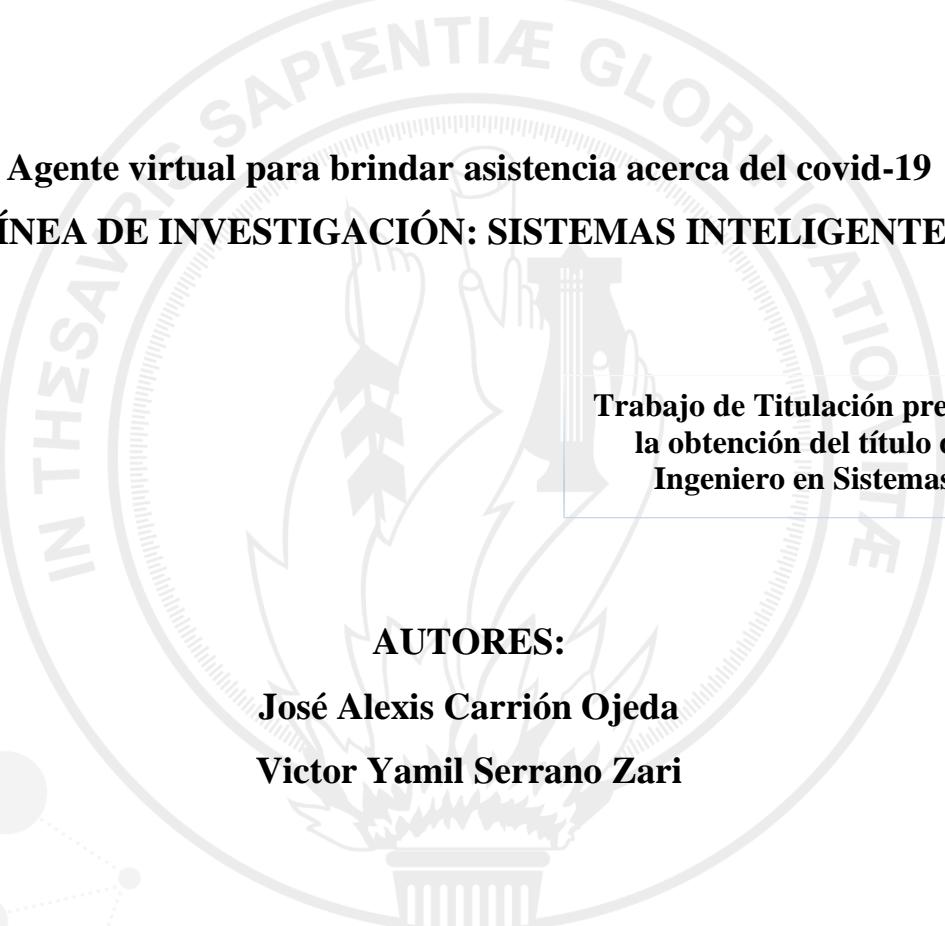




unl

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



Trabajo de Titulación previa a
la obtención del título de
Ingeniero en Sistemas

AUTORES:

José Alexis Carrión Ojeda

Victor Yamil Serrano Zari

DIRECTOR:

Ing. Oscar Miguel Cumbicus Pineda, Mg.Sc.

LOJA – ECUADOR
2022

Certificación

Loja, 24 de marzo del 2022

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

C E R T I F I C O:

Que he revisado y orientado todo el proceso de la elaboración del trabajo de titulación, así como para dar cumplimiento, a las directrices de la guía para la escritura y presentación del informe de Trabajo de Titulación, de conformidad a lo que dispone el Art. 231, inciso segundo del Reglamento de Régimen Académico de la Institución, se cambia el título del Trabajo de titulación de mayúsculas a minúsculas, el que se denomina: **Agente virtual para brindar asistencia acerca del covid-19**, de la autoría de los estudiantes **José Alexis Carrión Nro. de cédula 1803703972** y **Ojeda y Victor Yamil Serrano Zari Nro. de cédula 0750599391**, previa a la obtención del título de **Ingenieros en Sistemas**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación para la respectiva sustentación.

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Nosotros, **José Alexis Carrión Ojeda** y **Victor Yamil Serrano Zari**, declaramos ser autores del presente trabajo de titulación y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestro trabajo de titulación en el Repositorio Digital Institucional - Biblioteca Virtual.

Firma:

Autor: José Alexis Carrión Ojeda

Cédula: 1803703972

Fecha: 20/09/2022

Correo Electrónico:

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

Celular: 0960839300

Firma:

Autor: Victor Yamil Serrano Zari

Cédula: 0750599391

Fecha: 20/09/2022

Correo Electrónico:

victor.serrano@unl.edu.ec

Celular: 0985675436

Carta de autorización del Trabajo de Titulación por parte del autor, para la consulta de producción parcial o total, y publicación electrónica de texto completo

Nosotros, **José Alexis Carrión Ojeda, Víctor Yamil Serrano Zari**, declaramos ser autores del Trabajo de Titulación denominado: **Agente virtual para brindar asistencia acerca del covid-19**, autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional en las redes de información del país y del exterior, con los 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 esta autorización, en la ciudad de Loja, a los veinte días del mes de septiembre del dos mil veintidós.

Firma:

Firma:

Autor: José Alexis Carrión Ojeda

Autor: Victor Yamil Serrano Zari

Cédula: 1803703972

Cédula: 0750599391

Dirección: Loja

Dirección: Zaruma

Correo Electrónico:

Correo Electrónico:

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

victor.serrano@unl.edu.ec

Celular: 0960839300

Celular: 0985675436

DATOS COMPLEMENTARIOS:

Director del trabajo de titulación: Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

Dedicatoria

Dedico el presente trabajo de titulación a mi familia, a mis padres José Carrión y Teresa Ojeda, por su amor, sacrificio, confianza, así como los valores y principios que me han inculcado. A mis hermanos Jorge y María José por su cariño y apoyo incondicional, durante todo este proceso, por hacer más llevadera esta gran etapa, han sido mi soporte en los momentos difíciles.

A mis abuelos Rafael, Rosario, Vicente e Isabel María, son el tesoro más preciado de mi infancia que aún hoy conservo en mi corazón por sus sabias enseñanzas y cuidados, por enseñarme lo que es la rectitud, el esfuerzo y el gran ejemplo de vida que forjaron, no me alcanzará la vida para agradecer todo lo que han hecho por mí. De igual manera a mis tíos y primos por su apoyo, consejos y tiempo compartido. A ella, mi compañera incondicional María José S. que me ha brindado su apoyo y compañía en los buenos y malos momentos, sin dudar de mí, gracias por ser parte de esta meta. Finalmente, a mis compañeros con lo que inicié, me formé y aprendí de ellos, en especial a Leonardo, Prado, Cristian, Jorge y Yamil quienes sin esperar nada a cambio compartieron su conocimiento, tiempo y grandes momentos. A todas aquellas personas que durante estos cinco años estuvieron a mi lado apoyándome y lograron que este sueño se haga realidad.

José Alexis Carrión Ojeda

Dedico el presente trabajo de titulación a mis padres Victor Serrano y Gloria Zari, por su consejo, amor y esfuerzo, gracias a lo cual me ha permitido culminar mis estudios universitarios. A mi hermana Mabel Serrano, que siempre me ha apoyado incondicionalmente durante todas las etapas de este proceso universitario.

A mis amigos, que de una u otra manera me han apoyado a lo largo de todo este tiempo, y han hecho de esta etapa universitaria una parte de mi vida que me traerá grandes recuerdos. A todas las personas que nos han ayudado ya sea brindando su apoyo moral o compartiendo su conocimiento para que así culminemos satisfactoriamente nuestros estudios.

Víctor Yamil Serrano Zari

Agradecimiento

Agradecemos infinitamente a Dios que brindó salud, sabiduría y la perseverancia necesaria para cumplir con éxito todas las metas propuestas con trabajo y dedicación. A nuestras familias, en especial nuestros padres, quienes han sido un apoyo fundamental a lo largo de la formación académica.

A nuestro director de este trabajo, Oscar Cumbicus Pineda Mg. Sc., con su experiencia y conocimiento, supo orientar de la mejor manera para culminar con éxito, con tiempo y paciencia. Gracias a la Universidad Nacional de Loja, así como a todos los docentes y compañeros involucrados en la carrera de Ingeniería en Sistemas por todo el apoyo y conocimiento que generosamente nos brindaron.

**José Alexis Carrión Ojeda
Víctor Yamil Serrano Zari**

Índice de contenidos

Portada	i
Certificación.....	ii
Autoría.....	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento.....	vi
Índice de contenidos	vii
Índice de figuras	xi
Índice de tablas	xi
Índice de anexos.....	xv
1. Título.....	1
2. Resumen	2
2.1 Abstract	3
3. Introducción.....	4
4. Marco Teórico	6
4.1. Conceptos preliminares	6
4.1.1. Agente virtual.....	6
4.1.2. Inteligencia Artificial	6
4.1.3. Procesamiento de Lenguaje Natural (PLN).....	6
4.1.4. BERT (Representación de Codificador Bidireccional de Transformadores).....	6
4.1.5. SQuAD (Stanford Question Answering Dataset)	7
4.1.6. Arquitectura del modelo Bert:.....	7
4.1.7. Tipos de Chatbot:	25
4.1.8. Pytorch	25
4.1.9. Transformers	26
4.1.10. BM25	26
4.1.11. Capas de Autoatención.....	26
4.1.12. GoogleTranslation	26

4.1.13. Docker	27
4.1.14. Haystack.....	27
4.1.15. Retriever - Haystack	27
4.1.16. Reader- Haystack.....	27
4.1.17. FARM	27
4.1.18. Elasticserch	28
4.1.19. Streamlit	28
4.1.20. Microsoft Azure.....	28
4.1.21. Azure para estudiantes	28
4.1.22. FastAPI.....	28
4.1.23. Fine Tuning.....	28
4.1.24. Google Colaboratory.....	29
4.1.25. Semantic Scholar.....	29
4.1.26. Google Cloud.....	29
4.1.27. Coronavirus	29
4.1.28. SARS-CoV-2	30
4.2. Trabajos relacionados	30
5. Metodología	33
5.1. Tipo de investigación	33
5.2. Métodos de investigación	33
5.2.1. Método científico.....	33
5.2.2. Metodología Desing Thinking.....	34
5.3. Técnicas e instrumentos para la recolección de información	35
5.3.1. Recopilación documental.....	35
5.3.2. Instrumentos de recolección de datos.....	36
5.3.2.1. Encuesta.....	36
5.3.3. Tratamiento de la información: Metodología KDT	36
5.3.3.1. Entender el dominio de la aplicación y los objetivos del proceso KDT:	36
5.3.3.2. Adquisición o selección de un conjunto de datos objetivo:	37

5.3.3.3. Limpieza, preprocesamiento y transformación de los datos:	37
5.3.3.4. Desarrollo del modelo:	37
5.3.3.5. Elección y ejecución de los algoritmos de minería de datos adecuados:	38
5.3.3.6. Interpretación y visualización de los resultados:	38
6. Resultados	39
6.1. Objetivo 1: Definir el modelo de lenguaje BERT junto a SQuAD para generar un modelo que dé respuesta a las preguntas realizadas al agente virtual.	39
6.1.1. Tarea 1: Realizar revisión bibliográfica, basada en las fases de la metodología de RSL de Bárbara Kitchenham.....	39
6.1.2. Tarea 2: Extracción de información de artículos relacionados con las características y el funcionamiento respecto al empleo de BERT y SQuAD....	50
6.1.3. Tarea 3: Selección de artículos destacados respecto a temas tópicos sobre el covid-19.....	50
6.2. Objetivo 2: Desarrollar un agente virtual empleando NLP (Natural Language Processing) con el modelo de lenguaje BERT junto al conjunto de datos SQuAD.	55
6.2.1. Entrenar el modelo de lenguaje BERT, con la información de artículos científicos sobre información general del covid-19.	55
6.2.2. Analizar los resultados obtenidos del modelo entrenado.	68
6.2.3. Diseñar la arquitectura del agente virtual.....	70
6.2.4. Codificar en base al diseño.....	73
6.2.5. Desarrollo del prototipo de una aplicación web para la interacción de los usuarios con el agente virtual.....	77
6.3. Objetivo 3: Evaluación de la herramienta.....	79
6.3.1. Evaluar el agente virtual	80
7. Discusión	86
7.1. Desarrollo de la discusión de los objetivos específicos	86
Objetivo 1: Definir el modelo de lenguaje BERT junto a SQuAD para generar un modelo que dé respuesta a las preguntas realizadas al agente virtual.	86
Objetivo 2: Desarrollar un agente virtual empleando NLP (Natural Language	

Processing) con el modelo de lenguaje BERT junto al conjunto de datos SQuAD	87
Objetivo 3: Evaluación de la herramienta.	89
7.2. Valoración técnica, económica, ambiental y social	90
7.2.1. Valoración técnica	90
7.2.2. Valoración económica.....	90
7.2.3. Valoración ambiental	91
7.2.4. Valoración social.....	91
8. Conclusiones	93
9. Recomendaciones.....	95
9.1. Trabajos futuros.....	96
10. Bibliografía.....	97
11. Anexos	104

Índice de tablas

TABLA I .Estudios primarios seleccionados.....	30
TABLA II. Estudios seleccionados, temas tópicos del covid-19	32
TABLA III. Proceso de revisión sistemática de literatura propuesto [71]	40
TABLA IV. Preguntas de investigación	41
TABLA V. Scripts de búsqueda	42
TABLA VI. Resumen de estudios seleccionados	46
TABLA VII. Preguntas de investigación	51
TABLA VIII. Hiperparámetros Ajuste (Fine Tuning).....	57
TABLA IX. Test de preguntas	69
TABLA X. Test de Preguntas para Evaluación	80
TABLA XI. Evaluación del agente por un especialista en epidemióloga.....	83
TABLA XII. Recursos para talento humano	90
TABLA XIII. Recursos técnicos y tecnológicos	90
TABLA XIV. Recursos para servicios.....	91
TABLA XV. Totalidad de los recursos económicos	91

Índice de figuras

Figura 1. Preentrenamiento y Ajuste del Modelo BERT [9]	7
Figura 2. Tokenización de palabras [9]	8
Figura 3. Diagrama de bloques de una Red Transformer: la secuencia de entrada es procesada por el codificador y posteriormente enviada al decodificador [9].....	10
Figura 4. El embedding de entrada de la Red Transformer [9].....	11
Figura 5. El codificador de posición de la Red Transformer [9]	11
Figura 6. El bloque de codificación de la Red Transformer [9].....	12
Figura 7. Primer codificador en la Red Transformer [9].....	13
Figura 8. Relación entre palabras de la frase ‘I love Italian food’ [9]	13
Figura 9. Obtención de ‘querie’, ‘key’ y ‘value’ [9]	14
Figura 10. Comparación de ‘queries’ y ‘keys’ y obtención de puntajes [9]	14
Figura 11. Empleo de la función ‘softmax’ en la matriz de puntajes [9]	15
Figura 12. Salida del bloque atencional [9]	16
Figura 13. Bloques atencionales para localizar asociaciones entre palabras y grupos de palabras a distintos niveles [9]	17
Figura 14. Elementos del bloque residual [9]	18
Figura 15. Etapa de salida del codificador: una red neuronal seguida de otro bloque residual [9].....	18
Figura 16. Salida resultante del primer codificador [9]	19
Figura 17. Elementos del decodificador [9]	20
Figura 18. Funcionamiento del bloque atencional de enmascaramiento [9]	21
Figura 19. En el enmascaramiento no se presta atención a palabras futuras en la secuencia: ‘comida’ e ‘Italiana’ [9]	22
Figura 20. Resultado del enmascaramiento [9].....	22
Figura 21. El bloque atencional del decodificador toma como entrada la información producida por codificador de la anterior etapa [9]	23
Figura 22. Se requieren múltiples decodificadores para hallar relaciones entre	

palabras y grupos de palabras entre niveles [9].....	24
Figura 23. Capas de salida del bloque de decodificación [9]	25
Figura 24. Búsqueda ACM	43
Figura 25. Búsqueda IEEE	43
Figura 26. Búsqueda Google Scholar	44
Figura 27. Búsqueda Scopus	44
Figura 28. Proceso de selección de artículos	45
Figura 29. Creadores de Dataset CORD-19 [67]	51
Figura 30. Herramienta haystack annotation tool [77].....	56
Figura 31. Ajuste Modelo BERT N°1.....	58
Figura 32. Ajuste Modelo BERT N°2.....	58
Figura 33. Ajuste Modelo BERT N°3.....	58
Figura 34. Ajuste Modelo BERT N°4.....	58
Figura 35. Ajuste Modelo BERT N°5.....	58
Figura 36. Ajuste Modelo BERT N°6.....	59
Figura 37. Ajuste Modelo BERT N°7.....	59
Figura 38. Ajuste Modelo BERT N°8.....	59
Figura 39. Precisión del modelo ajustado	63
Figura 40. Proceso de aprendizaje	64
Figura 41. Proceso de aprendizaje ajustado	65
Figura 42. Artículos publicados por año y cantidad.....	67
Figura 43. Nube de palabras de títulos	67
Figura 44. Términos de Frecuencia en los documentos.....	68
Figura 45. Arquitectura del Agente Conversacional	71
Figura 46. Arquitectura General Modelo BERT.....	73
Figura 47. Contenedores de Docker resultantes.....	75
Figura 48. Aviso sobre la realización de la búsqueda	76

Figura 49. Aviso sobre la no contestación a la pregunta.....	76
Figura 50. Respuesta del agente conversacional	77
Figura 51. Levantamiento del API en Azure.....	78
Figura 52. Levantamiento de la Base de Conocimiento y la interfaz en Google Cloud	79
Figura 53. Prototipo levantado en la web.....	79
Figura 54 Interfaz del agente conversacional	85
Figura 55. Proceso de selección de artículos	116

Índice de anexos

Anexo 1: Revisión Sistemática de Literatura I.....	104
Anexo 2: Revisión Sistemática de Literatura II.....	113
Anexo 3: Cuestionario realizado a docentes.....	121
Anexo 4: Validación de Respuestas a profesionales de la salud	123
Anexo 5: Validación de Respuestas a especialista en epidemiología	130
Anexo 6: Certificado de traducción del resumen.....	137

1. Título

Agente virtual para brindar asistencia acerca del covid-19.

2. Resumen

En Ecuador, la emergencia sanitaria por el COVID-19 provocó colapsos en los centros hospitalarios, específicamente en la ciudad de Loja se atendieron casos graves durante el periodo de cuarentena establecido por el estado ecuatoriano y produjo que casos menores no tengan acceso a la atención médica pública, lo cual afecto a un gran número de personas que no podían costearse un profesional privado, frente a esta problemática el presente Trabajo de Titulación (TT) tiene como objetivo principal brindar asistencia acerca del Covid-19 mediante un agente conversacional (chatbot) para apoyar al sector de la salud en resolver cuestiones comunes que la gente tiene sin la necesidad de consultar en otras fuentes externas que fomentan la desinformación. Se utilizaron las fases de la metodología Desing Thinking. En la creación del modelo se utilizó la metodología Knowledge Discovery in Text (KDT) ya que las respuestas se obtuvieron de documentos extraídos de una base de datos, los cuales se procesaron en el lenguaje Python, gracias a librerías como Transformers de HuggingFace para tokenizar los textos, Pytorch para ejecutar y ajustar el modelo BERT, Haystack para extraer el contexto relacionado de las búsquedas, JSON para almacenar y manipular la información, Googletrans para la traducción de la información, el diseño de la interfaz se desarrolló en Streamlit para la interacción de los usuarios con el agente y el levantamiento de servicios mediante contenedores en Azure y Google Cloud. La información se la extrajo de artículos científicos empleando en el modelo preentrenado BERT en el conjunto de datos de SQuAD 2.0 para ajustar el modelo (fine-tune), el rendimiento del modelo se evaluó con el score o puntuación que se produce en cada una de las respuestas, obteniendo un modelo optimó en temas relacionados con el Covid-19. Se concluyó que se puede diseñar un agente virtual funcional con el empleo del modelo BERT junto a SQuAD, ya que mediante la evaluación realizada por profesionales de la salud corroboraron que las respuestas proporcionadas por el agente fueron apropiadas.

Palabras claves: Covid-19, agente virtual, chatbot, BERT, SQuAD, transformers.

2.1 Abstract

In Ecuador, the health emergency caused by COVID-19 caused collapses in hospital centers, specifically in Loja city, severe cases were treated during the quarantine period established by the Ecuadorian state and resulted in minor cases not having access to public medical care, which affected a large number of people who could not afford a private professional, faced with this problem, the main objective of this Degree Project (TT) is to provide assistance about Covid-19 through a conversational agent (chatbot) to support the health sector in resolving common issues that people have without the need to consult other external sources that promote misinformation. The phases of the Desing Thinking methodology were used. The Knowledge Discovery in Text (KDT) methodology was used in the creation of the model, since the answers were obtained from documents extracted from a database, which were processed in the Python language, thanks to libraries such as HuggingFace Transformers to tokenize the texts, Pytorch to execute and adjust the BERT model, Haystack to extract the related context from the searches, JSON to store and manipulate the information, Googletrans to translate the information, the interface design was developed in Streamlit for the interaction of the users with the agent and the lifting of services through containers in Azure and Google Cloud. The information was extracted from scientific articles using the pre-trained BERT model in the SQuAD 2.0 dataset to adjust the model (fine-tune), the performance of the model was evaluated with the score or score produced in each of the answers, obtaining an optimized model in topics related to Covid-19. It was concluded that a functional virtual agent can be designed with the use of the BERT model together with SQuAD, since the evaluation carried out by health professionals corroborated that the responses provided by the agent were appropriate.

Keywords: Covid-19, virtual agent, chatbot, BERT, SQuAD, transformers.

3. Introducción

En la actualidad con la pandemia del coronavirus que ha venido para quedarse, ha surgido información de todo tipo referente a Covid-19, de la cual no se puede constatar su origen ni su veracidad, lo que se puede considerar desinformación, y a su vez genera aglomeraciones en los centros de salud por parte de personas que buscan atención para resolver sus dudas con un profesional. En consecuencia, para abordar el inconveniente de combatir dudas y desinformación que surgen en la población sobre el Covid-19 [11], se plantea el siguiente problema de investigación ¿La creación de un agente virtual utilizando el modelo de lenguaje BERT (Bidirectional Encoder Representations from Transformers) [6] junto a SQuAD (Stanford Question Answering Dataset) [8] permitirá responder correctamente a preguntas tópicas acerca del Covid-19?, para dicho problema se planteó revisiones sistemáticas de literatura sobre el funcionamiento del modelo BERT y SQuAD, y de cuales temas fueron los más buscados sobre el Covid-19 en la web. Por consiguiente, se llegó a una solución viable, la cual sería un agente virtual [1], este es un sistema de software que interactúa con un usuario para resolver las dudas que este tenga de un tema en específico.

El acceso al Agente virtual permitiría a la sociedad en general solventar sus dudas frente al Covid-19, mediante información extraída de artículos científicos de bases de datos clínicas reconocidas como: ClinicalKey, BMJ, NEJM, PubMed, etc. Un Agente virtual emplea una gran cantidad de datos los cuales se obtiene de las diferentes bases de datos científicas, en nuestro caso el conjunto de datos de investigación abierta CORD-19, este proceso requiere de mucho tiempo y esfuerzo debido a la cantidad de información que necesita procesar el agente, para ofrecer una respuesta coherente y confiable en base a las interrogantes que se le planteen, surgieron limitaciones en el desarrollo del agente virtual como el procesamiento masivo de los datos que el hardware no podía procesar por la gran cantidad de documentos científicos que existen actualmente, (80 GB), por lo que se elaboró un conjunto de datos de un tamaño menor analizado y filtrado para eliminar documentos no relevantes, permitiendo un procesamiento más adecuado para las características del hardware disponible. De este modo las respuestas proporcionadas por el Agente virtual es información verídica, que puede ayudar a las personas sin necesidad de acudir a un profesional de la salud, a cualquier hora y minimizando los costes que conllevan una consulta médica, por muy leve o grave que sea. Para la validación del agente se contrastó las respuestas producidas frente a profesionales de la salud que se desempeñaron en la pandemia. El desarrollo del agente conversacional permitió aplicar los conocimientos adquiridos durante el periodo académico, tanto a nivel de inteligencia artificial, programación y diseño entre otros, ofreciendo soluciones a la sociedad y al campo ingenieril, fomentando el uso de las TIC en el campo de la salud.

Los siguientes apartados constituyen el presente trabajo de titulación:

- Revisión de literatura: sección que contiene la bibliografía primordial para la comprensión del trabajo de titulación.
- Materiales y métodos: se componen del tipo de investigación, la cual se trata de una investigación cuantitativa con un diseño experimental, de los métodos de investigación como el método científico y la metodología Design Thinking, los cuales se aprovecharon para llevar a cabo el desarrollo del trabajo de titulación.
- Resultados: abarca todo el proceso del trabajo de titulación, compuesto por tareas con sus referentes evidencias que conforman cada uno de los siguientes tres objetivos:
 - Objetivo 1: Definir el modelo de lenguaje BERT junto a SQuAD para generar un modelo que dé respuesta a las preguntas realizadas al agente virtual, se constató el modelo que mejor adaptabilidad posee frente a tareas de PNL, junto a sus características y funcionamiento, además se indagó sobre los temas tópicos que se buscan sobre el Covid-19, con el fin de tener una noción de las dudas de la población y poder adaptarlas posteriormente al modelo ajustado para ofrecer respuestas acordes.
 - Objetivo 2: Desarrollar un agente virtual empleando NLP (Natural Language Processing) con el modelo de lenguaje BERT junto al conjunto de datos SQuAD, el modelo BERT fue ajustado con los documentos científicos obtenidos en las revisiones de literatura, previa preparación del conjunto de datos y análisis exploratorio, llegando a un porcentaje del 94.5% de acierto, finalizado el ajuste se diseñó la arquitectura del agente y se codificó todo el sistema. Para el desarrollo del prototipo web, se usó las plataformas de Microsoft Azure [26] y Google Cloud [31] mediante contenedores Docker [18] divididos en módulos.
 - Objetivo 3: Evaluación de la herramienta, se aplicó la evaluación humana (human evaluation), exponiendo las respuestas generadas por el modelo a profesionales de la salud, ya que participaron activamente en la etapa más dura de la pandemia y la experiencia generada en primera línea es muy valiosa. Además, se constató posteriormente las respuestas con una especialista en neumología la cual corroboró las respuestas de forma satisfactoria.

4. Marco Teórico

A continuación, se presentan las bases teóricas que sustentan el presente Trabajo de Titulación (TT), dicha información ha sido obtenida a través de un proceso de revisión bibliográfica junto a búsquedas particulares para temas que no abarcó la revisión literaria. Esta sección inicia con las nociones más relevantes entorno a la temática, y posteriormente, se presentan los estudios relacionados.

4.1. Conceptos preliminares

4.1.1. Agente virtual

Un chatbot o agente conversacional es un sistema de software que puede interactuar con un usuario mediante el uso del lenguaje natural. No se encuentra únicamente ligado a mensajes de texto, si no a una serie de contenido multimedia, lo cual permite una mejor interacción con el usuario. Para una mejor interpretación del lenguaje natural un chatbot, en la actualidad, utiliza técnicas de procesamiento del lenguaje natural. Mediante el uso de computación cognitiva, a través del uso de inteligencia artificial un chatbot entiende lo que el usuario está intentando decir y responde con un mensaje coherente, relevante y directo relacionado con la tarea o petición que el usuario está solicitando. Un chatbot por la forma de interacción con el usuario viene a ser un nuevo tipo de interfaz de usuario que ingresa preguntas para establecer una comunicación y solventar dudas respecto a un tema particular [1].

4.1.2. Inteligencia Artificial

Se trata del campo de estudio que describe la capacidad del aprendizaje automático como los humanos y la capacidad de responder a ciertos comportamientos también conocidos como (A.I.). La necesidad de la Inteligencia Artificial está aumentando cada día. Desde que AI se introdujo por primera vez en el mercado, ha sido la razón del rápido cambio en los campos de la tecnología y los negocios. Los científicos informáticos predicen que para 2020, el 85% de las interacciones con los clientes se gestionarán sin un humano [2].

4.1.3. Procesamiento de Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una subdisciplina de la inteligencia artificial y rama de la lingüística computacional que hace uso de un conjunto de mecanismos que la computadora asimila en un lenguaje de programación definido (formal) y le permite comunicarse con el ser humano en su propio lenguaje [3].

4.1.4. BERT (Representación de Codificador Bidireccional de Transformadores)

BERT es un modelo de representación de lenguaje, que significa Representaciones de codificador bidireccional de Transformers. Está diseñado para entrenar representaciones bidireccionales profundas a partir de texto sin etiquetar a través de una capa contextual, que

se compone de capas de atención apiladas y redes de retroalimentación con incorporaciones de entrada y salida de la secuencia [4]. La representación del texto se integra dentro de las primeras capas (capas de representación), y las capas de salida se pueden cambiar para ofrecer la posibilidad de resolver una amplia variedad de tareas. La entrada es el texto representado como una secuencia de "tokens" (caracteres o serie de caracteres). Las capas de representación se entranan previamente en datos de texto sin etiquetar para aprender representaciones genéricas y contextualizadas de cada elemento de la secuencia de entrada. Luego, dada una tarea objetivo, las capas de salida se personalizan y todo el modelo se ajusta con precisión a los datos de destino [5].

4.1.5. SQuAD (Stanford Question Answering Dataset)

SQuAD 1.0: Es un conjunto de datos de comprensión de lectura, que consta de triples de pasaje-pregunta - respuesta. El conjunto de datos de SQuAD tiene más de 100.000 preguntas formuladas por trabajadores de multitudes en un conjunto de artículos de Wikipedia, donde la respuesta a cada pregunta es un segmento de texto del pasaje de lectura correspondiente [6]. SQuAD 2.0: Combina los datos de SQuAD 1.0 existentes con más de 50.000 preguntas sin respuesta escritas de forma contradictoria por los trabajadores de la plataforma para que parezcan similares a las que se pueden responder. Para tener un buen desempeño en SQuAD 2.0, los sistemas no solo deben responder preguntas cuando sea posible, sino también determinar cuándo el párrafo no respalda ninguna respuesta y abstenerse de responder [7].

4.1.6. Arquitectura del modelo Bert:

BERT aprovecha un enfoque basado en el ajuste fino para aplicar modelos de lenguaje previamente entrenados; es decir, se entrena una arquitectura común para una tarea relativamente genérica y, a continuación, se ajusta en tareas específicas posteriores que son más o menos similares a la tarea de preentrenamiento como se aprecia en la Figura 1.

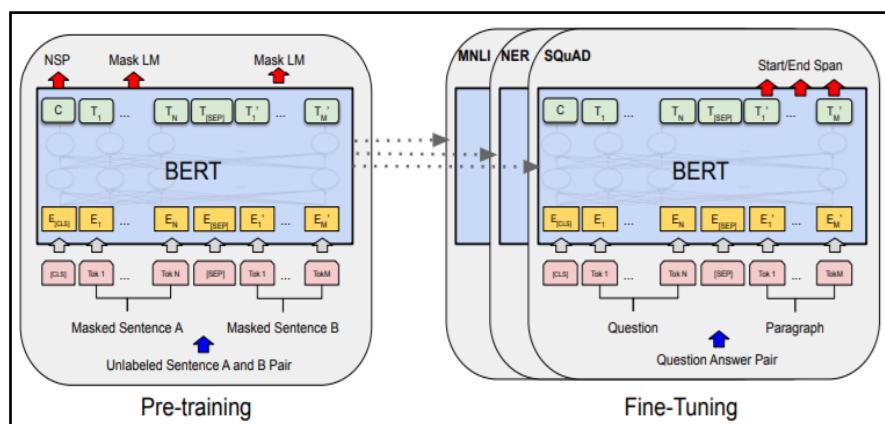


Figura 1. Preentrenamiento y Ajuste del Modelo BERT [9]

Los procedimientos generales de preentrenamiento y ajuste fino para BERT, aparte de las capas de salida, se utilizan las mismas arquitecturas tanto en el preentrenamiento como en el ajuste fino. Los mismos parámetros del modelo preentrenado se utilizan para inicializar los modelos para las diferentes tareas de flujo descendente. Durante el ajuste fino, se ajustan todos los parámetros. [CLS] es un símbolo especial es un símbolo especial que se añade delante de cada ejemplo de entrada, y [SEP] es un símbolo especial de separación (por ejemplo, para separar preguntas y respuestas).

Para la representación de la entrada BERT hace uso de WordPiece embeddings que tiene un vocabulario de 30000 tokens. El primer token de cada secuencia debe ser un token especial de clasificación ([CLS]). Para las parejas de oraciones, se compactan en un paquete para convertirlas en una única secuencia y son separadas con el token especial ([SEP]), posteriormente se averiguan los embeddings. Para cada token su representación de entrada se consigue sumando los correspondientes embeddings de token, segmento y de posición, tal como se muestra en la Figura 2 [8].

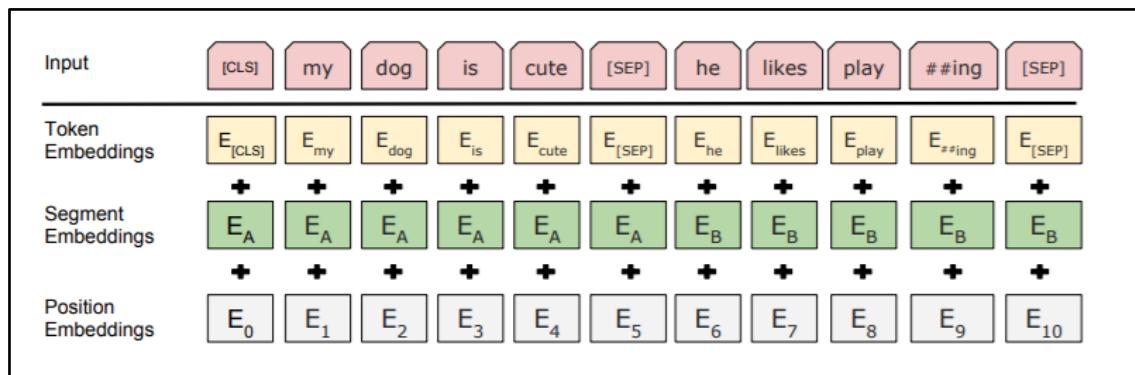


Figura 2. Tokenización de palabras [9]

Representaciones de entrada / salidas utilizadas en BERT.

- **[CLS]:** Denominado token de clasificación. Se utiliza al principio de una secuencia.
- **[SEP]:** Indica la separación de 2 secuencias, actúa como delimitador.
- **[MÁSCARA]:** se utiliza para indicar token enmascarado en la tarea MLM.
- Las incrustaciones de segmento se utilizan para indicar la secuencia a la que pertenece un token, es decir, si hay varias secuencias separadas por un token [SEP] en la entrada, luego, junto con las incrustaciones posicionales, se agregan a las incrustaciones de Word originales para el Modelo para identificar la secuencia del token.
- Los tokens alimentados al modelo BERT se tokenizan utilizando incrustaciones de WordPiece.

En el proceso de preentrenamiento de BERT se utilizan dos tareas no supervisadas distintas: "Masked LM" y "Next Sentence Prediction (NSP)".

Masked LM: Un porcentaje del input es enmascarado de manera aleatoria (un 15% de cada secuencia), posteriormente se deben predecir esos tokens enmascarados. Como consecuencia se obtiene un modelo bidireccional pre entrenado, sin embargo, se genera una asimetría entre el proceso de preentrenamiento y el de ajuste puesto que el token [MASK] que oculta esos tokens de entrada, no va a aparecer en el proceso de ajuste fino. Para mitigar este efecto, en el proceso de preentrenamiento, el generador de datos de entrenamiento oculta con un [MASK] los tokens en el 80% de los casos en los que se trata de utilizar [MASK], en otro 10% se utiliza un token aleatorio y en el diez por ciento restantes, no se modifica el i-ésimo token. Finalmente, el input token es utilizado para predecir el valor del token original a través de la pérdida de entropía cruzada [8].

Funcionamiento de la Red Transformer

La Red Transformer elimina el problema del gradiente de fuga (para relacionar palabras en un sentido bidireccional y que tengan un sentido sintáctico) empleando el mecanismo llamado “atención” (proceso para dar un valor de “importancia” a cada palabra) el cual está compuesto por un codificador y decodificador. Se trata de un mecanismo que permite al modelo mirar otras palabras en la oración durante la codificación, esto es exactamente lo que el mecanismo de atención de la arquitectura Transformer permite hacer a estos modelos.

La bidireccionalidad al ser una característica clave de BERT, por la “B” de su nombre que indica “Bidireccional”. Donde el mecanismo de atención de la arquitectura Transformer permite que modelos como BERT procesen texto bidireccionalmente al:

- Permitir el procesamiento en paralelo: puede procesar texto en paralelo, es decir pueden acceder a cualquier palabra de las oraciones que procesa.
- Almacenar la posición de la entrada: codifica la posición de la palabra en la incrustación y permite identificar palabras o secuencias de texto, lo que se puede extender a grandes volúmenes con distintas longitudes y saber exactamente el orden donde aparece la palabra en la oración.
- Facilitar la búsqueda: para los modelos basados en Transformers pueden buscar cualquier palabra en la oración en cualquier momento, posee una vista de todas las palabras de la secuencia en cada paso de la capa de atención, lo que le otorga la facilidad de observar o buscar hasta desde el final o inicio de las oraciones procesadas.

Como resultado de estos factores, al poder procesar texto en paralelo, incrustar la posición de la entrada en la incrustación y permitir una fácil búsqueda de cada entrada, el modelo BERT puede “leer” el texto de forma bidireccional. Aunque, técnicamente, no es bidireccional, ya que estos modelos realmente miran todo el texto a la vez, por lo que no es direccional. Pero es mejor entenderlo como una forma de intentar procesar el texto de forma bidireccional para mejorar la capacidad del modelo para aprender de la entrada. Al

procesar de esta manera el texto se producen problemas que el modelo los aborda mediante el “enmascaramiento”.

En estas redes la secuencia de entrada es procesada en paralelo por la red, a diferencia de las Redes Recurrentes en donde se procesan de forma serial los elementos de la secuencia Figura 3.

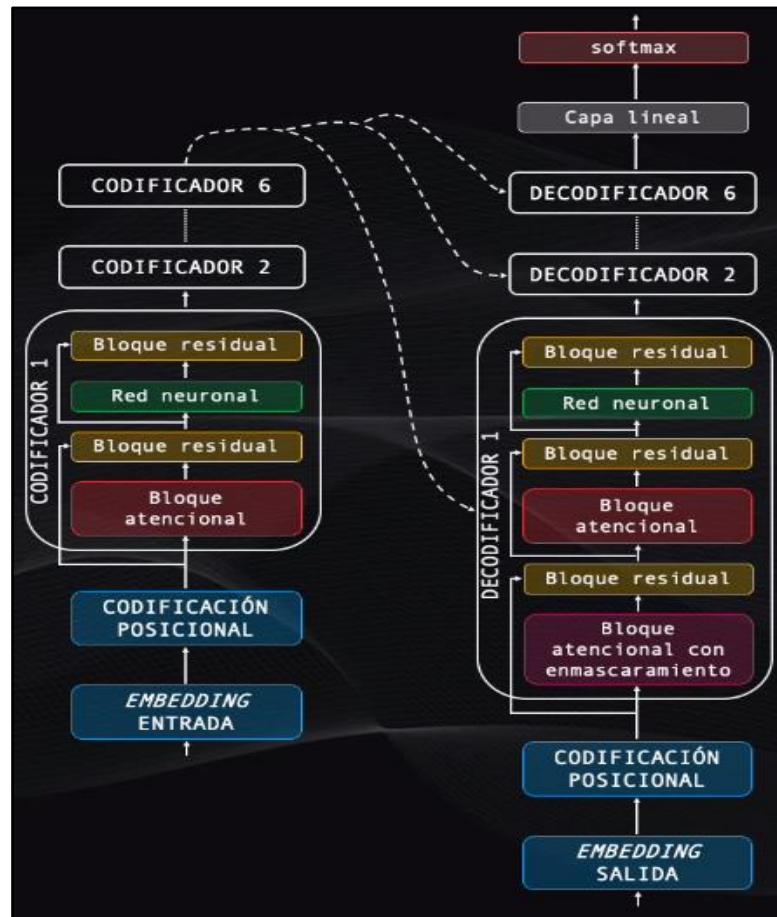


Figura 3. Diagrama de bloques de una Red Transformer: la secuencia de entrada es procesada por el codificador y posteriormente enviada al decodificador [9]

La secuencia se convierte inicialmente a una representación numérica mediante un embedding, luego se agrega una codificación posicional y el vector resultante ingresa a la etapa de codificación, que se encarga de obtener la información más relevante de la secuencia, la salida de esta etapa se conecta a un decodificador, que utiliza esta información para generar secuencialmente texto

Aunque se compone de varios elementos funciona de la siguiente forma:

- El embedding de entrada: el bloque embedding Figura 4, es un algoritmo que convierte el texto en una serie de vectores, o tokens, es decir en una representación numérica que puede ser “comprendida” por la red [12].



Figura 4. El embedding de entrada de la Red Transformer [9]

- El codificador de posición: indica a la red el orden en el que se encuentran las palabras dentro del texto ya que la secuencia se procesa en paralelo Figura 5, genera una serie de vectores que se sumarán a los tokens, y que indican la posición relativa de cada token dentro de la secuencia. Para esto se usan funciones senoidales para las posiciones pares, y cosenoidales para las impares, con lo que cada vector generado tendrá un patrón numérico único con la información de la posición [12].

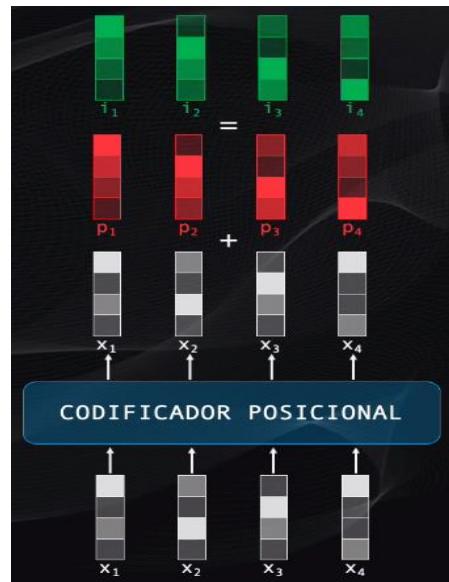


Figura 5. El codificador de posición de la Red Transformer [9]

- Codificación y el bloque atencional: contiene seis codificadores, todos con una estructura idéntica Figura 6.

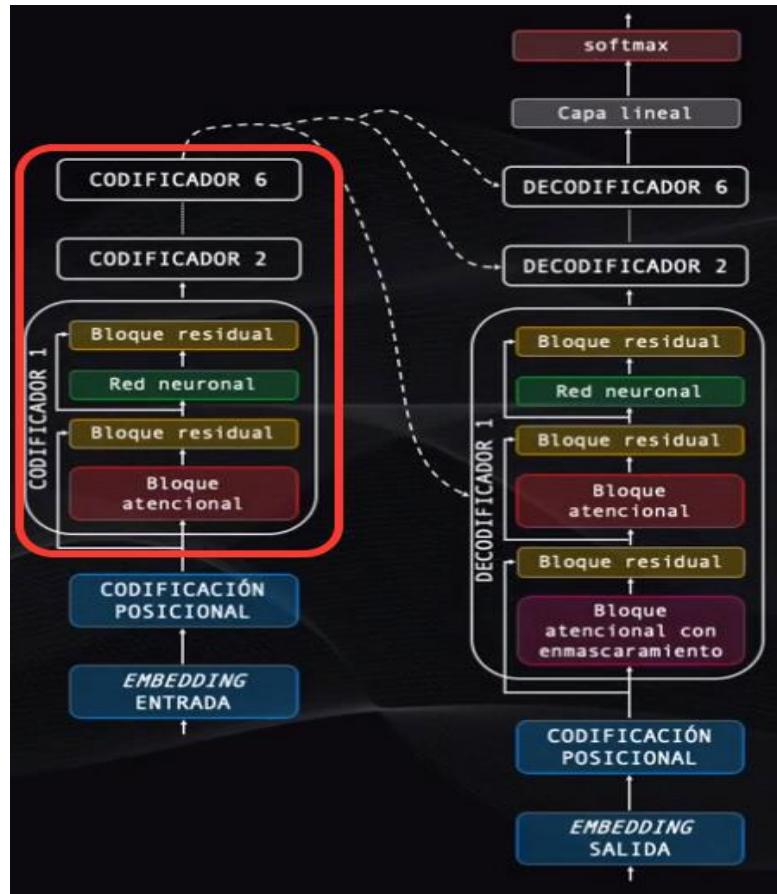


Figura 6. El bloque de codificación de la Red Transformer [9]

Cada codificador tiene cuatro elementos: un bloque atencional, un bloque de conexión residual, una red neuronal y otro bloque de conexión residual:

- Bloque atencional, es la más importante de toda la red, pues se encarga de analizar la totalidad de la secuencia de entrada (Ya que la red la procesa de manera simultánea) y de encontrar relaciones entre varias palabras de esta secuencia, Figura 7.

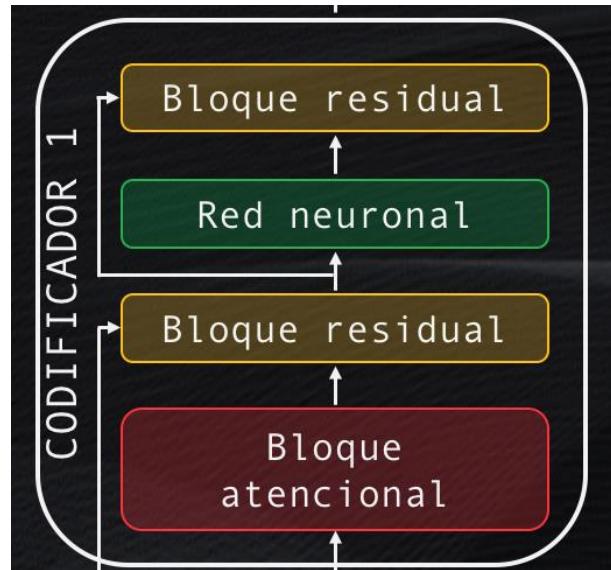


Figura 7. Primer codificador en la Red Transformer [9]

- Por ejemplo, si la secuencia de entrada es “I love Italian food” como en la Figura 8, se puede observar que existen al menos dos asociaciones posibles entre palabras: el sujeto “I” y el verbo “love” y el sustantivo “food” junto al adjetivo “Italian”. Pero además entre estas dos frases también existe una asociación:



Figura 8. Relación entre palabras de la frase ‘I love Italian food’ [9]

Esto permite que el bloque de atención exprese numéricamente las relaciones existentes en los diferentes niveles dentro de la secuencia, para posteriormente codificar cada uno con información del contexto, indicando a qué elementos del texto se les debe prestar más atención. De esta forma es como la red Transformer "entiende" este contexto para codificar cada palabra correctamente.

Para lograr esto, primero, los tokens se llevan al mismo tiempo a tres pequeñas redes neuronales, entrenadas para calcular vectores de "consulta", "clave" y "valor". Estos vectores son solo tres representaciones alternativas de los tokens originales.

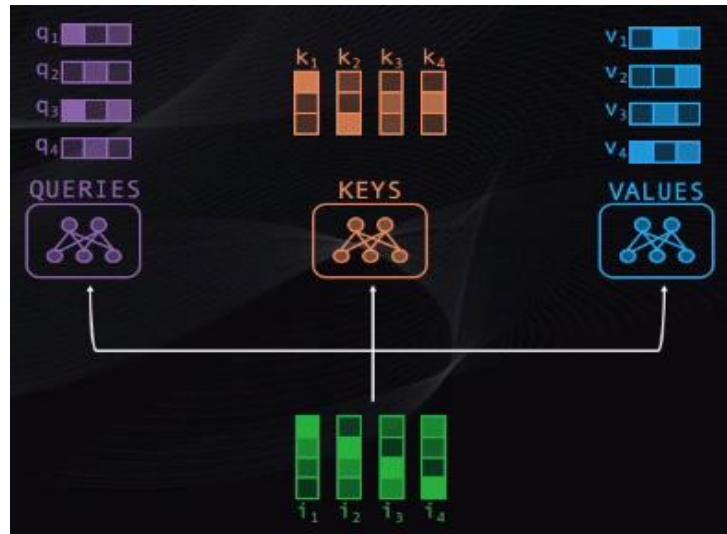


Figura 9. Obtención de ‘querie’, ‘key’ y ‘value’ [9]

- A continuación, se toma el ‘query’ de cada ‘token’ y se compara con cada uno de los ‘keys’ disponibles. Esta comparación solo es una multiplicación de vectores, y con esto se obtiene un puntaje que mide el grado de asociación entre pares de palabras, Figura 10.

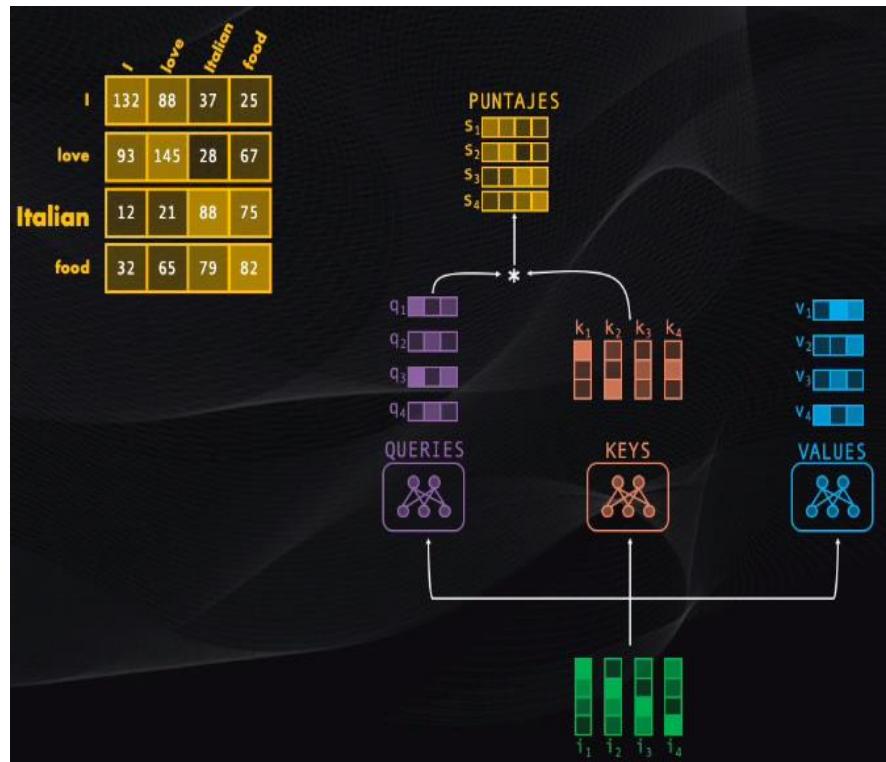


Figura 10. Comparación de ‘queries’ y ‘keys’ y obtención de puntajes [9]

Para la frase que deseamos traducir, si se analiza la palabra “Italian” los puntajes resultantes indican que al codificar este token se le debería prestar más atención a la misma palabra “Italian” seguida por la palabra “food”, y se debería considerar menos a las palabras “love” y “I”, que tienen puntajes

inferiores.

A continuación, se debe usar estos puntajes para ponderar cada uno de los vectores de valor, indicando de esta manera la importancia de cada palabra al codificar de los tokens.

Para hacer esto se deben escalar los puntajes, dividiéndolos entre el tamaño de cada vector, y posteriormente llevándolos a la función softmax (Figura 11), la misma que permite representar cada puntaje como una probabilidad entre cero y uno:

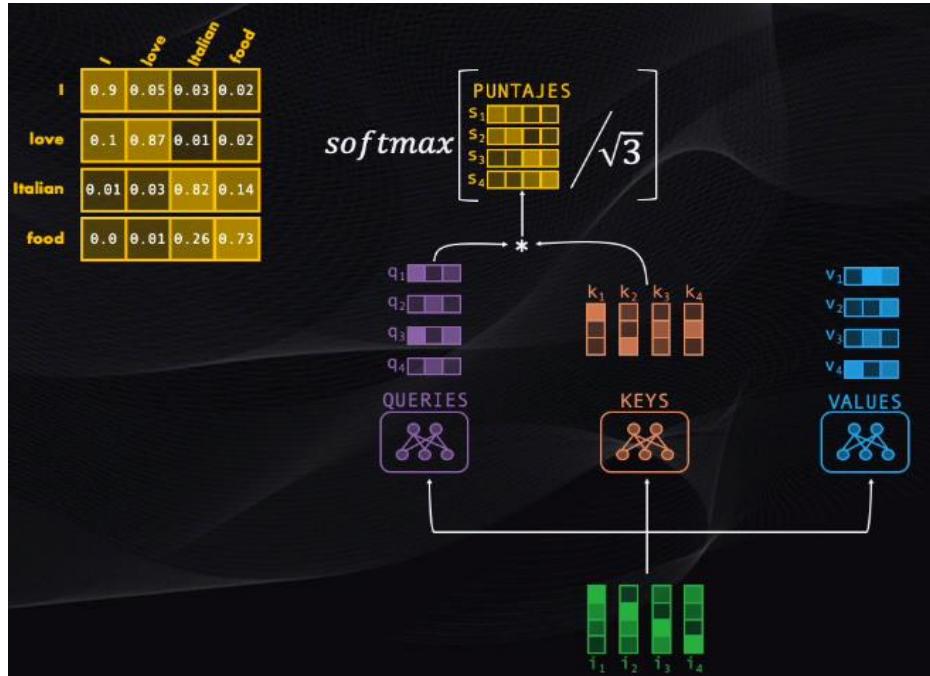


Figura 11. Empleo de la función ‘softmax’ en la matriz de puntajes [9]

Un valor próximo a 1 indica que se debe prestar más atención a ese token, y un valor menor o cercano a 0 significa que la palabra no es relevante.

Por último, se debe resumir la información resultante de la comparación en un vector por cada token. De esta forma se toma la matriz de puntajes y se la multiplica por la matriz de values: el resultado serán cuatro nuevos tokens, que contendrán la codificación de la información de contexto más relevante para cada palabra de la secuencia, como indica la Figura 12:

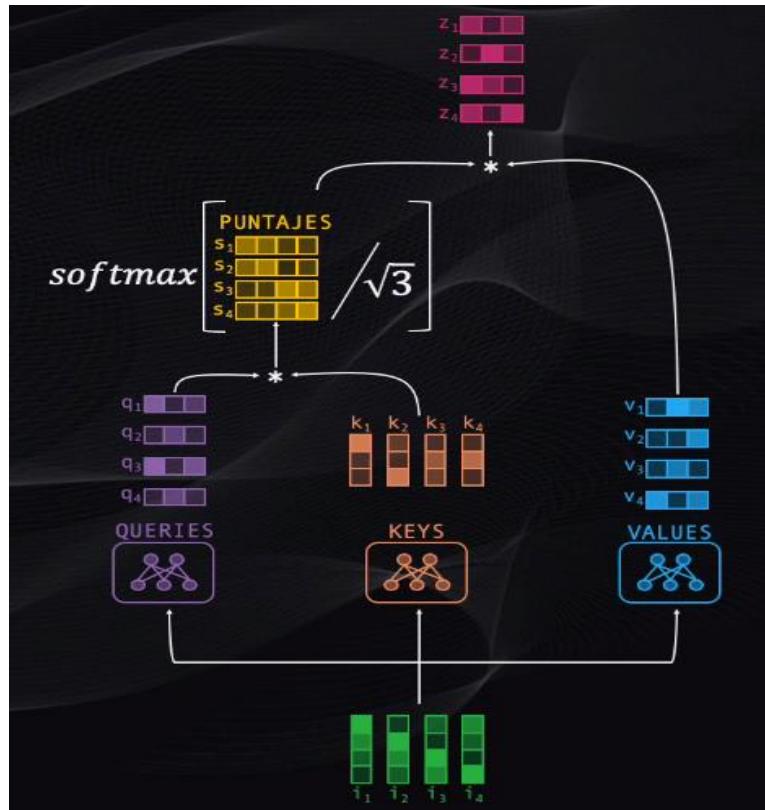


Figura 12. Salida del bloque atencional [9]

De forma general, el bloque atencional toma tokens iniciales y luego codifica en los tokens resultantes los elementos de la secuencia más relevantes. No obstante, también se producen asociaciones de palabras y frases, además de asociaciones entre palabras, asociaciones entre frases: para traducir la porción “Italian food” se necesita prestar atención a “I love”. Por este motivo un solo bloque de atención no es suficiente, y si se usa múltiples bloques atencionales es posible detectar y codificar asociaciones entre palabras y grupos de palabras a distintos niveles.

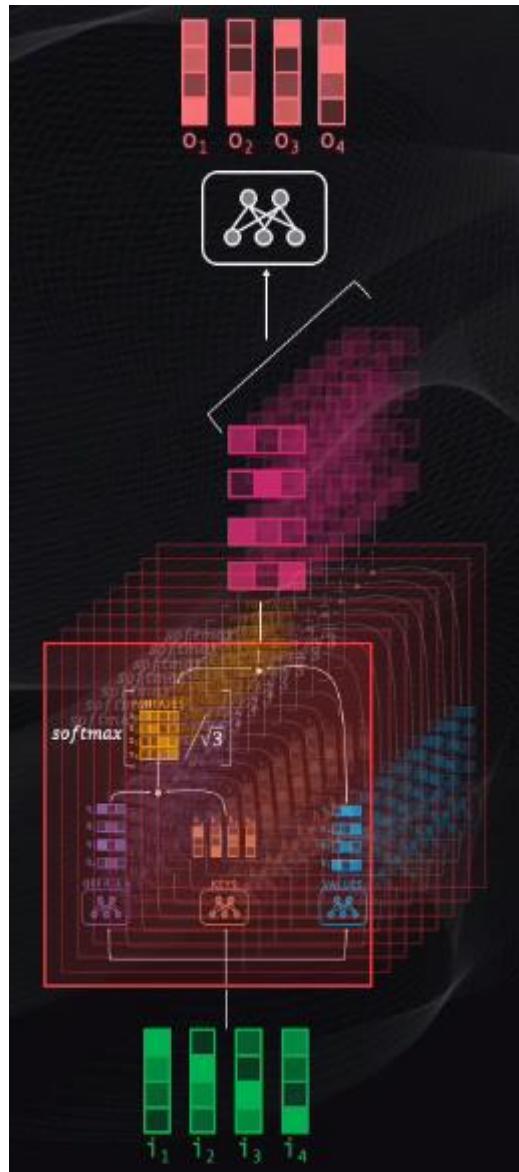


Figura 13. Bloques atencionales para localizar asociaciones entre palabras y grupos de palabras a distintos niveles [9]

Las salidas de estos bloques se combinan en una última red neuronal que condensa toda la información resultante en un único vector para cada token de entrada como se observa en la Figura 13, con lo que se tendría el bloque atencional de la red transformer [12].

- Codificación y el bloque residual: A este bloque se envían tanto la entrada como la salida del bloque atencional, y esto se hace pues la red es muy profunda y si tan solo se enviara la salida la información progresivamente se degradaría y esto dificultaría el entrenamiento y desempeño de la red. Esta etapa toma los dos datos, los suma y luego los normaliza para que tengan la escala adecuada requerida por el siguiente bloque, Figura 14 [12].

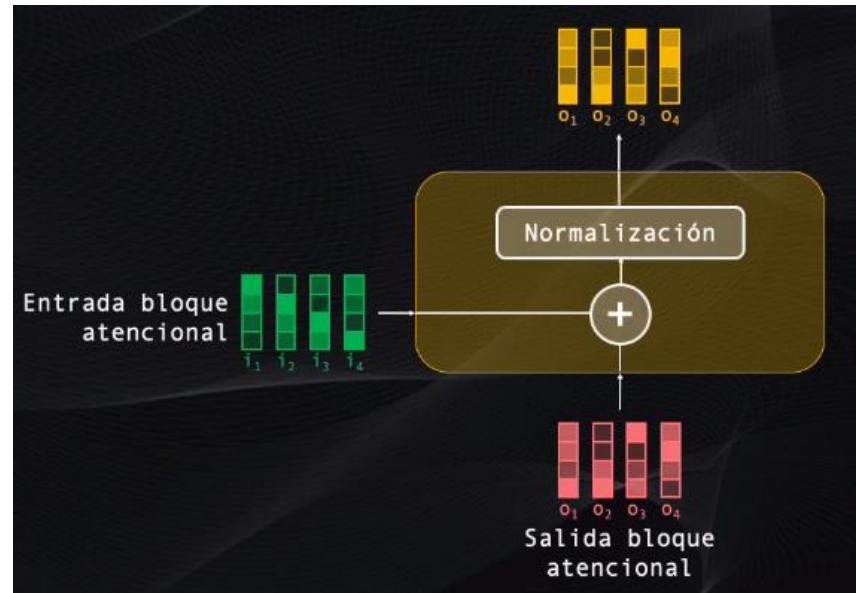


Figura 14. Elementos del bloque residual [9]

- Codificación: red neuronal y otro bloque residual; Después de pasar por la codificación y el bloque residual, tenemos una red neuronal seguida por otro bloque residual:

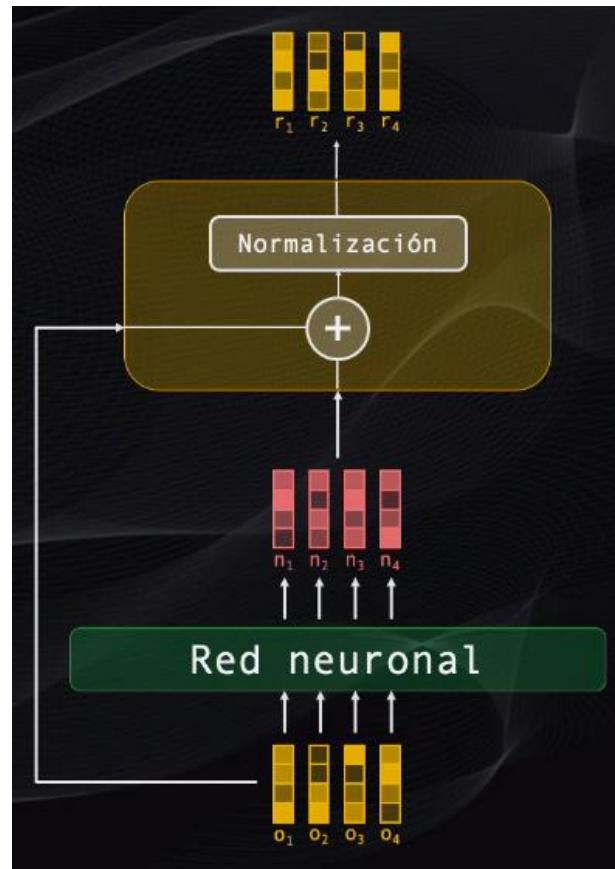


Figura 15. Etapa de salida del codificador: una red neuronal seguida de otro bloque residual [9]

La red neuronal procesa en paralelo todos los vectores de la secuencia, tomando la información atencional de las capas anteriores y consolidándola en una única

representación. La entrada y la salida de esta red neuronal son luego llevadas a un bloque residual que tiene exactamente las mismas características del bloque anterior: una suma seguida por una normalización de los datos, Figura 15 [12].

- Codificación: resultado final; Una vez listo el primer codificador, se puede resumir que el bloque en general, toma los tokens de entrada, los procesa en paralelo y entrega a la salida una representación que contiene información atencional sobre las diferentes relaciones entre palabras o grupos de palabras de la secuencia, importantes al momento de la traducción, Figura 16.

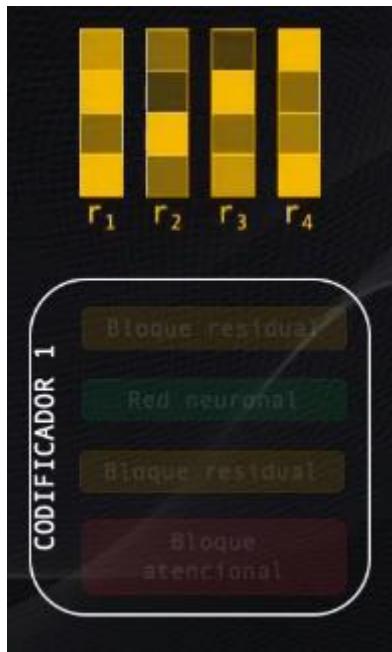


Figura 16. Salida resultante del primer codificador [9]

Este proceso se repite para los codificadores restantes, que son idénticos en estructura al codificador que acabamos de examinar [12].

- Decodificación: se desarrolla en el segundo bloque importante de la red transformer, que se encarga de hacer la traducción, Figura 17.

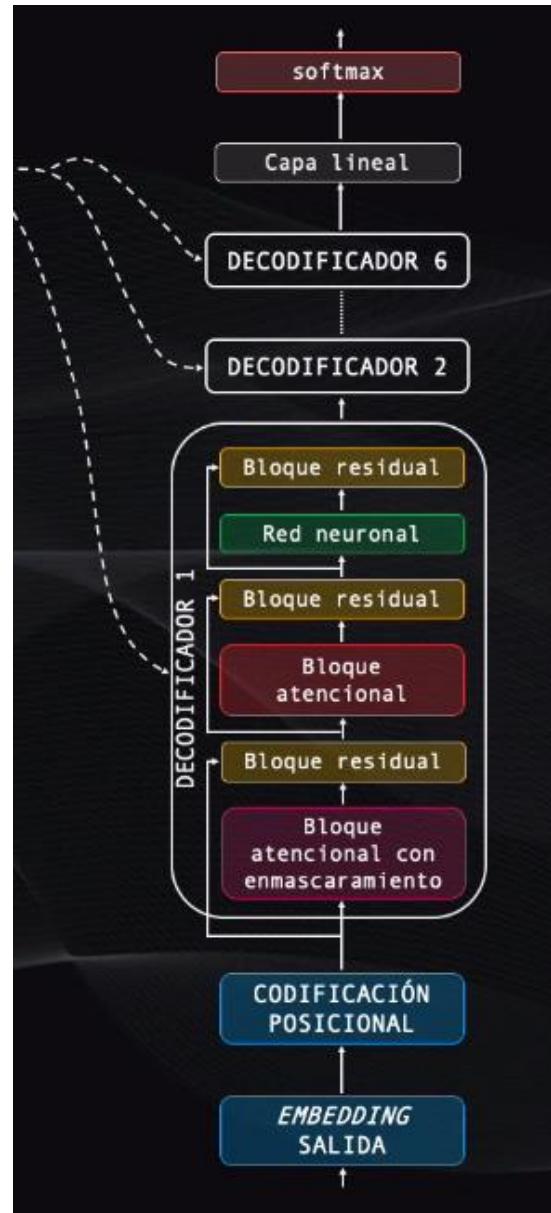


Figura 17. Elementos del decodificador [9]

Inicialmente, tenemos los bloques de embedding de salida y un codificador posicional, que cumplen la misma función de los bloques de la etapa de codificación. Luego el decodificador, que es muy parecido al bloque de codificación: el cual cuenta con 6 decodificadores, cada uno de ellos conectado al codificador, lo cual permite saber la información de entrada, en el idioma original, para realizar la traducción. Cada decodificador es parecido a los bloques de codificación que se vio anteriormente: cuenta con bloques de atención, residuales y redes neuronales con la misma estructura que el codificador. Sin embargo, poseen un bloque atencional de enmascaramiento y un bloque residual adicional. Posteriormente existe una capa lineal que, junto a la capa softmax, genera una a una las palabras de la secuencia de salida [12].

- Decodificación: bloque atencional enmascarado; La traducción empieza con la palabra clave “inicio”, que está codificada con el embedding y posicionalmente. Al ingresar al primer decodificador es procesada por el bloque atencional de enmascaramiento. Este bloque es igual al bloque atencional el cual codifica la relación entre elementos de la secuencia de salida, usando los ‘queries’, ‘keys’ y ‘values’, Figura 18.



Figura 18. Funcionamiento del bloque atencional de enmascaramiento [9]

Existe una gran diferencia, como se genera cada palabra de forma secuencial, el decodificador presta atención solamente a la palabra generada en ese momento y a las anteriores, no a las futuras. Por ejemplo, si en la secuencia nos ubicamos en la palabra “la”, el decodificador tiene acceso a esta y a la palabra “amo”, pero no a las palabras que se presentan luego en la secuencia (“comida” e “italiana”).



Figura 19. En el enmascaramiento no se presta atención a palabras futuras en la secuencia: ‘comida’ e ‘Italiana’ [9]

Para evitar esto, se agrega un bloque que enmascara, es decir que simplemente hace cero las palabras a las que durante la decodificación no se debe prestar atención.



Figura 20. Resultado del enmascaramiento [9]

Para evitar esto, se utiliza un bloque que se encarga de enmascarar, es decir que hace 0 las palabras a las que en la decodificación no se debe prestar atención. Al igual que con el codificador, también se emplean múltiples bloques atencionales para hallar relaciones entre niveles [12].

- Decodificación (bloque atencional): Todos los bloques residuales, al igual que la red neuronal de este decodificador funcionan de la misma forma a como ocurre en los codificadores.

El bloque atencional tiene la misma estructura, pero un funcionamiento ligeramente diferente al codificador: enfoca su atención en la secuencia original y en la de

salida, para ello toma la salida del codificador y las lleva a las redes “queries” y “keys”, mientras que el nodo “values” usa como entrada el dato proveniente del bloque residual, Figura 21.

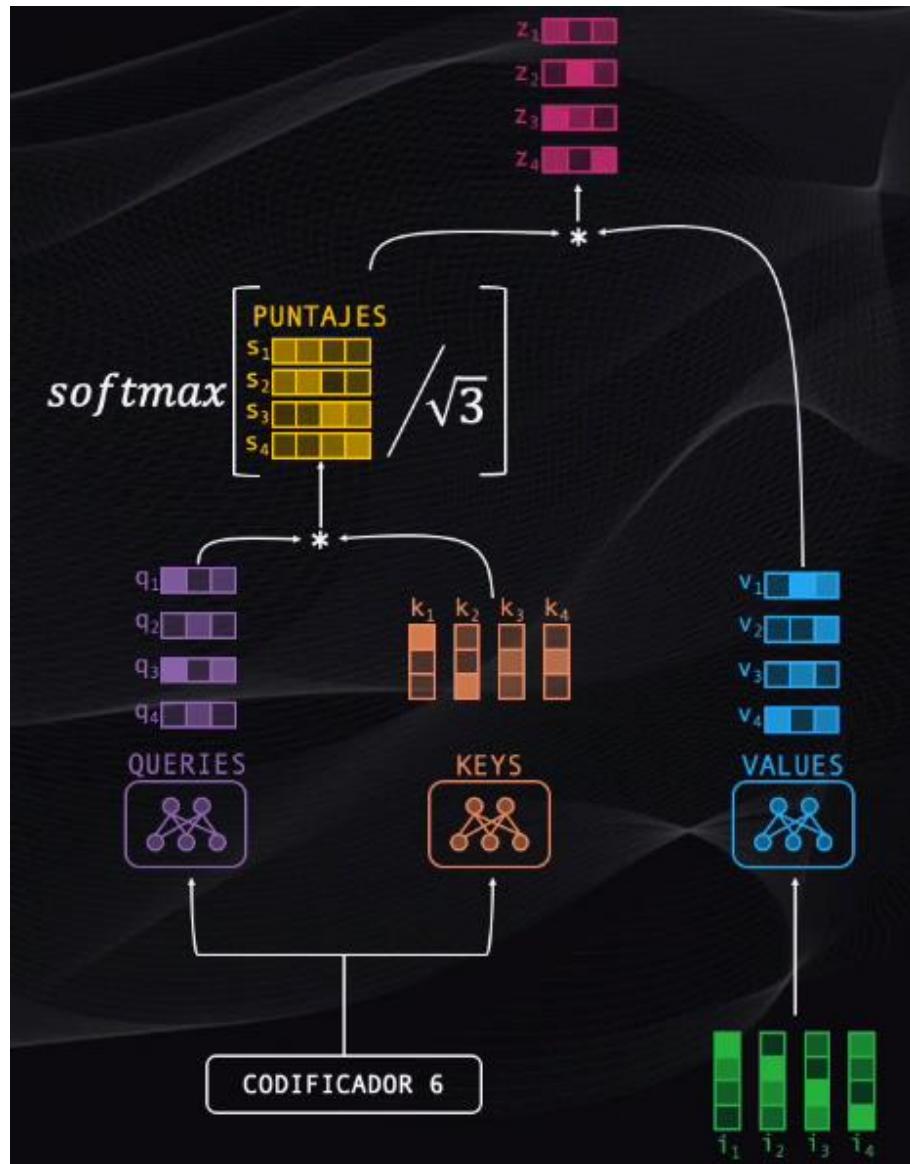


Figura 21. El bloque atencional del decodificador toma como entrada la información producida por codificador de la anterior etapa [9]

Es de esta forma el codificador le indica al decodificador a qué elementos debe prestar más atención al generar la secuencia resultante. En esta etapa nuevamente, se usan múltiples bloques atencionales al mismo tiempo para codificar asociaciones entre niveles. Con lo que se tendría el primer decodificador [12].

- Decodificación de decodificadores y etapas de salida: Este bloque se replica seis veces, y al final se produce un vector con cantidades numéricas, y faltaría por hacer es transformarlo, en una palabra, Figura 22.

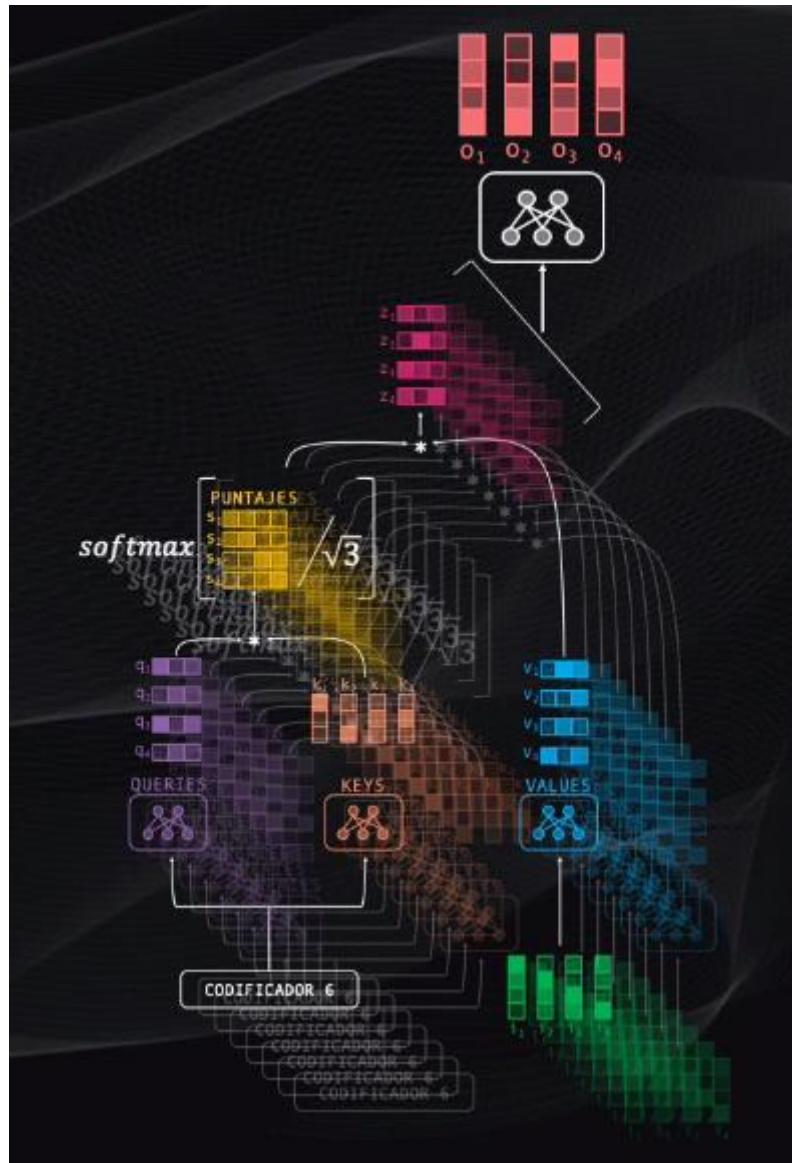


Figura 22. Se requieren múltiples decodificadores para hallar relaciones entre palabras y grupos de palabras entre niveles [9]

En primer lugar, se utiliza la capa lineal, que es una red neuronal que toma el vector resultante del decodificador y lo convierte en un vector más grande. Por ejemplo, si el traductor aprende 10000 palabras (el tamaño del vocabulario), entonces el vector de salida poseerá 10000 elementos.

La capa softmax utiliza cada elemento de este vector y lo transforma en una probabilidad, con valores positivos entre 0 y 1. La posición de la probabilidad más alta será elegida y la palabra asociada a dicha posición será la salida del modelo en ese instant, como se observa en la Figura 23.

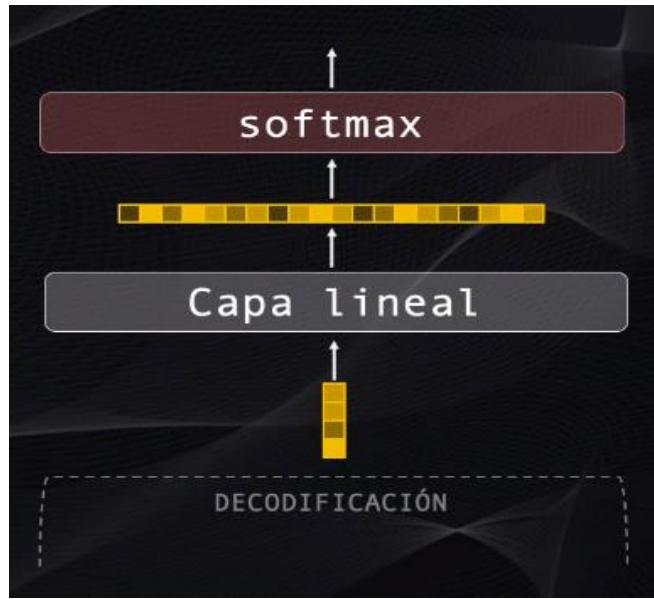


Figura 23. Capas de salida del bloque de decodificación [9]

Y este proceso se repite hasta generar la totalidad de la secuencia de salida. Esta Red Transformer analizada en el caso particular de la traducción, funciona con cualquier tipo de aplicación de NLP, como por ejemplo la generación de texto, cual se puede aplicar a otros contextos como en el caso presente es el de respuestas a preguntas [12].

4.1.7. Tipos de Chatbot:

Existen 3 tipos de chatbot, el primero basado en reglas, el cual tiene respuestas fijas establecidas por el desarrollador y solo responde a las preguntas que ya están en la base de datos, el segundo tipo de chatbot utiliza inteligencia artificial, este se entrena en base al procesamiento del lenguaje natural (NLP) con los conjuntos de datos, que son diálogos de conversación o información de un tema en particular, para extraer la combinación de la conversación, incluida la intención, el contexto y la entidad. Por último, tenemos los chatbot híbridos, el cual combina los conceptos anteriormente nombrados. Este chatbot puede comprender y comunicarse con los usuarios, pero permanece en el patrón determinado por el desarrollador.

4.1.8. Pytorch

PyTorch es una biblioteca de tensores optimizada para el aprendizaje profundo mediante GPU y CPU, proporciona dos funciones de alto nivel:

- Cálculo de tensor (como NumPy) con fuerte aceleración de GPU
- Redes neuronales profundas construidas en un sistema de autogrado basado en cintas

PyTorch no es un enlace de Python en un marco monolítico de C++. Está diseñado para integrarse profundamente en Python. Puede usarse de forma natural como con cualquier otra

librería de Python [10].

4.1.9. Transformers

Es una biblioteca de modelos preentrenados de última generación para el procesamiento del lenguaje natural (NLP). Proporciona arquitecturas de uso general (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet) para la comprensión del lenguaje natural (NLP) y la generación del lenguaje natural (NLG) con más de 32 modelos previamente entrenados en más de 100 idiomas y una interoperabilidad profunda entre Jax, PyTorch y TensorFlow. Se compone de codificador (capa de autoatención y la red reurnal de avance) y decodificador (capa de autoatención, atención del codificador-decodificador y red neuronal feed-forward [11] [9].

4.1.10. BM25

Se trata de una función de clasificación que clasifica un conjunto de documentos según los términos de consulta que aparecen en cada documento, independientemente de la interrelación entre los términos de consulta dentro de un documento (por ejemplo, su proximidad relativa). No es una función única, sino una familia completa de funciones de puntuación, con componentes y parámetros ligeramente diferentes. Los motores de búsqueda lo utilizan para clasificar los documentos coincidentes de acuerdo con su relevancia para una consulta de búsqueda determinada y, a menudo, se lo denomina "Okapi BM25", ya que el sistema de recuperación de información Okapi fue el primer sistema que implementó esta función. La fórmula de recuperación BM25 pertenece a la familia de modelos de recuperación BM (BM significa Best Match), es decir, el peso de un término en un documento [12].

4.1.11. Capas de Autoatención

Se tratan de capas de atención que codifican cada palabra de una frase en función del resto de la secuencia, permitiendo así introducir el contexto en la representación matemática del texto, motivo por el cual a los modelos basados en Transformer se les denomina también Embeddings Contextuales. Al no contar con redes neuronales recurrentes (RNN), emplea la estructura del decodificador del codificador, donde se componen de dos capas, el mecanismo de autoatención y una red de retroalimentación. El mecanismo de atención se usa para calcular las relaciones de dependencia entre palabras de la misma oración, las palabras de la oración de entrada que interactúan entre sí (self), es decir medida que el modelo procesa cada palabra (cada posición en la secuencia de entrada), la atención personal le permite buscar en otras posiciones en la secuencia de entrada pistas que puedan ayudar a llevar a una mejor codificación de esta palabra. La autoatención es el método que usa el Transformer para convertir la "comprensión" de otras palabras relevantes en la que estamos procesando actualmente [13].

4.1.12. GoogleTranslation

Googletrans es una biblioteca de Python gratuita e ilimitada, la cual implementó la API de Google Translate. La cual utiliza la API Ajax de Google Translate para realizar llamadas a métodos como detectar y traducir. Es compatible con versiones de Python 3.6 y superiores. Es rápido y fiable ya que utiliza los mismos servidores que Google emplea para su traductor predeterminado [14].

4.1.13. Docker

Docker es una plataforma abierta para desarrollar, enviar y ejecutar varias aplicaciones de forma más rápida. Docker permite que las aplicaciones se ejecuten por separado de la infraestructura del host y traten la infraestructura como una aplicación administrada. Docker también ayuda a enviar código, probar, implementar y acortar el ciclo entre escribir código y ejecutar código. Docker lo hace combinando una plataforma de virtualización de contenedores liviana con flujos de trabajo y herramientas que ayudan a administrar e implementar aplicaciones [15].

4.1.14. Haystack

Es un framework de Python para el desarrollo de sistemas de búsqueda en grandes colecciones de documentos, está disponible como una biblioteca de código abierto que proporciona diferentes aplicaciones entre las que destaca la recuperación de respuestas a preguntas. Su escalabilidad permite manejar un gran volumen de documentos mediante una API (Application Programming Interfaces) y a DocumentStore que actúa como una base de datos para almacenar textos y documentos y canaliza ofreciendo al retriever o recuperador en el momento de la consulta [16].

4.1.15. Retriever - Haystack

Retriever o Recuperador, se trata de una función que aporta la librería de Haystack, se desempeña como un filtro ligero que recorre el almacén de documentos al completo y obtiene un conjunto de documentos relevantes para la consulta que son enviados al “Lector” [17].

4.1.16. Reader- Haystack

Reader o Lector, es un sistema de control de calidad de dominio abierto en el lenguaje de aprendizaje automático, este componente permite a Haystack encontrar las respuestas junto a un modelo de lenguaje basado en transformadores, ya que aportan una buena comprensión de la semántica, es sensible a la estructura sintáctica y trabaja con modelos de control de calidad preentrenados [18] .

4.1.17. FARM

Farm “Framework for Adapting Representation Models” o Marco para la adecuación de los modelos de representación, se basa en transformadores y proporciona diferentes funciones como respuestas a preguntas y las transfiere a producción. Trabaja con los distintos modelos

preentrenados basados en BERT [19].

4.1.18. Elasticsearch

ElasticSearch es un motor de búsqueda y recuperación de documentos de tipo JSON basado en Apache Lucene, el cual permite realizar búsquedas por una gran cantidad de datos de un texto específico. Elasticsearch Gestiona información estructurada y no estructurada de múltiples fuentes, que se almacena de forma sofisticada y eficiente para optimizar búsquedas basadas en lenguaje/texto. De esta forma se puede almacenar, buscar y analizar grandes volúmenes de información de forma rápida y eficiente [20].

4.1.19. Streamlit

Es un framework de creación de aplicaciones de código abierto y gratuito creado para proyectos de aprendizaje automático y ciencia de datos, esta herramienta basada en Python permite crear aplicaciones web en tiempo real mientras se va codificando, sin la necesidad de saber HTML o CSS, permite la integración de Bootstrap y crea una interfaz interactiva y amigable con los usuarios [21].

4.1.20. Microsoft Azure

Se trata de una plataforma compuesta por productos y servicios en la nube, que permite crear, ejecutar y administrar aplicaciones gracias a las diferentes herramientas que posee. Ofrece diferentes infraestructuras como IaaS (Infraestructura como Servicio) servidores, máquinas virtuales, servicios de desarrollo, inteligencia artificial, IoT, almacenamiento y analítica, SaaS (Software como Servicio). PaaS (Plataforma como servicio) y CaaS (Alojamiento automatizado e implementación de paquetes de software en contenedores). Funciona con diferentes sistemas operativos y lenguajes de programación, todo bajo un esquema de pago por uso de sus productos y servicios [22].

4.1.21. Azure para estudiantes

Se trata de un crédito que ofrece la plataforma de Microsoft para poder usar los productos y servicios durante un año con un crédito de 100 USD, junto a varios servicios gratuitos. El crédito estudiantil está disponible para quienes cumplan ciertos requisitos como pertenecer a una institución educativa y acatar las condiciones de uso, límites cuotas y restricciones [23].

4.1.22. FastAPI

Se trata de un framework que permite construir APIs (Application Programming Interfaces o Interfaz de programación de aplicaciones) de forma rápida con Python, con la finalidad intercambiar datos entre los distintos módulos de un sistema [24].

4.1.23. Fine Tuning

Transformers brinda acceso a una gran cantidad de modelos preentrenados para diferentes tareas. Cuando se usa alguno de estos modelos, en ocasiones se entrena el modelo con un conjunto de datos específico para una tarea específica. Esto se conoce como ajuste fino o Fine Tuning, la cual una técnica de entrenamiento increíblemente poderosa. El ajuste fino, se trata de una modificación de la última capa del modelo, ya que las últimas capas están destinadas para realizar tareas específicas y, por lo tanto, cambian durante el proceso de ajuste, mientras que las primeras capas permanecen relativamente estables [25].

4.1.24. Google Colaboratory

Es una herramienta que permite a sus usuarios escribir y ejecutar código de Python en el navegador, su uso se enfoca a tareas como análisis de datos, aprendizaje automático y educación. Da acceso a las GPUs y TPUs y el contenido que se elabora se puede compartir y trabajar con otros usuarios [26].

4.1.25. Semantic Scholar

Es una base de datos que proporciona un buscador académico basado en inteligencia artificial, se desempeña en campos como el de la salud y geociencia, este diseño para investigadores que buscan resultados relevantes permitiendo ahorrar tiempo en las búsquedas y fomentar el conocimiento científico. Proporciona una API y conjuntos de datos de dominio público, fue fundado por el Instituto Allen para la Inteligencia Artificial (AI2) [27].

4.1.26. Google Cloud

Google Cloud es un proveedor de recursos computacionales en la nube que se utilizan para diseñar, testear y lanzar aplicaciones web. Es una infraestructura completa que cuenta con varias características y herramientas, para que sus clientes puedan realizar una gestión eficiente y con seguridad digital completa.

Esta plataforma cuenta con más de 100 productos y servicios, entre los cuales tenemos: máquinas virtuales, alojamiento de aplicaciones web, almacenamiento de archivos, banco de datos, servidores, compartir y procesar datos, creación de aplicaciones, herramientas de gestión [28].

4.1.27. Coronavirus

Los coronavirus constituyen una familia de virus ARN, monocatenario y de cadena positiva, envueltos. Desde 1968, se otorga su nombre por el aspecto en forma de corona observada en el microscopio electrónico, donde las proyecciones de la membrana del virus, conocidas como espículas, le dan la apariencia. Los coronavirus se dividen en tres géneros (I a III) en todos los casos de transmisión por animales. La subfamilia se clasifica en cuatro géneros: alfa, beta, gamma y delta, siendo los primeros dos los que infectan al humano. Se han descrito siete coronavirus que causan enfermedad en humanos: 229-E (α -CoV), NL63 (α -CoV), OC43

(β -CoV), HKU1 (β -CoV), MERS-CoV (β -CoV), SARS-CoV (β -CoV) y el séptimo miembro es el último descubierto SARS-CoV-2 (β -CoV) [11].

4.1.28. SARS-CoV-2

El Síndrome Respiratorio Agudo Severo 2 (SARS-CoV-2) proveniente de la familia de coronavirus que surgió en el 2019 en Wuhan, China y la enfermedad que produce denominada COVID-19 por la Organización Mundial de la Salud (OMS) cuyo origen específico aún no se ha determinado. Tiene una amplia gama de manifestaciones como un simple resfriado, una neumonía leve, moderada o grave, hasta sepsis, shock séptico o la muerte. De su estructura, el componente más importante es la glicoproteína espiga S, la cual tiene la función de unir y fusionar al virus con las células del hospedador y multiplicarse [12].

4.2. Trabajos relacionados

De acuerdo a la revisión bibliográfica realizada, en la **TABLA I** se presentan los trabajos o estudios relacionados con el presente TT.

TABLA I .Estudios primarios seleccionados

Nº	Estudios seleccionados	Ref.	Buscador
ES01	Prototipo de un chatbot para compras online utilizando bot framework	[1]	Google Académico
ES02	Asistente virtual web basado en inteligencia artificial para la escuela tic de la universidad piloto de Colombia	[2]	Google Scholar
ES03	Asistente Virtual Tipo Chatbot	[3]	Google Scholar
ES04	Implementación de un prototipo agente virtual basado en machine learning e inteligencia artificial aplicando Azure Bot Service para mejorar el servicio de una empresa de venta de soluciones tecnológicas.	[29]	Google Scholar
ES05	Learning to Extract Attribute Value from Product via Question Answering: A Multi-Task Approach	[4]	ACM
ES06	Multilingual Question Answering from Formatted Text applied to Conversational Agents	[5]	Google Scholar
ES07	A building regulation question answering system: A deep learning methodology	[6]	Google Scholar
ES08	Know What You Don't Know: Unanswerable Questions for SQuAD,	[7]	Google Scholar
ES09	Transfer Learning from BERT to Support Insertion of New Concepts into SNOMED CT	[30]	Google Scholar
ES10	Deep Learning Based Fusion Approach for Hate Speech Detection	[31]	IEEE
ES11	A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization	[32]	IEEE
ES12	BERT: Pre-training of deep bidirectional transformers for language understanding	[33]	Google Scholar
ES13	Performance Analysis of Deep Learning Workloads on a Composable System	[34]	IEEE

ES14	Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning	[35]	Google Scholar
ES15	Extracting family history of patients from clinical narratives: Exploring an end-to-end solution with deep learning models	[36]	Scopus
ES16	Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish	[37]	Scopus
ES17	Sentimental analysis of COVID-19 tweets using deep learning models	[38]	Scopus
ES18	Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0	[39]	Google Scholar
ES19	Controlling risk of web question answering	[40]	Google Scholar
ES20	SQuAD: 100,000+ questions for machine comprehension of text	[41]	Google Scholar
ES21	Combat COVID-19 infodemic using explainable natural language processing models	[42]	Scopus
ES22	Gestalt: a Stacking Ensemble for SQuAD2.0	[43]	Google Scholar
ES23	Target-dependent sentiment classification with BERT	[44]	IEEE
ES24	Generating contextual embeddings for emergency department chief complaints	[45]	Scopus
ES25	An ensemble learning strategy for eligibility criteria text classification for clinical trial recruitment: Algorithm development and validation	[46]	Scopus
ES26	Fine-tuning the BERTSUMEXT model for clinical report summarization	[47]	IEEE
ES27	Improving Fine-Tuned Question Answering Models for Electronic Health Records	[48]	ACM

Los hallazgos correspondientes a cada uno de los estudios seleccionados, fueron analizados en la revisión sistemática de literatura correspondiente al primer objetivo (véase sección 6.1.1), en la cual se obtuvo información relevante que aporto con el conocimiento necesario para el desarrollo del proyecto, tal como la elección del lenguaje de programación, el modelo adecuado para el desarrollo del agente, los parámetros y recursos necesarios para el ajuste del mismo, y las preguntas o palabras más comunes relacionadas con el COVID-19. Cada uno de estos hallazgos se plasman en la subsección 6.1.1 Tarea 1, en la que se realiza una RSL. Además, se realizó una segunda RSL como parte de la subsección 6.1.3, donde se seleccionaron estudios que posean información acerca de los temas y preguntas más buscadas por la población respecto al covid-19, en la **TABLA II** se muestran estos estudios.

TABLA II. Estudios seleccionados, temas tópicos del covid-19

Nº	Estudios seleccionados	Ref.	Buscador
ES27	The Laboratory Diagnosis of Coronavirus Disease 2019— Frequently Asked Questions	[49]	Google Scholar
ES28	Frequently asked questions (FAQs) about coronavirus disease 2019 (Covid-19)	[50]	Google Scholar
ES29	Exploratory analysis of internet search trends during the COVID-19 outbreak	[51]	Scopus
ES30	Informative treatment and media competencies on COVID-19 in Ecuador	[52]	Google Scholar
ES31	Aplicaciones de búsqueda de Google para la comunicación de riesgos en el manejo de la COVID-19	[53]	Google Scholar
ES32	Frequently Asked Questions and Answers	[54]	Google Scholar
ES33	Applications of Google Search Trends for risk communication in infectious disease management: A case study of the COVID-19 outbreak in Taiwan	[55]	Scopus
ES34	Frequently asked questions and myth busters on COVID-19	[56]	Google Scholar
ES35	More effective strategies are required to strengthen public awareness of COVID-19: Evidence from google trends	[57]	Scopus
ES36	Association of search query interest in gastrointestinal symptoms with COVID-19 diagnosis in the United States: Infodemiology study	[58]	Scopus
ES37	Google Trends Analysis of COVID-19 Pandemic	[59]	IEEE Digital Library
ES38	Correlations of online search engine trends with coronavirus disease (COVID-19) incidence: Infodemiology study	[60]	Scopus

Los artículos seleccionados fueron de gran ayuda para conocer cuáles son los temas y preguntas más buscadas por las personas en la época de la pandemia, y de esta forma generar los datos necesarios para el entrenamiento del modelo BERT y así desarrollar el agente virtual, cada uno de estos hallazgos se encuentran en la subsección 6.1.3. Tarea 3: Selección de artículos destacados respecto a temas tópicos sobre el covid-19.

5. Metodología

La presente sección expone que tipo de investigación, materiales y métodos se emplearon en el Trabajo de Titulación (TT), se aplicaron dos tipos de investigación: exploratoria y descriptiva. La investigación exploratoria [61], aplicada para el análisis y comprensión del modelo de lenguaje BERT e identificación de los modos de desarrollo de un agente conversacional para responder preguntas sobre el covid-19, empleando revisiones sistemáticas de literatura (RSL), la investigación descriptiva [61], se dedicó a la comprensión de la estructura y funcionamiento del modelo BERT junto a SQuAD junto con otra RSL, además, se empleó para consultar mediante cuestionarios el desempeño del agente conversacional en cuestiones de coherencia de respuestas, tiempos y funcionalidad, sirviendo así de apoyo a la población que tenga consultas sobre el covid-19; y posteriormente validar que el agente cumple con la finalidad para el cual fue creado. La sección se compone se cuatro apartados que se son: materiales, métodos, técnicas y metodología utilizados en el desarrollo del TT.

5.1. Tipo de investigación

Durante la ejecución del presente TT se empleó un tipo de investigación cuantitativa con un diseño experimental, ya que cuenta con objetivos y preguntas de investigación, revisiones de literatura y un marco teórico mediante el cual se recolectaron, analizaron, probaron datos y manipularon variables y parámetros, con el propósito de dar respuesta a preguntas sobre temas relacionados con el covid-19, a través de algoritmos y técnicas de procesamiento de lenguaje natural junto a los modelos BERT con SQuAD.

Así mismo, se realizaron investigaciones bibliográficas y de campo, ya que para el cumplimiento de dos objetivos se realizaron revisiones bibliográficas con el fin de recolectar estudios relacionados, sobre las características y el funcionamiento de los modelos BERT con SQuAD; en lo que concierne a la investigación de campo se empleó encuestas, realizadas a docentes de la Facultad de Salud Humana de la Universidad Nacional de Loja, de los cuales ciertos docentes laboran en el sector privado, con el objetivo de justificar y sustentar las razones por las que fue importante la ejecución del presente TT y se justifica la viabilidad del desarrollo del agente virtual para brindar asistencia acerca del covid-19.

5.2. Métodos de investigación

5.2.1. Método científico

Como precisa Castán [62], el método científico es un procedimiento empleado para generar más conocimiento a partir del que se domina, consiste en la observación sistemática, medición experimentación, formulación, interpretación y modificación de las hipótesis; permitiendo generar respuestas a las incógnitas de la investigación, específicamente se enfoca en unir la

ciencia y conocimiento científico, de modo que mediante dichos procedimientos el científico puede proyectar, planear y organizar un trabajo, así como las actividades para ejecutarlo. Este método se empleó durante el análisis de la documentación obtenida en la revisión bibliográfica, donde se hizo una selección de los modelos empleados en el desarrollo del trabajo, el lenguaje de programación en el que se desarrollan estos modelos, así como los algoritmos y herramientas de software para poder configurar y ajustar el modelo preentrenado, accediendo a información sobre el alcance y eficacia de los diferentes parámetros que fueron usados en los estudios relacionados, para posteriormente compararlos con los resultados obtenidos en el presente TT.

A través de la experimentación, se seleccionaron artículos sobre el covid-19 con base al parámetro “score”, que optimizó el modelo para clasificar y ofrecer las mejores respuestas, ofreciendo el mejor desempeño durante la validación y test del modelo, es decir, se trabajó con el fin de obtener un modelo entrenado en un tema particular, dando a lugar un agente conversacional experto en un tema.

5.2.2. Metodología Desing Thinking

La metodología Desing Thinking [63] implementado en el presente TT permitió seguir un orden en el cual se emplearon las fases: Empatizar, Definir e Idear para abordar el primer objetivo, para el segundo objetivo se usó: la creación de Prototipo, que permite constatar de primera mano el funcionamiento del agente conversacional en una aplicación web para la interacción de los usuarios con el agente virtual, donde se toma en cuenta las necesidades de las personas para poder aportar soluciones innovadoras y para la finalización del TT se empleó la fase llamada Testeo, en la cual se realizan pruebas al prototipo para comprobar su cometido. Se debe tener en cuenta que las fases, se pueden realizar sin un orden determinado; además, pueden ocurrir en afines y repetirse iterativamente. La finalidad de la metodología no consiste en ver las diferentes fases como un proceso secuencial si no como los pasos para llegar al desarrollo de un proyecto innovador, consta de cinco fases las cuales son las siguientes:

5.2.2.1. Empatizar

En esta primera fase se debe pensar empáticamente en lo que necesitan las personas. Esta fase fue amparada por un cuestionario realizado a los docentes de la Facultad de Salud de la Universidad Nacional de Loja, con el fin de comprobar la viabilidad y utilidad del agente conversacional para ayudar contra la pandemia.

5.2.2.2. Definir

Se necesita establecer que necesita el usuario, en este caso es información confiable sobre el covid-19, con lo que podríamos solucionar el problema de la desinformación y complicaciones que conlleva, además de la accesibilidad a esta información verificada y

justificada mediante una plataforma web. Se empleó el cuestionario mencionado en la fase anterior ya que también permite definir la necesidad y el aporte que se ofrece a la comunidad de la salud.

5.2.2.3. Idear

Se idearon alternativas para dar apoyo contra la lucha del covid-19, como buscadores de información y aplicaciones confiables de información, de lo que surgió una solución, un prototipo de agente conversacional para brindar asistencia, empleando inteligencia artificial, procesamiento de lenguaje natural y modelos especializados en dar respuestas a preguntas, todo esto integrando en una aplicación web.

5.2.2.4. Crear Prototipos

Se tomó como referencia, agentes conversacionales guiados, los cuales están programados en ciertos ámbitos predefinidos, con esta referencia como base se ideó la creación de un prototipo de agente que pueda aportar mayor flexibilidad frente a las consultas que se le realicen en un entorno web y así poder cumplir el fin de la presente fase que se centra en determinar la mejor solución para el problema identificado.

5.2.2.5. Testear

En esta última fase, se probó el prototipo, en este caso al ser un agente conversacional que ofrece información sensible sobre el covid-19, se seleccionaron a usuarios especializados en sector de la salud para poder comprobar la validez de las respuestas que ofrece el agente y si existe algún error poder solventarlo.

5.3. Técnicas e instrumentos para la recolección de información

5.3.1. Recopilación documental

A través de la revisión bibliográfica realizada en base a la metodología de Bárbara Kitchenham [64] se seleccionaron y analizaron los trabajos relacionados, los que contribuyeron a dar la sustentación científica al presente TT.

La flexibilidad de la metodología de Bárbara Kitchenham, permitió omitir ciertas subfases que fueron consideradas como no relevantes para dar cumplimiento a los objetivos del presente TT. A continuación, se presenta las fases y subfases que finalmente fueron realizadas.

Planificación de la revisión

- Identificación de la necesidad de una revisión.
- Especificación de las preguntas de investigación.
- El desarrollo de un protocolo de revisión.

Realizar la revisión

- Identificación de la investigación.
- Selección de los estudios primarios.

- Síntesis de los datos.

Revisión de informes

- Dar formato al informe principal.

5.3.2. Instrumentos de recolección de datos

5.3.2.1. Encuesta

La encuesta como menciona [61] se trata de una técnica que pretende obtener información que suministra un grupo de sujetos acerca de un tema en particular. En este caso particular fue mediante “Google Forms” empleando preguntas cerradas a través del correo electrónico de los diferentes docentes de la Facultad de Salud Humana de la Universidad Nacional de Loja, además permitió adquirir conocimiento de parte de los profesionales en el área de salud que trabajan en instituciones privadas como en clínicas o centros médicos, con el fin de aclarar las interrogantes que se pretendió conocer acerca de la viabilidad de la creación de un agente conversacional para apoyar al área de salud. Esta técnica se empleó para obtener información que justifique y sustente la elaboración del presente TT.

5.3.3. Tratamiento de la información: Metodología KDT

Para manipular la información se tomó como referencia la metodología llamada “Descubrimiento de Conocimiento en Textos” (KDT, Knowledge Discovery in Text) aplicada en procesos de procesamiento del lenguaje natural (para realizar el preprocesamiento y etiquetado de los textos), según Yves Kodratoff [65], KDT se conoce como la ciencia que descubre el conocimiento en los textos, donde el conocimiento extraído tiene que basarse en el mundo real y modificar el comportamiento de un agente humano o mecánico. La metodología mencionada según W. Elmenreich, J. T. Machado, and I. J. Rudas [66] tiene una similitud de las técnicas de preprocesamiento del proceso KDD, donde deduce conocimientos y consta de las siguientes etapas o pasos:

5.3.3.1. Entender el dominio de la aplicación y los objetivos del proceso KDT:

El usuario tiene que definir qué conceptos son los valiosos para poder abordar el marco de la investigación, como pueden ser los modelos empleados, el software, algoritmos, lenguajes de programación, técnicas de afinamiento de los modelos, validación y metodologías de los mismos, con el fin de entender todo el marco que abarca el proceso, junto a los objetivos que se pretenden culminar al emplear las diferentes etapas que forman este proceso o metodología KDT.

Para esta etapa se definió gracias al uso de una revisión de trabajos relacionados que figura

en la sección de Revisión de literatura, subsección 4.2. Trabajos relacionados, en la que se indagó sobre el uso y empleo que dan al modelo BERT entrenado con el conjunto de datos de SQuAD 2.0, que permitió comprender el dominio de la aplicación y los pasos que se debía desarrollar para poder generar un prototipo de un agente conversacional, donde se obtuvo el lenguaje de programación que emplea, sus librerías, frameworks, conjuntos de datos, servicios en la nube, despliegue y levantamiento del sistema.

5.3.3.2. Adquisición o selección de un conjunto de datos objetivo:

Esta etapa se basó en la recopilación de los artículos usando herramientas de recuperación de información y de forma manual de fuentes de datos abiertas como Kaggle y Google Scholar, las cuales contribuyeron como objeto de estudio para las etapas restantes del proceso KDT, los mencionados artículos se recolectaron basándose en temas, preguntas o palabras clave, se aplicaron ciertos criterios como el año de publicación, que sean publicados después del año 2020, en lenguaje inglés y que traten sobre el coronavirus, se obtuvieron los artículos mediante la página Kaggle y al desafío de conjunto de datos de investigación abierta COVID-19 (CORD-19) [67], donde abarca todos los criterios mencionados, ya que el conjunto de datos está desde el 13 de Marzo del 2020 en el inicio de la pandemia del coronavirus y se actualiza conforme a las nuevas investigaciones.

5.3.3.3. Limpieza, preprocesamiento y transformación de los datos:

Para la presente etapa, los artículos se analizaron y almacenaron en base a la estructura que ofrece el conjunto de datos CORD-19 (el cual posee título, doi, resumen, autores, journal, url, entre otros campos de cada artículo), con esta estructura se seleccionaron solo las columnas necesarias para la creación de la base de conocimiento y se escogieron los siguientes campos: sha, title, abstract, publish_time, authors, url y body_text. Donde se aplicaron filtros para eliminar campos vacíos, incompletos o duplicados que le generen errores posteriormente al modelo y que no aporten en dar respuestas acordes a las preguntas que se realicen. Los procedimientos se encuentran evidenciados en la subsección 6.2.1.1. Preparar el conjunto de datos, se aplicó un preprocesamiento de datos a los artículos, que se redujeron de 50.000 hasta llegar a los 400 documentos gracias a filtros creados para separar por año de publicación y por palabra claves, para posteriormente utilizarlos en la base de conocimiento del agente y que no produzca ningún error o alteración en el modelo.

5.3.3.4. Desarrollo del modelo:

En esta etapa se procedió a elaborar el modelo ajustado partiendo del modelo BERT junto a SQuAD con los artículos y preguntas anteriormente indicadas, se creó, validó y probó el modelo, mediante un ajuste fino (fine tune) con los parámetros definidos en las características y funcionamiento del modelo BERT, donde se obtuvo un modelo optimizado referente a

cuestiones relacionadas con el covid-19 (véase la sección 6.1.2).

5.3.3.5. Elección y ejecución de los algoritmos de minería de datos adecuados:

Para el desarrollo de esta etapa se utilizó el modelo entrenado y ajustado mencionado anteriormente, con la finalidad de que se ejecute correctamente con el modelo seleccionado en la revisión sistemática de literatura. En la ejecución de estos modelos y de su ajuste posterior, se organizaron los textos de los artículos y se colocó un token a cada palabra para poder asociarla correctamente cuando se realicen preguntas dentro de este contexto y así ofrecer una respuesta acorde a las preguntas. Además, se seleccionó la herramienta Haystack [68] para identificar las identidades de las preguntas y poder facilitar las búsquedas de una forma más rápida y optimizada ya que delimita el contexto en el campo de búsqueda de los textos disponibles en la base de conocimiento, la cual almacena el dataset principal de donde se suministraran los documentos para dar una respuesta afín a las preguntas planteadas por los usuarios.

5.3.3.6. Interpretación y visualización de los resultados:

En esta última etapa se debe interpretar los resultados obtenidos, mediante reportes del funcionamiento del sistema a través de diferentes Métricas para control de calidad, en este caso se seleccionaron dos métricas base que se emplean en modelos de PLN para conjuntos de datos de respuesta a preguntas, como puede ser: coincidencia exacta (exact match “EM”) que la puntuación varía entre 0.0 y 1.0 y el score, esta puntuación la produce el mismo modelo, estas puntuaciones generales de EM y score se calculan promediando las puntuaciones individuales de sus respuestas. De esta forma se pueden obtener conclusiones de acuerdo a los objetivos planteados en la primera parte de este proceso KDT (Knowledge Discovery in Text) y a los resultados conseguidos al ajustar el modelo BERT (Bidirectional Encoder Representations from Transformers) [69].

6. Resultados

En la presente sección se detallan los resultados de cada uno de los objetivos específicos del presente TT donde se evidencia las diferentes bases científicas que respaldan el desarrollo del proyecto para la obtención de información como: modelos de lenguaje, requerimientos, parámetros, características y funcionamiento.

A continuación, se detallan cada uno de los objetivos:

6.1. Objetivo 1: Definir el modelo de lenguaje BERT junto a SQuAD para generar un modelo que dé respuesta a las preguntas realizadas al agente virtual.

Para el desarrollo de este objetivo se aplicaron las tres primeras fases de la metodología Desing Thinking, en la que mediante la primera fase empatizar se realizó una encuesta para conocer si un agente conversacional puede apoyar al área de salud, para la segunda fase se definió que es lo que necesita el usuario, que en este caso sería cualquier persona que necesite información respecto al covid-19, para concluir el objetivo 1, se usó la fase de idear, se ejecutó una revisión sistemática de literatura para obtener información respecto al diseño y generación de modelos que permitan elaborar un agente conversacional.

Este objetivo 1, esta divido en tres tareas concretas que se detallan a continuación:

- **Tarea 1:** Realizar revisión bibliográfica, basada en las fases de la metodología de RSL de Bárbara Kitchenham
- **Tarea 2:** Extracción de información de artículos relacionados con las características y el funcionamiento respecto al empleo de BERT y SQuAD.
- **Tarea 3:** Selección de artículos destacados respecto a temas tópicos sobre el covid-19.

6.1.1. Tarea 1: Realizar revisión bibliográfica, basada en las fases de la metodología de RSL de Bárbara Kitchenham

La elaboración de la revisión sistemática de literatura se basa en el método de Bárbara Kitchenham [64] con la ayuda de la herramienta en línea Parsifal, que permite trabajar dentro de un espacio de trabajo compartido para diseñar protocolos y realizar estudios propuestos. Dado que proporciona el mecanismo para hacer cumplir la SLR [70], el proceso se resume en tres etapas principales:

- Planificación de la revisión
- Realizar la revisión
- Revisión o informe de la revisión.

De acuerdo a Kitchenham [64], algunas tareas de estas fases no son obligatorias, como por ejemplo:

- Puesta en marcha de una revisión, depende de la revisión sistemática que se está haciendo sobre una base comercial.
- Evaluar el protocolo de revisión y la evaluación del informe, son opcionales y dependen de los procedimientos de garantía de calidad decididas por el encargado de la revisión sistemática.

Para la RSL actual no se cumplen todos los pasos propuestos por Kitchenham, precisamente porque su metodología es flexible, ya que el investigador puede ampliar su revisión a su complacencia y el alcance que necesita dar a la presentación y publicación de su trabajo es flexible. Los resultados dependen del investigador. En consecuencia, en la **TABLA III**, se proponen las etapas y tareas a considerar:

TABLA III. Proceso de revisión sistemática de literatura propuesto [71]

Fases	Tareas
Planificación de la revisión	Identificación de la necesidad de una revisión.
	Especificación de las preguntas de investigación.
	El desarrollo de un protocolo de revisión.
Realizar la revisión	Identificación de la investigación.
	Selección de los estudios primarios.
	Síntesis de los datos
Revisión de informes	Dar formato al informe principal.

Las fases y tareas mencionadas anteriormente han sido realizadas para cumplir con la meta planeada en el alcance.

I. Planificación de la revisión: Definir el modelo de lenguaje BERT junto a SQuAD para generar un modelo que dé respuesta a las preguntas realizadas al agente virtual.

a. Identificación de la necesidad de una revisión

Descubrir el conocimiento que existe en torno a un tema de investigación a través de una revisión bibliográfica, permitió seleccionar los métodos, modelos, herramientas y software más adecuados para definir el modelo de lenguaje BERT, así como SQuAD, con el fin de generar un modelo que responde a la propuesta del agente virtual. Se han identificado ciertas preguntas de investigación para implementar y cumplir el alcance propuesto anteriormente.

b. Preguntas de investigación

El conocimiento existente sobre un tema específico se puede encontrar realizando una revisión sistemática de la literatura, en cuyo caso es necesario elegir la configuración y los

parámetros requeridos para desarrollar un agente virtual; de igual manera, comprender BERT y SQuAD que se utilizará en su construcción y sus diferentes versiones existentes. Para abarcar el tema, se presentan las siguientes preguntas de investigación para obtener información concisa y relevante.

En la **TABLA IV**, se presentan las preguntas de investigación que guiaron el desarrollo de la revisión sistemática de literatura, estas preguntas son:

TABLA IV. Preguntas de investigación

Preguntas de investigación	
P1	¿Cuáles son las características principales sobre el funcionamiento del modelo BERT y SQuAD?
P2	¿Cuáles son los parámetros que se emplean en los modelos de BERT con SQuAD?
P3	¿Qué versión de los modelos BERT y SQuAD es factible para el desarrollo del agente virtual?

c. El desarrollo de un protocolo de revisión

I. Diseño del protocolo de búsqueda

a. Estrategias de búsqueda

Petticrew y Roberts [72], recomendaron un enfoque estándar para las encuestas, mediante el uso de la regla nemotécnica PICOC (Población, Intervención, Comparación, Resultado y Contexto), se estructuran cinco componentes para definir la cadena de búsqueda para una mejor organización y selección de resultados. Se utilizó la herramienta en línea Parsifal, que facilita una revisión sistemática de literatura en el contexto de la ingeniería de software, gracias a la cual se puede monitorear la RSL, la misma que está diseñada para cubrir todos los campos necesarios y obtener el mejor resultado.

b. Fuentes bibliográficas

Como fuentes bibliográficas, se ha seleccionado algunas bibliotecas virtuales, como son:

- ACM Digital Library (<https://dl.acm.org/>)
- IEEE Digital Library (<https://ieeexplore.ieee.org/>)
- Google Scholar (<https://scholar.google.com/>)
- Scopus (<http://www.scopus.com>)

c. Definir palabras claves para el problema de estudio

Con la definición de los criterios PICOC se obtuvo un conjunto de palabras claves, las

mismas que permitieron construir las cadenas de búsquedas, estas son:

- BERT
- Bidirectional Encoder Representations from Transformer
- Configuration
- Model trained
- Parameters training
- SQuAD
- Stanford Question Answering Dataset
- Transformers parameters.

d. Cadenas de búsqueda

En la **TABLA V** se presentan las cadenas de búsqueda aplicadas que se realizaron de acuerdo a cada biblioteca virtual: ACM Digital Library (ver Figura 24), IEEE Digital Library (ver Figura 25), Google Scholar (ver Figura 26) y Scopus (ver Figura 27).

TABLA V. Scripts de búsqueda

Bibliotecas virtuales	Cadenas de búsqueda
ACM Digital Library	[[Publication Title: bert] OR [Publication Title: squad]] AND [All: performance of machine learning models] AND [Keywords: bert] AND [Abstract: answers to questions]
IEEE Digital Library	"BERT" AND "SQuAD" and "configuration" AND "training" AND "parameters" AND "Bidirectional Encoder Representations from Transformers" AND "Stanford Question Answering Dataset" AND performance of machine learning models
Google Scholar	"BERT" AND "SQuAD" and "configuration" AND "training" AND "parameters" AND "Bidirectional Encoder Representations from Transformers" AND "Stanford Question Answering Dataset" AND performance of machine learning models
Scopus	(TITLE-ABS-KEY (bert AND bidirectional AND encoder AND representations AND from AND transformer) OR TITLE-ABS-KEY (squad) AND TITLE-ABS-KEY (transformer AND models) OR TITLE-ABS-KEY (question AND answering AND systems) OR TITLE-ABS-KEY (performance AND of AND the AND bert AND models) AND TITLE-ABS-KEY (performance AND of AND machine AND learning AND models)) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018)) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English"))

The screenshot shows the ACM Digital Library search interface. The search bar at the top contains the query: "[Title: bert] OR [Title: squad]] AND [All: performance of machine learning models] AND [Keywords: bert] AND [Abstract: answers to questions]". Below the search bar, there are links to "Search ACM Digital Library" and "Register". The main content area displays 8 results. The first result is a research article titled "Chinese Intention Recognition Algorithm Based on BERT-GRU-Capsule Network" by Yang Xia and Chaobing Huang, published in November 2021. The results are filtered by "PEOPLE". On the left sidebar, there are filters for "Names", "Institutions", and "Authors" under the "People" section; and "Proceedings/Book Names", "All Publications", "Content Type", "Media Formats", and "Publisher" under the "Publications" section. The "Conferences" section is also partially visible.

Figura 24. Búsqueda ACM

The screenshot shows the IEEE Xplore search results page. The search bar at the top contains the query: ("Publication Title":BERT) AND ("Abstract":SQuAD) AND ("Abstract":configuration) OR ("Abstract":training) AND ("Abstract":parameters) AND ("Abstract":Bidirectional Encoder Representations from Transformers) OR ("Abstract":Stanford Question Answering Dataset) AND ("Abstract":performance of machine learning models). The results are shown for pages 1-8, with 8 total results. The first result is a conference paper titled "Can Knowledge Enhance Reading Comprehension? An Integrated Approach with Semantic Lexicon" by Kong Wei Kun, Teeradaj Racharak, and Le-Minh Nguyen, presented at the 2020 12th International Conference on Knowledge and Systems Engineering (KSE). The second result is another conference paper titled "A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization" by Ming-Hsiang Su, Chung-Hsien Wu, and Hao-Tse Cheng, from the IEEE/ACM Transactions on Audio, Speech, and Language Processing. The search interface includes filters for "Conferences" (6) and "Journals" (2), and an "Apply" button. There are also filters for "Show" (All Results selected, Open Access Only unselected), "Year" (Single Year selected, Range unselected), and sorting options ("Sort By: Relevance").

Figura 25. Búsqueda IEEE

The screenshot shows a Google Scholar search results page. The search query is: "BERT" AND "SQuAD" and "configuration" AND "training" AND "parameters". There are approximately 47 results found in 0.04 seconds. The results are listed in descending order of relevance. Each result includes a title, a brief abstract, the source (e.g., arxiv.org, diva-portal.org), and options to download as PDF or BibTeX, and to cite or import. The sidebar on the left provides filtering options for date (2020-2021), relevance, date, language (Spanish), type (revision articles), and alert creation.

Figura 26. Búsqueda Google Scholar

The screenshot shows a Scopus search results page. The search query is: (TITLE-ABS-KEY(bert AND bidirectional AND encoder AND representations AND from AND transformer) OR TITLE-ABS-KEY(squad) AND TITLE-ABS-KEY(transformer AND models) OR TITLE-ABS-KEY(question AND answering AND systems) OR TITLE-ABS-KEY(performance AND of AND the AND bert AND models) AND TITLE-ABS-KEY(performance AND of AND machine AND learning AND models)) AND (LIMIT-TO(PUBYEAR, 2021) OR LIMIT-TO(PUBYEAR, 2020) OR LIMIT-TO(PUBYEAR, 2019) OR LIMIT-TO(PUBYEAR, 2018)) AND (LIMIT-TO(DOCTYPE, "ar")) AND (LIMIT-TO(LANGUAGE, "English"))). There are 52 document results. The results are displayed in a table with columns for Document title, Authors, Year, Source, and Cited by. The first result is: "Text Mining of Adverse Events in Clinical Trials: Deep Learning Approach" by Chopard, D., Treder, M.S., Corcoran, P., Busse, M., Spasic, I. from JMIR Medical Informatics 9(12), e28632, 2021, with 0 citations.

Figura 27. Búsqueda Scopus

e. Criterios de inclusión

Durante la búsqueda, se consideraron los siguientes criterios de inclusión:

- Solo documentos que sean artículos científicos.
- Se mencione BERT.
- Contenga características de BERT y SQuAD

- Idioma: inglés-español
- Publicaciones a partir del año 2020.

f. Criterios de exclusión

- Que no aporten información relevante sobre BERT o SQuAD.
- Que no contesten las preguntas de investigación.
- Que no mencionen los parámetros de configuración de los modelos.
- Artículos redactados en idiomas que no sean en inglés o español.
- Publicaciones anteriores al año 2020.

II. Realizar la revisión

a. Identificación de la investigación

El propósito de esta revisión sistemática de la literatura es responder a la pregunta de investigación planteada mediante la búsqueda de estudios primarios que brinden información verídica con respecto al tema.

b. Selección de los estudios primarios

Para realizar la selección de los estudios se ha seguido el proceso que a continuación se muestra en la Figura 28.

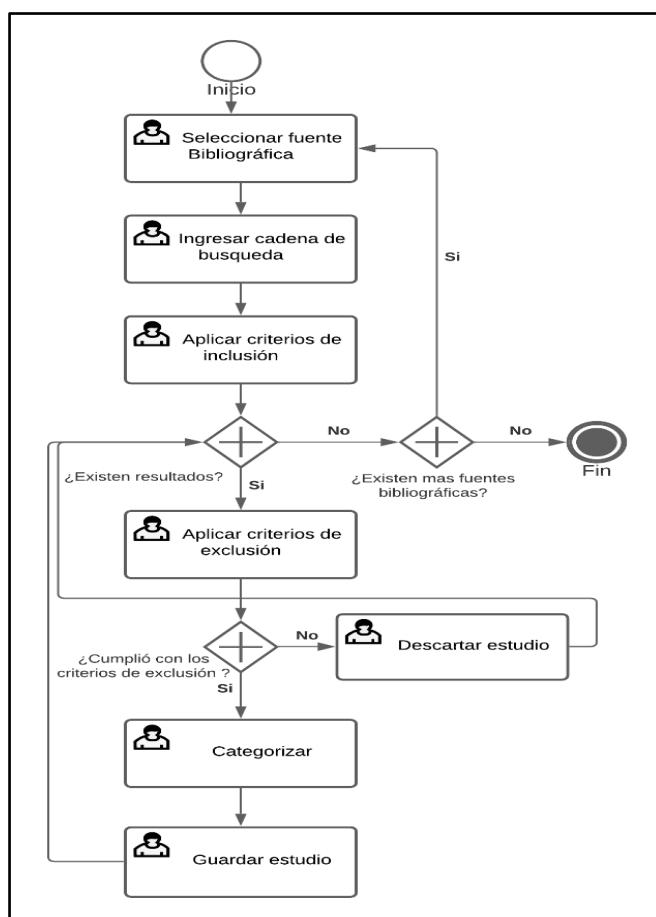


Figura 28. Proceso de selección de artículos

En la **TABLA VI** se presenta un resumen de los trabajos o estudios relacionados que fueron encontrados junto con el número de estudios seleccionados según la fuente bibliográfica.

TABLA VI. Resumen de estudios seleccionados

Fuente	Encontrados	Seleccionados
ACM Digital Library	4	2
IEEE Digital Library	5	3
Google Scholar	47	9
Scopus	52	18
Total	108	32

c. Síntesis de los datos

Cada artículo fue analizado para identificar su aporte más relevante y así finalmente obtener información relacionada a los modelos BERT y SQuAD requeridos que contestan a las preguntas de investigación inicialmente planteadas (ver **TABLA IV**).

a) ¿Cuáles son las características principales sobre el funcionamiento del modelo BERT y SQuAD?

De acuerdo a los estudios [30] [31] [37] [7] [43] [44] [45] [46] [47] [73] las características principales sobre el funcionamiento del modelo BERT y SQuAD son:

Para BERT:

- Se puede desarrollar en los Frameworks de Tensorflow y PyTorch.
- Dependiendo del framework usado se puede programar en los lenguajes Python (PyTorch y Tensorflow) o C++ (Tensorflow).
- Consta de un entrenamiento previo y ajuste fino, donde, se siguen dos pasos para aplicar BERT que incluyen preentrenamiento de un modelo BERT utilizando grandes corpus sin etiquetar y afinando el modelo preentrenado utilizando corpus anotados específicos de la tarea.
- Se recomienda ajustar la versión más pequeña de BERT en una tarjeta gráfica con 12 GB de VRAM. Donde el entrenamiento toma aproximadamente 2 días y el modelo base reporta un f1 de 65.28% en el conjunto de datos SQuAD 2.0.
- El entrenamiento de BERT Large en 16 Cloud TPU toma 4 días.
- El tiempo de refinamiento para BERT Large usando una GPU RTX 2080 Ti es de 1 día aproximadamente.
- BERT Large, en el ajuste fino es inestable en conjuntos de datos pequeños. Pero BERT Large logra un aprovechamiento de mayor precisión de tres veces y media más

precisión más parámetros.

En base a los artículos obtenidos se ha podido constatar que la mayoría de proyectos emplean como lenguaje Python junto a Pytorch [38] por su vinculación y puntos fuertes en el desarrollo de este tipo de proyectos [46]. Además, emplean BERT Base por los recursos limitados de la GPU y el tiempo que se toma en entrenar al modelo es más corto y la precisión no varía demasiado en comparación con BERT Large [30].

b) ¿Cuáles son los parámetros que se emplean en los modelos de BERT con SQuAD?

En base a los artículos seleccionados y estudiados, se constató que los siguientes parámetros son los más empleados como se mencionan en [30] [31] [32] [8] [34] [35] [36] [37] [38] [74] [39] [75] [7] [41] para trabajos en desarrollo.

- BERT Base emplea los siguientes parámetros previamente entrenados que son los siguientes: son $L = 12$, $H = 768$, $A = 12$, Parámetros totales = 110M, donde L es el número de capas (es decir, bloques de transformadores), H es el tamaño oculto y A es el número de cabezas de atención propia. El tamaño del filtro de alimentación hacia adelante se establece en $4H$, es decir, 3072 para $H = 768$.
- Se emplean métricas de evaluación como precisión micro promediada, recuperación y puntuación F1.
- BERT toma una secuencia de tokens con una longitud máxima de 512 y produce una representación de la secuencia en un vector de 768 dimensiones.
- Teniendo en cuenta la memoria de la GPU, si se elige BERT Base con el modelo uncase para un ajuste fino se puede evitar el desbordamiento de memoria de la GPU. De acuerdo con la sugerencia oficial de BERT, establece una la longitud máxima de la oración de 64, mini-lote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3, para el entrenamiento del modelo.
- En BERT, los datos se pasaron a tres capas de incrustación que incluyen una incrustación de token, segmento y posición capas. En el primer paso del procesamiento, las oraciones se tokenizaron y después de eso, cada token de entrada se pasó a través de una capa de incrustación de tokens para transformarlo en una representación vectorial de dimensión fija (es decir, vector de 768 dimensiones). Además, se agregaron tokens de clasificación adicional [CLS] y separador [SEP] al inicio y al final de la oración tokenizada para servir como una representación de entrada y un separador de oración para la tarea de clasificación. La capa de incrustación de segmentos ayuda a clasificar un texto dado un par de textos de entrada.
- Cabe afirmar que cada capa cumple una función de filtrado para lograr encontrar patrones en los datos que contienen datos invisibles a medida que pasa por cada capa,

la cual se basa en los patrones de la capa anterior, todo esto empleando los tokens creados en cada palabra al inicio de la primera capa en la creación de los tokens.

- Emplean la biblioteca PyTorch-Pretrained-BERT para construir el modelo BERT. Luego, ajustan su capa lineal (mediante la adición de nuevas palabras junto a la creación de sus respectivos tokens, mediante codificación de los parámetros mencionados en el anterior punto) y la activación sigmoidea (no importa cuál sea la entrada, la salida obtenida está entre valores de 0 y 1) para obtener las predicciones con el conjunto de datos etiquetado COVID-19. Durante el proceso de ajuste fino, se usaron el optimizador Adam con una tasa de aprendizaje de 3×10^{-6} y un tamaño de lote de 12. Y se ajustó el modelo en el conjunto de datos COVID-19 recopilado durante tres épocas, basado en BERT Base mencionado en el primer punto.

Dichos parámetros se consideraron como factibles para emplearlos en el entrenamiento del modelo BERT. De los estudios que contribuyeron a dar luz a la pregunta de investigación, fue el de Yanling Zhou [5], ya que evalúa el rendimiento de los diferentes modelos como el de ElMo y BERT entre otros, donde el que sigue destacando por encima de los demás es BERT con una precisión del 70 % y en el parámetro F1 del 62%, ofrece un margen considerable para poder mejorar con entrenamientos adicionales en modelos pre entrenados y enfocados a un tema en particular, [42] en el conjunto de [38] datos SQuAD 2.0. Cabe destacar que para el ajuste de cada modelo se basa en diferentes parámetros según el propósito que se tenga y donde se debe hacer diferentes ajustes para llegar a una configuración optima mediante los entrenamientos.

c) ¿Qué versión de los modelos BERT y SQuAD es factible para el desarrollo del agente virtual?

De acuerdo a los estudios [30][31][32][76][33][34][35][36][37][38][74][39] las principales versiones existentes de BERT son:

- BERT Base: Se compone de 12 bloques transformadores, 12 cabezas de atención y 110 millones de parámetros, el cual es más empleado con recursos limitados de hardware.
- BERT Large: Se compone de 24 capas, 16 cabezas de atención y 340 millones de parámetros, es más empleado en grandes equipos de hardware con alta capacidad computacional. Lo que dificulta su aplicación en tiempo real sin aceleración por hardware, como GPU y TPU.
- BERT Large emplea una cantidad de almacenamiento de 1,12 GB de registros, frente a BERT Base que sacrificando precisión en los resultados alcanza los 326 MB destinados en registros. Emplea parámetros de punto flotante 32b.

En cambio para SQuAD según [8] [75] [7] [41] menciona que:

- SQuAD 1.0: se trata de un conjunto de datos que contiene respuestas a preguntas que consta de más de 100.000 pares de preguntas, junto al párrafo relevante correspondiente que contiene a respuesta. Pero no es factible ya que para preguntas sin respuesta no tiene definido una respuesta lo que da lugar a malas interpretaciones sobre las respuestas.
- SQuAD 2.0: Permite entrenar modelos para que se abstengan de responder preguntas sin respuesta, la tarea de SQuAD 2.0 amplía la definición del problema de SQuAD1.1 al permitir la posibilidad de que no exista una respuesta corta en el párrafo proporcionado, lo que hace que el problema sea más realista. Por lo que es más factible emplear SQuAD 2.0, ya que, como se menciona es más realista por su funcionalidad antes preguntas que no tienen respuesta.

Dichas versiones tales como BERT Base y BERT Large [42] tienen diferentes contrastes dependiendo básicamente del hardware con el que se cuente para el desarrollo de los modelos. Cabe mencionar que, en los artículos encontrados la mayoría da por hecho el uso de modelos preentrenados, de los cuales solo se va mejorando a través de nuevos entrenamientos y parámetros. Por lo tanto, la versión de BERT base es más factible, ya que no demanda un hardware con altas prestaciones.

SQuAD posee varias versiones, pero las más usadas son SQuAD 1.0 [41] y SQuAD 2.0 [43], las que se diferencian básicamente en las preguntas sin respuesta que ofrece la última versión SQuAD 2.0 y se asemeja a la realidad por su función. Por lo que para que sea más fiel a la realidad, es más factible SQuAD 2.0, ya que combina el conjunto de datos de SQuAD 1.0 con preguntas sin respuesta.

d. Conclusión general de la Revisión de literatura

Como características principales sobre el funcionamiento del modelo BERT y SQuAD, se concluyó que se emplea el lenguaje Python junto a framework PyTorch para desarrollar en la nube o con la ayuda de una GPU, empleando además el modelo BERT Base, para tener un mejor desempeño si se posee hardware con recursos limitados, destacando que, se debe realizar un entrenamiento previo y ajuste fino posterior del modelo para su optimización. Los parámetros más destacados para proyectos relacionados con BERT y que no supongan un procesamiento extremadamente grande teniendo en cuenta los escasos recursos del hardware son el uso de tres capas de incrustación junto al modelo BERT base case, ya que debe procesar menos datos y establece una la longitud máxima de la oración de 64, minilote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3 para el entrenamiento del modelo. Para la elaboración del agente conversacional basado en BERT y SQuAD se concluyó que se usará el modelo de BERT base junto a SQuAD 2.0, con a una GPU Nvidia para entrenar de manera física o

en la nube empleando Google Colab, ya que también cuenta con tarjetas gráficas Nvidia.

III. Revisión de informes

a. Dar formato al informe principal

Como informe final de la revisión bibliográfica realizada, se redactó un artículo como producto final (ver Anexo1).

6.1.2. Tarea 2: Extracción de información de artículos relacionados con las características y el funcionamiento respecto al empleo de BERT y SQuAD.

Para la extracción de información sobre las características y el funcionamiento se utilizaron los artículos de la revisión sistemática de literatura anterior, mediante la extracción de datos para cada una de las preguntas de investigación. La cual sirvió de referencia para el ajuste del modelo BERT con nuestro conjunto de datos que se aborda en la Tarea 3. Donde se obtuvo las siguientes características principales:

- En base a los trabajos obtenidos mediante la RSL, los autores mencionan la versión BERT Base como la empleada por temas de recursos relacionados con los proyectos. Y SQuAD 2.0 por que combina el conjunto de datos de SQuAD 1.0 con preguntas sin respuesta, se enfoca más en preguntas de extrema confusión lo que asemeja la realidad y permite una adaptabilidad a más contextos. Por lo que se empleará BERT base junto a SQuAD 2.0 para dar respuestas a preguntas mediante el agente conversacional.
- Las características principales sobre el funcionamiento del modelo BERT y SQuAD, son el empleo del lenguaje Python junto a framework PyTorch que se pueden desarrollar en la nube o con la ayuda de una tarjeta gráfica con 12 GB de VRAM, para tener un mejor desempeño. Además de realizar un entrenamiento previo y ajuste fino posterior.
- Los parámetros más destacados para proyectos relacionados con BERT y que no supongan un procesamiento extremadamente grande teniendo en cuenta los escasos recursos del hardware son el uso de tres capas de incrustación junto al modelo BERT base uncase, ya que debe procesar menos datos y establece una la longitud máxima de la oración de 64, mini-lote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3 para el entrenamiento del modelo.

Para la elaboración del agente conversacional basado en BERT y SQuAD se concluyó que se usará el modelo de BERT base junto a SQuAD 2.0, con a una GPU Nvidia o Google Colab. Y los parámetros de acuerdo con la sugerencia oficial de los creadores de BERT, que establecen una la longitud máxima de la oración de 64, mini-lote de 32, una la tasa de aprendizaje de 2e-5 con un número de épocas de entrenamiento de 3 para el entrenamiento del modelo.

6.1.3. Tarea 3: Selección de artículos destacados respecto a temas tópicos

sobre el covid-19.

Para la presente tarea se realizó una revisión sistemática de literatura para identificar los temas, preguntas y palabras clave en torno a las búsquedas relacionadas con el Covid-19, con el fin de poder obtener preguntas para el ajuste del modelo con temas que estén en la línea de investigación. Además, se seleccionó un conjunto de datos (dataset) de investigación abierta (Open Source) de COVID-19 destinado para proyectos de procesamiento de lenguaje natural (PLN) y otras técnicas con el fin de luchar contra la pandemia, llamado CORD-19, el cual se compone de más de 500.000 artículos académicos sobre COVID-19, SARS-CoV-2 y coronavirus. Está respaldado por la página web Kaggle en conjunto con Semantic Scholar y el Instituto Allen de IA en asociación con la Iniciativa Chan Zuckerberg, el Centro de Seguridad y Tecnología Emergente de la Universidad de Georgetown, Microsoft Research, IBM y la Biblioteca Nacional de Medicina - Institutos Nacionales de Salud, en coordinación con la Casa Blanca, como se visualiza en la Figura 29. Una vez concluida la búsqueda se optó por la base de datos científica CORD-19 y las preguntas obtenidas en la revisión sistemática para poder ajustar el modelo BERT en temas tópicos sobre el covid-19.

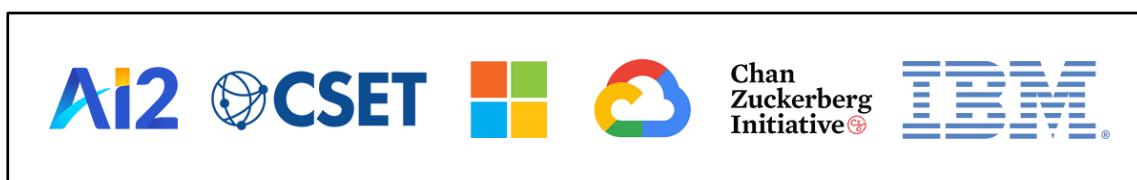


Figura 29. Creadores de Dataset CORD-19 [67]

I. Planificación de la revisión: Selección de artículos destacados respecto a temas tópicos sobre el covid-19.

a. Identificación de la necesidad de una revisión

Descubrir el conocimiento que existe en torno a un tema de investigación a través de una revisión bibliográfica, permitió seleccionar los métodos, modelos, herramientas y software más adecuados para definir el modelo de lenguaje BERT, así como SQuAD, con el fin de generar un modelo que responde a la propuesta del agente virtual. Se han identificado ciertas preguntas de investigación para implementar y cumplir el alcance propuesto anteriormente.

b. Preguntas de investigación

En la **TABLA VII**, se presentan las preguntas de investigación que guiaron el desarrollo de la revisión sistemática de literatura, estas preguntas son:

TABLA VII. Preguntas de investigación

Preguntas de investigación

P1	¿Cuáles son los temas más abordados relacionados con el covid-19?
P2	¿Qué preguntas o palabras clave son las más comunes acerca del covid-19?

c. El desarrollo de un protocolo de revisión

Consultar Anexo 2 (IV. Resultados, L. El desarrollo de un protocolo de revisión)

I. Realizar la revisión

a. Síntesis de los datos

Cada artículo fue analizado para identificar su aporte más relevante y así finalmente obtener información relacionada los temas y preguntas más frecuentes acerca del covid-19, que contestan a las preguntas de investigación inicialmente planteadas (ver **TABLA VII**).

a) ¿Cuáles son los temas más abordados relacionados con el covid-19?

El tema principal según [51] fue “coronavirus” en cuanto a temas para luchar contra el coronavirus mediante medidas preventivas, el término más buscado fue “quedarse en casa”, seguido de “máscaras faciales”, “distanciamiento social” y “lavarse las manos”.

En [52] se utiliza la herramienta Google Trends para establecer los temas informativos más consultados asociados al CoViD-19. Las palabras “coronavirus

Ecuador” y “Covid Ecuador” son las más utilizadas para indagar información en torno a la pandemia. Por otro lado, tratamientos con el uso de sustancias como la “hidroxicloroquina” y el “dióxido de cloro”, presentan búsquedas medianamente importantes. Además, las búsquedas mediante las palabras “vacuna coronavirus” o “Covid vacuna” así como “coronavirus síntomas” presentan diversos rangos de importancia.

En [53] se evidenció que a medida que los casos por COVID-19 se reportaban en Wuhan y Latinoamérica, el nivel de búsqueda en Google Trends para los términos “coronavirus”, “lavado de manos” y “mascarillas” aumentaban.

En [57] con ayuda de Google Trends la cual proporciona el número de búsquedas relativas dentro de una determinada región o globalmente para una consulta en particular. Los datos recopilados de Google Trends se ajustan a la hora y la ubicación, por lo que las comparaciones entre consultas pueden ser más fáciles. Los principales términos de búsqueda entre países de habla inglesa que se estudiaron fueron “2019-nCoV + SARS-CoV-2 + nuevo coronavirus + nuevo coronavirus + COVID-19 + corona virus de Enfermedades 2019”.

En [58] demuestra un fuerte aumento en el interés de las búsquedas en Internet sobre “fiebre”, “tos” y “diarrea” al inicio de la pandemia de COVID-19 en los Estados Unidos antes de la identificación de casos, mientras que la búsqueda relativa de muchos síntomas, que

incluyen “náuseas”, “vómitos”, “dolor abdominal” y “estreñimiento”, permanecieron cerca o por debajo de la línea de base durante el período de tiempo estudiado.

En el artículo [59] se menciona que las palabras clave en las consultas son: “casos de coronavirus”, “covid19” y “casos de covid19”. El coronavirus se relacionó principalmente con “casos”, “reino unido”, “síntomas”, “noticias” y “actualización”. El término “covid” estaba relacionado con el número “19”. Además, la palabra clave (casos de covid19) se produjo más en los términos de esta palabra clave compuesta con sus términos (covid19). El número entero “19” se correlacionó con el término “casos”.

En [54], los temas de Google Trends relacionados con el “coronavirus”, el “lavado de manos” y las “mascarillas” representaban las principales búsquedas de información sobre COVID-19, así como la práctica de la higiene personal para prevenir la transmisión de enfermedades.

b) ¿Qué preguntas o palabras clave son las más comunes acerca del covid-19?

En el artículo [51] menciona las palabras clave en base a los síntomas más buscados, donde fueron “fiebre”, “dolor de garganta” y “tos”, además, el interés de los usuarios por conocer las vías de transmisión del síndrome respiratorio agudo coronavirus 2, que genera hasta en la actualidad debates.

En [49] las principales preguntas abordadas en torno al tema del covid-19 son acerca de las pruebas de diagnóstico de esta enfermedad, entra las que se destacan: ¿Qué tan precisas son las pruebas de RTPCR?, ¿Qué tipos de muestras son adecuadas para realizar pruebas?, ¿Una muestra negativa descarta COVID-19?, ¿Vale la pena repetir una prueba después de que un paciente haya dado negativo?, ¿Cuánto tiempo permanecerán positivos los pacientes infectados?, ¿Una muestra positiva significa que un paciente es infeccioso?, ¿Afecta la edad a la carga viral?, ¿Qué tan pronto se desarrollan los anticuerpos?.

En el artículo [50] se mencionan y se dan respuestas a algunas de las preguntas más frecuentes relacionadas con el covid-19 entre las cuales tenemos: ¿Qué es COVID 19?, ¿De dónde provienen los coronavirus?, ¿Cuáles son los modos de transmisión del SARS CoV2?, ¿Cuáles son los síntomas comunes de COVID-19?, ¿Cómo puedo protegerme de la infección por COVID19?, ¿Qué es el distanciamiento social/físico?, ¿Cómo previene la propagación de COVID-19?, ¿Qué hacer si tuve un contacto con un paciente con COVID-19?, entre otras comunes.

En el artículo [55] se responden a preguntas frecuentes acerca del covid-19, relacionadas con información de la pandemia, métodos de prevención del covid, tratamientos, síntomas entre otros. Entre estas preguntas se destacan: ¿Qué es una pandemia?, ¿Qué es el distanciamiento social?, ¿Existe alguna vacuna, medicamento o tratamiento para COVID-

19?, ¿Cuál es la tasa de letalidad?, ¿Es lo mismo que la tasa de mortalidad por infecciones?, ¿El COVID-19 afecta a las personas mayores o las personas más jóvenes también son susceptibles?

En [58] los temas de búsqueda más destacados fueron los relacionados con síntomas gastrointestinales asociados con la infección por COVID-19 en la literatura, como "diarrea", "náuseas", "vómitos" y "dolor abdominal". Los términos "fiebre" y "tos" se incluyeron como destacados. Los términos "anosmia", "disgeusia", "pérdida del apetito", "pérdida del gusto" y "pérdida del olfato" fueron considerados; sin embargo, existió una baja frecuencia de búsquedas de estos términos.

En el artículo [59] la categoría más popular de búsqueda fue "Virus" la mayoría de los días de este período. La segunda categoría de búsqueda importante fue "Agente infeccioso". La categoría "Enfermedad" se presentó más a finales de febrero hasta finales de la segunda semana de marzo. Aunque las categorías de Asia ("País de Asia Oriental" y "País de Asia") fueron tendencias después del 11 de febrero, la categoría de Asia Oriental también fue frecuente a partir de la cuarta semana de enero.

En [60] mencionan un total de 15 de los términos de búsqueda de síntomas tenían correlación estadísticamente significativos con casos confirmados de COVID-19. En la lista de terminología de síntomas considerada fue: "fiebre", "dificultad para respirar", "tos", "anosmia", "fatiga", "rinorrea", "congestión nasal", "estornudo", "mialgia", "dolor de garganta", "diarrea", "anorexia", "dolor de pecho", "esputo", "dolor de cabeza", "náuseas", "ageusia", "dolor abdominal", "mareos", "vómitos" y "dolor ocular".

b. Conclusión general de la Revisión de literatura

De acuerdo a la revisión realizada sobre artículos relacionados con temas tópicos del covid-19, se han obtenido las siguientes conclusiones:

- En base a la RSL y los trabajos obtenidos, para los temas más abordados relacionados con el covid-19, se puede apreciar que los autores mencionan como tema principal la palabra coronavirus o covid-19, es el tema central de las búsquedas en la web y le preceden las búsquedas sobre medicación para el tratamiento, junto con búsquedas sobre vacunas y los síntomas principales que se presentan con este virus como fiebre, tos, dolor abdominal que estuvieron con alta demandas de consultas. También se apreció que a medida que aumentaban los casos en China las búsquedas sobre medidas de prevención se intensificaban, lo que ofrece una oportunidad de poder crear un sistema que ofrezca esta información de una manera asociada, confiable y precisa, además de la que ofrece la herramienta Google Trends.
- Las palabras clave más empleadas en búsquedas relacionadas con la sintomatología del covid-19 fueron: "fiebre, dolor de garganta, dificultad para respirar, náuseas, dolor ocular,

fatiga y tos” junto al interés de saber cuáles son las vías de transmisión, por lo que se podría ayudar a la población a dar un diagnóstico más fiable en base a las consultas que más generan incertidumbre sin la necesidad de recurrir a un especialista en primera instancia y si no concuerda designar a la persona a un centro médico para una atención oportuna.

- Se ha concluido que las preguntas que han realizado los usuarios entorno al covid-19 son acerca de que es el covid-19, el lugar de procedencia, síntomas, distanciamiento, prevención de la propagación, pruebas de diagnóstico, su precisión, tiempo, resultados, estadísticas de mortalidad y anticuerpos. Con lo que se puede deducir que mientras la pandemia exista, seguirán surgiendo dudas, las cuales tienen que ser confiables y concisas ya que la desinformación también causa un pánico innecesario en la población.
- En general la búsqueda de todos estos artículos que se relacionan con el covid-19 y la búsqueda de información que ha realizado la población para informarse sirve para poder ofrecer a la población una información con una categoría superior a la que se puede encontrar en cualquier página web, aportando más confianza respecto a los temas que se consultan, evitando que tengan que acudir a un médico sabiendo la exposición que esto implica en cualquier persona frente al coronavirus.

b. Revisión de informes

a. Dar formato al informe principal

Como informe final de la revisión bibliográfica realizada, se redactó un artículo como producto final (ver Anexo 2).

6.2. Objetivo 2: Desarrollar un agente virtual empleando NLP (Natural Language Processing) con el modelo de lenguaje BERT junto al conjunto de datos SQuAD.

Para el Objetivo 2, se empleó la cuarta fase de la metodología Design Thinking, el cual facilitó la creación del prototipo, basado en las anteriores fases abordadas en el objetivo uno donde se enfatizó en la necesidad de las personas, se definió lo que necesitaban las personas y se ideó la alternativa de un agente conversacional, esto permitió consumar la elaboración de un prototipo.

6.2.1. Entrenar el modelo de lenguaje BERT, con la información de artículos científicos sobre información general del covid-19.

Para el entrenamiento del modelo, como ya se mencionó anteriormente se utilizó el lenguaje de programación Python junto con la librería de aprendizaje automático Pytorch y Transformers la cual proporciona arquitecturas de uso general (en donde se encuentra el modelo BERT) para la comprensión del lenguaje natural.

Para la obtención de los datos de entrenamiento se utilizó la herramienta Haystack Annotation Tool [77] la cual permite seleccionar preguntas y respuestas en diferentes contextos como se observa en la Figura 30, luego descargarlos en diferentes formatos, en este caso el formato SQuAD, que contiene: la pregunta con su respectiva respuesta, su posición en el contexto y el contexto. Además utilizamos el dataset Covid_SQuAD el cual es proyecto de código abierto diseñado que contiene diferentes preguntas y respuestas relacionadas con el Covid-19 [78].

The screenshot shows the Haystack annotation tool interface. At the top, there is a navigation bar with options like 'Bert' (selected), 'Documents', 'Questions', 'Import', 'Export Labels', and an email address 'victor.serrano@unl.edu.ec'. Below the navigation bar, the URL 'Bert / Labels' is displayed. The main area is titled 'All project labels' and shows a table of 'All labels'. The table has columns for 'Id', 'Question text', 'Answer', and 'Actions'. There are six rows of data:

Id	Question text	Answer	Actions
55995 9	what is covid-19?	a global pandemic	
55996 0	what is covid-19?	global pandemic	
55996 2	How to prevent covid-19?	mask/covering when outside	
55996 3	How to prevent covid-19?	social distancing	
55996 4	how to protect myself from covid-19?	face mask/covering when outside	
55996 5	how to protect myself from covid-19?	social distancing behaviors	

A blue button 'Export answers' is located at the top right of the table area.

Figura 30. Herramienta haystack annotation tool [77]

Luego de obtener los datos, se inició con el afinado del modelo, para esto se utilizó la herramienta Google Colab, ya que permite la utilización de GPU's con mejores características. Antes de iniciar con el ajuste del modelo se empleó el método Fine Tuning, que permite la modificación de los ajustes prioritarios llamados hiperparámetros que juegan un papel imprescindible en cualquier ajuste de un modelo, los fundamentales son: en primer lugar, el learning Rate "lr" (Tasa de Aprendizaje): Este valor determina si el aprendizaje se propaga hacia atrás, lo que significa que tiende al valor mínimo, si el valor es mucho más alto de lo que debería ser, hará que el modelo diverja, lo que significa que cuanto menor sea la tasa de aprendizaje, más rápido converge el modelo; sin embargo, si el valor utilizado en esta variable es incorrecto, hará que el modelo no alcance el máximo nivel de precisión posible. En segundo lugar, tenemos el Batch Size (Tamaño del Lote): este valor es responsable de definir la cantidad de veces que iterará la entrada y que pasará por la red, es decir, divide el conjunto de datos en número de lotes o conjuntos o partes. Cuando los pesos convergen a un cierto valor, cualquier aumento más allá de este valor no mejorará la eficiencia del modelo, sino que

limitará su rendimiento. Como tercer hiperparámetro tenemos, Epoch (Épocas): se trata de la cantidad de ejemplos que se deben poner en la red al mismo tiempo para que se ajusten. Una época es igual a una iteración completa sobre los datos de entrenamiento. A medida que aumenta el número de épocas, cambia el peso en la red neuronal y la curva pasa de un ajuste insuficiente a una curva óptima e incluso a una curva de sobreajuste si se coloca un valor muy alto. Depende del conjunto de datos que se posea y de la divergencia de información que contenga. Por último, tenemos el Gradient Descent o Descenso de gradiente: Es un algoritmo de optimización iterativo utilizado en aprendizaje automático para encontrar el mejor resultado (mínimo de la curva). Los ajustes se basaron en la información obtenida de la revisión de literatura que permitió ser la base para el desarrollo del modelo ajustado descrito en la Figura 37.

El resultado del proceso de ajuste o Fine tuning se debe probar con distintos valores tanto como en lr, batch y epoch, ya que varía dependiendo del conjunto de datos y las características que tenga posea como los tipos de datos si son similares o distintos se debe obtener un promedio adecuado a través de diferentes pruebas, teniendo en cuenta los hiperparámetros obtenidos en las revisiones de literatura que se tomaron como base como por ejemplo una Tasa de Aprendizaje (LR) para el ajuste fino de BERT de 5e-5, 3e-5, 2e-5 [28][49], un tamaño de Lote (batch) que varía entre 5 a 12 y por último una cantidad de Épocas (Epoch) de entre 3 a 7, con las que se obtuvo diferentes valores buscando una pérdida mínima que pueda ofrecer resultados óptimos, las pruebas realizadas con distintas configuraciones se muestran en la **TABLA VIII**.

TABLA VIII. Hiperparámetros Ajuste (Fine Tuning)

Hiperparámetros del Ajuste en el Modelo Bert					
Nº	Hiperparámetros			Resultados	
	LR	Batch Size	Epoch	Pérdida	Figura
1	1e-5	7	7	0.323	Figura 31
2	2e-5	12	3	Error Memoria GPU	Figura 32
3	2e-5	10	3	0.151	Figura 33
4	5e-5	8	7	0.0114	Figura 34
5	5e-5	10	3	0.714	Figura 35
6	3e-5	5	5	0.063	Figura 36
7	2e-5	7	10	0.0539	Figura 37
8	5e-5	7	7	0.0114	Figura 38

Observamos que el aumento en la pérdida depende de los hiperparámetros, iniciando con una pérdida de 0.323 en la Figura 32, donde su LR es baja y esto influye en el aprendizaje, a

medida que aumenta el Batch se puede observar el fallo por el límite de la memoria de la GPU dado por el ajuste excesivo como se indica en la Figura 33. Estableciendo distintos valores en las configuraciones podemos obtener una pérdida aceptable pero no la deseada, como se puede ver en la Figura 34, Figura 35 o en la Figura 36, que si se compara con los resultados de la Figura 38 se observa que las épocas juegan un papel importante en el aprendizaje y esto se refleja en la pérdida final. El ajuste seleccionado y aplicado al modelo BERT fue la octava configuración por la pérdida de 0.0114 como se distingue en la Figura 39, ya que posee unos valores donde no se tuvieron ni sobreajustes ni pérdidas significativas.

Epoch 0: 100%	18/18 [00:26<00:00, 1.48s/it, loss=3.35]
Epoch 1: 100%	18/18 [00:26<00:00, 1.45s/it, loss=2.22]
Epoch 2: 100%	18/18 [00:25<00:00, 1.40s/it, loss=1.38]
Epoch 3: 100%	18/18 [00:25<00:00, 1.40s/it, loss=0.121]
Epoch 4: 100%	18/18 [00:25<00:00, 1.40s/it, loss=0.388]
Epoch 5: 100%	18/18 [00:25<00:00, 1.40s/it, loss=0.157]
Epoch 6: 100%	18/18 [00:25<00:00, 1.40s/it, loss=0.323]

Figura 31. Ajuste Modelo BERT Nº1

```

-----  

RuntimeError                               Traceback (most recent call last)  

<ipython-input-14-e2daed0ff441> in <module>()  

    15 device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')  

    16 # se mueve el modelo al dispositivo detectado  

--> 17 model.to(device)  

    18 # se activa el modo de entrenamiento del modelo  

    19 model.train()  

  
      5 frames  

/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py in convert(t)  

  848     return t.to(device, dtype if t.is_floating_point() or t.is_complex() else None,  

  849                 non_blocking, memory_format=convert_to_format)  

-> 850     return t.to(device, dtype if t.is_floating_point() or t.is_complex() else None, non_blocking)  

  851  

  852     return self._apply(convert)  

  
RuntimeError: CUDA out of memory. Tried to allocate 86.00 MiB (GPU 0; 11.17 GiB total capacity; 10.53 GiB already allocated; 69.81 MiB free;  
10.65 GiB reserved in total by Pytorch

```

Figura 32. Ajuste Modelo BERT Nº2

Epoch 0: 100%	13/13 [00:24<00:00, 1.89s/it, loss=2.25]
Epoch 1: 100%	13/13 [00:24<00:00, 1.87s/it, loss=2.01]
Epoch 2: 100%	13/13 [00:24<00:00, 1.86s/it, loss=0.151]

Figura 33. Ajuste Modelo BERT Nº3

Epoch 0: 100%	16/16 [00:25<00:00, 1.57s/it, loss=1.58]
Epoch 1: 100%	16/16 [00:24<00:00, 1.56s/it, loss=0.459]
Epoch 2: 100%	16/16 [00:24<00:00, 1.55s/it, loss=2.64]
Epoch 3: 100%	16/16 [00:24<00:00, 1.54s/it, loss=0.0261]
Epoch 4: 100%	16/16 [00:24<00:00, 1.54s/it, loss=0.201]
Epoch 5: 100%	16/16 [00:24<00:00, 1.54s/it, loss=0.0133]
Epoch 6: 100%	16/16 [00:24<00:00, 1.54s/it, loss=0.0114]

Figura 34. Ajuste Modelo BERT Nº4

Epoch 0: 100%	13/13 [00:24<00:00, 1.89s/it, loss=3.49]
Epoch 1: 100%	13/13 [00:24<00:00, 1.87s/it, loss=0.945]
Epoch 2: 100%	13/13 [00:24<00:00, 1.86s/it, loss=0.714]

Figura 35. Ajuste Modelo BERT Nº5

Epoch 0: 100%		25/25 [00:26<00:00, 1.05s/it, loss=2.88]
Epoch 1: 100%		25/25 [00:26<00:00, 1.05s/it, loss=2.06]
Epoch 2: 100%		25/25 [00:26<00:00, 1.04s/it, loss=0.0459]
Epoch 3: 100%		25/25 [00:26<00:00, 1.04s/it, loss=0.0477]
Epoch 4: 100%		25/25 [00:26<00:00, 1.05s/it, loss=0.063]

Figura 36. Ajuste Modelo BERT Nº6

Epoch 0: 100%		18/18 [00:25<00:00, 1.42s/it, loss=2.62]
Epoch 1: 100%		18/18 [00:25<00:00, 1.42s/it, loss=4.14]
Epoch 2: 100%		18/18 [00:25<00:00, 1.40s/it, loss=0.604]
Epoch 3: 100%		18/18 [00:25<00:00, 1.39s/it, loss=1.23]
Epoch 4: 100%		18/18 [00:25<00:00, 1.40s/it, loss=0.242]
Epoch 5: 100%		18/18 [00:25<00:00, 1.40s/it, loss=0.196]
Epoch 6: 100%		18/18 [00:25<00:00, 1.40s/it, loss=0.0184]
Epoch 7: 100%		18/18 [00:25<00:00, 1.39s/it, loss=0.00542]
Epoch 8: 100%		18/18 [00:25<00:00, 1.39s/it, loss=0.0106]
Epoch 9: 100%		18/18 [00:25<00:00, 1.39s/it, loss=0.0539]

Figura 37. Ajuste Modelo BERT Nº7

Epoch 0: 100%		49/49 [00:15<00:00, 3.17it/s, loss=0.387]
Epoch 1: 100%		49/49 [00:15<00:00, 3.16it/s, loss=1.79]
Epoch 2: 100%		49/49 [00:15<00:00, 3.18it/s, loss=0.225]
Epoch 3: 100%		49/49 [00:15<00:00, 3.17it/s, loss=0.0781]
Epoch 4: 100%		49/49 [00:15<00:00, 3.17it/s, loss=0.0149]
Epoch 5: 100%		49/49 [00:15<00:00, 3.17it/s, loss=0.00038]
Epoch 6: 100%		49/49 [00:15<00:00, 3.18it/s, loss=0.0114]

Figura 38. Ajuste Modelo BERT Nº8

A continuación, a cada archivo json de entrenamiento el cual está estructurado con una serie de preguntas y respuestas para cada contexto. Se lo divide en listas paralelas de contextos, preguntas y respuestas (Los contextos se repiten ya que pueden existir varias preguntas por contexto):

```
def read_squad(path):
    # se abre el archivo JSON y cargue el diccionario de introducción
    with open(path, 'rb') as f:
        squad_dict = json.load(f)
    # inicializar listas para contextos, preguntas y respuestas // título, pregunta, respuesta,
    # resumen (contextos)
    contexts = []
    questions = []
    answers = []
    # iterar a través de todos los datos
    for i in squad_dict["data"]:
        try:
            for x in i['paragraphs'][0]['qas']:
                for xx in x['answers']:
                    # print(x)
                    context = i['paragraphs'][0]['context']
                    question = x['question']
                    answer = xx
                    # comprueba si necesitamos extraer de 'answers' o 'plausible_answers'
                    if 'plausible_answers' in i.keys():
                        access = 'plausible_answers'
                    else:
```

```

        access = 'answers'

    # agregar datos a listas // título, pregunta, respuesta, resumen
    contexts.append(context)
    questions.append(question)
    answers.append(answer)

except:
    print("error")

# devolver listas de datos
return contexts, questions, answers

# se ejecuta la función de lectura SQuAD para conjuntos de entrenamiento y validación
train_contexts, train_questions, train_answers = read_squad(answers.json)
val_contexts, val_questions, val_answers = read_squad('COVID-QA_Answers_Validator.json')

```

Primero, se obtuvo la posición de la respuesta en el pasaje (se nos da la posición inicial) y con ella calculamos la posición final de la respuesta.

```

def add_end_idx(answers, contexts):
    # se recorre cada par respuesta-contexto
    for answer, context in zip(answers, contexts):
        # gold_text se refiere a la respuesta que esperamos encontrar en contexto
        gold_text = answer['text']
        # se conoce el índice de inicio
        start_idx = answer['answer_start']
        # idealmente este sería el índice final
        end_idx = start_idx + len(gold_text)

        # A veces las respuestas se desvían por un personaje o dos
        if context[start_idx:end_idx] == gold_text:
            # si la respuesta no es apagada
            answer['answer_end'] = end_idx
        else:
            # esto significa que la respuesta está desviada por 1-2 tokens
            for n in [1, 2]:
                if context[start_idx - n:end_idx - n] == gold_text:
                    answer['answer_start'] = start_idx - n
                    answer['answer_end'] = end_idx - n

    # y se aplica la función a nuestras dos listas de respuestas
add_end_idx(train_answers, train_contexts)
add_end_idx(val_answers, val_contexts)

```

A continuación, debemos convertir las posiciones de inicio y fin de la respuesta en posiciones de inicio y fin de token, además se inicializa el tokenizador en el que añadimos las configuraciones de padding y truncation, las cuales van a ser útiles para secuencias largas, en este caso la configuración truncation cortará secuencias demasiado extensas, y padding llenará las secuencias cortas para asegurarse que estas tengan el mismo tamaño.

```
max_length = 384 # La longitud máxima de cadena (pregunta y contexto)
```

```

doc_stride = 128 # La cantidad que se solapará entre dos partes del contexto si es necesario
# print(train_answers[:5])
from transformers import BertTokenizerFast
# inicializar el tokenizador
tokenizer = BertTokenizerFast.from_pretrained("deepset/bert-base-cased-squad2")
# tokenizador
train_encodings = tokenizer( train_contexts, train_questions, max_length=max_length, padding =
    True, truncation= True, stride=doc_stride)
val_encodings = tokenizer(val_contexts, val_questions, max_length=max_length, padding = True,
truncation= True, stride=doc_stride)
# print(tokenizer.decode(train_encodings['input_ids'][0]))

def add_token_positions(encodings, answers):
    # se inicializa las listas para contener los indices de token de inicio / final de respuesta
    start_positions = []
    end_positions = []

    for i in range(len(answers)):
        print(answers[i]['answer_start'])
        print(answers[i]['text'])
        # agregar la posición del token de inicio/final usando el método char_to_token
        start_positions.append(encodings.char_to_token(i, answers[i]['answer_start']))
        end_positions.append(encodings.char_to_token(i, answers[i]['answer_end']))
        # si la posición inicial es None, el pasaje de respuesta se ha truncado
        if start_positions[-1] is None:
            start_positions[-1] = tokenizer.model_max_length

        # si no se encuentra la posición final , se emplea char_to_token y cambiará
        # la posición hasta encontrarla
        shift = 1
        while end_positions[-1] is None:
            end_positions[-1] = encodings.char_to_token(i, answers[i]['answer_end'] - shift)
            shift += 1
    #Actualiza encodings con las nuevas posiciones de inicio/final basadas en tokens.
    encodings.update({'start positions': start_positions, 'end positions': end_positions})

    # Se aplica la función a los datos
add_token_positions(train_encodings, train_answers)
add_token_positions(val_encodings, val_answers)

```

Se colocan los datos en conjunto de datos de PyTorch para poder usarlos fácilmente para el entrenamiento.

```

import torch

class SquadDataset(torch.utils.data.Dataset):
    def __init__(self, encodings):
        self.encodings = encodings

```

```

    def __getitem__(self, idx):
        return {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
    def __len__(self):
        return len(self.encodings.input_ids)

train_dataset = SquadDataset(train_encodings)
val_dataset = SquadDataset(val_encodings)

```

Finalmente se realiza el entrenamiento y se procede a guardar el modelo entrenado. Aquí se inicializa el tamaño de lote, la tasa de aprendizaje y el número de épocas, con las seleccionadas en la RSL.

```

#Entrenamiento
from torch.utils.data import DataLoader
from transformers import AdamW
from tqdm import tqdm

model_path = "deepset/bert-base-cased-squad2"

model = AutoModelForQuestionAnswering.from_pretrained(model_path)
tokenizer = AutoTokenizer.from_pretrained(model_path)

gc.collect()
torch.cuda.empty_cache()
# configurar GPU
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
# se mueve el modelo al dispositivo detectado
model.to(device)
# se activa el modo de entrenamiento del modelo
model.train()
# se inicializa adam optimizer con caída de peso (reduce la posibilidad de sobreajuste)
optim = AdamW(model.parameters(), lr=5e-5) #tasa de aprendizaje
# Se inicializa el cargador de datos para los datos de entrenamiento
train_loader = DataLoader(train_dataset, batch_size=7, shuffle=True)

for epoch in range(7):
    # configurar el modelo en modo de entrenamiento
    model.train()
    # bucle de configuración (usamos tqdm para la barra de progreso)
    loop = tqdm(train_loader, leave=True)
    for batch in loop:
        # Se inicializa los gradientes calculados (del paso anterior)
        optim.zero_grad()
        # extraer todos los lotes de tensores necesarios para el entrenamiento
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        start_positions = batch['start_positions'].to(device)
        end_positions = batch['end_positions'].to(device)
        # Salida del modelo de entrenamiento por lotes y devoluciones (incluida la pérdida)

```

```

outputs = model(input_ids, attention_mask=attention_mask,
                start_positions=start_positions,
                end_positions=end_positions)

# extraer la pérdida
loss = outputs[0]

# calcula la pérdida para cada parámetro que necesite actualización gradual
loss.backward()

# actualiza los parámetros
optim.step()

# imprime información relevante en la barra de progreso
loop.set_description(f'Epoch {epoch}')
loop.set_postfix(loss=loss.item())

```

Para la evaluación del entrenamiento del modelo con 7 épocas (epoch) y de tamaño de lote 7 (batch size), se empleó el siguiente código que permite medir la precisión del modelo en base a las respuestas con las que se entrenó y las que genera el modelo.

```

dump_equals=False
model.eval()
acc = []
train_loader = DataLoader(train_dataset, batch_size=7, shuffle=True)
dump_count = 0

eval_data = []
for batch in train_loader:
    with torch.no_grad():
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        start_true = batch['start_positions'].to(device)
        end_true = batch['end_positions'].to(device)
        # realiza estimaciones
        outputs = model(input_ids, attention_mask=attention_mask)
        # obtiene la mejor predicción con argmax
        start_pred = torch.argmax(outputs['start_logits'], dim=1)
        end_pred = torch.argmax(outputs['end_logits'], dim=1)
        # calcula la precisión
        acc.append(((start_pred == start_true).sum()/len(start_pred)).item())
        acc.append(((end_pred == end_true).sum()/len(end_pred)).item())
# calcula de la precisión total, promediando las precisiones
acc = sum(acc)/len(acc)
print("Precisión del modelo ajustado:", acc)
print("eval_data", eval_data)
return acc, eval_data
acc, eval_data = eval_loader(1,dump_equals=False)

```

Precisión del modelo ajustado: 0.9451038575667656

Figura 39. Precisión del modelo ajustado

Como se observa en la Figura 39, se obtuvo una precisión general del modelo ajustado del

0.9451, la cual fue puesta a evaluación del entrenamiento del modelo con 7 épocas (epoch) y de tamaño de lote 7 (batch size) ya que fue el ajuste que menor pérdida ofreció (ver **TABLA VIII**).

Para comprobar que el ajuste se realizó correctamente se elaboraron dos gráficas donde se expone la diferencia entre el modelo base de Bert, para el primer caso se seleccionó el modelo preentrenado: deepset/bert-base-cased-squad2 proveniente de Huggingface, en el que se realizó el ajuste para que trabaje en campos relacionados del Covid-19, como se observa en proceso de aprendizaje de la Figura 400. Se seleccionaron solo las respuestas inferiores a 100 caracteres (índice de inicio e índice de fin de predicción), se evaluó el modelo Bert Base, con los hiperparámetros siguientes: batch 7 y 1 epoch, en el resultado final de la evaluación se puede observar tres gráficas en las que se dividen en Predicción, Real, y Combinado, se expone el estado actual del modelo Bert Base, en el gráfico Combinado se muestra la disparidad que existe entre los resultados de la Predicción y los valores Reales. El modelo base Bert estaba entrenado con SQuAD 2.0, por lo que no predijo correctamente. SQuAD es un conjunto de datos un poco sesgado, la mayoría de las respuestas y preguntas tienen el mismo vocabulario, como resultado el modelo base no se ajusta a un campo en concreto, así que puede dar respuestas, pero no serán precisas como en un modelo ajustado.

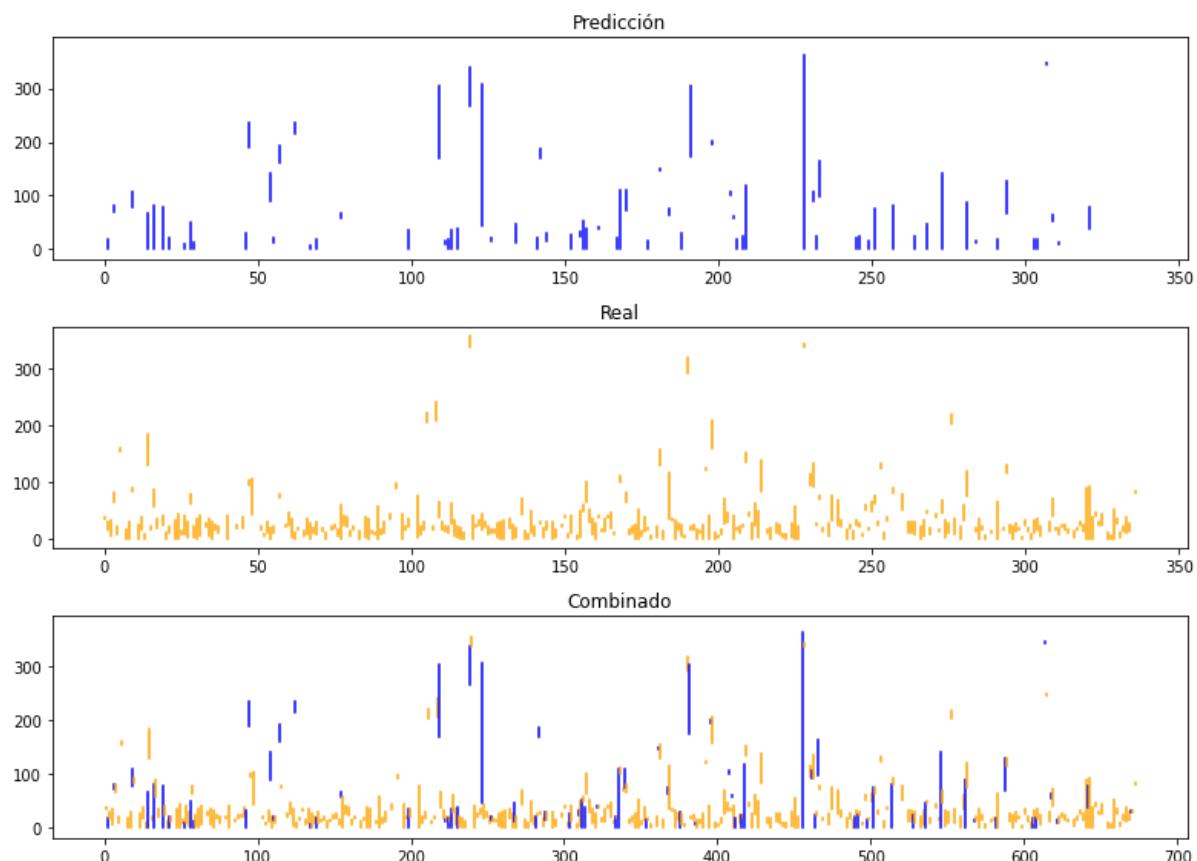


Figura 40. Proceso de aprendizaje

Para finalizar la comparación entre un modelo base sin ajuste frente a un modelo ajustado en un ámbito en concreto como lo es el Covid-19, se presenta en el Figura 411 el resultado del aprendizaje del modelo ajustado. Se muestra el resultado de la combinación de la predicción con los resultados reales datos por el conjunto de datos del entrenamiento, se puede apreciar que la paridad en este caso es mayor a la que se observaba con el modelo Bert Base, se aprecia una alineación notable en el gráfico Combinado, lo que significa que el modelo se pudo ajustar exitosamente gracias a los hiperparámetros definidos en la **Tabla VIII**. Se debe recalcar que el conjunto de datos influye en la predicción ya si homogeneidad suele facilitar el trabajo al modelo.

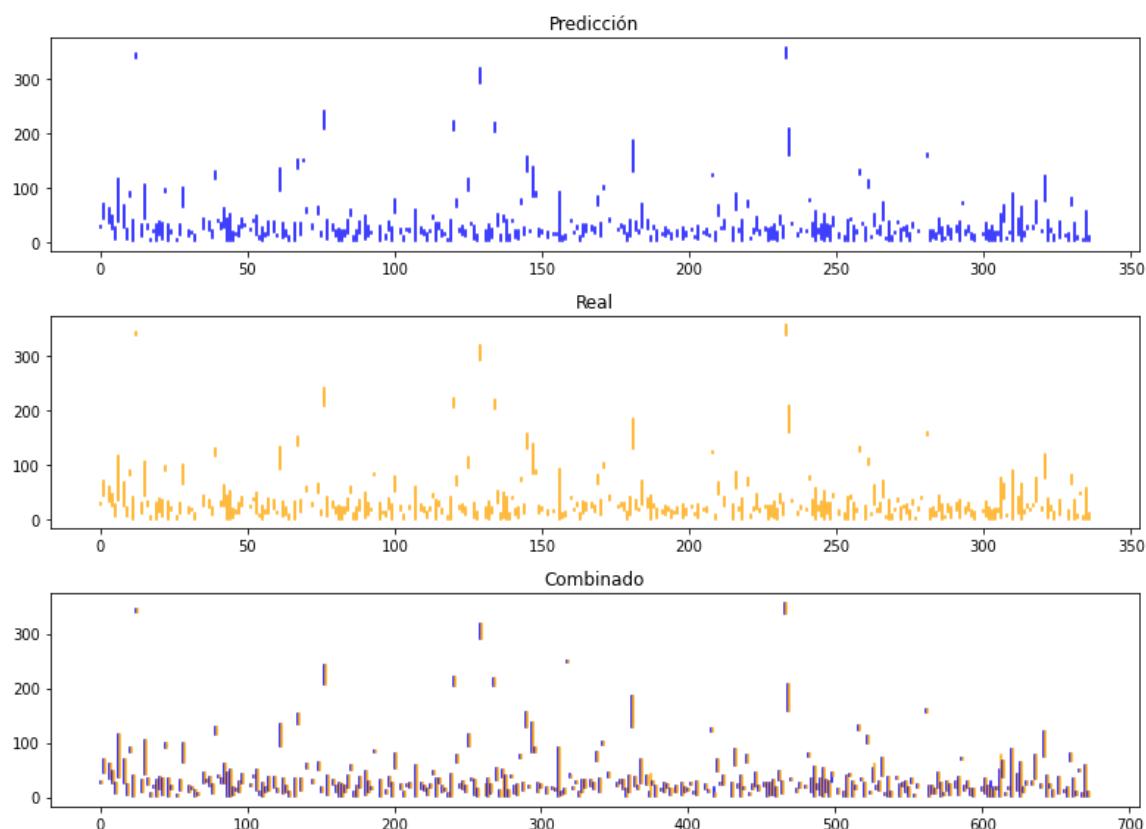


Figura 41. Proceso de aprendizaje ajustado

6.2.1.1. Preparar el conjunto de datos

Para la elaboración del conjunto de datos se realizó un preprocesamiento del conjunto de datos CORD-19, ya que cuenta con un total aproximado de un millón de artículos y trabajar con esta basta cantidad de datos generaría cuellos de botella en el procesamiento del sistema por la limitación de hardware. Por lo cual se prepararon aproximadamente 50.000 artículos procedentes de bases de datos científicas y medicas como lo son ScienceDirect, Elsevier, NCBI (National Center for Biotechnology Information) y PuMed; donde se fusionaron en un conjunto de datos en el cual se extrajeron diferentes campos como son: sha, title, abstract, publish_time, authors, url, body_text, con los que se creó un dataframe de todos los trabajos

de investigación necesarios para poder buscar en ellos, posteriormente se realizó un filtrado de las filas de los metadatos que tenían campos nulos o no poseían datos en sus campos, como el resumen, quedando sobre 35.000 documentos; el modelo BERT utiliza en este caso el campo “body text”, el cual es el documento completo para obtener las respuestas a las preguntas, esta es la principal razón por la que el modelo demora en obtener una respuesta, solventando este contrariedad se realizó otro filtrado mediante dos filtros, el primer filtro que limitó por año de publicación en los años 2020, 2021 y 2022. Seguidamente se aplicó el segundo filtro por palabras clave, redujo drásticamente la cantidad de documentos quedando un aproximado de 400 documentos, con los que trabajará el modelo, permitiéndole un desempeño adecuado.

```
# Filtro para seleccionar los artículos en fechas determinadas
df_finalfiltro=df_finalfiltro.loc[df_finalfiltro["publish_time"].between('2020-03-01', '2022-12-31')]
len(df_finalfiltro)

# Filtro para seleccionar los artículos por su título y cuerpo de texto, se filtran
# por palabras claves
import numpy as np
mask = np.column_stack([df_finalfiltro['title'].astype(str).str.contains("symptom" and "covid",
, na=False, case=False) for col in df_final])
filtro = df_finalfiltro.loc[mask.any(axis=1)]
```

Además, se debe tener en cuenta que al dataset se le puede añadir documentos más específicos dependiendo del desarrollo de las investigaciones y avances de la medicina en la lucha contra el coronavirus, permitiendo seguir actualizando y fortaleciendo la base de conocimientos a futuro y resolviendo dudas referentes al Covid-19.

6.2.1.2. Análisis Exploratorio de Datos - Data Analysis (EDA)

Concluida la elaboración del conjunto de datos elaborado y filtrado, se procedió al análisis de los 400 documentos, donde se obtuvo que los documentos corresponden al periodo desde que la pandemia dio inicio hasta la actualidad, en el año 2019 no se data ningún documento ya que la mayoría de investigaciones inició en el año 2020 y posteriormente fueron aumentando para el siguiente año 2021, como se observa en el Figura 42.

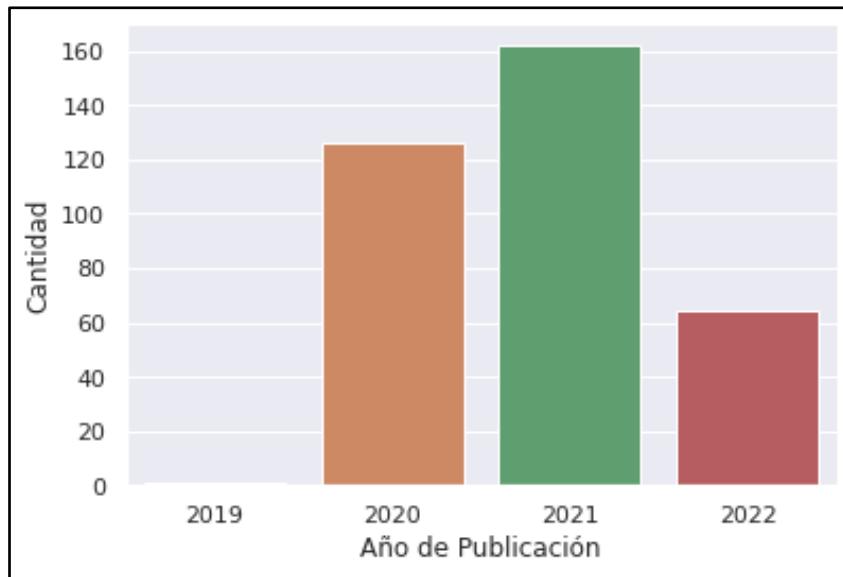


Figura 42. Artículos publicados por año y cantidad

Para indagar sobre las palabras más empleadas en los títulos de los artículos de investigación seleccionados, se elaboró una nube de palabras para poder distinguir cuales tienen un mayor peso significativo en cada investigación, entre las que más destacan son: análisis, vacuna, enfermedad, factor, mortalidad, tratamiento, entre otras palabras, las cuales se puede apreciar en la Figura 43.

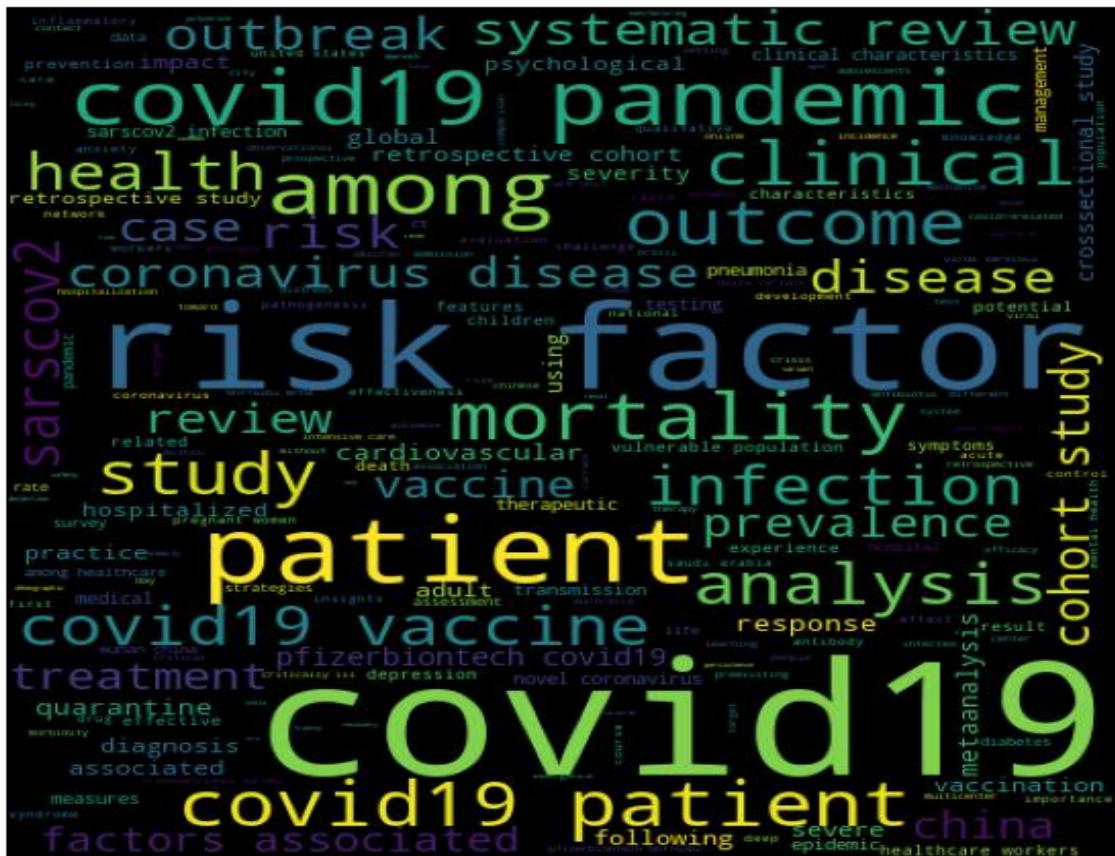


Figura 43. Nube de palabras de títulos

Para verificar y constatar la cantidad de veces que aparecen los terminos más usados en los artículos se creo la siguiente nube de palabras, donde se puede apreciar que los terminos como enfermedad (disease), covid, paciente, tratamiento y enfermedad, son las palabras que aparecen con mayor frecuencia en los documentos del dataset elaborado, como se aprecia en la Figura 44.



Figura 44. Términos de Frecuencia en los documentos

6.2.2. Analizar los resultados obtenidos del modelo entrenado.

Para analizar los resultados obtenidos por el modelo ajustado se codifico con ayuda de un contenedor de ElasticSerch la base del conocimiento y una API para obtener las respuestas con el framework FastAPI, este proceso se detalla más adelante en la subsección correspondiente a la codificación del agente virtual.

En base a la puntuación o score ofrecido por el modelo BERT ajustado se tomaron como resultados para las respuestas a preguntas. Se analizaron 50.000 artículos del conjunto de datos CORD-19, de los cuales se obtienen las respuestas, para el análisis realizado por el recuperador o retriever, se tomó la mejor puntuadas en base al score de los 5 mejores artículos científicos con los que coincide el contexto de la pregunta, se expone en la interfaz de usuario: la respuesta, título, autores, fuente de donde se obtuvo esa respuesta y el contexto del documento con la respuesta marcada en color verde, para aportar mayor información y confianza al usuario.

Los resultados obtenidos del modelo se enfocan en las respuestas válidas de las preguntas realizadas, siempre y cuando estén bien escritas y las respuestas tengan coherencia y den un aporte informativo en lo que respecta al Covid-19. Se elaboró un conjunto de preguntas mediante una revisión de literatura para obtener los temas, preguntas y palabras clave relacionadas con el Covid-19, se obtuvieron 20 preguntas de la mezcla de los temas, cuestiones y palabras clave, con el fin de realizar un formulario de preguntas y constatar el rendimiento del agente conversacional. Además, se incluyó una puntuación sobre el Acierto

en base al criterio propio, junto a la evaluación de buena y mala (1 - 0), que fue validada con los datos de entrenamiento, recalmando que la validación solo sirve para saber que el modelo está ajustado con los datos adecuados, los resultados obtenidos están plasmados en la **TABLA IX**, para los cuales se limitó la cantidad de respuestas a tres y se obtuvo las respuestas con mayor confianza creadas por el modelo para cada pregunta.

TABLA IX. Test de preguntas

Test de preguntas			
Nº	Pregunta	Resultado	Acierto (%)
1	¿Qué síntomas son menos comunes del covid-19?	• Respuesta: fiebre y tos	0
2	¿Cuáles son los factores de riesgo del covid-19?	• Respuesta: la salud del corazón y los pulmones, la edad y el sobrepeso o la insuficiencia ponderal	1
3	¿Qué efectos secundarios tienen las vacunas?	• Respuesta: dolor local, fatiga, dolor muscular, hinchazón local, dolor en las articulaciones, dolor de cabeza, fiebre, escalofríos, enrojecimiento local y náuseas	1
4	¿Qué tratamientos existen contra el covid-19?	• Respuesta: tratamiento antiviral, terapia con antibióticos, terapia con glucocorticoides, terapia con gammaglobulina, terapia con albúmina, inhalación de oxígeno, ventilación mecánica y TRRC	1
5	¿Qué es el covid-19?	• Respuesta: un virus respiratorio emergente	1
6	¿Cuál fue el lugar de origen del covid-19?	• Respuesta: Ciudad de Wuhan, provincia de Hubei, China	1
7	¿Qué tipos de pruebas para covid-19 están disponibles?	• Respuesta: Ensayos de reacción en cadena de la polimerasa con transcriptasa inversa en tiempo real (RT-PCR)	1
8	¿Cuáles son las recomendaciones para quienes tienen síntomas de covid-19?	• Respuesta: que las personas con EII continúen siguiendo los consejos de salud pública sobre el distanciamiento físico, el uso de mascarillas, que se les haga una prueba de inmediato para detectar síntomas que sugieran COVID-19 o si se exponen a un caso conocido de COVID-19 y se vacunen en la primera oportunidad disponible.	1
9	¿Cuál es la tasa de mortalidad del covid-19?	• Respuesta: 12.6%	1
10	¿Cuánto tiempo se tarda en generar anticuerpos para el covid-19?	• Respuesta: 7 días	1
11	¿Cómo se puede prevenir el covid-19?	• Respuesta: uso adecuado de la máscara.	1
12	¿Cuáles son los	• Respuesta: la mayoría de las personas con fiebre o	1

	síntomas a largo plazo del covid-19?	síntomas respiratorios	
13	¿Qué efectos secundarios producen las vacunas contra el covid-19?	<ul style="list-style-type: none"> • Respuesta: dolores de cabeza, dolor en el lugar de la inyección de la vacuna, reacciones anafilácticas de leves a moderadas y muerte (n=1000 muertes, 0,003 %) que no estaban directamente relacionados 	1
14	¿Qué vacunas contra el covid-19 existen?	<ul style="list-style-type: none"> • Respuesta: dos vacunas basadas en ARNm de Pfizer-BioNTech y Moderna 	1
15	¿Cuántos días de cuarentena debo realizar si tengo covid-19?	<ul style="list-style-type: none"> • Respuesta: 14 	1
16	¿Son eficaces los antibióticos para prevenir o tratar la COVID-19?	<ul style="list-style-type: none"> • Respuesta: Las vacunas fueron el medio de prevención más eficaz 	0
17	¿Cuáles son los síntomas comunes del covid-19?	<ul style="list-style-type: none"> • Respuesta: fiebre y tos 	1
18	¿A los cuantos días aparecen los síntomas del covid-19?	<ul style="list-style-type: none"> • Respuesta: 21 días 	0
19	¿Qué personas son más vulnerables al covid-19?	<ul style="list-style-type: none"> • Respuesta: ancianos 	1
20	¿Cuánto tiempo permanece el covid-19 en una superficie?	<ul style="list-style-type: none"> • Respuesta: hasta 72 horas 	1
TOTAL		Porcentaje de aciertos	85%

Como se observa en la **TABLA IX**, se obtiene un **85% de aciertos** respecto a las respuestas de las preguntas realizadas, se configuró para obtener una respuesta de los mejores 5 documentos analizados por el modelo.

6.2.3. Diseñar la arquitectura del agente virtual.

La arquitectura fue destinada para implementarla en una aplicación web donde se emplearon las plataformas de Microsoft Azure y Google Cloud junto herramienta Docker con los contenedores que almacenaron cada módulo del sistema para su desarrollo y posterior levantamiento web, gracias a que permite exponer hacia el exterior cualquier servicio web desarrollado, además permite especificar en qué puerto se realiza la conexión al servicio de Streamlit (puerto 9200), FastAPI (puerto 8080) y de la aplicación web (<http://34.171.25.128:8080/>), con el fin de facilitar el acceso al modelo ajustado desde cualquier navegador web. Para el uso de este prototipo, el usuario puede realizar diferentes

consultas mediante la interfaz diseñada en Streamlit, en el cuadro de búsquedas y tras un breve procesamiento del modelo se presentará la respuesta gracias al servidor cargado con el servicio de FastAPI y ElasticSearch que se comunica con el modelo BERT ajustado con Pytorch el cual funciona con Haystack y permite ofrecer un tiempo de respuesta equivalente a la cantidad de documentos y respuestas seleccionadas en el panel lateral de opciones de búsqueda.

La arquitectura se compone de los siguientes elementos:

- El usuario puede utilizar el agente conversacional para resolver sus dudas desde cualquier dispositivo a través de internet.
- Módulo UI: Compuesto por el framework Streamlit que permite crear aplicaciones web de manera sencilla, la cual nos proporciona una interfaz intuitiva para que los usuarios realicen sus preguntas.
- Módulo Q&A: Compuesto por Haystack que desempeña la tarea de indexador de los diferentes artículos para agilizar búsquedas masivas de documentos. Seguido del modelo BERT el cual trabaja con transformadores o transformers gracias a estos el modelo establece una relación semántica entre los elementos de una oración y devuelve una respuesta inferida en base a los documentos otorgados por Haystack. Finalmente, el módulo está compuesto por FastAPI permite intercambiar datos entre los dos anteriores módulos del sistema mediante archivos Json.
- Base de Conocimientos: Compuesta por ElasticSearch, la cual realiza la función de base de datos, que contiene información estructurada y no estructurada de múltiples fuentes, almacena de forma sofisticada y eficiente para optimizar búsquedas basadas en textos.

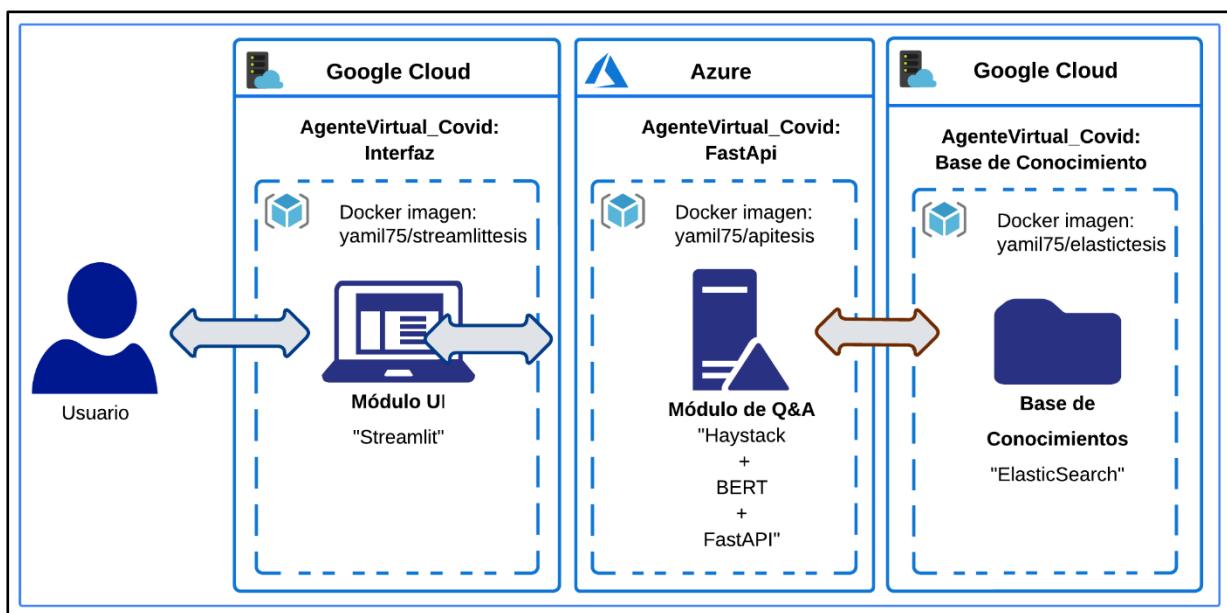


Figura 45. Arquitectura del Agente Conversacional

Arquitectura del modelo Bert:

La arquitectura como se manifiesta en la Figura 45, permite observar la estructura, funcionamiento e interacciones entre los componentes empleados en el desarrollo del agente conversacional. Los elementos principales, están distinguidos entre Usuario y el conjunto de todo el sistema que está almacenado en contenedores y estos son administrados desde la nube por los servicios de Microsoft Azure y Google Cloud que hacen de hospedadores para los tres contenedores en los que se divide el sistema. Como elementos secundarios tenemos los distintos subcomponentes que están divididos en los contenedores, donde: Streamlit, FastAPI y ElasticSearch, se comunican a través del protocolo Http, con el lenguaje de programación Python y las respuestas se realizan en formato Json.

El usuario, realiza preguntas dirigidas hacia el agente conversacional mediante la interfaz y puede seleccionar la cantidad de respuestas junto a la cantidad de artículos de los que desea información. La interfaz, realizada con la herramienta Streamlit, se compone principalmente por la presentación de la página, un menú desplegable de opciones, un cuadro de búsquedas y un botón para realizar la búsqueda. El resultado de la búsqueda presenta el título, autores, URL con el doi del documento, la respuesta que se colorea de un fondo color verde para identificar de que parte del documento científico se extrajo la respuesta y la confianza con la que el modelo ajustado devuelve una respuesta. Recibirá los datos provenientes desde FastAPI el cual está integrado con Haystack y el modelo ajustado, los cuales reciben los documentos desde ElasticSearch para ser procesados. Al tener la base de conocimientos creada con la herramienta ElasticSearch y almacenar documentos en idioma inglés el agente brinda la opción de traducir las preguntas que ingresa el usuario del español al inglés, ya que el modelo está entrenado en corpus en ese idioma y las respuestas que proporciona el modelo es en el mismo idioma por tal motivo se realiza una traducción al idioma local para presentarlas.

El servicio de FastAPI, trabaja con la herramienta Haystack la cual hace las funciones de recuperador y lector de documentos provenientes de la base de conocimientos de ElasticSearch, la cual realiza la tarea de buscar y administrar los documentos solicitados en base al contexto de la pregunta traducida.

El proceso general como se lo describe en la Figura 4646, se trata de que el usuario realice una consulta mediante la interfaz, la consulta se traducirá y consultará a través de FastAPI que solicitará los documentos a ElasticSearch y según el contexto de la pregunta ingresada, el retriever o recuperador seleccionará en su búsqueda dentro de la base de conocimiento cinco documentos que tengan la mayor cantidad de coincidencias en base a los contextos hallados y serán enviados para ser procesados por el modelo Bert ajustado, posteriormente,

devolverá una salida que será traducida y enviada a la interfaz.

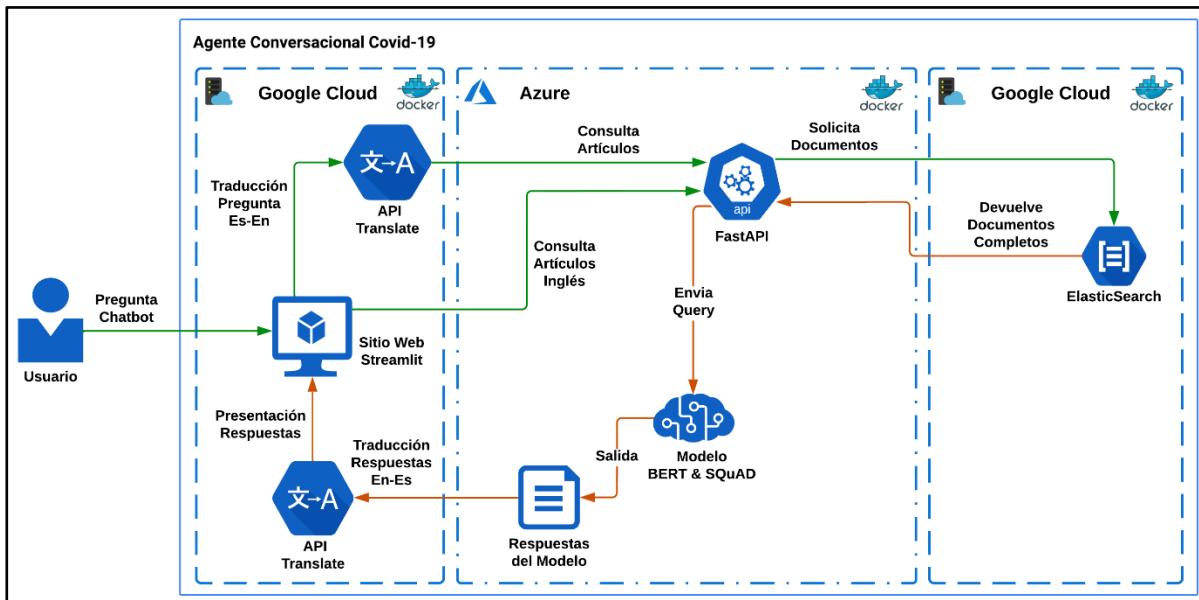


Figura 46. Arquitectura General Modelo BERT

6.2.4. Codificar en base al diseño.

El diseño visual o frontend se implementó con Streamlit un framework basado en el lenguaje Python destinado al aprendizaje automático, que aporta las herramientas necesarias para poder diseñar aplicaciones web con interfaces interactivas, ya que posee HTML y CSS. Además, acepta en el mismo diseño todas las librerías que estén vinculadas con Python. Para la base de conocimiento se utilizó los artículos del conjunto de datos CORD19 anteriormente preparados, los cuales se agregaron a un contenedor de ElasticSearch para luego obtenerlos cuando sea necesario.

```
from haystack.document_store.elasticsearch import ElasticsearchDocumentStore
document_store = ElasticsearchDocumentStore(host="localhost", username="", password="", index="document")

import pandas as pd
df = pd.read_csv("Dataset_DatosFiltro.csv")
print(df.head())

dicts = df.to_dict('records')
final_dicts = []
for each in dicts:
    tmp = {}
    tmp['text'] = each.pop('body_text')
    tmp['meta'] = each
    final_dicts.append(tmp)
```

```
document_store.write_documents(final_dicts)
```

El siguiente paso fue la creación de la API, mediante FastAPI, aquí se realizó la conexión con el contenedor de ElasticSearch y luego mediante el framework Haystack se logró obtener la respuesta a la pregunta. Posteriormente el api fue agregado a un contenedor de Docker para facilitar su uso.

```
from fastapi import FastAPI
from pydantic import BaseModel
from haystack.document_store.elasticsearch import ElasticsearchDocumentStore
import os
from haystack.pipeline import ExtractiveQAPipeline
from haystack.reader.farm import FARMReader
from haystack.retriever.sparse import ElasticsearchRetriever

ELASTIC_SEARCH_HOST = os.environ.get('es_ip', 'host.docker.internal')
ELASTIC_SEARCH_PORT = os.environ.get('es_port', 9200)

document_store = ElasticsearchDocumentStore(host=ELASTIC_SEARCH_HOST,
                                             port=ELASTIC_SEARCH_PORT,
                                             username="", password="",
                                             index="document")
retriever = ElasticsearchRetriever(document_store=document_store)
reader = FARMReader(model_name_or_path="JAlexis/ModelTT_Covid19", use_gpu=True)
pipe = ExtractiveQAPipeline(reader, retriever)
app = FastAPI()

class Queobj(BaseModel):
    question: str
    num_answers: int
    num_docs: int

@app.post('/query')
async def query(que_obj: Queobj):
    question = que_obj.question
    k_retriver = que_obj.num_docs
    k_reader = que_obj.num_answers
    prediction = pipe.run(query=question, top_k_retriever=k_retriver,
    top_k_reader=k_reader)
    return {'answer': prediction}
```

Finalmente se codifico la parte de la interfaz, la cual como previamente se menciona se la realizo con Streamlit, se realizó la conexión con el contenedor del api y se agregaron los campos necesarios para su funcionamiento.

Para facilitar el uso de la aplicación se dividieron e implementaron en contenedores de Docker cada parte de la aplicación, en la Figura 47 se muestran las imágenes resultantes, las mismas que luego fueron cargadas al repositorio de Docker Hub, para luego utilizarlas para el

despliegue de la aplicación (<https://hub.docker.com/u/yamil75>).

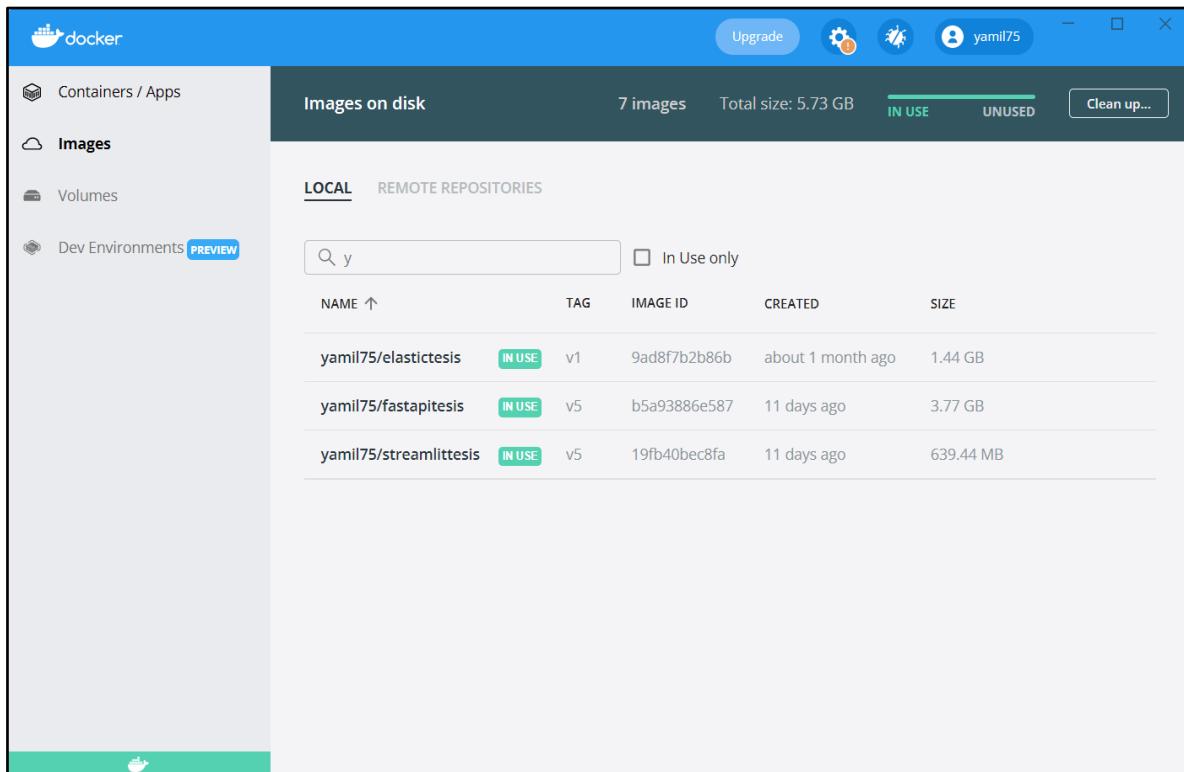


Figura 47. Contenedores de Docker resultantes

Para el diseño, al ser una agente virtual, debe aceptar preguntas relacionadas sobre el Covid-19, se estableció que debería tener una caja de búsqueda, otra de presentación de la respuesta junto a la fuente de donde se extrajo la respuesta, la confianza con la que el modelo produjo esa respuesta, un enlace directo al documento de donde se obtuvo la respuesta y avisos para cuando está realizando la búsqueda de las diferentes preguntas, así como cuando falla la búsqueda informar que se debe realizar nuevamente otra pregunta. Además, se agregó un panel lateral para que el usuario elija la cantidad de respuestas y de artículos que desee obtener. En la Figura 48 se presenta el funcionamiento del agente al momento de realizar una consulta, en la Figura 49 se observa el aviso que se devuelve si no se ingresa una pregunta en el cuadro de texto o si existe algún error en la consulta, y en la Figura 50 se muestra como el agente presenta la respuesta a una consulta.

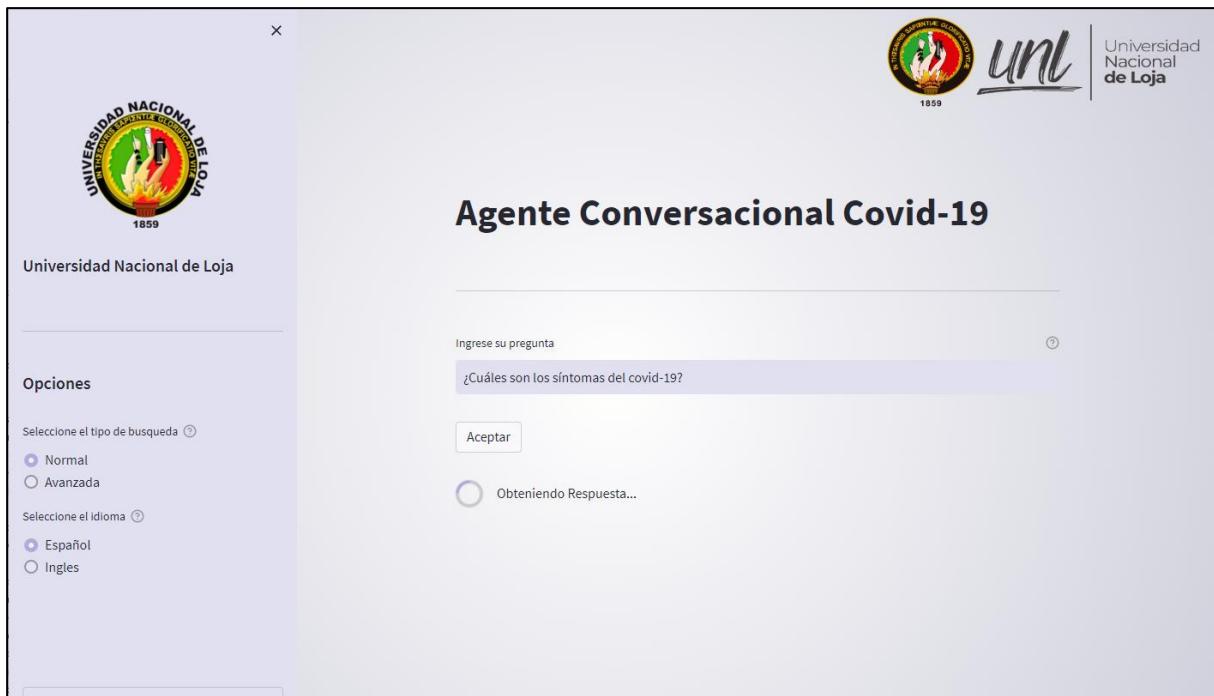


Figura 48. Aviso sobre la realización de la búsqueda

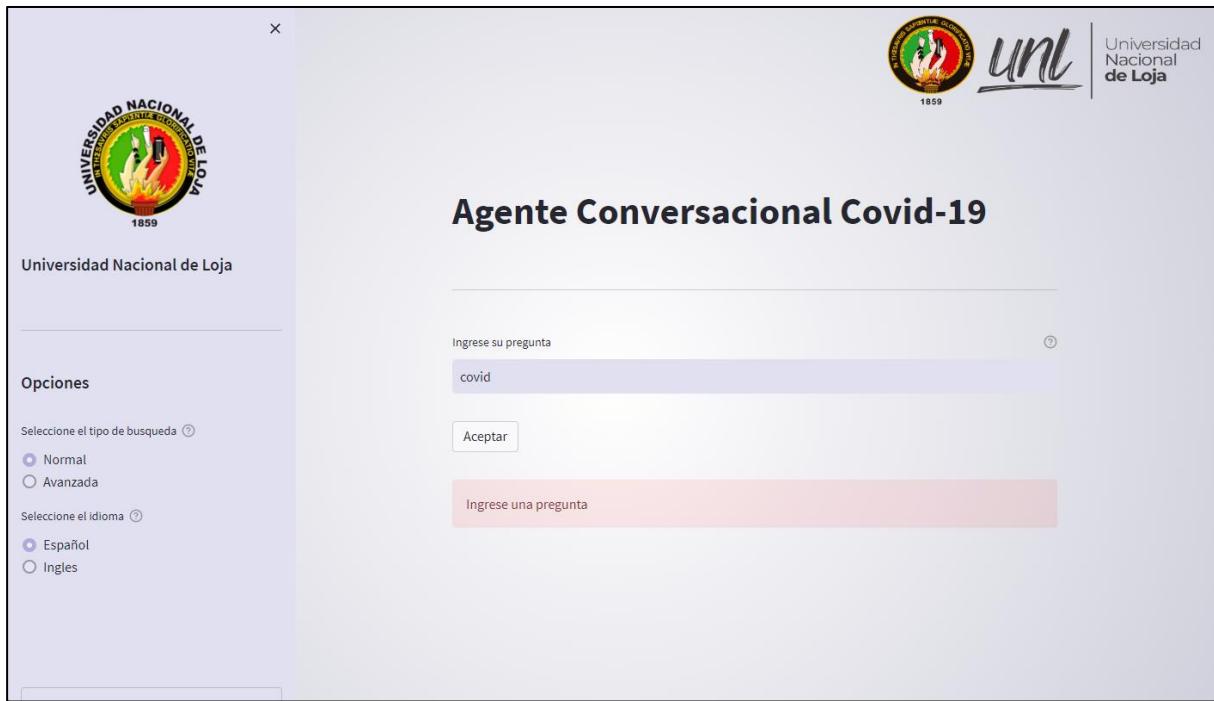


Figura 49. Aviso sobre la no contestación a la pregunta

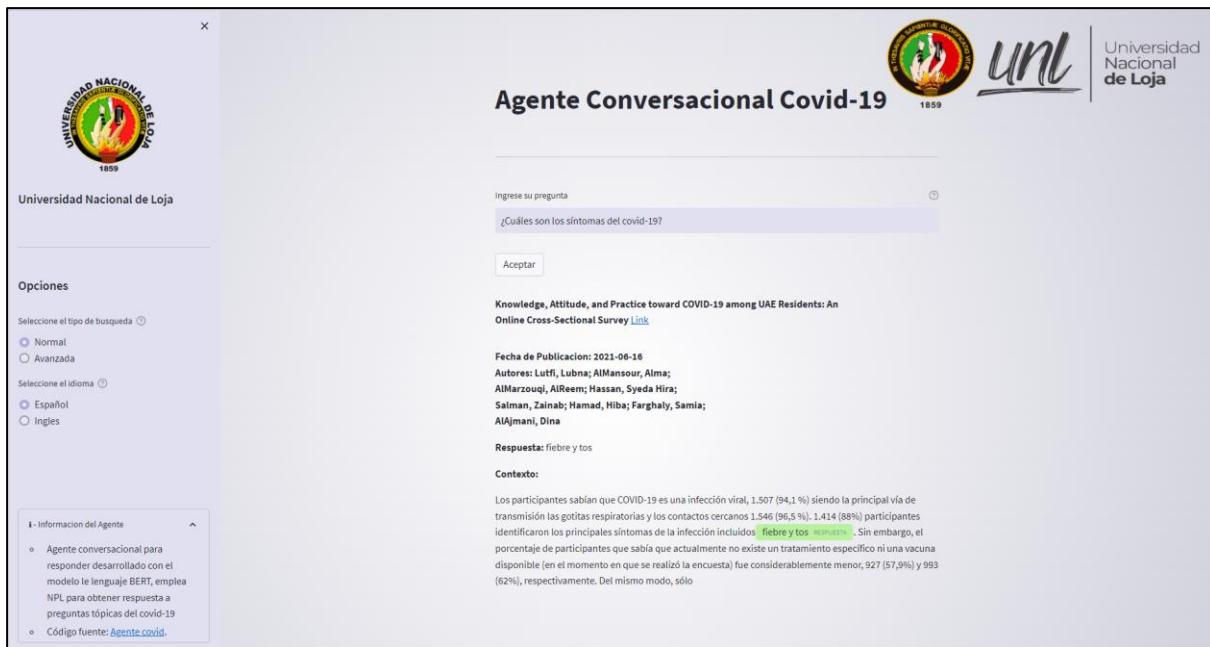


Figura 50. Respuesta del agente conversacional

6.2.5. Desarrollo del prototipo de una aplicación web para la interacción de los usuarios con el agente virtual.

Para el desarrollo eficiente del prototipo Agente Virtual Covid se implementó la metodología Desing Thinking, la cual se compone de cinco etapas con el fin de presentar un prototipo del agente virtual amigable hacia los usuarios y que les pueda servir para atenuar la desinformación que existe sobre el coronavirus, para la interfaz se empleó el framework Streamlit por su rápido desarrollo y adaptabilidad en proyectos de PLN tal y como se refleja en la fase de creación de prototipos donde se empleó una interfaz siguiendo los modelos ya realizados en otros proyectos de chatbots. Por último, para levantar el servicio web se emplearon los servicios de contenedores de Azure, en el cual se creó un contenedor que posee la API junto al modelo ajustado, como se muestra en la Figura 51. Figura 46 El código fuente se encuentra alojado y disponible en el repositorio github Tesis Agente Covid-19 (<https://github.com/victorSerrano98/Tesis-Agente-Covid-19>).

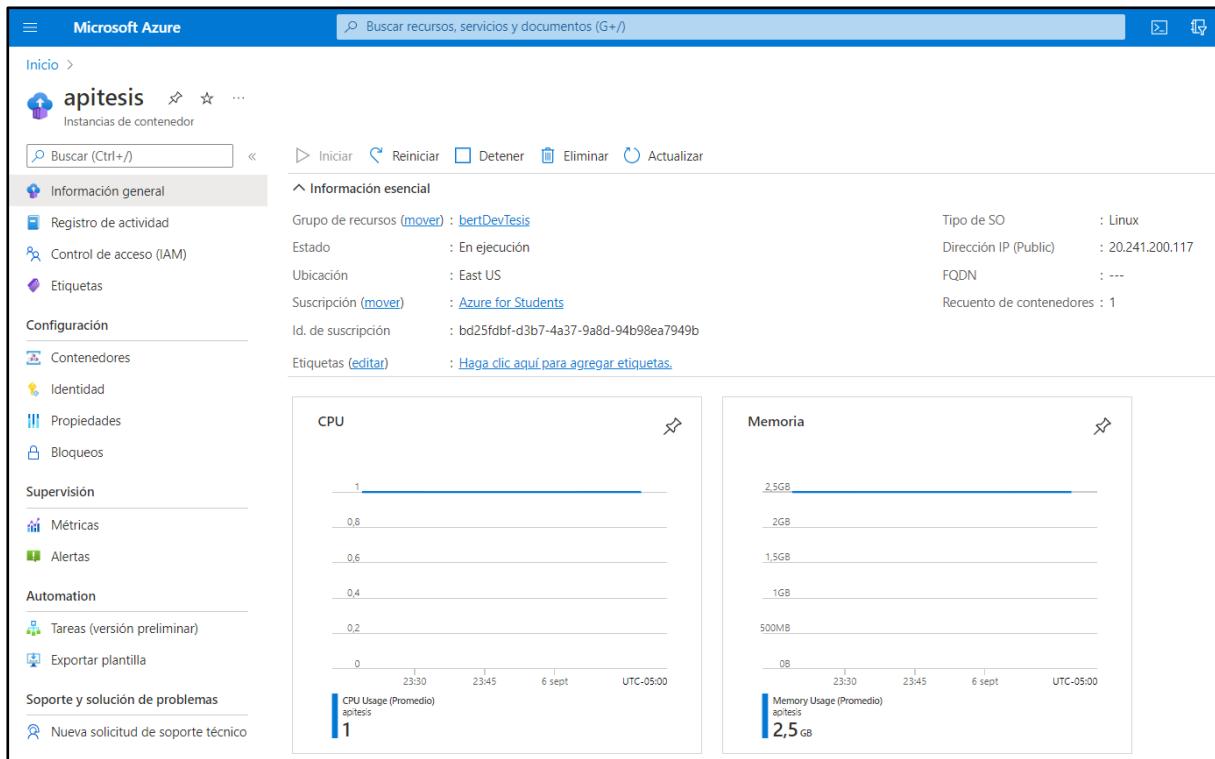


Figura 51. Levantamiento del API en Azure

La plataforma de Azure y su suscripción para estudiantes ofrece el despliegue de contenedores, en el que se ubicó la API y para la interfaz, así como la base de conocimientos, se empleó la plataforma de Google Cloud, la cual presenta mejores prestaciones en su suscripción, permitiendo la creación de varios con, como se muestra en la Figura 512. El resultado final del levantamiento de la aplicación se muestra en la Figura 522 y Figura 533.

El Agente Conversacional se implementó en la siguiente dirección: <http://34.171.25.128:8080/>, siguiendo el diseño establecido junto a su arquitectura y codificado en base a los parámetros, como se puede observar en la Figura 53.

Figura 52. Levantamiento de la Base de Conocimiento y la interfaz en Google Cloud

Figura 53. Prototipo levantado en la web

6.3. Objetivo 3: Evaluación de la herramienta.

Para la elaboración del último objetivo, utilizando la metodología Desing Thinking y siguiendo sus fases, se aplicó el testeo del prototipo diseñado en el segundo objetivo, en donde se comprobó que funcione correctamente proporcionando respuestas a las diferentes preguntas que se le ingresa, para constatar que el modelo devuelva información viable y constatada se

recurrió a profesionales de la salud como: doctores, enfermeras, laboratoristas y por último a una epidemióloga, los cuales aportaron con una retroalimentación en base a su percepción y experiencia en lo que se refiere al contexto del Covid-19.

6.3.1. Evaluar el agente virtual

a) Realizar las pruebas del software al agente virtual.

Para la realización de pruebas conforme la metodología Desing Thinking se empleó la evaluación humana (human evaluation), se expusieron las respuestas generadas por el modelo a profesionales de la salud para que valoren dichas respuestas respecto a la concordancia y aceptabilidad de sus conocimientos en el campo y verificar el porcentaje de respuestas válidas que ofrece el modelo, ya que los profesionales de la salud son los que tienen los conocimientos y pueden considerar si las respuestas son coherentes, gracias a una evaluación que tiene como parámetros: errónea, incompleta y correcta, según la apreciación que le den las respuestas.

Se emplearon algunas preguntas anteriormente usadas para obtener todas las posibles respuestas del modelo, en este caso se obtuvieron las respuestas por parte del modelo con una configuración que implica una búsqueda en cinco documentos, los cuales coinciden mayormente con el contexto de la pregunta y se busca dentro de estos una respuesta acorde la cual es devuelta por el modelo como la mejor puntuada, este proceso se realizó con todas las preguntas y las respuestas se enviaron a evaluación como se aprecia en la Tabla X.

TABLA X. Test de Preguntas para Evaluación

Test de preguntas de Evaluación			
Nº	Pregunta	Resultado	Evaluación
1	¿Qué síntomas son menos comunes del covid-19?	• Respuesta: fiebre y tos	40% Correcta 50% Incompleta 10% Errónea
2	¿Cuáles son los factores de riesgo del covid-19?	• Respuesta: la salud del corazón y los pulmones, la edad y el sobrepeso o la insuficiencia ponderal	80% Correcta 20% Incompleta 0% Errónea
3	¿Qué efectos secundarios tienen las vacunas?	• Respuesta: dolor local, fatiga, dolor muscular, hinchazón local, dolor en las articulaciones, dolor de cabeza, fiebre, escalofríos, enrojecimiento local y náuseas	80% Correcta 20% Incompleta 0% Errónea
4	¿Qué tratamientos existen contra el covid-19?	• Respuesta: tratamiento antiviral, terapia con antibióticos, terapia con glucocorticoides, terapia con gammaglobulina, terapia con albúmina,	40% Correcta 40% Incompleta 20% Errónea

		inhalación de oxígeno, ventilación mecánica y TRRC	
5	¿Qué es el covid-19?	<ul style="list-style-type: none"> • Respuesta: un virus respiratorio emergente 	60% Correcta 30% Incompleta 10% Errónea
6	¿Cuál fue el lugar de origen del covid-19?	<ul style="list-style-type: none"> • Respuesta: Ciudad de Wuhan, provincia de Hubei, China 	90% Correcta 10% Incompleta 0% Errónea
7	¿Qué tipos de pruebas para covid-19 están disponibles?	<ul style="list-style-type: none"> • Respuesta: Ensayos de reacción en cadena de la polimerasa con transcriptasa inversa en tiempo real (RT-PCR) 	50% Correcta 50% Incompleta 0% Errónea
8	¿Cuáles son las recomendaciones para quienes tienen síntomas de covid-19?	<ul style="list-style-type: none"> • Respuesta: que las personas con EI continúen siguiendo los consejos de salud pública sobre el distanciamiento físico, el uso de mascarillas, que se les haga una prueba de inmediato para detectar síntomas que sugieran COVID-19 o si se exponen a un caso conocido de COVID-19 y se vacunen en la primera oportunidad disponible. 	70% Correcta 30% Incompleta 0% Errónea
9	¿Cuál es la tasa de mortalidad del covid-19?	<ul style="list-style-type: none"> • Respuesta: 12.6% 	0% Correcta 30% Incompleta 70% Errónea
10	¿Cuánto tiempo se tarda en generar anticuerpos para el covid-19?	<ul style="list-style-type: none"> • Respuesta: 7 días 	70% Correcta 30% Incompleta 0% Errónea
11	¿Cómo se puede prevenir el covid-19?	<ul style="list-style-type: none"> • Respuesta: uso adecuado de la máscara. 	60% Correcta 40% Incompleta 0% Errónea
12	¿Cuáles son los síntomas a largo plazo del covid-19?	<ul style="list-style-type: none"> • Respuesta: la mayoría de las personas con fiebre o síntomas respiratorios 	20% Correcta 30% Incompleta 70% Errónea
13	¿Qué efectos secundarios producen las vacunas contra el covid-19?	<ul style="list-style-type: none"> • Respuesta: dolores de cabeza, dolor en el lugar de la inyección de la vacuna, reacciones anafilácticas de leves a moderadas y muerte (n=1000 muertes, 0,003 %) que no estaban directamente relacionados 	40% Correcta 60% Incompleta 0% Errónea

14	¿Qué vacunas contra el covid-19 existen?	<ul style="list-style-type: none"> Respuesta: dos vacunas basadas en ARNm de Pfizer-BioNTech y Moderna 	60% Correcta 40% Incompleta 0% Errónea
15	¿Cuántos días de cuarentena debo realizar si tengo covid-19?	<ul style="list-style-type: none"> Respuesta: 14 	80% Correcta 20% Incompleta 0% Errónea
16	¿Son eficaces los antibióticos para prevenir o tratar la COVID-19?	<ul style="list-style-type: none"> Respuesta: Las vacunas fueron el medio de prevención más eficaz 	20% Correcta 70% Incompleta 10% Errónea
17	¿Cuáles son los síntomas comunes del covid-19?	<ul style="list-style-type: none"> Respuesta: fiebre y tos 	70% Correcta 30% Incompleta 0% Errónea
18	¿A los cuantos días aparecen los síntomas del covid-19?	<ul style="list-style-type: none"> Respuesta: 21 días 	0% Correcta 20% Incompleta 80% Errónea
19	¿Qué personas son más vulnerables al covid-19?	<ul style="list-style-type: none"> Respuesta: ancianos 	60% Correcta 40% Incompleta 0% Errónea
20	¿Cuánto tiempo permanece el covid-19 en una superficie?	<ul style="list-style-type: none"> Respuesta: hasta 72 horas 	70% Correcta 30% Incompleta 0% Errónea
TOTAL		Porcentaje De Aciertos Totales	60% Correcta 25% Incompleta 15% Errónea

Los resultados de la evaluación humana sometida a veinte preguntas más comunes sobre el Covid-19, se expuso a los profesionales de la salud que evaluaron las respuestas con mediante tres distintas opciones:

- Correcta: para las respuestas generadas por el modelo en las que tengan concordancia y sea información adecuada para solventar la pregunta.
- Incompleta: para las respuestas que aportan información, pero de forma parcial desde el punto de vista de los profesionales de la salud. No están mal, pero podría ser más exacta la información.
- Errónea: Ninguna de las respuestas proporciona información relevante que sirva como contestación.

Los resultados de la evaluación realizada por profesionales de la salud: un 60% de las respuestas es plenamente correcta, es decir solventa las dudas realizadas al agente. Un 25% de las respuestas evaluadas tiene concordancia en sus respuestas, no necesariamente está mal la respuesta ya que no desinforma, pero le faltan datos para poder considerarla como una respuesta correcta. El porcentaje restante que es el 15%, son respuestas que se las consideraron erróneas, debido a que no tiene alguna relación con la pregunta. Toda esta información con sus respectivas gráficas se encuentra en el Anexo 4.

Además, para tener una mejor evaluación en cuanto a la veracidad de las respuestas se optó por evaluar el agente con ayuda de una especialista en epidemiología, a la cual se le envió el mismo formulario anteriormente mencionado, el resultado de esta evaluación se la puede observar en la **TABLA XI.** Evaluación del agente por un especialista en epidemióloga

TABLA XI. Evaluación del agente por un especialista en epidemióloga

Test de preguntas de Evaluación			
N.º	Pregunta	Resultado	Evaluación
1	¿Qué síntomas son menos comunes del covid-19?	<ul style="list-style-type: none"> • Respuesta: fiebre y tos 	Incompleta
2	¿Cuáles son los factores de riesgo del covid-19?	<ul style="list-style-type: none"> • Respuesta: la salud del corazón y los pulmones, la edad y el sobrepeso o la insuficiencia ponderal 	Correcta
3	¿Qué efectos secundarios tienen las vacunas?	<ul style="list-style-type: none"> • Respuesta: dolor local, fatiga, dolor muscular, hinchazón local, dolor en las articulaciones, dolor de cabeza, fiebre, escalofríos, enrojecimiento local y náuseas 	Correcta
4	¿Qué tratamientos existen contra el covid-19?	<ul style="list-style-type: none"> • Respuesta: tratamiento antiviral, terapia con antibióticos, terapia con glucocorticoides, terapia con gammaglobulina, terapia con albúmina, inhalación de oxígeno, ventilación mecánica y TRRC 	Incompleta
5	¿Qué es el covid-19?	<ul style="list-style-type: none"> • Respuesta: un virus respiratorio emergente 	Correcta
6	¿Cuál fue el lugar de origen del covid-19?	<ul style="list-style-type: none"> • Respuesta: Ciudad de Wuhan, provincia de Hubei, China 	Correcta
7	¿Qué tipos de pruebas para covid-19 están disponibles?	<ul style="list-style-type: none"> • Respuesta: Ensayos de reacción en cadena de la polimerasa con transcriptasa inversa en tiempo real (RT-PCR) 	Incompleta
8	¿Cuáles son las recomendaciones para quienes tienen síntomas de covid-19?	<ul style="list-style-type: none"> • Respuesta: que las personas con EII continúen siguiendo los consejos de salud pública sobre el distanciamiento físico, el uso de mascarillas, que se les haga una prueba de inmediato para detectar síntomas que sugieran COVID-19 o si se exponen a un caso conocido de COVID-19 y se vacunen en la primera oportunidad 	Correcta

		disponible.	
9	¿Cuál es la tasa de mortalidad del covid-19?	• Respuesta: 12.6%	Errónea
10	¿Cuánto tiempo se tarda en generar anticuerpos para el covid-19?	• Respuesta: 7 días	Incompleta
11	¿Cómo se puede prevenir el covid-19?	• Respuesta: uso adecuado de la máscara.	Incompleta
12	¿Cuáles son los síntomas a largo plazo del covid-19?	• Respuesta: la mayoría de las personas con fiebre o síntomas respiratorios	Errónea
13	¿Qué efectos secundarios producen las vacunas contra el covid-19?	• Respuesta: dolores de cabeza, dolor en el lugar de la inyección de la vacuna, reacciones anafilácticas de leves a moderadas y muerte (n=1000 muertes, 0,003 %) que no estaban directamente relacionados	Incompleta
14	¿Qué vacunas contra el covid-19 existen?	• Respuesta: dos vacunas basadas en ARNm de Pfizer-BioNTech y Moderna	Incompleta
15	¿Cuántos días de cuarentena debo realizar si tengo covid-19?	• Respuesta: 14	Correcta
16	¿Son eficaces los antibióticos para prevenir o tratar la COVID-19?	• Respuesta: Las vacunas fueron el medio de prevención más eficaz	Incompleta
17	¿Cuáles son los síntomas comunes del covid-19?	• Respuesta: fiebre y tos	Incompleta
18	¿A los cuantos días aparecen los síntomas del covid-19?	• Respuesta: 21 días	Errónea
19	¿Qué personas son más vulnerables al covid-19?	• Respuesta: ancianos	Correcta
20	¿Cuánto tiempo permanece el covid-19 en una superficie?	• Respuesta: hasta 72 horas	Correcta
TOTAL		Porcentaje de aciertos totales	40% Correcta 45% Incompleta 15% Errónea

Los resultados de la evaluación realizada por la epidemióloga son: un 40% de las respuestas correctas, un 45% tiene concordancia en sus respuestas, pero se encuentran incompletas y un 15%, son respuestas erróneas.

Obtenidos los resultados finales de la evaluación, se considera que el Agente Conversacional sobre el Covid-19, alcanza un 85% de efectividad en su función de dar respuestas que den información respecto a preguntas tópicas como se indica en la TABLA XI.

b) Evidenciar el funcionamiento de la aplicación integrada con el agente virtual.

El prototipo del agente conversacional que se ha desarrollado, responde a las preguntas

de los usuarios mediante su interfaz gráfica, cómo se observa en la Figura 544, sigue el proceso de recuperar los documentos desde el contenedor de ElasticSearch, la cual devuelve un json con todos los artículos científicos previamente agregados, para que posteriormente mediante nuestro modelo afinado se obtengan las respuestas a la pregunta. Se presenta la respuesta junto con el título y autores del artículo, y la confiabilidad de la respuesta.

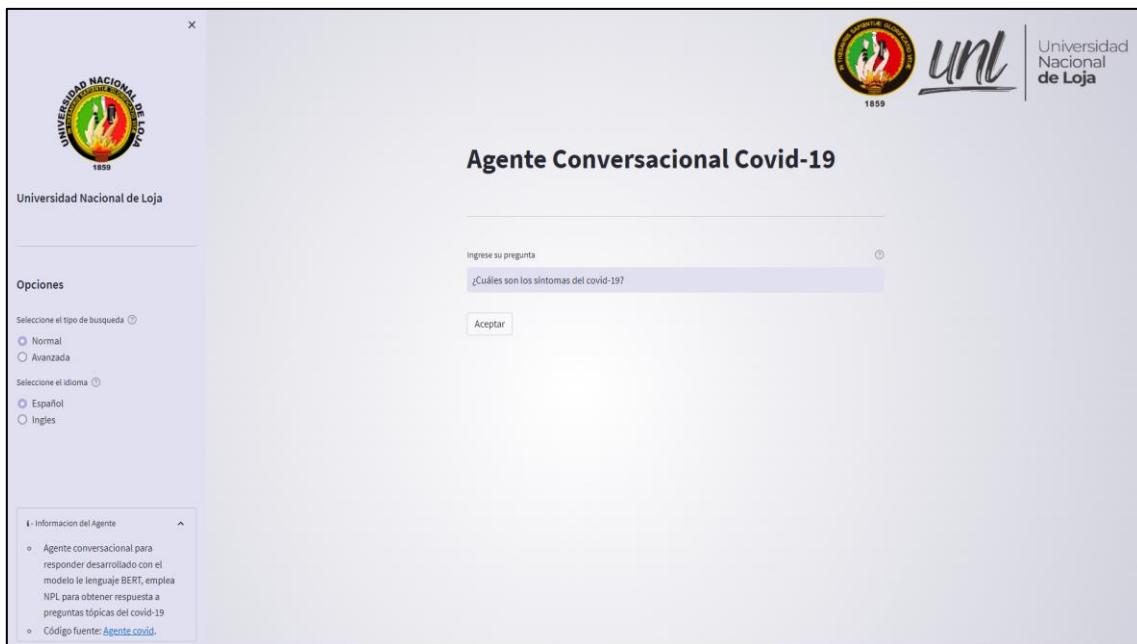


Figura 54 Interfaz del agente conversacional

Si la respuesta del agente no es satisfactoria, el usuario tiene la opción de elegir en la barra lateral una consulta avanzada, en la cual el agente realizará la búsqueda en un número mayor de documentos y por consiguiente el tiempo de espera será mayor.

7. Discusión

El propósito de la presente sección es ratificar los objetivos que conforman el Trabajo de Titulación (TT), cuyo fin general fue la creación de un agente conversacional para cuestiones relacionadas con el coronavirus para brindar apoyo a la sociedad y mitigar la carga a los profesionales de la salud, además de solventar el problema de investigación el cual trata de: ¿La creación de un agente virtual utilizando el modelo de lenguaje BERT (Bidirectional Encoder Representations from Transformers) junto a SQuAD (Stanford Question Answering Dataset) permitirá responder correctamente a preguntas tópicas acerca del Covid-19?. De acuerdo con los resultados obtenidos en los objetivos planteados, se ratifica la creación de un agente virtual, donde según las evaluaciones realizadas se justifica que el modelo ajustado en base a BERT y SQuAD, permite responder apropiadamente a preguntas sobre el Covid-19.

Las siguientes discusiones se desglosan por cada uno de los objetivos propuesto respecto a otros trabajos, donde se relacionan los puntos más importantes de cada proceso, sus limitaciones, mejoras y los aportes más importantes del desarrollo de este TT.

7.1. Desarrollo de la discusión de los objetivos específicos

Objetivo 1: Definir el modelo de lenguaje BERT junto a SQuAD para generar un modelo que dé respuesta a las preguntas realizadas al agente virtual.

Para la culminación del primer objetivo, se realizó mediante la elaboración de una Revisión Sistemática de Literatura (Anexo 1), la RSL nos aportó estudios relacionados con los modelos BERT y SQuAD que permitió arrojar conocimientos para poder seleccionar un modelo y mediante el cual se sustente el agente conversacional. Se adquirió información respecto a las características y funcionamiento, versiones y parámetros, los cuales fueron publicados en el volumen 11 de la revista CEDAMAZ el 15 de septiembre del 2021 [79], consecutivamente se obtuvo información relacionada a las herramientas que se utilizaron para desarrollar el agente conversacional tanto de software como hardware en el trabajo [32], en donde predominaba el uso de Pytorch pero la documentación era limitada frente a la de Tensorflow, además, se obtuvo en los trabajos [30] [37], que para el ajuste del modelo es conveniente tener equipos con unos requisitos mínimos de procesamiento para la unidad de procesamiento gráfico (GPU) y para la unidad de central de procesamiento (CPU).

La RSL ayudó inicialmente con la información necesaria a emplear en la selección y desarrollo del modelo, no obstante, las características y parámetros solo sirvieron como referencia para el uso de BERT con SQuAD 2.0, mas no como los cimientos de donde partir, frente al ajuste del modelo las variables halladas no se aplican a un modelo destinado a preguntas y respuestas ya que el modelo propuesto no se limita a longitudes de oración máxima de 64 como se observa en el trabajo [39] el cual, como otros trabajos se enfocan a trabajar con

oraciones y no con documentos completos los cuales sobrepasan los 512 caracteres de oración y el procesamiento que conlleva es significativamente mayor, además de las 3 épocas y tamaño de lote de 12 mencionados en el trabajo [42], para los cuales se emplea parámetros distintos y hardware más potente. Sin embargo, la RSL facilitó la elección del modelo, el cual fue: deepset/bert-base-cased-squad2 [80], ya que había sido previamente ajustado con SQuAD 2.0, y permitía emplearlo con el hardware ofrecido por Google Colaboratory, ya que las características de GPU y RAM eran superiores frente a las del desarrollo local. Como material exclusivamente necesario se utilizaron artículos que traten sobre temas tópicos del Covid-19, ya que en estos se sustentan las respuestas a las preguntas realizadas al agente virtual, para lo cual se seleccionó una base de datos científica conocida globalmente como es Semantic Scholar, esta base de datos tiene en su almacenamiento el conjunto de datos llamado CORD-19, al se puede acceder libremente y mediante el cual se aprovecha en las búsquedas de contexto para dar respuestas coherentes referentes al coronavirus, sin embargo, no todos los documentos tratan el Covid-19 como tema central si no como un término que emplean como referencia y puede producir sesgos en las respuestas, por lo que se recomienda realizar un preprocesamiento del dataset.

Objetivo 2: Desarrollar un agente virtual empleando NLP (Natural Language Processing) con el modelo de lenguaje BERT junto al conjunto de datos SQuAD

El presente objetivo fue desarrollado empleando la metodología Design Thinking, la primera actividad radicó en entrenar el modelo de lenguaje BERT, con la información de artículos científicos sobre información general del Covid-19, esta información fue obtenida mediante la herramienta Haystack Annotation Tool [77] ya que permitió solventar la tarea de crear manualmente un dataset para el ajuste y donde se obtuvo en formato SQuAD agilizando todo el proceso. Se aplicó el lenguaje Python como base junto a librerías como Pytorch y Transformers, permitiendo realizar el ajuste en una GPU Nvidia, donde se emplearon los parámetros obtenidos en trabajos como [37][42][45][46], que establecen un tamaño de lote mínimo de 16, tamaño de épocas de 4 como máximo y longitud máxima de la oración de 64, los cuales no concordaron con los parámetros finales del ajuste, que difirieron en tamaño de lote de 7, épocas de 7 y longitud máxima de oración de 384, empleando el truncamiento y zancada o stride en la longitud para solapar las partes que no entran en la longitud máxima de oración y poder buscar una respuesta si fuese necesario en el resto del documento, gracias a Haystack, se pudo configurar la búsqueda dentro de todo el cuerpo de los documentos y no solo limitarse al resumen o introducción ya que esto daría lugar a sesgos en las respuestas, a pesar de la diferencia en los parámetros se los pudo tomar como guía en el ajuste hasta llegar a la configuración idónea mediante diferentes pruebas como se presenta en la Tabla VIII. Los resultados del ajuste en comparación con otros trabajos como [48], se reflejan en la

precisión final del modelo ajustado en la que el modelo arrojó un 94.51 de precisión, frente al 85.02 de BioBERT-QA, un modelo preentrenado en tareas de minería de textos biomédicos con distintas funciones como reconocimiento de entidades nombradas biomédicas, extracción de relaciones, respuesta a preguntas, etc. Además, se corroboró el resultado del ajuste en la Figura 41, donde se puede verificar el resultado con y sin ajuste. Recalcando que las configuraciones y parámetros cambian en base a los tipos de datos con los que se trabaja y dependen de la función a la que se destine el modelo.

Como subactividad se obtuvo un conjunto de datos, de donde provienen las respuestas, el conjunto de datos CORD-19, se lo sometió a un preprocesamiento para filtrar los documentos completos que aporten información integra en todos los campos y evitar problemas posteriores en el procesamiento del modelo. Se analizaron los datos obtenidos donde se obtuvo que los documentos que se usaron corresponden a la pandemia del coronavirus y es el corpus ideal para trabajar con el modelo, al tener inconvenientes del hardware respecto a la GPU y con la máquina de desarrollo por los limitantes del basto tamaño del conjunto de datos se limitó a cerca de 400 artículos los cuales fueron filtrados por años y palabras clave, para evitar cuellos de botella en el procesamiento del modelo y facilitando a posterior la adición de nuevos documentos con el fin de mantener a la población informada y actualizada respecto al Covid-19. Como limitantes se puede obtener que no todas las respuestas serán contestadas ya cada día avanzan las investigaciones y para actualizar el dataset se necesitaría procesar cada nuevo documento y agregarlo al dataset mediante Elasticsearch.

Se realizó un análisis exploratorio al dataset, que aportó más conocimientos sobre los datos que se usa para dar respuestas, como los años de publicación (véase la Figura 42), palabras más empleadas en títulos (véase la Figura 43) y los términos más frecuentes en los documentos (véase Figura 44), arrojando un léxico relacionado con el Covid-19, excelente para obtener respuestas.

Los procedimientos ejecutados y sumados al modelo previamente entrenado con el conjunto de datos SQuAD nos permitió generar un modelo centrado en los temas sobre el Covid-19, generando respuestas a preguntas con un índice de acierto superior al 85% (véase la Tabla X), satisfaciendo el propósito del desarrollo del presente objetivo, dando a la población un medio para poder solucionar sus dudas referentes al Covid-19 basados en documentos científicos de las mejores bases de datos, sin la necesidad de recurrir a un profesional exponiéndose en el proceso de consulta.

Respecto al diseño y arquitectura del agente virtual se encontró con la limitante del desarrollo local por la exigencia que produce el sistema al procesar, por lo cual se recurrió a contenedores que gracias a su ligereza y eficiencia permitió implementar el agente en una arquitectura distribuida con una mayor eficacia en lo que respecta al tiempo de respuesta (véase la sección 6.2.3), siendo esta arquitectura una ventaja para su implementación, ya que

sin la arquitectura diseñada el tiempo incrementaría en un 50%, al no disponer de los recursos necesarios.

Continuando con la arquitectura y a su codificación en base al diseño (véase la sección 6.2.4), fue elaborado en el lenguaje Python por su relación al aprendizaje automático, que permite un desarrollo e implementación íntegra. El desarrollo del prototipo se obtuvo gracias a la aplicación de la metodología Desing Thinking, la cual permitió modificar, ajustar y subsanar cualquier error y volver a probarlo hasta obtener un prototipo apropiado, en el que se usó junto a los servicios de Microsoft Azure y Google Cloud, por la integración de contenedores y el levantamiento del sistema en línea. Cabe mencionar que el prototipo posee un menú de opciones desplegable para modificar el tipo de búsqueda e idioma, permitiendo obtener respuestas más precisas a cambio del tiempo que puede aumentar según el contexto de la pregunta realizada, donde se estima un tiempo de respuesta de entre 10 segundos a 30 segundos según las opciones seleccionadas. En consideración al ingreso de las preguntas, se apreció que las preguntas se pueden plantear y estructurar de distintas formas, lo que puede producir inexactitudes al momento de ser traducidas del español a inglés o viceversa por el extenso léxico de los lenguajes.

Objetivo 3: Evaluación de la herramienta.

Para la evaluación del prototipo una vez definido el modelo y desarrollado el agente virtual con el conjunto de datos del CORD-19, se obtuvo resultados satisfactorios en las respuestas ofrecidas por el agente, junto a la puntuación (score) producida por el modelo. Ya que el modelo se centra principalmente en devolver respuestas confiables, se recurrió a diferentes profesiones de la salud los cuales evaluaron la información devuelta por el modelo y basándose en sus conocimientos pudieron dar una retroalimentación respecto a las respuestas generadas, con el fin de establecer que tan confiables son.

Se obtuvo en una primera evaluación humana o human evaluation, por parte profesionales de la salud los siguientes resultados: el 60% de las respuestas fueron correctas, el 25% fueron respuestas incompletas con una leve falta de información, por lo que no se las considera como erróneas y el 15% fueron respuestas erróneas ofrecidas por el modelo, donde las respuestas no daban razón alguna referente a la pregunta (véase la Tabla X). Cotejado los primeros resultados de la evaluación contra los obtenidos posteriormente por una especialista en epidemiología en donde se obtuvo los siguientes resultados: un 40% de las respuestas correctas, un 45% incompletas y un 15%, son respuestas erróneas. Con lo que se puede determinar que el modelo responde correctamente, pero podría mejorar al devolver las respuestas con un contexto mayor, el cual aporte información integra, el inconveniente de mejorar las respuestas incompletas sería el reajuste que se debería hacer al modelo como se observa en la sección 6.2.1, donde se detalla todo el procedimiento de ajuste del modelo.

Los resultados obtenidos, justifican que el agente conversacional cumple con su propósito de aportar información válida sobre el Covid-19. Tanto las respuestas como el funcionamiento del mismo sistema se desempeñaron correctamente según lo configurado y ajustado hacia temas del Covid-19, por ende, a pesar de las respuestas incorrectas, se puede tomar las respuestas correctas e incompletas ya que la información faltante es leve, pero aportan información dando un parámetro final de un 85% de precisión el agente conversacional respecto a las respuestas producidas según la evaluación realizada por profesionales de la salud.

7.2. Valoración técnica, económica, ambiental y social

7.2.1. Valoración técnica

El presente TT se valora técnicamente mediante la integración de varias tecnologías y múltiples herramientas de software que fueron utilizadas conjuntamente con el objetivo de realizar un modelo basado en BERT y SQuAD que responda a preguntas comunes sobre el covid-19.

7.2.2. Valoración económica

Para la ejecución del presente TT fueron necesarios ciertos recursos económicos, los mismos que se presentan a continuación, en las **TABLA XII, TABLA XIII, TABLA XIV:**

TABLA XII. Recursos para talento humano

Talento Humano			
Responsable	Número de horas	Costo por hora	Costo total
Tesistas	480 (2)	\$10	\$4.800
Director	80	\$10.48	\$838.40
TOTAL			\$5,638.40

TABLA XIII. Recursos técnicos y tecnológicos

Recursos Técnicos y Tecnológicos	
Recursos de Software	
Nombre	Costo total
Mendeley Desktop	\$0.00
Google Chrome	\$0.00
Google Colab	\$20.00
Visual Studio IDE	\$0.00
Google Cloud	\$300.00
Microsoft Azure	\$100.00
Python	\$0.00
Zoom	\$0.00

Firma electrónica (2)		\$34.00	
SUBTOTAL		\$454.00	
Recursos de Hardware			
Nombre	Cantidad	Costo unitario	Costo total
Laptop Asus	1	\$1300.00	\$1300.00
Laptop Dell	1	\$1200.00	\$1200.00
SUBTOTAL		\$2500.00	
TOTAL		\$2954.00	

TABLA XIV. *Recursos para servicios*

Servicios			
Nombre	Meses	Costo unitario	Costo total
Internet 20Mb	5	\$25.00	\$125.00
TOTAL			\$125.00

De acuerdo a todos los recursos económicos anteriormente presentados, se genera la **TABLA XV**, en la que se presenta la sumatoria de cada uno de ellos, con el valor total de los gastos que llevó realizar el presente TT.

TABLA XV. *Totalidad de los recursos económicos*

Presupuesto General	
Descripción	Costo total
Talento Humano	\$5,638.40
Recursos Técnicos y Tecnológicos	\$3034.00
Servicios	\$125.00
TOTAL	\$8,797.40

7.2.3. Valoración ambiental

El presente TT se realizó en su totalidad con recursos tecnológicos y digitales que no tienen un mayor impacto al medio ambiente, salvo el consumo eléctrico, además que se emplearon pocos recursos materiales o de otros elementos que puedan llegar a perjudicar al planeta.

7.2.4. Valoración social

En el ámbito social, el presente TT tiene una afanosa valoración ya que determinó la necesidad de profesionales de la salud que den respuestas a la población que tiene dudas sobre temas relacionados con el Covid-19. Ya que en los picos más altos de contagio del

coronavirus los hospitales estuvieron al borde del colapso por la cantidad de infectados y no podían atender a dudas menores que surgían sobre el coronavirus en la gente. Por lo cual se consultó de que sí un agente conversacional ayudaría a evitar el colapso en los centros médicos solventando los cuellos de botella producidos por la atención a pacientes graves por el covid-19, lo que de acuerdo a varias entrevistas como por ejemplo, la realizadas a enfermeras que se desempeñaron en centros médicos, aceptaron que si sería viable ya que descongestionaría los centros de atención y evitarían la exposición y el riesgo que conlleva acercarse a donde se encuentran internados los pacientes. Además, que puede ser accesible a la población en general sin importar su clase social, política o religiosa, lo que apoyaría a los más desfavorecidos, así como a la población más vulnerable.

8. Conclusiones

De acuerdo al trabajo de titulación realizado, se puede concluir lo siguiente:

- El desarrollo del agente conversacional permitió conocer sobre el uso de modelos de lenguaje como BERT, el cual principalmente se lo utiliza en diferentes tareas no relacionadas a dar respuesta a preguntas, así como los parámetros base que se emplean para su uso, lo que permitió indagar y experimentar con distintos parámetros, logrando establecer que para longitudes de oración mayores de 512 caracteres lo mejor es emplear diversos parámetros como el truncamiento o truncation de la oración, el relleno o padding, zancada o stride y un tamaño de 384 de longitud máxima, lo que facilita el ajuste y funcionamiento posterior del modelo, ya que al procesar documentos se deben analizar al completo para evitar sesgos en las respuestas.
- De acuerdo a la revisión sistemática de literatura se determinaron las características y configuraciones óptimas para el modelo de lenguaje BERT, gracias a lo cual se pudo elegir el modelo adecuado para el desarrollo del proyecto, en este caso el modelo BERT case, ya que se desempeña mejor con bajos recursos. Así mismo los parámetros óptimos establecidos para el ajuste del modelo se establecieron en una la longitud máxima de oración de 512, batch de 7, una la tasa de aprendizaje de 5e-5 y un número de épocas de entrenamiento de 7. Tras obtener una pérdida baja en numerosas pruebas, se generó un modelo ajustado con una precisión del 94.5%. Además, se pudo conocer las principales preguntas que han realizado los usuarios entorno al covid-19, las cuales se relacionan con, el lugar de procedencia, síntomas, distanciamiento, prevención de la propagación, pruebas de diagnóstico, su precisión, tiempo, resultados, estadísticas de mortalidad y anticuerpos, estos resultados aportaron a las búsquedas de artículos para el afinado del modelo.
- Al reducir el conjunto de datos CORD19 en el proceso de filtrado, puede ocasionar limitaciones respecto a la generación de respuestas, debido a la menor la cantidad de documentos que existen en la base de conocimientos, ya que el fin de la reducción de documentos fue centrarse en los temas más comunes del covid-19 y buscar el rendimiento óptimo de todo el sistema, lo que puede dar lugar a sesgos en temas más profundos sobre el Covid-19, los cuales requieren muchos más documentos para ofrecer respuestas correctas.
- Finalmente, por medio de una evaluación al modelo, se obtuvieron respuestas correctas provenientes del modelo BERT ajustado al ámbito del Covid-19, la cual fue realizada por profesionales de la salud, los cuales validaron que en un 85% de las respuestas existía

concordancia. Posteriormente se reafirmaron estos resultados con una segunda evaluación realizada por parte de una especialista en epidemiología la cual corroboró que solamente el 15% de las respuestas fueron incorrectas. Este porcentaje obtenido es el aceptado en modelos ajustados, ya que con menos 80% no se lo puede considerar óptimo, pero se debe aclarar que para temas relacionados con la salud podría ser mejor y el sesgo de respuestas malas podría causar desinformación. Por lo tanto, se puede considerar al Agente Virtual como un servicio para la ciudadanía que tenga interrogantes sobre el Covid-19, pero recalando que cualquier información sensible respecto a la enfermedad se debería consultar con un médico.

9. Recomendaciones

En base al Trabajo de Titulación elaborado, se puede dar las siguientes recomendaciones:

- En el proceso de ajuste o fine tune, lo óptimo es el uso de modelos preentrenados, ya que estos, aunque sea necesario afinarlos para obtener resultados óptimos, permiten ahorrar mucho tiempo y recursos computacionales. Además, el uso de estos modelos, especializados en un campo en específico permite la creación de diferentes sistemas los cuales podrían de cierta manera suplir o ayudar a diferentes profesionales en sus campos.
- En la etapa de selección de datos, para afinar (fine-tune) el modelo en un campo específico, los datos deben pasar por una clasificación y filtrado óptimo para lograr realizar un buen ajuste y que así el modelo se desempeñe correctamente a posterior.
- Se deben emplear datos actualizados, que no superen en gran medida el año de publicación ya que, en este caso frente a virus o enfermedades no comunes, no se desarrollan investigaciones suficientes que aporten y sustenten el desarrollo del proyecto, sea el caso de la investigación de vacunas contra el covid-19, que tuvo que transcurrir un año para su investigación, aprobación y desarrollo.
- El ajuste del modelo y su posterior uso emplea un hardware potente en lo referente a procesamiento gráfico, es decir se necesita tarjetas gráficas (GPU) con una memoria igual o superior a los 12 Gb de memoria RAM para realizar un ajuste acorde a las especificaciones dadas por el modelo BERT, en cuanto al tamaño de lote y épocas que se establecen como parámetros de entrenamiento.
- Para la optimización del sistema respecto al modelo BERT y su desempeño al emplear la biblioteca Tokenizer, se debe reducir o segmentar los documentos que se vayan a procesar, ya que la tokenización de varios documentos conlleva un tiempo determinado que puede influir al dar una respuesta.

9.1. Trabajos futuros

Como trabajos futuros se propone:

- Ampliar la base de conocimiento ya que está limitada por las características del hardware disponible, ya que se limita a 16 GB de memoria RAM, lo cual limita la cantidad de artículos de la base a 50 mil.
- Actualizar la base de conocimientos a los últimos documentos de investigación sobre el COVID-19, debido a las constantes investigaciones y desarrollo de nuevas medidas frente a la pandemia, por la importancia que esto supone para la estabilidad social.
- Mejorar las características de las máquinas virtuales que alojan a los contenedores para obtener mayor capacidad de procesamiento, lo que equivaldría a una mejoría en la velocidad de respuesta del agente virtual.
- Con todo lo anterior mencionado aplicar al modelo un ajuste más optimo, para que así el agente logre responder con una precisión más acorde a los últimos avances, empleando un tiempo menor de espera.
- Comparar el desempeño del agente con el rendimiento de otros modelos como RoBERTa, DistilBERT, BioBERT; ya que estos últimos han sido desarrollados a partir del modelo BERT base y tienen una base de conocimientos más desarrollada.
- Exportar la aplicación web a plataformas móviles en concreto a: Android e iOS.

10. Bibliografía

- [1] E. D. Gamboa Teneta and I. Hernán Fabricio Naranjo Ávalos Mg ii, "Prototipo de un chatbot para compras online utilizando bot framework," Universidad Técnica de Ambato, 2019.
- [2] C. L. Mendoza Durán and S. A. Pedraza Gutiérrez, "Asistente virtual web basado en inteligencia artificial para la escuela tic de la universidad piloto de Colombia," Universidad Piloto de Colombia, 2018.
- [3] L. García, "Asistente Virtual Tipo Chatbot," Universidad Católica de Colombia, 2018.
- [4] Q. Wang *et al.*, "Learning to Extract Attribute Value from Product via Question Answering: A Multi-Task Approach," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 47–55, doi: 10.1145/3394486.3403047.
- [5] W. Siblini, C. Pasqual, A. Lavielle, M. Challal, and C. Cauchois, "Multilingual Question Answering from Formatted Text applied to Conversational Agents," Oct. 2019, Accessed: Jun. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1910.04659>.
- [6] B. Zhong, W. He, Z. Huang, P. E. D. Love, J. Tang, and H. Luo, "A building regulation question answering system: A deep learning methodology," *Adv. Eng. Informatics*, vol. 46, p. 101195, Oct. 2020, doi: 10.1016/j.aei.2020.101195.
- [7] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.)*, vol. 2, pp. 784–789, Jun. 2018, Accessed: May 28, 2021. [Online]. Available: <http://arxiv.org/abs/1806.03822>.
- [8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Oct. 2019, vol. 1, pp. 4171–4186, Accessed: May 26, 2021. [Online]. Available: <https://github.com/tensorflow/tensor2tensor>.
- [9] "Redes Transformer (... o el fin de las Redes Recurrentes) | Codificando Bits." <https://www.codificandobits.com/blog/redes-transformer/> (accessed Sep. 19, 2021).
- [10] "PyTorch documentation — PyTorch 1.9.0 documentation." <https://pytorch.org/docs/stable/index.html#> (accessed Aug. 11, 2021).
- [11] "Transformers — transformers 4.7.0 documentation." <https://huggingface.co/transformers/> (accessed Aug. 11, 2021).
- [12] G. Amati, "BM25," *Encycl. Database Syst.*, pp. 257–260, 2009, doi: 10.1007/978-0-387-

39940-9_921.

- [13] A. Vaswani *et al.*, “Attention Is All You Need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 5999–6009, Jun. 2017, Accessed: Sep. 17, 2021. [Online]. Available: <https://arxiv.org/abs/1706.03762v5>.
- [14] “Googletrans: Free and Unlimited Google translate API for Python — Googletrans 3.0.0 documentation.” <https://py-googletrans.readthedocs.io/en/latest/> (accessed Sep. 17, 2021).
- [15] E. N. Preeth, J. P. Mulerickal, B. Paul, and Y. Sastri, “Evaluation of Docker containers based on hardware utilization,” *2015 Int. Conf. Control. Commun. Comput. India, ICCC 2015*, pp. 697–700, Mar. 2016, doi: 10.1109/ICCC.2015.7432984.
- [16] “Haystack Docs.” <https://haystack.deepset.ai/overview/intro> (accessed Mar. 02, 2022).
- [17] “Haystack Docs.” <https://haystack.deepset.ai/docs/latest/retrievermd> (accessed Mar. 02, 2022).
- [18] “Haystack Docs.” <https://haystack.deepset.ai/docs/readermd> (accessed Mar. 02, 2022).
- [19] “Welcome to the FARM! — FARM documentation.” <https://farm.deepset.ai/> (accessed Mar. 02, 2022).
- [20] “¿Qué es Elasticsearch? | Elastic.” <https://www.elastic.co/es/what-is/elasticsearch> (accessed Mar. 02, 2022).
- [21] “Streamlit documentation.” <https://docs.streamlit.io/> (accessed Mar. 02, 2022).
- [22] “Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure.” <https://azure.microsoft.com/es-es/overview/what-is-azure/> (accessed Mar. 02, 2022).
- [23] “Azure para estudiantes universitarios: detalles de la oferta | Microsoft Azure.” <https://azure.microsoft.com/es-es/offers/ms-azr-0170p/> (accessed Mar. 02, 2022).
- [24] “FastAPI.” <https://fastapi.tiangolo.com/es/> (accessed Mar. 02, 2022).
- [25] “What exactly happens when we fine-tune BERT? | by Samuel Flender | Feb, 2022 | Towards Data Science.” <https://towardsdatascience.com/what-exactly-happens-when-we-fine-tune-bert-f5dc32885d76> (accessed Mar. 10, 2022).
- [26] “Te damos la bienvenida a Colaboratory - Colaboratory.” <https://colab.research.google.com/?hl=es> (accessed Mar. 10, 2022).
- [27] “Semantic Scholar | About Us.” <https://www.semanticscholar.org/about> (accessed Sep. 07, 2021).
- [28] “Servicios de cloud computing | Google Cloud.” <https://cloud.google.com/?hl=es> (accessed Mar. 18, 2022).
- [29] K. S. Galarza Veriñaz, “Implementación de un prototipo agente virtual basado en machine learning e inteligencia artificial aplicando Azure Bot Service para mejorar el servicio de una empresa de venta de soluciones tecnológicas.,” Universidad Católica de Santiago de Guayaquil, 2020.

- [30] H. Liu, Y. Perl, and J. Geller, "Transfer Learning from BERT to Support Insertion of New Concepts into SNOMED CT," *AMIA ... Annu. Symp. proceedings. AMIA Symp.*, vol. 2019, pp. 1129–1138, 2019.
- [31] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, "Deep Learning Based Fusion Approach for Hate Speech Detection," *IEEE Access*, vol. 8, pp. 128923–128929, 2020, doi: 10.1109/ACCESS.2020.3009244.
- [32] M. H. Su, C. H. Wu, and H. T. Cheng, "A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2061–2072, 2020, doi: 10.1109/TASLP.2020.3006731.
- [33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 4171–4186, Oct. 2018, Accessed: Mar. 15, 2021. [Online]. Available: <http://arxiv.org/abs/1810.04805>.
- [34] K. El Maghraoui, L. M. Herger, C. Choudary, K. Tran, T. Deshane, and D. Hanson, "Performance Analysis of Deep Learning Workloads on a Composable System," no. 1, pp. 1–10, Mar. 2021, Accessed: May 26, 2021. [Online]. Available: <http://arxiv.org/abs/2103.10911>.
- [35] R. G. Bruke Mammo, Praveer Narwelkar, "Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning," pp. 1–25, 2018.
- [36] X. Yang, H. Zhang, X. He, J. Bian, and Y. Wu, "Extracting family history of patients from clinical narratives: Exploring an end-to-end solution with deep learning models," *JMIR Med. Informatics*, vol. 8, no. 12, Dec. 2020, doi: 10.2196/22982.
- [37] A. Özçift, K. Akarsu, F. Yumuk, and C. Söylemez, "Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish," *Automatika*, 2021, doi: 10.1080/00051144.2021.1922150.
- [38] N. Chintalapudi, G. Battineni, and F. Amenta, "Sentimental analysis of COVID-19 tweets using deep learning models," *Infect. Dis. Rep.*, vol. 13, no. 2, Apr. 2021, doi: 10.3390>IDR13020032.
- [39] E. Hulburg, "Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0," Feb. 2020, Accessed: May 26, 2021. [Online]. Available: <http://arxiv.org/abs/2002.10670>.
- [40] L. Su, J. Guo, Y. Fan, Y. Lan, and X. Cheng, "Controlling Risk of Web Question Answering," *SIGIR 2019 - Proc. 42nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 115–124, May 2019, doi: 10.1145/3331184.3331261.
- [41] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for

machine comprehension of text," in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, Jun. 2016, pp. 2383–2392, doi: 10.18653/v1/d16-1264.

- [42] J. Ayoub, X. J. Yang, and F. Zhou, "Combat COVID-19 infodemic using explainable natural language processing models," *Inf. Process. Manag.*, vol. 58, no. 4, Jul. 2021, doi: 10.1016/j.ipm.2021.102569.
- [43] M. El-Geish, "Gestalt: a Stacking Ensemble for SQuAD2.0," Apr. 2020, Accessed: May 26, 2021. [Online]. Available: <http://arxiv.org/abs/2004.07067>.
- [44] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with BERT," *IEEE Access*, vol. 7, pp. 154290–154299, 2019, doi: 10.1109/ACCESS.2019.2946594.
- [45] D. Chang, W. S. Hong, and R. A. Taylor, "Generating contextual embeddings for emergency department chief complaints," *JAMIA Open*, vol. 3, no. 2, pp. 160–166, Jul. 2020, doi: 10.1093/jamiaopen/ooaa022.
- [46] K. Zeng, Z. Pan, Y. Xu, and Y. Qu, "An ensemble learning strategy for eligibility criteria text classification for clinical trial recruitment: Algorithm development and validation," *JMIR Med. Informatics*, vol. 8, no. 7, Jul. 2020, doi: 10.2196/17832.
- [47] P. Vinod, S. Safar, D. Mathew, P. Venugopal, L. M. Joly, and J. George, "Fine-tuning the BERTSUMEXT model for clinical report summarization," in *2020 International Conference for Emerging Technology, INCET 2020*, Jun. 2020, doi: 10.1109/INCET49848.2020.9154087.
- [48] T. Mairitha, N. Mairitha, and S. Inoue, "Improving Fine-Tuned Question Answering Models for Electronic Health Records," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 688–691, doi: 10.1145/3410530.3414436.
- [49] F. C. Fang, S. N. Naccache, and A. L. Greninger, "The laboratory diagnosis of coronavirus disease 2019-frequently asked questions," *Clinical Infectious Diseases*, vol. 71, no. 11. Oxford University Press, pp. 2996–3001, Dec. 01, 2020, doi: 10.1093/cid/ciaa742.
- [50] J. Kishore, "Frequently Asked Questions (FAQs) about Coronavirus Disease 2019 (Covid-19)," *Epidemiol. Int.*, vol. 05, no. 01, pp. 14–21, 2020, doi: 10.24321/2455.7048.202004.
- [51] C. R. Aquino-Canchari, S. G. Chávez-Bustamante, and B. S. Caira-Chuquineyra, "Exploratory analysis of internet search trends during the COVID-19 outbreak, Análisis exploratorio de las tendencias de búsqueda en internet durante el brote de la COVID-19," *Rev. Cuba. Inf. en Ciencias la Salud*, 2020, doi: 10.36512/rcics.v31i3.1631.

- [52] A. M. López, P. de Casas Moreno, and J. M. B. Mera, “Informative treatment and media competencies on COVID-19 in Ecuador,” *Rev. Comun.*, vol. 20, no. 1, pp. 137–152, Mar. 2021, doi: 10.26441/RC20.1-2021-A8.
- [53] “Aplicaciones de búsqueda de Google para la comunicación de riesgos en el manejo de la COVID-19.” http://scielo.sld.cu/scielo.php?pid=S1561-31942020000400015&script=sci_arttext&tlang=pt (accessed Jun. 22, 2021).
- [54] O. Frequently and A. Questions, “Frequently Asked Questions and Answers,” *Is My J. Elig. to be Index. Int. Databases?* 3rd Ed., vol. 2019, 2011, doi: 10.5082/xmlarchive.2010.1.13.
- [55] A. Husnayain, A. Fuad, and E. C. Y. Su, “Applications of Google Search Trends for risk communication in infectious disease management: A case study of the COVID-19 outbreak in Taiwan,” *Int. J. Infect. Dis.*, vol. 95, pp. 221–223, Jun. 2020, doi: 10.1016/j.ijid.2020.03.021.
- [56] “Frequently Asked Questions and Myth Busters on COVID-19.” <http://www.saraya.com.au/index.php/about-us/news/item/frequently-asked-questions-and-myth-busters-on-covid-19> (accessed Jun. 22, 2021).
- [57] D. Hu *et al.*, “More effective strategies are required to strengthen public awareness of COVID-19: Evidence from google trends,” *J. Glob. Health*, vol. 10, no. 1, Jun. 2020, doi: 10.7189/jogh.10.011003.
- [58] A. Rajan, R. Sharaf, R. S. Brown, R. Z. Sharaiha, B. Lebwohl, and S. H. Mahadev, “Association of search query interest in gastrointestinal symptoms with COVID-19 diagnosis in the United States: Infodemiology study,” *JMIR Public Heal. Surveill.*, vol. 6, no. 3, Jul. 2020, doi: 10.2196/19354.
- [59] H. L. Nguyen, Z. Pan, H. Abu-gellban, F. Jin, and Y. Zhang, “Google Trends Analysis of COVID-19,” Nov. 2020, Accessed: Jun. 22, 2021. [Online]. Available: <http://arxiv.org/abs/2011.03847>.
- [60] T. S. Higgins *et al.*, “Correlations of online search engine trends with coronavirus disease (COVID-19) incidence: Infodemiology study,” *JMIR Public Heal. Surveill.*, vol. 6, no. 2, Apr. 2020, doi: 10.2196/19702.
- [61] F. Arias, “El proyecto de Investigacion Introducción a la metodología Científica,” *J. Chem. Inf. Model.*, vol. 53, no. 9, p. 144, 2012, Accessed: Aug. 11, 2021. [Online]. Available: <https://ebevidencia.com/wp-content/uploads/2014/12/EL-PROYECTO-DE-INVESTIGACIÓN-6ta-Ed.-FIDIAS-G.-ARIAS.pdf>.
- [62] Y. Castán, “Metodo Científico y Sus Etapas,” *Castán, Yolanda. Inst. Aragon. Ciencias La Salud*, vol. 2, pp. 1–6, 2006.
- [63] C. Walker, T. Nolen, J. Du, and H. Davis, “Applying Design Thinking:,” pp. 19–19, 2019, doi: 10.1145/3347709.3347775.

- [64] “Guidelines for performing Systematic Literature Reviews in Software Engineering | BibSonomy.”
<https://www.bibsonomy.org/bibtex/aed0229656ada843d3e3f24e5e5c9eb9> (accessed May 28, 2021).
- [65] Y. Kodratoff, “About knowledge discovery in texts: A definition and an example,” *Unpubl. Pap.*, pp. 1–16, 1999, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.183.2058&rep=rep1&type=pdf>.
- [66] “Intelligent Systems at the Service of the Mankind, vol. 1.”
https://www.researchgate.net/publication/262010081_Intelligent_Systems_at_the_Service_of_the_Mankind_vol_1 (accessed Aug. 11, 2021).
- [67] “COVID-19 Open Research Dataset Challenge (CORD-19) | Kaggle.”
<https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge> (accessed Aug. 11, 2021).
- [68] “spaCy · Industrial-strength Natural Language Processing in Python.” <https://spacy.io/> (accessed Sep. 07, 2021).
- [69] “sklearn metrics f1_score documentation ” https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (accessed Aug. 11, 2021).
- [70] “Parsifal.” <https://parsif.al/> (accessed May 28, 2021).
- [71] R. Vega, “Análisis de Sentimientos en Twitter para Descubrir Contenido Xenófobo hacia los Inmigrantes Venezolanos en Ecuador,” Universidad nacional de Loja, 2021.
- [72] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Blackwell Publishing Ltd, 2008.
- [73] A. Otegi, J. A. Campos, G. Azkune, A. Soroa, and E. Agirre, “Automatic Evaluation vs. User Preference in Neural Textual QuestionAnswering over COVID-19 Scientific Literature,” 2020, doi: 10.18653/v1/2020.nlp covid19-2.15.
- [74] A. Balagopalan, B. Eyre, J. Robin, F. Rudzicz, and J. Novikova, “Comparing Pre-trained and Feature-Based Models for Prediction of Alzheimer’s Disease Based on Speech,” *Front. Aging Neurosci.*, vol. 13, Apr. 2021, doi: 10.3389/fnagi.2021.635945.
- [75] L. Su, J. Guo, Y. Fan, Y. Lan, and X. Cheng, “Controlling risk of web question answering,” *SIGIR 2019 - Proc. 42nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 115–124, 2019, doi: 10.1145/3331184.3331261.
- [76] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 4171–4186, Oct. 2018, Accessed: Sep. 07, 2021. [Online]. Available: <https://arxiv.org/abs/1810.04805v2>.

- [77] “Haystack Annotation Tool.” <https://annotate.deepset.ai/projects> (accessed Sep. 05, 2022).
- [78] “GitHub - deepset-ai/COVID-QA: API & Webapp to answer questions about COVID-19. Using NLP (Question Answering) and trusted data sources.” <https://github.com/deepset-ai/COVID-QA> (accessed Sep. 05, 2022).
- [79] J. Carrión and V. Serrano, “Revisión sistemática de literatura: características y funcionamiento respecto a los modelos BERT y SQuAD,” CEDAMAZ, vol. 11, no. 1, pp. 79–86, Jul. 2021, Accessed: Mar. 15, 2022.
- [80] “deepset/bert-base-cased-squad2 · Hugging Face.” <https://huggingface.co/deepset/bert-base-cased-squad2> (accessed Sep. 05, 2022).

11. Anexos

Anexo 1: Revisión Sistemática de Literatura I

A continuación, se adjunta el producto final de la Revisión Sistemática de Literatura, presentado como artículo.

Revisión sistemática de literatura: características y funcionamiento respecto a los modelos BERT y SQuAD

Systematic literature review: characteristics and functioning of the BERT and SQuAD models.

José Carrión^{1,*} y Victor Serrano¹

¹*Carrera de Ingeniería en Sistemas/Computación, Universidad Nacional de Loja, Loja, Ecuador*
jose.a.carrion.o@unl.edu.ec, victor.serrano@unl.edu.ec

*Autor para correspondencia: *jose.a.carrion.o@unl.edu.ec*

Fecha de recepción del manuscrito: 27/06/2021 Fecha de aceptación del manuscrito: 07/07/2021 Fecha de publicación: 15/07/2021

Resumen—En la actualidad con la pandemia que se padece, se han producido colapsos en el sistema de salud lo que ha ocasionado pérdidas humanas y económicas en mayor parte, ha provocado el resguardo de la población y a limitado el acceso a centros de salud. Lo que ha provocado decesos en la población por no poder tener acceso a atención médica básica, como pueden ser consultas sobre los principales síntomas. La presente Revisión Sistemática de Literatura (RSL) asumió el propósito de identificar qué características y funcionamiento óptimo son necesarios para empleo de BERT y SQuAD con el fin de desarrollar posteriormente un agente virtual centrado en dar Respuesta a Preguntas sobre temas comunes del Covid-19. Ya que al no dar abasto los centros de salud el agente ofrecería una mayor cobertura en temas de asistencia sobre el covid a la población. La presente RSL se basó en las fases de la metodología de Bárbara Kitchenham, la revisión se planteó en base a tres preguntas de investigación y definió el transcurso de la revisión; obteniendo a PyTorch y TensorFlow como frameworks para el desarrollo de software, como lenguaje programación a Python por su vinculación en aprendizaje automático, el modelo BERT BASE empleado para hardware de pocos recursos y SQuAD 2.0 por ser más completo respecto a pares de preguntas y respuestas razonables.

Palabras clave—BERT, SQuAD, Covid, Respuestas a preguntas, Agentes conversacionales.

Abstract—Currently, with the current pandemic, there have been collapses in the health system, which has caused human and economic losses in most cases, has caused the protection of the population and has limited access to health centers. This has caused deaths in the population due to lack of access to basic medical care, such as consultations on the main symptoms. This Systematic Literature Review (SLR) was undertaken to identify what features and optimal performance are necessary for the use of BERT and SQuAD in order to further develop a virtual agent focused on answering questions on common Covid-19 topics. The agent would provide greater coverage of Covid assistance issues to the population, since the health centers are not able to meet the needs of the population. The present RSL was based on the phases of Barbara Kitchenham's methodology, the review was based on three research questions and defined the course of the review; obtaining PyTorch and TensorFlow as frameworks for software development, Python as programming language for its linkage in machine learning, the BERT BASE model used for low-resource hardware and SQuAD 2.0 for being more complete with respect to pairs of questions and reasonable answers.

Keywords—BERT, SQuAD, Covid, Answers to Questions, Conversational agents.

INTRODUCCIÓN

Actualmente la población por temas relacionados al Covid19, no puede acceder a la sanidad pública ni privada, lo que genera miedo e inestabilidad en las familias al no poder consultar a un profesional de la salud cuando se encuentran con síntomas leves. Las condiciones de los centros de salud de solo atender a pacientes graves no

suponen un alivio para las personas, si a esto se le suma la desinformación que circula sobre diferentes temas de la pandemia la situación se complica aún más (Ayoub et al., 2021). Surgen formas de ayudar a la gente en los momentos de duda respecto al Covid-19 con la ayuda de la tecnología y en este caso mediante agentes conversacionales. En la presente RSL, mediante la metodología de

Bárbara Kitchenham y empleando sus fases se ha podido conseguir artículos sobre estudios y trabajos relacionados que permitieron identificar la información correspondiente a las características y funcionamiento de los modelos BERT y SQuAD.

El presente estudio se realizó en secciones como: la sección de Metodología donde se definieron las fases a seguir que propone Barbara Kitchenham para realizar revisiones de literatura, posteriormente, se realiza el proceso para cada una de las fases propuestas detallando el proceso y las salidas en la sección de Resultados, en la sección de Discusión para analizar, explicar e interpretar los resultados y finalmente, se plantean las Conclusiones obtenidas durante la realización de la presente RSL.

MATERIALES Y MÉTODOS

La realización de la revisión sistemática de literatura se basa en el proceso de la metodología de Bárbara Kitchenham (Kitchenham Charters, 2007), dicho proceso se resume en tres fases principales:

- Planificación de la revisión
- Realizar de la revisión
- Revisión de informes

De acuerdo a Kitchenham (Kitchenham Charters, 2007), algunas tareas de estas fases no son obligatorias, como por ejemplo:

- Puesta en marcha de una revisión, depende de la revisión sistemática que se está haciendo sobre una base comercial.
- Evaluar el protocolo de revisión y la evaluación del informe, son opcionales y dependen de los procedimientos de garantía de calidad decididas por el encargado de la revisión sistemática

Para la presente RSL no se ejecutaron todos los pasos propuestos por Kitchenham, justamente porque su metodología es flexible en cuanto a la extensión que el investigador necesite dar a su revisión y depende del investigador el alcance que requiera dar a la presentación y publicación de sus resultados. De acuerdo a esto, en la TABLA 1, se presentan las fases y tareas que fueron consideradas:

Tabla 1: Proceso de revisión sistemática de literatura propuesto

Fases	Tareas
Planificación de la revisión	Identificación de la necesidad de una revisión.
	Especificación de las preguntas de investigación.
	El desarrollo de un protocolo de revisión.
Realizar la revisión	Identificación de la investigación. Selección de los estudios primarios. Síntesis de los datos
Revisión de informes	Dar formato al informe principal.

RESULTADOS

Planificación de la revisión

Identificación de la necesidad de una revisión

Mediante la ejecución de una revisión sistemática de literatura, se puede encontrar el conocimiento ya existente acerca de un tema en particular, en este caso es necesaria para optar por las configuraciones y parámetros necesarios para el desarrollo de un agente virtual; así mismo conocer las diferentes versiones existentes de los modelos BERT y SQuAD que se emplearan en su construcción. Por consecuente, para abarcar este tema se propusieron tres preguntas de investigación para obtener información concisa y relevante.

Preguntas de investigación

En la TABLA 2, se presentan las preguntas de investigación que guiaron el desarrollo de la revisión sistemática de literatura, estas preguntas son:

Tabla 2: Preguntas de investigación

Preguntas de investigación	
P1	¿Cuáles son las características principales sobre el funcionamiento del modelo BERT y SQuAD?
P2	¿Cuáles son los parámetros que se emplean en los modelos de BERT con SQuAD?
P3	¿Qué versión de los modelos BERT y SQuAD es factible para el desarrollo del agente virtual?

El desarrollo de un protocolo de revisión

Diseño del protocolo de búsqueda

Estrategias de búsqueda

Petticrew y Roberts (Petticrew Roberts, 2008), recomiendan criterios para plantear la investigación, mediante el uso de la nemotécnica PICOC (Población, Intervención, Comparación, Resultado y Contexto), se estructuran los cinco componentes para definir la cadena de búsqueda; así como también, para una mejor organización y selección de todos los resultados, se utilizó la herramienta en línea Parsifal1, la cual ayuda en el contexto de la Ingeniería del Software para realizar revisiones sistemáticas de literatura, gracias a esta herramienta que permite dar seguimiento a la RSL, la cual está diseñada para abarcar todos los campos necesarios y obtener resultados óptimos.

Fuentes bibliográficas

Como fuentes bibliográficas, se ha seleccionado algunas bibliotecas virtuales, como son:

- ACM Digital Library (<https://dl.acm.org/>)
 - IEEE Digital Library (<https://ieeexplore.ieee.org/>)
 - Scopus (<http://www.scopus.com>)
- Además, se realizó la investigación con ayuda del buscador:
- Google Scholar (<https://scholar.google.com/>)

Tabla 3: Scripts de Búsqueda

Bibliotecas virtuales	Cadenas de búsqueda
ACM Digital Library	[[Publication Title: bert] OR [Publication Title: squad]] AND [All: performance of machine learning models] AND [Keywords: bert] AND [Abstract: answers to questions]
IEEE Digital Library	"BERT.^AND "SQuAD.^and configuration.^AND "training.^AND "parameters.^AND "Bidirectional Encoder Representations from Transformers.^AND "Stanford Question Answering Dataset.^AND performance of machine learning models
Google Scholar	("Título de la publicación": BERT) Y (Resumen": SQuAD) Y (Resumen": configuración) O (Resumen": formación) Y (Resumen": parámetros) Y (Resumen": Representaciones de codificador bidireccional de Transformers) O (Resumen": conjunto de datos de respuesta a preguntas de Stanford) Y (-resumen": rendimiento de los modelos de aprendizaje automático)
Scopus	(TITLE-ABS-KEY (bert AND bidirectional AND encoder AND representations AND from AND transformer) OR TITLE-ABS-KEY (squad) AND TITLE-ABS-KEY (transformer AND models) OR TITLE-ABS-KEY (question AND answering AND systems) OR TITLE-ABSKEY (performance AND of AND the AND bert AND models) AND TITLE-ABS-KEY (performance AND of AND machine AND learning AND models)) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018)) AND (LIMIT-TO (DOCTYPE , .ar")) AND (LIMIT-TO (LANGUAGE , .English"))

Tabla 4: Resumen de estudios seleccionados

Fuente	Encontrados	Seleccionados
ACM Digital Library	4	2
IEEE Digital Library	5	3
Google Scholar	11	9
Scopus	27	18
Total	47	32

Se obtuvieron 32 estudios relacionados, los mismos que luego de pasar por la evaluación de calidad, resultaron en 21

Definir palabras claves para el problema de estudio

Con la definición de los criterios PICOC se obtuvo un conjunto de palabras claves, las mismas que permitieron construir las cadenas de búsquedas, estas son: **BERT, Bidirectional Encoder, Representations from Transformer, Configuration, Model trained, Parameters training, SQuAD, Stanford Question Answering Dataset, Transformers parameters.**

Cadenas de búsqueda

En la TABLA 3, se presentan las cadenas de búsqueda aplicadas que se realizaron de acuerdo a cada biblioteca virtual: ACM Digital Library, IEEE Digital Library, Google Scholar, y Scopus.

Criterios de inclusión

Durante la búsqueda, se consideraron los siguientes criterios de inclusión:

- Solo documentos que sean artículos científicos.
- Se mencione BERT.
- Contenga características de BERT y SQuAD
- Idioma: Inglés-Español
- Publicaciones superiores a los últimos 4 años

Se consideraron los siguientes criterios de exclusión

- Documentos que no sean artículos científicos.
- Idiomas que no sean inglés o español.
- Publicaciones mayores a los últimos 4 años.
- Que no mencionen los modelos BERT o SQuAD

Planificación de la revisión

Identificación de la investigación

El objetivo de la presente revisión sistemática de literatura es dar respuesta a las preguntas de investigación planteadas, mediante la búsqueda de estudios primarios que aporten información verídica con respecto al tema. Selección de los estudios primarios.

Para realizar la selección de los estudios se ha seguido el proceso que a continuación se muestra en la FIGURA 1.

En la TABLA 4, se presenta un resumen de los trabajos o estudios relacionados que fueron encontrados junto con el número de estudios seleccionados según la fuente bibliográfica.

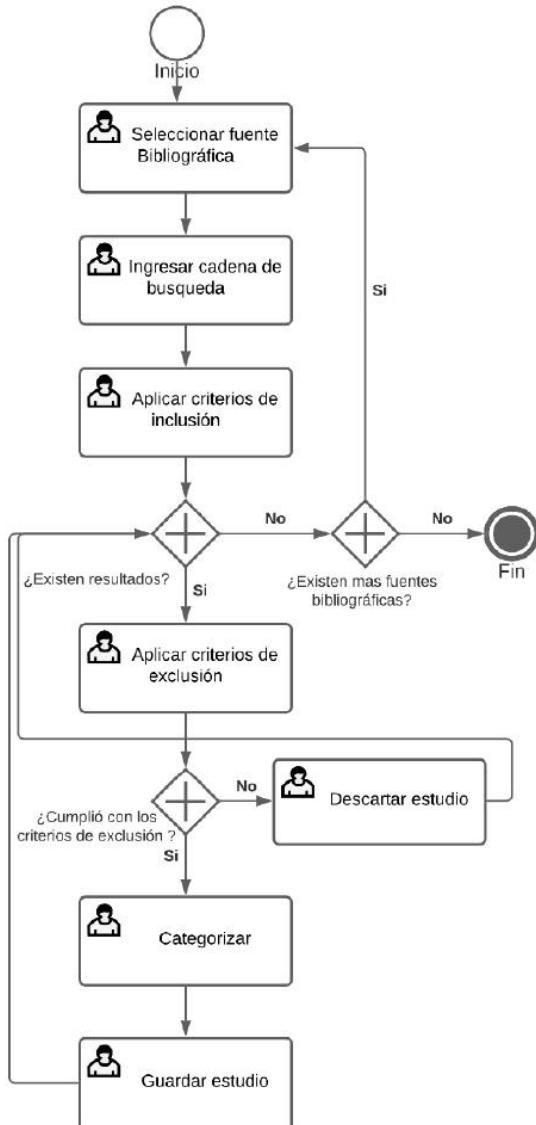


Fig. 1: Proceso de selección de artículos

artículos, los cuales poseían la información necesaria para la realización de la RSL, estos se detallan en la TABLA 5.

Síntesis de los datos

Cada artículo fue analizado para identificar su aporte más relevante y así finalmente obtener información relacionada a los modelos BERT y SQuAD requeridos que contestan a las preguntas de investigación inicialmente planteadas (ver TABLA 2).

¿Cuáles son las características principales sobre el funcionamiento del modelo BERT y SQuAD?

De acuerdo a los estudios (Liu et al., 2019) (Zhou et al., 2020) (Yang et al., 2020) (L. Su et al., 2019b) (Gao et al., 2019) (El-Geish, 2020) (Chang et al., 2020) (Zeng et al., 2020) (Vinod et al., 2020) las características principales sobre el funcionamiento del modelo BERT y SQuAD son:

Para BERT:

Se puede desarrollar en los Frameworks de Tensorflow y PyTorch.

Dependiendo del framework usado se puede programar en los lenguajes Python (PyTorch y Tensorflow) o C++ (Tensorflow).

Consta de un entrenamiento previo y ajuste fino, donde, se siguen dos pasos para aplicar BERT que incluyen preentrenamiento de un modelo BERT utilizando grandes corpus sin etiquetar y afinando el modelo pre entrenado utilizando corpus anotados específicos de la tarea.

Se recomienda ajustar la versión más pequeña, BERT Base, en una tarjeta gráfica con 12 GB de RAM. Donde el entrenamiento toma aproximadamente 2 días y el modelo base reporta un f1 de 65.28 % en el conjunto de datos SQuAD 2.0

El entrenamiento de BERT Large en 16 Cloud TPU toma 4 días.

Para el refinamiento de BERT Large empleando una GPU RTX 2080 Ti es de 1 día aproximadamente.

BERT Large, en el ajuste fino es inestable en conjuntos de datos pequeños. Pero BERT Large logra un aprovechamiento de mayor precisión de tres veces y media más precisión más parámetros.

En base a los artículos obtenidos se ha podido constatar que la mayoría de proyectos emplean como lenguaje Python junto a PyTorch (Bruke Mammo, Praveer Narwelkar, 2018) por su vinculación y puntos fuertes en el desarrollo de este tipo de proyectos (Zeng et al., 2020). Además, emplean BERT Base por los recursos limitados de la GPU y el tiempo que se toma en entrenar al modelo es más corto y la precisión no varía demasiado en comparación con BERT Large (Liu et al., 2019).

¿Cuáles son los parámetros que se emplean en los modelos de BERT con SQuAD?

En base a los artículos seleccionados y estudiados, se constató que los siguientes parámetros son los más empleados como se mencionan en (Liu et al., 2019) (Zhou et al., 2020) (Devlin et al., 2019a) (Maghraoui et al., 2021) (Zadeh et al., 2020)(Bruke Mammo, Praveer Narwelkar, 2018) (Yang et al., 2020) (Özçift et al., 2021) (Chintalapudi et al., 2021) (Balagopalan et al., 2021) (Hulburd, 2020) para trabajos en desarrollo.

BERT Base emplea los siguientes parámetros previamente entrenados que son los siguientes: son $L = 12$, $H = 768$, $A = 12$, Parámetros totales = 110M, donde L es el número de capas (es decir, bloques de Transformers), H es el tamaño oculto y A es el número de capas de atención propia. El tamaño del filtro de alimentación hacia adelante se establece en $4H$, es decir, 3072 para $H = 768$.

BERT toma una secuencia de tokens con una longitud máxima de 512 y produce una representación de la secuencia en un vector de 768 dimensiones, según la documentación base de BERT. Donde se prefiere un tamaño de lote más grande para obtener una estimación suficientemente estable de cuál sería el gradiente del conjunto de datos completo. Donde, el tamaño del lote siempre se establece en potencia

Tabla 5: Estudios seleccionados

Nº.	Estudios seleccionados	Ref.	Fuente
ES01	Combat COVID-19 infodemic using explainable natural language processing models	(Ayoub et al., 2021)	Scopus
ES02	Transfer Learning from BERT to Support Insertion of NewConcepts into SNOMED CT	(Liu et al., 2019)	Scopus
ES03	Deep Learning Based Fusion Approach for Hate Speech Detection	(Zhou et al., 2020)	Scopus
ES04	A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization	(M. H. Su et al., 2020)	IEEE Digital Library
ES05	BERT: Pre-training of deep bidirectional transformers for language understanding	(Devlin et al., 2019b)	Google Scholar
ES06	Performance Analysis of Deep Learning Workloads on a Composable System	(Maghraoui et al., 2021)	Google Scholar
ES07	GOBO: Quantizing attention-based nlp models for low latency and energy efficient inference	(Zadeh et al., 2020)	Google Scholar
ES08	Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning	(Bruke Mammo, Praveer Narwelkar, 2018)	Google Scholar
ES09	Extracting family history of patients from clinical narratives: Exploring an end-to-end solution with deep learning models	(Yang et al., 2020)	Scopus
ES10	Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish	(Özçift et al., 2021)	Scopus
ES11	Sentiment analysis of COVID-19 tweets using deep learning models	(Chintalapudi et al., 2021)	Scopus
ES12	Comparing Pre-trained and Feature-Based Models for Prediction of Alzheimer's Disease Based on Speech	(Balagopalan et al., 2021)	Scopus
ES13	Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0	(Hulburg, 2020)	Google Scholar
ES14	Target-dependent sentiment classification with BERT	(Gao et al., 2019)	Scopus
ES15	Gestalt: a Stacking Ensemble for SQuAD2.0	(El-Geish, 2020)	Google Scholar
ES16	Controlling Risk of Web Question Answering	(L. Su et al., 2019a)	Google Scholar
ES17	Know What You Don't Know: Unanswerable Questions for SQuAD	(Rajpurkar et al., 2018)	Google Scholar
ES18	SQuAD: 100,000+ Questions for Machine Comprehension of Text	(Rajpurkar et al., 2016)	Google Scholar
ES19	Generating contextual embeddings for emergency department chief complaints	(Chang et al., 2020)	Scopus
ES20	An ensemble learning strategy for eligibility criteria text classification for clinical trial recruitment: Algorithm development and validation	(Zeng et al., 2020)	Scopus
ES21	Fine-tuning the BERTSUMEXT model for Clinical Report Summarization	(Vinod et al., 2020)	IEEE Digital Library

de dos, por ejemplo, 512, 1024, 2048.

Teniendo en cuenta la memoria de la GPU, si se elige BERT Base con el modelo uncase para un ajuste fino se puede evitar el desbordamiento de memoria de la GPU. De acuerdo con la sugerencia oficial de BERT, establece una la longitud máxima de la oración de 64, mini-lote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3,0 para el entrenamiento del modelo.

En BERT, los datos se pasaron a tres capas de incrustación que incluyen una incrustación de token, segmento y posición capas. En el primer paso del procesamiento, las oraciones se tokenizaron y después de eso, cada token de entrada se pasó a través de una capa de incrustación de tokens para transformarlo en una representación vectorial de dimensión fija (es decir, vector de 768 dimensiones). Además, se agregaron tokens de clasificación adicional [CLS] y separador [SEP] al inicio y al final de la oración tokenizada para servir como una representación de entrada y un separador de oración para la tarea de clasificación. La capa de incrustación de segmentos ayuda a clasificar un texto dado un par de textos de entrada.

Cabe afirmar que cada capa cumple una función de filtrado para lograr encontrar patrones en los datos que contienen datos invisibles a medida que pasa por cada capa, la cual se basa en los patrones de la capa anterior, todo esto empleando los token creados en cada palabra al inicio de la primera capa en la creación de los tokens.

Emplean la biblioteca PyTorch-Pretrained-BERT para construir el modelo BERT. Luego, ajustan su capa lineal (mediante la adición de nuevas palabras junto a la creación de sus respectivos tokens, mediante codificación de los parámetros mencionados en el anterior punto) y la activación sigmoidea (no importa cuál sea la entrada, la salida obtenida está entre valores de 0 y 1) para obtener las predicciones con el conjunto de datos etiquetado COVID-19. Durante el proceso de ajuste fino, se usaron el optimizador Adam con una tasa de aprendizaje de 3×10^{-6} y un tamaño de lote de 12. Y se ajustó el modelo en el conjunto de datos COVID-19 recopilado durante tres épocas, basado en BERT Base mencionado en el primer punto.

Dichos parámetros se consideraron como factibles para emplearlos en el entrenamiento del modelo BERT. De los estudios que contribuyeron a dar luz a la pregunta de investigación, fue el de Yanling Zhou (Zhou et al., 2020), ya que evalúa el rendimiento de los diferentes modelos como el de EIMo y BERT entre otros, donde el que sigue destacando por encima de los demás es BERT con una precisión del 70% y en el parámetro F1 del 62%, ofrece un margen considerable para poder mejorar con entrenamientos adicionales en modelos pre entrenados y enfocados a un tema en particular, (Ayoub et al., 2021) en el conjunto de (Bruke Mammo, Praveer Narwelkar, 2018) datos SQuAD 2.0. Cabe destacar que para el ajuste de cada modelo se basa en diferentes parámetros según el propósito que se tenga y donde se debe hacer diferentes ajustes para llegar a una configuración optima mediante los entrenamientos.

¿Qué versión de los modelos BERT y SQuAD es factible para el desarrollo del agente virtual?

De acuerdo a los estudios (Liu et al., 2019) (Zhou et al., 2020)(M. H. Su et al., 2020) (Devlin et al., 2019a) (Maghraoui et al., 2021) (Zadeh et al., 2020) (Bruke Mammo, Praveer Narwelkar, 2018)(Yang et al., 2020) (Özçift et al., 2021) (Chintalapudi et al., 2021) (Balagopalan et al., 2021) las principales versiones existentes de BERT son:

BERT Base: Se compone de 12 bloques transformers, 12 capas de atención y 110 millones de parámetros, el cual es más empleado con recursos limitados de hardware.

BERT Large: Se compone de 24 capas, 16 capas de atención y 340 millones de parámetros, es más empleado en grandes equipos de hardware con alta capacidad computacional. Lo que dificulta su aplicación en tiempo real sin aceleración hardware, como GPU y TPU.

BERT Largue emplea una cantidad de almacenamiento de 1,12 GB de registros, frente a BERT Base que sacrificando precisión en los resultados alcanza los 326 MB destinados en registros. [9]Emplea parámetros de punto flotante 32b.

En cambio, para SQuAD según (Devlin et al., 2019a) (L. Su et al., 2019b) (Rajpurkar et al., 2018) (Rajpurkar et al., 2016) menciona que:

SQuAD 1.0 se trata de un conjunto de datos que contiene respuestas a preguntas que consta de más de 100.000 pares de preguntas, junto al párrafo relevante correspondiente que contiene la respuesta. Pero no es factible ya que para preguntas sin respuesta no tiene definido una respuesta lo que da lugar a malas interpretaciones sobre las respuestas.

SQuAD 2.0: Permite entrenar modelos para que se abstengan de responder preguntas sin respuesta, la tarea de SQuAD 2.0 amplía la definición del problema de SQuAD 1.1 al permitir la posibilidad de que no exista una respuesta corta en el párrafo proporcionado, lo que hace que el problema sea más realista. Por lo que es más factible emplear SQuAD 2.0, ya que, como se menciona es más realista por su funcionalidad antes preguntas que no tienen respuesta.

Dichas versiones tales como BERT Base y BERT Large (Ayoub et al., 2021) tienen diferentes contrastes dependiendo básicamente del hardware con el que se cuente para el desarrollo de los modelos. Cabe mencionar que, en los artículos encontrados la mayoría da por hecho el uso de modelos pre entrenados, de los cuales solo se va mejorando a través de nuevos entrenamientos y parámetros. Por lo tanto, la versión de BERT base es más factible, ya que no demanda un hardware con altas prestaciones.

SQuAD posee varias versiones, pero las más usadas son SQuAD 1.0 (Rajpurkar et al., 2016) y SQuAD 2.0 (El-Geish, 2020), las que se diferencian básicamente en las preguntas sin respuesta que ofrece la última versión SQuAD 2.0 y se asemeja a la realidad por su función. Por lo que para que sea

más fiel a la realidad, es más factible SQuAD 2.0, ya que combina el conjunto de datos de SQuAD 1.0 con preguntas sin respuesta.

DISCUSIÓN

Esta RSL tuvo como propósito investigar acerca de los principales parámetros y configuraciones de los modelos BERT y SQuAD para el desarrollo de agentes conversacionales, así como de las diferentes versiones existentes de estos modelos.

De los resultados obtenidos en la mayoría de artículos se mencionan el empleo del lenguaje Python junto a PyTorch (Bruke Mammo, Praveer Narwelkar, 2018) (Zeng et al., 2020), por su vinculación y puntos fuertes en el desarrollo de este tipo de proyectos. Además, emplean BERT Base por los recursos limitados de la GPU y el tiempo que se toma en entrenar al modelo (Liu et al., 2019).

En relación a los parámetros y configuraciones relacionados con el modelo BERT teniendo en cuenta los escasos recursos del hardware los parámetros más factibles serían el uso de tres capas de incrustación junto al modelo BERT base uncase, ya que debe procesar menos datos y establece una la longitud máxima de la oración de 64, mini-lote de 32, una tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3,0 para el entrenamiento del modelo (Zhou et al., 2020).

Por lo cual sería la configuración idónea para empezar a entrenar un modelo junto a BERT y SQuAD 2.0 para un equipo de hardware no tan robusto.

Se han identificado algunas versiones de BERT, tales como BERT Base y BERT Large (Ayoub et al., 2021) las cuales tienen diferentes contrastes dependiendo del hardware con el que se cuente para el desarrollo de los modelos. En la mayoría de artículos encontrados se da por hecho el uso de modelos pre entrenados, de los cuales solo se va mejorando a través del nuevos entrenamientos y parámetros. La versión que se ha encontrado como más factible ya que no demanda un hardware con altas prestaciones es BERT Base.

Por parte de SQuAD se han encontrado varias versiones, pero las más usadas son SQuAD 1.0 y SQuAD 2.0 (Rajpurkar et al., 2016) (El-Geish, 2020), las que se diferencian básicamente en las preguntas sin respuesta que ofrece la última versión y se asemeja a la realidad por su función.

CONCLUSIONES

Como características principales sobre el funcionamiento del modelo BERT y SQuAD, se concluyó que se emplea del lenguaje Python junto a framework PyTorch para desarrollar en la nube o con la ayuda de una GPU, empleando además el modelo BERT Base, para tener un mejor desempeño si se posee hardware con recursos limitados, destacando que, se debe realizar un entrenamiento previo y ajuste fino posterior del modelo para su optimización.

Los parámetros más destacados para proyectos relaciona-

dos con BERT y que no supongan un procesamiento extremadamente grande teniendo en cuenta los escasos recursos del hardware son el uso de tres capas de incrustación junto al modelo BERT base uncase, ya que debe procesar menos datos y establece una la longitud máxima de la oración de 64, mini-lote de 32, una la tasa de aprendizaje de 2e-5 y un número de épocas de entrenamiento de 3,0 para el entrenamiento del modelo.

Para la elaboración del agente conversacional basado en BERT y SQuAD se concluyó que se usará el modelo de BERT base junto a SQuAD 2.0, con una GPU Nvidea para entrenar de manera física o en la nube empleando Google Colab, ya que también cuenta con tarjetas gráficas Nvidea. Y los parámetros de acuerdo con la sugerencia oficial de los creadores de BERT, que establecen una la longitud máxima de la oración de 64, un mini-lote de 32, una la tasa de aprendizaje de 2e-5, con un número de épocas de entrenamiento de 3 para el entrenamiento del modelo.

AGRADECIMIENTOS

A la Universidad Nacional de Loja y todos sus docentes que nos formaron tanto intelectualmente y profesionalmente.

CONTRIBUCIONES DE LOS AUTORES

Conceptualización: JC y VS; metodología: JC y VS; análisis formal: JC y VS.; investigación: JC y VS; recursos: JC y VS; curación de datos: JC y VS; redacción — preparación del borrador original: JC y VS; redacción — revisión y edición: JC y VS; visualización: JC y VS; supervisión: JC y VS; administración de proyecto: JC y VS; adquisición de financiamiento para la investigación: JC y VS. Todos los autores han leído y aceptado la versión publicada del manuscrito. José Carrión: JC. Victor Serrano: VS.

FINANCIAMIENTO

El presente estudio tuvo un financiamiento de procedencia propia por parte de los integrantes.

REFERENCIAS

Ayoub, J., Yang, X. J., Zhou, F. (2021). Combat COVID-19 infodemic using explainable natural language processing models. *Information Processing and Management*, 58(4). <https://doi.org/10.1016/j.ipm.2021.102569>

Balagopalan, A., Eyre, B., Robin, J., Rudzicz, F., Novikova, J. (2021). Comparing Pre-trained and Feature-Based Models for Prediction of Alzheimer's Disease Based on Speech. *Frontiers in Aging Neuroscience*, 13. <https://doi.org/10.3389/fnagi.2021.635945>

Bruke Mammo, Praveer Narwelkar, R. G. (2018). Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning, 1–25.

Chang, D., Hong, W. S., Taylor, R. A. (2020). Generating contextual embeddings for emergency department chief complaints. *JAMIA Open*, 3(2), 160–166.

- <https://doi.org/10.1093/jamiaopen/ooaa022>
- Chintalapudi, N., Battineni, G., Amenta, F. (2021). Sentimental analysis of COVID-19 tweets using deep learning models. *Infectious Disease Reports*, 13(2). <https://doi.org/10.3390/IDR13020032>
- Devlin, J., Chang, M. W., Lee, K., Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, 4171–4186. <https://github.com/tensorflow/tensor2tensor>
- El-Geish, M. (2020). Gestalt: a Stacking Ensemble for SQuAD2.0. <http://arxiv.org/abs/2004.07067>
- Gao, Z., Feng, A., Song, X., Wu, X. (2019). Target-dependent sentiment classification with BERT. *IEEE Access*, 7, 154290–154299. <https://doi.org/10.1109/ACCESS.2019.2946594>
- Hulburd, E. (2020). Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0. <http://arxiv.org/abs/2002.10670>
- Kitchenham, B., Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering.
- Liu, H., Perl, Y., Geller, J. (2019). Transfer Learning from BERT to Support Insertion of New Concepts into SNOMED CT. *AMIA ... Annual Symposium Proceedings. AMIA Symposium*, 2019, 1129–1138.
- Maghraoui, K. El, Herger, L. M., Choudary, C., Tran, K., Deshane, T., Hanson, D. (2021). Performance Analysis of Deep Learning Workloads on a Composable System. 1, 1–10. <http://arxiv.org/abs/2103.10911>
- Özçift, A., Akarsu, K., Yumuk, F., Söylemez, C. (2021). Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. *Automatika*. <https://doi.org/10.1080/00051144.2021.1922150>
- Petticrew, M., Roberts, H. (2008). Systematic Reviews in the Social Sciences: A Practical Guide. In *Systematic Reviews in the Social Sciences: A Practical Guide*. Blackwell Publishing Ltd. <https://doi.org/10.1002/9780470754887>
- Rajpurkar, P., Jia, R., Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *ArXiv Preprint ArXiv:1806.03822*.
- Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2383–2392. <https://doi.org/10.18653/v1/d16-1264>
- Su, L., Guo, J., Fan, Y., Lan, Y., Cheng, X. Controlling Risk of Web Question Answering. *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 115–124. <https://doi.org/10.1145/3331184.3331261>
- Su, M. H., Wu, C. H., Cheng, H. T. (2020). A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28, 2061–2072. <https://doi.org/10.1109/TASLP.2020.3006731>
- Vinod, P., Safar, S., Mathew, D., Venugopal, P., Joly, L. M., George, J. (2020, June 1). Fine-tuning the BERTSUMEXT model for clinical report summarization. *2020 International Conference for Emerging Technology, INCET 2020*. <https://doi.org/10.1109/INCET49848.2020.9154087>
- Yang, X., Zhang, H., He, X., Bian, J., Wu, Y. (2020). Extracting family history of patients from clinical narratives: Exploring an end-to-end solution with deep learning models. *JMIR Medical Informatics*, 8(12). <https://doi.org/10.2196/22982>
- Zadeh, A. H., Edo, I., Awad, O. M., Moshovos, A. (2020). GOBO: Quantizing attention-based nlp models for low latency and energy efficient inference. *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, 2020-Octob, 811–824. <https://doi.org/10.1109/MICRO50266.2020.00071>
- Zeng, K., Pan, Z., Xu, Y., Qu, Y. (2020). An ensemble learning strategy for eligibility criteria text classification for clinical trial recruitment: Algorithm development and validation. *JMIR Medical Informatics*, 8(7). <https://doi.org/10.2196/17832>
- Zhou, Y., Yang, Y., Liu, H., Liu, X., Savage, N. (2020). Deep Learning Based Fusion Approach for Hate Speech Detection. *IEEE Access*, 8, 128923–128929. <https://doi.org/10.1109/ACCESS.2020.3009244>

Anexo 2: Revisión Sistemática de Literatura II

A continuación, se adjunta el producto final de la Revisión Sistemática de Literatura indicado en el objetivo 2, presentado como artículo.

Revisión Sistemática de Literatura: Artículos destacados respecto a temas tópicos sobre el Covid-19.

**José Alexis CarrIÓN Ojeda, Victor Yamil Serrano Zari,
Cumbicus Pineda Oscar Miguel**

Carrera de Ingeniería en Sistemas
Universidad Nacional de Loja

jose.a.carrion.o@unl.edu.ec, victor.serrano@unl.edu.ec,
oscar.cumbicus@unl.edu.ec

Abstract – Currently, the population uses social networks and search engines that index information to facilitate the search for any information on the web.

The present literature review aims to identify the articles that talk about the most common topics about covid-19 in the current pandemic to find what are the questions that most interest the population about the coronavirus to subsequently use this information in different automated systems that provide reliable and verifiable information. Since searches thanks to Big Data can be classified and measured to determine what people are looking for. The present RSL was based on the phases of Barbara Kitchenham's methodology, the review was based on three research questions and defined the course of the review; obtaining that the central theme with which the searches coincide is coronavirus, along with keywords such as symptoms and protective measures.

Resumen – En la actualidad la población para informarse recurre a redes sociales y buscadores que indexan la información para que sea más fácil buscar cualquier información en la web.

La revisión de literatura presente tiene como fin identificar artículos que hablen sobre los temas más comunes sobre el covid-19 en la actual pandemia con el fin de hallar cuales son las preguntas que más le interesan a la gente sobre el coronavirus para emplear posteriormente esta información en diferentes sistemas automatizados que brinden una información confiable y verificable. Ya que las búsquedas gracias al Big Data se pueden clasificar y medir para determinar qué es lo que la gente busca. La presente RSL se basó en las fases de la metodología de Bárbara Kitchenham, la revisión se planteó en base a tres preguntas de investigación y definió el transcurso de la revisión; obteniendo que el tema centrar con que coinciden las búsquedas es coronavirus, junto a las palabras claves como síntomas y medidas de protección.

Keywords – Covid-19, Coronavirus, Information, search statistics.

I. INTRODUCCIÓN

Actualmente la mayoría de la población mundial a emplea buscadores para obtener información sobre diferentes temas de interés y mediante una empresa mundialmente conocida como Google y su buscador, facilita las consultas en la web gracias a su accesibilidad, además, ofrece diferentes herramientas para analizar los contenidos buscados y obtener diferentes datos segmentados. En la presente RSL, mediante la metodología de Bárbara

Kitchenham y empleando sus fases se ha podido conseguir artículos sobre estudios y trabajos relacionados que permitieron identificar la información y temas relacionados con temas tópicos sobre el covid-19. Ya que es de interés saber qué es lo que la población busca frecuentemente sobre el covid-19.

La presente revisión de literatura está dividida en secciones como: el Alcance, para determinar el propósito que tiene realizar la presente RSL, a continuación, en la sección de Metodología se definieron las fases a seguir que propone Barbara Kitchenham para realizar revisiones de literatura, posteriormente, se realiza el proceso para cada una de las fases propuestas detallando el proceso y las salidas en la sección de Resultados y finalmente, se plantean las Conclusiones obtenidas durante la realización de la presente RSL.

II. ALCANCE

La Revisión Sistemática de Literatura (RSL) tiene como intención analizar los artículos que contengan temas sobre el covid-19 comunes, en lo que se refiere a temas que fueron más buscados para obtener información relevante sobre lo que a la población le interesa saber acerca de lo que abarca la pandemia.

III. METODOLOGÍA

La realización de la revisión sistemática de literatura se basa en el proceso de la metodología de Bárbara Kitchenham [1], dicho proceso se resume en tres fases principales:

- Planificación de la revisión.
- Realizar la revisión.
- Revisión de informes.

De acuerdo a Kitchenham[1], algunas tareas de estas fases no son obligatorias, como por ejemplo:

- Puesta en marcha de una revisión, depende de la revisión sistemática que se está haciendo sobre una base comercial.
- Evaluar el protocolo de revisión y la evaluación del informe, son opcionales y dependen de los procedimientos de garantía de calidad decididas por el encargado de la revisión sistemática.

Para la presente RSL no se ejecutaron todos los pasos

propuestos por Kitchenham, justamente porque su metodología es flexible en cuanto a la extensión que el investigador necesite dar a su revisión y depende del investigador el alcance que requiera dar a la presentación y publicación de sus resultados. De acuerdo a esto, en la TABLA XVI, se presentan las fases y tareas que fueron consideradas:

TABLA XVI.
Proceso de revisión sistemática de literatura propuesto

Fases	Tareas
Planificación de la revisión	Identificación de la necesidad de una revisión.
	Especificación de las preguntas de investigación.
	El desarrollo de un protocolo de revisión.
Realizar la revisión	Identificación de la investigación.
	Selección de los estudios primarios.
	Síntesis de los datos
Revisión de informes	Dar formato al informe principal.

Las fases y tareas mencionadas anteriormente han sido realizadas para cumplir con la meta planeada en el alcance.

IV. RESULTADOS

A. Planificación de la revisión

J. Identificación de la necesidad de una revisión

Mediante la ejecución de una revisión sistemática de literatura, se puede encontrar el conocimiento ya existente acerca de un tema en particular, en este caso es necesaria para determinar los temas y preguntas frecuentes que las personas realizan relacionado con el covid-19, para de esta manera elegir artículos relacionados a estos temas para la posterior creación de un agente virtual.

K. Preguntas de investigación

En la TABLA XVII, se presentan las preguntas de investigación que guiaron el desarrollo de la revisión sistemática de literatura, estas preguntas son:

TABLA XVII.
Preguntas de investigación

Preguntas de investigación	
P1	¿Cuáles son los temas más abordados relacionados con el covid-19?
P2	¿Qué preguntas o palabras clave son las más comunes acerca del covid-19?

L. El desarrollo de un protocolo de revisión

a. Diseño del protocolo de búsqueda

i. Estrategias de búsqueda

Petticrew y Roberts [2], recomiendan criterios para plantear la investigación, mediante el uso de la nemotécnica PICOC (Población, Intervención, Comparación, Resultado y Contexto), se estructuran los cinco componentes para definir la cadena de búsqueda; así como también, para una mejor organización y selección de todos los resultados, se utilizó la herramienta en línea Parsifal¹, la cual ayuda en el contexto de la Ingeniería del Software para realizar revisiones sistemáticas de literatura, gracias a esta herramienta que permite dar seguimiento a la RSL, la cual está diseñada para abarcar todos los campos necesarios y obtener resultados óptimos.

ii. Fuentes bibliográficas

Como fuentes bibliográficas, se ha seleccionado algunas bibliotecas virtuales, como son:

- IEEE Digital Library (<https://ieeexplore.ieee.org/>)
- Google Scholar (<https://scholar.google.com/>)
- Scopus (<http://www.scopus.com>)

iii. Definir palabras claves para el problema de estudio

Con la definición de los criterios PICOC se obtuvo un conjunto de palabras claves, las mismas que permitieron construir las cadenas de búsquedas, estas son:

- Coronavirus
- covid-19
- Information
- Search
- search statistics
- statistics

iv. Cadenas de búsqueda

En la TABLA VII se presentan las cadenas de búsqueda aplicadas que se realizaron de acuerdo a cada biblioteca virtual: IEEE Digital Library, Google Scholar, y Scopus.

v. Criterios de inclusión

Durante la búsqueda, se consideraron los siguientes criterios de inclusión:

¹ "Parsifal." <https://parsif.al/>.

- Solo documentos que sean artículos científicos.
- Se mencione Covid-19.
- Sobre cuestiones relacionadas con el covid-19.
- Idioma: inglés-español
- Publicaciones superiores a los últimos 3 años

vi. Criterios de exclusión

Se consideraron los siguientes criterios de exclusión:

- Documentos que no aporten información relevante sobre datos de búsquedas del covid-19.
- Artículos en idiomas que no sean inglés o español.
- Publicaciones anteriores al año 2020.
- Que no mencione al covid-19 como tema.

B. Realizar la revisión

a. Identificación de la investigación

El objetivo de la presente revisión sistemática de literatura es dar respuesta a las preguntas de investigación planteadas, mediante la búsqueda de estudios primarios que aporten información verídica con respecto al tema.

b. Selección de los estudios primarios

Para realizar la selección de los estudios se ha seguido el proceso que a continuación se muestra en la Figura 55.

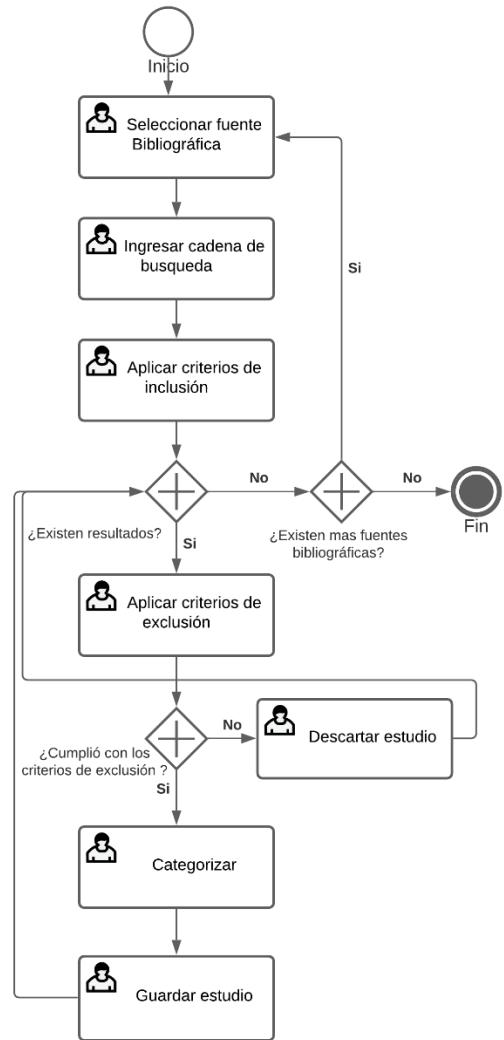


Figura 55. Proceso de selección de artículos

En la TABLA XVIII, se presenta un resumen de los trabajos o estudios relacionados que fueron encontrados junto con el número de estudios seleccionados según la fuente bibliográfica.

TABLA XVIII.
Resumen de estudios seleccionados

Fuente	Encontrados	Seleccionados
IEEE Digital Library	1	1
Google Scholar	12	6
Scopus	21	8
Total	34	15

Se obtuvieron 16 estudios relacionados, los mismos que luego de pasar por la evaluación de calidad, resultaron en 12 artículos, los cuales poseían la información necesaria para la realización de la RSL, estos se detallan en la TABLA VII del Anexo 1.

c. Síntesis de los datos

Cada artículo fue analizado para identificar su aporte más relevante y así finalmente obtener información

relacionada los temas y preguntas más frecuentes acerca del covid-19, que contestan a las preguntas de investigación inicialmente planteadas (ver TABLA XX II).

a) ¿Cuáles son los temas más abordados relacionados con el covid-19?

El tema principal según [3] fue “coronavirus” en cuanto a temas para luchar contra el coronavirus mediante medidas preventivas, el término más buscado fue “quedarse en casa”, seguido de “máscaras faciales”, “distanciamiento social” y “lavarse las manos”.

En [4] se utiliza la herramienta Google Trends para establecer los temas informativos más consultados asociados al CoViD-19. Las palabras “coronavirus Ecuador” y “Covid Ecuador” son las más utilizadas para indagar información en torno a la pandemia. Por otro lado, tratamientos con el uso de sustancias como la “hidroxicloroquina” y el “dióxido de cloro”, presentan búsquedas medianamente importantes. Además, las búsquedas mediante las palabras “vacuna coronavirus” o “Covid vacuna” así como “coronavirus síntomas” presentan diversos rangos de importancia.

En [5] se evidenció que a medida que los casos por COVID-19 se reportaban en Wuhan y Latinoamérica, el nivel de búsqueda en Google Trends para los términos “coronavirus”, “lavado de manos” y “mascarillas” aumentaban.

En [6] con ayuda de Google Trends la cual proporciona el número de búsquedas relativas dentro de una determinada región o globalmente para una consulta en particular, Los datos recopilados de Google Trends se ajustan a la hora y la ubicación, por lo que las comparaciones entre consultas pueden ser más fáciles. Los principales términos de búsqueda entre países de habla inglesa que se estudiaron fueron “2019-nCoV + SARS-CoV-2 + nuevo coronavirus + nuevo coronavirus + COVID-19 + corona virus de Enfermedades 2019”.

En [7] demuestra un fuerte aumento en el interés de las búsquedas en Internet sobre “fiebre”, “tos” y “diarrea” al inicio de la pandemia de COVID-19 en los Estados Unidos antes de la identificación de casos, mientras que la búsqueda relativa de muchos síntomas, que incluyen “náuseas”, “vómitos”, “dolor abdominal” y “estreñimiento”, permanecieron cerca o por debajo de la línea de base durante el período de tiempo estudiado.

En el artículo [8] se menciona que las palabras clave en las consultas son: “casos de coronavirus”, “covid19” y “casos de covid19”. El coronavirus se relacionó principalmente con “casos”, “reino unido”, “síntomas”, “noticias” y “actualización”. El término “covid” estaba relacionado con el número “19”. Además, la palabra clave (casos de covid19) se produjo más en los términos de esta palabra clave compuesta con sus términos (covid19). El número entero “19” se correlacionó con el término “casos”.

En [12], los temas de Google Trends relacionados con el “coronavirus”, el “lavado de manos” y las “mascarillas” representaban las principales búsquedas de información sobre COVID-19, así como la práctica de la higiene personal para prevenir

la transmisión de enfermedades.

b) ¿Qué preguntas o palabras clave son las más comunes acerca del covid-19?

En el artículo [3] menciona las palabras clave en base a los síntomas más buscados, donde fueron “fiebre”, “dolor de garganta” y “tos”, además, el interés de los usuarios por conocer las vías de transmisión del síndrome respiratorio agudo coronavirus 2, que genera hasta en la actualidad debates.

En [10] las principales preguntas abordadas en torno al tema del covid-19 son acerca de las pruebas de diagnóstico de esta enfermedad, entra las que se destacan: ¿Qué tan precisas son las pruebas de RT-PCR?, ¿Qué tipos de muestras son adecuadas para realizar pruebas?, ¿Una muestra negativa descarta COVID-19?, ¿Vale la pena repetir una prueba después de que un paciente haya dado negativo?, ¿Cuánto tiempo permanecerán positivos los pacientes infectados?, ¿Una muestra positiva significa que un paciente es infeccioso?, ¿Afecta la edad a la carga viral?, ¿Qué tan pronto se desarrollan los anticuerpos?.

En el artículo [11] se mencionan y se dan respuestas a algunas de las preguntas más frecuentes relacionadas con el covid-19 entre las cuales tenemos: ¿Qué es COVID 19?, ¿De dónde provienen los coronavirus?, ¿Cuáles son los modos de transmisión del SARS Cov-2?, . ¿Cuáles son los síntomas comunes de COVID-19?, ¿Cómo puedo protegerme de la infección por COVID-19?, ¿Qué es el distanciamiento social / físico? ¿Cómo previene la propagación de COVID-19?, ¿Qué hacer si tuve un contacto con un paciente con COVID-19?, entre otras comunes.

En el artículo [13] se responden a preguntas frecuentes acerca del covid-19, relacionadas con información de la pandemia, métodos de prevención del covid, tratamientos, síntomas entre otros. Entre estas preguntas se destacan: ¿Qué es una pandemia?, ¿Qué es el distanciamiento social?, ¿Existe alguna vacuna, medicamento o tratamiento para COVID-19?, ¿Cuál es la tasa de letalidad? ¿Es lo mismo que la tasa de mortalidad por infecciones?, ¿El COVID-19 afecta a las personas mayores o las personas más jóvenes también son susceptibles?

En [7] los temas de búsqueda más destacados fueron los relacionados con síntomas gastrointestinales asociados con la infección por COVID-19 en la literatura, como “diarrea”, “náuseas”, “vómitos” y “dolor abdominal”. Los términos “fiebre” y “tos” se incluyeron como destacados. Los términos “anosmia”, “disgeusia”, “pérdida del apetito”, “pérdida del gusto” y “pérdida del olfato” fueron considerados; sin embargo, existió una baja frecuencia de búsquedas de estos términos.

En el artículo [8] la categoría más popular de búsqueda fue “Virus” la mayoría de los días de este

período. La segunda categoría de búsqueda importante fue "Agente infeccioso". La categoría "Enfermedad" se presentó más a finales de febrero hasta finales de la segunda semana de marzo. Aunque las categorías de Asia ("País de Asia Oriental" y "País de Asia") fueron tendencias después del 11 de febrero, la categoría de Asia Oriental también fue frecuente a partir de la cuarta semana de enero.

En [9] mencionan un total de 15 de los términos de búsqueda de síntomas tenían correlación estadísticamente significativa con casos confirmados de COVID-19. De los términos incluidos, solo la "diarrea" no logró una correlación estadísticamente significativa con las muertes por COVID-19. Los síntomas de "disnea", "anosmia", "ageusia", "dolor de cabeza", "dolor en el pecho" y "estornudos" tuvieron fuertes correlaciones tanto con los casos nuevos como con las muertes.

En [9] la lista de terminología de síntomas considerada fue: "fiebre", "dificultad para respirar", "tos", "anosmia", "fatiga", "rinorrea", "congestión nasal", "estornudo", "mialgia", "dolor de garganta", "diarrea", "anorexia", "dolor de pecho", "esputo", "dolor de cabeza", "náuseas", "ageusia", "dolor abdominal", "mareos", "vómitos" y "dolor ocular".

A. Revisión de informes

a. Dar formato al informe principal

Esta tarea hace referencia a la creación del informe con el proceso realizado para la obtención de resultados, es decir, el presente trabajo de RSL es el informe a considerar.

V. CONCLUSIONES

De acuerdo a la revisión realizada sobre artículos relacionados con temas tópicos del covid-19, se han obtenido las siguientes conclusiones:

- En base a la RSL y los trabajos obtenidos, para los temas más abordados relacionados con el covid-19, se puede apreciar que los autores mencionan como tema principal la palabra coronavirus o covid-19, es el tema central de las búsquedas en la web y le preceden las búsquedas sobre medicación para el tratamiento, junto con búsquedas sobre vacunas y los síntomas principales que se presentan con este virus como fiebre, tos, dolor abdominal que estuvieron con alta demandas de consultas. También se apreció que a medida que aumentaban los casos en China las búsquedas sobre medidas de prevención se intensificaban, lo que ofrece una oportunidad de poder crear un sistema que ofrezca esta información de una manera asociada, confiable y precisa, además de la que ofrece la herramienta Google Trends.
- Las palabras clave más empleadas en búsquedas relacionadas con la sintomatología del covid-19

fueron "fiebre, dolor de garganta, dificultad para respirar, náuseas, dolor ocular, fatiga y tos" junto al interés de saber cuáles son las vías de transmisión, por lo que se podría ayudar a la población a dar un diagnóstico más fiable en base a las consultas que más generan incertidumbre sin la necesidad de recurrir a un especialista en primera instancia y si no concuerda designar a la persona a un centro médico para una atención oportuna.

- Se ha concluido que las preguntas que han realizado los usuarios entorno al covid-19 son acerca de que es el covid-19, el lugar de procedencia, síntomas, distanciamiento, prevención de la propagación, pruebas de diagnóstico, su precisión, tiempo, resultados, estadísticas de mortalidad y anticuerpos. Con lo que se puede deducir que mientras la pandemia exista, seguirán surgiendo dudas, las cuales tienen que ser confiables y concisas ya que la desinformación también causa un pánico innecesario en la población.
- En general la búsqueda de todos estos artículos que se relacionan con el covid-19 y la búsqueda de información que ha realizado la población para informarse sirve para poder ofrecer a la población una información con una categoría superior a la que se puede encontrar en cualquier página web, aportando más confianza respecto a los temas que se consultan, evitando que tengan que acudir a un médico sabiendo la exposición que esto implica en cualquier persona frente al coronavirus.

VI. REFERENCIAS

- [1] "Guidelines for performing Systematic Literature Reviews in Software Engineering | BibSonomy." <https://www.bibsonomy.org/bibtex/aed0229656ada843d3e3f24e5e5c9eb9> (accessed May 28, 2021).
- [2] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Blackwell Publishing Ltd, 2008.
- [3] C. R. Aquino-Canchari, S. G. Chávez-Bustamante, and B. S. Caira-Chuquineyra, "Exploratory analysis of internet search trends during the COVID-19 outbreak, Análisis exploratorio de las tendencias de búsqueda en internet durante el brote de la COVID-19," *Rev. Cuba. Inf. en Ciencias la Salud*, 2020, doi: 10.36512/rcics.v3i3.1631.
- [4] A. M. López, P. de Casas Moreno, and J. M. B. Mera, "Informative treatment and media

- [5] competencies on COVID-19 in Ecuador," *Rev. Comun.*, vol. 20, no. 1, pp. 137–152, Mar. 2021, doi: 10.26441/RC20.1-2021-A8.
 "Aplicaciones de búsqueda de Google para la comunicación de riesgos en el manejo de la COVID-19."
http://scielo.sld.cu/scielo.php?pid=S1561-31942020000400015&script=sci_arttext&tlang=pt (accessed Jun. 22, 2021).
- [6] D. Hu *et al.*, "More effective strategies are required to strengthen public awareness of COVID-19: Evidence from google trends," *J. Glob. Health*, vol. 10, no. 1, Jun. 2020, doi: 10.7189/jogh.10.011003.
- [7] A. Rajan, R. Sharaf, R. S. Brown, R. Z. Sharaiha, B. Lebwohl, and S. H. Mahadev, "Association of search query interest in gastrointestinal symptoms with COVID-19 diagnosis in the United States: Infodemiology study," *JMIR Public Heal. Surveill.*, vol. 6, no. 3, Jul. 2020, doi: 10.2196/19354.
- [8] H. L. Nguyen, Z. Pan, H. Abu-gellban, F. Jin, and Y. Zhang, "Google Trends Analysis of COVID-19," Nov. 2020, Accessed: Jun. 22, 2021.
 [Online]. Available:
<http://arxiv.org/abs/2011.03847>.
- [9] T. S. Higgins *et al.*, "Correlations of online search engine trends with coronavirus disease (COVID-19) incidence: Infodemiology study," *JMIR Public Heal. Surveill.*, vol. 6, no. 2, Apr. 2020, doi: 10.2196/19702.
- [10] F. C. Fang, S. N. Naccache, and A. L. Greninger, "The laboratory diagnosis of coronavirus disease 2019-frequently asked questions," *Clinical Infectious Diseases*, vol. 71, no. 11. Oxford University Press, pp. 2996–3001, Dec. 01, 2020, doi: 10.1093/cid/ciaa742.
- [11] J. Kishore, "Frequently Asked Questions (FAQs) about Coronavirus Disease 2019 (Covid-19)," *Epidemiol. Int.*, vol. 05, no. 01, pp. 14–21, 2020, doi: 10.24321/2455.7048.202004.
- [12] O. Frequently and A. Questions, "Frequently Asked Questions and Answers," *Is My J. Elig. to be Index. Int. Databases? 3rd Ed.*, vol. 2019, 2011, doi: 10.5082/xmlarchive.2010.1.13.
- [13] A. Husnayain, A. Fuad, and E. C. Y. Su, "Applications of Google Search Trends for risk communication in infectious disease management: A case study of the COVID-19 outbreak in Taiwan," *Int. J. Infect. Dis.*, vol. 95, pp. 221–223, Jun. 2020, doi: 10.1016/j.ijid.2020.03.021.
- [14] "Frequently Asked Questions and Myth Busters on COVID-19."
<http://www.saraya.com.au/index.php/about-us/news/item/frequently-asked-questions-and-myth-busters-on-covid-19> (accessed Jun. 22, 2021).

VII. Anexo 1

TABLA XIX. Estudios Seleccionados

Nº.	Estudios seleccionados	Ref.	Fuente
ES01	The Laboratory Diagnosis of Coronavirus Disease 2019— Frequently Asked Questions	[10]	Google Scholar
ES02	Frequently asked questions (FAQs) about coronavirus disease 2019 (Covid-19)	[11]	Google Scholar

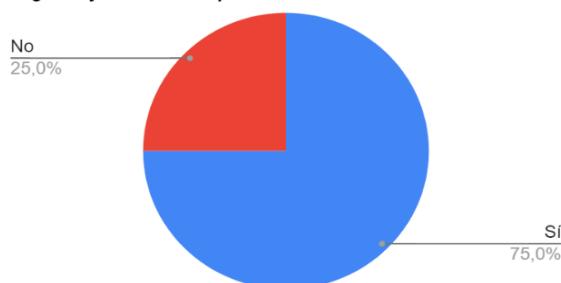
ES03	Exploratory analysis of internet search trends during the COVID-19 outbreak	[3]	Scopus
ES04	Informative treatment and media competencies on COVID-19 in Ecuador	[4]	Google Scholar
ES05	Aplicaciones de búsqueda de Google para la comunicación de riesgos en el manejo de la COVID-19	[5]	Google Scholar
ES06	Frequently Asked Questions and Answers	[14]	Google Scholar
ES07	Applications of Google Search Trends for risk communication in infectious disease management: A case study of the COVID-19 outbreak in Taiwan	[12]	Scopus
ES08	Frequently asked questions and myth busters on COVID-19	[13]	Google Scholar
ES09	More effective strategies are required to strengthen public awareness of COVID-19: Evidence from google trends	[6]	Scopus
ES10	Association of search query interest in gastrointestinal symptoms with COVID-19 diagnosis in the United States: Infodemiology study	[7]	Scopus
ES11	Google Trends Analysis of COVID-19 Pandemic	[8]	IEEE Digital Library
ES12	Correlations of online search engine trends with coronavirus disease (COVID-19) incidence: Infodemiology study	[9]	Scopus

Anexo 3: Cuestionario realizado a docentes

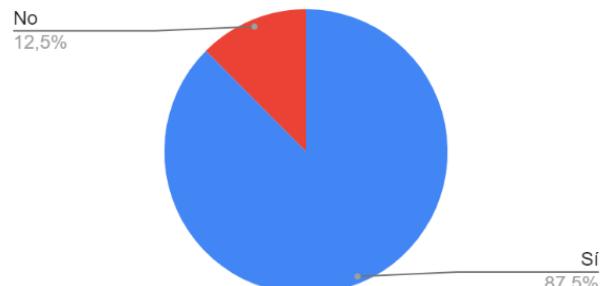
Se adjunta el los resultados del cuestionario realizado a los docentes de Facultad de Salud Humana de la Universidad Nacional de Loja.

Dirección de correo electrónico
anamaria.granda@unl.edu.ec
carmen.ullauri@unl.edu.ec
yadira.gavilanes@unl.edu.ec
patricio.espinosa@unl.edu.ec
deisy.saraguro@unl.edu.ec
catalina.araujo@unl.edu.ec
byron.salazar@unl.edu.ec
alvaro.quinche@unl.edu.ec

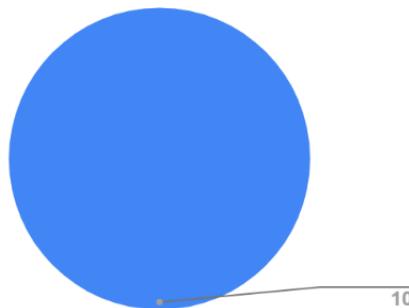
1. ¿Trabaja en el sector privado del área de salud?



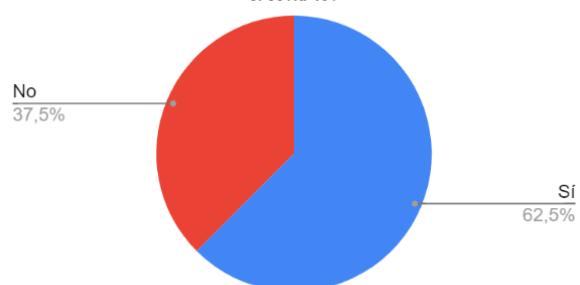
3. ¿Cree que un Chatbot (agente conversacional) sería de ayuda para los diferentes centros de salud en cuestiones el apoyo a solventar dudas comunes sobre el covid-19?



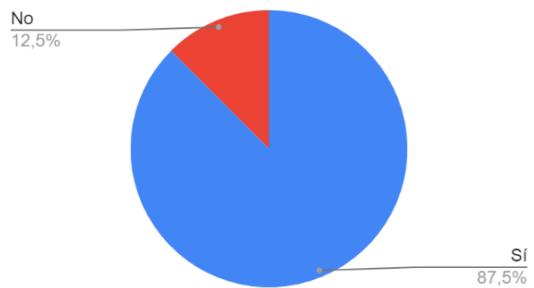
2. ¿Un chatbot ayudaría a combatir la desinformación acerca del covid-19 en temas como síntomas, vacunas y tratamiento general?



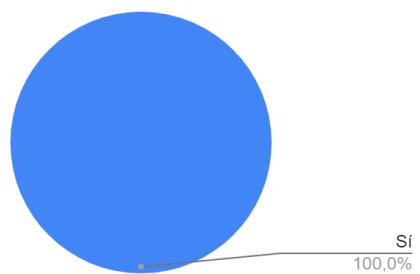
4. ¿Basándose en su experiencia, cree que un chatbot ayudaría a disminuir la población que acude al sector de salud por cuestiones relacionadas con el covid-19?



5. ¿Un chatbot ayudaría a las personas que tienen escasos recursos en solventar dudas respecto al covid-19?



6. ¿Basándose en su experiencia, cree que un chatbot ayudaría a disminuir la población que acude al sector de salud por cuestiones relacionadas con el covid-19?

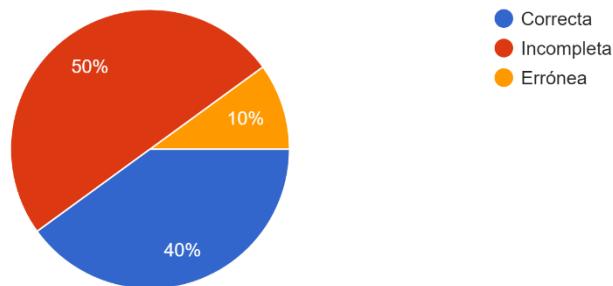


Anexo 4: Validación de Respuestas a profesionales de la salud

Se adjunta el los resultados de la Validación de las Respuestas realizado a profesionales de la salud de centros de atención médica de Loja.

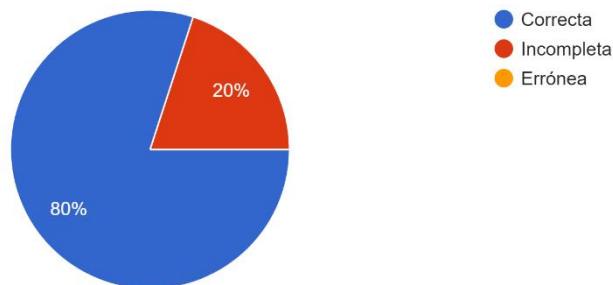
1. ¿Qué síntomas son menos comunes del covid-19?

10 respuestas



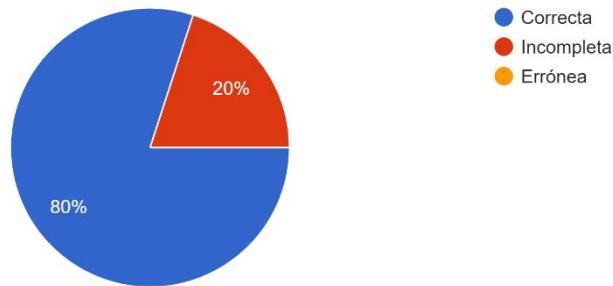
2. ¿Cuáles son los factores de riesgo del covid-19?

10 respuestas



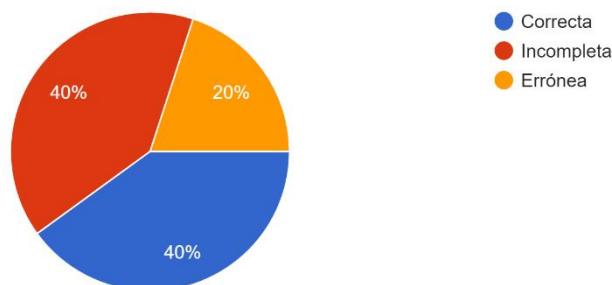
3. ¿Qué efectos secundarios tienen las vacunas?

10 respuestas



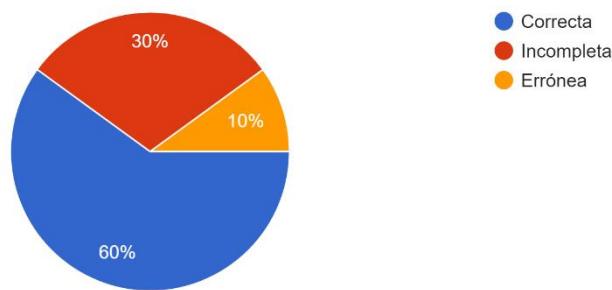
4. ¿Qué tratamientos existen contra el covid-19?

10 respuestas



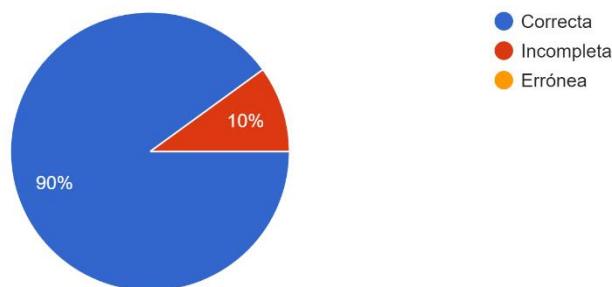
5. ¿Qué es el covid-19?

10 respuestas



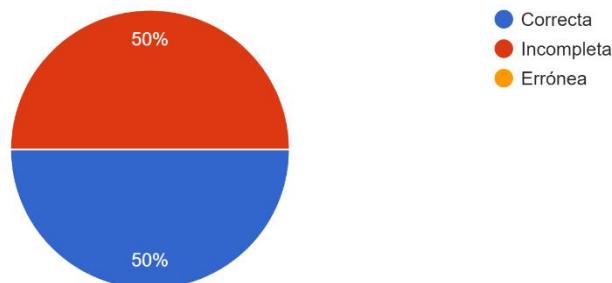
6. Cual es el lugar de origen de la enfermedad del covid-19?

10 respuestas



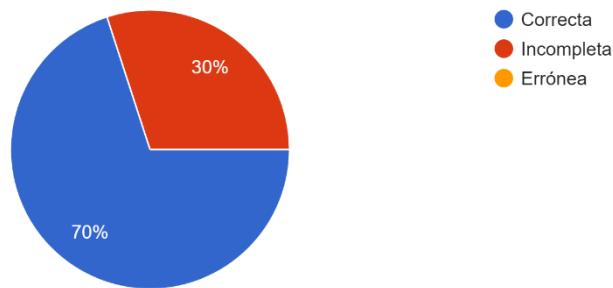
7. ¿Qué tipos de pruebas para covid-19 están disponibles?

10 respuestas



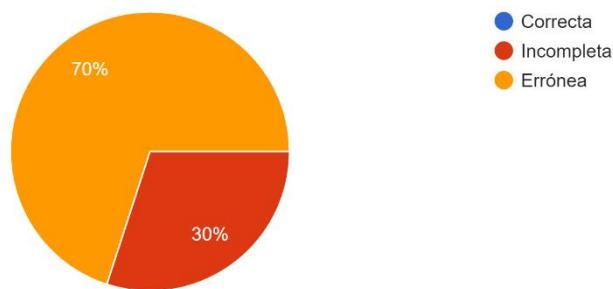
8. ¿Cuáles son las recomendaciones para aquellos con síntomas de covid-19?

10 respuestas



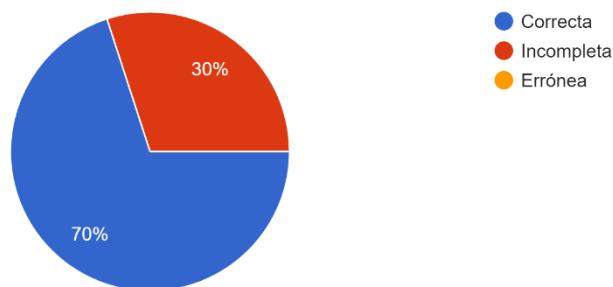
9. ¿Cuál es la tasa de mortalidad del covid-19?

10 respuestas



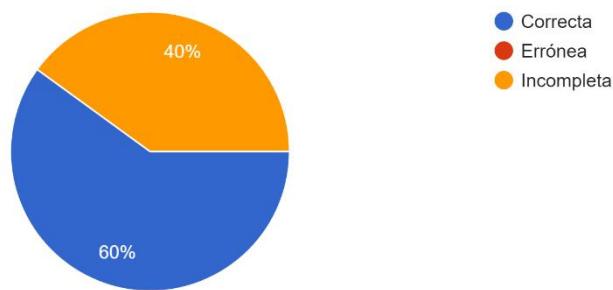
10. ¿Cuánto tiempo se tarda en generar anticuerpos para el covid-19?

10 respuestas



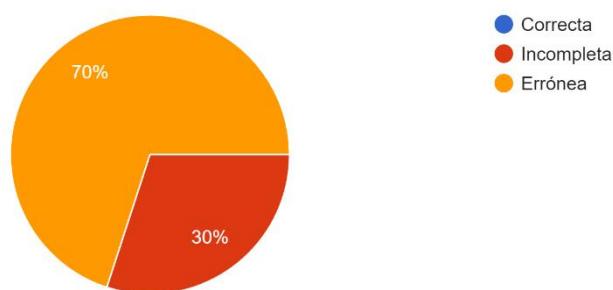
11. ¿Cómo se puede prevenir el covid-19?

10 respuestas



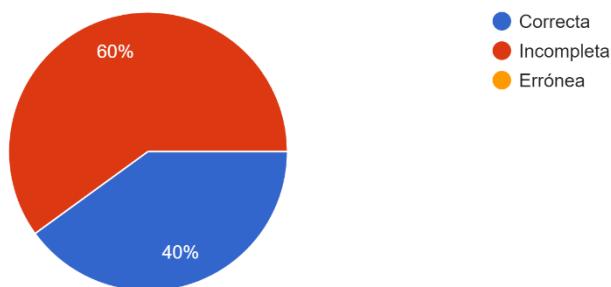
12. ¿Cuáles son los síntomas a largo plazo del covid-19?

10 respuestas



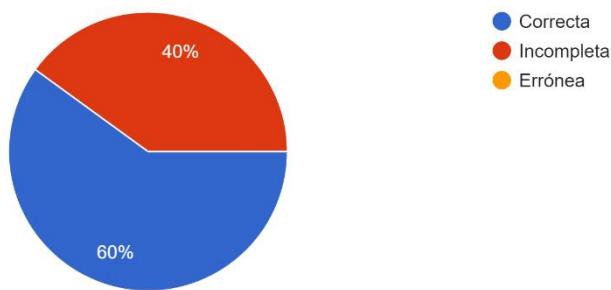
13. ¿Qué efectos secundarios producen las vacunas contra el covid-19?

10 respuestas



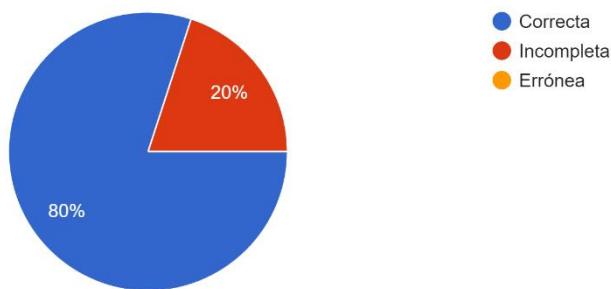
14. ¿Qué vacunas covid-19 hay disponibles?

10 respuestas



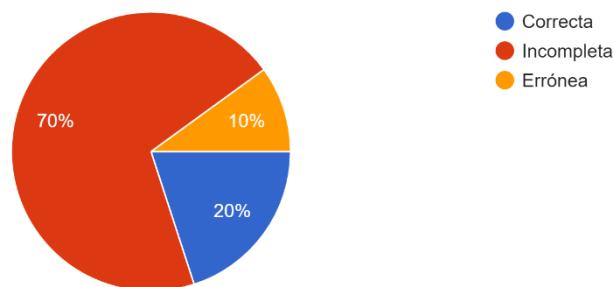
15. ¿Cuántos días de cuarentena debo realizar si tengo covid-19?

10 respuestas



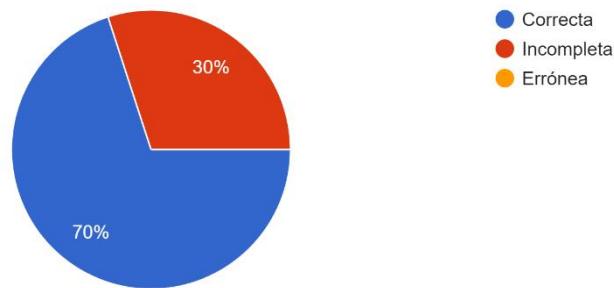
16. ¿Son eficaces los antibióticos para prevenir o tratar la COVID-19?

10 respuestas



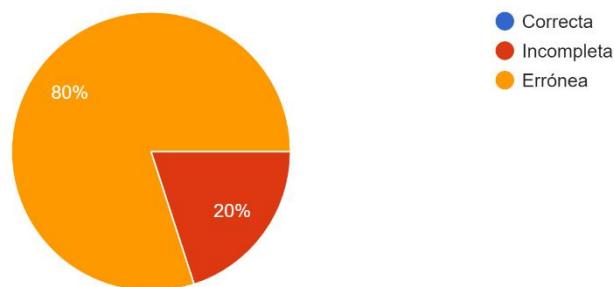
17. ¿Cuáles son los síntomas comunes del covid-19?

10 respuestas



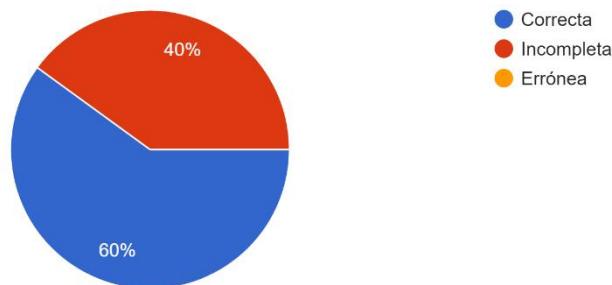
18. ¿A los cuantos días aparecen los sintomas del covid-19?

10 respuestas



19. ¿Qué personas son más vulnerables al covid-19?

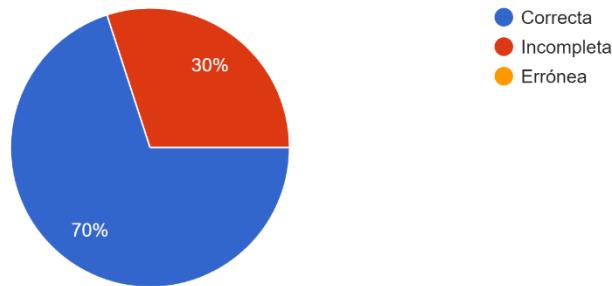
10 respuestas



- Correcta
- Incompleta
- Errónea

20. ¿Cuánto tiempo permanece el covid-19 en una superficie?

10 respuestas



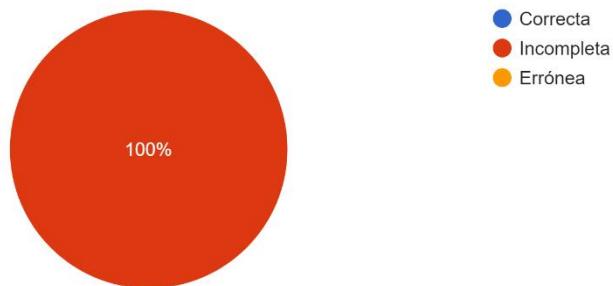
- Correcta
- Incompleta
- Errónea

Anexo 5: Validación de Respuestas a especialista en epidemiología

Se adjunta el los resultados de la Validación de las Respuestas realizadas a la especialista en epidemiología de la zona 7 Dra. Enideth Marlene Moreno Ortega Mgtr.

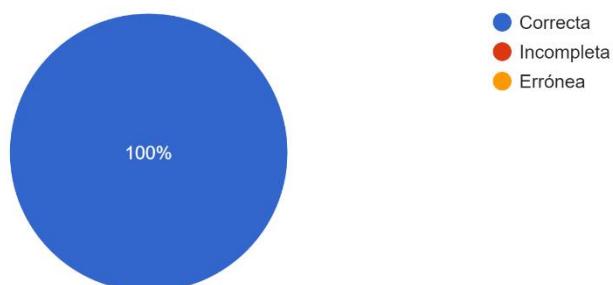
1. ¿Qué síntomas son menos comunes del covid-19?

1 respuesta



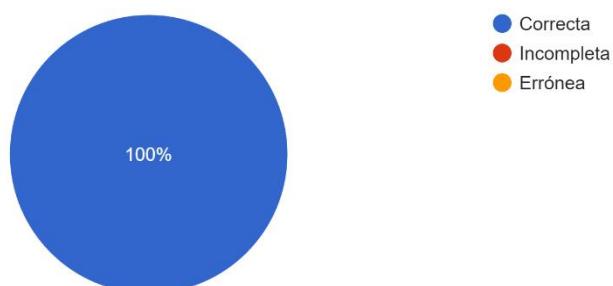
2. ¿Cuáles son los factores de riesgo del covid-19?

1 respuesta



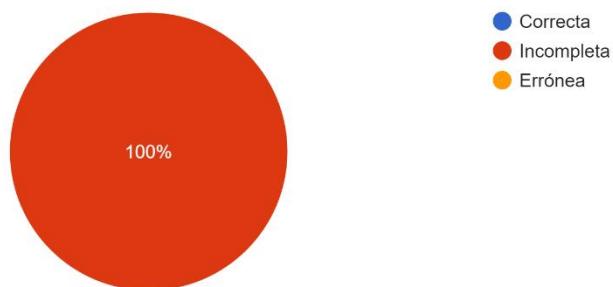
3. ¿Qué efectos secundarios tienen las vacunas?

1 respuesta



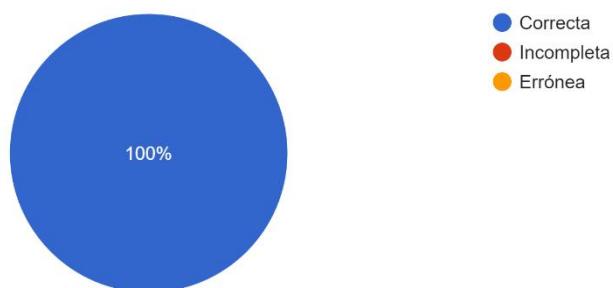
4. ¿Qué tratamientos existen contra el covid-19?

1 respuesta



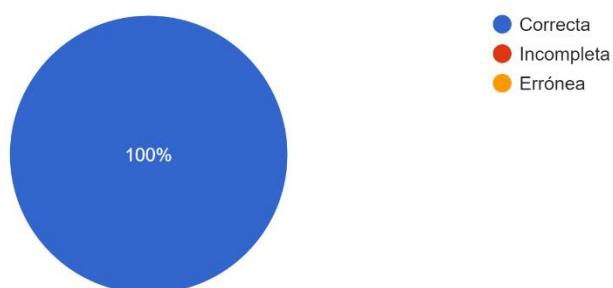
5. ¿Qué es el covid-19?

1 respuesta



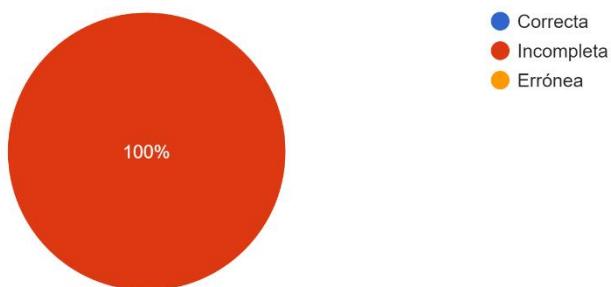
6. Cual es el lugar de origen de la enfermedad del covid-19?

1 respuesta



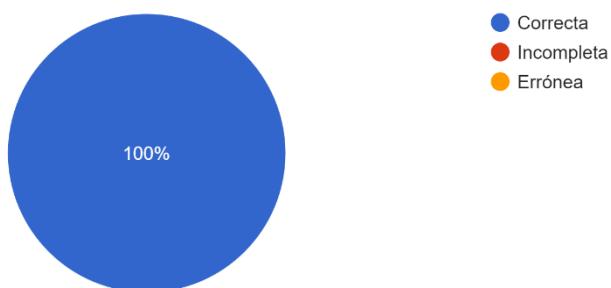
7. ¿Qué tipos de pruebas para covid-19 están disponibles?

1 respuesta



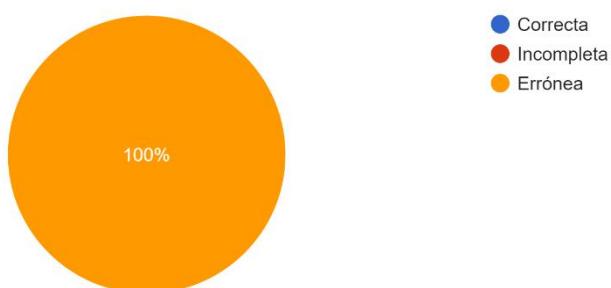
8. ¿Cuáles son las recomendaciones para aquellos con síntomas de covid-19?

1 respuesta



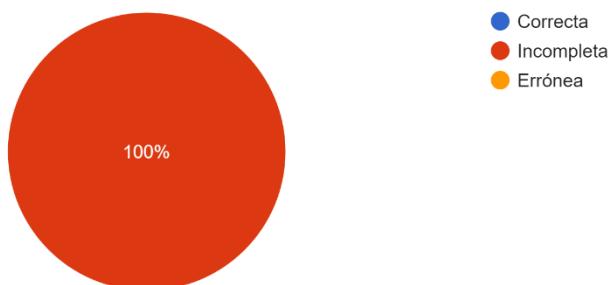
9. ¿Cuál es la tasa de mortalidad del covid-19?

1 respuesta



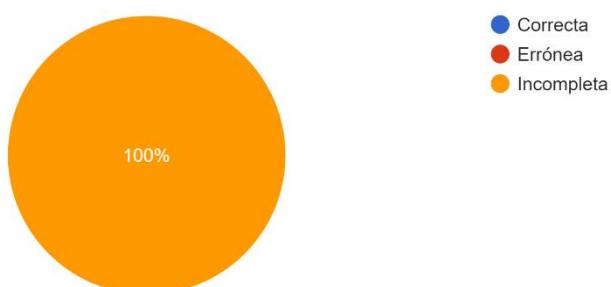
10. ¿Cuánto tiempo se tarda en generar anticuerpos para el covid-19?

1 respuesta



11. ¿Cómo se puede prevenir el covid-19?

1 respuesta



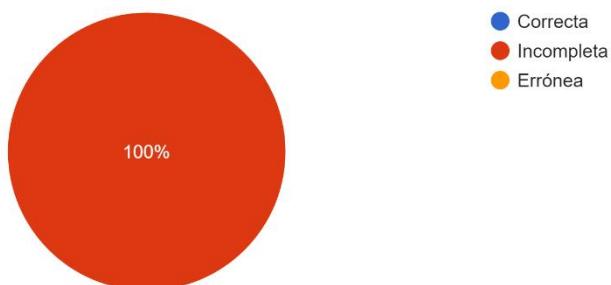
12. ¿Cuáles son los síntomas a largo plazo del covid-19?

1 respuesta



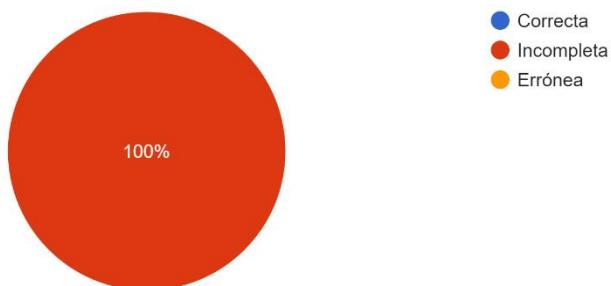
13. ¿Qué efectos secundarios producen las vacunas contra el covid-19?

1 respuesta



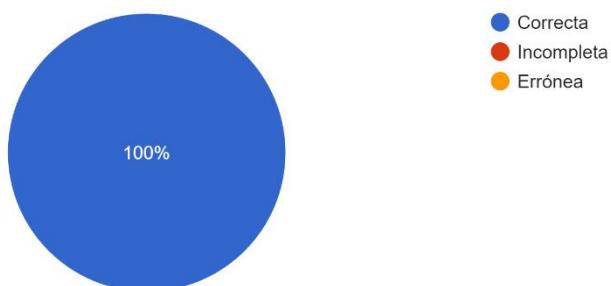
14. ¿Qué vacunas covid-19 hay disponibles?

1 respuesta



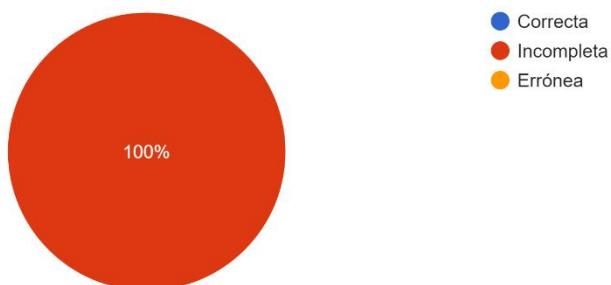
15. ¿Cuántos días de cuarentena debo realizar si tengo covid-19?

1 respuesta



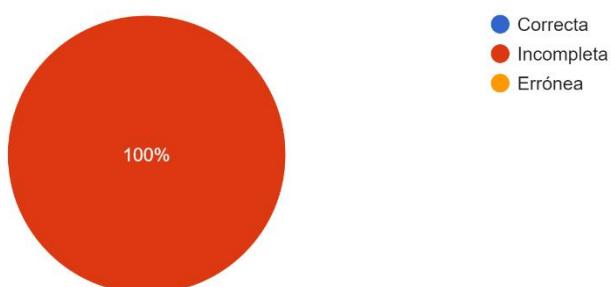
16. ¿Son eficaces los antibióticos para prevenir o tratar la COVID-19?

1 respuesta



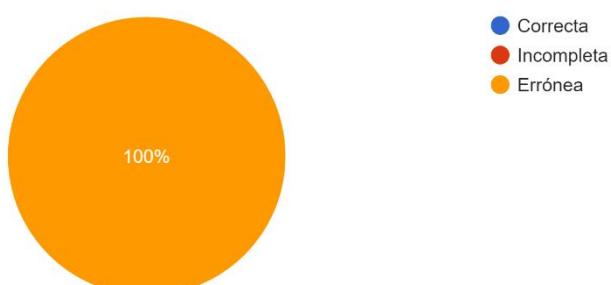
17. ¿Cuáles son los síntomas comunes del covid-19?

1 respuesta



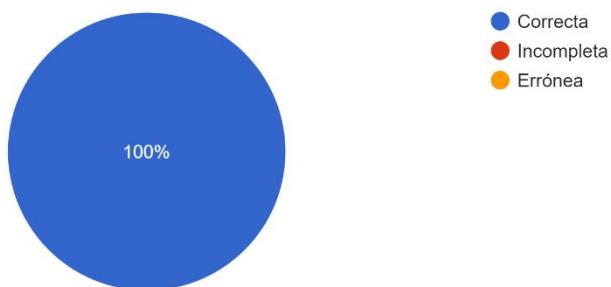
18. ¿A los cuantos días aparecen los sintomas del covid-19?

1 respuesta



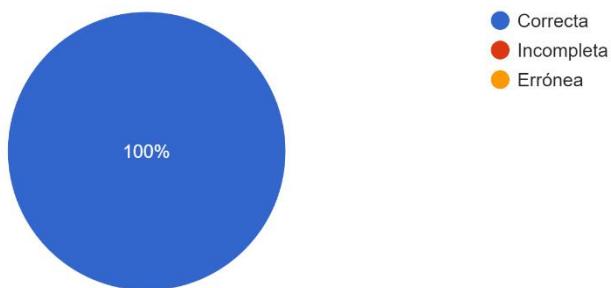
19. ¿Qué personas son más vulnerables al covid-19?

1 respuesta



20. ¿Cuánto tiempo permanece el covid-19 en una superficie?

1 respuesta



Anexo 6: Certificado de traducción del resumen

Loja, 14 de septiembre de 2022

CERTIFICACIÓN DE TRADUCCIÓN

Doctora.
Erika Lucía González Carrión, Ph.D.
Docente de la Facultad de la Educación, el Arte y la Comunicación de la Universidad Nacional de Loja

CERTIFICO:

En mi calidad de traductora del idioma Inglés, con capacidades que pueden ser probadas a través de las traducciones realizadas para revistas de alto impacto como: Comunicar(Q1): <https://bit.ly/3v0JggL> así como a través de la Certificación de conocimiento del Inglés, nivel B2, que la traducción del Resumen (Abstract) del Trabajo de Titulación denominado: "**AGENTE VIRTUAL PARA BRINDAR ASISTENCIA ACERCA DEL COVID-19**"; de la autoría de los egresados: **José Alexis Carrión Ojeda**, con CI: 1803703972 y **Víctor Yamil Serrano Zari**, con CI: 0750599391, es correcta y completa, según las normas internacionales de traducción de textos.

Es cuanto puedo certificar en honor a la verdad, facultando a los interesados, **José Alexis Carrión Ojeda y Víctor Yamil Serrano Zari**, hacer uso legal del presente, según estime conveniente.

Atentamente,



Dra. Erika González Carrión. PhD.

Docente de la Facultad de la Educación, el Arte y la Comunicación
Universidad Nacional de Loja