



Universidad
Nacional
de Loja

Universidad Nacional de Loja

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

Carrera de Ingeniería en Sistemas

Chatbot para resolver inquietudes académicas sobre estudios de posgrado en la Carrera de Sistemas/Computación de la UNL

Chatbot to solve academic queries about the postgraduate studies of the Computer Science/Systems degree at UNL

Trabajo de Titulación
previo a la obtención del
título de Ingeniero en
Sistemas.

AUTOR:

Leonardo Vicente Paredes Rivas

DIRECTOR:

Ing. Roberth Gustavo Figueroa Díaz, Mg.Sc.

Loja – Ecuador
2023

Certificación

Loja, 01 de agosto del 2023

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo Titulación denominado: **Chatbot para resolver inquietudes académicas sobre estudios de posgrado en la Carrera de Sistemas/Computación de la UNL**, previo a la obtención del título de **Ingeniero en Sistemas**, de la autoría del estudiante **Leonardo Vicente Paredes Rivas**, con **cédula de identidad Nro.1900553908**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo **Leonardo Vicente Paredes Rivas**, declaro ser autor del presente Trabajo de Titulación denominado: **Chatbot para resolver inquietudes académicas sobre estudios de Posgrado en la Carrera de Sistemas/Computación de la UNL** y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi Trabajo de Titulación en el Repositorio Institucional - Biblioteca Virtual.

Firma:

Autor: Leonardo Vicente Paredes Rivas.

Cédula: 1900553908

Correo electrónico: leonardo.v.paredes@unl.edu.ec

Fecha: 02 de agosto del 2023

Carta de autorización por parte del autor para la Consulta de producción parcial o total, y/o publicación electrónica del texto completo.

Yo; **Leonardo Vicente Paredes Rivas**, declaro ser autor del Trabajo de Titulación denominado : **Chatbot para resolver inquietudes académicas sobre estudios de Posgrado en la Carrera de Sistemas/Computación de la UNL**, como requisito para optar por el **título de Ingeniero en Sistemas**; 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 RDI, en las redes de información del país y el exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de la autorización, firma el autor, en la ciudad de Loja, a los dos días del mes de agosto del dos mil veintitrés.

Firma:

Autor: Leonardo Vicente Paredes Rivas

Cédula: 1900553908

Dirección: Loja (Miraflores)

Correo electrónico: leonardo.v.paredes@unl.edu.ec

Teléfono: 0985563326

DATOS COMPLEMENTARIOS:

Director de Trabajo de Titulación: Ing. Roberth Gustavo Figueroa Díaz, Mg.Sc.

Dedicatoria

El presente trabajo se lo dedico en primer lugar a Diosito, quien me ha dado fortaleza y sabiduría para poder iniciar y finalizar mi formación universitaria a pesar de las dificultades presentadas en el camino, me ha guiado por el camino del bien y me ha otorgado salud y vida para hacer realidad una meta tan preciada como es la de ser ingeniero.

Dedico este Trabajo de Titulación a mis padres, Daniel Paredes y Graciela Rivas, los seres más preciados, que me han apoyado durante toda mi formación profesional brindándome su apoyo económico y emocional, inculcándome valores y principios, fomentando el sentido de la responsabilidad, los resultados fructíferos de la constancia y el sacrificio.

A mi hermana, María Paredes quien me ha compartido su conocimiento y facilitado su ayuda cuando lo he requerido y principalmente incentivándome a la positividad y el compromiso.

A mis amigos, Alexis Carrión, Raquel Espinosa, Mirian Astimbay y Cristian Aguirre, personas con las que he compartido la etapa estudiantil y quienes me han enseñado a valorar cada segundo de mi vida.

Leonardo Vicente Paredes Rivas

Agradecimiento

Agradezco en primera instancia este trabajo a Dios por ser mi guía en el buen camino y por darme salud y vida para enfrentarme a todos los obstáculos que se han presentado en el transcurso de mi vida y que me han permitido cumplir con todas las metas planteadas.

También agradezco todo el sacrificio y esfuerzo puesto en este Trabajo de Titulación a toda mi familia, en especial a mis padres, Daniel Paredes y Graciela Rivas y a mi hermana María Paredes, quienes me han apoyado, me han dado ánimos y han creído en mí, desde que inicié mi formación profesional como ingeniero.

A mi director del Trabajo de Titulación, Roberth Figueroa, que con su experiencia y sabiduría me pudo dar las pautas necesarias para culminar todo el Trabajo de Titulación sin ningún tipo de contratiempos.

A la Universidad Nacional de Loja y en general a los docentes y compañeros que de forma directa o indirectamente formaron parte de todo este proceso.

Leonardo Vicente Paredes Rivas

Índice de contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas	x
Índice de figuras	xi
Índice de anexos	xiii
1. Título	1
2. Resumen	2
2.1. Abstract	3
3. Introducción	4
4. Marco Teórico	6
4.1. Modelo y Notación de Procesos de Negocio (BPMN)	6
4.2. Reglamento de Régimen Académico de la UNL	6
4.3. Inteligencia Artificial	6
4.4. Machine Learning	6
4.5. Procesamiento del Lenguaje Natural.....	7
4.6. BERT de Google	8
4.7. Arquitectura de BERT	9
4.8. Transformer	12
4.9. DistilBERT	19
4.10. Destilación de conocimientos.....	19
4.11. Hugging Face	19
4.12. Chatbot	19
4.12.1. Tipos de Chatbot	20
4.12.2. Arquitectura básica de un Chatbot	21
4.12.3. Características de un Chatbot	22
4.13. Tecnologías y Herramientas	22
4.13.1. JavaScript	22
4.13.2. HTML.....	23

4.13.3. CSS	23
4.13.4. Python.....	23
4.13.5. Flask	23
4.13.6. Google Colab.....	23
4.13.7. GitHub	24
4.13.8. GitLab.....	24
4.13.9. PyTorch	24
5. Metodología.....	25
5.1. Contexto	25
5.2. Proceso	25
5.3. Métodos.....	26
5.4. Técnicas.....	26
5.5. Metodología	27
6. Resultados	28
Objetivo 1: Definir los procesos académicos de posgrados y examinar el estado del arte sobre el modelo BERT empleado en el Procesamiento del Lenguaje Natural.....	28
Tarea 1: Revisar normativas y reglamentos de posgrado de la UNL.....	28
Tarea 2: Definir los procesos académicos de posgrados a considerar en el Chatbot	29
Tarea 3: Realizar una búsqueda sistemática de literatura.....	33
Objetivo 2: Desarrollar una arquitectura de Chatbot o Agente Conversacional utilizando como red neuronal el modelo de lenguaje BERT.	43
Tarea 1: Probar el modelo BERT con la ayuda de Google Colab.....	44
Tarea 2: Ajustar el modelo BERT mediante un entrenamiento con información del Reglamento de la UNL en lo que concierne a posgrados.....	49
Tarea 3: Diseñar la arquitectura del Agente Conversacional	55
Tarea 4: Desarrollar un prototipo web utilizando Python y JavaScript.	58
Tarea 5: Documentar la aplicación.....	63
Objetivo 3: Evaluar el funcionamiento del Chatbot en un ambiente con estudiantes, egresados y profesionales de la carrera de Sistemas/Computación de la UNL.	67
Tarea 1: Realizar pruebas de validación, carga, estrés y rendimiento.	67
Tarea 2: Probar el agente conversacional	73
Tarea 3: Corregir los errores producidos.....	75

7. Discusión	78
7.1. Valoración técnica, económica y ambiental.....	83
7.1.1. Valoración técnica.....	83
7.1.2. Valoración económica.....	83
7.1.3. Valoración ambiental.....	84
8. Conclusiones	85
9. Recomendaciones	86
9.1. Trabajos Futuros.....	87
10. Bibliografía	88
11. Anexos	93

Índice de tablas:

Tabla 1. Preguntas de investigación.	34
Tabla 2. Palabras clave.	35
Tabla 3. Cadena de búsqueda.	35
Tabla 4. Bases de datos científicas.	36
Tabla 5. Criterios de inclusión.	36
Tabla 6. Criterios de exclusión.	37
Tabla 7. Ejecución de las cadenas de búsqueda.	38
Tabla 8. Tecnologías y herramientas utilizadas para ajustar el modelo BERT.	42
Tabla 9. Cuadro comparativo de modelos de lenguaje basados en BERT.	45
Tabla 10. Contextos usados en el modelo DistilBERT.	48
Tabla 11. Precisión del Modelo BERT (DistilBERT).	49
Tabla 12. Resultados del entrenamiento y la evaluación.	54
Tabla 13. Recursos hardware para el chatbot.	64
Tabla 14. Formato de encuesta.	68
Tabla 15. Método GET: ingreso de datos para las pruebas de carga.	70
Tabla 16. Método GET: resumen de las pruebas de carga.	70
Tabla 17. Método GET: ingreso de datos para las pruebas estrés.	71
Tabla 18. Método GET: resumen de las pruebas de estrés.	72
Tabla 19. Método POST: resumen de las pruebas de carga.	72
Tabla 20. Método POST: ingreso de datos para las pruebas de estrés.	73
Tabla 21. Método POST: resumen de las pruebas de estrés.	73
Tabla 22. Formato de preguntas aplicadas a los encuestados.	74
Tabla 23. Valoración técnica.	83
Tabla 24. Valoración económica: Talento humano.	83
Tabla 25. Valoración económica: Hardware y software.	84
Tabla 26. Presupuesto final.	84

Índice de figuras:

Figura 1. Bidireccionalidad.....	8
Figura 2. Pre-entrenamiento y ajuste fino.....	9
Figura 3. Enmascaramiento que se produce en una oración.....	10
Figura 4. Representación de la entrada de BERT.....	11
Figura 5. Arquitectura de BERT en sus dos versiones originales.....	12
Figura 6. Arquitectura simple del Transformer.....	12
Figura 7. Pila de codificadores y decodificadores de un Transformer.....	13
Figura 8. Subcapas del codificador.....	13
Figura 9. Embedding y codificador posicional.....	14
Figura 10. Representación de palabras mediante vectores de cuadros simples.....	15
Figura 11. Vectores queries, keys y values.....	16
Figura 12. Calcular la puntuación.....	16
Figura 13. Normalización de la puntuación con la función softmax.....	17
Figura 14. Salida de la capa de auto atención.....	18
Figura 15. Múltiples bloques de atención.....	18
Figura 16. Interfaz básica de un chatbot.....	21
Figura 17. Esquema resumen de los procesos de matrículas y homologaciones.....	29
Figura 18. Proceso de inscripción a un programa de posgrado.....	30
Figura 19. Proceso de selección para un programa de posgrado.....	31
Figura 20. Proceso de registro de matrícula.....	32
Figura 21. Proceso de homologación.....	33
Figura 22. Proceso de selección de los estudios primarios.....	39
Figura 23. Proceso de selección de los artículos finales.....	40
Figura 24. Importación del modelo y las librerías.....	46
Figura 25. Tokenización del contexto y la pregunta.....	47
Figura 26. Ejemplo simple de una tokenización.....	47
Figura 27. Estructura del conjunto de datos.....	50
Figura 28. Lectura y almacenamiento del conjunto de datos en listas.....	51
Figura 29. Cálculo de la posición final donde termina la respuesta.....	51
Figura 30. Tokenización de los contextos y preguntas en vectores numéricos.....	52
Figura 31. Parámetros de entrenamiento.....	52
Figura 32. Descarga del modelo y selección del entorno de ejecución.....	52
Figura 33. Entrenamiento del modelo.....	53
Figura 34. Ejemplo de una historia de usuario.....	55

Figura 35. Arquitectura del chatbot.	57
Figura 36. Funcionamiento interno del chatbot.	58
Figura 37. Estructura de la aplicación.	59
Figura 38. Importación del modelo y entorno de ejecución.	59
Figura 39. Fase de predicción de DistilBERT.	60
Figura 40. Ejemplo de uno de los contextos de la base de conocimientos.	60
Figura 41. Extracción del contexto de la base de conocimientos	61
Figura 42. Envío de las preguntas al modelo de lenguaje.	62
Figura 43. Ejecución del chatbot.	62
Figura 44. Archivo app.js.	63
Figura 45. Creación del servidor virtual	64
Figura 46. Dirección IP para la conexión remota.	64
Figura 47. Aplicación alojada en el servidor virtual.	65
Figura 48. Ejecución de la aplicación.	65
Figura 49. Chatbot en funcionamiento en la página Web: Ejemplo 1.	66
Figura 50. Chatbot en funcionamiento en la página Web: Ejemplo 2.	66
Figura 51. Chatbot en funcionamiento en la página Web: Ejemplo 3.	67
Figura 52. Interfaz gráfica de ApacheJMeter.	69
Figura 53. Resultados de la aceptación del chatbot.	75
Figura 54. Adaptación del chatbot a dispositivos móviles.	76
Figura 55. Interfaz del chatbot para dispositivos móviles.	77

Índice de anexos:

Anexo 1. Ejecución de las cadenas de búsqueda.	93
Anexo 2. Encuesta realizada a la secretaria de posgrado.....	95
Anexo 3. Preguntas realizadas al modelo DistilBERT.	99
Anexo 4. Entrenamiento del modelo DistilBERT.	101
Anexo 5. Historias de usuario.	115
Anexo 6. Encuesta aplicada a tres interesados, para comprobar los criterios de validación de las historias de usuario del agente conversacional.....	118
Anexo 7. Ejecución de las pruebas con los métodos GET y POST en la herramienta ApacheJMeter.....	126
Anexo 8. Encuesta aplicada a la secretaria Nataly Mendoza, para conocer el promedio de interesados que realizan preguntas sobre los programas de posgrados.	132
Anexo 9. Encuesta aplicada a estudiantes de último ciclo, egresados y profesionales para determinar el grado de satisfacción del agente conversacional.....	134
Anexo 10. Certificación de traducción	138

1. Título

Chatbot para resolver inquietudes académicas sobre estudios de Posgrado en la Carrera de Sistemas/Computación de la UNL.

Chatbot to solve academic queries about the postgraduate studies of the Computer Science/Systems degree at UNL

2. Resumen

En la carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja (UNL), los estudiantes tras culminar sus estudios de tercer nivel suelen optar por continuar su formación profesional con programas de posgrados. Sin embargo, obtener información de forma inmediata respecto a ello es tedioso debido a los tiempos largos de respuesta que dan los administrativos a través del correo electrónico o las redes sociales (Instagram, Twitter y Facebook) de la UNL. Por tal motivo surge la siguiente pregunta de investigación, “**¿Un Chatbot implementado con el modelo de lenguaje BERT, ayudará a los estudiantes y profesionales a tener una respuesta frente a las inquietudes sobre los procesos de posgrados que se dan en la Carrera de Sistemas/Computación de la UNL?**”. En este contexto se planteó el siguiente objetivo, “**Implementar un Chatbot para la Carrera de Sistemas/Computación de la UNL, utilizando el modelo de lenguaje BERT, que permita solventar las dudas de estudiantes y profesionales sobre los procesos que se rigen antes y durante los estudios de posgrados**”.

El proyecto se culminó empleando las metodologías de Bárbara Kitchenham y la programación extrema (XP). La primera permitió hacer una revisión sistemática de literatura (RSL) para conocer el funcionamiento del modelo BERT y las tecnologías necesarias a usar en el entrenamiento y ajuste. La segunda se utilizó en el desarrollo de la interfaz web utilizando las fases de planeación, diseño, codificación y pruebas. Durante este proceso se creó un dataset con información sobre los estudios de posgrados, que se usó para entrenar el modelo (se escogió una versión denominada DistilBERT) obteniendo una precisión de 89,2%. Posteriormente, se diseñó la arquitectura del chatbot, la programación y una serie de pruebas en un entorno con estudiantes y profesionales que permitieron evaluar a través de una encuesta la aceptación del agente virtual, obteniendo un resultado del 83,9%, valor que indicó que el chatbot fue capaz de responder “todas” las preguntas planteadas por los interesados. De esta forma, se concluyó que el desarrollo del asistente virtual usando el modelo BERT puede ayudar en mayor medida a estudiantes y profesionales a solventar sus dudas con respecto a los estudios de cuarto nivel.

Palabras claves: Agente Conversacional, DistilBERT, BERT, Inteligencia Artificial

2.1. Abstract

In the Systems/Computer Engineering career at the National University of Loja (UNL), students after completing their third level studies often choose to continue their professional training with postgraduate programs. However, obtaining immediate information about it is tedious due to the long response times given by the administrative staff through email or social networks of the UNL. For such reason the following research question arises, **"Will a Chatbot implemented with the BERT language model, help students and professionals to have an answer in front of the concerns about the postgraduate processes that occur in the Systems/Computing Career of the UNL?"**. In this context, the following objective was set, **"To implement a Chatbot for the UNL Systems/Computing Career, using the BERT language model, which allows to solve the doubts of students and professionals about the processes that are governed before and during the postgraduate studies"**.

The project was completed using the methodologies of Barbara Kitchenham and Extreme Programming (XP). The former allowed for a systematic literature review (SLR) to understand the functioning of the BERT model and the technologies needed for its training and adjustment. The second was used in the development of the web interface using the planning, design, coding and testing phases. During this process, a dataset was created with information on postgraduate studies, which was used to train the model (a version called DistilBERT was chosen), obtaining an accuracy of 89.2%. Subsequently, the chatbot architecture, programming and a series of tests were designed in an environment with students and professionals that allowed evaluating through a survey the acceptance of the virtual agent, obtaining a result of 83.9%, which affirmed that the chatbot was able to answer "all" the questions posed by the interested parties. Thus, it was concluded that the development of the virtual assistant using the BERT model can help students and professionals to a greater extent to solve their doubts regarding fourth level studies.

Keywords: Conversational Agent, DistilBERT, BERT, Artificial Intelligence

3. Introducción

Hoy en día, el auge de la Inteligencia Artificial (IA) ha permitido el desarrollo de sistemas inteligentes con la capacidad de realizar tareas automatizadas sin la intervención de un ser humano. Uno de estos sistemas son los “agentes conversacionales” o “chatbots”, aplicaciones informáticas con la capacidad de simular una conversación natural con un ser humano [1] utilizando el Procesamiento del Lenguaje Natural (NLP). Las empresas los suelen emplear como medio de comunicación para dar asistencia a las peticiones y solicitudes de sus clientes en el menor tiempo posible. En las instituciones educativas los chatbots se están convirtiendo en herramientas esenciales para responder preguntas frecuentes que los estudiantes hacen sobre temas específicos, evitando de esta forma acudir al centro educativo para solventar sus inquietudes de manera personal frente a un administrativo, o esperar por largos periodos de tiempo cuando las consultas son realizadas mediante otros medios de comunicación [2]. Incluso se los están empleando para reforzar el aprendizaje en diferentes campos de la educación [3]. Sin embargo, para que un asistente virtual sea lo suficientemente preciso en cualquier tarea (análisis de sentimientos, responder preguntas, traducción de texto, etc) se requiere el uso de redes neuronales. Actualmente, lo más conocido son los modelos de lenguajes basados en BERT entrenados con datos masivos (SQuAD, GLUE, RACE) [4] cuya arquitectura se asemejan a las redes Transformers [5]. Emplear estos modelos requiere el uso de grandes recursos de cómputo (hardware), por lo que existen autores que han trabajado en versiones de redes neuronales más livianas que estén al alcance para ser implementados en equipos limitados [6], [7].

En este contexto y como un aporte a la sociedad, se planteó en este Trabajo de Titulación (TT) el desarrollo de un **“Chatbot para resolver inquietudes académicas sobre estudios de Posgrado en la Carrera de Sistemas/Computación de la UNL¹”**, utilizando como red neuronal una versión ligera del modelo BERT (DistilBERT) que permita proporcionar una herramienta de asistencia virtual para que los interesados resuelvan sus dudas en el menor tiempo posible.

Anteproyecto

¹ <https://github.com/Leo646/Anteproyecto>

Los objetivos planteados para este TT se presentan a continuación:

General:

- Implementar un Chatbot para la Carrera de Sistemas/Computación de la UNL, utilizando el modelo de lenguaje BERT, que permita solventar las dudas de los estudiantes y profesionales sobre los procesos que se rigen antes y durante los estudios de posgrados.

Específicos:

- Definir los procesos académicos de posgrados y examinar el estado del arte sobre el modelo BERT empleado en el Procesamiento del Lenguaje Natural.
- Construir una arquitectura de Chatbot o Agente Conversacional utilizando como red neuronal el modelo de lenguaje BERT.
- Evaluar el funcionamiento del Chatbot en un ambiente con estudiantes, egresados y profesionales de la Carrera de Sistemas/Computación de la UNL.

Este TT está enmarcado en 6 secciones, iniciando con el apartado de revisión bibliográfica, donde se presenta la información fundamental que permitió obtener los conocimientos principales para la ejecución del proyecto. Seguidamente, se encuentra el apartado de materiales y métodos en los que se muestran los recursos empleados en toda la investigación. A continuación, siguen los resultados, a través de los cuales se presenta la evidencia obtenida para cada uno de los objetivos planteados. Posteriormente, se muestra la sección de discusión donde se analiza cada uno de los resultados obtenidos. Por último, están las secciones de conclusiones y recomendaciones, que se emitieron tras la finalización de este trabajo.

4. Marco Teórico

4.1. Modelo y Notación de Procesos de Negocio (BPMN)

Es un estándar que se utiliza a nivel internacional como un lenguaje global empleado para representar visualmente diferentes procesos de negocio y de esta manera facilitar su comprensión. BPMN permite elaborar diagramas de flujos de actividades, eventos, compuertas lógicas y elementos adicionales para tener una visión detallada con respecto al funcionamiento, la estructura de un proceso y de la interacción que puede existir entre uno y otro [8].

4.2. Reglamento de Régimen Académico de la UNL

Es un documento compuesto de 372 artículos encaminado a regular y orientar el desarrollo de las funciones de investigación, docencia y vinculación con la sociedad en la UNL. Mediante el Reglamento de Régimen Académico se establecen reglas o normas para la formación, permanencia, ingreso, movilidad, promoción, titulación y movilidad de los estudiantes en la Universidad Nacional de Loja. Además, se decretan procesos para la ejecución, planificación, y evaluación de las tareas relacionadas con la sociedad para dar una variedad de soluciones respecto a las necesidades y problemas desde el punto de vista académico².

4.3. Inteligencia Artificial

De acuerdo a la Real Academia Española (RAE), la Inteligencia Artificial (IA) es una “disciplina que se ocupa de elaborar algoritmos informáticos empleados para ejecutar operaciones comparables a las que efectúa la mente humana, como el razonamiento o el aprendizaje”³. Según el autor Caiafa César en su artículo [9] considera la IA como la facultad que adquieren las máquinas para tomar decisiones al momento de resolver los errores generados en diferentes tareas, gracias a la elaboración de algoritmos basados en redes neuronales que presentan la cualidad de predecir y aprender un resultado a partir de una fuente de datos; su finalidad es elaborar sistemas informáticos con inteligencia, capaces de simular las funciones intelectuales del cerebro humano.

4.4. Machine Learning

El Machine Learning (ML) es un subcampo de la IA, que gracias a la creación de algoritmos dan la capacidad a los dispositivos de identificar secuencias o patrones en grandes corpus de datos, para realizar predicciones y que las computadoras hagan tareas específicas de forma autónoma.

El ML hace posible que los programas informáticos aprendan sobre la base de experiencias

² Reglamento de Régimen Académico de la UNL: <https://bit.ly/3RS9pbR>

³ RAE: <https://dle.rae.es/inteligencia>

previas obtenidas en situaciones similares. Los sistemas informáticos, que se espera que aprendan, en primera instancia, toman un ejemplo y aprenden cierta información de este, seguidamente, se fijan en un segundo ejemplo para obtener más datos. Este proceso ayuda a hacer generalizaciones para la situación que se va a aprender. Es posible verlo como una forma de aprender de las experiencias.

De acuerdo al artículo [10], los algoritmos del ML se dividen en cuatro categorías, de las cuales las dos primeras son las más frecuentes:

- **Aprendizaje supervisado:** presentan algoritmos que tienen un aprendizaje previo a partir de datos etiquetados, los cuales permiten hacer predicciones.
- **Aprendizaje no supervisado:** los algoritmos no tienen un entendimiento previo, por lo que se enfrentan a un desorden de datos, en donde se deben hallar patrones que permitan ordenarlos de alguna forma.
- **Aprendizaje semi-supervisado:** es una técnica donde una parte de los datos de entrenamiento están etiquetados y los datos restantes no lo están; la cantidad de datos sin etiquetar a menudo supera con creces la cantidad de datos etiquetados.
- **Aprendizaje por refuerzo:** estos algoritmos tienen como objetivo, que sean capaces de aprender a partir de la propia experiencia, es decir, que tomen decisiones en situaciones de prueba y error para realizar predicciones correctas. Actualmente se utilizan en el reconocimiento facial, diagnósticos médicos o para la clasificación de secuencias de ADN.

4.5. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (NLP) es una área de la Inteligencia Artificial, que busca establecer una comunicación entre una máquina y un humano, y por lo tanto se utiliza en diversas situaciones como la traducción automática, recuperación de información, minería de texto, reconocimiento de voz, respuestas a preguntas, traducciones automáticas y otros campos de la IA [11]. Hay varias fases básicas en el NLP, incluyendo la fonética, morfología, sintaxis, semántica y pragmática. Para que una máquina sea capaz de comprender el dialecto humano, se necesita dividir todo el texto en párrafos, oraciones o palabras [12], así como reconocer las relaciones entre las diferentes palabras, extraer el significado exacto del texto, comprender oraciones en diferentes situaciones y considerar el contexto discursivo previo [13].

4.6. BERT de Google

BERT es un modelo de red neuronal que se aplica en la IA, específicamente en el NLP, en un subcampo llamado Entendimiento del Lenguaje Natural (NLU). BERT permite que un sistema informático comprenda mejor el lenguaje humano. Este algoritmo fue creado en 2018 por Google y anunciado públicamente en 2019 por Pandu Nayak vicepresidente de Google Search. Además, por esa misma fecha fue incorporado a su motor de búsqueda para mejorar las consultas que hacían los usuarios en el menor tiempo posible [14].

BERT son las siglas de, “Representación de Codificadores Bidireccionales de Transformadores”. Este significado se lo puede dividir de la siguiente manera:

- **Representación de codificadores:** se utiliza un codificador para obtener una representación del sistema del lenguaje. BERT entra dentro de la familia de un sistema de modelado del lenguaje y utiliza información pre-entrenada sin etiquetar de ninguna forma, esto quiere decir, que BERT no ha sido creado para resolver una tarea específica, lo que hace es dar la representación de palabras y frases más eficientes y flexibles para utilizarlas en una tarea concreta [5].
- **Bidireccionalidad:** permite analizar una oración en dos direcciones (tiene acceso a todo el contexto de la oración), es decir, examina los términos que se encuentran a la izquierda y a la derecha de las palabras clave que está analizando para que el modelo entienda en profundidad el contexto de la oración [5]. En la **Figura 1** se puede observar la representación de la bidireccionalidad, donde los términos clave están subrayados de color negro, y el resto de la oración aporta significado a las palabras clave haciendo que el modelo analice las oraciones en ambas direcciones.

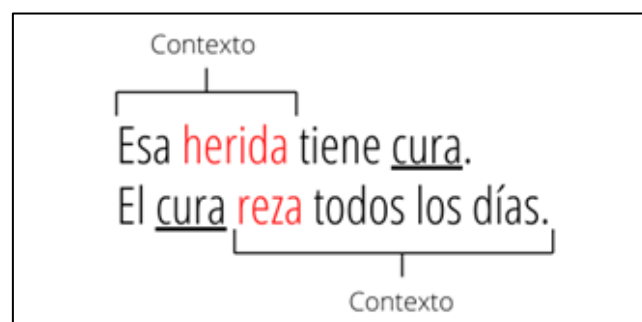


Figura 1. Bidireccionalidad.

Fuente: [15].

- **De transformadores:** hace referencia a la arquitectura de BERT [5]. Un transformador es un algoritmo muy poderoso del NLP diseñado para resolver tareas secuenciales; traducción de texto, predicciones, etc.

4.7. Arquitectura de BERT

La arquitectura de BERT se basa en un codificador bidireccional multicapa de un “Transformer”. Un Transformer o Transformador [16], es una red neuronal, cuya arquitectura es un codificador-decodificador, la cual utiliza mecanismos de atención para calcular las representaciones de su entrada y salida. Esto quiere decir, que los mecanismos de atención se ocupan de codificar cada una de las palabras de acuerdo a las demás que conforman el texto, permitiendo separar la información del contexto y representarla numéricamente [17]. BERT es un modelo que utiliza dos estrategias en su marco: El Pre-entrenamiento y Ajuste Fino (Figura 2). El proceso de pre-entrenamiento, se basa en dos tareas no supervisadas únicas: Modelado de Lenguaje Enmascarado (MLM) y Predicción de la Siguiente Oración (NSP).

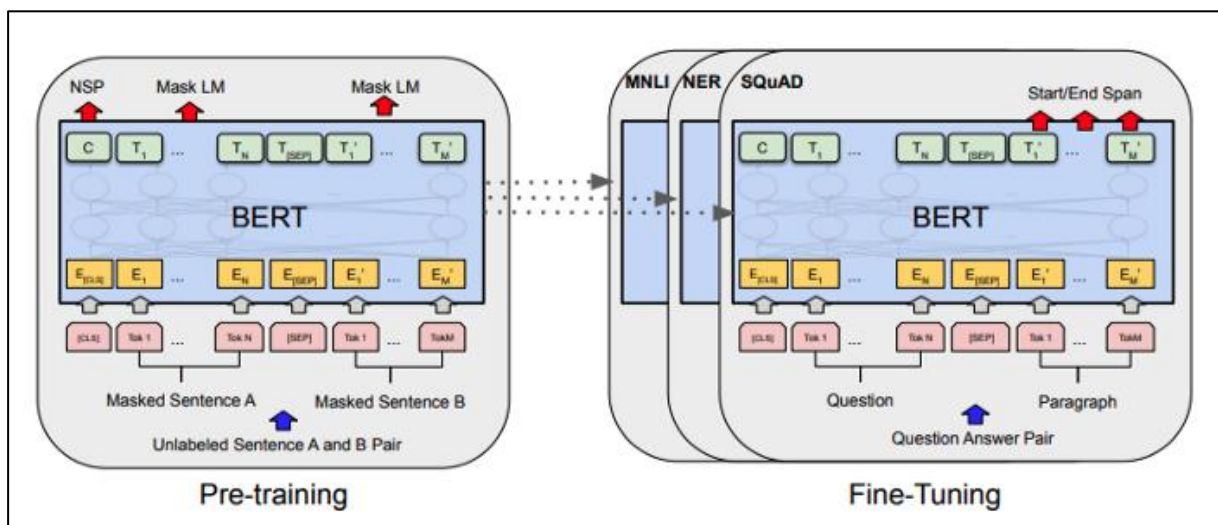


Figura 2. Pre-entrenamiento y ajuste fino.

Fuente: [5].

El objetivo del MLM es ocultar la palabra de una oración y luego hacer que el modelo la prediga en función del contexto. En el MLM, normalmente se enmascara un 15% de palabras, que son reemplazadas por el token especial [MASK], y se espera que la red sea capaz de predecir las palabras enmascaradas basándose en otras de la misma oración. Este procedimiento de entrenamiento, hace que el modelo sea de naturaleza bidireccional, porque la representación de la palabra enmascarada se aprende en función de los términos que aparecen tanto a la izquierda como a la derecha del contexto. También se puede visualizar como el enunciado de un problema, en donde es necesario rellenar los espacios en blanco. Sin embargo, la consecuencia de ello es que se produce un desequilibrio tanto en el pre-entrenamiento como en el ajuste fino, debido a que la palabra [MASK] que oculta algunos tokens de entrada no se va a evidenciar en el ajuste. Para controlar este resultado, durante el pre-entrenamiento se oculta en un 80% de los casos las palabras con el token [MASK], en

otro 10% se utiliza un token aleatorio y el 10% restante, no se modifica nada [18]. En la **Figura 3** se presenta un ejemplo.

```
80% of the time: Replace the word with the
[MASK] token, e.g., my dog is hairy →
my dog is [MASK]

10% of the time: Replace the word with a
random word, e.g., my dog is hairy → my
dog is apple

10% of the time: Keep the word un-
changed, e.g., my dog is hairy → my dog
is hairy. The purpose of this is to bias the
representation towards the actual observed
word.
```

Figura 3. Enmascaramiento que se produce en una oración.

Fuente: [5].

La segunda tarea del pre-entrenamiento es la predicción de la siguiente oración. La mayor parte de las tareas en las que se puede emplear BERT son para preguntas y respuestas (QA) o inferencia del lenguaje natural (NLI). En el proceso de entrenamiento previo de BERT, el modelo acepta parejas de oraciones como entrada (un 50%) y aprende a deducir si la oración contigua del par es la expresión subsiguiente que debe aparecer.

BERT puede representar la entrada de una oración o un conjunto de oraciones como una secuencia de tokens (por ejemplo, una pregunta y una respuesta) de acuerdo a las siguientes características [19]:

- Emplea “WordPiece embeddings”, que consta de un vocabulario de 30.522 tokens.
- El primer token que se emplea en cada secuencia es una palabra especial llamada [CLS].
- Cuando hay más de una oración, estas se unen en una única pieza que van a estar separadas con el token especial [SEP], y a cada token se le añade una incrustación que indica la frase a la que pertenece.
- Para la representación de la entrada de los tokens estos se consiguen sumando los embeddings del token, del segmento y de la posición.

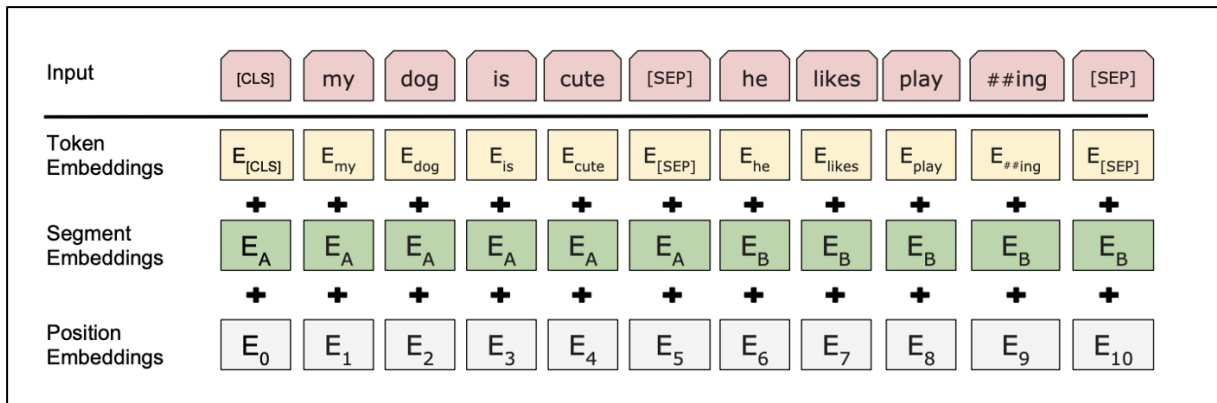


Figura 4. Representación de la entrada de BERT.

Fuente: [5].

El modelo BERT muestra su arquitectura en dos tamaños (**Figura 5**): BERT_{base} con, L=12, H=768, A=12, parámetros totales=110M; y BERT_{large} con, L= 24, H=1024, A=16, parámetros totales=340M). Siendo:

- L: Número de capas.
- H: Tamaño de la capa oculta (tamaño de los vectores).
- A: Número de encabezados de auto atención.

La diferencia entre estas dos versiones radica en el número de capas de codificadores. El modelo base tiene 12 capas apiladas una encima de la otra, mientras que la versión large presenta 24 capas. Conforme aumenta la cantidad de codificadores del modelo, también incrementa el número de parámetros (pesos) y el número de encabezados de atención. BERT_{base} se entrenó en 4 TPUs y para BERT_{large} se utilizaron 16 TPUs. Ambos modelos fueron entrenados durante 4 días en la nube [5].

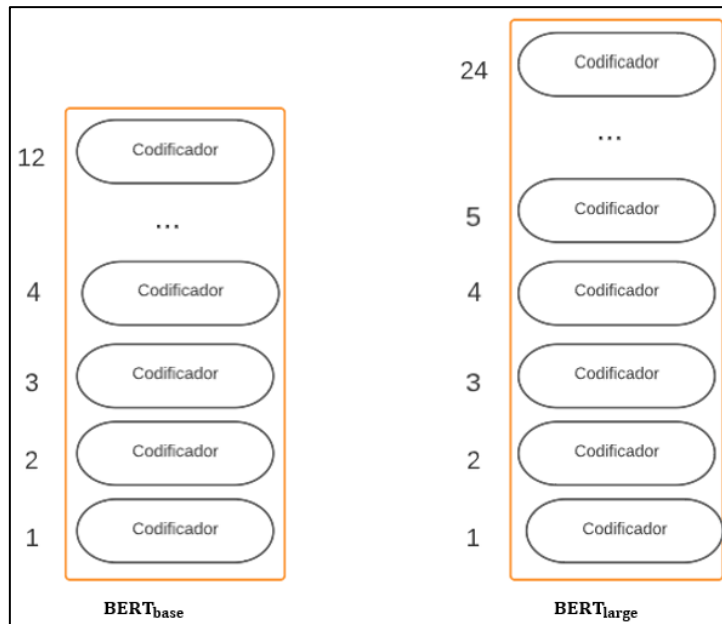


Figura 5. Arquitectura de BERT en sus dos versiones originales.

4.8. Transformer

El funcionamiento de BERT se basa en el codificador del Transformer, por lo que es necesario explicar de manera general el funcionamiento de este. Se puede comenzar mirando el modelo como una caja (**Figura 6**), la cual recibe como entrada una frase en un idioma y genera su representación en otro como salida.

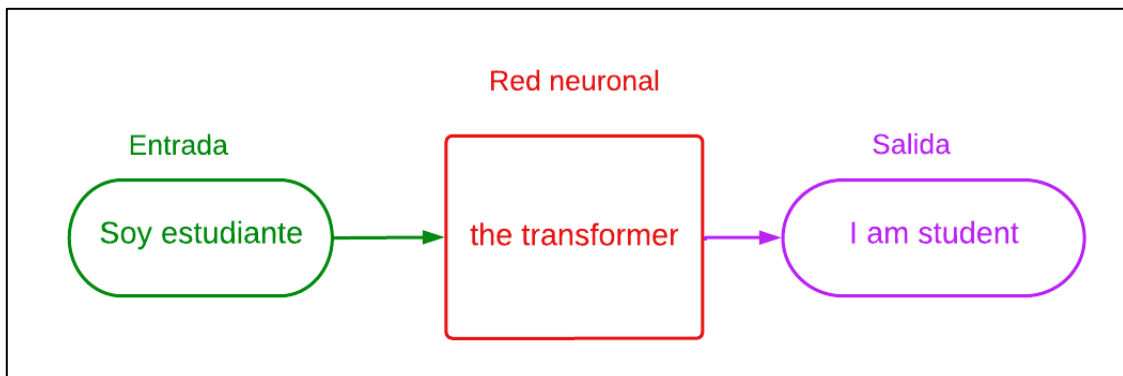


Figura 6. Arquitectura simple del Transformer.

Al explorar la caja se muestran dos componentes importantes; una pila de codificadores (seis de ellos, uno encima del otro) los cuales presentan una estructura idéntica, pero con pesos diferentes; y una pila de decodificadores del mismo número [20] (**ver Figura 7**).

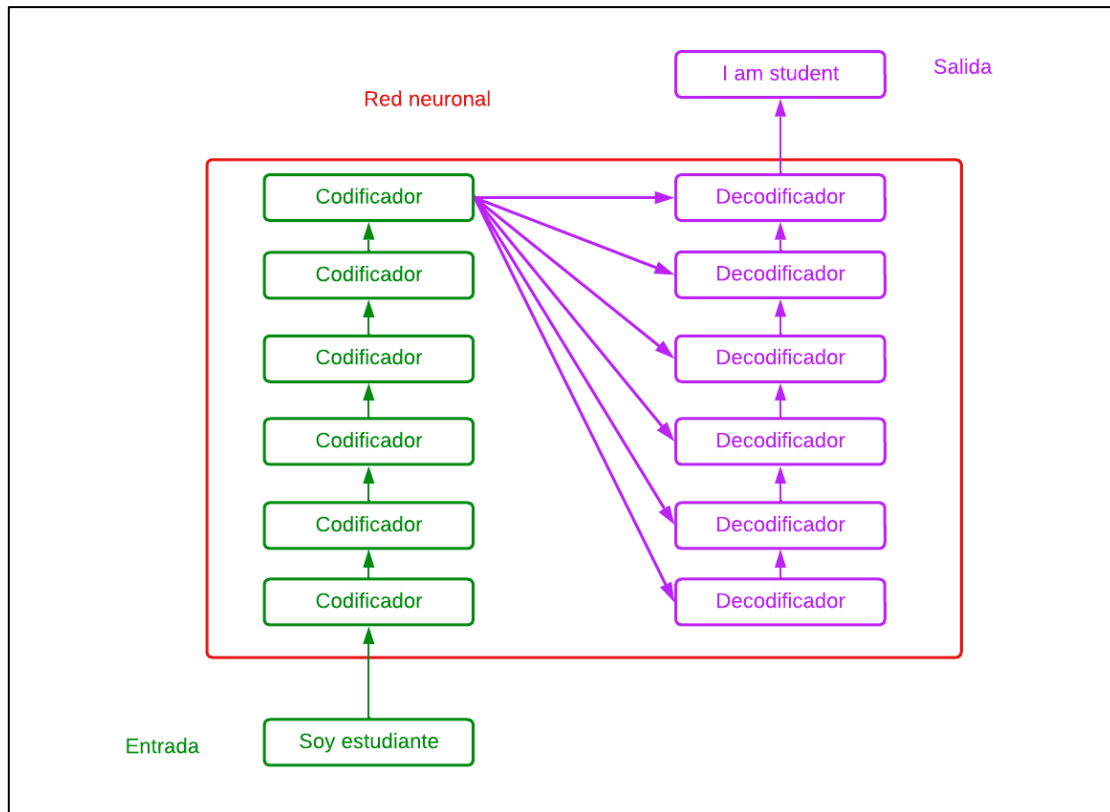


Figura 7. Pila de codificadores y decodificadores de un Transformer.

Cada codificador se divide en dos subcapas (**Figura 8**), una capa de “Auto atención” y una red neuronal “Feed Forward”. En el codificador las palabras pasan primeramente a través de la capa de atención, la cual ayudará al codificador a observar la similitud con otras palabras en la frase de entrada mientras codifica una palabra en particular. La salida de la capa de Auto atención se envía a la red neuronal Feed Forward y después al siguiente codificador. La misma red Feed Forward se usa independientemente en cada posición [16], [20].

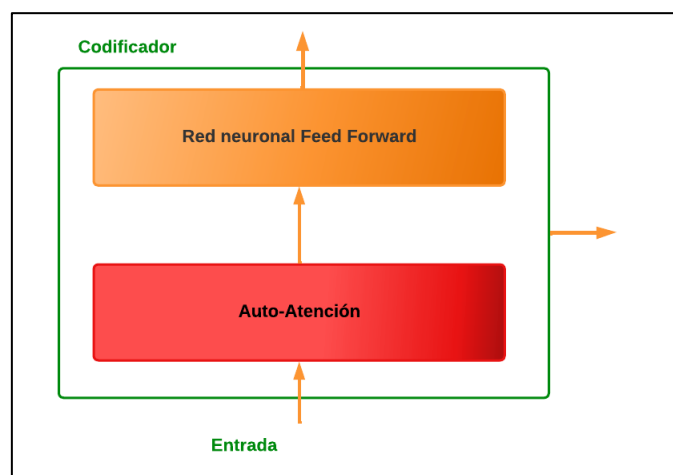


Figura 8. Subcapas del codificador.

Para que el modelo pueda entender una oración, se debe convertir cada palabra en vectores numéricos utilizando un algoritmo de incrustación. Para comprender un poco mejor esto, es necesario tener en cuenta dos conceptos importantes:

- **Embedding.** Es un algoritmo que se encarga de transformar texto en una secuencia de vectores numéricos (**ver Figura 9**) para que el modelo pueda entender la secuencia de entrada [21].
- **Codificación Posicional.** Genera un tipo de vectores que se adjuntan a los tokens (**ver Figura 9**) indicando la posición en la que se encuentra cada token dentro de la secuencia, para ello hace uso de funciones matemáticas para las posiciones pares (funciones senoidales) e impares (funciones cosenoidales), generando una secuencia numérica con la información de la posición [21].

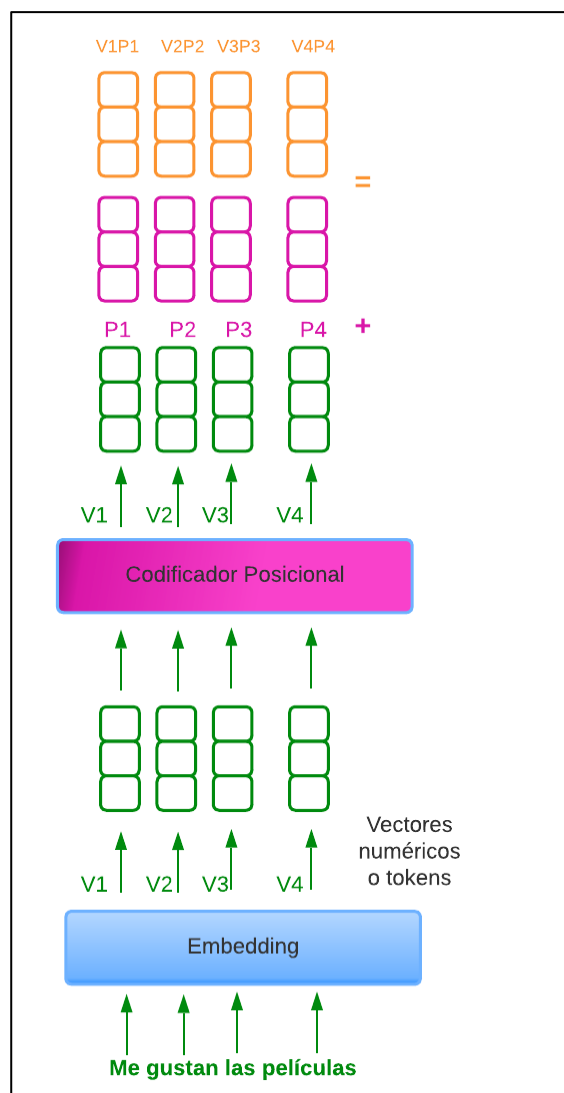


Figura 9. Embedding y codificador posicional.

Los codificadores aceptan una lista de vectores con un tamaño de 512; en el codificador que se encuentra al inicio de la arquitectura se producirán las incrustaciones de palabras y en los codificadores restantes se generará la salida del codificador que está directamente debajo. El tamaño de la lista es un parámetro que se puede configurar, en el cual se indica la longitud de la frase más larga en el data set de entrenamiento [17]. Después de ingresar las palabras en la secuencia de entrada, estas pasan a través de las dos capas del codificador. Se pueden representar los vectores mediante cuadros simples para tener una visión gráfica (ver **Figura 10**).

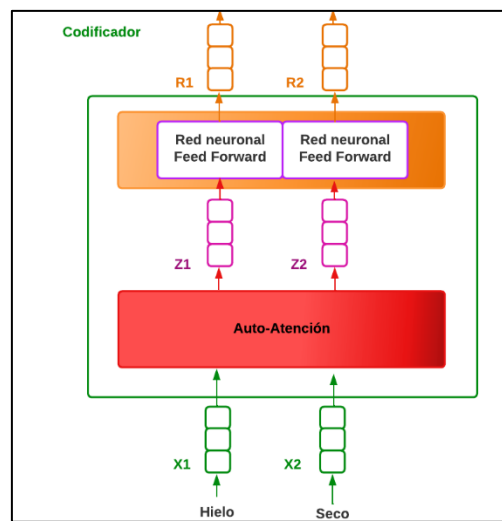


Figura 10. Representación de palabras mediante vectores de cuadros simples.

Cálculo de la Autoatención

Como primer paso, para el cálculo de la auto atención hay que generar tres vectores (Queries, Keys y Values) para cada palabra que recibe el codificador como entrada, los cuales se obtienen al multiplicar la entrada mediante tres matrices entrenadas previamente. Hay que tener en cuenta que estos vectores nuevos tienen una dimensión más pequeña que el vector incrustado. Su dimensión es 64, mientras que los vectores de entrada tienen una dimensionalidad de 512.

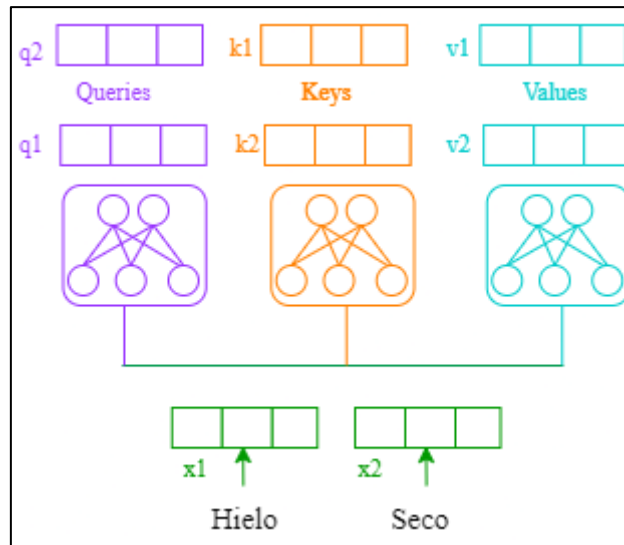


Figura 11. Vectores queries, keys y values.

Como segundo paso se necesita calcular un score (puntuación). Si se quiere calcular la auto atención de la primera palabra en este ejemplo, "Hielo", se necesita puntuar cada token de la frase de entrada contra esta palabra. La puntuación indica la cantidad de enfoque que se debe colocar en el resto de palabras que conforman la oración de entrada a medida que se codifica una palabra en una posición específica. Esta puntuación se consigue calculando el producto escalar del vector Queries con el vector Keys del token que se está puntuando. Si se está procesando la auto atención del token en la posición 1, la primera puntuación sería el producto de q1 y k1 y la segunda sería el producto de q2 y k2.

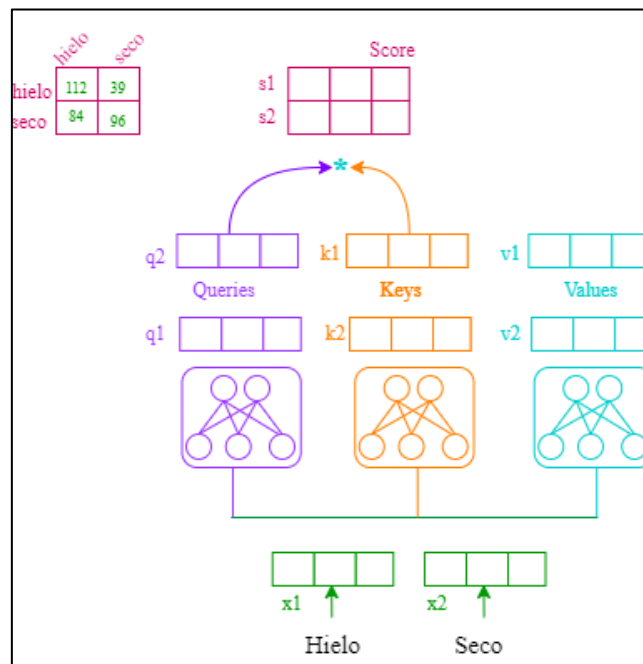


Figura 12. Calcular la puntuación.

El tercer paso consiste en dividir las puntuaciones entre el tamaño de la dimensión de los vectores Keys y luego pasar el resultado mediante la función softmax, que normalizará o representará cada puntuación como una probabilidad entre 0 y 1, donde, si el valor es cercano a 1, la red debe enfocará más atención a dicho token.

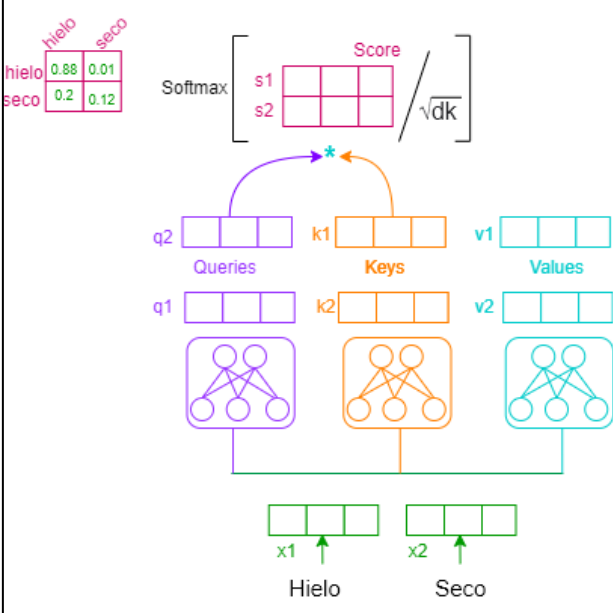


Figura 13. Normalización de la puntuación con la función softmax.

Como paso final se debe reducir la información en un único vector para cada token mediante la multiplicación de la matriz de puntajes con la matriz de Values, dando como resultado una serie de tokens nuevos, que tendrán la información más relevante para cada secuencia de palabras.

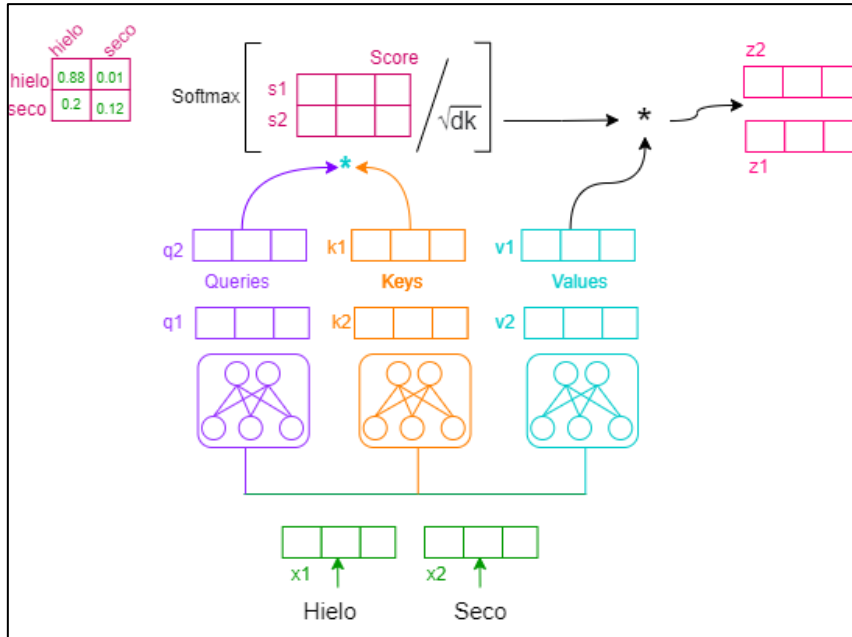


Figura 14. Salida de la capa de auto atención.

Sin embargo, hay que tener en cuenta que cuando se analiza una frase, se deben hacer asociaciones no solo entre palabras, sino también entre grupos de palabras, por lo que únicamente un bloque de atención es insuficiente. Al emplear múltiples bloques de atención es posible encontrar y codificar agrupaciones de palabras a diferentes niveles, donde la salida se combina en una última red neuronal que contiene la información resultante en un solo vector para cada token de entrada.

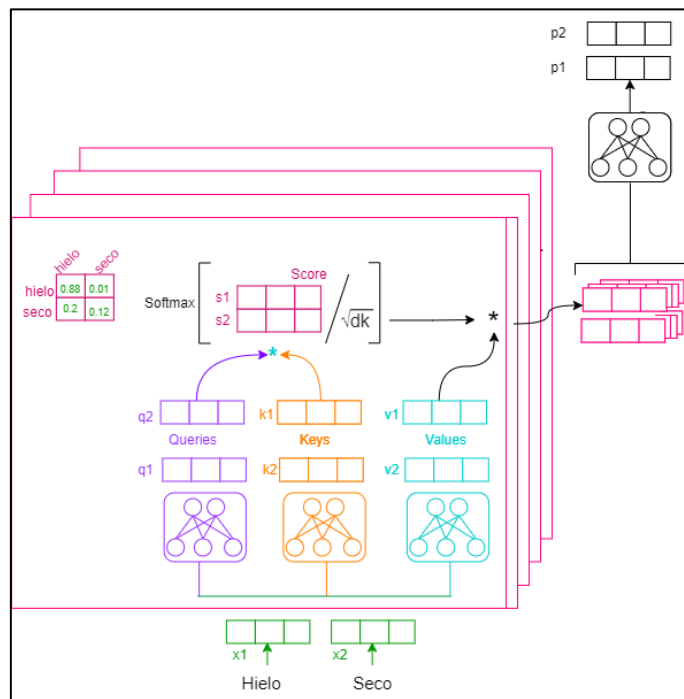


Figura 15. Múltiples bloques de atención.

4.9. DistilBERT

DistilBERT es un modelo de lenguaje que ha sido pre-entrenado mediante la técnica de Destilación de Conocimientos y tiene la misma arquitectura del modelo BERT_{base}. Sin embargo, la diferencia que existe con la versión original es que posee un 40% menos de parámetros (su número de capas se reduce a 6), es un 60% más rápido y presenta un 97% del rendimiento del modelo original [7].

4.10. Destilación de conocimientos

Este término fue acuñado por Bucila [22] en 2006 y desarrollado en 2015 de la mano de Hinton [23]. La técnica consiste en transferir el conocimiento de un modelo grande o complejo (Profesor) a un modelo de menor tamaño (Estudiante), sin perder el rendimiento del original y que se pueda ejecutar con recursos limitados [24].

4.11. Hugging Face

Hugging Face es una empresa construida sobre herramientas y datos de código abierto de resolución a problemas en el NLP. Esta empresa se centra en la creación de una comunidad entorno a los conceptos de consolidación de mejores prácticas y herramientas de última generación; ofrece acceso básico a su API de forma gratuita bajo una licencia “Apache 2.0”, donde se incluyen sus bibliotecas principales del NLP (transformadores, tokenizadores, conjuntos de datos y aceleradores), aunque se centra particularmente en modelos Transformers. Los Transformers han sido los modelos de aprendizaje profundo predominantes, utilizados en NLP durante los últimos años, como ejemplos bien conocidos son GPT-3 de OpenAI y sus predecesores, el modelo de Representaciones de Codificador Bidireccional de Transformers (BERT) desarrollado por Google, XLNet de Carnegie Mellon y Google, y muchos otros modelos y variantes [25].

4.12. Chatbot

Un chatbot es un programa de computadora, que actúa como un ente inteligente al establecer una conversación a través de texto o voz y entiende uno o más lenguajes humanos mediante el Procesamiento del Lenguaje Natural [26]. Los chatbots también son conocidos como bots inteligentes, agentes interactivos, asistentes digitales, agentes conversacionales o entidades de conversación artificial que pueden simular una conversación humana y también entretener a los usuarios con diversas actividades. Además, son útiles en aplicaciones como negocios, educación, comercio electrónico y recuperación de información.

4.12.1. Tipos de Chatbot

Los chatbots se pueden clasificar de acuerdo a una serie de parámetros: el dominio del conocimiento, el servicio proporcionado, los objetivos, el procesamiento de entrada y el método de generación de respuestas.

- **Clasificación basada en el dominio del conocimiento.** Son aquellos que pueden acceder a la cantidad de datos que se utilizan para su entrenamiento, y estos pueden ser de dominio abierto, que abarquen temas generales y responder adecuadamente; o de dominio cerrado donde se centran en una parte del conocimiento y pueden no responder a otras preguntas [27].
- **Clasificación basada en el servicio prestado.** Considera la aproximación sentimental del chatbot con el interesado, es decir la cantidad de interacción íntima que tiene con respecto a la tarea que realiza el agente conversacional con el usuario. Dentro de esta clasificación están [27]:
 - **Chatbots interpersonales.** Se encuentran en el dominio de la comunicación y brindan servicios como reservas de restaurantes, reservas de vuelos y bots de preguntas frecuentes. No son acompañantes del usuario, pero obtienen información y se la transmiten a los internautas. Pueden tener una personalidad amigable y probablemente recordarán información sobre el usuario, pero no están obligados a ello ni se espera que lo hagan.
 - **Chatbots intrapersonales.** Existen dentro del dominio personal del usuario, por ejemplo, Messenger, Slack o WhatsApp, y realizan tareas que se encuentran en el dominio personal del usuario.
 - **Chatbots Inter Agentes.** Son aquellos que establecen una comunicación entre dos o más sistemas inteligentes para poder realizar una tarea. La integración Alexa-Cortana es un ejemplo de comunicación entre agentes.
- **Clasificación basada en los objetivos.** Se clasifican según el objetivo principal que pretenden lograr [27]:
 - **Informativos.** Están diseñados para proporcionar al usuario información que se almacena de antemano o está disponible desde una fuente fija, como los chatbots de preguntas frecuentes.
 - **Conversacionales.** Hablan con el usuario, como otro ser humano, y su objetivo es responder correctamente a la oración que se les ha dado.
 - **Basados en tareas.** Realizan una tarea específica, como reservar un vuelo o ayudar a alguien. Estos chatbots son inteligentes en el contexto de solicitar

información y comprender la entrada del usuario. Los bots de reserva de restaurantes y los chatbots de preguntas frecuentes son ejemplos de asistentes virtuales basados en tareas.

- **Clasificación basada en el método de procesamiento de entradas y generación de respuestas.** Hay tres modelos utilizados para producir las respuestas adecuadas:
 - **Modelos basados en reglas.** La mayoría de chatbots presentan este tipo de arquitectura. Las respuestas dadas por estas inteligencias se basan en un conjunto fijo predefinido de reglas, a través del reconocimiento de la forma léxica del texto de entrada sin crear nuevas respuestas de texto. El conocimiento utilizado en el chatbot está codificado a mano humanamente y está organizado y presentado con patrones conversacionales [28].
 - **Basado en recuperación.** Ofrece más flexibilidad a medida que consulta y analiza los recursos disponibles mediante las APIs [29].
 - **El modelo generativo.** Genera respuestas de mejor manera que los otros tres modelos, basándose en los mensajes de los usuarios actuales y anteriores. Estos chatbots son más parecidos a los humanos y utilizan algoritmos de Machine Learning y técnicas de Deep Learning. Sin embargo, existen dificultades para construirlos y entrenarlos [29].

4.12.2. Arquitectura básica de un Chatbot

Los requerimientos para el diseño y desarrollo de un chatbot o agente conversacional, se basan en una representación de una estrategia para generar respuestas de forma precisa a partir del conocimiento, en las que se crea un conjunto de respuestas predefinidas que se pueden usar para responder las preguntas realizadas por el usuario.

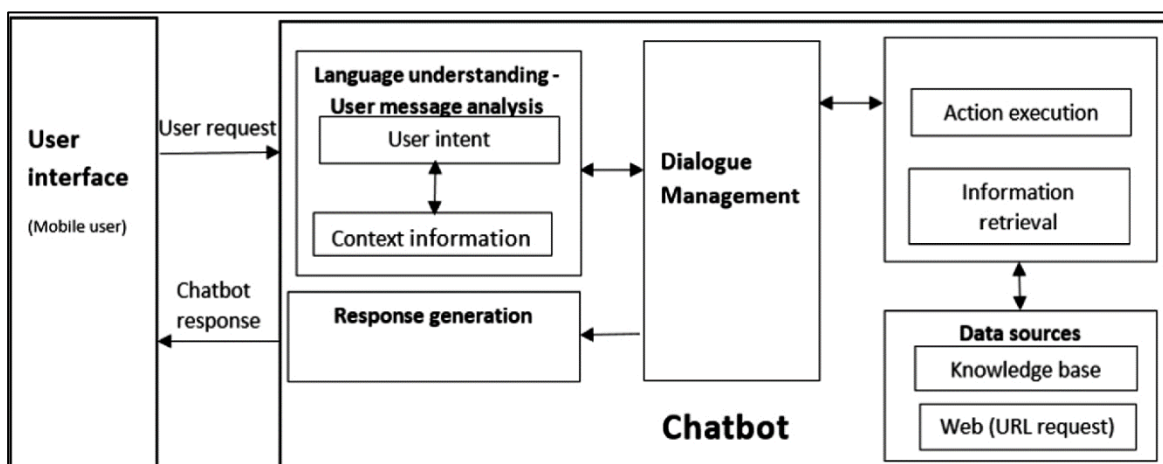


Figura 16. Interfaz básica de un chatbot.

El proceso inicia al momento en el que el usuario realiza la solicitud al chatbot mediante una interfaz gráfica, la cual recibe como entrada texto o audio. Después, el chatbot mediante un componente de comprensión del lenguaje analiza la solicitud para predecir la intención del usuario. Cuando el asistente conversacional logra la mejor interpretación, este debe decidir como continuar. Normalmente puede actuar sobre la nueva información, recordar lo que aprendió y esperar a ver qué sucede más adelante, pedir datos del contexto o requerir de una explicación. Cuando el agente conversacional entiende la solicitud, este procede a ejecutar una acción para recuperar la información de interés que el usuario desea ver. El chatbot recupera los datos de interés desde diferentes fuentes, como puede ser una base de datos, a la cual se la suele llamar Knowledge Base, o también desde otros recursos que se pueden tener acceso mediante una API. Después de la recuperación, el componente de generación de respuesta utiliza la Generación de Lenguaje Natural (NLG), para construir una respuesta lo más parecida a la que realiza una persona en lenguaje natural, teniendo en cuenta la intención y la información extraída de los contextos. Un componente de gestión del diálogo actualiza y mantiene la información que forma parte de los contextos de una conversación, que representa la intención actual, las entidades identificadas o las entidades faltantes requeridas para cumplir con las solicitudes de los usuarios. Además, requiere de información ausente, solicita aclaraciones de los interesados y realiza preguntas de seguimiento [26].

4.12.3. Características de un Chatbot

A continuación se presentan las características que resaltan en un agente iterativo [30]:

- **Autónomo.** Capacidad de actuar en base a su propia experiencia y de adecuarse a cualquier entorno.
- **Sociabilidad.** Capacidad de establecer una comunicación con otros sistemas o agentes.
- **Racionalidad.** Capacidad de formular, analizar y brindar respuestas adecuadas.
- **Adaptabilidad.** Capacidad de establecer diferentes comportamientos en su entorno de acuerdo al tipo de aprendizaje obtenido.
- **Movilidad.** Capacidad de adaptarse en diferentes plataformas.
- **Veracidad.** Capacidad de informar con la verdad.

4.13. Tecnologías y Herramientas

4.13.1. JavaScript

JavaScript es un lenguaje empleado para el desarrollo de páginas web permitiendo agregar interacciones como por ejemplo animaciones, menús emergentes, botones con efectos, entre

otros. JavaScript es un lenguaje de programación interpretado, es decir, se puede probar directamente en diferentes navegadores sin la instalación de intérpretes intermedios. [31].

4.13.2. HTML

HTML es un lenguaje que permite definir la estructura de las páginas web, el cual contiene una gran variedad de etiquetas que se usan para detallar los elementos que contendrá una página web, como listas, imágenes, tablas, formularios, vídeos, etc. HTML es un lenguaje de marcado de elementos para el desarrollo de documentos de hipertexto, de fácil aprendizaje y que permite incorporar otros lenguajes como JavaScript y CSS [32].

4.13.3. CSS

Es un lenguaje que permite presentar documentos con estilo, es decir, define como se va a presentar un documento de forma visual en un navegador. CSS permite dar estilos a los contenidos escritos en HTML como por ejemplo, colores, márgenes, fondos, bordes, tipo de letra, entre otros, permitiendo la modificación de un sitio web de una manera más atractiva y sencilla [32].

4.13.4. Python

Se trata de un lenguaje interpretado o de script orientado a objetos, el cual utiliza un software intermedio llamado intérprete para la ejecución de código. Python presenta una sintaxis clara, sencilla y simple, además, tiene una gran cantidad de librerías disponibles que hacen que el desarrollo de código sea muy rápido a comparación de otros lenguajes [33].

4.13.5. Flask

Es un microframework diseñado para crear aplicaciones web en poco tiempo con pocas líneas de código. Es un marco de trabajo que puede ser utilizado tanto para el backend puro como para el frontend si es necesario [34]. La palabra “micro” no significa que Flask se utilice para proyectos pequeños, sino que, al instalar el framework éste viene con las herramientas imprescindibles para la creación una página web, pero si en algún momento se requiere de nuevas funcionalidades, existe un conjunto de extensiones (plugins) que se pueden instalar con Flask para dar funcionalidad a las aplicaciones web.

4.13.6. Google Colab

Es un cuaderno Jupyter o un servicio gratuito alojado en la nube que permite a los usuarios el uso de GPUs y TPUs sin necesidad de realizar un pago extra. Colab permite hacer la programación en una estructura de cuadernos en Python facilitando a los usuarios el uso de diversas librerías como Keras, TensorFlow o PyTorch, así como la ejecución de comandos Shell, enseñar aprendizaje automático escribiendo texto, explicaciones y códigos de Python a través del navegador.

4.13.7. GitHub

Es un servicio de control de versiones basado en la nube, que permite llevar a cabo la colaboración entre desarrolladores de software y realizar cambios en proyectos compartidos; y al mismo tiempo llevar un seguimiento minucioso de cada fase del proyecto. El control de versiones es un software que permite gestionar y rastrear los diferentes cambios que se producen en los archivos, principalmente utilizado para ver las modificaciones generadas en el código.

4.13.8. GitLab

Es una herramienta que los desarrolladores de software usan para almacenar, colaborar, probar y compartir una gran cantidad de proyectos de tamaños diferentes hasta un máximo de 10 GB por repositorio. GitLab también ofrece un seguimiento mediante un control de versiones que ayuda a controlar los cambios que se producen en los archivos de un proyecto.

4.13.9. PyTorch

PyTorch es una biblioteca de código abierto creada por el laboratorio de Inteligencia Artificial de Facebook para programas de Python, que fomenta el software de aprendizaje profundo. Está basada en la biblioteca de Torch, y se utiliza en el Deep Learning para aplicaciones como visión artificial y Procesamiento de Lenguajes Naturales. El objetivo de este framework es permitir la implementación y el entrenamiento de modelos basados en el aprendizaje profundo de forma eficiente y sencilla.

PyTorch fue diseñado para minimizar el tiempo de ejecución de los algoritmos y usar hardware de una manera más eficiente al representar los datos en tablas multidimensionales, llamadas “tensores”. Los tensores almacenan las entradas de las redes neuronales, los parámetros de los hidden, layers y las salidas. Mediante esos tensores, PyTorch puede realizar a cabo 4 etapas para entrenar una red neuronal [35]:

- Montar un gráfico a partir de los tensores de la red neuronal, lo que permite una estructura dinámica, es posible modificarla (número de nudos, conexiones entre ellos, etc.) durante el aprendizaje.
- Efectuar las predicciones de la red (“forward pass”).
- Calcular la pérdida o error respecto a las predicciones.
- Atravesar la red en el sentido inverso: “backpropagation”, y ajustar los tensores, para que la red haga predicciones más precisas basadas en la pérdida/error calculado.

5. Metodología

5.1. Contexto

El presente TT, se desarrolló en la carrera de Sistemas/Computación de la Universidad Nacional de Loja (UNL), en la Facultad de Energía, las Industrias y los Recursos Naturales no Renovables (FEIRNNR); este trabajo se enfocó en el desarrollo de un chatbot para resolver inquietudes académicas sobre los estudios de posgrados, empleando como red neuronal el modelo de lenguaje BERT (en específico la versión DistilBERT). Para ello, se empezó realizando un análisis del Reglamento de Régimen Académico de la UNL, tomando en cuenta los artículos relacionados con los estudios de posgrados. A continuación, se elaboró una RSL, la cual permitió obtener las bases de conocimientos para la elaboración de todo el proyecto. Posteriormente se procedió a desarrollar el chatbot siguiendo las 4 fases de la metodología XP; y finalmente se ejecutaron pruebas de validación, rendimiento, carga, estrés y pruebas en un entorno con estudiantes de la carrera de Sistemas/Computación para ver la eficiencia del mismo.

5.2. Proceso

En el proceso de investigación se realizó una serie de etapas que permitieron seguir un orden lógico para la elaboración del presente TT.

Análisis

Inicialmente, fue necesario hacer un análisis del Reglamento de Régimen Académico de la UNL para seleccionar los procesos de posgrado que se tendrán en cuenta en el desarrollo del chatbot. Por otro lado, también fue imprescindible conocer el funcionamiento de la arquitectura del Modelo de Lenguaje BERT en el procesamiento del Lenguaje Natural, así como las tecnologías y herramientas más utilizadas para su entrenamiento y ajuste fino. Para obtener la información, se procedió a llevar a cabo una Revisión Sistemática de Literatura siguiendo las etapas propuestas por Bárbara Kitchenham [36]. La revisión se dividió en 4 etapas, que son:

- **Planificación de la RSL.** Etapa en la que se especificaron los cimientos para realizar la investigación; se establecieron preguntas de investigación, palabras clave, selección de las bases de datos científicas, estrategias de búsqueda, elaboración de las cadenas de búsqueda, criterios de inclusión y exclusión para seleccionar los artículos finales.
- **Ejecución de la Revisión Sistemática de Literatura:** Etapa donde se realizó la ejecución de la búsqueda de información en las respectivas bases de datos científicas.

- **Evaluación de la calidad de los artículos.** Fase en la cual se empezó aplicar los filtros y lecturas críticas para descartar y seleccionar los documentos finales que sirvieron en el TT.
- **Extracción y documentación de los resultados.** Etapa final donde se extrae la información que ayudó a responder las preguntas de investigación planteadas en la primera etapa de la RSL.

5.3. Métodos

- **Método Científico.** Es un método empleado en la producción de conocimiento, mediante el cual se puede obtener un conocimiento de la realidad, para responder a las interrogantes. Este método se usó en la obtención de información relacionada con el TT, mediante una investigación en diferentes documentos de fuentes confiables. Se recopiló, analizó y seleccionó la información necesaria para realizar las actividades del primer objetivo.
- **Método Analítico.** Es un método que ayudó a descomponer un todo, una idea o un objeto en elementos, yendo de lo particular a lo específico, ya que permitió analizar minuciosamente hechos particulares. El método se lo empleó en el primer objetivo para analizar la información sobre los estudios de cuarto nivel que están regidos en el Reglamento de la UNL, así como también en la RSL para seleccionar los estudios que dieron respuesta a las preguntas de investigación.

5.4. Técnicas

- **Revisión Bibliográfica.** Es una técnica que permite encontrar, recolectar, y extraer información científica de fuentes confiables como: páginas web, artículos, libros, bases de datos científicas, revistas científicas, entre otros. Gracias a esta técnica se pudo obtener los conocimientos necesarios para la elaboración del presente TT.
- **Encuesta.** Es una técnica muy efectiva al momento de recolectar datos de forma rápida y eficiente ya que permite obtener muestras de temas muy diversos. La técnica fue empleada para obtener información sobre los posgrados y también para ver el grado de satisfacción de los usuarios al usar el chatbot.
- **Tutorías.** Se realizaron reuniones con el docente asesor/director, que fueron de mucha importancia para la elaboración del TT, ya que establecieron los parámetros que se debían seguir para el cumplimiento de los objetivos y la elaboración del documento.

- **Muestreo por conveniencia.** Es una técnica que permitió al investigador escoger una cantidad de individuos que formaron parte de una muestra de acuerdo a ciertos criterios de facilidad, como la proximidad o la disponibilidad de los sujetos.

5.5. Metodología

- **Metodología de Bárbara Kitchenham** [36]. Es una metodología muy flexible con la cual fue posible realizar una Revisión Sistemática de Literatura, permitiendo seleccionar los estudios más importantes para dar respuesta a las preguntas de investigación sobre una temática o área en específica, y que fue de vital importancia para adquirir conocimientos sobre el objeto de estudio.
- **Metodología XP.** Permitted realizar de manera rápida, el análisis, diseño, desarrollo y las pruebas del agente conversacional que se elaboró. La metodología consta de las siguientes fases que fueron implementadas en el desarrollo del agente conversacional.
 - **Planeación.** Fase donde se identificaron los requisitos necesarios (versión y entrenamiento del modelo) para la elaboración del chatbot. Además, en esta fase se crearon las historias de usuario, las cuáles permitieron describir el funcionamiento del agente conversacional.
 - **Diseño.** Etapa que permitió desarrollar una arquitectura general del agente conversacional y la interacción de todos los elementos que intervinieron, como el usuario, la interfaz gráfica y las tecnologías que se usaron.
 - **Codificación.** Fase donde se realizó la programación para darle funcionalidad al chatbot empleando las tecnologías de Python, JavaScript, HTML, CSS y Flask.
 - **Pruebas.** Etapa final empleada para realizar las pruebas al agente conversacional.

6. Resultados

En esta parte se presentan las actividades, fases y tareas empleadas para el desarrollo del presente trabajo.

Objetivo 1: Definir los procesos académicos de posgrados y examinar el estado del arte sobre el modelo BERT empleado en el Procesamiento del Lenguaje Natural.

Para el cumplimiento del siguiente objetivo se procedió a realizar un análisis del Reglamento de Régimen Académico de la Universidad Nacional de Loja, donde se definieron los procesos académicos que se tomaron en cuenta sobre los estudios de posgrados. Posterior a ello, se realizó una revisión sistemática de literatura acerca del modelo BERT en el Procesamiento del Lenguaje Natural, donde se pudo identificar el funcionamiento, las tecnologías y herramientas más adecuadas para su entrenamiento y ajuste fino.

Tarea 1: Revisar normativas y reglamentos de posgrado de la UNL

En esta actividad, se analizó mediante una lectura el Reglamento de Régimen Académico de la UNL, que permitió conocer la estructura y los artículos relacionados con los estudios de posgrados. El Reglamento consta de 372 artículos divididos en seis partes de la siguiente manera:

- Título I. Aspectos Generales.
- Título II. Docencia.
- Título III. Régimen Estudiantil.
- Título IV. Títulos Otorgados por la Universidad.
- Título V. Función de Investigación.
- Título VI. Vinculación con la Colectividad.

El análisis del reglamento permitió separar la información relacionada con los estudios de cuarto nivel, tomando énfasis en el Título número III, denominado, Régimen Estudiantil; en esta temática se abordó el Capítulo I, de la Admisión y Matrícula; y el Capítulo VI, Reconocimiento u Homologaciones. En el capítulo I se abarcó la Sección I: De la admisión, Sección IV: De la Matrícula y la Sección V: De los Costos y Financiamientos de los Estudios de Posgrado. El capítulo VI se lo abordó de forma general debido a que no está dividido en secciones. Para tener una mejor visión de los temas que se contemplaron, en la **Figura 17** se muestra un esquema resumen detallando los dos elementos principales que se trataron en este proyecto.

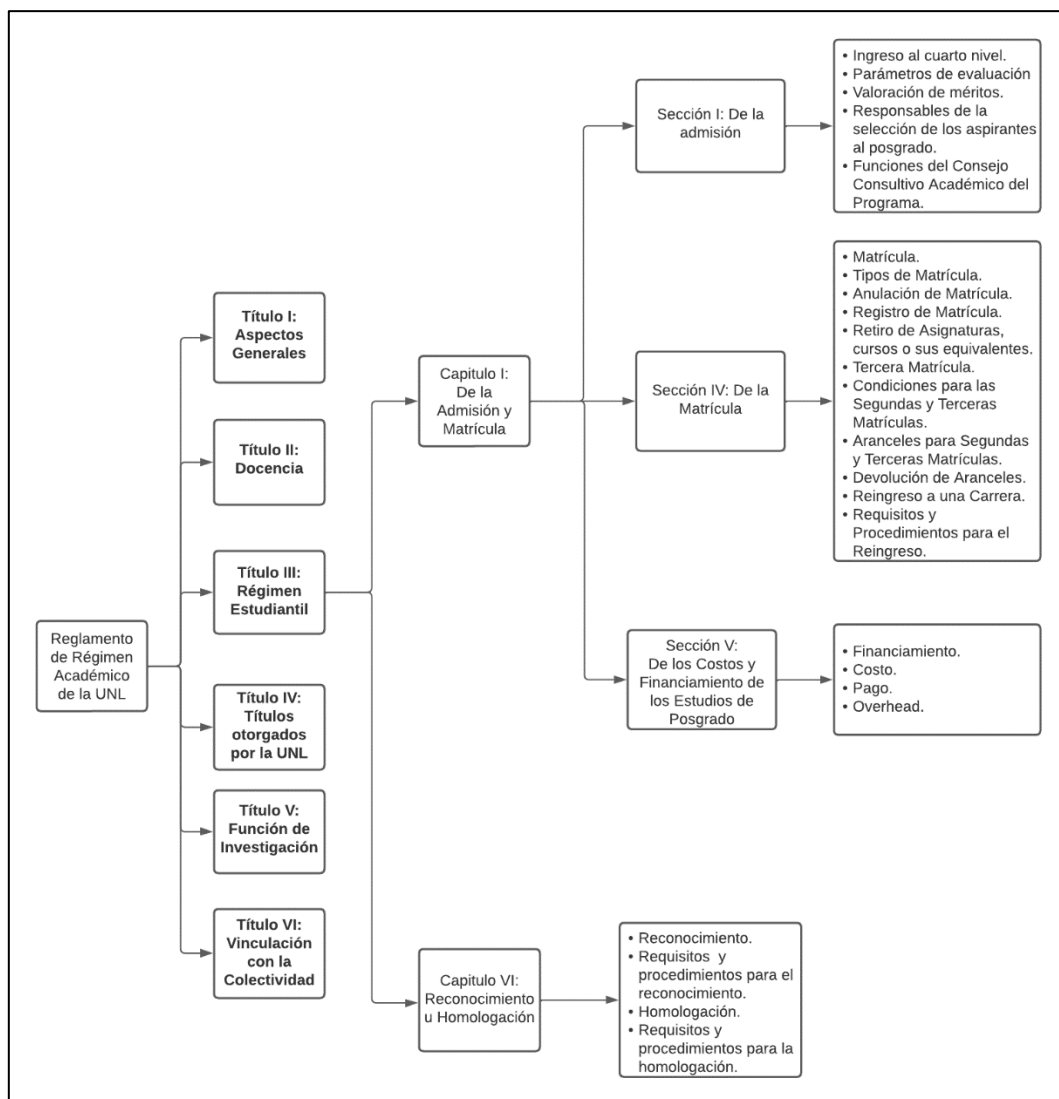


Figura 17. Esquema resumen de los procesos de matrículas y homologaciones.

Tarea 2: Definir los procesos académicos de posgrados a considerar en el Chatbot

En la siguiente actividad, se procedió a plasmar las dos temáticas mencionadas en la tarea anterior mediante diagramas de procesos, en la cual se utilizó el estándar BPMN, un lenguaje de modelado para representar de forma gráfica los procesos [8] de matrículas y homologaciones, tomando como referencia la información estipulada en el Reglamento de Régimen Académico de la UNL.

El proceso de matriculación se lo dividió en tres partes: Inscripción, Selección y Registro de matrícula.

En la inscripción a un programa de posgrado el aspirante debe seguir los siguientes pasos:

- Registrarse en el Sistema de Gestión de Inscripciones (SGI), en el cual deberá proporcionar sus datos personales (nombres y apellidos, teléfono, cédula, etc) y la información del domicilio. Con ello, el SGI creará las credenciales y notificará (vía

correo electrónico) el acceso al aspirante.

- Adjuntar los requisitos. Una vez que el SGI haya creado las credenciales del aspirante, este puede acceder al sistema, donde podrá escoger el programa de posgrado al que desea inscribirse, en el cual deberá adjuntar la siguiente documentación:
 - La solicitud de admisión al posgrado, dirigida al decano o decana de la facultad o al director/a de la Unidad de Educación a Distancia.
 - Copia del título de tercer nivel o de grado.
 - Documentos personales.
 - Hoja de vida.
 - Récord académico.
 - Carta donde el aspirante argumente las razones por las que desea estudiar el programa de posgrado.

Para evidenciar el proceso de inscripción del aspirante, se desarrolló un diagrama BPMN usando la herramienta “Bizagi Modeler”, donde se muestran las actividades o flujo de trabajo del proceso (**Figura 18**).

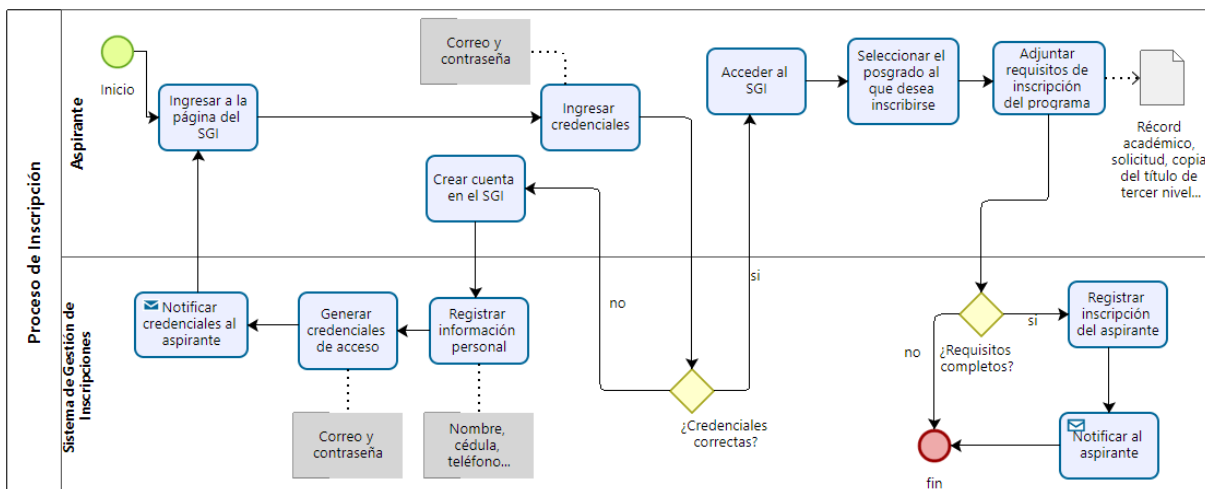


Figura 18. Proceso de inscripción a un programa de posgrado.

El proceso de selección es llevado a cabo por el Consejo Consultivo Académico, conformado por integrantes que poseen un título de cuarto nivel. El consejo será responsable de:

- Ejecutar el proceso de selección.
- Evaluar los méritos del aspirante sobre 10 puntos. Los parámetros de evaluación a considerar son:
 - Récord académico.
 - Carta de motivación.
 - Entrevista.

- Calificar la autenticidad de todos los documentos presentados por el aspirante.
- Elaborar el informe para el decano con los resultados de la selección.
- Publicar los resultados obtenidos de cada aspirante en el proceso de selección.
- Notificar los resultados a los aspirantes (puntaje y asignación de cupos).

El proceso de selección se lo puede observar en la **Figura 19**.

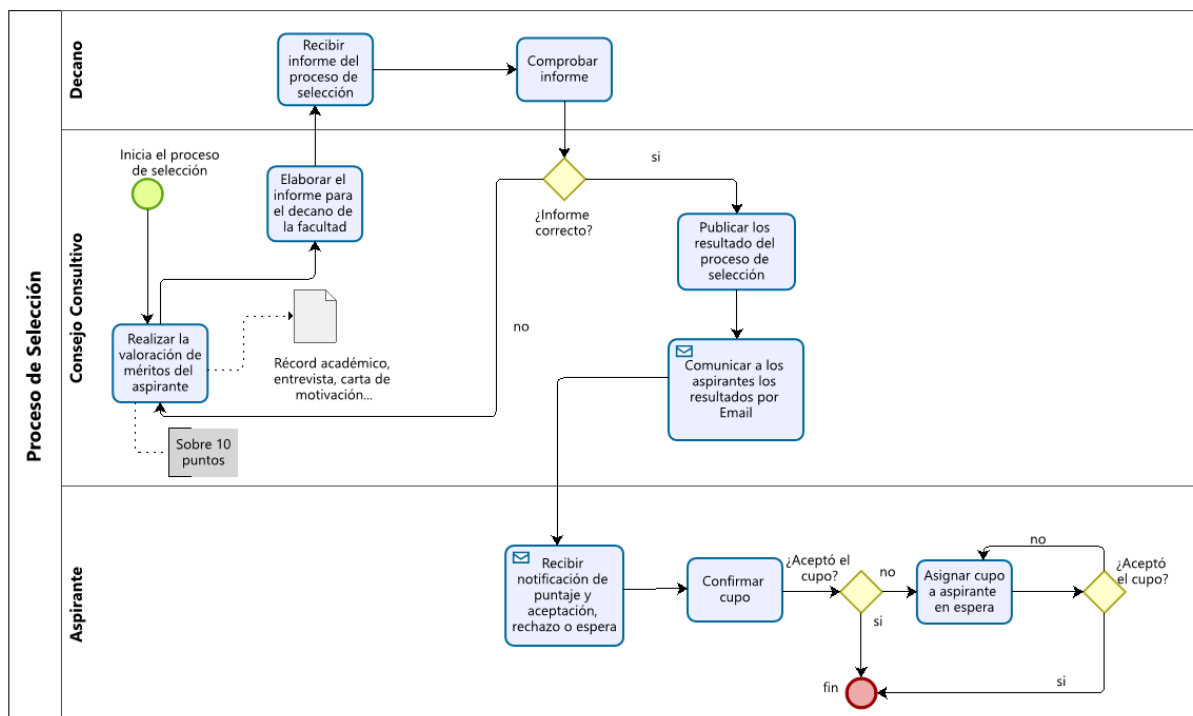


Figura 19. Proceso de selección para un programa de posgrado.

El registro de la matrícula es inicializado por el estudiante, el cual deberá acceder al SGA, donde realizará la solicitud de matrícula y adjuntará la siguiente documentación:

- Título de tercer nivel.
- Documentos personales.
- Fotografía tamaño carnet.

Posteriormente, el estudiante debe cancelar los pagos de aranceles correspondientes al posgrado y enviar el comprobante de pago a tesorería, quien se encargará de registrar la matrícula del estudiante. Este proceso se observa en la **Figura 20**.

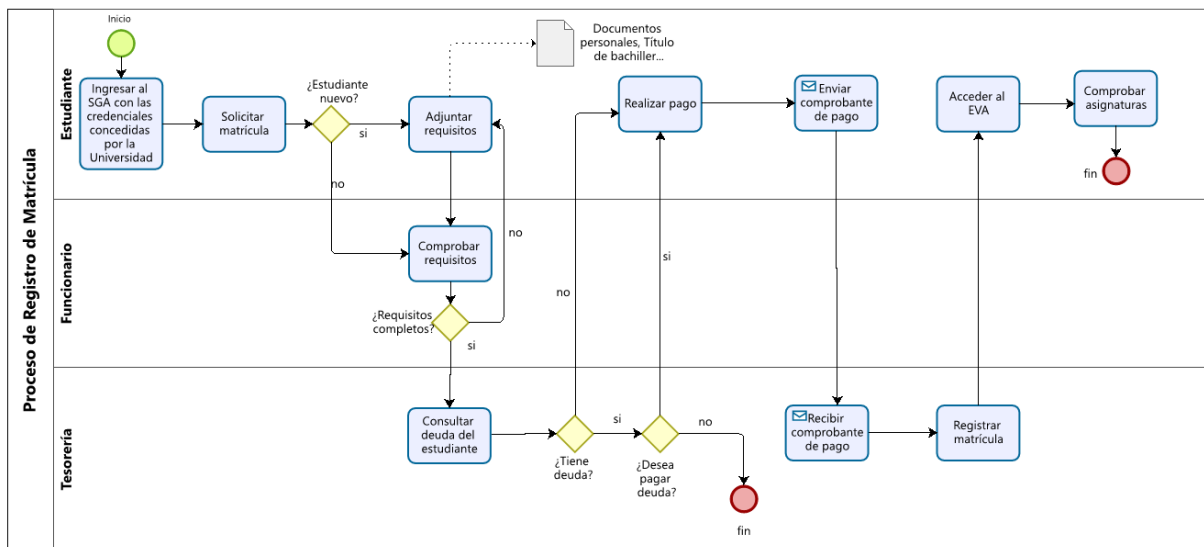


Figura 20. Proceso de registro de matrícula.

Para realizar la homologación el estudiante necesita contar con los siguientes requerimientos:

- Elaborar la solicitud de homologación dirigida al decano o decana de la facultad o al director de la unidad de educación a distancia. Para este requerimiento el estudiante debe adjuntar lo siguiente:
 - Récord académico.
 - Certificación de no haber agotado la tercera matrícula en la carrera de destino.
 - Programas analíticos, sílabos o similares de las asignaturas aprobadas.
 - Si los aspirantes son extranjeros o sus estudios fueron realizados en el exterior, la documentación que deberá presentarse tiene que estar traducida al español.
- El decano acepta el trámite y remitirá la solicitud con la documentación del estudiante al director del programa.
- El director del programa escogerá los docentes pertinentes para que realicen el proceso y el informe de la homologación.
- El director del programa enviará un informe al decano.
- El decano enviará la aprobación o la negación de la homologación.
- La resolución de la homologación será notificada al estudiante donde constará los cursos, asignaturas o equivalentes que han sido homologadas y las que deberá aprobar si fuese el caso. En la **Figura 21** se presenta el proceso.

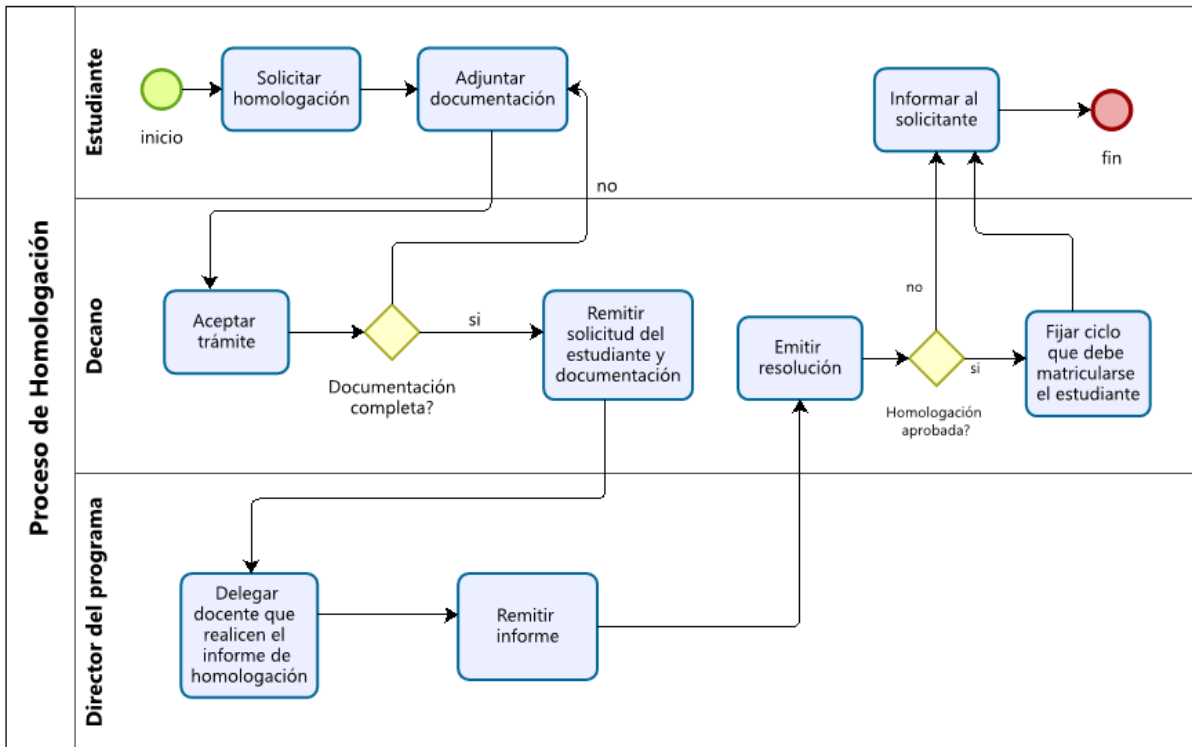


Figura 21. Proceso de homologación.

Tarea 3: Realizar una búsqueda sistemática de literatura.

La elaboración de la RSL se llevó a cabo mediante algunas fases propuestas por Bárbara Kitchenham [36]. Esta revisión literaria consta de cuatro fases: Planificar la RSL, ejecutar la RSL, evaluar la calidad de los artículos, y extraer y documentar los resultados.

Según Barbara, es posible omitir la ejecución de algunas fases en una RSL como por ejemplo: la puesta en marcha de una revisión, la evaluación de un protocolo de revisión y la valuación del informe. A continuación, se muestra detalladamente las cuatro fases que sirvieron para la elaboración y obtención de los resultados presentados a través de la RSL.

- Planificación de la RSL:
 - Elaborar las preguntas de investigación.
 - Determinar las palabras clave.
 - Crear las cadenas de búsqueda.
 - Seleccionar las bases de datos.
 - Establecer criterios de selección.
- Ejecución de la RSL:
 - Ejecutar las cadenas de búsqueda en las bases de datos seleccionadas.
- Evaluar la calidad de los artículos:
 - Selección de los estudios primarios.
 - Síntesis de los datos.

- Extraer y documentar los resultados.

Planificación de la revisión

En esta fase se definieron los cimientos necesarios para realizar la presente investigación mediante una Revisión Sistemática de Literatura, la cual comprende las siguientes actividades.

Preguntas de investigación

Para comenzar con la RSL, fue necesario determinar las preguntas de investigación que ayudaron a obtener la información más relevante acerca del objeto de estudio. De esta manera se plantearon 3 preguntas de investigación (**véase la Tabla 1**), las cuales fueron de gran importancia en todo el proceso de ejecución de este trabajo.

Tabla 1. Preguntas de investigación.
ID: código que identifica la pregunta. **Preguntas de investigación:** consultas elaboradas para la búsqueda de información.

ID	Preguntas de Investigación
P1	¿Cómo funciona la arquitectura del modelo BERT en el Procesamiento del Lenguaje Natural?
P2	¿Qué tecnologías y herramientas son compatibles para el entrenamiento y ajuste del modelo BERT en la tarea de preguntas y respuestas?
P3	¿Cuál es el tamaño máximo de secuencia de entrada que el modelo BERT permite y qué técnicas se pueden utilizar para resolver este problema?

Palabras clave

Se hizo uso del control terminológico IEEE Thesaurus [37] el cual permitió obtener un conjunto de vocabulario técnico y científico para la elaboración de las cadenas de búsqueda. En la **Tabla 2**, se presentan las palabras que se obtuvieron y utilizaron para ejecutar una búsqueda exhaustiva de información de mayor calidad en la obtención de resultados que aporten valor para responder las preguntas de investigación.

Tabla 2. Palabras clave.

Palabras clave: términos para la búsqueda de información.

Traducción: términos traducidos al idioma inglés.

Palabras clave	Traducción
Modelo BERT	BERT Model
Representación de Codificadores Bidireccionales de Transformadores	Bidirectional Encoder Representation from Transformers
Corpus	Corpus
Procesamiento del Lenguaje Natural	Natural Language Processing
Pregunta	Question
Respuesta	Answer
Arquitectura del modelo BERT	Architecture of te BERT Model
Tecnologías	Tecnologies

Cadenas de Búsqueda

Las cadenas de búsqueda de la presente RSL se crearon siguiendo la estructura del lenguaje SQL, donde se usaron los operadores lógicos OR y AND que se pueden aplicar en la mayoría de bases de datos científicas. Con todo esto se procedió a construir una cadena de búsqueda general, tal y como se muestra en la **Tabla 3**.

Tabla 3. Cadena de búsqueda.

ID: código que identifica la cadena de búsqueda. **Cadena de Búsqueda:** estructura elaborada a partir de términos clave para la búsqueda de información.

ID	Cadena de Búsqueda
CB_1	("BERT Model" OR "Bidirectional lenguaje model" OR "BERT") AND ("Architecture of the BERT" OR " Architecture of te BERT Model") AND ("Dataset" OR Corpus) AND ("Natural Language Processing") AND ("Question") AND ("Answer") AND ("Tecnologies" Or "Tols")

Bases de Datos

En la **Tabla 4**, se aprecian las bases de datos científicas que se utilizaron en la presente RSL con su dirección URL.

Tabla 4. Bases de datos científicas.

Base de datos: repositorios que contienen una gran variedad de artículos científicos. **URL:** dirección web de cada respectiva base de datos.

Base de datos	URL
ACM Digital Library	https://www.acm.org/publications/digital-library
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
Scopus	https://www.scopus.com/home.uri
arXiv	https://arxiv.org/

Criterios de Selección

En esta actividad se definieron las pautas que se utilizaron para la elección de los documentos relacionados con las preguntas de investigación. A continuación, se detallan los criterios de inclusión y exclusión que se aplicaron a los artículos científicos, teniendo en cuenta las preguntas de investigación.

- **Criterios de inclusión.** En la **Tabla 5** se especificaron los criterios de inclusión que fueron aplicados para seleccionar los artículos que ayudaron a responder las preguntas de investigación.

Tabla 5. Criterios de inclusión.

Criterio: característica que permitió incluir los estudios finales.
Descripción: explicación que se tomó en cuenta para cada criterio.

Criterio	Descripción
Fecha de publicación	Artículos publicados entre en el año 2018 a 2022.
Motores de Búsqueda	ACM, IEEE, Scopus, arXiv.
Idioma	Se consideraron todos los artículos en el idioma inglés y español.
Tipo de estudios	Artículos de revistas y artículos de conferencias.
Contenido	Artículos que se relacionen con las preguntas de investigación.

- **Criterios de exclusión.** Se usaron para descartar todos los artículos que no permitieron dar respuesta a las preguntas de investigación, los cuales se describen en la **Tabla 6**.

Tabla 6. Criterios de exclusión.

Criterio: característica que permitió excluir los estudios finales.
Descripción: explicación que se tomó en cuenta para cada criterio.

Criterio	Descripción
Fecha de publicación	Artículos publicados en fechas inferiores al año 2018.
Accesibilidad	Artículos a los que no se tiene acceso.
Idioma	Artículos diferentes al idioma inglés y español.
Duplicados	Artículos duplicados.
Contenido	Artículos no relacionados con las preguntas de investigación.

Ejecución de la revisión

Ejecutar las cadenas de búsqueda

En la **Tabla 7** se presentan las cadenas de búsqueda que se ejecutaron en cada respectiva base de datos científica (vea el **Anexo 1**) para esta RSL, mismas que arrojaron un resultado de 178 artículos. (**Nota colocada al pie de la siguiente página**).

Tabla 7. Ejecución de las cadenas de búsqueda.

Base de datos: repositorio que contiene artículos científicos. **Cadena de búsqueda:** secuencia ejecutada en cada base de datos. **Número de artículos:** cantidad de documentos encontrados en cada base de datos.

Base de datos	Cadenas de búsqueda	Número de Artículos
ACM Library	Title:(bert OR bidirectional encoder representation from transformers) AND Abstract:(question) AND Abstract:(architecture) AND Abstract:(data set) AND E-publication date:(01/01/2018 TO 12/31/2022)	13
IEEE Xplore	("Document Title": bidirectional encoder representation from transformers) OR ("Document Title":Bert) AND ("Document Title":Question) AND ("Document Title":Legal) AND ("Document Title":Answer) AND ("Abstract":Architecture)	21
Scopus	(TITLE ("bert" OR "Bidirectional Encoder Representation from Transformers") AND TITLE ("question" OR "answer" OR "tols" OR "technologies") AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019)) AND (LIMIT-TO (LANGUAGE , "English"))	120
arXiv	order: -announced_date_first; size: 50; hide_abstracts: True; date_range: from 2018-01-01 to 2022-12-31; include_cross_list: True; terms: AND title=BERT; AND title=question; AND bstract=answer ⁴	24
Total		178

Evaluar la calidad de los artículos

Selección de los estudios primarios

En la **Figura 22** se muestra el proceso de forma ordenada que se estableció para la selección de los estudios primarios.

⁴ La cadena de búsqueda de la base de datos arXiv presente en la **Tabla 7**, solo funciona al colocar los términos “BERT”, “Question” y “Answer” en la opción de “Búsqueda Avanzada” de la página web. Para mayor detalle siga los pasos del **Anexo 1 – Figura 4**, de esa manera podrá visualizar los resultados de los artículos obtenidos en dicha base de datos científica.

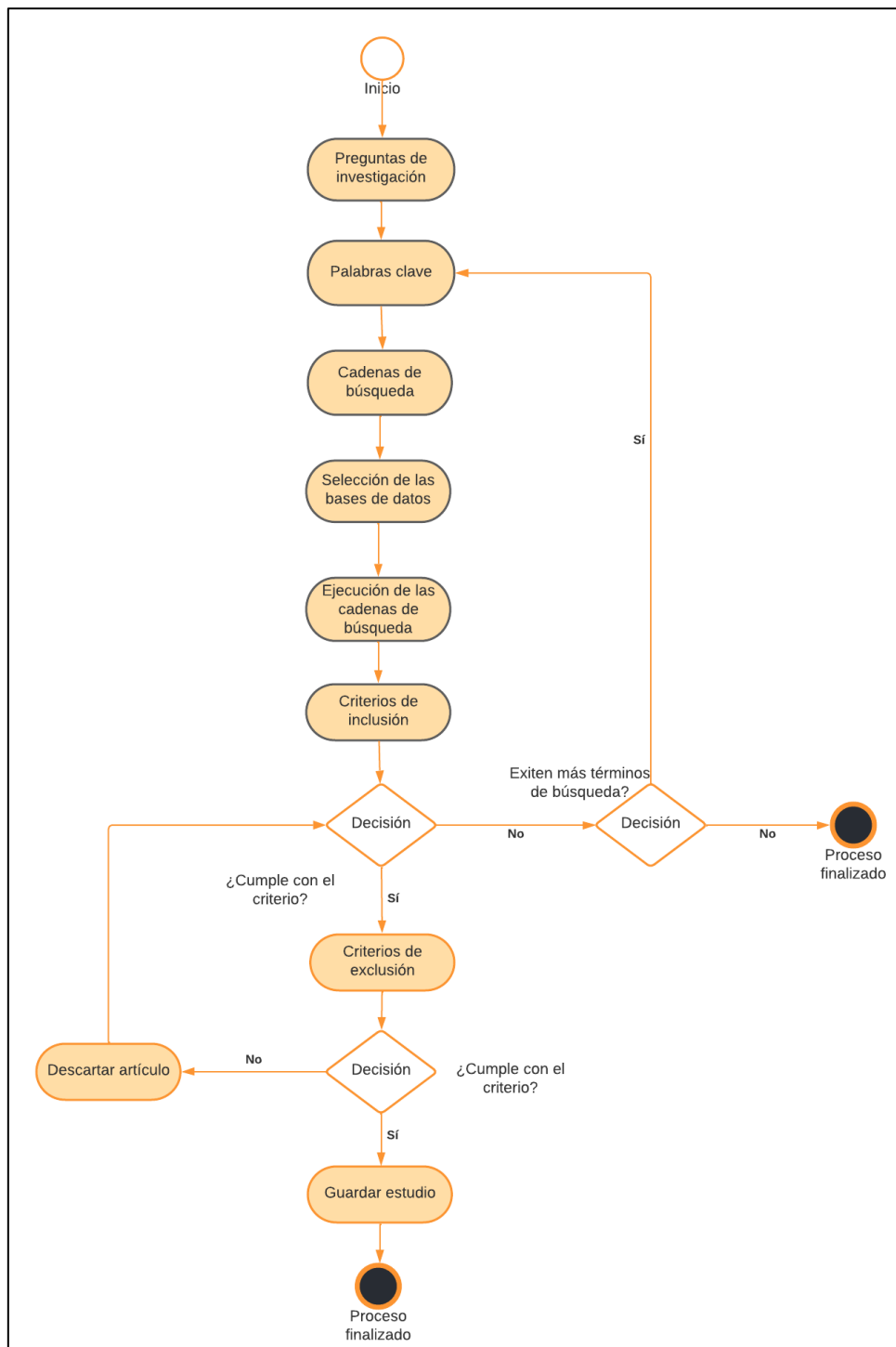


Figura 22. Proceso de selección de los estudios primarios.

Síntesis de los datos.

Para seleccionar los estudios que ayuden a responder las preguntas de investigación, se procedió a realizar un análisis detallado de todos los documentos proporcionados por las bases de datos científicas, permitiendo reducir la cantidad de los mismos, para posteriormente extraer la información más importante que ayude al objeto de estudio de esta RSL. En la **Figura 23** se presenta el proceso que se siguió para la selección de los artículos finales.

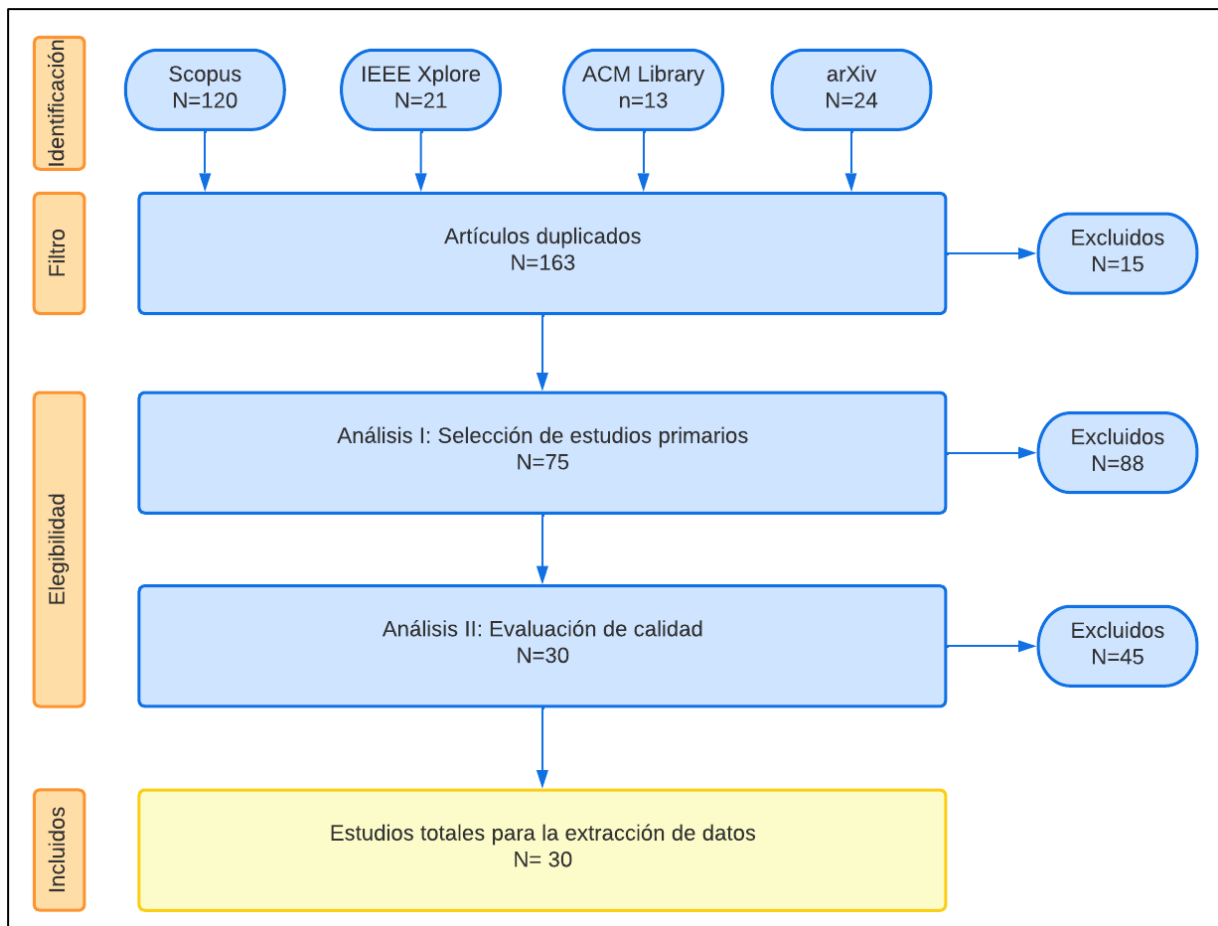


Figura 23. Proceso de selección de los artículos finales.

Extraer y documentar los resultados

En esta tarea se procedió a llevar a cabo la extracción de la información de los artículos seleccionados anteriormente y se procedió a contestar las preguntas de investigación:

¿Cómo funciona la arquitectura del modelo BERT en el Procesamiento del Lenguaje Natural?

Según los artículos [38], [4], [39], [40], [41], [42], [43], [44] y en especial, el artículo [5], el modelo de lenguaje BERT funciona de la siguiente manera:

- Hace uso de la bidireccionalidad, donde el modelo analiza una oración por completo a partir de una palabra clave, es decir, es capaz de interpretar un contexto tanto a la izquierda como a la derecha de una palabra principal, permitiendo a la red neuronal entender una frase entera.
- El modelo utiliza un codificador que es parte de la arquitectura de un Transformer, el cual hace uso de mecanismos de auto atención que se encargan de relacionar una palabra o grupos de palabras con los términos restantes de la secuencia de la oración que recibió como entrada, y de esta forma ver el grado de similitud que existe entre las

mismas mediante una puntuación, la cual se obtiene calculando el producto escalar de dos matrices seguida de una función softmax, para obtener dos valores altos que corresponden a los tokens de inicio y final de la respuesta que se extraen de un contexto.

- Para la secuencia de entrada el modelo utiliza WordPiece embedding, que posee una cantidad de 30.522 tokens. Al inicio de la secuencia de entrada se introduce un token especial llamado [CLS]. Si la secuencia de entrada son dos oraciones, el modelo interpretará dicha secuencia como una sola frase y coloca el token [CLS] al inicio de la oración y el token [SEP] en medio de ambas frases para diferenciarlas.
- El modelo presenta dos fases importantes: El Pre-entrenamiento, que se lo realiza con datos sin etiquetar; y el Ajuste Fino, que se lleva a cabo con texto etiquetado para una tarea específica que se quiera resolver. El Pre-entrenamiento tiene dos etapas; Modelado de lenguaje enmascarado y predicción de la siguiente oración. La primera etapa consiste en ocultar aleatoriamente un 15% de la secuencia de entrada con el token [MASK], para que el modelo sea capaz de predecir la palabra oculta y complete la oración. La segunda etapa se encarga de ver la relación que puede existir en un par de oraciones, es decir, el modelo recibe parejas de frases como entrada y aprende a predecir si la segunda oración del par es la que corresponde. Durante el entrenamiento, el 50% de las entradas son un par en el que la segunda frase es la siguiente que corresponde al documento fuente, mientras que en el otro 50% se elige una frase al azar del conjunto de datos como segunda oración.
- La arquitectura de BERT presenta dos tamaños originales: BERT_{base} (L=12, H=768, A=12, Parámetros: 110 M) y BERT_{large} (L=24, H=1024, A=16, Parámetros: 340 M), siendo L el número de capas, H tamaño de la capa oculta (tamaño de la dimensión) y A el número de encabezados de auto atención.

¿Qué tecnologías y herramientas son compatibles para el entrenamiento y ajuste del modelo BERT?

Para esta pregunta de investigación, se escogieron los documentos que más información mencionaban sobre herramientas y tecnologías que se pueden utilizar para ajustar el modelo BERT mediante un entrenamiento con algún conjunto de datos. En la **Tabla 8** se presenta el resultado de la investigación.

Tabla 8. Tecnologías y herramientas utilizadas para ajustar el modelo BERT.
Código: identificador único de cada artículo. **Título:** nombre principal de los documentos.
Tecnologías: herramientas tecnológicas empleadas en cada estudio.

Código	Título	Tecnologías
TR01	BERT-CoQAC: BERT-Based Conversational Question Answering in Context [45]	TensorFlow Optimizador Adam RAM 16 GB GPU NVIDIA Tesla P100 Biblioteca Transformers
TR02	Efficient Fine-Tuning of BERT Models on the Edge [46]	PyTorch Python Biblioteca de Transformers RAM 16 GB + GPU NVIDIA Tesla V100 Volta
TR03	BERT with history answer embedding for conversational question answering [47]	Python TensorFlow Optimizador Adam
TR04	Evaluating BERT-based Rewards for Question Generation with Reinforcement Learning [48]	Python PyTorch Biblioteca de Transformers
TR05	Automatic Numerical Question Answering on Table using BERT-GNN [49]	Python PyTorch
TR06	A Recurrent BERT-based Model for Question Generation [50]	Python PyTorch Optimizador Adam RAM 24 GB + GPU TITAN RTX
TR07	BERT for Conversational Question Answering Systems Using Semantic Similarity Estimation [51]	PyTorch Biblioteca de Transformers Optimizador Adam
TR08	BERT for Question Generation [52]	PyTorch Biblioteca de Transformers Optimizador Adam RAM 24 GB + GPU TITAN RTX
TR09	Infusing Disease Knowledge Into BERT for Health Question Answering, Medical Inference and Disease Name Recognition [53]	PyTorch Optimizador Adam RAM 16 GB + GPU NVIDIA V100
TR10	Pre-training BERT on Resources for Short Answer Grading [54]	TensorFlow Python

TR11	Predicting Subjective Features from Questions on QA Websites Using BERT [55]	TensorFlow Python Biblioteca de Transformers RAM 16 GB + CPU 2.00GHz. Intel(R) Xeon(R)
TR12	Questions and Answers on Legal Texts Based on BERT-BiGRU [56]	PyTorch Python Biblioteca de Transformers RAM 16 GB + GPU GTX-1080Ti
TR13	Two-stage Semantic Answer Type Prediction for Question Answering Using BERT and Class-specificity Rewarding [57]	PyTorch Google Colab RAM 24 GB + GPU Nvidia Tesla K80

¿Cuál es el tamaño máximo de secuencia de entrada que el modelo BERT permite y qué técnicas se pueden utilizar para resolver este problema?

Según el análisis de los artículos [38], [4], [40], [58], [59], [51], [60], [55] cualquier versión del modelo BERT permite una secuencia máxima de entrada de 512 tokens, siendo un problema cuando se quiere procesar textos relativamente grandes, ya que los tokens de entrada más largos no tendrán una incrustación en el modelo.

En el artículo [38] se presentan métodos que se pueden utilizar para tratar con textos largos cuya secuencia de entrada es mayor a 512 tokens. Las técnicas que se usan se denominan métodos de truncamiento que pueden emplearse de tres maneras: quedarse con los primeros 510 tokens; conservar los últimos 510 tokens; o seleccionar empíricamente los primeros 128 y los últimos 382 tokens. Dado que la información normalmente se encuentra al inicio o al final de un artículo, estos métodos son la mejor alternativa al tratar con textos extensos, sin embargo, de acuerdo a los autores, la mejor forma es tomar los primeros 128 y los últimos 382 tokens, ya que ofrecieron mejores resultados. Por otro lado, los estudios [58], [60] mencionan que uno de los mejores métodos, consiste en dividir los textos en partes o categorías según las preguntas realizadas.

Objetivo 2: Desarrollar una arquitectura de Chatbot o Agente Conversacional utilizando como red neuronal el modelo de lenguaje BERT.

El objetivo 2, consta de 5 tareas: 1) probar el modelo BERT con la ayuda de Google Colab; 2) ajustar el modelo BERT mediante un entrenamiento con información del reglamento de la UNL en lo que concierne a posgrados; 3) diseñar la arquitectura del agente conversacional; 4) desarrollar un prototipo web utilizando como lenguaje de programación Python y JavaScript; y 5) documentar la aplicación.

Para el cumplimiento del siguiente objetivo se emplearon las tres primeras fases de la metodología XP: Planeación, Diseño y Codificación. En la fase de la Planeación se identificaron los requisitos necesarios para el desarrollo del chatbot, que consta de lo siguiente:

- Realizar una comparativa de modelos de lenguaje basados en BERT y seleccionar una versión como red neuronal.
- Construir un dataset a partir de la información obtenida del Reglamento de Régimen Académico de la UNL y de datos obtenidos mediante una encuesta realizada a la secretaria de posgrados.
- Ejecutar el entrenamiento del modelo escogido.
- Crear las historias de usuario que describan el comportamiento del chatbot.

En la fase de diseño se creó la arquitectura del agente conversacional; y en la fase de codificación se llevó a cabo la programación, utilizando Python, JavaScript, Html, Css y el microframework FLask.

Tarea 1: Probar el modelo BERT con la ayuda de Google Colab.

Uno de los requisitos que se estableció en esta tarea fue la selección del modelo que serviría para la implementación de un chatbot. En la **Tabla 9** se presenta un cuadro comparativo de modelos de lenguaje ajustados previamente con el conjunto de datos SQuAD 2.0 (el conjunto de datos está en español) para la tarea de preguntas y respuestas. Por un lado, BERT_{base} y BERT_{large} son los modelos originales propuestos en el artículo [5]; y por otro, DistilBERT [7], versión ligera basada en BERT_{base} que se caracteriza por la poca cantidad de recursos de cómputo que necesita para tareas específicas.

Tabla 9. Cuadro comparativo de modelos de lenguaje basados en BERT.

Características: elementos comunes de los modelos de lenguaje. **DistilBERT:** modelo de lenguaje de 6 capas. **BERT_{base}:** modelo de lenguaje de 12 capas. **BERT_{large}:** modelo de lenguaje de 24 capas. Datos obtenidos de los artículos [6], [42], [5], [7].

Característica	DistilBERT	BERT_{base}	BERT_{large}
Número de capas	6	12	24
Número de encabezados de autoatención	12	12	16
Número de parámetros	66 millones	110 millones	340 millones
Tiempo de entrenamiento	3,5 días con 8 GPUs	4 días con 4 GPUs	4 días con 16 GPUs
Conjunto de datos de entrenamiento	SQuAD 2.0 en español	SQuAD 2.0 en español	SQuAD 2.0 en español
Velocidad de entrenamiento	Rápido	Más rápido que BERT _{large}	Muy lento
Recursos computacionales mínimos	1 GPU de 16 GB de memoria	4 GPU de 16 GB de memoria	8 GPU de 16 GB de memoria
Precisión (F1)	77,5	80,4	85,0

En base a la tabla anterior, se descartaron las versiones BERT_{base} y BERT_{large} que a pesar de que son los modelos con mejor precisión, estos requieren de bastantes recursos en cuanto al hardware (4 y 8 GPUs con memoria de 16 GB como mínimo). Hay que tener en cuenta que al utilizar la herramienta Google Colab ésta pone a disposición únicamente 1 GPU de 16 GB de memoria, provocando que el entrenamiento de los dos modelos mencionados sea muy lento y tarde demasiado, aunque esto también depende de la cantidad de datos que se utilicen. La versión restante (DistilBERT) es más rápida y solo necesitan 1 GPU de 16 GB, cumpliendo con la característica principal que exige Google Colab. De esta manera, se optó por escoger un modelo de 417 MB de peso, denominado “**mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es**”⁵.

Para probar el modelo se inició la tarea instalando la biblioteca Transformers desde Hugging Face en Google Colab. Esta biblioteca permitió usar ciertos elementos como (**ver Figura 24**):

Versión del modelo BERT (DistilBERT):

⁵ <https://huggingface.co/mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es>

- **AutoTokenizer.** Librería que ayudó a separar el texto en palabras para convertirlas en representaciones numéricas y agregar unos tokens especiales, ([CLS] al inicio del contexto y [SEP] en medio de dos oraciones para distinguir el contexto de la pregunta).
- **AutoModelForQuestionAnswering.** Con esta librería se pudo cargar el modelo pre-entrenado que se descargó desde Hugging Face.
- **Pipelin.** Es una librería que permitió hacer la inferencia con el modelo, es decir, que al introducir un contexto y una pregunta, se le va a pedir a DistilBERT que haga la deducción o encuentre la respuesta dentro de un contexto a una pregunta dada.

```
# Importar
from transformers import AutoTokenizer, AutoModelForQuestionAnswering, pipeline
the_model = 'mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es'
tokenizer = AutoTokenizer.from_pretrained(the_model, do_lower_case=False)
model = AutoModelForQuestionAnswering.from_pretrained(the_model)
```

Figura 24. Importación del modelo y las librerías.

Como se mencionó anteriormente, fue necesario usar la tokenización para representar las entradas hacia el modelo en valores numéricos, la cual funciona de la siguiente manera; se parte de un contexto y una pregunta, mismas que deben ser codificadas retornando un valor numérico para cada elemento del contexto y de la pregunta; la función `tokenizer.encode_plus()` permitió hacer esa transformación y de la misma forma agregar los tokens especiales [CLS] y [SEP]. En la Figura 25 se puede observar el script usado para el proceso de tokenización; y en la **Figura 26** se muestra el resultado que se obtuvo, donde cada palabra corresponde a un número (por ejemplo, “**Cómo**”, tiene el valor de 2.144). Además, también se agregó el token [CLS] al inicio de la pregunta y el token [SEP] al final de la misma, permitiendo la introducción del contexto. La secuencia numérica generada es la que se va a incrustar en el modelo y luego será representada como un Embedding que comenzará a pasar por los codificadores hasta que al final, a la salida de la última capa neuronal, el modelo pueda entregar dos posiciones que corresponden a la respuesta.

```
contexto="Mi nombre es Roberto"
pregunta="¿Cómo me llamo?"
codificar= tokenizer.encode_plus(pregunta,contexto, return_tensors='pt')
input_ids = codificar['input_ids'].tolist()
tokens = tokenizer.convert_ids_to_tokens(input_ids[0])
for id, token in zip(input_ids[0], tokens):
    print('{:<12} {:>1}'.format(token, id))
```

Figura 25. Tokenización del contexto y la pregunta.

[CLS]	4
¿	1067
Como	2144
me	1129
llamo	6861
?	1064
[SEP]	5
Mi	1451
nombre	1952
es	1058
Roberto	14600
[SEP]	5

Figura 26. Ejemplo simple de una tokenización.

Para comprobar el comportamiento del modelo que se escogió para el presente Trabajo de Titulación, se realizó una serie de preguntas en base a dos contextos (**observe la Tabla 10**) que nunca antes se habían utilizado en el entrenamiento previo, los cuales corresponden a los temas de “Requisitos de postulación” y “Formas de pago y financiamiento” que figuran en los artículos 155 y 189 del Reglamento de Régimen Académico de la UNL. El motivo por el cual se escogió estos dos temas fue debido a que los estudiantes suelen realizar preguntas frecuentes con respecto a requisitos y formas de pago (**vea la pregunta 1 Anexo 2**). Esta información se la pudo obtener mediante una encuesta que se realizó a la secretaria de posgrados Nataly Mendoza.

Tabla 10. Contextos usados en el modelo DistilBERT.
Código: identificador único del contexto. **Contexto:** información obtenida textualmente del Reglamento de Régimen Académico⁶.

Código	Contexto
Art.155	<p>Los postulantes a los programas de postgrado que ofrece la Universidad Nacional de Loja, para la formación en especialización, maestría académica con trayectoria profesional y maestría académica con trayectoria de investigación, deberán presentar lo siguiente:</p> <ul style="list-style-type: none"> • Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia. • Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior, el postulante deberá presentarlo debidamente apostillado o legalizado por vía consular, de conformidad al Art. 22 del Reglamento de Régimen Académico del Consejo de Educación Superior. • Documentos personales: cédula de ciudadanía y certificado de votación, actualizados. • Nivel de dominio de lengua diferente a la materna, al menos a nivel B2 tomando como referencia el Marco Común Europeo para lenguas, si el proyecto curricular del programa de posgrado lo establece. • Hoja de vida, donde conste: formación y experiencia profesional. • Récord académico del nivel de grado. • Carta donde argumente las razones por las que quiere estudiar y/o investigar en el programa de posgrado. • Ensayo en temas relacionados con el posgrado, para la maestría académica con trayectoria de investigación.
Art.189	<p>Para el pago de la matrícula y arancel se establecen las siguientes formas: Efectivo (directamente en tesorería), Transferencia bancaria, Tarjeta de crédito, Crédito bancario, Descuento a rol de pago (funcionarios UNL y externos con convenio), Convenio de pago directo con la UNL.</p> <p>Los costos de aranceles del programa de posgrado serán cancelados máximo hasta el tercer mes de iniciado el primer período académico ordinario, sin intereses; pasada la fecha se pagarán intereses de acuerdo a la Ley.</p>

En la **Tabla 11** se puede observar el resultado que se obtuvo al pedir al modelo que realice la inferencia en base a los contextos de la **Tabla 10**. Para cada pregunta que se realizó, DistilBERT extrae la respuesta del propio contexto y genera un valor numérico (precisión) de que tan exacta es la respuesta. La idea general de realizar esta prueba, fue para ver el comportamiento y que tan eficiente puede llegar a ser el modelo antes de llevar a cabo el entrenamiento. Como resultado final se obtuvo una precisión general de 0,2372 siendo un valor bastante bajo. La ejecución de este proceso se lo puede ver detalladamente en el **Anexo 3**.

⁶ Reglamento de Régimen Académico de la UNL: <https://bit.ly/3RS9pbR>

Tabla 11. Precisión del Modelo BERT (DistilBERT).

Contexto: código que identifica el contexto de la Tabla 10. **Preguntas:** interrogantes realizadas al modelo DistilBERT. **Respuestas:** contestación proporcionada por el modelo. **Precisión:** valor numérico que indica la exactitud de la respuesta que da el modelo.

Contexto	Preguntas	Respuesta	Precisión
Art.155	¿Qué requisitos se deben presentar para la inscripción a un posgrado?	Solicitud.	0,0080
	¿Requisitos para la inscripción?	Copia simple del título de tercer nivel o grado.	0,0215
	¿Qué requisitos debo presentar para postular a un posgrado?	Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia; Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior.	0,5537
	Dime los requisitos que debo presentar	Solicitud de admisión al programa de posgrado.	0.2013
Art.189	¿Qué formas de pago existen?	Efectivo (directamente en tesorería), Transferencia bancaria.	0,2469
	¿De qué formas puedo pagar una maestría?	Efectivo (directamente en tesorería), Transferencia bancaria, Tarjeta de crédito, Crédito bancario.	0.6169
	Formas de pago	Para el pago de la matrícula y arancel se establecen las siguientes formas.	0.0124
Promedio			0,2372

Tarea 2: Ajustar el modelo BERT mediante un entrenamiento con información del Reglamento de la UNL en lo que concierne a posgrados.

En la presente tarea se procedió a realizar el entrenamiento y ajuste del modelo que se escogió en la etapa anterior. Para ello, se utilizó la herramienta Google Colab que trabaja con el lenguaje de programación Python. Como primera instancia, fue necesario elaborar un cuestionario, que se aplicó a la secretaria de posgrados Rubiela Nataly Mendoza Patiño (**revisar el Anexo 2**), y que permitió obtener la información necesaria para crear un conjunto de datos (se siguió el formato de un JSON para el dataset, **vea la Figura 27**) con respecto a los temas de matrículas y homologaciones para los estudios de cuarto nivel.

```
"data": [
  { // 2 items
    "title": "Tipos de posgrado",
    "paragraphs": [
      {
        "context": "La Universidad Nacional de Loja ofrece actualmente dos tipos de posgrado: Maestría Académica con Trayectoria Profesional y Maestría Académica con Trayectoria de Investigación",
        "qas": [
          {
            "question": "¿Qué tipos de posgrados ofrece la UNL?",
            "answers": [
              {
                "text": "Maestría Académica con Trayectoria Profesional y Maestría Académica con Trayectoria de Investigación",
                "answer_start": 74
              }
            ]
          },
          {
            "question": "¿Cuáles son los posgrados que hay en la Universidad Nacional de Loja?",
            "answers": [
              {
                "text": "Maestría Académica con Trayectoria Profesional y Maestría Académica con Trayectoria de Investigación",
                "answer_start": 74
              }
            ]
          }
        ]
      }
    ]
  }
],
```

Figura 27. Estructura del conjunto de datos.

El conjunto de datos se dividió en dos partes: 80% para el entrenamiento (con 60 contextos y 170 preguntas) y un 20% para la validación (26 contextos y 47 preguntas), mismas que fueron almacenados en el repositorio GitHub⁷ desde donde se consumieron (vea la Figura 28). Para el entrenamiento y la validación se rescataron 4 elementos del dataset: el contexto, la pregunta, la respuesta y la posición de inicio de la respuesta. Sin embargo, también se tuvo que agregar la posición en donde termina la respuesta, para que el modelo de lenguaje aprenda hasta donde realmente debe hacerlo. Este procedimiento se lo realizó mediante código Python en Colab, tal y como se observa en la Figura 29.

Repositorio que contiene los scripts usados en el TT.

⁷ <https://github.com/Leo646/TrabajoTitulacion>

```

#Cargar los datos desde GitHub
url1 = 'https://raw.githubusercontent.com/Leo646/TrabajoTitulacion/main/Cojunto_de
_datos/Entrenamiento.json'
url2 = 'https://raw.githubusercontent.com/Leo646/TrabajoTitulacion/main/Cojunto_de
_datos/Validacion.json'
def leer_datos(ruta):
    # Leer el archivo json y almacenar los contextos, las preguntas y
    #respuestas en arreglos contextos
    resp = requests.get(ruta)
    data = json.loads(resp.text)
    contexto = []
    preguntas = []
    respuestas= []
    for group in data['data']:
        for passage in group['paragraphs']:
            context = passage['context']
            for qa in passage['qas']:
                question = qa['question']
                for answer in qa['answers']:
                    contexto.append(context)
                    preguntas.append(question)
                    respuestas.append(answer)
    return contexto, preguntas, respuestas

```

Figura 28. Lectura y almacenamiento del conjunto de datos en listas.

```

def posicion_final(respuestas, contextos):
    for respuesta, contexto in zip(respuestas, contextos):
        texto_respuesta = respuesta['text'] #se extrae la respuesta
        inicio_respuesta = respuesta['answer_start'] # se extrae la posición de inicio
        de la respuesta
        final_respuesta = inicio_respuesta+ len(texto_respuesta)# se calcula
        la posición final de la respuesta
        #En ocasiones las respuestas del conjunto de datos están mal por uno o dos
        caracteres, así que se debe arreglar eso
        if contexto[inicio_respuesta:final_respuesta ] == texto_respuesta:
            respuesta['answer_end'] = final_respuesta
        elif contexto[inicio_respuesta-1:final_respuesta 1] == texto_respuesta:
            respuesta['answer_start'] = inicio_respuesta - 1
            respuesta['answer_end'] = final_respuesta - 1
        elif contexto[inicio_respuesta-2:final_respuesta -2] == texto_respuesta:
            respuesta['answer_start'] = inicio_respuesta - 2
            respuesta['answer_end'] = final_respuesta - 2
        posicion_final(train_respuestas,train_contextos,)
        posicion_final(valid_respuestas,valid_contextos)

```

Figura 29. Cálculo de la posición final donde termina la respuesta.

El siguiente paso consistió en transformar los datos que se iban a introducir al modelo en tokens numéricos, utilizando para ello la librería BertTokenizerFast, la cual recibió como parámetros los contextos y las preguntas que se almacenaron anteriormente en arreglos.


```

from transformers import BertTokenizerFast
#se tokenizan los datos en vectores numéricos
tokenizador = BertTokenizerFast.from_pretrained('mrm8488/distill-bert-base-
spanish-wwm-cased-finetuned-spa-squad2-es',do_lower_case=False)
#se tokenizan únicamente los contextos y las preguntas
train_codificacion = tokenizador(train_contextos, train_preguntas, truncation
=True, padding=True)

```

Figura 30. Tokenización de los contextos y preguntas en vectores numéricos.

A continuación, se establecieron los parámetros de entrenamiento; en este caso se modificaron tres de ellos: el tamaño del lote (`batch_size`), el número de épocas (`N_epochs`) y la tasa de aprendizaje (`lr`); a los cuales se les dio diferentes valores para obtener un modelo que estuviera lo mejor entrenado posible.

```

#parámetros
batch_size=10
N_EPOCHS =8
lr=5e-5

```

Figura 31. Parámetros de entrenamiento.

El siguiente paso fue descargar el modelo, para ello se empleó la librería `AutoModelForQuestionAnswering`, importada de la biblioteca `Transformers`. Del mismo modo se escogió un entorno de ejecución GPU (Tesla T4 de 16 GB), para acelerar el entrenamiento en el menor tiempo posible, siendo la razón principal por la cual se eligió trabajar con Google Colab.

```

from transformers import AutoModelForQuestionAnswering
#Descarga del modelo para el entrenamiento
modelo = AutoModelForQuestionAnswering.from_pretrained('mrm8488/distill-bert-base-
spanish-wwm-cased-finetuned-spa-squad2-es')
#Detemita si hay una GPU disponible para el entrenamiento caso contrario se usa una CPU
dispositivo = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
print(f'trabajando con {dispositivo}')

```

Figura 32. Descarga del modelo y selección del entorno de ejecución.

En la última fase, se llevó a cabo el entrenamiento del modelo de lenguaje, donde se ingresó la información obtenida del conjunto de datos que pasó por el proceso de tokenización.

```

from transformers import AdamW
optimizador = AdamW(modelo.parameters(), lr)
modelo.to(dispositivo)#el modelo trabaja con GPU Tesla T4
modelo.train() #función que permite entrar en modo de entrenamiento para el modelo
for iteracion in range(N_EPOCHS):#cantidad de iteraciones
#permite visualizar el progreso del entrenamiento en cada iteración
    progreso = tqdm(train_loader, leave=True)
    for batch in progreso: #por cada iteración el modelo recibe una cantidad de datos
        optimizador.zero_grad()#permite mejorar el rendimiento
        #se pide los datos para el entrenamiento
        entrada = batch['input_ids'].to(dispositivo)
        atencion= batch['attention_mask'].to(dispositivo)
        posicion_inicial = batch['start_positions'].to(dispositivo)
        posicion_final = batch['end_positions'].to(dispositivo)
        #se envía los datos al modelo para que genere una salida
        salida = modelo(entrada, attention_mask=atencion, start_positions=posicion_inicial, end
            _positions=posicion_final)
        perdida = salida[0]#el modelo genera una pérdida por cada iteración
        perdida.backward()
        optimizador.step()
        #se imprime el progreso del entrenamiento en cada iteración incluida la pérdida
        progreso.set_description(f'Epoch {iteracion+1}')
        progreso.set_postfix(loss=perdida.item())

```

Figura 33. Entrenamiento del modelo.

El resultado de este proceso se lo puede visualizar en la **Tabla 12**, en la cual se contemplan los parámetros que permitieron realizar el entrenamiento: número de épocas, tasa de aprendizaje (lr) y el tamaño del lote; que se fueron alterando con diferentes valores para realizar 34 ejecuciones. También se aprecia una pérdida (cuanto más pequeña sea ésta más eficiente será el modelo); una precisión que se obtuvo mediante una evaluación con el 20% de datos restantes del dataset; y el tiempo medido en segundos que tardó cada entrenamiento. El resultado más alto que se obtuvo como precisión fue de 0.892 (89%) al establecer los hiperparámetros mencionados anteriormente en $9 \cdot 10^{-5}$ y 12 respectivamente (vea el **Anexo 4 Figura 42**).

Tabla 12. Resultados del entrenamiento y la evaluación.

Parámetros de entrenamiento: LR (Taza de aprendizaje), Épocas y Tamaño del lote; **Pérdida:** es una función que indica la eficiencia del aprendizaje del modelo de lenguaje. **Precisión:** valor numérico que señala la exactitud de las predicciones. **Tiempo de ejecución:** duración de cada entrenamiento.

LR	Épocas	Tamaño del lote	Pérdida	Precisión	Tiempo de ejecución (s)	Figuras
1e-5	5	4	0.403	0.821	56 s	Anexo 4: Figura 13
	7	6	0.547	0.758	77 s	Anexo 4: Figura 14
	8	8	0.547	0.765	89 s	Anexo 4: Figura 15
	10	12	0.040	0.791	100 s	Anexo 4: Figura 16
	12	10	0.013	0.816	124 s	Anexo 4: Figura 17
	12	15	0.044	0.839	107 s	Anexo 4: Figura 18
2e-5	15	15	0.078	0.723	159 s	Anexo 4: Figura 19
	6	5	0.013	0.772	66 s	Anexo 4: Figura 20
	8	7	0.393	0.821	91 s	Anexo 4: Figura 21
	9	9	0.013	0.847	98 s	Anexo 4: Figura 22
	11	13	0.010	0.730	115 s	Anexo 4: Figura 23
	13	11	0.033	0.755	136 s	Anexo 4: Figura 24
3e-5	13	16	0.051	0.817	128 s	Anexo 4: Figura 25
	15	15	0.048	0.820	152 s	Anexo 4: Figura 26
	6	10	0.050	0.795	63 s	Anexo 4: Figura 27
	7	15	0.087	0.787	75 s	Anexo 4: Figura 28
	10	11	0.006	0.785	100 s	Anexo 4: Figura 29
	12	10	0.257	0.745	121 s	Anexo 4: Figura 30
4e-5	12	15	0.037	0.803	129 s	Anexo 4: Figura 31
	13	15	0.004	0.820	139 s	Anexo 4: Figura 32
	15	15	0.034	0.803	162 s	Anexo 4: Figura 33
	6	5	0.010	0.800	72 s	Anexo 4: Figura 34
	7	10	0.148	0.762	81 s	Anexo 4: Figura 35
	8	8	0.003	0.888	98 s	Anexo 4: Figura 36
5e-5	8	13	0.068	0.784	91 s	Anexo 4: Figura 37
	10	12	0.005	0.812	113 s	Anexo 4: Figura 38
	12	12	0.037	0.835	132 s	Anexo 4: Figura 39
	15	15	0.003	0.700	161 s	Anexo 4: Figura 40
	7	9	0.005	0.763	86 s	Anexo 4: Figura 41
	9	12	0.011	0.892	102 s	Anexo 4: Figura 42
5e-5	9	15	0.056	0.722	104 s	Anexo 4: Figura 43
	10	13	0.034	0.784	114 s	Anexo 4: Figura 44
	11	20	0.012	0.800	121 s	Anexo 4: Figura 45
	15	16	0.001	0.812	171 s	Anexo 4: Figura 46

Otro aspecto que se llevó a cabo fue la creación de las historias de usuario (**ver Anexo 5**), en las cuales se describió el funcionamiento de una forma breve del agente conversacional. La estructura y componentes que tienen las historias de usuario se pueden observar en la **Figura**

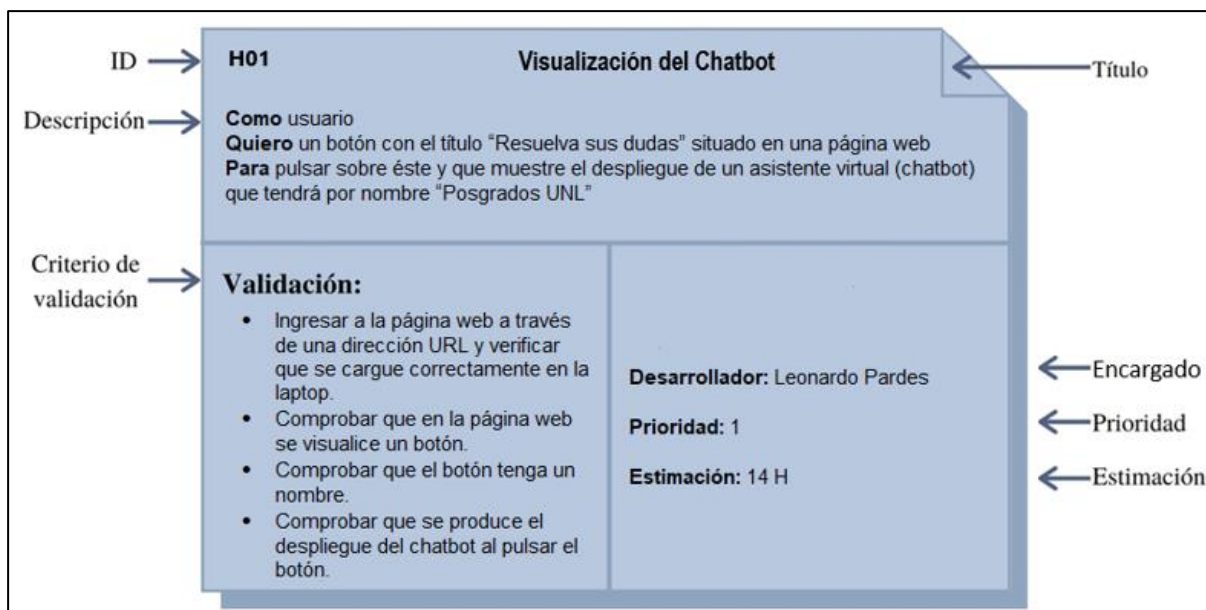


Figura 34. Ejemplo de una historia de usuario.

Los elementos más destacados que conformaron las historias de usuario fueron las siguientes:

- ID. Identificador único de la historia de usuario.
- Título. Nombre de la historia de usuario.
- Descripción. Explicación breve de la historia de usuario. Debe responder tres interrogantes ¿Cuál es el beneficiario?, ¿Qué se requiere? y ¿Cuál es el beneficio?
- Criterios. Pruebas aplicadas al sistema que debe validar el usuario.
- Desarrollador. Persona encargada de la codificación de la historia de usuario.
- Prioridad. Determina el orden en el que las historias de usuario deben ser implementadas.
- Estimación. Tiempo estimado para la elaboración del desarrollo de la historia de usuario.

Tarea 3: Diseñar la arquitectura del Agente Conversacional

En la fase de diseño se elaboró una arquitectura general del funcionamiento del agente conversacional (vea la **Figura 35**). La arquitectura muestra la interacción de todos los elementos que conforman el chatbot:

- Usuario final. Es cualquier persona que puede interactuar con la interfaz gráfica. Este puede plantear preguntas sobre los procesos de matrículas y homologaciones, y esperar que el agente conversacional le devuelva una respuesta.
- Interfaz gráfica. Entorno visual por donde el usuario puede realizar preguntas pertinentes al agente conversacional y visualizar las respuestas dadas por el mismo. La interfaz gráfica se desarrolló utilizando las siguientes tecnologías:

- HTML. Es un lenguaje de marcado de hipertexto que permitió generar y desplegar la estructura web del agente conversacional.
- JavaScript. Lenguaje de programación que se empleó para darle dinamismo al chatbot, es decir, generar efectos llamativos y al mismo tiempo rescatar las preguntas realizadas por los usuarios, para posteriormente, enviarlas al modelo neuronal.
- CSS: Es un lenguaje que se utilizó para dar cualidades visuales y estéticas al agente conversacional.
- Flask. Es un microframework que se usó para montar un servidor web con Python y ejecutar toda la aplicación.
- DigitalOcean⁸. Es una plataforma en la nube, donde se pudo desplegar la aplicación web (el agente conversacional) para que diferentes usuarios puedan tener acceso.
- SpaCy. Librería usada para descomponer las preguntas de los usuarios en términos clave y calcular la similitud con las palabras de la base de conocimientos, para extraer el contexto donde se encuentra la respuesta a una pregunta dada.
- DistilBERT. Modelo de lenguaje que se empleó para la creación del chatbot. El modelo se encargó de interpretar las preguntas del usuario y generar una respuesta en base a la misma pregunta y un contexto de donde se extrae la respuesta.
- Base de conocimientos. Información que contiene de los temas de matrículas y homologaciones sobre los posgrados.

⁸ <https://www.digitalocean.com/>

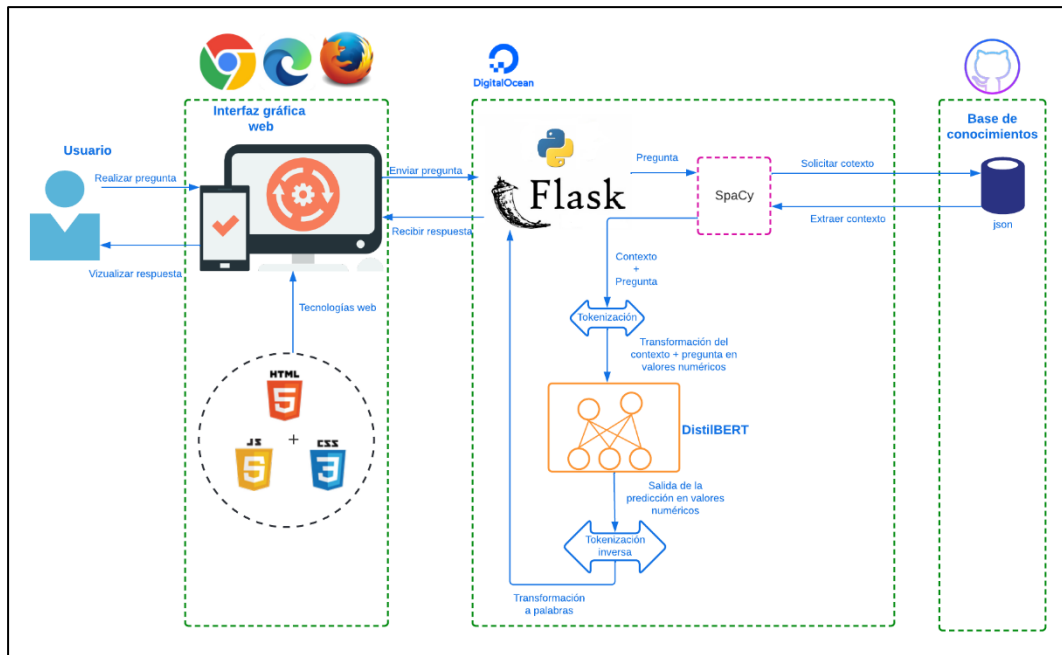


Figura 35. Arquitectura del chatbot.

En la arquitectura, el usuario genera una consulta en forma de pregunta a través de la interfaz gráfica web que se puede acceder mediante una laptop; Python junto a Flask permiten usar la librería spaCy, la cual se ocupa de descomponer la pregunta en términos y calcular la similitud que existe con las palabras que hay en la base de conocimientos y proceder a la extracción del contexto que más se asemeje a la pregunta (**ver Figura 36**). Posteriormente, el texto y la pregunta son enviados al modelo pasando previamente por un proceso de tokenización (transformar las palabras en vectores numéricos). DistilBERT se encarga de realizar la predicción en base a los parámetros que recibió como entrada y extraer una respuesta del contexto en forma numérica. Finalmente, la predicción pasa por una tokenización inversa (convierte los vectores numéricos a palabras), la cual es recibida por Python que se encarga de enviar la respuesta a la interfaz gráfica y pueda ser visualizada por el usuario.

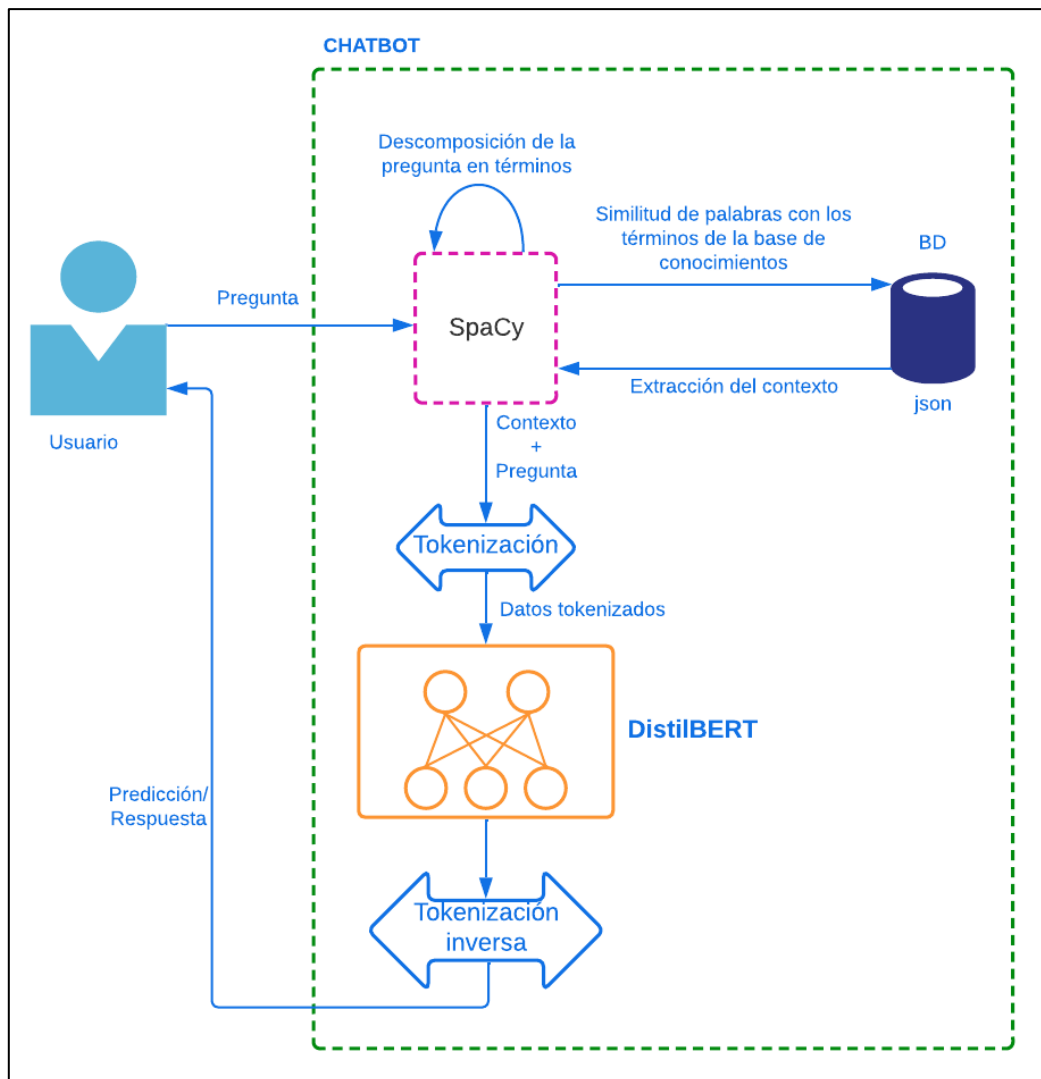


Figura 36. Funcionamiento interno del chatbot.

Tarea 4: Desarrollar un prototipo web utilizando Python y JavaScript.

En la siguiente tarea se empleó la fase de codificación de la metodología XP, en donde se utilizó el microframework Flask que permitió crear de forma sencilla y con pocas líneas de código una aplicación web funcional. El motivo principal por el que se escogió este microframework fue debido a que trabaja de la mano con el lenguaje de programación Python, lo que facilitó el desarrollo del chatbot. En la **Figura 37** se contemplan todos las carpetas y archivos que formaron parte de la aplicación. En primera instancia se programó la estructura y el diseño visual que debía tener el agente conversacional utilizando los lenguajes HTML y CSS. Las tecnologías Python y JavaScript permitieron realizar la funcionalidad del asistente virtual, para que los usuarios puedan interactuar con éste.

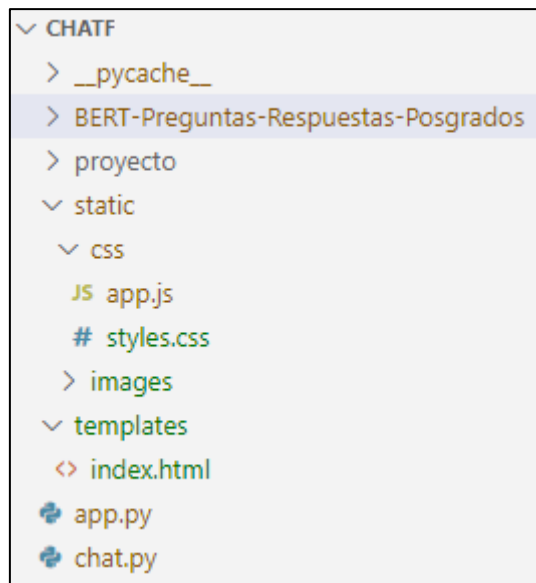


Figura 37. Estructura de la aplicación.

En el archivo “chat.py” se programó toda la funcionalidad que debe realizar el modelo DistilBERT. En este documento se usó código para cargar la red neuronal que se entrenó en la Tarea 2 del presente objetivo y que se encuentra instalada localmente. Las librerías empleadas para el proceso fueron “BertForQuestionAnswering” (para importar el modelo) y “BertTokenizerFast” (para tokenizar las entradas). Adicional a ello, se especificó un entorno CPU que permita a DistilBERT el procesamiento de los datos.

```
# Importar el modelo
modelo= 'BERT-Preguntas-Respuestas-Posgrados'
#Se carga el modelo entrenado para la predicción
model_preentrenado = BertForQuestionAnswering.from_pretrained(modelo)
#Transforma las entradas en tokens numéricos
tokenizador = BertTokenizerFast.from_pretrained(modelo)
#Establece si se trabaja con CPU o GPU
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
model_preentrenado = model_preentrenado.to(device)
```

Figura 38. Importación del modelo y entorno de ejecución.

DistilBERT recibe dos parámetros tokenizados como entrada (pregunta y contexto), que pasan por las diferentes capas del modelo hasta que en la última de ellas queden dos valores correspondientes al inicio y final de la respuesta que extrae del propio contexto. Posteriormente, se produce una tokenización inversa de la predicción, es decir, los vectores numéricos que salen de la red neuronal se convertirán en palabras en forma de oraciones, de tal forma que serán retornadas hacia la interfaz gráfica.


```

def get_prediction(contexto, pregunta):
    #se tokeniza la pregunta y el contexto
    entradas = tokenizador.encode_plus(pregunta, contexto, return_tensors='pt').to(device)
    salida = modelo_preentrenado(**entradas)
    #se calcula el inicio y final de la respuesta que el modelo predice
    inicio_respuesta = torch.argmax(salida[0])
    final_respuesta = torch.argmax(salida[1]) + 1
    #almacena el inicio-final de la respuesta que el modelo predice
    respuesta = tokenizador.convert_tokens_to_string(tokenizador.convert_ids_to_tokens
    | | | | | (entradas['input_ids'][0][inicio_respuesta:final_respuesta]))
    return respuesta

```

Figura 39. Fase de predicción de DistilBERT.

Para que el modelo realice la predicción es necesario acceder a la información de la base de conocimientos, la cual se compone de contextos, palabras clave y una dirección URL que será incorporada en algunas respuestas.

```

{
  "palabras_clave": [ "inscripción", "requisitos", "maestrías", "posgrados", "postular" ],
  "parrafo": [
    {
      "context": "Los aspirantes que desean inscribirse en un programa de posgrado deberán presentar la siguiente información. Solicitud de admisión al programa de posgrado dirigida al Decano/a de la Facultad que corresponda o al Director/a de la Unidad de Educación a Distancia, Copia simple del título de tercer nivel o de grado debidamente registrado por el SENESCYT; en caso de que el título de grado sea obtenido en el exterior presentarlo debidamente apostillado o legalizado por vía consular, Documentos personales: cédula de ciudadanía y certificado de votación actualizados, Nivel de dominio de lengua diferente a la materna al menos a nivel B2 tomando como referencia el Marco Común Europeo para lenguas si el proyecto curricular del programa de posgrado lo establece, Hoja de vida donde conste: formación y experiencia profesional, Récord académico del nivel de grado, Carta donde argumente las razones por las que quiere estudiar y/o investigar en el programa de posgrado, Ensayo en temas relacionados con el posgrado para la maestría académica con trayectoria de investigación.",
      "enlace1": ""
    }
  ]
}

```

Figura 40. Ejemplo de uno de los contextos de la base de conocimientos.

El uso de spaCy permitió extraer el contexto correcto de la base de conocimientos. Para que este proceso sea posible, la librería se encargó de calcular cierta similitud que existe entre la pregunta que el interesado formuló con las palabras de la base de conocimientos. La similitud corresponde a una cifra que se encuentra en el rango de 0 a 1; en este caso se estableció un valor mínimo de 0.7, es decir, cada vez que spaCy llegue a ese valor se encargará de extraer el texto de la BD (vea la **Figura 41**). Esta parte de la codificación se la desarrolló en el archivo “app.py”.

```

import spacy
nlp = spacy.load("es_core_news_sm")
url1 = 'https://raw.githubusercontent.com/Leo646/TrabajoTitulacion/main/Base_Conocimientos/Base_Conocimientos2'
resp = requests.get(url1)
datos = json.loads(resp.text)
#almacena la pregunta hecha por el usuario
text1= request.get_json().get("message").lower()
text=text1.strip()#permite eliminar espacios
text= re.sub("\,|\?|\|", "",text)#elimina caracteres especiales
mayor=0.7 #similitud mínima
dato=''
indice=0 #índice de cada contexto
#se recorre la base de conocimientos
for i,data in enumerate(datos['data']):
    #Se recorren las palabras clave de cada contexto
    for e,pregunta in enumerate(data['clave']):
        pre=nlp(dato)
        pre1=nlp(pregunta)
        #se calcula la similitud, la cual debe ser mayor a 0.7
        if pre.similarity(pre1)>mayor:
            indice=i #se almacena el índice del contexto que mayor similitud tuvo
            mayor=pre.similarity(pre1)

```

Figura 41. Extracción del contexto de la base de conocimientos

Sin embargo, existe un problema, los usuarios que realicen preguntas mediante la interfaz gráfica lo hacen de forma muy distinta, por lo que el modelo DistilBERT genera diferentes valores para las predicciones, de la misma manera que sucedió en la **Tabla 11** correspondiente a la actividad 1 del primer objetivo, ya que al cambiar o jugar con algunas palabras de las preguntas se obtuvieron respuestas incompletas o diferentes.

Para mitigar esto, se optó por realizar una asociación de las preguntas hechas por los usuarios con las preguntas utilizadas en el entrenamiento del modelo. Esto se lo puede ver de la siguiente manera:

- En primer lugar, se utilizó spaCy para descomponer la pregunta formulada por el usuario en palabras y comparar la similitud con los términos clave de la base de conocimientos. Si la similitud es mayor a 0.7 se extrae el índice o posición donde se encuentra el contexto.
- En segundo lugar, a partir del índice se pudo consumir el contexto de la BD, el cual se lo envió al modelo junto con una pregunta utilizada en el entrenamiento. Con este proceso se pudo asegurar que DistilBERT de una respuesta completa (**ver procedimiento en la Figura 42**).

```

elif(indice==1):
    response=get_response('{que requisitos se necesitan para la inscripción?}')
    message={"number":indice,"answer":response}
elif(indice==2):
    response=get_response('{dónde puedo registrar la matrícula de un posgrado?}')
    message={"number":indice,"answer":response}
elif(indice==3):
    response=get_response('{tipos de posgrados?}')
    message={"number":indice,"answer":response}

```

Figura 42. Envío de las preguntas al modelo de lenguaje.

También se estableció la codificación para ejecutar la aplicación web (chatbot) en algún navegador.

```

@app.get("/")
def index():
    return render_template("index3.html")
@app.post("/predict")
def predict():
    while True:
        text= request.get_json().get("message").lower()
        message={"answer":response}
        return jsonify(message)
#dirección donde se ejecuta la aplicación
if __name__ == '__main__':
    port = int(os.environ.get("PORT", 5050))
    app.run(host='0.0.0.0', port=port)

```

Figura 43. Ejecución del chatbot.

En el archivo “app.js” se estableció la funcionalidad que permite visualizar la conversación que hay entre el chatbot y el usuario, es decir, ésta permite ver de forma visual las preguntas realizadas por los interesados y las respuestas dadas por el modelo. Dentro de este archivo también se estableció la codificación que permitió recuperar las preguntas echas por el usuario a través de la interfaz gráfica.

```

onSendButton(chatbox) {
  //accede a la variable input, que es la caja de texto donde se ingresa la pregunta
  var textField = chatbox.querySelector("input");
  //extrae el valor(pregunta) ingresada por el usuario desde la caja de texto
  let text1 = textField.value
  if (text1 === "") {
    return;}
  //se define un nombre que identifique al Usuario
  let msg1 = { name: "Usuario", message: text1 }
  this.messages.push(msg1);
  //se recibe la pregunta con el método POST desde la interfaz para que se pueda visualizar
  fetch("http://127.0.0.1:5000/predict", {
    method: 'POST',
    body: JSON.stringify({ message: text1 }),
    mode: 'cors',
    headers: {
      'Content-Type': 'application/json',
    }) // recibe la respuesta que predijo el Moledo
  .then(r => r.json())
  .then(r => {
    //se define un nombre que identifique al chatbot
    let msg2 = { name: "UNL", message:r.answer };
    this.messages.push(msg2);
    //se llama al método que permite visualizar la respuesta en la interfaz dada por
    //el modelo
    this.updateChatText(chatbox)
    textField.value = ""
  }).catch((error) => {
    console.error("Error:", error);
    this.updateChatText(chatbox)
    textField.value = ""
  });
};

```

Figura 44. Archivo app.js.

Tarea 5: Documentar la aplicación

Para ver el funcionamiento se optó por instalar el agente conversacional en dos servidores; uno de ellos corresponde al server de la carrera de Sistemas/Computación de la UNL y el otro es un servicio llamado DigitalOcean que permite el alojamiento de aplicaciones en la nube, cuya característica principal es la escalabilidad, lo que se traduce, en que se pueden usar diferentes tipos de recursos a medida que el programa informático vaya creciendo, además, ofrece un servicio gratuito de 60 días para utilizar todas las prestaciones en cualquier sistema desarrollado. En la **Tabla 13** se muestran los componentes y características de los recursos del hardware que tienen ambos servidores.

Tabla 13. Recursos hardware para el chatbot.

Componentes: elementos físicos de los servidores. **Servidor de la carrera de computación:** dispositivo físico que permite el almacenamiento de recursos informáticos. **Comandos:** instrucción que permite ver las características de un sistema informático.

Componentes	Servidor de la carrera de Computación	Servidor de Digital Ocean	Comandos
RAM	7,6 GB	8 GB	free -h
CPU	Genuine Intel, 3 cpu de 4 núcleos, 2,4 GHz	Genuine Intel, 4 cpu de 4 núcleos, 2,8 GHz	lscpu
Almacenamiento	117 GB	160 GB	df -h
Tipo de disco de almacenamiento	HDD	SSD	cat /sys/block/sda/queue/rotational
SO	CentOS 7	Ubuntu 22.04	S/n

Para el alojamiento del agente conversacional fue necesario crear en la plataforma un Droplet (vea la **Figura 45**), que es un servidor virtual donde se instaló un sistema operativo basado en la distribución de Linux (se escogió Ubuntu 22.04).

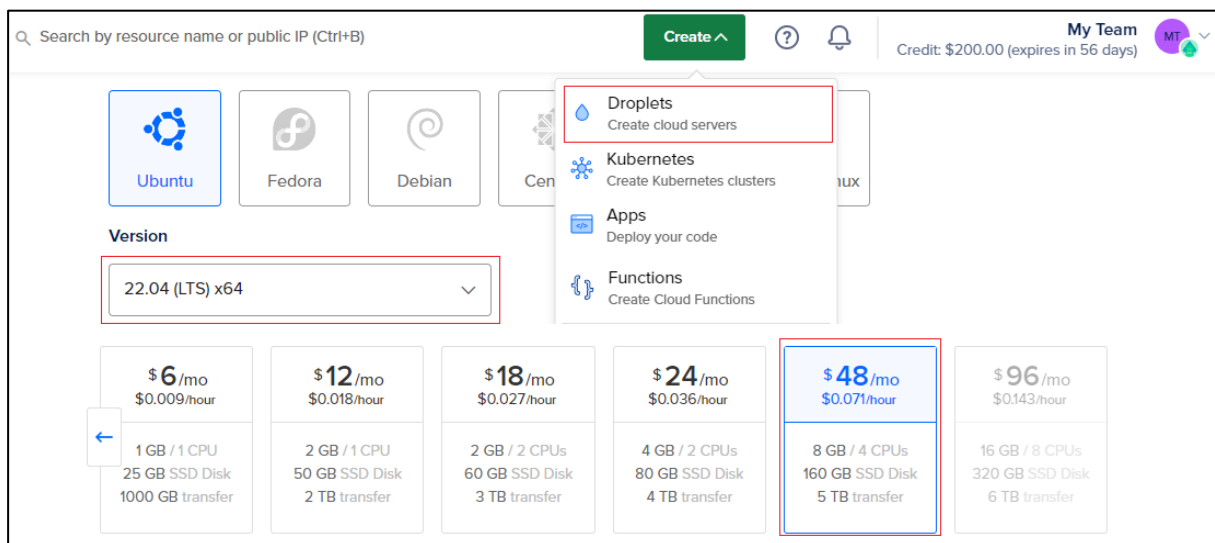


Figura 45. Creación del servidor virtual

Al momento de crear la máquina virtual se proporcionó una dirección IP que permitió conectarse de forma remota al servidor, aunque también se necesitó de una clave ssh para hacer la conexión desde la máquina de origen.

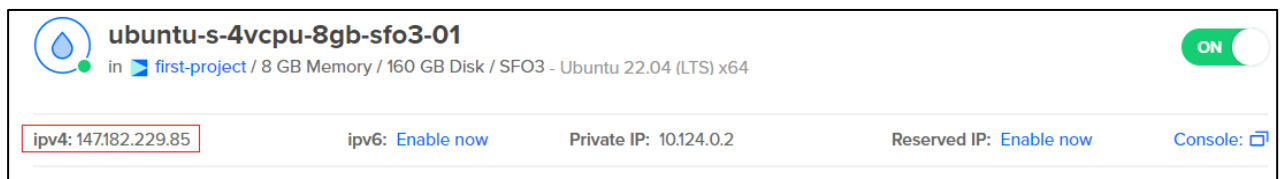


Figura 46. Dirección IP para la conexión remota.

La conexión remota se realizó con el CMD del sistema operativo Windows. Dentro del servidor virtual se creó una carpeta que contendría toda la aplicación del chatbot. Una de las desventajas de trabajar con un servidor basado en el sistema operativo Linux, es que cada operación se realiza mediante la consola del sistema, es decir, que para desarrollar una aplicación web directamente en el servidor se deben crear carpetas y archivos a través de líneas de comando; para mitigar este problema fue necesario subir todo el proyecto al repositorio GitLab⁹, desde donde se descargó los archivos que formaban parte del agente conversacional, además de instalar los módulos necesarios para su funcionamiento en la carpeta que se elaboró en el Droplet.

```

root@ubuntu-s-2vcpu-4gb-ams3-01: ~/proyecto
carpeta que contiene todo el proyecto
root@ubuntu-s-2vcpu-4gb-ams3-01:~# ls
proyecto  snap
root@ubuntu-s-2vcpu-4gb-ams3-01:~# cd proyecto
root@ubuntu-s-2vcpu-4gb-ams3-01:~/proyecto# ls
BERT-Preguntas-Respuestas-Posgrados  __pycache__  app.py  cha1.py  chatbot  static  templates
root@ubuntu-s-2vcpu-4gb-ams3-01:~/proyecto# source chatbot/bin/activate
(chatbot) root@ubuntu-s-2vcpu-4gb-ams3-01:~/proyecto# pip list
Package      Version
-----
Flask        2.1.3
huggingface-hub  0.8.1
torch        1.12.0+cpu
transformers  4.20.1
archivos que forman parte de la aplicación
módulos necesarios para el funcionamiento del chatbot

```

Figura 47. Aplicación alojada en el servidor virtual.

Finalmente se ejecutó la aplicación, especificando la dirección y puerto que permitieron la salida hacia el exterior de la red, para que los usuarios puedan hacer uso de la misma desde cualquier lugar. Nótese que el enlace <http://147.182.229.85:80/>, corresponde a la dirección IP que proporcionó la plataforma Digital Ocean. Si se quiere acceder al chatbot instalado en el servidor de la carrera de Sistemas/Computación, lo puede hacer mediante la siguiente dirección url: <https://computacion.unl.edu.ec/chatpos/>.

```

(chatbot) root@ubuntu-s-4vcpu-8gb-sfo3-01:~/Chatbot# flask run -h 0.0.0.0 -p 80
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on http://147.182.229.85:80
Press CTRL+C to quit

```

Figura 48. Ejecución de la aplicación.

Carpetas y archivos del Chatbot

⁹ <https://gitlab.com/leonardo.v.paredes/chatbot-unl>

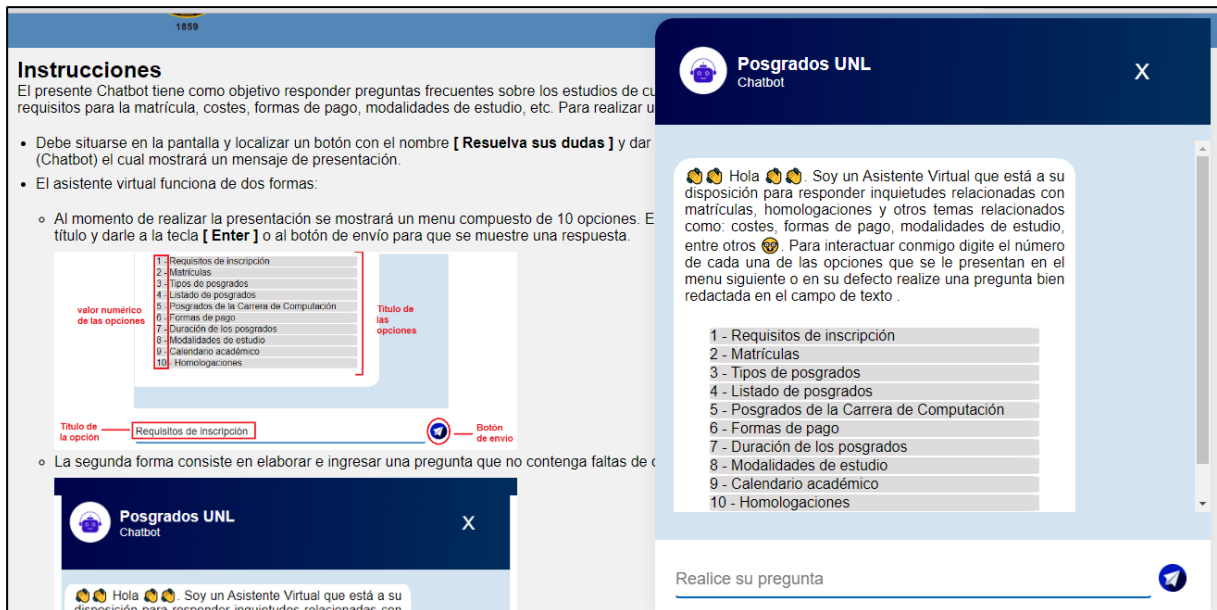


Figura 49. Chatbot en funcionamiento en la página Web: Ejemplo 1.

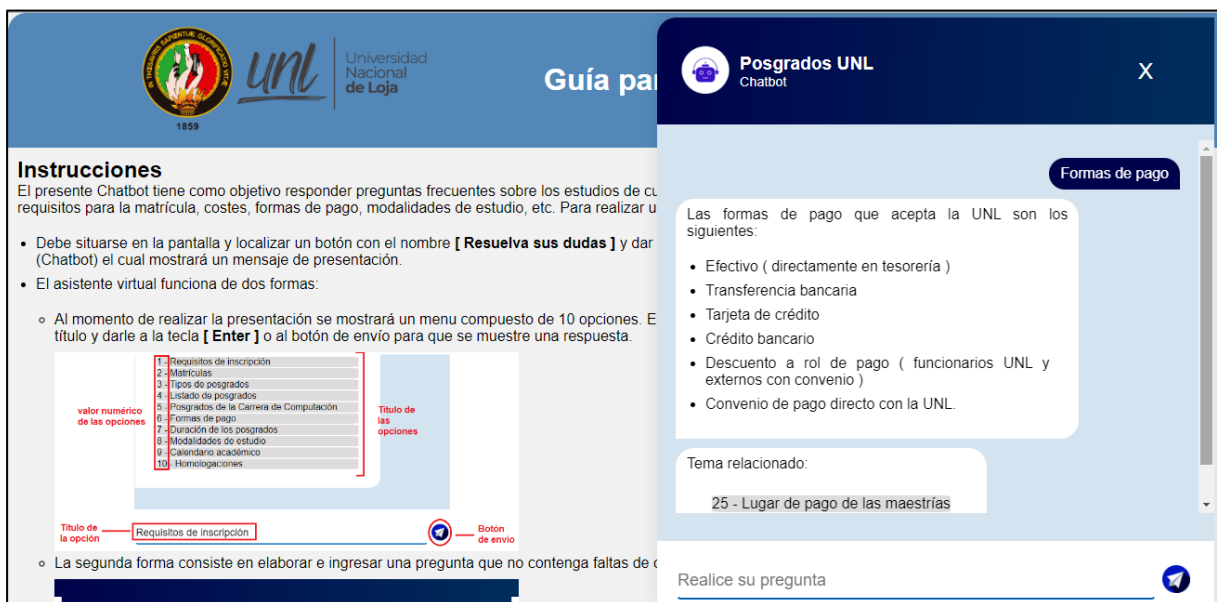


Figura 50. Chatbot en funcionamiento en la página Web: Ejemplo 2.

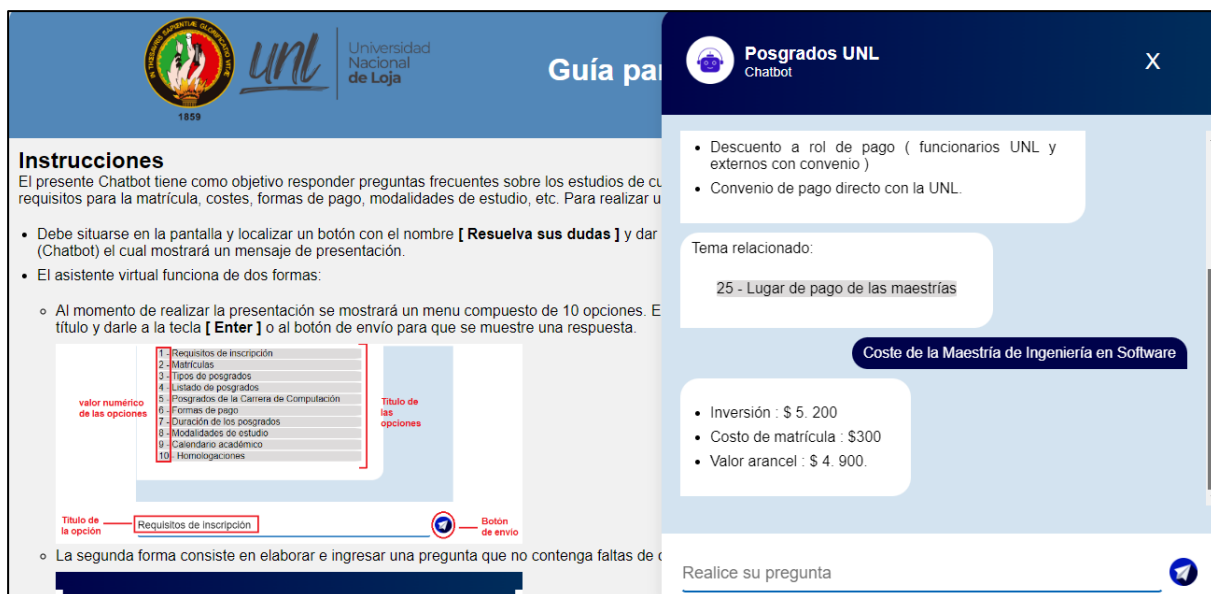


Figura 51. Chatbot en funcionamiento en la página Web: Ejemplo 3.

Objetivo 3: Evaluar el funcionamiento del Chatbot en un ambiente con estudiantes, egresados y profesionales de la carrera de Sistemas/Computación de la UNL.

Tarea 1: Realizar pruebas de validación, carga, estrés y rendimiento.

Pruebas de validación

Estas pruebas tienen la finalidad de ver si se cumplen los criterios de validación establecidas en las historias de usuario en el **Anexo 5**. Por tal motivo, se realizó una encuesta a tres personas de la Carrera de Sistemas/Computación de la UNL, 2 profesionales y un estudiante egresado de 10º ciclo. El formato que se siguió para la encuesta se presenta en la **Tabla 14**.

Tabla 14. Formato de encuesta.

Código: identificador de las historias de usuario. **Preguntas:** interrogantes que permiten comprobar los criterios de validación. **Si:** valor afirmativo de que se cumplió el criterio de validación. **No:** valor negativo de que no se cumplió el criterio de validación.

Código	Preguntas	Si	No
H01	¿Se puede acceder a la página web donde está implementado el chatbot?		
	¿En la página web se presenta un botón con el nombre "Resuelva sus dudas"?		
	¿Al pulsar sobre el botón se despliega el chatbot en la pantalla?		
	¿El chatbot se visualiza correctamente en la laptop?		
H02	¿El chatbot muestra los comandos que se deben emplear para iniciar el diálogo?		
	¿Los comandos para iniciar el diálogo funcionan correctamente?		
H03	¿El chatbot muestra un mensaje de presentación?		
	¿El chatbot menciona que temas puede dar respuesta?		
	¿El chatbot presenta un menú compuesto por dígitos y nombres sobre temas que puede responder?		
H04	¿El chatbot muestra un campo de texto para ingresar alguna consulta?		
	En el campo de texto, ¿el chatbot permite ingresar el valor numérico o el nombre de las opciones del menú?		
	¿Las opciones del menú funcionan correctamente?		
	¿El chatbot permite ingresar una pregunta de autoría propia del usuario?		
	¿El chatbot muestra una respuesta a cada consulta hecha por el usuario?		
H05	¿El chatbot presenta un mensaje cuando no encontró respuesta a una pregunta?		
	¿El chatbot muestra el uso del comando "Ayuda"?		
	¿El comando "Ayuda" funciona correctamente?		
	¿El chatbot presenta un menú de 25 opciones tras usar el comando "Ayuda"?		
	¿Cada una de las opciones del menú funcionan correctamente?		

Los resultados obtenidos (**ver Anexo 6**) demuestran que todos los criterios de validación se cumplieron, asegurando que el chatbot funciona correctamente.

Pruebas de carga, estrés y rendimiento

Para realizar las pruebas de carga, estrés y rendimiento se empleó la herramienta ApacheJMeter¹⁰, una aplicación de código abierto que trabaja de la mano con Java y que fue diseñada para medir el desempeño de las aplicaciones con respecto al rendimiento, tanto para recursos dinámicos y estáticos (objetos java, base de datos, lenguajes de programación, protocolos, entre otros). La herramienta permite simular una cantidad de usuarios (grupo de hilos), para determinar la respuesta que tiene la aplicación con una carga simultánea de

¹⁰ <https://jmeter.apache.org/>

peticiones, y de esta forma comprobar la resistencia o analizar el rendimiento general del aplicativo cuando se lo somete a diferentes tipos de carga. A continuación, en la **Figura 52** se presenta la interfaz gráfica de la herramienta

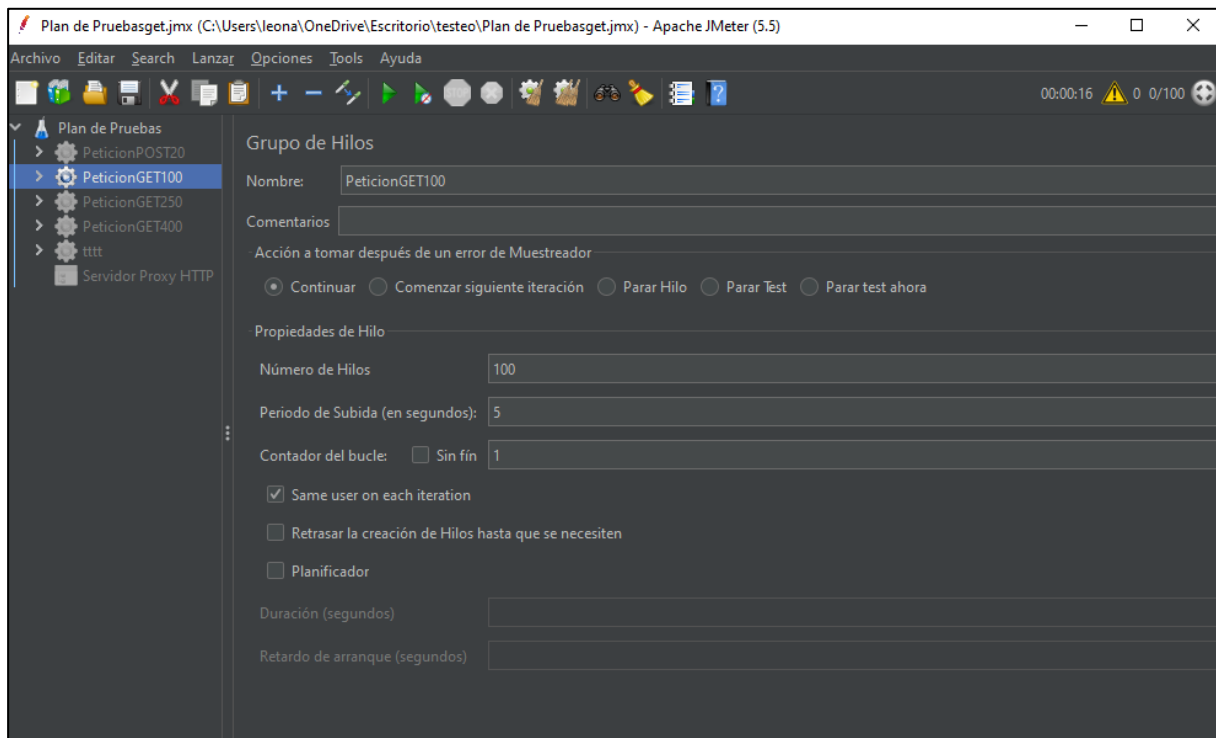


Figura 52. Interfaz gráfica de ApacheJMeter.

Ejecución de las pruebas.

En las pruebas se realizaron solicitudes HTTP utilizando los métodos GET y POST. A través del método GET se hicieron peticiones para recuperar los datos que conforman la interfaz gráfica del agente conversacional, es decir, la estructura, las imágenes y los estilos que forman parte de la página web. Con el método POST se llevó a cabo el envío de datos, representando las preguntas que los interesados hacen a través del chatbot, alojado en la plataforma DigitalOcean. A continuación, se van a presentar los resultados de ambos métodos, ejecutados en la herramienta ApacheJMeter; la evidencia de estos resultados se los puede consultar en el **Anexo 7**. Cabe mencionar que las pruebas se realizaron al chatbot alojado en la plataforma DigitalOcean, por lo que el dominio que se utilizó para el proceso es el siguiente, <http://147.182.229.85:80/>. El motivo por el que se escogió esta plataforma fue por que al momento de hacer las pruebas en el servidor de la carrera de Sistemas/Computación, en ocasiones este se encontraba desactivado.

Método GET

Para tener un promedio aproximado de la cantidad de usuarios que realizan las preguntas sobre los programas de posgrados, se decidió realizar una encuesta aplicada a la secretaria

Nataly Mendoza (ver Anexo 6) y determinar el número de peticiones atendidas diariamente. De acuerdo a los resultados existe una media general de 70 interesados que establecen preguntas sobre los estudios de cuarto nivel, específicamente en la etapa de postulación; dentro de la carrera de computación el promedio disminuye a 30 o 40 preguntas diarias. Dado que la cantidad de solicitudes es bastante baja se decidió iniciar la simulación con un mínimo de 100 usuarios.

Pruebas de Rendimiento y Carga

Mediante estas pruebas, se determinó la respuesta que tiene el chatbot al ser sometido a una carga de usuarios o peticiones de forma simultánea. Los datos que se ingresaron en ApacheJMeter se presentan en la **Tabla 15**.

Tabla 15. Método GET: ingreso de datos para las pruebas de carga.

Número de hilos	100 usuarios
Tiempo de ejecución de todos los hilos	5 segundos
Número de repeticiones	1 ciclo
Protocolo	http
Dirección de la página web	http://147.182.229.85:80/
Puerto	80
Rutas	/ /static/css/styles.css /static/css/app.js /static/images/logo.png /static/images/burbuja.png /static/images/send.png

La **Tabla 16** muestra que el chatbot soportó 600 peticiones de solicitud-respuesta a través del método GET, en un tiempo de ejecución de 5 segundos, con un rendimiento total de 87,1 segundos, indicando que fue capaz de procesar 87,1 peticiones por segundo. Además, se puede observar que la mayoría de peticiones fueron atendidas en un tiempo promedio de 224 ms, siendo el tiempo más pequeño de 15 ms y el más elevado de 4.265 ms. El porcentaje de error fue del 0%, indicando que las peticiones realizadas por los 100 usuarios de forma simultánea, se ejecutaron en su totalidad sin ningún problema.

Tabla 16. Método GET: resumen de las pruebas de carga.

Etiqueta: nombre de las rutas a las cuales se realizaron las peticiones http. **#Muestras:** cantidad de usuarios usados para la simulación. **Media:** tiempo promedio medido en milisegundos, que tardó la aplicación web en responder la mayoría de peticiones. **Min:** tiempo mínimo medido en milisegundos, que tardó el aplicativo en devolver una respuesta. **Max:** tiempo máximo expresado en milisegundos que tardó el chatbot en devolver una respuesta. **%Error:** porcentaje de peticiones que no obtuvieron respuesta. **Rendimiento:** cantidad de peticiones que pueden ser atendidas en segundos, minutos u horas.

Etiqueta	#Muestra	Media	Min	Max	%Error	Rendimiento
/	100	220	37	1369	0,00%	17,8/sec
/static/css/styles.css	100	270	16	4265	0,00%	15,7/sec
/static/css/app.js	100	197	16	3033	0,00%	14,4/sec
/static/images/logo.png	100	300	16	3087	0,00%	13,4/sec
/static/images/burbuja.png	100	187	15	3041	0,00%	13,7/sec
/static/images/send.png	100	173	17	1354	0,00%	12,1/sec
Total	600	224	15	4265	0,00%	87,1/sec

Pruebas de Rendimiento y estrés

En las pruebas de estrés, se incrementó el número de usuarios y de repeticiones (**ver Tabla 17**) para determinar los límites del agente conversacional y ver el comportamiento que tiene ante esta situación extrema.

Tabla 17. Método GET: ingreso de datos para las pruebas de estrés.

Número de hilos	200 usuarios
Tiempo de ejecución de todos los hilos	5 segundos
Número de repeticiones	4 ciclos
Protocolo	http
Dirección de la página web	http://147.182.229.85:80/
Puerto	80
Rutas	/
	/static/css/styles.css
	/static/css/app.js
	/static/images/logo.png
	/static/images/burbuja.png
	/static/images/send.png

Los resultados presentados en la **Tabla 18**, muestran que el chatbot procesó 4.800 peticiones en 4 iteraciones, con un tiempo promedio de 707 ms para la mayoría de solicitudes. El tiempo mínimo empleado para responder las peticiones fue de 14 ms y el máximo de 63.860 ms. También se tuvo un rendimiento total de 79,3 peticiones por segundo y se puede evidenciar un porcentaje mínimo de error de 0,08%, señalando que el chatbot no fue capaz de responder a ciertas peticiones, cuando éste se sometió a un estrés considerable, sin embargo, continuó funcionando sin ningún inconveniente.

Tabla 18. Método GET: resumen de las pruebas de estrés.

Etiqueta	#Muestra	Media	Min	Max	%Error	Rendimiento
/	800	473	35	15457	0,00%	11,8/sec
/static/css/styles.css	800	198	16	63860	0,13%	13,7/sec
/static/css/app.js	800	227	16	13123	0,00%	14,4/sec
/static/images/logo.png	800	172	15	56218	0,13%	13,4/sec
/static/images/burbuja.png	800	135	14	62330	0,13%	13,9/sec
/static/images/send.png	800	159	15	59825	0,13%	12,1/sec
Total	4800	707	14	63860	0,08%	79,3/sec

Método POST

Con el método POST, se pudieron realizar pruebas en donde se hicieron peticiones, simulando una conversación entre el chatbot y el usuario final, para determinar los tiempos que tardaba el agente conversacional en procesar el texto y dar una respuesta.

Pruebas de Rendimiento y Carga

En estas pruebas se emplearon los mismos datos de la **Tabla 15**, obteniendo como resultado lo que se observa en la **Tabla 19**.

Tabla 19. Método POST: resumen de las pruebas de carga.

Etiqueta	#Muestra	Media	Min	Max	%Error	Rendimiento
/predict	100	37798	15553	70346	0,00%	46,0/min
Total	100	37798	15553	70346	0,00%	46,0/min

Los resultados indicaron que el agente conversacional pudo responder las 100 peticiones realizadas por los usuarios, con un tiempo promedio de 37.798 ms. El tiempo mínimo de respuesta fue de 15.553 ms y el máximo de 70.346 ms, con un rendimiento de 46 peticiones por minuto y un porcentaje de error del 0%. Como se puede apreciar los tiempos de las peticiones son más elevados a través del método POST. Esto se debe, a que al momento de que un usuario realiza una pregunta al agente conversacional, el modelo DistilBERT consume bastante memoria y CPU para analizar un texto completo y dar una respuesta, haciendo que el tiempo se incremente, sin embargo, a pesar de todo ello, el agente fue capaz de responder a las peticiones sin ningún error, mostrando que puede manejar 100 peticiones simultáneas durante un ciclo.

Pruebas de Rendimiento y Estrés

En estas pruebas se utilizó la misma cantidad de usuarios descritos en la **Tabla 15**, sin embargo, se incrementó las iteraciones a 4 repeticiones para comprobar si el chatbot puede soportar un estrés constante.

Tabla 20. Método POST: ingreso de datos para las pruebas de estrés.

Número de hilos	100 usuarios
Tiempo de ejecución de todos los hilos	5 segundos
Número de repeticiones	4 ciclos
Protocolo	http
Dirección de la página web	http://147.182.229.85:80/
Puerto	80
Rutas	/predict

De acuerdo a los resultados obtenidos en la **Tabla 21**, el agente conversacional pudo procesar las 400 peticiones realizadas por los usuarios, en un tiempo promedio de 4.374 ms. El tiempo mínimo de respuesta fue de 12.587 ms y el máximo de 81.366 ms, con un rendimiento de 45 peticiones por minuto y un porcentaje de error del 78%. Con esta información se dedujo que la mayor parte de peticiones no obtuvieron una respuesta por parte del chatbot, pues en los resultados se refleja que existió un porcentaje de error del 78%, incluso se pudo notar que el servidor donde se encontraba alojada la aplicación se había reiniciado, ya que los recursos computacionales de los que disponía el agente conversacional no tenían tanta capacidad para procesar demasiado texto.

Tabla 21. Método POST: resumen de las pruebas de estrés.

Etiqueta	#Muestra	Media	Mín	Max	%Error	Rendimiento
/predict	400	4374	12587	81366	78,00%	45,0/min
Total	400	4374	12587	81366	78,00%	45,0/min

Tarea 2: Probar el agente conversacional

En la presente tarea, se procedió a realizar las pruebas al chatbot en un entorno real, para lo cual mediante el uso de la técnica de muestreo por conveniencia [61] se pudo seleccionar una muestra compuesta por 31 interesados (7 estudiantes de último ciclo, 13 egresados y 11 profesionales) de la Carrera de Sistemas/Computación de la Universidad Nacional de Loja, con el objetivo de determinar el funcionamiento del chatbot al momento de responder las inquietudes que tuviesen los interesados. Para llevar a cabo este proceso se realizó una encuesta de 5 preguntas obteniendo los siguientes resultados (**Ver Anexo 9**):

Tabla 22. Formato de preguntas aplicadas a los encuestados.

Código: identificador de la pregunta. **Preguntas:** interrogantes aplicadas a los interesados.

Opciones: respuesta escogida por el interesado.

Código	Preguntas	Opciones
Pregunta 1	¿Cómo califica la interfaz gráfica del agente conversacional?	Mala Regular Buena Excelente
Pregunta 2	¿Considera que el agente conversacional respondió las preguntas que usted realizó?	Ninguna Algunas La mayoría Todas
Pregunta 3	¿Cómo calificaría el tiempo que tarda el agente conversacional en proporcionarle una respuesta a sus preguntas?	Mala Regular Buena Excelente
Pregunta 4	¿Cree que la implementación del agente conversacional resultó útil para responder sus inquietudes?	Si No
Pregunta 5	¿Considera que el agente conversacional proporcionó respuestas entendibles a sus preguntas?	Ninguna Algunas La mayoría Todas

- Pregunta 1. El 83.9% de los encuestados calificaron la presentación de la interfaz gráfica del chatbot como “excelente”; el 12.9% lo puntuaron como “buena; y el 3.2% como “regular”.
- Pregunta 2. Un 83.9% de la muestra afirmó que el chatbot respondió a “todas” las preguntas planteadas; y un 16.1% manifestó que el asistente conversacional respondió “la mayoría” de las consultas.
- Pregunta 3. El 87.1% de los encuestados calificó como “Excelente” el tiempo que tardó el asistente conversacional en dar respuesta a las preguntas planteadas por el usuario, mientras que un 9.7% lo puntuó como “buena”, y tan solo un 3.2% lo calificó como “regular”.
- Pregunta 4. El 100% de los encuestados aseguraron que la implementación del agente conversacional, “sí” resultó útil para responder las inquietudes que tenían.
- Pregunta 5. Se determinó que el 90.3% de la muestra afirmó que el chatbot proporcionó a “todas” las preguntas planteadas una respuesta entendible; y tan solo un 9.1% manifestaron que “la mayoría” de sus inquietudes tuvieron respuestas comprensibles.

En la **Figura 53** se puede observar un resumen de los resultados obtenidos de la encuesta.

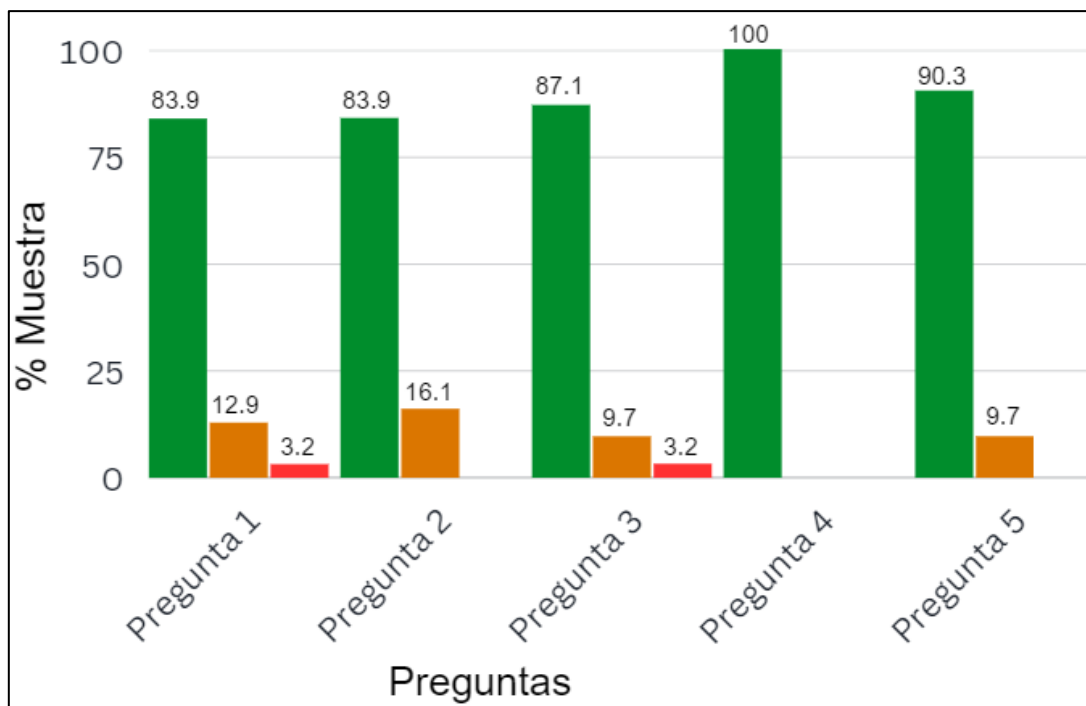


Figura 53. Resultados de la aceptación del chatbot.

Tarea 3: Corregir los errores producidos

En esta parte, se procedió a realizar las correcciones al agente conversacional que aparecieron al momento de realizar las pruebas con los estudiantes. El problema que se encontró fue la adaptabilidad del chatbot en las pantallas de los dispositivos celulares, debido a que la interfaz no se expandía correctamente por todas las dimensiones de la pantalla del teléfono móvil (**ver Figura 55, imagen izquierda**). Se tuvo que hacer modificaciones en la hoja de estilos de la aplicación web, utilizando un módulo del lenguaje CSS denominado Media Queries, que permitió establecer reglas para ejecutar bloques de código especificando ciertas condiciones para diferentes dispositivos. En este caso la regla que se estableció, fue que cada vez que la pantalla llegue a un tamaño máximo de 500px, la aplicación web debía acoplarse a toda la pantalla del dispositivo, en la **Figura 54** se puede ver el script que se usó para el proceso.


```
@media screen and (max-width: 500px) {
  .conta{
    padding: 0px;
  }
  .chatbox {
    position: absolute;
    height: 100%;
    width: 100%;
    z-index: 1;
    top:0%;
  }
  .chatbox__support {
    position: absolute;
    width: 100%;
    height: 100%;
    margin: auto;
    left: 0;
    right: 0;
  }
  .chatbox__header {
    border-top-left-radius: 0px;
  }
  .boton_cerrar{
    left: 30%;
    float: right;
  }
}
```

Figura 54. Adaptación del chatbot a dispositivos móviles.

Además, fue necesario también, ocultar el botón que permitía que el chatbot se mostrara en la pantalla, ya que resultaba un obstáculo al momento de escribir la pregunta y visualizar la respuesta, porque al usar el teclado del celular se ocultaba por completo la interfaz y no se podía observar nada. Al expandir la aplicación por toda la pantalla (**ver Figura 55, imagen derecha**) el interesado tuvo mejor comodidad para interactuar con el agente conversacional desde el dispositivo.

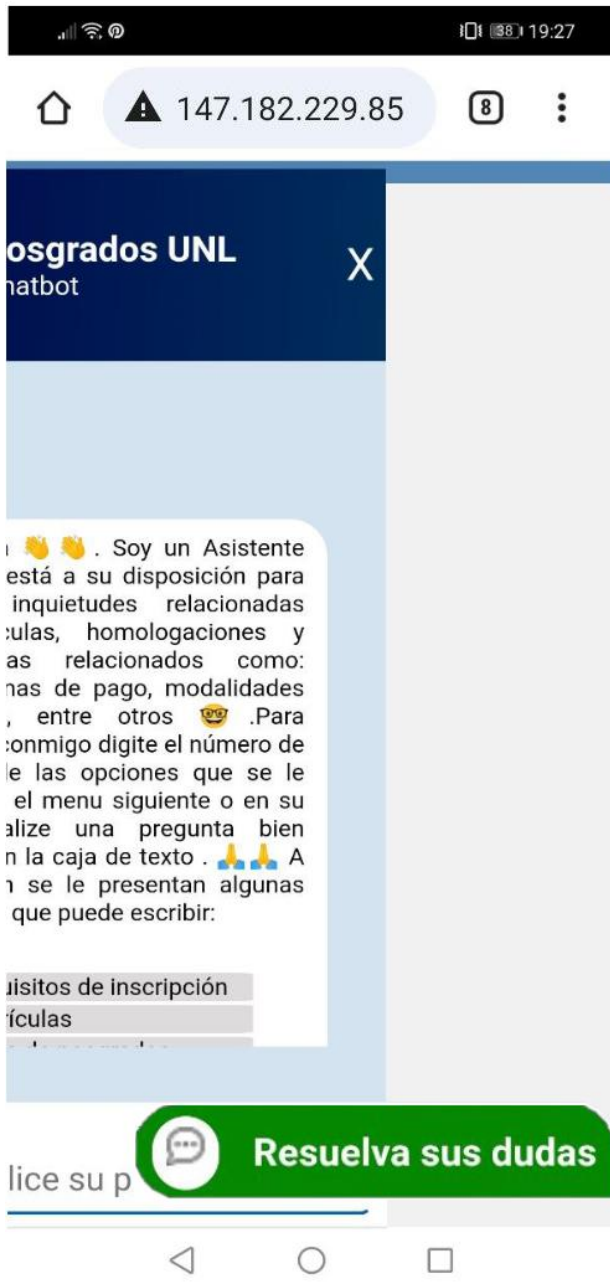


Figura 55. Interfaz del chatbot para dispositivos móviles

7. Discusión

Para llevar a cabo el desarrollo del chatbot del presente TT, se inició con un análisis del Reglamento de Régimen Académico de la UNL, abarcando la temática de posgrados en materia de matrículas y homologaciones. Posteriormente, se realizó una RSL siguiendo la metodología de Barbara Kitchenham, la cual permitió conocer los fundamentos teóricos en cuanto a la arquitectura y las tecnologías o herramientas necesarias para realizar el ajuste del modelo BERT en la tarea de preguntas y respuestas. Por último, utilizando la metodología XP en sus cuatro fases se procedió al desarrollo del agente conversacional. A continuación, se detalla los aspectos más importantes que se realizaron en este trabajo.

Objetivo 1: Definir los procesos académicos de posgrados y examinar el estado del arte sobre el modelo BERT empleado en el Procesamiento del Lenguaje Natural.

Para el cumplimiento de este objetivo, se comenzó realizando una lectura al Reglamento de Régimen Académico de la Universidad Nacional de Loja, sobre los temas de matrículas y homologaciones de los estudios de posgrados. Este análisis permitió hacer 4 diagramas de procesos de negocio siguiendo el estándar BPMN (**Tarea 2 del Objetivo 1**), donde se plasmaron con mayor detalle el funcionamiento interno de las dos temáticas mencionadas.

En segunda instancia, se realizó una revisión sistemática de literatura utilizando la metodología de Barbara Kitchenham [36], y que permitió encontrar artículos de trabajos relacionados con la información del modelo de lenguaje BERT. Esta RSL sirvió como punto de partida para el presente proyecto, ya que en los trabajos [7] y [6] se hallaron datos relevantes de modelos basados en BERT entrenados con datasets (SQuAD en la versión 1.0 o 2.0) para la tarea de responder preguntas y que se podían utilizar en este TT, además en los mismos documentos se especificaban que parámetros eran necesarios emplear durante el entrenamiento y que valores se les debía asignar a cada uno de ellos, aunque de acuerdo al artículo [39] estos valores dependen del tamaño del conjunto de datos y de los recursos tecnológicos a usar. Según los trabajos [45], [46], [48] y [49] se pueden usar tecnologías de software como Python, las librerías PyTorch o TensorFlow por su relación con el DeepLearning y la biblioteca Transformers por contener una gran variedad de conjunto de datos y modelos entrenados que se pueden usar para diversas tareas. En cuanto a los recursos de hardware, se puede emplear como mínimo una GPU de 16 GB de memoria según lo manifestado en los trabajos [53] y [55]. Otro detalle que proporcionó la RSL, fue el uso de técnicas que se podían aplicar al momento de introducir como entrada al modelo un contexto mayor a 512 tokens (según los trabajos [38] y [4] cualquier modelo BERT permite una secuencia de entrada máxima de 512 palabras). En el artículo [38] se valieron de los métodos

de truncamiento que consisten en seleccionar los primeros o últimos 512 tokens de un contexto; o quedarse con los primeros 128 y los últimos 382 tokens. Por otro lado, los estudios [57], [59] mencionaban que uno de los mejores métodos, consiste en dividir los textos en partes o categorías según las preguntas realizadas. Con esta información se tuvo las bases necesarias para la ejecución de los siguientes objetivos.

Objetivo 2: Construir una arquitectura de chatbot o agente conversacional utilizando como red neuronal el modelo de lenguaje BERT

Para el presente objetivo, se utilizaron las 3 primeras fases de la metodología XP: planeación, diseño y codificación, teniendo en cuenta algunas cuestiones. La primera consistió en escoger un modelo BERT mediante una comparación de 4 redes neuronales (**vea la Tabla 9**) destacando la versión DistilBERT; el motivo principal por el que se escogió este modelo fue por las limitaciones del hardware que proporcionaba Google Colab para el entrenamiento (1 GPU Tesla T4 de 16 GB); en el artículo [57] se menciona el uso de la herramienta. Posteriormente, a esta versión se la probó para ver la eficiencia que tenía frente a datos que no se habían usado en su entrenamiento previo. Los datos que el modelo recibió como entrada se basaron en los artículos 155 y 189 del Reglamento de Régimen Académico de la UNL (**vea la Tabla 10**), y con base a ellos se realizaron una serie de preguntas que permitieron obtener una precisión general de 0,2372 (23,72%) (**revisar la Tabla 11**). Teniendo en cuenta esto se pasó a la segunda cuestión, donde se ejecutó el entrenamiento de DistilBERT con un conjunto de datos elaborado con información obtenida de una encuesta efectuada a la secretaria Nataly Mendoza (**ver Anexo 2**); y con datos del Reglamento de Régimen Académico de la UNL sobre los posgrados. Del conjunto de datos se escogió un 80% para el entrenamiento (60 contextos y 170 preguntas) y el 20% restante para la validación (26 contextos y 47 preguntas). Se aplicó al modelo 34 entrenamientos ajustando tres parámetros con valores diferentes: el número de lotes (batch_size), el número de épocas (epoch) y la tasa de aprendizaje(lr), adicional a ello se hizo una evaluación para cada ejecución hasta obtener una precisión de 0,892 equivalente al 89,2%, estableciendo el tamaño de lote en 12, el número de épocas en 9 y la tasa de aprendizaje 5e-5 (**Consulte el Anexo 4: Figura 41**). Los resultados del proceso se los puede revisar en la **Tabla 12**. La precisión obtenida del modelo supera a los trabajos [6] (TinyBERT 71,4%, DistilBERT 69,5%) y [42] (BERT_{base} 80,4%, BERT_{large} 85,0%, ALBERT_{base} 80,0%, ALBERT_{large} 82,3% y DistilBERT 77,7%), siendo estos modelos entrenados con el dataset SQuAD 2.0. En estos dos trabajos, DistilBERT presenta precisiones diferentes debido a que cada autor estableció los parámetros de entrenamiento acorde a lo que ellos consideraban adecuados a utilizar. Cabe mencionar que en ninguno de los trabajos

relacionados se especificaba la cantidad de entrenamientos realizados a los modelos, ni tampoco como se los puede implementar en el desarrollo de un agente conversacional, pues los autores solo se dedicaron a mostrar las precisiones obtenidas.

Posteriormente se construyeron las historias de usuario (ver **Anexo 5**) en las que se describió de forma general el funcionamiento que debía tener el chatbot; también, se diseñó la arquitectura del agente conversacional, donde se especificaron todos los elementos que se iban a involucrar en el desarrollo del aplicativo y las relaciones existentes entre ellos; finalmente, se llevó a cabo la programación utilizando Python como lenguaje principal de programación, ya que este, permitió crear el código para acceder a los contextos que se almacenaron en el repositorio GitHub con la ayuda de la librería spaCy, además, permitió utilizar la biblioteca de PyTorch por su relación con el DeepLearnig; y librerías adicionales para que el modelo pueda realizar las predicciones. Con HTML se elaboró la estructura del agente conversacional y se le dió un estilo o una estética visual con el lenguaje CSS. JavaScript se lo empleó para crear efectos y responder a eventos causados por el usuario, como puede ser, el pulsar un botón cuando el interesado escriba una pregunta en el campo de texto del chatbot. Finalizada la programación se optó por buscar un servicio en la nube para poder alojar la aplicación. Inicialmente se tuvo en consideración desplegar el chatbot en la plataforma Heroku, sin embargo, debido a las limitaciones que presentaba, en cuanto a espacio de almacenamiento (512 Mb como máximo) y la incompatibilidad de algunas librerías que se estaban usando, se optó por escoger la plataforma Digital Ocean, que ofrecía mejores prestaciones y recursos en cuanto a hardware y software. En esta plataforma se creó un servidor virtual (Droplet) con las siguientes características: Memoria Ram de 8 GB, 4 CPUs, Disco SSD de 160 GB (ver **Figura 45**). En el servidor se instaló el sistema operativo Ubuntu 22.04, donde se creó una carpeta que almacenaría todo los archivos y librerías utilizadas para el desarrollo del agente conversacional, y que permitieron ejecutar la aplicación para que los interesados puedan acceder desde cualquier lugar con conexión a internet.

Objetivo 3: Evaluar el funcionamiento del chatbot en un ambiente con estudiantes, egresados y profesionales de la Carrera de Sistemas/Computación de la UNL.

En este objetivo se utilizó la última fase de la metodología XP, en la que se realizó una serie de pruebas al agente conversacional. Se inició con las pruebas de validación, aplicando una encuesta a tres interesados de la carrera de Sistemas/Computación de la UNL, que verificaron el correcto funcionamiento de los criterios de validación establecidos en las historias de

usuario que debía cumplir el chatbot (los resultados se presentan en el **Anexo 6**). Seguidamente, se ejecutaron las pruebas de rendimiento, carga y estrés empleando la herramienta ApacheJMeter, que permitió conocer las limitaciones del aplicativo en cuanto a la cantidad de usuarios que podía atender simultáneamente. Se hicieron peticiones http con los métodos GET y POST; se evidenció que el chatbot soportó un mínimo de 100 usuarios sin problemas (**ver la Tabla 16 y la Tabla 19**). Sin embargo, al incrementar la cantidad a 200, atendiendo 4.800 peticiones para el método GET, hubo un error del 0.08% (**ver la Tabla 18**), siendo un valor muy pequeño que permitió que la aplicación continuara funcionando correctamente. Con el método POST, al incrementar las repeticiones a 4 ciclos para tener un total de 400 peticiones, se generó un error del 78% (**observe la Tabla 21**), provocando que el servidor donde estaba alojada la aplicación se llegase a reiniciar. Esto no fue un inconveniente, ya que de acuerdo a una encuesta realizada a la secretaria de posgrados (**Anexo 8**), se pudo saber que el promedio diario de estudiantes que realizan preguntas con respecto a los estudios de cuarto nivel llega a un máximo de 70 interesados, cifra que no se acercaba a la cantidad mínima que el agente conversacional era capaz de atender simultáneamente. Por otro lado, en el trabajo [62] se presenta un chatbot (basado en una serie de reglas con preguntas y respuestas predefinidas) destinado a responder inquietudes sobre la materia de inteligencia artificial, donde el autor utilizó la herramienta ApacheJMeter para hacer pruebas de carga atendiendo 1.750 peticiones sin ningún inconveniente (este valor es la suma de las solicitudes realizadas al frontend y backend). De esta manera, se demostró que en este TT, el desarrollo del chatbot fue capaz de atender más cantidad de peticiones a diferencia del trabajo mencionado anteriormente.

Por último, se hicieron pruebas en un entorno con estudiantes de último ciclo, egresados y graduados profesionales de la carrera de Sistemas/Computación de la UNL (**observe la Figura 53**), a los cuales se les aplicó una encuesta sobre el grado de satisfacción al momento de usar el chatbot y que permitió responder la pregunta de investigación planteada en este TT: “¿Un chatbot implementado con el modelo de lenguaje BERT, ayudará a los estudiantes y profesionales a tener una respuesta frente a las inquietudes sobre los procesos de posgrados que se dan en la Carrera de Sistemas/Computación de la UNL?”. De acuerdo a los resultados, el 83.9% de los encuestados afirmó que el agente conversacional respondió a todas las preguntas que los interesados realizaron; el 90,3% aseguraron que las respuestas proporcionadas por el chatbot eran entendibles; y el 100% de la muestra manifestó que la implementación del asistente conversacional fue muy útil. Con estos datos se pudo observar que la acogida del chatbot para responder las preguntas más frecuentes por parte de los

interesados fue muy buena.

Por otro lado, se consideró necesario mencionar las limitaciones que existieron en este trabajo:

- En la RSL no se encontraron artículos que mencionaran el número de entrenamientos aplicados a los modelos basados en BERT para la tarea de responder preguntas, por lo que se interpretó que se ejecutó únicamente un solo entrenamiento.
- El chatbot solo podrá responder preguntas relacionadas con matrículas, homologaciones y temas relacionados como costes, tipos de posgrados, malla curricular etc; por lo que no podrá abarcar cuestiones como titulaciones, becas, entre otros.
- El agente conversacional se encuentra instalado en dos servidores (en DigitalOcean y el server de la carrera de Sistemas/Computación de la UNL) que cuentan con recursos de cómputo similares en cuanto al hardware (ver la **Tabla 13**), sin embargo, el tiempo que le toma en dar una respuesta al chatbot que está en DigitalOcean oscila entre los 3 y 7 segundos. Este tiempo se incrementa al doble al ser instalado en el servidor de la carrera de Sistemas/Computación, debido a que en éste existen otras aplicaciones usando los recursos computacionales del dispositivo.
- La base de conocimientos que usa el chatbot es estática, por lo que si se quiere aumentar la información, solo se podrá hacer mediante código Python, además, de que es necesario entrenar el modelo con dicha información para que sea más preciso.
- El chatbot no será capaz de generar una respuesta personalizada, simplemente la extrae de un contexto dado.

7.1. Valoración técnica, económica y ambiental

7.1.1. Valoración técnica.

En el presente TT, se hizo uso de una gran variedad de recursos tecnológicos (hardware, software y servicios) que emplean la inteligencia artificial, para el desarrollo de un agente conversacional usando como red neuronal, el modelo de lenguaje DistilBERT.

Tabla 23. Valoración técnica.

Recursos: elementos de software, hardware y servicios empleados para el TT.

Tecnologías: herramientas tecnológicas utilizadas en el TT. **Descripción:** definición de la utilidad de cada tecnología en el TT.

Recursos	Tecnologías	Descripción
Hardware	Laptop Accer AMD A4.	Dispositivo utilizado para la ejecución de todo el proyecto.
	GPU de Google Colab.	Unidad de procesamiento gráfico usado para el entrenamiento del modelo DistilBERT.
Software	Google Colab.	Herramienta empleada para el entrenamiento del modelo, debido a que permite usar librerías basadas en IA con código Python.
	Visual Studio Code.	IDE esencial para la escritura de código.
	Python	Lenguaje de programación empleado para la programación lógica del chatbot
	JavaScript	Lenguaje usado para dar dinamismo a la interfaz gráfica.
	Html	Lenguaje de marcado de hipertexto para crear la estructura del agente conversacional
	Css	Lenguaje que permitió dar estilos y efectos al asistente virtual.
Servicios	Flask	Microframework que se usó para crear un servicio web.
	DigitalOcean	Plataforma que permitió el alojamiento del chatbot en la nube.
	Internet	Servicio empleado para hacer consultas en diferentes documentos científicos y obtener información para el desarrollo del TT.

7.1.2. Valoración económica.

En el presenta TT, se tuvo en cuenta también el uso de recursos económicos, los cuales se presentan a continuación.

Tabla 24. Valoración económica: Talento humano.

Responsable: personas encargadas del TT. **Horas empleadas:** tiempo empleado en el TT. **Valor por hora:** sueldo para cada responsable. **Subtotal:** sueldo total para los responsables del TT.

Responsables	Horas empleadas	Valor por hora	Subtotal
Tesista	500	\$ 6,00	\$ 3.000,00
Director de tesis	90	\$ 12,00	\$ 1.080,00
Total			\$ 4.080,00

Tabla 25. Valoración económica: Hardware y software.

Recursos: tecnologías empleadas en el TT. **Tiempo empleado en meses:** duración del TT. **Costo por mes:** gastos producidos en el uso de recursos tecnológicos por mes en el TT. **Subtotal:** gastos totales por cada recurso tecnológico al finalizar el TT.

Recursos	Tiempo empleado en meses	Costo por mes	Subtotal
Python	5	\$ 0,00	\$ 0,00
Visual Studio Code	5	\$ 0,00	\$ 0,00
Google Colab	5	\$ 0,00	\$ 0,00
DigitalOcean	3	\$ 0,00	\$ 0,00
Laptop Accer AMD A4	6	\$ 0,00	\$ 0,00
Flask	5	\$ 0,00	\$ 0,00
Google Drive	3	\$ 0,00	\$ 0,00
Repositorio GitLab	4	\$ 0,00	\$ 0,00
Repositorio GitHub	4	\$ 0,00	\$ 0,00
Internet	6	\$ 40,00	\$ 240,00
Total			\$ 240,00

Tabla 26. Presupuesto final.

Recursos económicos	Subtotal
Talento humano	\$ 4.080,00
Hardware y software	\$ 240,00
Subtotal	\$ 4.320,00
IVA (+10% del subtotal)	\$ 432,90
Total	\$ 4.752,90

7.1.3. Valoración ambiental.

Durante todo el proceso del TT, se emplearon únicamente recursos digitales que no afectaron al medioambiente, reduciendo significativamente el uso de materiales físicos como el papel y otros elementos que puedan producir contaminación.

8. Conclusiones

- La Revisión Sistemática de Literatura permitió obtener 30 artículos de trabajos relacionados, donde se pudo conocer el uso de tecnologías como Python, PyTorch y Transformers, que se podían utilizar para el entrenamiento de modelos basados en BERT, además de los recursos computacionales mínimos necesarios para dicho proceso (1 GPU de 16 GB), específicamente para redes modelos livianos como DistilBERT o TinyBERT.
- La herramienta ApacheJMeter permitió conocer las limitaciones del chatbot en cuanto a la cantidad de solicitudes que puede atender simultáneamente. Como mínimo fue capaz de asistir 100 peticiones concurrentemente con los métodos GET y POST, puesto que el promedio de preguntas que suelen realizar los interesados diariamente sobre los posgrados no es mayor a 70 (según el **Anexo 8**), con la herramienta se pudo demostrar que el agente conversacional fue apto para manejar dicha cantidad.
- El uso de DistilBERT resultó muy eficiente para el desarrollo del chatbot, ya que al tratarse de un modelo pre-entrenado, solo se necesitó realizar algunos ajustes en cuanto a los parámetros de entrenamiento (número de épocas, tasa de aprendizaje y tamaño del lote), para realizar 34 ejecuciones y en base a ello obtener un modelo con una precisión del 89,2%.
- De acuerdo a los resultados que se obtuvieron en el **Anexo 9**, se pudo demostrar que el chatbot desarrollado en el presente Trabajo de Titulación, ayuda o permite a estudiantes de último ciclo, egresados y profesionales de la Carrera de Sistemas/Computación de la UNL, a solventar la problemática de obtener información de manera rápida y sin necesidad de acudir a la institución, sobre los temas de matrículas y homologaciones para los estudios de posgrados, además, estos datos permitieron responder de forma afirmativa la pregunta de investigación planteada en el proyecto.

9. Recomendaciones

- Se recomienda, que para cualquier tipo de proyecto, es necesario hacer una búsqueda de información en artículos científicos utilizando un método investigativo flexible para obtener los conocimientos teóricos necesarios de acuerdo al objeto de estudio, y que ayuden a culminar el trabajo sin ningún tipo de inconvenientes.
- Al desarrollar un chatbot o aplicación web, se puede usar otro framework diferente a Flask como puede ser Django o Laravel, ya que estos poseen bastante documentación sobre el uso de librerías de terceros que pueden facilitar el desarrollo de una página web robusta.
- Usar versiones ligeras de modelos pre-entrenados para desarrollar un sistema en la tarea de preguntas y respuestas, o cualquier otro tipo de tarea en el Procesamiento del Lenguaje Natural, ya que estas versiones permiten emplear recursos computacionales tanto en hardware como en software que estén al alcance para poner en producción una aplicación.

9.1. Trabajos Futuros

- Realizar una comparativa de diferentes modelos de lenguajes basados en BERT que utilicen bajos recursos tanto en hardware como software, de forma que se puedan poner en producción para cualquier tipo de plataforma.
- Desarrollar una interfaz gráfica para que un administrativo pueda realizar modificaciones en la base de conocimientos con otros temas académicos.
 - Dado que el ajuste del modelo que se escogió para este TT se lo entrenó con un pequeño conjunto de datos en las temáticas de matrículas y homologaciones, en un futuro se lo puede volver a entrenar con más información tomada del Reglamento de Régimen Académico de la UNL, para que el modelo tenga mejor precisión en otras temáticas académicas.
 - Desplegar el chatbot en las plataformas de mensajería que más se utilicen actualmente como Telegram, WhatsApp o Facebook.
 - Incorporar al chatbot un sistema de reconocimiento de voz, que permita la conversión de las preguntas realizadas por los interesados mediante audio a formato de texto.

10. Bibliografía

- [1] D. Bailey and N. Almusharraf, “Investigating the Effect of Chatbot-to-User Questions and Directives on Student Participation,” *2021 1st Int. Conf. Artif. Intell. Data Anal. CAIDA 2021*, pp. 85–90, 2021, doi: 10.1109/CAIDA51941.2021.9425208.
- [2] S. Saraswat, S. Mishra, V. Mani, and S. Priya, “GALGOBOT - The college companion chatbot,” *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, no. Iccics, pp. 1374–1378, 2021, doi: 10.1109/ICICCS51141.2021.9432101.
- [3] L. Lee, Y. Fung, Y. Pun, K. Wong, M. Yu, and N. Wu, “Using a Multiplatform Chatbot as an Online Tutor in a University Course,” *Proc. - 2020 Int. Symp. Educ. Technol. ISET 2020*, pp. 53–56, 2020, doi: 10.1109/ISET49818.2020.00021.
- [4] M. Koroteev, “BERT: A Review of Applications in Natural Language Processing and Understanding,” 2021, [Online]. Available: <http://arxiv.org/abs/2103.11943>.
- [5] J. Devlin, W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Assoc. Comput. Linguist.*, vol. 1, pp. 4171–4186, 2019, [Online]. Available: <https://doi.org/10.48550/arXiv.1810.04805>.
- [6] X. Jiao *et al.*, “TinyBERT: Distilling BERT for natural language understanding,” *Find. Assoc. Comput. Linguist. Find. ACL EMNLP 2020*, pp. 4163–4174, 2020, doi: 10.18653/v1/2020.findings-emnlp.372.
- [7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” pp. 2–6, 2019, [Online]. Available: <https://doi.org/10.48550/arXiv.1910.01108>.
- [8] F. Jakob, R. Bernd, and H. Bernhard, *BPMN 2.0 Manual de Referencia y Guía Práctica*, vol. 1, no. 69. 2014.
- [9] C. Caiafa and S. Lew, “¿Qué es la Inteligencia Artificial?,” *IAR:Boletín Radio@stronómico*, vol. 69, pp. 1–7, 2020.
- [10] J. Wei, X. Chu, and X. Sun, “Machine learning in materials science,” *InfoMat*, vol. 1, no. 3, pp. 338–358, 2019, doi: 10.1002/inf2.12028.
- [11] J. Baby, F. Khan, and J. Swathi, “Home automation using IoT and a chatbot using natural language processing,” *2017 Innov. Power Adv. Comput. Technol. i-PACT 2017*, pp. 1–6, 2017, doi: 10.1109/IPACT.2017.8245185.
- [12] A. Jiao, “An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network,” *J. Phys. Conf. Ser.*, vol. 1487, no. 1, 2020, doi: 10.1088/1742-

6596/1487/1/012014.

- [13] A. Vásquez, H. Huerta, J. Quispe, and M. Huayna, “Procesamiento de lenguaje natural,” *Rev. Ing. Sist. e Informática*, vol. 6, no. 2, pp. 45–54, 2009.
- [14] INBOUNDCYCLE, “Análisis profundo de Google BERT para responsables de marketing,” *academia.edu*, 2020.
- [15] R. Lucca, “¿Qué es BERT y cómo funciona?,” *DX Media*, 2020. <https://dxmedia.net/algoritmo-bert-google/>.
- [16] A. Vaswani, N. Shazeer, and N. Parmar, “Attention Is All You Need,” *Curran Assoc. Inc.*, vol. 30, pp. 1–15, 2017, doi: 10.48550/ARXIV.1706.03762.
- [17] M. Mascarell, “Clasificación de Textos Basado en los Modelos Pre-entrenados BERT,” Universidad Politecnica de Valencia, 2021.
- [18] C. González, “Procesamiento del lenguaje natural con BERT : Análisis de sentimientos en tuits,” Universidad Carlos III de Madrid, 2020.
- [19] C. González, “Estudio del rendimiento de BERT frente a métodos clásicos de procesamiento de lenguaje natural,” Universidad Autónoma de Madrid, 2020.
- [20] J. Alammari, “The Illustrated Transformer,” 2018. <https://jalammar.github.io/illustrated-transformer/>.
- [21] M. Sotaquirá, “Redes Transformer o el fin de las Redes Recurrentes,” 2020. <https://www.codificandobits.com/blog/redes-transformer/#el-codificador-de-posición>.
- [22] C. Bucila, R. Caruana, and A. Niculescu, “Model compression,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 2006, no. May, pp. 535–541, 2006, doi: 10.1145/1150402.1150464.
- [23] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” pp. 1–9, 2015, [Online]. Available: <http://arxiv.org/abs/1503.02531>.
- [24] L. Wang and K. Yoon, “Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, 2022, doi: 10.1109/TPAMI.2021.3055564.
- [25] J. Montantes, “Getting Started with Hugging Face Transformers for NLP,” *Becoming Human: Artificial Intelligence Magazine*, 2020. <https://becominghuman.ai/getting-started-with-hugging-face-transformers-for-nlp-36bf2d5a2805>.
- [26] E. Adamopoulou and L. Moussiades, “An Overview of Chatbot Technology,” *IFIP Adv. Inf. Commun. Technol.*, vol. 584 IFIP, pp. 373–383, 2020, doi: 10.1007/978-3-030-49186-4_31/FIGURES/3.
- [27] K. Nimavat and T. Champaneria, “Chatbots: An Overview Types, Architecture, Tools

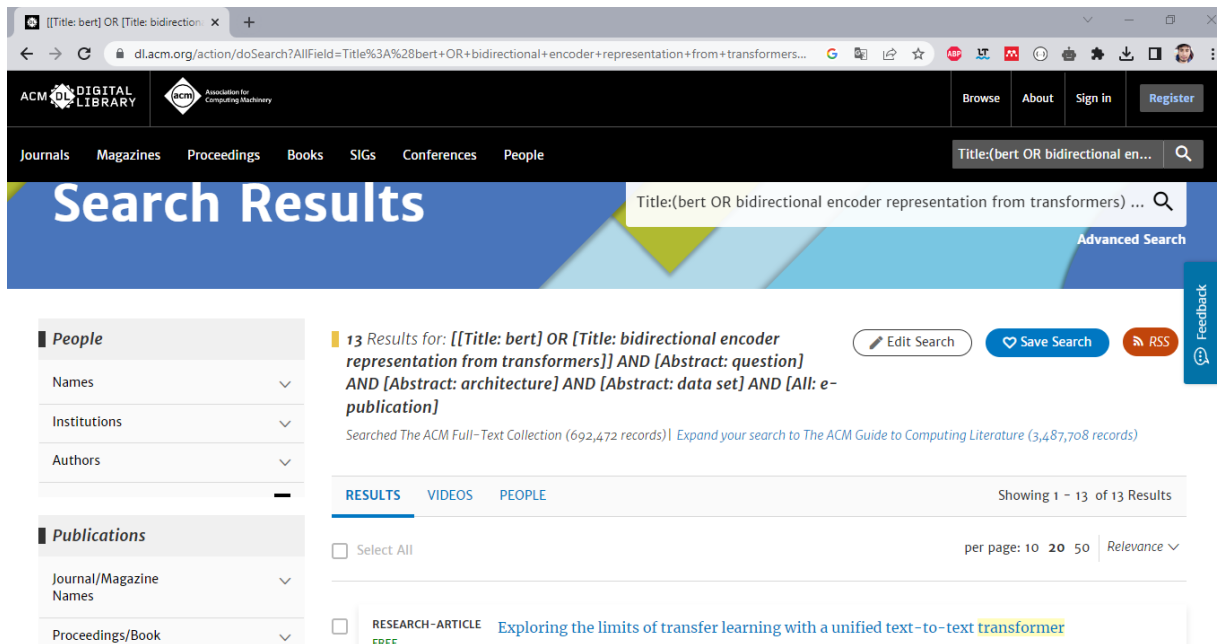
- and Future Possibilities,” *Int. J. Sci. Res. Dev.*, vol. 5, no. 7, pp. 1019–1026, 2017.
- [28] K. Ramesh, S. Ravishankaran, A. Joshi, and K. Chandrasekaran, “A Survey of Design Techniques for Conversational Agents,” *Commun. Comput. Inf. Sci.*, vol. 750, pp. 336–350, 2017, doi: 10.1007/978-981-10-6544-6_31.
- [29] T. Hien, P. Cuong, L. Nam, H. Nhung, and L. Thang, “Intelligent assistants in higher-education environments: The FIT-EBOt, a chatbot for administrative and learning support,” *ACM Int. Conf. Proceeding Ser.*, no. January 2020, pp. 69–76, 2018, doi: 10.1145/3287921.3287937.
- [30] V. Elvis, “Implementación de un Chatbot en Lenguaje Natural, Utilizando Técnicas de Inteligencia Artificial y Aprendizaje Profundo para el Aprendizaje de las Tecnologías de la Información.,” Universidad De Las Américas, 2020.
- [31] M. Corporation, “JavaScript,” 2021. <https://developer.mozilla.org/es/docs/Learn/JavaScript>.
- [32] D. Gauchat, *El gran libro de HTML, CSS y JavaScript*. 2012.
- [33] R. González, *Python Para Todos*, Creative C. 2011.
- [34] D. Ghimire, “Comparative study on Python web frameworks: Flask and Django,” Metropolia University of Applied Sciences Bachelor of Engineering Media Engineering Bachelor’s, 2020.
- [35] M. Attla, “PyTorch: saber todo sobre el marco de trabajo de Deep Learning de Facebook,” *DataScientest*, 2021. .
- [36] B. Kitchenham and C. Sturart, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” *Keele Univ. Durham Univ. Jt. Rep.*, 2007.
- [37] The Institute Electrical And Engineers Electronics, “2022 IEEE Thesaurus,” 2022.
- [38] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to Fine-Tune BERT for Text Classification?,” *CoRR*, vol. abs/1905.0, no. 2, pp. 194–206, 2019, doi: 10.1007/978-3-030-32381-3_16.
- [39] R. Laskar, E. Hoque, and J. Xiangji, “Utilizing Bidirectional Encoder Representations from Transformers for Answer Selection,” *Springer Proc. Math. Stat.*, vol. 343, pp. 693–703, 2021, doi: 10.1007/978-3-030-63591-6_63.
- [40] G. Hirst, J. Lin, R. Nogueira, and A. Yates, “Pretrained Transformers for Text Ranking: BERT and Beyond,” *Synth. Lect. Hum. Lang. Technol.*, vol. 14, no. 4, pp. 1–325, 2021, doi: 10.2200/S01123ED1V01Y202108HLT053.
- [41] N. Trang and M. Shcherbakov, “Vietnamese Question Answering System from Multilingual BERT Models to Monolingual BERT Model,” *Proc. 2020 9th Int. Conf.*

- Syst. Model. Adv. Res. Trends, SMART 2020*, pp. 201–205, 2020, doi: 10.1109/SMART50582.2020.9337155.
- [42] L. Xiao, D. Haoran, T. Matthew, G. Priyanka, Z. Zhan, and H. Zhe, “Attention Mechanism with BERT for Content Annotation and Categorization of Pregnancy-Related Questions on a Community Q&A Site,” *Physiol. Behav.*, vol. 32, no. 1, pp. 41–52, 2020, doi: 10.1109/bibm49941.2020.9313379.Attention.
- [43] S. Bhattacharjee and R. Haque, *Investigating Query Expansion and Coreference Resolution in Question Answering on BERT Santanu*, vol. 2. Springer International Publishing, 2020.
- [44] Y. Cai and D. Wang, “Question Mark Prediction by Bert,” pp. 18–21, 2019.
- [45] M. Zaib, D. Tran, S. Sagar, A. Mahmood, W. Zhang, and Q. Sheng, “BERT-CoQAC: BERT-Based Conversational Question Answering in Context,” *Commun. Comput. Inf. Sci.*, vol. 1362, pp. 47–57, 2021, doi: 10.1007/978-981-16-0010-4_5.
- [46] D. Vucetic, M. Tayaranian, M. Ziaeeafard, J. Clark, B. Meyer, and W. Gross, “Efficient Fine-Tuning of BERT Models on the Edge,” 2022, [Online]. Available: <http://arxiv.org/abs/2205.01541>.
- [47] C. Qu, L. Yang, M. Qiu, B. Croft, Y. Zhang, and M. Iyyer, “BERT with history answer embedding for conversational question answering,” *SIGIR 2019 - Proc. 42nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 1133–1136, 2019, doi: 10.1145/3331184.3331341.
- [48] P. Zhu and C. Hauff, *Evaluating BERT-based Rewards for Question Generation with Reinforcement Learning*, vol. 1, no. 1. Association for Computing Machinery, 2021.
- [49] R. Bagwe and K. George, “Automatic Numerical Question Answering on Table using BERT-GNN,” *2020 11th IEEE Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2020*, pp. 118–124, 2020, doi: 10.1109/UEMCON51285.2020.9298028.
- [50] Y. Chan and Y. Fan, “A Recurrent BERT-based Model for Question Generation,” *MRQA@EMNLP 2019 - Proc. 2nd Work. Mach. Read. Quest. Answering*, pp. 154–162, 2019, doi: 10.18653/v1/d19-5821.
- [51] A. Al-Besher, K. Kumar, M. Sangeetha, and T. Butsa, “BERT for Conversational Question Answering Systems Using Semantic Similarity Estimation,” *Comput. Mater. Contin.*, vol. 70, no. 3, pp. 4763–4780, 2022, doi: 10.32604/cmc.2022.021033.
- [52] Y. Chan and Y. Fan, “BERT for Question Generation,” *INLG 2019 - 12th Int. Conf. Nat. Lang. Gener. Proc. Conf.*, vol. 04, no. Bleu 4, pp. 173–177, 2019, doi:

- 10.18653/v1/w19-8624.
- [53] Y. He, Z. Zhu, Y. Zhang, Q. Chen, and J. Caverlee, “Infusing Disease Knowledge Into BERT for Health Question Answering, Medical Inference and Disease Name Recognition,” *EMNLP 2020 - 2020 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 4604–4614, 2020, doi: 10.18653/v1/2020.emnlp-main.372.
- [54] C. Sung, T. Ma, T. Dhamecha, V. Reddy, S. Saha, and R. Arora, “Pre-training BERT on Domain Resources for Short Answer Grading,” *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 6071–6075, 2019, doi: 10.18653/v1/d19-1628.
- [55] I. Annamoradnejad, M. Fazli, and J. Habibi, “Predicting Subjective Features from Questions on QA Websites Using BERT,” *2020 6th Int. Conf. Web Res. ICWR 2020*, pp. 240–244, 2020, doi: 10.1109/ICWR49608.2020.9122318.
- [56] N. Zhang and Y. Xing, “Questions and Answers on Legal Texts Based on BERT-BiGRU,” *J. Phys. Conf. Ser.*, vol. 1828, no. 1, 2021, doi: 10.1088/1742-6596/1828/1/012035.
- [57] C. Nikas, P. Fafalios, and Y. Tzitzikas, “Two-stage Semantic Answer Type Prediction for Question Answering Using BERT and Class-specificity Rewarding,” *CEUR Workshop Proc.*, vol. 2774, pp. 19–28, 2020.
- [58] E. Amer, A. Hazem, O. Farouk, A. Louca, Y. Mohamed, and M. Ashraf, “A Proposed Chatbot Framework for COVID-19,” *2021 Int. Mobile, Intelligent, Ubiquitous Comput. Conf. MIUCC 2021*, no. May, pp. 263–268, 2021, doi: 10.1109/MIUCC52538.2021.9447652.
- [59] C. Chau, T. Nguyen, and L. Nguyen, “VNLawBERT: A Vietnamese Legal Answer Selection Approach Using BERT Language Model,” *Proc. - 2020 7th NAFOSTED Conf. Inf. Comput. Sci. NICS 2020*, pp. 298–301, 2020, doi: 10.1109/NICS51282.2020.9335906.
- [60] M. Mahmud, S. Schaffer, and T. Scheffler, “Factoid and Open-ended Question Answering With BERT in the Museum Domain,” *CEUR Workshop Proc.*, vol. 2836, pp. 1–15, 2021.
- [61] T. Otzen and C. Manterola, “Técnicas de Muestreo sobre una Población a Estudio,” *Int. J. Morphol.*, vol. 35, no. 1, pp. 227–232, 2017, doi: 10.4067/S0717-95022017000100037.
- [62] M. M. Manuel, “Chatbot para resolver dudas frecuentes de los estudiantes referentes a una materia,” 2020.

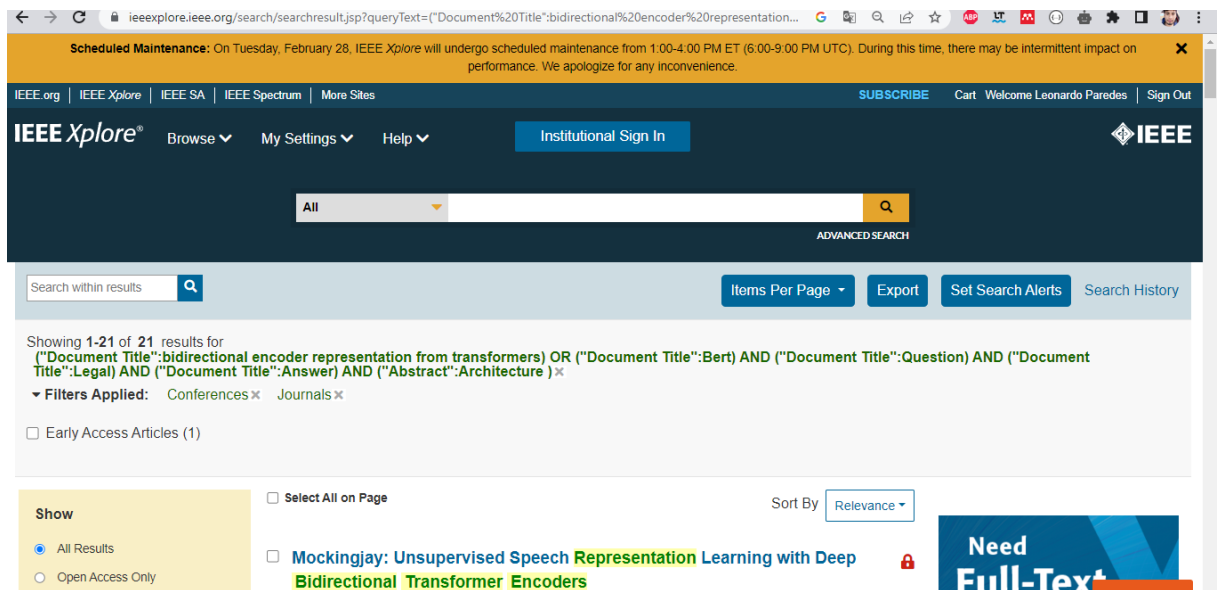
11. Anexos

Anexo 1. Ejecución de las cadenas de búsqueda.



The screenshot shows the ACM Digital Library search results page. The search query is: `Title:(bert OR bidirectional encoder representation from transformers) AND (Abstract: question) AND (Abstract: architecture) AND (Abstract: data set) AND [All: e-publication]`. The results show 13 items. The first result is a research article titled "Exploring the limits of transfer learning with a unified text-to-text transformer". The page includes navigation links for Journals, Magazines, Proceedings, Books, SIGs, Conferences, and People. There are also buttons for "Edit Search", "Save Search", and "RSS".

Figura 1. Resultados de la búsqueda en la base de datos ACM.



The screenshot shows the IEEE Xplore search results page. The search query is: `("Document Title":bidirectional encoder representation from transformers) OR ("Document Title":Bert) AND ("Document Title":Question) AND ("Document Title":Legal) AND ("Document Title":Answer) AND ("Abstract":Architecture) *`. The results show 21 items. The first result is a research article titled "Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders". The page includes navigation links for IEEE.org, IEEE Xplore, IEEE SA, IEEE Spectrum, and More Sites. There are also buttons for "SUBSCRIBE", "Cart", "Welcome Leonardo Paredes", and "Sign Out".

Figura 2. Resultados de la búsqueda en la base de datos IEEE Xplore.

120 document results

(TITLE ("bert" OR "Bidirectional Encoder Representation from Transformers") AND TITLE ("question" OR "answer" OR "tols" OR "technologies")) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019)) AND (LIMIT-TO (LANGUAGE, "English"))

Edit Save Set alert

BibTeX file exported. See your downloaded file for more details.

Search within results...



Documents Secondary documents Patents

View Mendeley Data (399)

Refine results

Analyze search results

Show all abstracts Sort on: Date (newest)

Figura 3. Resultados de la búsqueda en la base de datos Scopus.

Figura 4. Inserción de términos en la base de datos arXiv.

Figura 5. Resultados de la búsqueda en la base de datos arXiv.

Anexo 2. Encuesta realizada a la secretaria de posgrados.

Encuesta para conocer la información más relevante que los interesados preguntan sobre los programas de posgrado que oferta la Universidad Nacional de Loja

La presente encuesta tiene como objetivo recolectar información relevante sobre los programas de posgrado que oferta la Universidad Nacional de Loja. Dicha información es de vital importancia para justificar los requisitos necesarios en la elaboración de un chatbot que permita responder preguntas frecuentes que realizan estudiantes y profesionales con respecto a los estudios de cuarto nivel.

Por favor, contestar las siguientes preguntas de forma honesta. De antemano agradezco su colaboración.

Este formulario recoge automáticamente los correos de todos los encuestados. [Cambiar configuración](#)

Correo electrónico

rnmendezap@unl.edu.ec

Ayúdeme con los nombres y apellidos completos

1 respuesta

Rubiela Nataly Mendoza Patiño

Especifique el área o departamento en el que labora y el cargo que ocupa

1 respuesta

Dirección de Posgrado, Asistente de apoyo académico y administrativo

¿Cuáles son las preguntas más frecuentes que realizan los interesados con respecto a los programas de posgrado de la UNL? (Por favor elabore un listado).

1 respuesta

1. Costos
2. Formas de pago y financiamiento
3. Horarios
4. Modalidad
5. Docentes
6. Requisitos de postulación
7. Duración de los programas
8. Malla curricular

¿Qué requisitos son necesarios para el acceso a un programa de posgrado o una maestría que oferta actualmente la UNL?

1 respuesta

1. Solicitud de admisión al programa de posgrado dirigida a la Decana de la Facultad correspondiente.
2. Cédula de ciudadanía para los postulantes nacionales; y, para los postulantes extranjeros, la cédula de identidad o pasaporte.
3. Foto tamaño carnet digital actualizada.
4. Copia simple del título de tercer nivel de grado debidamente registrado en la SENESCYT. En caso de que el título de grado sea obtenido en el extranjero, presentarlo debidamente apostillado o legalizado por vía consular.
5. Hoja de vida en la que conste formación académica y experiencia profesional.
6. Récord académico del nivel de grado.
7. Carta donde argumente las razones por las que desea estudiar en el programa de posgrado.

¿Cuál es el listado actual de los programas de posgrado que ofrece la UNL?

1 respuesta

1. Maestría en Dirección de Empresas
2. Maestría en Contabilidad y Finanzas mención Gestión Fiscal, Financiera y Tributaria
3. Maestría en Derecho Laboral y Seguridad Social
4. Maestría en Comunicación Política
5. Maestría en Agronegocios Sostenibles
6. Especialidad en Medicina Familiar y Comunitaria
7. Maestría en Políticas Públicas
8. Maestría en Finanzas mención en Dirección Financiera
9. Maestría en Gestión del Talento Humano
10. Maestría en Educación con Mención en Inclusión Educativa y Atención a la Diversidad
11. Maestría en Educación con mención en Innovación y Liderazgo Educativo
12. Maestría en Educación con Mención en Intervención Psicopedagógica
13. Maestría en Trabajo Social con Mención en Proyectos Sociales
14. Maestría en Educación con Mención en Orientación Educativa
15. Maestría en Economía y Dirección de Empresas
16. Maestría en Educación, Tecnología e Innovación
17. Maestría en Sanidad Animal
18. Maestría en Restauración de Paisajes Tropicales
19. Maestría en Educación Inicial
20. Maestría en Contabilidad y Finanzas
21. Maestría en Educación Básica
22. Maestría en Reproducción Animal con mención en Rumiantes
23. Maestría Educación con Mención en Docencia e Investigación en Educación Superior
24. Maestría en Ingeniería en Software
25. Maestría en Minas
26. Maestría en Telecomunicaciones
27. Maestría en Agroecología y Desarrollo Sostenible
28. Maestría en Turismo, Mención en Gestión e Innovación de Destinos
29. Maestría en Electricidad
30. Maestría en Biodiversidad y Cambio Climático
31. Maestría en Derecho Constitucional con Mención en Derechos Humanos

¿Qué parámetros (requisitos) son considerados para homologar en un posgrado de la UNL?

1 respuesta

No hemos realizado hasta la fecha procesos de homologaciones

¿Qué mecanismos o formas existen para homologar en un posgrado de la UNL?

1 respuesta

No hemos realizado hasta la fecha procesos de homologaciones

¿Cuáles son los requisitos para matricularse en los programas de posgrado de la Carrera de Computación de la UNL?

1 respuesta

1. Solicitud de admisión al programa de posgrado dirigida a la Decana de la Facultad correspondiente.
2. Cédula de ciudadanía para los postulantes nacionales; y, para los postulantes extranjeros, la cédula de identidad o pasaporte.
3. Foto tamaño carnet digital actualizada.
4. Copia simple del título de tercer nivel de grado debidamente registrado en la SENESCYT. En caso de que el título de grado sea obtenido en el extranjero, presentarlo debidamente apostillado o legalizado por vía consular.
5. Hoja de vida en la que conste formación académica y experiencia profesional.
6. Récord académico del nivel de grado.
7. Carta donde argumente las razones por las que desea estudiar en el programa de posgrado.

¿Qué modalidades de estudio se imparten en los programas de posgrado que se dan en la Carrera de Computación de la UNL?

1 respuesta

En línea

¿Cuál es la forma de pago o financiación de cada programa, en especial, los programas de posgrado de la Carrera de Computación de la UNL?

1 respuesta

En cuanto a nuestras Formas de pago tenemos:

- Efectivo (directamente en tesorería)
- Transferencia bancaria
- Tarjeta de crédito
- Crédito bancario
- Descuento a rol de pago (funcionarios UNL y externos con convenio)
- Convenio de pago directo con la UNL

¿Cuál es el tiempo o duración de los programas de posgrado que se imparten en la Carrera de Computación de la UNL?

1 respuesta

Dos periodos académicos (cada Periodo dura 4 meses de labores académicas)

¿Cuáles son los costos de los programas de posgrado de la Carrera de Computación de la UNL?

1 respuesta

-Maestría en Telecomunicaciones: Inversión: \$4.500, Matrícula: \$250, Arancel: \$4.250

¿Cómo se puede obtener información sobre el plan de estudios de los programas de posgrado de la Carrera de Computación de la UNL?

1 respuesta

Los interesados pueden encontrar la malla curricular en los brochure publicados en nuestra página web: www.unl.edu.ec/posgrados, o contactándose a los número telefónicos publicados en las redes sociales y afiches publicitarios.

¿Qué otros temas o consultas realizan los interesados sobre los programas de posgrado de la UNL?

1 respuesta

Aparte de las dudas que ya nombré en algún item anterior: Una consulta adicional frecuente es: si la Universidad recibe a estudiantes graduados en nivel tecnológico y sobre el manejo de los sistemas institucionales al momento de inscribirse.

Anexo 3. Preguntas realizadas al modelo DistilBERT.

```
contexto="''Los requisitos que deberán presentar los postulantes a los programas de posgrado que ofrece la Universidad Nacional de Loja, para la formación en especialización, maestría académica con trayectoria profesional y maestría académica con trayectoria de investigación, serán los siguientes: Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia; Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior, el postulante deberá presentarlo debidamente apostillado o legalizado por vía consular, de conformidad al Art. 22 del Reglamento de Régimen Académico del Consejo de Educación Superior; Documentos personales: cédula de ciudadanía y certificado de votación, actualizados; Nivel de dominio de lengua diferente a la materna, al menos a nivel B2 tomando como referencia el Marco Común Europeo para lenguas, si el proyecto curricular del programa de posgrado lo establece; Hoja de vida, donde conste: formación y experiencia profesional. Récord académico del nivel de grado; Carta donde argumente las razones por las que quiere estudiar y/o investigar en el programa de posgrado; Ensayo en temas relacionados con el posgrado, para la maestría académica con trayectoria de investigación.'''

pregunta="¿Qué requisitos se deben presentar para la inscripción a un posgrado?"

{'score': 0.008094879798591137, 'start': 300, 'end': 309, 'answer': 'Solicitud'}
```

Figura 6. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 1

```
contexto="''Los requisitos que deberán presentar los postulantes a los programas de posgrado que ofrece la Universidad Nacional de Loja, para la formación en especialización, maestría académica con trayectoria profesional y maestría académica con trayectoria de investigación, serán los siguientes: Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia; Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior, el postulante deberá presentarlo debidamente apostillado o legalizado por vía consular, de conformidad al Art. 22 del Reglamento de Régimen Académico del Consejo de Educación Superior; Documentos personales: cédula de ciudadanía y certificado de votación, actualizados; Nivel de dominio de lengua diferente a la materna, al menos a nivel B2 tomando como referencia el Marco Común Europeo para lenguas, si el proyecto curricular del programa de posgrado lo establece; Hoja de vida, donde conste: formación y experiencia profesional. Récord académico del nivel de grado; Carta donde argumente las razones por las que quiere estudiar y/o investigar en el programa de posgrado; Ensayo en temas relacionados con el posgrado, para la maestría académica con trayectoria de investigación.'''

pregunta="¿Requisitos para la inscripción?"

{'score': 0.021527700126171112, 'start': 467, 'end': 517, 'answer': 'Copia simple del título de tercer nivel o de grado'}
```

Figura 7. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 2

```
contexto="''Los requisitos que deberán presentar los postulantes a los programas de posgrado que ofrece la Universidad Nacional de Loja, para la formación en especialización, maestría académica con trayectoria profesional y maestría académica con trayectoria de investigación, serán los siguientes: Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia; Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior, el postulante deberá presentarlo debidamente apostillado o legalizado por vía consular, de conformidad al Art. 22 del Reglamento de Régimen Académico del Consejo de Educación Superior; Documentos personales: cédula de ciudadanía y certificado de votación, actualizados; Nivel de dominio de lengua diferente a la materna, al menos a nivel B2 tomando como referencia el Marco Común Europeo para lenguas, si el proyecto curricular del programa de posgrado lo establece; Hoja de vida, donde conste: formación y experiencia profesional. Récord académico del nivel de grado; Carta donde argumente las razones por las que quiere estudiar y/o investigar en el programa de posgrado; Ensayo en temas relacionados con el posgrado, para la maestría académica con trayectoria de investigación.'''

pregunta="¿Qué requisitos se deben presentar para postular a un posgrado?"

{'score': 0.5537686143070459, 'start': 300, 'end': 643, 'answer': 'Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia; Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior'}
```

Figura 8. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 3


```

1 s
contexto="''Los requisitos que deberán presentar los postulantes a los programas de posgrado que ofrece la Universidad Nacional de Loja, para la formación en especialización, maestría académica con trayectoria profesional y maestría académica con trayectoria de investigación, serán los siguientes: Solicitud de admisión al programa de posgrado, dirigida al Decano/a de la Facultad que corresponda o al director/a de la Unidad de Educación a Distancia; Copia simple del título de tercer nivel o de grado, debidamente registrado por el órgano rector de la política pública de Educación Superior. En caso de que el título de grado sea obtenido en el exterior, el postulante deberá presentarlo debidamente apostillado o legalizado por vía consular, de conformidad al Art. 22 del Reglamento de Régimen Académico del Consejo de Educación Superior; Documentos personales: cédula de ciudadanía y certificado de votación, actualizados; Nivel de dominio de lengua diferente a la materna, al menos a nivel B2 tomando como referencia el Marco Común Europeo para lenguas, si el proyecto curricular del programa de posgrado lo establece; Hoja de vida, donde conste: formación y experiencia profesional. Récord académico del nivel de grado; Carta donde argumente las razones por las que quiere estudiar y/o investigar en el programa de posgrado; Ensayo en temas relacionados con el posgrado, para la maestría académica con trayectoria de investigación.''"

pregunta="Dime los requisitos que debo presentar"

{ 'score': 0.20139334704726936, 'start': 300, 'end': 345, 'answer': 'Solicitud de admisión al programa de posgrado' }

```

Figura 9. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 4

```

1 s
contexto="''Para el pago de la matrícula y arancel se establecen las siguientes formas: Efectivo (directamente en tesorería) Transferencia bancaria, Tarjeta de crédito, Crédito bancario, Descuento a rol de pago (funcionarios UNL y externos con convenio), Convenio de pago directo con la UNL. Los costos de aranceles del programa de posgrado serán cancelados máximo hasta el tercer mes de iniciado el primer período académico ordinario, sin intereses; pasada la fecha se pagarán intereses de acuerdo a la Ley.''"

pregunta="¿Qué formas de pago existen?"

{ 'score': 0.2469973236322403, 'start': 76, 'end': 135, 'answer': 'Efectivo (directamente en tesorería),Transferencia bancaria' }

```

Figura 10. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 5

```

0 s
contexto="''Para el pago de la matrícula y arancel se establecen las siguientes formas: Efectivo (directamente en tesorería) Transferencia bancaria, Tarjeta de crédito, Crédito bancario, Descuento a rol de pago (funcionarios UNL y externos con convenio), Convenio de pago directo con la UNL. Los costos de aranceles del programa de posgrado serán cancelados máximo hasta el tercer mes de iniciado el primer período académico ordinario, sin intereses; pasada la fecha se pagarán intereses de acuerdo a la Ley.''"

pregunta="¿De qué formas puedo pagar una maestría?"

{ 'score': 0.6169973236322402, 'start': 76, 'end': 285, 'answer': 'Efectivo (directamente en tesorería),Transferencia bancaria, Tarjeta de crédito, Crédito bancario ' }

```

Figura 11. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 6

```

0 s
contexto="''Para el pago de la matrícula y arancel se establecen las siguientes formas: Efectivo (directamente en tesorería) Transferencia bancaria, Tarjeta de crédito, Crédito bancario, Descuento a rol de pago (funcionarios UNL y externos con convenio), Convenio de pago directo con la UNL. Los costos de aranceles del programa de posgrado serán cancelados máximo hasta el tercer mes de iniciado el primer período académico ordinario, sin intereses; pasada la fecha se pagarán intereses de acuerdo a la Ley.''"

pregunta="Formas de pago"

{ 'score': 0.012476273812353611, 'start': 0, 'end': 74, 'answer': 'Para el pago de la matrícula y arancel se establecen las siguientes formas' }

```

Figura 12. Funcionamiento del modelo BERT (DistilBERT): Ejemplo 7

Anexo 4. Entrenamiento del modelo DistilBERT.

```
0s ✓ #parámetros
batch_size=4
N_EPOCHS =5
lr=1e-5

Epoch 1: 100%|██████████| 41/41 [00:11<00:00, 3.49it/s, loss=1.77]
Epoch 2: 100%|██████████| 41/41 [00:11<00:00, 3.49it/s, loss=0.367]
Epoch 3: 100%|██████████| 41/41 [00:11<00:00, 3.47it/s, loss=0.501]
Epoch 4: 100%|██████████| 41/41 [00:11<00:00, 3.43it/s, loss=0.486]
Epoch 5: 100%|██████████| 41/41 [00:12<00:00, 3.39it/s, loss=0.403]

100%|██████████| 7/7 [00:00<00:00, 15.97it/s]
0.8214285714285714
```

Figura 13. Entrenamiento y ajuste de DistilBERT: 1º ejecución.

```
0s ✓ #parámetros
batch_size=6
N_EPOCHS =7
lr=1e-5

Epoch 1: 100%|██████████| 28/28 [00:11<00:00, 2.46it/s, loss=2.41]
Epoch 2: 100%|██████████| 28/28 [00:11<00:00, 2.45it/s, loss=0.367]
Epoch 3: 100%|██████████| 28/28 [00:11<00:00, 2.45it/s, loss=0.911]
Epoch 4: 100%|██████████| 28/28 [00:11<00:00, 2.43it/s, loss=0.0776]
Epoch 5: 100%|██████████| 28/28 [00:11<00:00, 2.40it/s, loss=0.206]
Epoch 6: 100%|██████████| 28/28 [00:11<00:00, 2.37it/s, loss=0.0567]
Epoch 7: 100%|██████████| 28/28 [00:11<00:00, 2.34it/s, loss=0.0431]

100%|██████████| 5/5 [00:00<00:00, 12.41it/s]
0.7583333402872086
```

Figura 14. Entrenamiento y ajuste de DistilBERT: 2º ejecución.

```
0s ✓ #parámetros
batch_size=8
N_EPOCHS =8
lr=1e-5

Epoch 1: 100%|██████████| 21/21 [00:11<00:00, 1.89it/s, loss=2.35]
Epoch 2: 100%|██████████| 21/21 [00:11<00:00, 1.87it/s, loss=0.677]
Epoch 3: 100%|██████████| 21/21 [00:11<00:00, 1.85it/s, loss=0.429]
Epoch 4: 100%|██████████| 21/21 [00:11<00:00, 1.80it/s, loss=0.0734]
Epoch 5: 100%|██████████| 21/21 [00:11<00:00, 1.78it/s, loss=0.353]
Epoch 6: 100%|██████████| 21/21 [00:12<00:00, 1.75it/s, loss=0.588]
Epoch 7: 100%|██████████| 21/21 [00:11<00:00, 1.76it/s, loss=0.0227]
Epoch 8: 100%|██████████| 21/21 [00:11<00:00, 1.79it/s, loss=0.547]

100%|██████████| 4/4 [00:00<00:00, 9.20it/s]
0.765625
```

Figura 15. Entrenamiento y ajuste de DistilBERT: 3º ejecución.

```

✓ 0s ▶ #parámetros
batch_size=12
N_EPOCHS =10
lr=1e-5

Epoch 1: 100% ██████████ | 14/14 [00:10<00:00, 1.36it/s, loss=2.4]
Epoch 2: 100% ██████████ | 14/14 [00:10<00:00, 1.36it/s, loss=1.57]
Epoch 3: 100% ██████████ | 14/14 [00:10<00:00, 1.35it/s, loss=0.656]
Epoch 4: 100% ██████████ | 14/14 [00:10<00:00, 1.33it/s, loss=1.3]
Epoch 5: 100% ██████████ | 14/14 [00:10<00:00, 1.32it/s, loss=0.18]
Epoch 6: 100% ██████████ | 14/14 [00:10<00:00, 1.30it/s, loss=0.291]
Epoch 7: 100% ██████████ | 14/14 [00:10<00:00, 1.29it/s, loss=0.174]
Epoch 8: 100% ██████████ | 14/14 [00:10<00:00, 1.29it/s, loss=0.302]
Epoch 9: 100% ██████████ | 14/14 [00:10<00:00, 1.31it/s, loss=0.163]
Epoch 10: 100% ██████████ | 14/14 [00:10<00:00, 1.30it/s, loss=0.0407]

100% ██████████ | 3/3 [00:00<00:00, 8.04it/s]
0.7916666766007742

```

Figura 16. Entrenamiento y ajuste de DistilBERT: 4º ejecución.

```

✓ 0s ▶ #parámetros
batch_size=10
N_EPOCHS =12
lr=1e-5

Epoch 1: 100% ██████████ | 17/17 [00:10<00:00, 1.64it/s, loss=2.97]
Epoch 2: 100% ██████████ | 17/17 [00:10<00:00, 1.64it/s, loss=1.02]
Epoch 3: 100% ██████████ | 17/17 [00:10<00:00, 1.63it/s, loss=0.577]
Epoch 4: 100% ██████████ | 17/17 [00:10<00:00, 1.62it/s, loss=0.166]
Epoch 5: 100% ██████████ | 17/17 [00:10<00:00, 1.61it/s, loss=0.406]
Epoch 6: 100% ██████████ | 17/17 [00:10<00:00, 1.60it/s, loss=0.32]
Epoch 7: 100% ██████████ | 17/17 [00:10<00:00, 1.57it/s, loss=0.157]
Epoch 8: 100% ██████████ | 17/17 [00:10<00:00, 1.55it/s, loss=0.045]
Epoch 9: 100% ██████████ | 17/17 [00:11<00:00, 1.51it/s, loss=0.25]
Epoch 10: 100% ██████████ | 17/17 [00:11<00:00, 1.50it/s, loss=0.0167]
Epoch 11: 100% ██████████ | 17/17 [00:11<00:00, 1.51it/s, loss=0.0587]
Epoch 12: 100% ██████████ | 17/17 [00:11<00:00, 1.54it/s, loss=0.0134]

100% ██████████ | 3/3 [00:00<00:00, 7.21it/s]
0.8166666726271311

```

Figura 17. Entrenamiento y ajuste de DistilBERT: 5º ejecución.

```

0 s ✓ #parámetros
batch_size=15
N_EPOCHS =12
lr=1e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.04it/s, loss=2.1]
Epoch 2: 100%|██████████| 11/11 [00:10<00:00, 1.03it/s, loss=1.84]
Epoch 3: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=0.867]
Epoch 4: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=0.932]
Epoch 5: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=0.8]
Epoch 6: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=0.255]
Epoch 7: 100%|██████████| 11/11 [00:10<00:00, 1.03it/s, loss=0.286]
Epoch 8: 100%|██████████| 11/11 [00:10<00:00, 1.03it/s, loss=0.213]
Epoch 9: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=0.344]
Epoch 10: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=0.0641]
Epoch 11: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.373]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.03s/it, loss=0.0441]

100%|██████████| 2/2 [00:00<00:00, 5.27it/s]
0.8397436439990997

```

Figura 18. Entrenamiento y ajuste de DistilBERT: 6ª ejecución.

```

0 s ✓ #parámetros
batch_size=15
N_EPOCHS =15
lr=1e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.05it/s, loss=1.98]
Epoch 2: 100%|██████████| 11/11 [00:10<00:00, 1.05it/s, loss=1.44]
Epoch 3: 100%|██████████| 11/11 [00:10<00:00, 1.03it/s, loss=1.03]
Epoch 4: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=0.522]
Epoch 5: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=0.429]
Epoch 6: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=0.329]
Epoch 7: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.284]
Epoch 8: 100%|██████████| 11/11 [00:11<00:00, 1.03s/it, loss=0.169]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.185]
Epoch 10: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.244]
Epoch 11: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0851]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.103]
Epoch 13: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0822]
Epoch 14: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.143]
Epoch 15: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0784]

100%|██████████| 2/2 [00:00<00:00, 5.18it/s]
0.7230769634246826

```

Figura 19. Entrenamiento y ajuste de DistilBERT: 7ª ejecución.

```

✓ [155] #parámetros
0s batch_size=5
    N_EPOCHS =6
    lr=2e-5

↳ Epoch 1: 100%|██████████| 33/33 [00:11<00:00, 2.91it/s, loss=1.19]
    Epoch 2: 100%|██████████| 33/33 [00:11<00:00, 2.89it/s, loss=2.02]
    Epoch 3: 100%|██████████| 33/33 [00:11<00:00, 2.86it/s, loss=0.854]
    Epoch 4: 100%|██████████| 33/33 [00:11<00:00, 2.85it/s, loss=0.193]
    Epoch 5: 100%|██████████| 33/33 [00:11<00:00, 2.82it/s, loss=0.109]
    Epoch 6: 100%|██████████| 33/33 [00:11<00:00, 2.79it/s, loss=0.0138]
↳ 100%|██████████| 6/6 [00:00<00:00, 14.38it/s]
    0.7722222271064917

```

Figura 20. Entrenamiento y ajuste de DistilBERT: 8ª ejecución.

```

✓ #parámetros
0s batch_size=7
    N_EPOCHS =8
    lr=2e-5

↳ Epoch 1: 100%|██████████| 24/24 [00:11<00:00, 2.11it/s, loss=1.46]
    Epoch 2: 100%|██████████| 24/24 [00:11<00:00, 2.10it/s, loss=0.285]
    Epoch 3: 100%|██████████| 24/24 [00:11<00:00, 2.08it/s, loss=0.509]
    Epoch 4: 100%|██████████| 24/24 [00:11<00:00, 2.06it/s, loss=0.0251]
    Epoch 5: 100%|██████████| 24/24 [00:11<00:00, 2.02it/s, loss=0.0388]
    Epoch 6: 100%|██████████| 24/24 [00:12<00:00, 1.98it/s, loss=0.0203]
    Epoch 7: 100%|██████████| 24/24 [00:12<00:00, 1.96it/s, loss=0.25]
    Epoch 8: 100%|██████████| 24/24 [00:12<00:00, 1.95it/s, loss=0.393]
↳ 100%|██████████| 4/4 [00:00<00:00, 10.02it/s]
    0.8214285895228386

```

Figura 21. Entrenamiento y ajuste de DistilBERT: 9ª ejecución.

```

✓ #parámetros
0s batch_size=9
    N_EPOCHS =9
    lr=2e-5

↳ Epoch 1: 100%|██████████| 19/19 [00:10<00:00, 1.74it/s, loss=2.68]
    Epoch 2: 100%|██████████| 19/19 [00:11<00:00, 1.72it/s, loss=0.228]
    Epoch 3: 100%|██████████| 19/19 [00:11<00:00, 1.70it/s, loss=1.99]
    Epoch 4: 100%|██████████| 19/19 [00:11<00:00, 1.67it/s, loss=0.0217]
    Epoch 5: 100%|██████████| 19/19 [00:11<00:00, 1.65it/s, loss=0.00989]
    Epoch 6: 100%|██████████| 19/19 [00:11<00:00, 1.62it/s, loss=0.00646]
    Epoch 7: 100%|██████████| 19/19 [00:11<00:00, 1.62it/s, loss=0.218]
    Epoch 8: 100%|██████████| 19/19 [00:11<00:00, 1.63it/s, loss=0.0197]
    Epoch 9: 100%|██████████| 19/19 [00:11<00:00, 1.63it/s, loss=0.0145]
↳ 100%|██████████| 4/4 [00:00<00:00, 10.26it/s]
    0.8472222313284874

```

Figura 22. Entrenamiento y ajuste de DistilBERT: 10ª ejecución.

```

✓ 2 s ▶ #parámetros
batch_size=13
N_EPOCHS =11
lr=2e-5

Epoch 1: 100% ██████████ | 13/13 [00:10<00:00, 1.22it/s, loss=2.79]
Epoch 2: 100% ██████████ | 13/13 [00:10<00:00, 1.22it/s, loss=0.896]
Epoch 3: 100% ██████████ | 13/13 [00:10<00:00, 1.21it/s, loss=0.528]
Epoch 4: 100% ██████████ | 13/13 [00:10<00:00, 1.20it/s, loss=0.257]
Epoch 5: 100% ██████████ | 13/13 [00:10<00:00, 1.19it/s, loss=0.246]
Epoch 6: 100% ██████████ | 13/13 [00:11<00:00, 1.17it/s, loss=0.245]
Epoch 7: 100% ██████████ | 13/13 [00:11<00:00, 1.16it/s, loss=0.245]
Epoch 8: 100% ██████████ | 13/13 [00:11<00:00, 1.16it/s, loss=0.032]
Epoch 9: 100% ██████████ | 13/13 [00:11<00:00, 1.17it/s, loss=0.0929]
Epoch 10: 100% ██████████ | 13/13 [00:11<00:00, 1.18it/s, loss=0.0177]
Epoch 11: 100% ██████████ | 13/13 [00:10<00:00, 1.18it/s, loss=0.0103]
100% ██████████ | 3/3 [00:00<00:00, 8.10it/s]
0.7307692468166351

```

Figura 23. Entrenamiento y ajuste de DistilBERT: 11ª ejecución.

```

✓ 0 s ▶ #parámetros
batch_size=11
N_EPOCHS =13
lr=2e-5

Epoch 1: 100% ██████████ | 15/15 [00:10<00:00, 1.44it/s, loss=1.95]
Epoch 2: 100% ██████████ | 15/15 [00:10<00:00, 1.43it/s, loss=0.513]
Epoch 3: 100% ██████████ | 15/15 [00:10<00:00, 1.39it/s, loss=0.362]
Epoch 4: 100% ██████████ | 15/15 [00:10<00:00, 1.37it/s, loss=0.0856]
Epoch 5: 100% ██████████ | 15/15 [00:10<00:00, 1.39it/s, loss=0.0268]
Epoch 6: 100% ██████████ | 15/15 [00:10<00:00, 1.37it/s, loss=0.139]
Epoch 7: 100% ██████████ | 15/15 [00:11<00:00, 1.35it/s, loss=0.0349]
Epoch 8: 100% ██████████ | 15/15 [00:11<00:00, 1.34it/s, loss=0.0131]
Epoch 9: 100% ██████████ | 15/15 [00:11<00:00, 1.34it/s, loss=0.232]
Epoch 10: 100% ██████████ | 15/15 [00:11<00:00, 1.35it/s, loss=0.0487]
Epoch 11: 100% ██████████ | 15/15 [00:11<00:00, 1.36it/s, loss=0.0539]
Epoch 12: 100% ██████████ | 15/15 [00:10<00:00, 1.36it/s, loss=0.0388]
Epoch 13: 100% ██████████ | 15/15 [00:11<00:00, 1.36it/s, loss=0.0338]
100% ██████████ | 3/3 [00:00<00:00, 7.31it/s]
0.7550505101680756

```

Figura 24. Entrenamiento y ajuste de DistilBERT: 12ª ejecución.

```

✓ 0s #parámetros
batch_size=16
N_EPOCHS =13
lr=2e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.05it/s, loss=1.41]
Epoch 2: 100%|██████████| 11/11 [00:10<00:00, 1.04it/s, loss=1.28]
Epoch 3: 100%|██████████| 11/11 [00:10<00:00, 1.03it/s, loss=1.71]
Epoch 4: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=0.527]
Epoch 5: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.223]
Epoch 6: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0244]
Epoch 7: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.0263]
Epoch 8: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.0542]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.0869]
Epoch 10: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.279]
Epoch 11: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.167]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.0404]
Epoch 13: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0511]

100%|██████████| 2/2 [00:00<00:00, 5.13it/s]
0.8177083432674408

```

Figura 25. Entrenamiento y ajuste de DistilBERT: 13ª ejecución.

```

✓ [317] 0s #parámetros
batch_size=15
N_EPOCHS =12
lr=3e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=2.27]
Epoch 2: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.935]
Epoch 3: 100%|██████████| 11/11 [00:11<00:00, 1.03s/it, loss=0.468]
Epoch 4: 100%|██████████| 11/11 [00:11<00:00, 1.04s/it, loss=0.424]
Epoch 5: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.0803]
Epoch 6: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.053]
Epoch 7: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=0.326]
Epoch 8: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=0.0961]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.198]
Epoch 10: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.232]
Epoch 11: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0292]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0371]

100%|██████████| 2/2 [00:00<00:00, 5.02it/s]
0.8038461804389954

```

Figura 26. Entrenamiento y ajuste de DistilBERT: 14ª ejecución.

```

✓ 0s ▶ #parámetros
batch_size=15
N_EPOCHS =15
lr=2e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.02it/s, loss=1.96]
Epoch 2: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=1.06]
Epoch 3: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.418]
Epoch 4: 100%|██████████| 11/11 [00:11<00:00, 1.04s/it, loss=0.531]
Epoch 5: 100%|██████████| 11/11 [00:11<00:00, 1.03s/it, loss=0.0835]
Epoch 6: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.128]
Epoch 7: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.0315]
Epoch 8: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=0.0208]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.228]
Epoch 10: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0831]
Epoch 11: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.12]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0486]
Epoch 13: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0184]
Epoch 14: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.00464]
Epoch 15: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0483]
100%|██████████| 2/2 [00:00<00:00, 4.95it/s]
0.8205128610134125

```

Figura 27. Entrenamiento y ajuste de DistilBERT: 15° ejecución.

```

✓ 0s ▶ #parámetros
batch_size=10
N_EPOCHS =6
lr=3e-5

Epoch 1: 100%|██████████| 17/17 [00:10<00:00, 1.60it/s, loss=0.812]
Epoch 2: 100%|██████████| 17/17 [00:10<00:00, 1.58it/s, loss=0.132]
Epoch 3: 100%|██████████| 17/17 [00:10<00:00, 1.56it/s, loss=0.498]
Epoch 4: 100%|██████████| 17/17 [00:11<00:00, 1.54it/s, loss=0.193]
Epoch 5: 100%|██████████| 17/17 [00:11<00:00, 1.51it/s, loss=0.00906]
Epoch 6: 100%|██████████| 17/17 [00:11<00:00, 1.49it/s, loss=0.0509]
100%|██████████| 3/3 [00:00<00:00, 7.84it/s]
0.7958333392937978

```

Figura 28. Entrenamiento y ajuste de DistilBERT: 16° ejecución.

```

✓ 0s ▶ #parámetros
batch_size=15
N_EPOCHS =7
lr=3e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=1.27]
Epoch 2: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.615]
Epoch 3: 100%|██████████| 11/11 [00:11<00:00, 1.04s/it, loss=0.787]
Epoch 4: 100%|██████████| 11/11 [00:11<00:00, 1.04s/it, loss=0.171]
Epoch 5: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.355]
Epoch 6: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.471]
Epoch 7: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=0.0871]
100%|██████████| 2/2 [00:00<00:00, 5.01it/s]
0.7871795147657394

```

Figura 29. Entrenamiento y ajuste de DistilBERT: 17° ejecución.


```

✓ 0 s #parámetros
batch_size=11
N_EPOCHS =10
lr=3e-5

Epoch 1: 100%|██████████| 15/15 [00:10<00:00, 1.39it/s, loss=1.5]
Epoch 2: 100%|██████████| 15/15 [00:10<00:00, 1.37it/s, loss=0.464]
Epoch 3: 100%|██████████| 15/15 [00:11<00:00, 1.34it/s, loss=0.231]
Epoch 4: 100%|██████████| 15/15 [00:11<00:00, 1.32it/s, loss=0.0375]
Epoch 5: 100%|██████████| 15/15 [00:11<00:00, 1.33it/s, loss=0.263]
Epoch 6: 100%|██████████| 15/15 [00:11<00:00, 1.35it/s, loss=0.0427]
Epoch 7: 100%|██████████| 15/15 [00:11<00:00, 1.36it/s, loss=0.114]
Epoch 8: 100%|██████████| 15/15 [00:10<00:00, 1.37it/s, loss=0.0184]
Epoch 9: 100%|██████████| 15/15 [00:11<00:00, 1.36it/s, loss=0.0846]
Epoch 10: 100%|██████████| 15/15 [00:11<00:00, 1.36it/s, loss=0.00671]

100%|██████████| 3/3 [00:00<00:00, 7.35it/s]
0.785353551308314

```

Figura 30. Entrenamiento y ajuste de DistilBERT: 18ª ejecución.

```

✓ [317] 0 s #parámetros
batch_size=15
N_EPOCHS =12
lr=3e-5

Epoch 1: 100%|██████████| 11/11 [00:10<00:00, 1.01it/s, loss=2.27]
Epoch 2: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.935]
Epoch 3: 100%|██████████| 11/11 [00:11<00:00, 1.03s/it, loss=0.468]
Epoch 4: 100%|██████████| 11/11 [00:11<00:00, 1.04s/it, loss=0.424]
Epoch 5: 100%|██████████| 11/11 [00:11<00:00, 1.02s/it, loss=0.0803]
Epoch 6: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.053]
Epoch 7: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=0.326]
Epoch 8: 100%|██████████| 11/11 [00:10<00:00, 1.00it/s, loss=0.0961]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.00s/it, loss=0.198]
Epoch 10: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.232]
Epoch 11: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0292]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.01s/it, loss=0.0371]

100%|██████████| 2/2 [00:00<00:00, 5.02it/s]
0.8038461804389954

```

Figura 31. Entrenamiento y ajuste de DistilBERT: 19ª ejecución.

```

0 s ✓ ▶ #parámetros
batch_size=15
N_EPOCHS =13
lr=3e-5

Epoch 1: 100% ██████████ | 11/11 [00:10<00:00, 1.03it/s, loss=1.48]
Epoch 2: 100% ██████████ | 11/11 [00:10<00:00, 1.02it/s, loss=0.735]
Epoch 3: 100% ██████████ | 11/11 [00:11<00:00, 1.00s/it, loss=0.334]
Epoch 4: 100% ██████████ | 11/11 [00:11<00:00, 1.02s/it, loss=0.209]
Epoch 5: 100% ██████████ | 11/11 [00:11<00:00, 1.04s/it, loss=0.208]
Epoch 6: 100% ██████████ | 11/11 [00:11<00:00, 1.02s/it, loss=0.068]
Epoch 7: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0209]
Epoch 8: 100% ██████████ | 11/11 [00:10<00:00, 1.00it/s, loss=0.0129]
Epoch 9: 100% ██████████ | 11/11 [00:10<00:00, 1.00it/s, loss=0.0969]
Epoch 10: 100% ██████████ | 11/11 [00:11<00:00, 1.00s/it, loss=0.0489]
Epoch 11: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.00738]
Epoch 12: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0498]
Epoch 13: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0041]

100% ██████████ | 2/2 [00:00<00:00, 5.27it/s]
0.8205128610134125

```

Figura 32. Entrenamiento y ajuste de DistilBERT: 20ª ejecución.

```

3 s ✓ ▶ #parámetros
batch_size=15
N_EPOCHS =15
lr=3e-5

Epoch 1: 100% ██████████ | 11/11 [00:10<00:00, 1.02it/s, loss=1.63]
Epoch 2: 100% ██████████ | 11/11 [00:10<00:00, 1.00it/s, loss=0.587]
Epoch 3: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.599]
Epoch 4: 100% ██████████ | 11/11 [00:11<00:00, 1.04s/it, loss=0.661]
Epoch 5: 100% ██████████ | 11/11 [00:11<00:00, 1.03s/it, loss=0.261]
Epoch 6: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.167]
Epoch 7: 100% ██████████ | 11/11 [00:11<00:00, 1.00s/it, loss=0.105]
Epoch 8: 100% ██████████ | 11/11 [00:10<00:00, 1.00it/s, loss=0.135]
Epoch 9: 100% ██████████ | 11/11 [00:11<00:00, 1.00s/it, loss=0.0268]
Epoch 10: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0216]
Epoch 11: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.00569]
Epoch 12: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0443]
Epoch 13: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0191]
Epoch 14: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0101]
Epoch 15: 100% ██████████ | 11/11 [00:11<00:00, 1.01s/it, loss=0.0348]

100% ██████████ | 2/2 [00:00<00:00, 5.33it/s]
0.8038461804389954

```

Figura 33. Entrenamiento y ajuste de DistilBERT: 21ª ejecución.

```

0 s ✓ #parámetros
batch_size=5
N_EPOCHS =6
lr=4e-5

Epoch 1: 100%|██████████| 33/33 [00:15<00:00, 2.12it/s, loss=0.639]
Epoch 2: 100%|██████████| 33/33 [00:11<00:00, 2.79it/s, loss=0.606]
Epoch 3: 100%|██████████| 33/33 [00:11<00:00, 2.83it/s, loss=0.169]
Epoch 4: 100%|██████████| 33/33 [00:11<00:00, 2.79it/s, loss=0.372]
Epoch 5: 100%|██████████| 33/33 [00:12<00:00, 2.72it/s, loss=0.0149]
Epoch 6: 100%|██████████| 33/33 [00:12<00:00, 2.67it/s, loss=0.0106]

100%|██████████| 7/7 [00:00<00:00, 14.47it/s]
0.8000000055347171

```

Figura 34. Entrenamiento y ajuste de DistilBERT: 22° ejecución.

```

0 s ✓ #parámetros
batch_size=10
N_EPOCHS =7
lr=4e-5

Epoch 1: 100%|██████████| 17/17 [00:11<00:00, 1.52it/s, loss=0.906]
Epoch 2: 100%|██████████| 17/17 [00:11<00:00, 1.48it/s, loss=0.276]
Epoch 3: 100%|██████████| 17/17 [00:11<00:00, 1.46it/s, loss=1.26]
Epoch 4: 100%|██████████| 17/17 [00:12<00:00, 1.40it/s, loss=0.0683]
Epoch 5: 100%|██████████| 17/17 [00:12<00:00, 1.36it/s, loss=0.0336]
Epoch 6: 100%|██████████| 17/17 [00:12<00:00, 1.36it/s, loss=0.449]
Epoch 7: 100%|██████████| 17/17 [00:12<00:00, 1.40it/s, loss=0.148]

100%|██████████| 4/4 [00:00<00:00, 8.22it/s]
0.7625000104308128

```

Figura 35. Entrenamiento y ajuste de DistilBERT: 23° ejecución.

```

0 s ✓ #parámetros
batch_size=8
N_EPOCHS =8
lr=4e-5

Epoch 1: 100%|██████████| 21/21 [00:12<00:00, 1.74it/s, loss=1.75]
Epoch 2: 100%|██████████| 21/21 [00:12<00:00, 1.71it/s, loss=0.195]
Epoch 3: 100%|██████████| 21/21 [00:12<00:00, 1.66it/s, loss=1.48]
Epoch 4: 100%|██████████| 21/21 [00:13<00:00, 1.60it/s, loss=0.0547]
Epoch 5: 100%|██████████| 21/21 [00:13<00:00, 1.60it/s, loss=0.0588]
Epoch 6: 100%|██████████| 21/21 [00:12<00:00, 1.65it/s, loss=0.0461]
Epoch 7: 100%|██████████| 21/21 [00:12<00:00, 1.68it/s, loss=0.0423]
Epoch 8: 100%|██████████| 21/21 [00:12<00:00, 1.68it/s, loss=0.00331]

100%|██████████| 4/4 [00:00<00:00, 8.14it/s]
0.888125

```

Figura 36. Entrenamiento y ajuste de DistilBERT: 24° ejecución.

```

0s #parámetros
batch_size=13
N_EPOCHS =8
lr=4e-5

Epoch 1: 100%|██████████| 13/13 [00:11<00:00, 1.14it/s, loss=1.4]
Epoch 2: 100%|██████████| 13/13 [00:11<00:00, 1.12it/s, loss=0.303]
Epoch 3: 100%|██████████| 13/13 [00:11<00:00, 1.09it/s, loss=1.07]
Epoch 4: 100%|██████████| 13/13 [00:12<00:00, 1.07it/s, loss=0.0161]
Epoch 5: 100%|██████████| 13/13 [00:12<00:00, 1.04it/s, loss=0.112]
Epoch 6: 100%|██████████| 13/13 [00:12<00:00, 1.07it/s, loss=0.0952]
Epoch 7: 100%|██████████| 13/13 [00:11<00:00, 1.09it/s, loss=0.00549]
Epoch 8: 100%|██████████| 13/13 [00:11<00:00, 1.11it/s, loss=0.068]

100%|██████████| 3/3 [00:00<00:00, 6.54it/s]
0.784188042084376

```

Figura 37. Entrenamiento y ajuste de DistilBERT: 25° ejecución.

```

0s #parámetros
batch_size=12
N_EPOCHS =10
lr=4e-5

Epoch 1: 100%|██████████| 14/14 [00:11<00:00, 1.23it/s, loss=1.39]
Epoch 2: 100%|██████████| 14/14 [00:11<00:00, 1.21it/s, loss=0.48]
Epoch 3: 100%|██████████| 14/14 [00:12<00:00, 1.16it/s, loss=0.185]
Epoch 4: 100%|██████████| 14/14 [00:12<00:00, 1.14it/s, loss=0.053]
Epoch 5: 100%|██████████| 14/14 [00:12<00:00, 1.15it/s, loss=0.0851]
Epoch 6: 100%|██████████| 14/14 [00:11<00:00, 1.17it/s, loss=0.0248]
Epoch 7: 100%|██████████| 14/14 [00:11<00:00, 1.18it/s, loss=0.0333]
Epoch 8: 100%|██████████| 14/14 [00:11<00:00, 1.18it/s, loss=0.077]
Epoch 9: 100%|██████████| 14/14 [00:11<00:00, 1.18it/s, loss=0.00786]
Epoch 10: 100%|██████████| 14/14 [00:11<00:00, 1.17it/s, loss=0.00536]

100%|██████████| 3/3 [00:00<00:00, 6.37it/s]
0.8125000099341074

```

Figura 38. Entrenamiento y ajuste de DistilBERT: 26° ejecución.

```

0s #parámetros
batch_size=12
N_EPOCHS =12
lr=4e-5

Epoch 1: 100%|██████████| 14/14 [00:11<00:00, 1.23it/s, loss=1.58]
Epoch 2: 100%|██████████| 14/14 [00:11<00:00, 1.20it/s, loss=0.488]
Epoch 3: 100%|██████████| 14/14 [00:12<00:00, 1.16it/s, loss=0.329]
Epoch 4: 100%|██████████| 14/14 [00:12<00:00, 1.12it/s, loss=0.272]
Epoch 5: 100%|██████████| 14/14 [00:12<00:00, 1.15it/s, loss=0.127]
Epoch 6: 100%|██████████| 14/14 [00:11<00:00, 1.18it/s, loss=0.0153]
Epoch 7: 100%|██████████| 14/14 [00:11<00:00, 1.19it/s, loss=0.136]
Epoch 8: 100%|██████████| 14/14 [00:11<00:00, 1.19it/s, loss=0.206]
Epoch 9: 100%|██████████| 14/14 [00:11<00:00, 1.19it/s, loss=0.0124]
Epoch 10: 100%|██████████| 14/14 [00:11<00:00, 1.18it/s, loss=0.0289]
Epoch 11: 100%|██████████| 14/14 [00:11<00:00, 1.18it/s, loss=0.108]
Epoch 12: 100%|██████████| 14/14 [00:11<00:00, 1.17it/s, loss=0.0373]

100%|██████████| 3/3 [00:00<00:00, 6.68it/s]
0.8355555621782938

```

Figura 39. Entrenamiento y ajuste de DistilBERT: 27° ejecución.

```

0 s ✓ #parámetros
batch_size=15
N_EPOCHS =15
lr=4e-5

Epoch 1: 100% ██████████ | 11/11 [00:11<00:00, 1.04s/it, loss=1.32]
Epoch 2: 100% ██████████ | 11/11 [00:11<00:00, 1.06s/it, loss=0.671]
Epoch 3: 100% ██████████ | 11/11 [00:11<00:00, 1.08s/it, loss=0.225]
Epoch 4: 100% ██████████ | 11/11 [00:12<00:00, 1.12s/it, loss=0.16]
Epoch 5: 100% ██████████ | 11/11 [00:12<00:00, 1.14s/it, loss=0.0643]
Epoch 6: 100% ██████████ | 11/11 [00:12<00:00, 1.11s/it, loss=0.0205]
Epoch 7: 100% ██████████ | 11/11 [00:11<00:00, 1.08s/it, loss=0.0105]
Epoch 8: 100% ██████████ | 11/11 [00:11<00:00, 1.07s/it, loss=0.0042]
Epoch 9: 100% ██████████ | 11/11 [00:11<00:00, 1.08s/it, loss=0.0074]
Epoch 10: 100% ██████████ | 11/11 [00:12<00:00, 1.10s/it, loss=0.0459]
Epoch 11: 100% ██████████ | 11/11 [00:12<00:00, 1.10s/it, loss=0.0362]
Epoch 12: 100% ██████████ | 11/11 [00:12<00:00, 1.10s/it, loss=0.0271]
Epoch 13: 100% ██████████ | 11/11 [00:12<00:00, 1.10s/it, loss=0.00992]
Epoch 14: 100% ██████████ | 11/11 [00:12<00:00, 1.10s/it, loss=0.0172]
Epoch 15: 100% ██████████ | 11/11 [00:12<00:00, 1.10s/it, loss=0.00393]

100% ██████████ | 3/3 [00:00<00:00, 6.54it/s]
0.7000000278155009

```

Figura 40. Entrenamiento y ajuste de DistilBERT: 28ª ejecución.

```

0 s ✓ #parámetros
batch_size=9
N_EPOCHS =7
lr=5e-5

Epoch 1: 100% ██████████ | 19/19 [00:12<00:00, 1.55it/s, loss=0.668]
Epoch 2: 100% ██████████ | 19/19 [00:12<00:00, 1.49it/s, loss=0.318]
Epoch 3: 100% ██████████ | 19/19 [00:13<00:00, 1.44it/s, loss=0.0506]
Epoch 4: 100% ██████████ | 19/19 [00:13<00:00, 1.46it/s, loss=1.3]
Epoch 5: 100% ██████████ | 19/19 [00:12<00:00, 1.50it/s, loss=0.0994]
Epoch 6: 100% ██████████ | 19/19 [00:12<00:00, 1.52it/s, loss=0.141]
Epoch 7: 100% ██████████ | 19/19 [00:12<00:00, 1.53it/s, loss=0.0054]

100% ██████████ | 4/4 [00:00<00:00, 8.89it/s]
0.7638888955116272

```

Figura 41. Entrenamiento y ajuste de DistilBERT: 29ª ejecución.

```

0 s ✓ #parámetros
batch_size=12
N_EPOCHS =9
lr=5e-5

Epoch 1: 100% ██████████ | 14/14 [00:11<00:00, 1.22it/s, loss=1.67]
Epoch 2: 100% ██████████ | 14/14 [00:11<00:00, 1.17it/s, loss=0.993]
Epoch 3: 100% ██████████ | 14/14 [00:12<00:00, 1.13it/s, loss=0.629]
Epoch 4: 100% ██████████ | 14/14 [00:12<00:00, 1.13it/s, loss=0.301]
Epoch 5: 100% ██████████ | 14/14 [00:12<00:00, 1.17it/s, loss=0.0135]
Epoch 6: 100% ██████████ | 14/14 [00:11<00:00, 1.19it/s, loss=0.063]
Epoch 7: 100% ██████████ | 14/14 [00:11<00:00, 1.19it/s, loss=0.0551]
Epoch 8: 100% ██████████ | 14/14 [00:11<00:00, 1.19it/s, loss=0.0143]
Epoch 9: 100% ██████████ | 14/14 [00:11<00:00, 1.18it/s, loss=0.011]

100% ██████████ | 3/3 [00:00<00:00, 6.46it/s]
0.8925000099341074

```

Figura 42. Entrenamiento y ajuste de DistilBERT: 30ª ejecución.

```

0 s ✓ #parámetros
batch_size=15
N_EPOCHS =9
lr=5e-5

Epoch 1: 100%|██████████| 11/11 [00:11<00:00, 1.04s/it, loss=1.6]
Epoch 2: 100%|██████████| 11/11 [00:11<00:00, 1.09s/it, loss=0.452]
Epoch 3: 100%|██████████| 11/11 [00:12<00:00, 1.15s/it, loss=0.329]
Epoch 4: 100%|██████████| 11/11 [00:12<00:00, 1.16s/it, loss=0.204]
Epoch 5: 100%|██████████| 11/11 [00:12<00:00, 1.11s/it, loss=0.239]
Epoch 6: 100%|██████████| 11/11 [00:12<00:00, 1.09s/it, loss=0.35]
Epoch 7: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.605]
Epoch 8: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.0678]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.0566]

100%|██████████| 3/3 [00:00<00:00, 6.40it/s]
0.722222258647283

```

Figura 43. Entrenamiento y ajuste de DistilBERT: 31° ejecución.

```

0 s ✓ #parámetros
batch_size=13
N_EPOCHS =10
lr=5e-5

Epoch 1: 100%|██████████| 13/13 [00:11<00:00, 1.13it/s, loss=0.88]
Epoch 2: 100%|██████████| 13/13 [00:11<00:00, 1.11it/s, loss=0.513]
Epoch 3: 100%|██████████| 13/13 [00:12<00:00, 1.06it/s, loss=0.777]
Epoch 4: 100%|██████████| 13/13 [00:12<00:00, 1.05it/s, loss=0.349]
Epoch 5: 100%|██████████| 13/13 [00:12<00:00, 1.07it/s, loss=0.0455]
Epoch 6: 100%|██████████| 13/13 [00:12<00:00, 1.08it/s, loss=0.279]
Epoch 7: 100%|██████████| 13/13 [00:11<00:00, 1.09it/s, loss=0.109]
Epoch 8: 100%|██████████| 13/13 [00:11<00:00, 1.09it/s, loss=0.0636]
Epoch 9: 100%|██████████| 13/13 [00:11<00:00, 1.09it/s, loss=0.0228]
Epoch 10: 100%|██████████| 13/13 [00:11<00:00, 1.09it/s, loss=0.0347]

100%|██████████| 3/3 [00:00<00:00, 6.44it/s]
0.784188042084376

```

Figura 44. Entrenamiento y ajuste de DistilBERT: 32° ejecución.

```

0 s ✓ #parámetros
batch_size=20
N_EPOCHS =11
lr=5e-5

Epoch 1: 100%|██████████| 9/9 [00:11<00:00, 1.24s/it, loss=1.35]
Epoch 2: 100%|██████████| 9/9 [00:11<00:00, 1.26s/it, loss=0.468]
Epoch 3: 100%|██████████| 9/9 [00:11<00:00, 1.29s/it, loss=0.0569]
Epoch 4: 100%|██████████| 9/9 [00:11<00:00, 1.31s/it, loss=0.551]
Epoch 5: 100%|██████████| 9/9 [00:11<00:00, 1.31s/it, loss=0.998]
Epoch 6: 100%|██████████| 9/9 [00:11<00:00, 1.29s/it, loss=0.0728]
Epoch 7: 100%|██████████| 9/9 [00:11<00:00, 1.28s/it, loss=0.0108]
Epoch 8: 100%|██████████| 9/9 [00:11<00:00, 1.26s/it, loss=0.0359]
Epoch 9: 100%|██████████| 9/9 [00:11<00:00, 1.26s/it, loss=0.199]
Epoch 10: 100%|██████████| 9/9 [00:11<00:00, 1.27s/it, loss=0.00615]
Epoch 11: 100%|██████████| 9/9 [00:11<00:00, 1.29s/it, loss=0.012]

100%|██████████| 2/2 [00:00<00:00, 4.42it/s]
0.800000011920929

```

Figura 45. Entrenamiento y ajuste de DistilBERT: 33° ejecución.

```

0 s #parámetros
batch_size=16
N_EPOCHS =15
lr=5e-5

Epoch 1: 100%|██████████| 11/11 [00:11<00:00, 1.07s/it, loss=1.18]
Epoch 2: 100%|██████████| 11/11 [00:12<00:00, 1.12s/it, loss=0.761]
Epoch 3: 100%|██████████| 11/11 [00:12<00:00, 1.14s/it, loss=0.223]
Epoch 4: 100%|██████████| 11/11 [00:12<00:00, 1.12s/it, loss=0.124]
Epoch 5: 100%|██████████| 11/11 [00:11<00:00, 1.09s/it, loss=0.0419]
Epoch 6: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.0237]
Epoch 7: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.0762]
Epoch 8: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.147]
Epoch 9: 100%|██████████| 11/11 [00:11<00:00, 1.09s/it, loss=0.00199]
Epoch 10: 100%|██████████| 11/11 [00:12<00:00, 1.09s/it, loss=0.0241]
Epoch 11: 100%|██████████| 11/11 [00:12<00:00, 1.10s/it, loss=0.003]
Epoch 12: 100%|██████████| 11/11 [00:11<00:00, 1.09s/it, loss=0.0293]
Epoch 13: 100%|██████████| 11/11 [00:11<00:00, 1.08s/it, loss=0.00336]
Epoch 14: 100%|██████████| 11/11 [00:11<00:00, 1.09s/it, loss=0.0272]
Epoch 15: 100%|██████████| 11/11 [00:12<00:00, 1.10s/it, loss=0.00167]

100%|██████████| 2/2 [00:00<00:00, 4.22it/s]
0.8125

```

Figura 46. Entrenamiento y ajuste de DistilBERT: 34ª ejecución.

Anexo 5. Historias de usuario.

Tabla 1. Roles de los involucrados en el proyecto.

Rol	Descripción
Programador	Persona encargada de desarrollar e investigar todo el proyecto.
Usuario Final	Personas que tendrán interacción con el agente conversacional para determinar su correcta funcionalidad.

Tabla 2. Historia de usuario 1: Visualización del agente conversacional.

H01	Visualización del chatbot
<p>Como usuario</p> <p>Quiero un botón con el título “Resuelva sus dudas” situado en una página web</p> <p>Para pulsar sobre éste y que muestre el despliegue de un asistente virtual (chatbot) que tendrá por nombre “Posgrados UNL”</p>	
<p>Validación</p> <ul style="list-style-type: none"> • Ingresar a la página web a través de una dirección URL y verificar que se cargue correctamente en la laptop. • Comprobar que en la página web se visualice un botón. • Comprobar que el botón tenga un nombre. • Comprobar que se produce el despliegue del chatbot al pulsar el botón. 	<p>Desarrollador: Leonardo Paredes</p> <p>Prioridad: 1</p> <p>Estimación: 14 H</p>

Tabla 3. Historia de usuario 2: Presentación del chatbot.

H02	Presentación del agente conversacional
<p>Como usuario Quiero que después del despliegue del chatbot se presente diciendo que es un asistente virtual creado para responder preguntas frecuentes sobre posgrados y que mencione brevemente los temas de los que tiene conocimientos. Además, debe mostrar un pequeño menú de opciones compuesto por números y nombres Para facilitar a los usuarios algunos de los temas que el asistente virtual pueda responder.</p>	
<p>Validación</p> <ul style="list-style-type: none"> • Comprobar que se muestre el mensaje de presentación. • Comprobar que se mencionan los temas de los que tiene información el chatbot. • Comprobar que se muestre un menú de opciones. 	<p>Desarrollador: Leonardo Paredes</p> <p>Prioridad: 3</p> <p>Estimación: 72 H</p>

Tabla 4. Historia de usuario 3: Diálogo de preguntas.

H03	Diálogo de preguntas
<p>Como usuario Quiero un campo de texto dentro del chatbot, Para escribir las consultas sobre los posgrados y que el envío de la pregunta sea detectado mediante la tecla “Enter” o algún botón. Además, se debe presentar una respuesta para cada consulta que se realice.</p>	
<p>Validación</p> <ul style="list-style-type: none"> • Comprobar que exista el campo de texto en el chatbot. • Comprobar que se pueda ingresar en el campo de texto el valor numérico o el nombre de cada opción del menú que apareció durante la presentación del asistente conversacional. • Comprobar que el campo de texto permita escribir una pregunta. • Comprobar que las consultas se envíen al pulsar la tecla “Enter” o el botón de envío. • Comprobar que cada opción del menú funcione correctamente. • Comprobar que se muestre una respuesta. 	<p>Desarrollador: Leonardo Paredes</p> <p>Prioridad: 4</p> <p>Estimación: 672 H</p>

Tabla 5. Historia de usuario 4: Opciones de ayuda.

H04	Opciones de ayuda
<p>Como usuario</p> <p>Quiero que se presente un mensaje de ayuda, cuando el chatbot no pueda responder alguna pregunta, y en dicho mensaje, se mencione el uso de un comando</p> <p>Para visualizar todos los temas en forma de menú que el asistente virtual puede responder</p>	
<p>Validación</p> <ul style="list-style-type: none"> • Comprobar que se visualice un diálogo donde se especifique el uso del comando “ayuda”. • Comprobar que se presenta un menú de opciones al utilizar el comando “ayuda” • Comprobar que cada opción del menú que se presente tras usar el comando “ayuda” muestre una respuesta. 	<p>Desarrollador: Leonardo Paredes</p> <p>Prioridad: 5</p> <p>Estimación: 46 H</p>

Anexo 6. Encuesta aplicada a tres interesados, para comprobar los criterios de validación de las historias de usuario del agente conversacional.

No se pueden editar las respuestas

La presente encuesta tiene como objetivo recoger información de las pruebas de validación hechas al agente conversacional y ver si cumple con lo establecido en las historias de usuario

Por favor, para realizar las pruebas ingrese al siguiente enlace: <http://147.182.229.85/>

Se ha registrado el correo del encuestado (deiby.calva@unl.edu.ec) al enviar este formulario.

* Indica que la pregunta es obligatoria

Nombres y apellidos completos *

Deiby Patricio Calva

Código de la Historia de Usuario: H01 *

	Si	No
¿Se puede acceder a la página web donde está implementado el chatbot?	<input checked="" type="radio"/>	<input type="radio"/>
¿En la página web se presenta un botón con el nombre "Resuelva sus dudas"?	<input checked="" type="radio"/>	<input type="radio"/>
¿Al pulsar sobre el botón se despliega el chatbot en la pantalla?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot se visualiza correctamente en la laptop?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H02 *

	Si	No
¿El chatbot muestra un mensaje de presentación?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot menciona a que temas puede dar respuesta?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot presenta un menú compuesto por dígitos y nombres sobre temas que puede responder?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H03 *

	Si	No
¿El chatbot muestra un campo de texto para ingresar alguna consulta?	<input checked="" type="radio"/>	<input type="radio"/>
En el campo de texto, ¿el chatbot permite ingresar el valor numérico o el nombre de las opciones del menú?	<input checked="" type="radio"/>	<input type="radio"/>
¿Las opciones del menú funcionan correctamente?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot permite ingresar una pregunta de autoría propia del usuario?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot muestra una respuesta a cada consulta echa por el usuario?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H04 *

	Si	No
¿El chatbot presenta un mensaje cuando no encontró respuesta a una pregunta?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot muestra el uso del comando "Ayuda"?	<input checked="" type="radio"/>	<input type="radio"/>
¿El comando "Ayuda" funciona correctamente?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot presenta un menú de 25 opciones tras usar el comando "Ayuda"?	<input checked="" type="radio"/>	<input type="radio"/>
¿Cada una de las opciones del menú funcionan correctamente?	<input checked="" type="radio"/>	<input type="radio"/>

No se pueden editar las respuestas

La presente encuesta tiene como objetivo recoger información de las pruebas de validación hechas al agente conversacional y ver si cumple con lo establecido en las historias de usuario

Por favor, para realizar las pruebas ingrese al siguiente enlace: <http://147.182.229.85/>

Se ha registrado el correo del encuestado (jose.a.carrion.o@unl.edu.ec) al enviar este formulario.

* Indica que la pregunta es obligatoria

Nombres y apellidos completos *

José Alexis Carrión Ojeda

Código de la Historia de Usuario: H01 *

	Si	No
¿Se puede acceder a la página web donde está implementado el chatbot?	<input checked="" type="radio"/>	<input type="radio"/>
¿En la página web se presenta un botón con el nombre "Resuelva sus dudas"?	<input checked="" type="radio"/>	<input type="radio"/>
¿Al pulsar sobre el botón se despliega el chatbot en la pantalla?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot se visualiza correctamente en la laptop?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H02 *

	Si	No
¿El chatbot muestra un mensaje de presentación?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot menciona a que temas puede dar respuesta?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot presenta un menú compuesto por dígitos y nombres sobre temas que puede responder?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H03 *

	Si	No
¿El chatbot muestra un campo de texto para ingresar alguna consulta?	<input checked="" type="radio"/>	<input type="radio"/>
En el campo de texto, ¿el chatbot permite ingresar el valor numérico o el nombre de las opciones del menú?	<input checked="" type="radio"/>	<input type="radio"/>
¿Las opciones del menú funcionan correctamente?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot permite ingresar una pregunta de autoría propia del usuario?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot muestra una respuesta a cada consulta echa por el usuario?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H04 *

	Si	No
¿El chatbot presenta un mensaje cuando no encontró respuesta a una pregunta?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot muestra el uso del comando "Ayuda"?	<input checked="" type="radio"/>	<input type="radio"/>
¿El comando "Ayuda" funciona correctamente?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot presenta un menú de 25 opciones tras usar el comando "Ayuda"?	<input checked="" type="radio"/>	<input type="radio"/>
¿Cada una de las opciones del menú funcionan correctamente?	<input checked="" type="radio"/>	<input type="radio"/>

No se pueden editar las respuestas

La presente encuesta tiene como objetivo recoger información de las pruebas de validación hechas al agente conversacional y ver si cumple con lo establecido en las historias de usuario

Por favor, para realizar las pruebas ingrese al siguiente enlace: <http://147.182.229.85/>

Se ha registrado el correo del encuestado (raquel.espinosa@unl.edu.ec) al enviar este formulario.

* Indica que la pregunta es obligatoria

Nombres y apellidos completos *

Raquel Teresa Espinosa Hurtado

Código de la Historia de Usuario: H01 *

	Si	No
¿Se puede acceder a la página web donde está implementado el chatbot?	<input checked="" type="radio"/>	<input type="radio"/>
¿En la página web se presenta un botón con el nombre "Resuelva sus dudas"?	<input checked="" type="radio"/>	<input type="radio"/>
¿Al pulsar sobre el botón se despliega el chatbot en la pantalla?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot se visualiza correctamente en la laptop?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H02 *

	Si	No
¿El chatbot muestra un mensaje de presentación?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot menciona a que temas puede dar respuesta?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot presenta un menú compuesto por dígitos y nombres sobre temas que puede responder?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H03 *

	Si	No
¿El chatbot muestra un campo de texto para ingresar alguna consulta?	<input checked="" type="radio"/>	<input type="radio"/>
En el campo de texto, ¿el chatbot permite ingresar el valor numérico o el nombre de las opciones del menú?	<input checked="" type="radio"/>	<input type="radio"/>
¿Las opciones del menú funcionan correctamente?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot permite ingresar una pregunta de autoría propia del usuario?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot muestra una respuesta a cada consulta echa por el usuario?	<input checked="" type="radio"/>	<input type="radio"/>

Código de la Historia de Usuario: H04 *

	Si	No
¿El chatbot presenta un mensaje cuando no encontró respuesta a una pregunta?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot muestra el uso del comando "Ayuda"?	<input checked="" type="radio"/>	<input type="radio"/>
¿El comando "Ayuda" funciona correctamente?	<input checked="" type="radio"/>	<input type="radio"/>
¿El chatbot presenta un menú de 25 opciones tras usar el comando "Ayuda"?	<input checked="" type="radio"/>	<input type="radio"/>
¿Cada una de las opciones del menú funcionan correctamente?	<input checked="" type="radio"/>	<input type="radio"/>

Anexo 7. Ejecución de las pruebas con los métodos GET y POST en la herramienta ApacheJMeter.

En este documento se presenta el proceso de ejecución de las pruebas de rendimiento, carga y estrés para cierta cantidad de usuarios (grupo de hilos), aplicados en la herramienta ApacheJMeter.

Pruebas de carga para el método GET

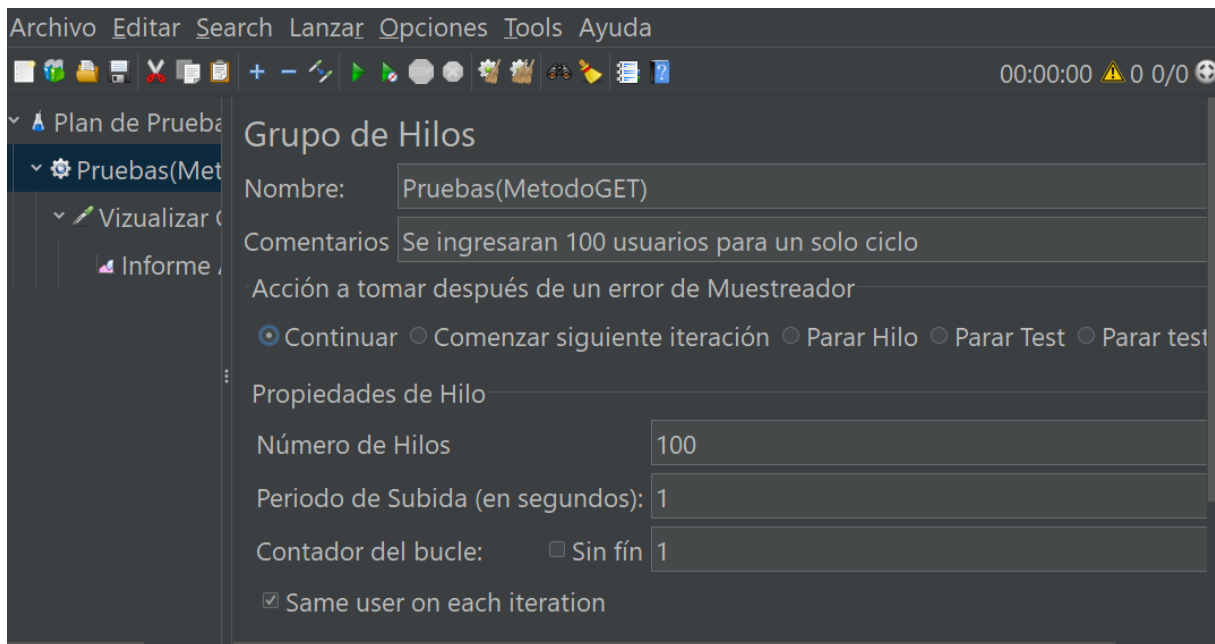


Figura 47. Ingreso de 100 usuarios para 1 iteración con el método GET.

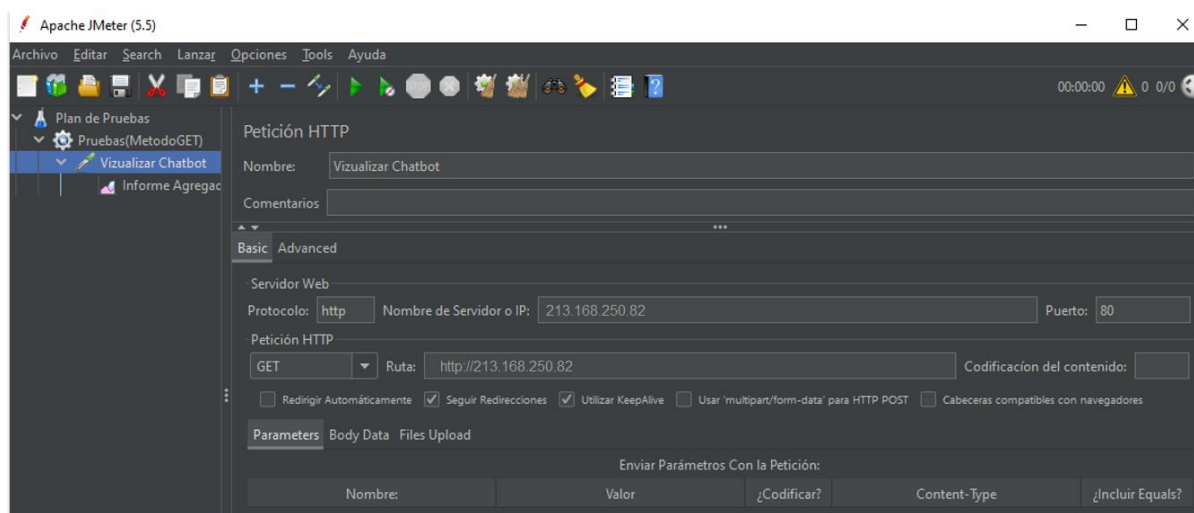


Figura 48. Ruta y puerto establecido para las pruebas de carga para el método GET.

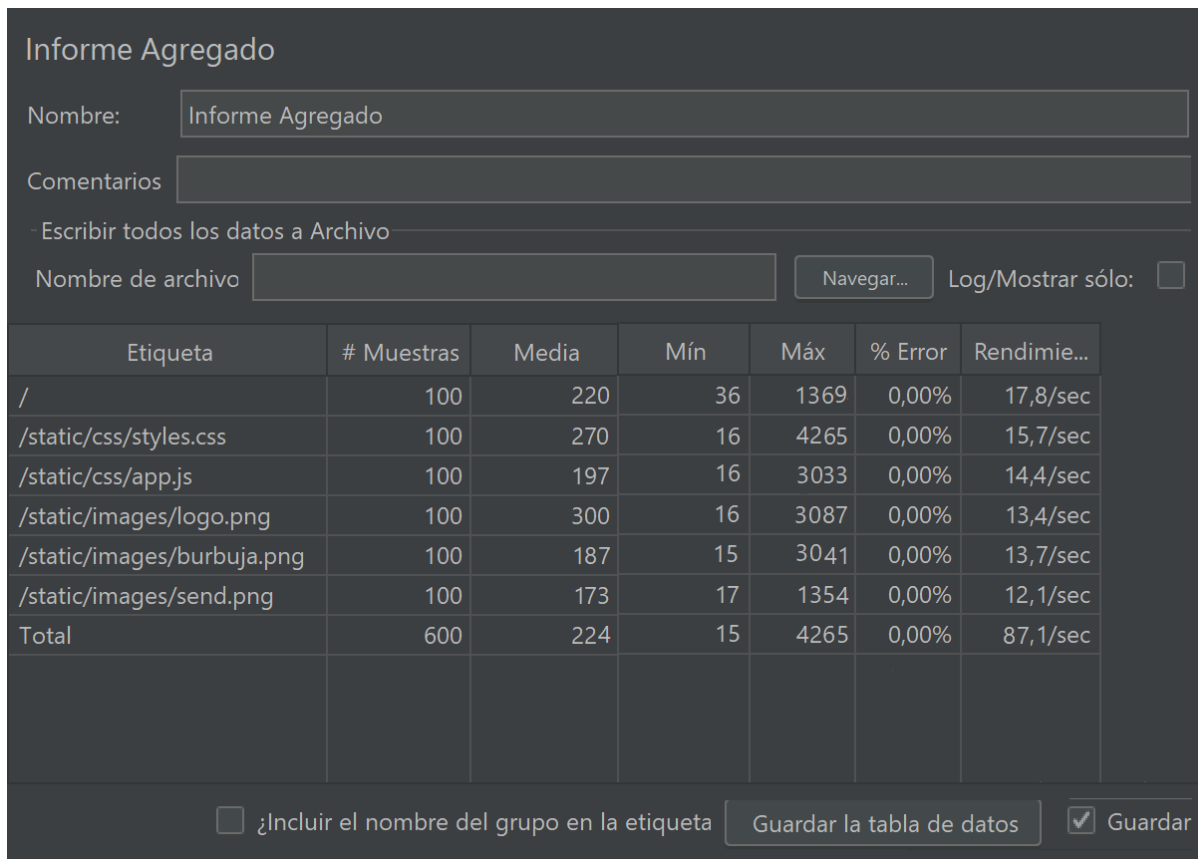


Figura 49. Reporte de las pruebas de carga para el método GET.

Pruebas de estrés para el método GET

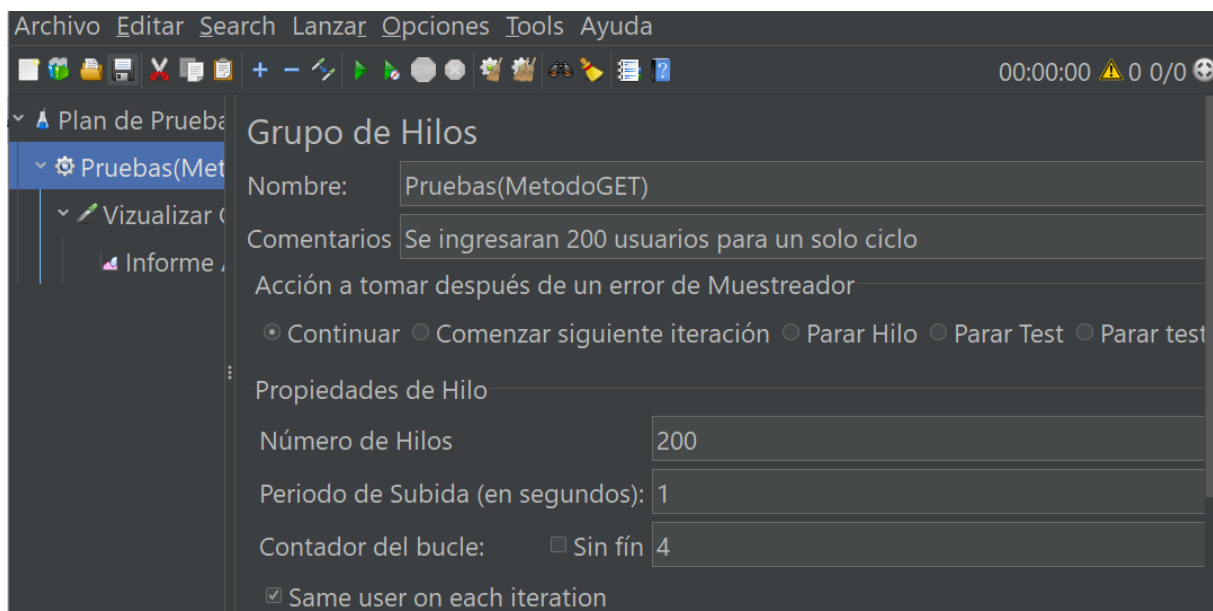


Figura 50. Ingreso de 200 usuarios para 4 iteración con el método GET.

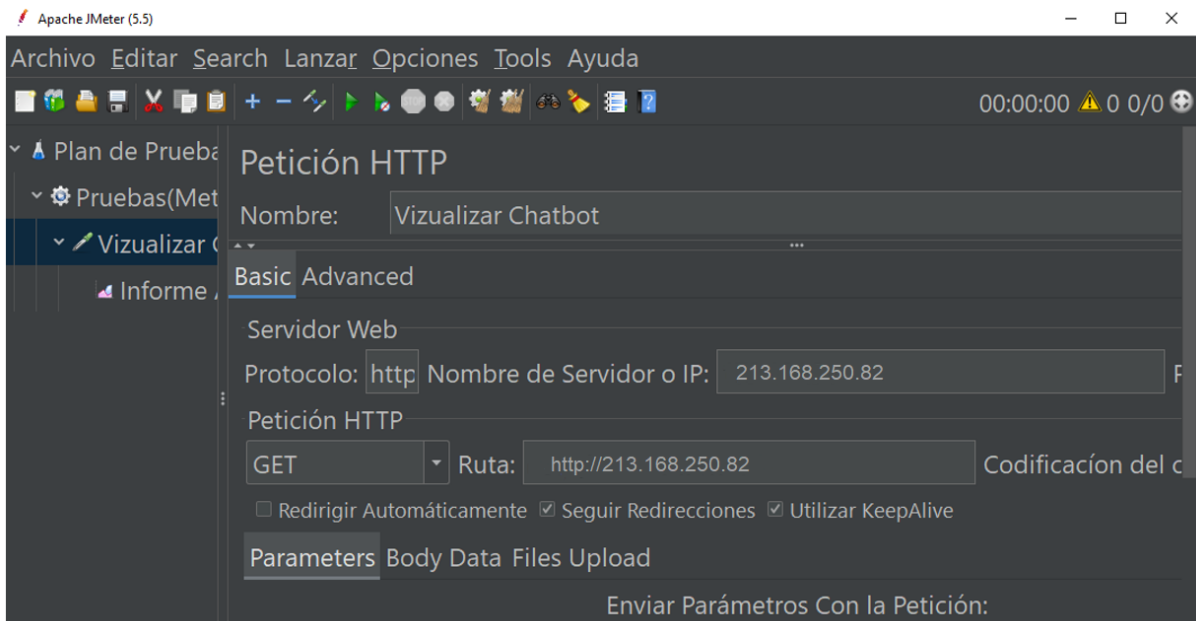


Figura 51. Ruta y puerto establecido para las pruebas de estrés para el método GET.

Informe Agregado

Nombre: Informe Agregado

Comentarios

- Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo:

Etiqueta	# Muestras	Media	Mín	Máx	% Error	Rendimie...
/	800	473	35	15457	0,00%	11,8/sec
/static/css/styles.css	800	198	16	63860	0,13%	13,7/sec
/static/css/app.js	800	227	16	13123	0,00%	14,4/sec
/static/images/logo.png	800	172	15	56218	0,13%	13,4/sec
/static/images/burbuja.png	800	135	14	62330	0,13%	13,9/sec
/static/images/send.png	800	159	15	59825	0,13%	12,1/sec
Total	4800	707	14	63860	0,08%	79,3/sec

¿Incluir el nombre del grupo en la etiqueta Guardar

Figura 52. Reporte de las pruebas de estrés para el método GET.

Pruebas de carga para el método POST

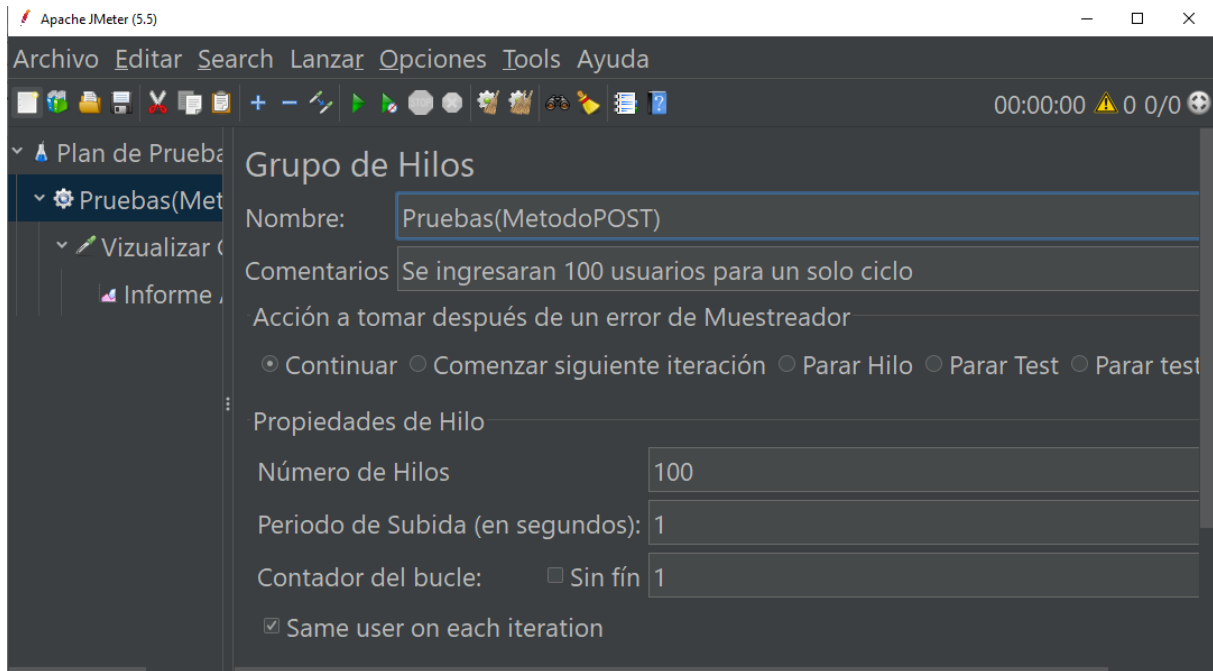


Figura 53. Ingreso de 100 usuarios para 1 iteración con el método POST.

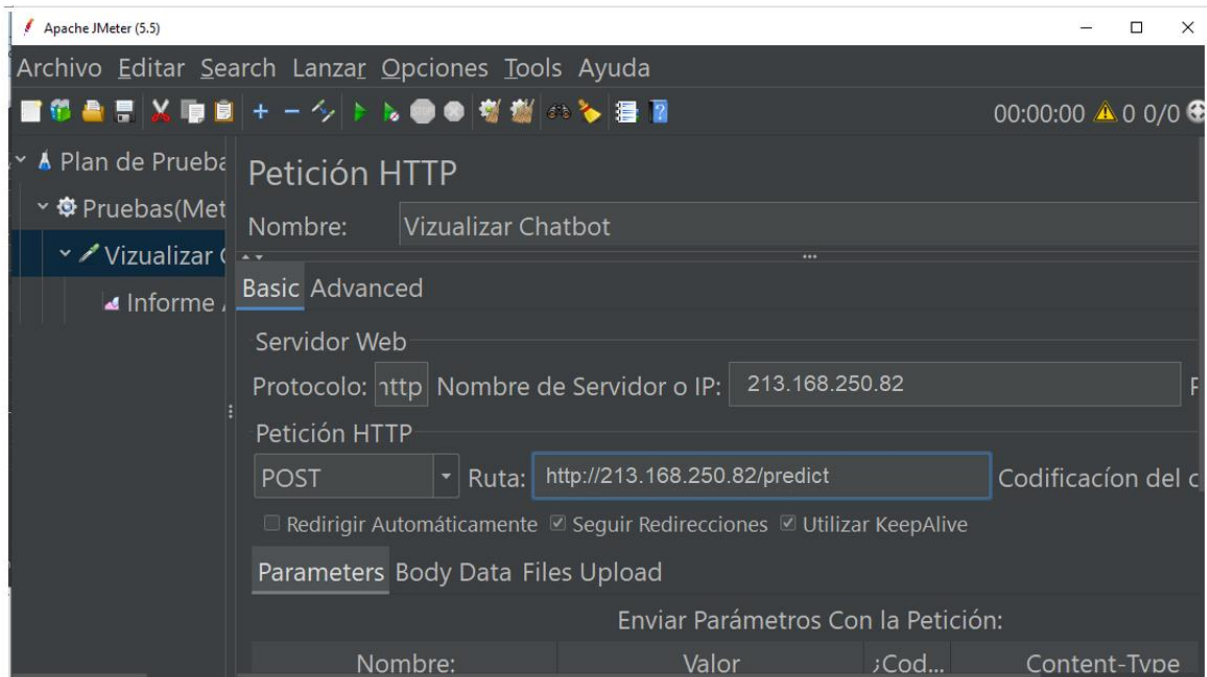


Figura 54. Ruta y puerto establecido para las pruebas de carga para el método POST.

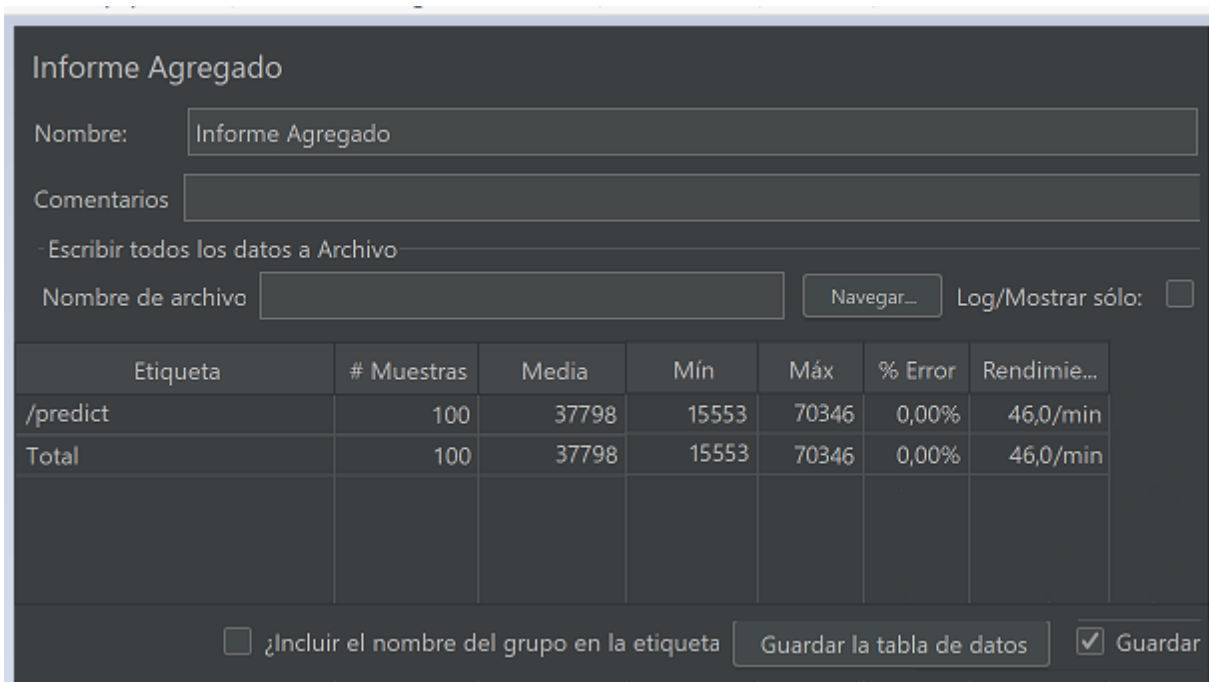


Figura 55. Reporte de las pruebas de carga para el método POST.

Pruebas de estrés para el método POST



Figura 56. Ingreso de 100 usuarios para 4 iteración con el método POST.

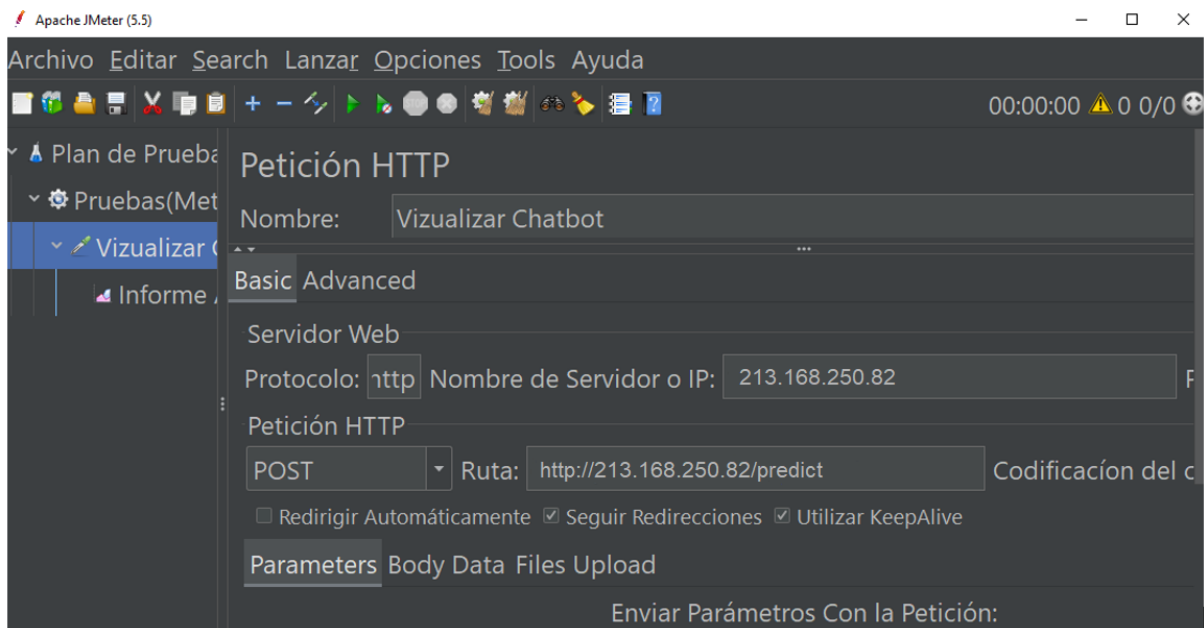


Figura 57. Ruta y puerto establecido para las pruebas de estrés para el método POST.

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo:

Etiqueta	# Muestras	Media	Mín	Máx	% Error	Rendimie...
/predict	400	4374	12587	81366	0,78%	45,0/min
Total	400	4374	12587	81366	0,78%	45,0/min

¿Incluir el nombre del grupo en la etiqueta Guardar

Figura 5. Reporte de las pruebas de estrés para el método POST.

Anexo 8. Encuesta aplicada a la secretaria Nataly Mendoza, para conocer el promedio de interesados que realizan preguntas sobre los programas de posgrados.

Encuesta orientada a conocer la cantidad de estudiantes que realizan preguntas frecuentes con respecto a los programas de posgrado

Esta encuesta tiene el objetivo de recolectar información del promedio de estudiantes que realizan preguntas sobre los estudios de posgrados, con el fin de simular en la herramienta Apache JMeter una cantidad de usuarios reales y determinar la cantidad de carga que puede soportar un aplicativo web (chatbot) que se desarrolló con TT.

Este formulario registra automáticamente los correos de los usuarios de Universidad Nacional de Loja. [Cambiar configuración](#)

Usuarios que han respondido

Correo electrónico

rnmendozap@unl.edu.ec

Ayúdeme con los nombres y apellidos completos

1 respuesta

RUBIELA NATALY MENDOZA PATIÑO

Especifique el área o departamento en el que labora y el cargo que ocupa

1 respuesta

Dirección de Posgrado, Asistente de apoyo académico y administrativo

¿Cuál es el promedio de preguntas o peticiones que realizan al día, los interesados respecto a los programas de posgrados de la UNL?

1 respuesta

En etapa de postulaciones aproximadamente unas 70 preguntas al día, si no es matrículas aproximadamente unas 15 diarias.

¿Cuál es el promedio de preguntas que hacen los interesados sobre los programas de posgrados de la Carrera de Computación de la UNL?

1 respuesta

En etapa de postulaciones aproximadamente unas 30 preguntas al día si no es etapa de postulaciones, aproximadamente unas 3 a 5 máximo.

¿Qué día o días de la semana se producen más preguntas por parte de los estudiantes sobre los posgrados de la UNL?

1 respuesta

De lunes a viernes.

¿Cuántas preguntas realiza en promedio un estudiante sobre los posgrados durante el proceso de postulación y matriculación?

1 respuesta

Aproximadamente unos 30 o 40 al día.

¿En qué fase del proceso de matriculación a los programas de posgrados de la Carrera de Computación de la UNL se realizan más preguntas?

1 respuesta

Fase de inscripción y etapa de matriculación.

Anexo 9. Encuesta aplicada a estudiantes de último ciclo, egresados y profesionales de la carrera de Sistemas/Computación de la UNL para determinar el grado de satisfacción del agente conversacional.

Chatbot para resolver inquietudes académicas sobre estudios de Posgrado en la Carrera de Sistemas / Computación de la UNL

La presente encuesta tiene como finalidad la recopilación de información acerca del grado de satisfacción de los usuarios al utilizar un Chatbot, orientado a responder preguntas frecuentes sobre los estudios de posgrado en materia de matrículas y homologaciones, y que fue desarrollado utilizando como red neuronal, una versión ligera del modelo BERT.

Para acceder a la página web donde se encuentra alojada la aplicación, debe dar clic en el siguiente enlace <http://147.182.229.85/>

Este formulario recoge automáticamente los correos de todos los encuestados. [Cambiar configuración](#)

Correo electrónico

deiby.calva@unl.edu.ec

cristian.aguirre@unl.edu.ec

andy.camacho@unl.edu.ec

edmundoj.pezantes@unl.edu.ec

dayanna.alvarado@unl.edu.ec

wmmaldonadoa@unl.edu.ec

mario.a.orellana@unl.edu.ec

raquel.espinosa@unl.edu.ec

jose.a.carrion.o@unl.edu.ec

victoria.coronel@unl.edu.ec

david.pacheco@unl.edu.ec

bryan.requenes@unl.edu.ec

jorge.pucha@unl.edu.ec

franz.flores@unl.edu.ec

juan.c.armijos@unl.edu.ec

yamilka.erazo@unl.edu.ec

jean.x.agreda@unl.edu.ec

byron.montano@unl.edu.ec

josselyn.guazha@unl.edu.ec

jonathan.tillaguango@unl.edu.ec

angel.capa@unl.edu.ec

maria.paredes@unl.edu.ec

gilson.quezada@unl.edu.ec

patricio.benitez@unl.edu.ec

franz.flores@unl.edu.ec

juan.c.armijos@unl.edu.ec

yamilka.erazo@unl.edu.ec

jean.x.agreda@unl.edu.ec

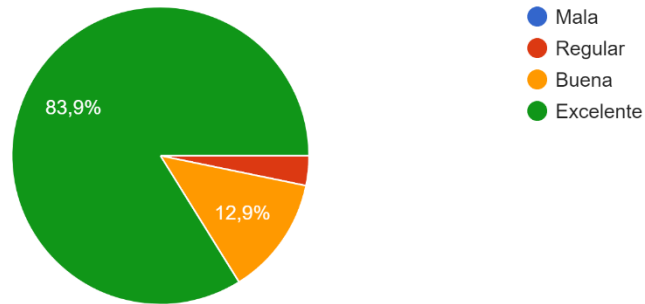
byron.montano@unl.edu.ec

josselyn.guazha@unl.edu.ec

jonathan.tillaguango@unl.edu.ec

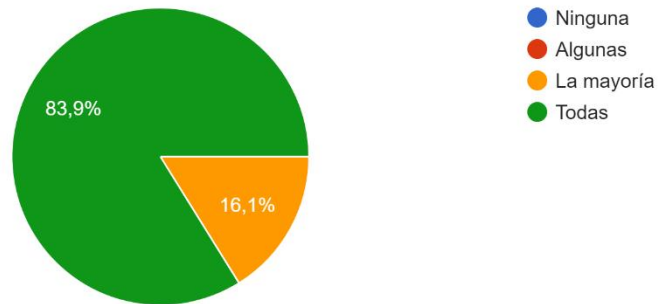
¿Cómo califica la interfaz gráfica del agente conversacional?

31 respuestas



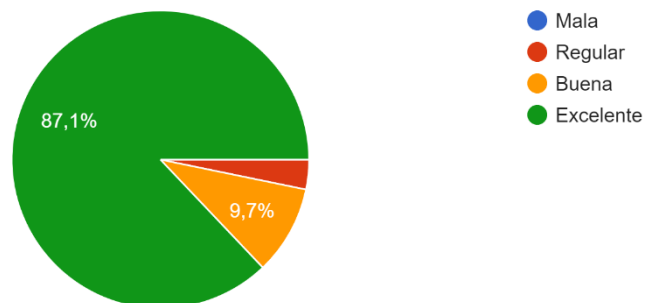
¿Considera que el agente conversacional respondió las pregunta que usted realizó?

31 respuestas



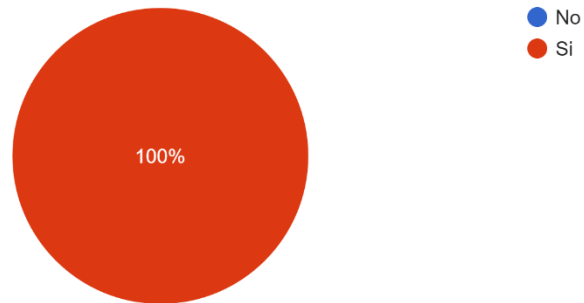
¿Cómo calificaría el tiempo que tarda el agente conversacional en proporcionarle una respuesta a sus preguntas?

31 respuestas



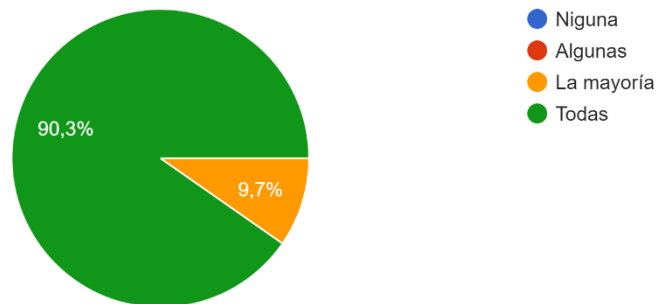
¿Cree que la implementación del agente conversacional resultó útil para responder sus inquietudes?

31 respuestas



¿Considera que el agente conversacional proporcionó respuestas entendibles a sus preguntas?

31 respuestas



Anexo 10. Certificación de traducción

Loja, 23 de julio del 2023

Angie Nicole Valarezo Lima

Licenciada en Ciencias de la Educación Mención Idioma Inglés

CERTIFICO:

Que he realizado la traducción de español a inglés del resumen del Trabajo de Titulación: **Chatbot para resolver inquietudes académicas sobre estudios de posgrado en la Carrera de Sistemas/Computación de la UNL**, de autoría del estudiante **Leonardo Vicente Paredes Rivas**, con cédula N.º **1900553908**, estudiante de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Es todo cuanto puedo certificar en honor a la verdad, facultando al interesado hacer uso del presente en lo que considere conveniente.



Lcda. Angie Nicole Valarezo Lima

C.I: 1150760435

Registro Senescyt: 1008-2021-2368840

Celular: 0989301637

Loja, 23 de julio del 2023

Angie Nicole Valarezo Lima

Licenciada en Ciencias de la Educación Mención Idioma Inglés

CERTIFICO:

Que he realizado la traducción de español a inglés del resumen del Trabajo de Titulación: **Chatbot para resolver inquietudes académicas sobre estudios de posgrado en la Carrera de Sistemas/Computación de la UNL**, de autoría del estudiante Leonardo Vicente Paredes Rivas, con cédula N° 1900553908, estudiante de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Es todo cuanto puedo certificar en honor a la verdad, facultando al interesado hacer uso del presente en lo que considere conveniente.



Firmado electrónicamente por:
ANGIE NICOLE
VALAREZO LIMA

Lcda. Angie Nicole Valarezo Lima

C.I: 1150760435

Registro Senescyt: 1008-2021-2368840

Celular: 0989301637