye, A. O. J. ale, & Zainon, W. M. N. W. (2017). A framework for detecting ambiguity in software requirement specification. *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, 209–213. <https://doi.org/10.1109/ICITECH.2017.8080002>.
- [11] Shah, U. S., & Jinwala, D. C. (2015). Resolving Ambiguities in Natural Language Software Requirements. *ACM SIGSOFT Software Engineering Notes*, 40(5), 1–7. <https://doi.org/10.1145/2815021.2815032>.
- [12] A. Benabbou, S. N. Bahloul, and P. Dhaussy, "An Automated Transformation Approach for Requirement Specification," *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 891–900, 2016, doi: 10.1016/j.procs.2016.07.107.
- [13] J.-L. Boulanger, "Requirements Specification of a Software Application," *Certifiable Softw. Appl.*, 3, pp. 43–73, 2018, doi: 10.1016/b978-1-78548-119-2.50004-2.
- [14] B. DeVries and B. H. C. Cheng, "Automatic detection of incomplete requirements via symbolic analysis," *Proc. - 19th ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Model.* 2016, pp. 385–395, 2016, doi: 10.1145/2976767.2976791.
- [15] A. Hayrapetian and R. Raje, "Empirically analyzing and evaluating security features in software requirements," *ACM Int. Conf. Proceeding Ser.*, pp. 1–11, 2018, doi: 10.1145/3172871.3172879.
- [16] M. S. Husain and M. Akheela Khanum, "Word sense disambiguation in software requirement specifications using Wordnet and association mining rule," *ACM Int. Conf. Proceeding Ser.*, vol. 04-05-Marc, pp. 4–7, 2016, doi: 10.1145/2905055.2905179.
- [17] E. Amer, A. Hazem, O. Farouk, A. Louca, Y. Mohamed, and M. Ashraf, "A proposed chatbot framework for covid-19," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*.
- [18] Kalia, Anup, Vukovic, Maja, Pandita, Rahul, and Singh, Munindar P., "A conceptual framework for engineering chatbots," *IEEE Internet Computing*, vol. 22, no. 6, pp. 54–59, 2018.
- [19] Z. Zhou, V. W. L. Tam, and E. Y. Lam, "SignBERT: A BERT-Based Deep Learning Framework for Continuous Sign Language Recognition," *IEEE Access*, vol. 9, pp. 161669–161682, 2021, doi: 10.1109/ACCESS.2021.3132668.
- [20] J. Cerwall, "What the BERT? Fine-tuning KB-BERT for Question Classification," *Degree Proj. Comput. Sci. Eng.*, 2021.
- [21] B. Prakash, R. Guda, A. Garimella, and N. Chhaya, "BERT: A BERT-based Framework for Demographic-aware Empathy Prediction," pp. 3072–3079, 2016.

11.2 Anexo 2: Revisión Sistemática de Literatura.

Tecnologías y Elementos técnicos que componen el desarrollo de un framework de asistentes virtuales: Revisión Sistemática de Literatura

Agreda-Sánchez Francisco, Auquilla-Villamagua Jorge
Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas
Loja - Ecuador
(francisco.c.agreda, jorge.auquilla)@unl.edu.ec

Resumen — Este artículo describe un estudio sobre el mapeo y revisión sistemática de literatura (RSL) con el fin de identificar, analizar y clasificar los artículos publicados sobre las tecnologías, métodos y elementos técnicos implicados en el desarrollo de un framework de asistentes virtuales. Para el desarrollo de la RSL se empleó la metodología propuesta de Bárbara Kitchenham, la misma que permitió seleccionar 8 artículos que presentan información relacionada al objeto de estudio. La selección de los artículos se realizó acorde a un rango máximo de 4 años desde su publicación de acuerdo a los criterios de inclusión; los estudios se obtuvieron de las bases de datos IEEE Xplore, Science@Direct y ACM Digital Library. Para realizar la presentación de los resultados, se elaboraron tablas que muestran datos relevantes sobre las tecnologías, frameworks, y arquitectura empleadas en los frameworks de desarrollo de asistentes virtuales.

Palabras Clave –Tecnología; métodos; asistentes virtuales; estructuras, frameworks.

Abstract — This article describes a study on the mapping and systematic review of the literature (RSL) in order to identify, analyze and classify the articles published on the technologies, methods and technical elements involved in the development of a virtual assistant framework. For the development of the RSL, the methodology proposed by Bárbara Kitchenham was used, which allowed the selection of 8 articles that present information related to the object of study. The articles were selected according to a maximum range of 4 years from their publication according to the inclusion criteria; studies were obtained from the IEEE Xplore, Science@Direct, and ACM Digital Library databases. To present the results, tables were prepared that show relevant data on the technologies, frameworks, and architecture used in the virtual assistant development frameworks.

Keywords –Technology; methods; virtual assistant; structures; Frameworks.

I. INTRODUCCIÓN

La tecnología evoluciona a un ritmo vertiginoso y año tras año las nuevas herramientas de desarrollo de software parecen estar simplificando nuestro trabajo, dada la demanda actual y el poco tiempo de ejecución de los proyectos, las utilidades parecen estar solucionando este problema tan importante, solucionando estos frameworks y muchos otros problemas [1].

Con el término framework se refiere una estructura de software que consta de componentes adaptables e intercambiables para el desarrollo de aplicaciones. En otras

palabras, un framework se define como un esquema de reutilización del software conformado por componentes y sus relaciones, por ejemplo: abstracción de clases, objetos o componentes, además de proveer distintos componentes de conexión a base de datos [2].

“Un framework integra una funcionalidad avanzada a un lenguaje de programación, también automatiza muchos de los patrones de programación para dirigirlos a un determinado objetivo, proporcionando una estructura al código, lo mejora y lo hace más entendible y sostenible, y permite que la aplicación se divida en capas [3]”. Diseñar un aplicación basada en un framework implementa una instancia del mismo, es decir, rellena los huecos que faltan al framework para ser una aplicación concreta, el mismo soporta la estructura general de cualquier aplicación del dominio al que se adscribe, a modo de esqueleto, y son esos huecos los que aportan la flexibilidad requerida para ajustar una aplicación a intereses concretos [4].

Un asistente virtual es un sistema de software que puede interactuar con un usuario mediante el uso del lenguaje natural. Un asistente virtual no está sujeto no solo a mensajes de texto, si no a una gama de contenido multimedia que permite una mejor interacción con el usuario. Para una mejor interpretación del lenguaje natural un chatbot, en la actualidad, utiliza técnicas de procesamiento del lenguaje natural [5].

Mediante el uso de computación cognitiva, a través del uso de inteligencia artificial, un chatbot entiende lo que el usuario está intentando decir y responde con un mensaje coherente, relevante y directo relacionado con la tarea o petición que el usuario está solicitando [6].

Además, los agentes para mejorar las respuestas al usuario se pueden conectar con aplicaciones externas que responden a las peticiones que el bot haga. Las características propias que contiene un chatbot lo convierten en una especie de sistema experto, que basado en el conocimiento que contiene simula un diálogo inteligente con el usuario [7].

Los frameworks de chatbot ayudan a crear, conectar, publicar y gestionar chatbots inteligentes e interactivos para ofrecer la mejor experiencia de usuario. Ayudan a crear fácilmente nuevos chatbots. Los marcos de chatbot que se identificaron son: Bert, Microsoft Bot Framework, Rasa, Wit AI, Dialogflow, IBM Watson, Amazon Lex, Pandorabots, Botpress, Botkit, ChatterBot. Todas estas plataformas se basan

en algoritmos de aprendizaje automático. La principal operación que realizan los frameworks es identificar la intención y la entidad. Una intención representa una coincidencia entre lo que dice un usuario y la acción que debe realizar el chatbot. Representa una parte de la conversación. Una entidad, por otro lado, es una herramienta para extraer valores de parámetros de la entrada del lenguaje natural. Cualquier palabra clave importante que se quiera obtener de la petición de un usuario se vincularía a una entidad. Por ejemplo, "¿Reservar un vuelo a París?" Es la entrada de la intención de solicitar reservar un vuelo, mientras que "París" es un individuo de una posible entidad llamada Ciudad que recoge los nombres de cualquier ciudad del mundo [8].

II. MÉTODO DE INVESTIGACIÓN

En [9] menciona que toda investigación debe estar basada en evidencias que permitan identificar cómo ha sido abordado el objeto de investigación por otros autores. Sin embargo, es necesario buscar y agregar evidencias usando estudios secundarios como son las RSL y estudios de mapeo sistemático [10]. Para esta investigación se utilizó el protocolo definido por [11] y se estructuró el documento en referencia a [12] [13] [14] [15].

A. Preguntas de Investigación

A partir de la temática central denominada "Tecnologías y elementos técnicos que componen el desarrollo de un framework de asistentes virtuales." se planteó cuatro preguntas de investigación clasificadas en preguntas para el mapeo sistemático (MQ) y la revisión sistemática (RQ):

- RQ1: ¿Tecnologías usadas en el desarrollo de chatbots?
- RQ2: ¿Frameworks usados para construir Chatbots?
- RQ3: ¿Metodología para desarrollar Frameworks?
- RQ4: ¿Herramientas para el desarrollo de frameworks?

B. Proceso de Búsqueda

Se determinaron un conjunto de términos basados en las preguntas de investigación para construir la cadena de búsqueda. Se utilizó el método PICOC propuesto por [16] para definir el ámbito de la RSL:

- Población (P): "Frameworks, Chatbots".
- Intervención (I): "Frameworks, Tecnologías, Elementos Técnicos".
- Comparación (C): "Conocimiento General, Tecnologías".
- Resultados (O): "Tecnologías, Estructuras, Métodos".
- Contexto (C): Frameworks, Tecnologías y Métodos usados en el desarrollo de Chatbots.

C. Definición de los criterios de inclusión y exclusión

Se especificaron 3 criterios de inclusión (IC):

- IC1: Documento con rango de publicación no mayor a 4 años.
- IC2: Documento en Español o Inglés.
- IC3: Documento relacionado con Frameworks de Chatbots.

Se especificaron 3 criterios de exclusión (EC):

- EC1: Documento relacionado con Frameworks de Chatbots.
- EC2: Documento sin relación con Frameworks de Chatbots.
- EC3: Documento en Español o Inglés.

D. Cadenas de Búsquedas

Se definieron palabras clave a través del método PICOC que permitieron junto con la sinonimia de cada palabra realizar diversas combinaciones, usando operadores lógicos "OR" para los conceptos similares, "AND" para los conceptos complementarios y "NOT" para los términos excluyentes. Las búsquedas aplicadas en las bases de datos seleccionadas fueron las siguientes:

IEEE Library:

- ("Frameworks" OR "Marco de Trabajo") AND ("Chatbots" OR "Asistentes Virtuales") AND ("Elementos Técnicos" OR "Tecnologías") AND ("Tecnologías") AND ("Arquitectura" OR "Estructura" OR "Metodologías" OR "Metodos")

Science@Direct:

- ("Frameworks" OR "Marco de Trabajo") AND ("Chatbots" OR "Asistentes Virtuales") AND ("Elementos Técnicos" OR "Tecnologías") AND ("Tecnologías") AND ("Arquitectura" OR "Estructura" OR "Metodologías" OR "Metodos")

ACM Digital Library:

- ("Chatbots" OR "Frameworks") AND ("Tecnologías") AND ("Arquitectura" OR "Metodologías" OR "Metodos")

E. Evaluación de Calidad

Definida la cadena de búsqueda es importante realizar la evaluación de calidad de los documentos seleccionados. Cada artículo es evaluado siguiendo los criterios de importancia, picoc, keywords, synonyms, and questions, haciendo uso de la herramienta parsifal que permite realizar revisiones sistemáticas.

Las siguientes preguntas se establecieron para evaluar la calidad de los artículos preseleccionados:

- QA1: ¿El documento utiliza un framework de desarrollo de chatbots?
- QA2: ¿El documento implica alguna tecnología de desarrollo de frameworks?

- QA3: ¿El autor detalla alguna arquitectura de framework?
- QA4: ¿El autor menciona alguna metodología de desarrollo de frameworks?

III. RESULTADOS

Los resultados obtenidos de la RSL en los siguientes pasos:

1) Se ejecutaron las cadenas de búsqueda en cada base de datos y se obtuvieron 125 artículos.

2) Los 125 artículos fueron revisados y analizados en su título y resumen, tomando en consideración los criterios de inclusión y exclusión. Del total se eliminaron 105 artículos (84%) que son irrelevantes al objeto de estudio, además se descartaron porque su argumentación referente a las tecnologías de desarrollo de frameworks es débil y no dan contestación a las preguntas de investigación planteadas. En la Tabla I se detalla el proceso de selección de los estudios, de los cuales se seleccionó 20 documentos para evaluarlos a través de los criterios de calidad.

TABLA I ARTÍCULOS REVISADOS

Base de Datos	Artículos				
	Encontrados	Duplicados	Revisados	Eliminados	Seleccionados
IEEE Xplore	50	0	50	35	15
Science@Direct	25	0	25	20	5
ACM	50	0	50	50	0
Total	125	0	125	105	20

3) Cada una de las preguntas planteadas para el desarrollo del control de calidad tiene un puntaje de 1.5 si se califica con “Útil”, 1 si la respuesta es “Medio” y 0 si la respuesta se evaluó con “No útil”. Cada artículo obtuvo un puntaje de 0 a 6 puntos. Si un artículo tiene un puntaje igual o superior a 3, será seleccionado para extraer su información. Los artículos que cumplieron con los criterios de calidad son 20.

A. Informe del mapeo sistemático

MQ1 “¿Cuántos estudios se publicaron a lo largo de los años en el área de desarrollo de frameworks de chatbots?”
Como se detalla en la Fig.1. se muestra el año de publicación de cada artículo tomando en cuenta que en el año 2021 se encuentran 37 que representan el valor más alto.

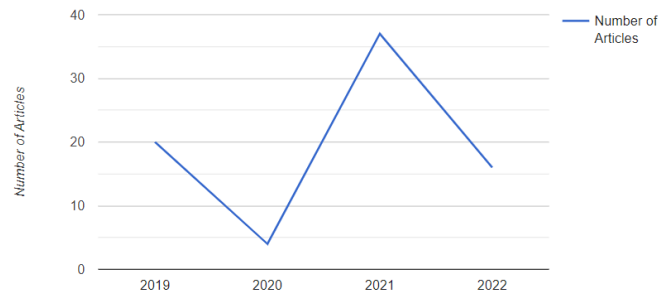


Figure 1. Publicaciones por año

En la Tabla II se presentan los 36 artículos seleccionados donde se detalla la información por: el título, el año de publicación y su fuente de publicación (nombre del congreso o revista) para identificar posibles escenarios de publicación.

Tabla II ARTÍCULOS EVALUADOS

Título	Año / Revista / Congreso
College Enquiry Chatbot using Rasa Framework [16]	2021
Implementation of Chatbot on University Website Using RASA Framework [17]	2021
An Intelligent Arabic Chatbot System Proposed Framework [18]	2021
Designing a Transactional Smart Assistant in Indonesian using Rasa Framework [19]	2021
Bilingual Chatbot For Covid-19 Detection Based on Symptoms Using Rasa NLU [20]	2022
Design and Implementation of Chatbot Framework For Network Security Cameras[21]	2019
Chatbot development using Java tools and libraries [22]	2020
Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States[23]	2019 Procedia Computer Science

B. Extracción de la Información

Los criterios de selección de estudios establecen la pauta de extracción de información relevante. Por cada uno se sintetizó y clasificó a cada artículo para tener una visión clara de las preguntas RQ1, RQ2, RQ3 y RQ4 como se describe en la Tabla III y Tabla IV.

TABLE III FRAMEWORKS POR ARTÍCULO

Frameworks utilizadas	Artículo
RASA Framework	[16] [17] [19] [20]
ELIZA	[18]
ALICE	[18]

TABLE IV TECNOLOGÍAS POR ARTÍCULO

Tecnología	Artículo
Rasa	[16][19]
Entity Transformer (DIET)	[17]
JAVA	[18][22]
Plataforma Facebook Messenger	[21]
AIML	[23]

TABLE V ARQUITECTURA POR ARTÍCULO

Arquitectura	Artículo
Verdadero	[17] [19] [20] [23]
Falso	[16] [18] [21] [22]

II. DISCUSIÓN

A. Datos históricos sobre los frameworks

Como se puede evidenciar en [1][24] a lo largo de los años la tecnología ha tenido un crecimiento exponencial, naciendo la necesidad de usar un marco desarrollo, porque combinan la funcionalidad y desarrollo, todo esto implementado en un lenguaje de programación único que permite a los desarrolladores crear fácil y rápidamente funcionalidades de diferente índole y de acuerdo a sus necesidades, así como mantener el código generado y crear configuraciones sobre él proporcionando patrones de diseño que permite una creación de código de línea de comandos mucho más potentes y redundantes ya que tienen muchas características y especificaciones, que los programadores pueden adoptar. El framework está diseñado precisamente para evitar estos problemas y generar aplicaciones a partir de la calidad.

Los asistentes virtuales no son una idea nueva. El primer programa que simulaba una conversación fue ELIZA, creado por el profesor del MIT Joseph Weizenbaum en la década de 1960. ELIZA opera reconociendo palabras clave o frases para reproducir una respuesta usando esas palabras clave de respuestas preprogramadas [25].

Como se mencionó antes los asistentes virtuales tiene más de 60 años, y en estas últimas décadas según [26][27][28][29] los asistentes virtuales serán capaces de entender y responder un porcentaje promedio más alto de preguntas sin intervención humana, con mayor precisión y mayor velocidad, generando un Índice de Felicidad promedio y un NPS más altos. A medida que el mercado madure, el 40% de las aplicaciones de chatbot / asistente virtual lanzadas en 2018 habrán sido abandonadas para 2020. Las plataformas que queden ganarán impulso y desarrollarán más casos de uso de segunda generación, lo que dará a conocer mejor la capacidad avanzada que ofrecen algunas empresas. Los proveedores se inclinarán por nichos de mercado que ofrezcan el mayor ahorro de costes, ofreciendo la capacidad de desarrollar e implementar soluciones personalizadas en menor tiempo con el uso de librerías de conocimiento preconstruidos específico para la industria. Los chatbots continuarán mejorando a través de datos de aprendizaje automático, donde cada industria será más eficiente en la colaboración entre sus chatbots y los empleados humanos.

B. Frameworks de Chatbots.

En la elaboración de una Revisión Sistemática de Literatura permitió obtener de 125 estudios de los cuales 8 de ellos fueron seleccionados para el criterio de evaluación según el objeto de estudio, evidenciando de esta forma que la aplicación de un marco de trabajo en el desarrollo de frameworks[24].

A continuación se pudo evidenciar de manera objetiva que los frameworks para el desarrollo de si son aplicables dentro del desarrollo de chatbots son RASA Framework en [16] [17] [19] [20]. Así mismo se destacó en otro artículo el apoyo de framework que en este caso viene hacer ELIZA, ALICE [18].

C. Tecnologías usadas en el desarrollo de frameworks de asistentes virtuales.

Se muestran algunos datos relacionados a las tecnologías empleadas en el desarrollo de frameworks de chatbot que han sido desarrollados en cada uno de los estudios primarios, son varias las herramientas encontradas en los distintos artículos, pero no todas resultan útiles para el objeto de estudio en esta revisión.

Se puede apreciar que la herramienta usada con más frecuencia para el desarrollo de frameworks de Chatbots es RASA, seguida de JAVA , Entity Transformer ,IML y una no tan aplicada comúnmente Plataforma Facebook Messenger [16][17][18][19][21][22][23].

Según [30], RASA es una pila de tecnología NLP de código abierto que ayuda a los desarrolladores a crear chatbots altamente inteligentes utilizando modelos probabilísticos en lugar del enfoque habitual basado en diagramas de flujo. En lugar de depender de estructuras lógicas si/entonces, RASA implementa modelos de aprendizaje automático no sólo para reconocer identidades e intenciones, sino también para determinar el flujo de la conversación. Esto convierte a RASA en el sueño de los desarrolladores de chatbots cuando crean experiencias de conversación altamente sofisticadas que evolucionan continuamente sin necesidad de intervención manual.

Desafortunadamente, la misma riqueza y complejidad lo hacen inadecuado para usuarios no técnicos o para organizaciones que buscan construir bots simples basados en reglas. RASA NLU se puede utilizar para crear bots basados en voz y texto y se integra con la mayoría de los canales de mensajería a través de API [31].

III. CONCLUSIONES

En este artículo se elaboró una revisión sistemática de literatura sobre las tecnologías, métodos y elementos técnicos implicados en el desarrollo de frameworks de chatbots, para ello se aplicó la metodología de Barbara Kitchenham. Teniendo como resultados una revisión sistemática de la literatura de los datos relevantes sobre las tecnologías y métodos utilizados en frameworks de chatbot, cabe mencionar que los estudios relacionados no son a nivel nacional, sino internacional dicho esto se puede concluir de los estudios considerados como relevantes, se evidencia que la aplicación de distintas tecnologías de frameworks si son aplicables al proceso de desarrollo de chatbots a nivel estándar. La herramienta más usada por su capacidad de personalización del y su riqueza en desarrollo para crear bots es RASA.

REFERÊNCIAS BIBLIOGRÁFICA

- [1] M. P. S. Pankaj Telang, Anup Kalia, Maja Vukovic, Rahul Pandita, "A Conceptual Framework for Engineering Chatbots".
- [2] J. Ortega, Dinarle Guevara, María Benavides, "Elementary: Un Framework De Programación Web," *Télématique*, vol. 15, no. 2, pp. 144–171, 2016. [Online]. Available: <https://www.redalyc.org/pdf/784/78457627004.pdf>
- [3] D. Biancha Gutiérrez, G. Camacho Sánchez, and G. Martínez Villalobos, "Design Framework for the Development Dynamic Web Applications," *Sci. Tech.*, vol. XVI, pp. 178–183, 2010. [Online]. Available: <https://www.redalyc.org/pdf/849/84917316032.pdf>
- [4] R. Johnson and B. Foote, "Designing reusable classes," *J. Object-Oriented Program.*, vol. 1, no. 2, pp. 22–30, 35, 1988.
- [5] V. Hristidis, "Chatbot Technologies and Challenges," 2018 First International Conference on Artificial Intelligence for Industries (AI4I), 2018, pp. 126–126, doi: 10.1109/AI4I.2018.8665692.
- [6] [8] A. Bozzon, "Enterprise Crowd Computing for Human Aided Chatbots," 2018 IEEE/ACM 1st International Workshop on Software Engineering for Cognitive Services (SE4COG), 2018, pp. 29–30.
- [7] [9] J. Rodríguez, H. Merlino, and E. Fernández, "Comportamiento Adaptable de Chatbots Dependiente del Contexto," *Rev. Latinoam. Ing. Softw.*, vol. 2, no. 2, p. 115, 2014, doi: 10.18294/relais.2014.115-136.
- [8] Á. G. Abad, "BFMB : Framework Base para Bots Modulares," 2019.
- [9] Sedelmaier, Y., & Landes, D. (2017). Experiences in teaching and learning requirements engineering on a sound didactical basis. Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, Part F1286, 116–121. <https://doi.org/10.1145/3059009.305901>.
- [10] Sabriye, A. O. J. ale, & Zainon, W. M. N. W. (2017). A framework for detecting ambiguity in software requirement specification. ICIT 2017 - 8th International Conference on Information Technology, Proceedings, 209–213. <https://doi.org/10.1109/ICITECH.2017.8080002>
- [11] Shah, U. S., & Jinwala, D. C. (2015). Resolving Ambiguities in Natural Language Software Requirements. ACM SIGSOFT Software Engineering Notes, 40(5), 1–7. <https://doi.org/10.1145/2815021.2815032>
- [12] A. Benabbou, S. N. Bahloul, and P. Dhaussy, "An Automated Transformation Approach for Requirement Specification," *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 891–900, 2016, doi: 10.1016/j.procs.2016.07.107.
- [13] J.-L. Boulanger, "Requirements Specification of a Software Application," *Certifiable Softw. Appl.* 3, pp. 43–73, 2018, doi: 10.1016/b978-1-78548-119-2.50004-2.
- [14] B. DeVries and B. H. C. Cheng, "Automatic detection of incomplete requirements via symbolic analysis," *Proc. - 19th ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Model.* 2016, pp. 385–395, 2016, doi: 10.1145/2976767.2976791.
- [15] A. Hayrapetian and R. Raje, "Empirically analyzing and evaluating security features in software requirements," *ACM Int. Conf. Proceeding Ser.*, pp. 1–11, 2018, doi: 10.1145/3172871.3172879.
- [16] S. Meshram, N. Naik, M. VR, T. More and S. Kharche, "College Enquiry Chatbot using Rasa Framework," 2021 Asian Conference on Innovation in Technology (ASIANCON), 2021, pp. 1–8, doi: 10.1109/ASIANCON51346.2021.9544650.
- [17] L. Fauzia, R. B. Hadiprakoso and Girinoto, "Implementation of Chatbot on University Website Using RASA Framework," 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2021, pp. 373–378, doi: 10.1109/ISRITI54043.2021.9702821.
- [18] N. A. Al-Madi, K. A. Maria, M. A. Al-Madi, M. A. Alia and E. A. Maria, "An Intelligent Arabic Chatbot System Proposed Framework," 2021 International Conference on Information Technology (ICIT), 2021, pp. 592–597, doi: 10.1109/ICIT52682.2021.9491699.
- [19] L. W. Narendra and E. R. Setyaningsih, "Designing a Transactional Smart Assistant in Indonesian using Rasa Framework," 2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE), 2021, pp. 1–6, doi: 10.1109/ICEEIE52663.2021.9616946.
- [20] Y. Mellah, T. Bouchentouf, N. Rahmoun and M. Rahmoun, "Bilingual Chatbot For Covid-19 Detection Based on Symptoms Using Rasa NEU," 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2022, pp. 1–5, doi: 10.1109/IRASET52964.2022.9738103.
- [21] T. Van Cuong and T. M. Tan, "Design and Implementation of Chatbot Framework For Network Security Cameras," 2019 International Conference on System Science and Engineering (ICSSE), 2019, pp. 324–328, doi: 10.1109/ICSSE.2019.8823516.
- [22] A. Ilić, A. Ličina and D. Savić, "Chatbot development using Java tools and libraries," 2020 24th International Conference on Information Technology (IT), 2020, pp. 1–4, doi: 10.1109/IT48810.2020.9070294.
- [23] R. Sutoyo, A. Chowanda, A. Kurniati, and R. Wongso, "Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States," *Procedia Comput. Sci.*, vol. 157, pp. 621–628, 2019, doi: <https://doi.org/10.1016/j.procs.2019.08.226>.
- [24] R. Espinosa-Hurtado, "Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web," *Cedamaz*, vol. 11, no. 2, pp. 133–141, 2021, doi: 10.54753/cedamaz.v11i2.1182.
- [25] F. W. Neiva, J. M. N. David, R. Braga y F. Campos, «Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature.» *Information and Software Technology*, vol. 72, n° C, pp. 137-150, 4 2016.
- [26] P. V. Torres-Carrion, C. S. Gonzalez-Gonzalez, S. Aciar y G. Rodriguez-Morales, «Methodology for systematic literature review applied to engineering and education.» de 2018 IEEE Global Engineering Education Conference (EDUCON), 2018.
- [27] Jin, Z., Chen, X., Li, Z., & Yu, Y. (2019). RE4CPS: Requirements engineering for cyber-physical systems. Proceedings of the IEEE International Conference on Requirements Engineering, 2019-Septe, 496–497. <https://doi.org/10.1109/RE.2019.00072>
- [28] T. Shah and S. Patel, "A Novel Approach for Specifying Functional and Non-functional Requirements Using RDS (Requirement Description Schema)," *Procedia Comput. Sci.*, vol. 79, pp. 852–860, 2016, doi: 10.1016/j.procs.2016.03.083.
- [29] Smullen, D., & Breaux, T. D. (2016). Modeling, analyzing, and consistency checking privacy requirements using eddy, 118–120. <https://doi.org/10.1145/2898375.2898381>
- [30] Mantha, Mady. Introducing DIET: state-of-the-art architecture that outperforms fine-tuning BERT and is 6X faster to train. Rasa Research website.
- [31] Vegesna, A.; Jain, P.; Porwal, D. Ontology based Chatbot (For E-Commerce Website). *Int. J. Comput. Appl.* 2018, 179, 51–55.