



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

DESARROLLO DE UN SERVICIO WEB PARA LA CLASIFICACIÓN DE NOTICIAS FALSAS APLICANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Trabajo De Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO DE SOFTWARE

AUTOR: JEFFERSON JOSÉ CHARIGUAMÁN CUJI

DIRECTORA: ING. GISEL KATERINE BASTIDAS GUACHO

Riobamba – Ecuador

2023

© 2023, Jefferson José Chariguamán Cuji

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Yo, Jefferson José Chariguamán Cuji, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 20/01/2023



Jefferson José Chariguamán Cuji

160050396-3.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo Proyecto Técnico, **DESARROLLO DE UN SERVICIO WEB PARA LA CLASIFICACIÓN DE NOTICIAS FALSAS, APLICANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO**, realizado por el señor: **JEFFERSON JOSÉ CHARIGUAMÁN CUJI**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

FIRMA

FECHA

Ing. Oscar Danilo Gavilanez Alvarez
PRESIDENTE DEL TRIBUNAL



2023-01-20

Ing. Gisel Katerine Bastidas Guacho.
DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR



2023-01-20

Ing. Patricio Rene Moreno Costales.
MIEMBRO DEL TRIBUNAL



2023-01-20

AGRADECIMIENTO

Doy un agradecimiento a mis padres por darme su apoyo emocional y espiritual, a nuestra querida y prestigiosa Institución Académica ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, que nos brindó su apertura para ser parte de la gran comunidad politécnica y llena de coraje, con valentía para alcanzar metas y objetivos propuestos y formarnos como personas que aportan a la sociedad.

Un agradecimiento especial a quienes colaboraron con la culminación del presente proyecto, a la Ing. Gisel Bastidas, directora y al Ing. Patricio Moreno, asesor, gracias por compartir su conocimiento, tiempo, dedicación y apoyo.

Jefferson

INDICE DE CONTENIDO

INDICE DE TABLAS.....	x
INDICE DE ILUSTRACIONES	xi
INDICE DE GRÁFICOS.....	xii
INDICE DE ANEXOS	xiii
RESUMEN	xiv
INTRODUCCIÓN	1

CAPÍTULO I

1	DIAGNÓSTICO DEL PROBLEMA	3
1.1	Antecedentes	3
1.2	Formulación del problema.....	4
1.3	Sistematización del problema	4
1.4	Justificación	5
1.4.1	<i>Justificación teórica</i>	5
1.4.2	<i>Justificación aplicada</i>	6
1.5	Objetivos	7
1.5.1	<i>Objetivo general</i>	7
1.5.2	<i>Objetivos específicos</i>	7

CAPÍTULO II

2	MARCO TEÓRICO.....	8
2.1	Aprendizaje automático	8
2.1.1	<i>Tipos de aprendizaje</i>	8
2.1.1.1	<i>Aprendizaje supervisado</i>	8
2.1.1.2	<i>Aprendizaje por refuerzo</i>	8
2.1.1.3	<i>Aprendizaje no supervisado</i>	8
2.1.2	<i>Técnicas de aprendizaje automático para la clasificación de texto</i>	9
2.1.2.1	<i>Naive Bayes</i>	9

2.1.2.2	<i>Regresión logística</i>	10
2.1.2.3	<i>Máquinas de soporte vectorial</i>	10
2.1.2.4	<i>K-vecino cercano</i>	11
2.1.2.5	<i>Árboles de decisión</i>	12
2.1.2.6	<i>Redes neuronales</i>	13
2.1.3	<i>Aplicaciones del aprendizaje automático</i>	18
2.1.3.1	<i>Medicina</i>	18
2.1.3.2	<i>Educación</i>	18
2.1.3.3	<i>Construcción</i>	19
2.1.3.4	<i>Robótica</i>	19
2.1.4	<i>Lenguajes para el aprendizaje automático</i>	19
2.1.4.1	<i>Lenguaje Python</i>	19
2.1.4.2	<i>Lenguaje R</i>	20
2.1.4.3	<i>Lenguaje Matlab</i>	20
2.1.5	<i>Librerías y Frameworks</i>	21
2.1.5.1	<i>Keras</i>	21
2.1.5.2	<i>Scikit-learn</i>	21
2.1.5.3	<i>Tensorflow</i>	22
2.2	<i>Servicios Web</i>	22
2.2.1	<i>API REST</i>	22
2.3	<i>Metodología CRISP-DM</i>	23
2.3.1	<i>Comprensión del negocio</i>	23
2.3.2	<i>Entendiendo los datos</i>	24
2.3.3	<i>Preparación de los datos</i>	24
2.3.4	<i>Modelado</i>	24
2.3.5	<i>Evaluación</i>	24
2.3.6	<i>Despliegue</i>	24
2.4	<i>Metodología SCRUM</i>	24
2.4.1	<i>Roles Scrum</i>	25

2.4.2	<i>Eventos scrum</i>	25
2.4.3	<i>Artefactos Scrum</i>	26
2.5	Métodos de evaluación	27
2.5.1	<i>F1 score</i>	27
2.5.2	<i>Eficiencia de desempeño</i>	27
2.6	Trabajos relacionados	28

CAPÍTULO III

3	MARCO METODOLÓGICO	30
3.1	Tipo de estudio	30
3.2	Métodos y técnicas	30
3.3	Determinación de la población y muestra	33
3.4	Estudio de factibilidad	33
3.4.1	<i>Factibilidad técnica</i>	33
3.4.2	<i>Factibilidad operativa</i>	35
3.4.3	<i>Factibilidad económica</i>	35
3.5	Metodología de desarrollo CRISP-DM	36
3.5.1	<i>Comprensión del negocio</i>	36
3.5.1.1	<i>Objetivo de negocio</i>	36
3.5.1.2	<i>Evaluar la situación</i>	36
3.5.1.3	<i>Objetivo de la minería de datos</i>	36
3.5.2	<i>Comprensión de los datos</i>	36
3.5.2.1	<i>Recopilar los datos iniciales</i>	36
3.5.2.2	<i>Describir los datos</i>	36
3.5.2.3	<i>Explorar los datos</i>	37
3.5.3	<i>Preparación de los datos</i>	38
3.5.3.1	<i>Recolección de los conjuntos de datos desde diferentes fuentes</i>	38
3.5.3.2	<i>Descartar las noticias correspondientes a tweets</i>	38
3.5.3.3	<i>Integración de los conjuntos de datos en uno solo</i>	38

3.5.3.4	<i>Eliminar documentos con campos nulos o vacíos</i>	38
3.5.3.5	<i>Eliminación de ruido en los documentos</i>	38
3.5.4	<i>Modelado</i>	38
3.5.4.1	<i>Escoger la técnica de modelado</i>	39
3.5.4.2	<i>Generar el plan de prueba</i>	39
3.5.4.3	<i>Construcción del modelo</i>	40
3.5.5	<i>Evaluación</i>	43
3.5.5.1	<i>Red MLP</i>	43
3.5.5.2	<i>Red CNN</i>	44
3.5.5.3	<i>Red LSTM</i>	45
3.5.6	<i>Despliegue</i>	47
3.6	<i>Metodología de desarrollo SCRUM</i>	47
3.6.1	<i>Fases de planificación</i>	47
3.6.1.1	<i>Requerimientos Funcionales</i>	47
3.6.1.2	<i>Personas y roles involucrados en el proyecto</i>	48
3.6.1.3	<i>Product Backlog</i>	49
3.6.1.4	<i>Sprint Backlog</i>	50
3.6.1.5	<i>Historias de usuario</i>	51
3.6.2	<i>Fase de desarrollo</i>	53
3.6.2.1	<i>Arquitectura del sistema</i>	53
3.6.2.2	<i>Diseño de interfaz</i>	53
3.6.3	<i>Fase de pruebas</i>	55
3.6.4	<i>Fase de despliegue</i>	56

CAPÍTULO IV

4	RESULTADOS	58
4.1	Eficiencia de desempeño	58
4.1.1	<i>Tiempo de respuesta</i>	58
4.2	Despliegue de la aplicación	59

4.3	Tiempos de respuesta	59
4.3.1	<i>Tiempos de respuesta con noticias de longitud corta</i>	60
4.3.2	<i>Tiempos de respuesta con noticias de longitud media</i>	63
4.3.3	<i>Tiempos de respuesta con noticias de longitud larga.....</i>	67
4.3.4	<i>Tiempos de respuestas totales</i>	71
	CONCLUSIONES.....	78
	RECOMENDACIONES.....	80
	GLOSARIO	
	BIBLIOGRAFÍA	
	ANEXOS	

INDICE DE TABLAS

Tabla 1-3: Métodos y técnicas.	31
Tabla 2-3: Recursos hardware.	33
Tabla 3-3: Recursos Software.	34
Tabla 4-3: Recursos humanos.	34
Tabla 5-3: Costos del proyecto.	35
Tabla 6-3: Descripción de los conjuntos de datos.	37
Tabla 7-4: Comparativa entre técnicas de aprendizaje automático.	39
Tabla 8-3: Resultados de entrenamiento de redes neuronales.	47
Tabla 9-3: Requerimientos funcionales.	48
Tabla 10-3: Requerimiento no funcional.	48
Tabla 11-3: Personas y roles involucrados en el proyecto.	49
Tabla 12-3: Valoraciones de tiempo según T-Shirt.	49
Tabla 13-3: Estimación de tareas.	49
Tabla 14-3: Sprint Backlog.	50
Tabla 15-3: HU1 Crear interfaz de usuario.	52
Tabla 16-3: TI_HU1 Crear interfaz de usuario.	52
Tabla 17-3: T2_HU1 Crear interfaz de usuario.	52
Tabla 18-3: PA_TI_HU Crear interfaz de usuario.	55
Tabla 19-3: PA_T2_HU1 Crear interfaz de usuario.	56
Tabla 20-3: Métricas para la característica de calidad Eficiencia en el desempeño.	58
Tabla 21-4: Descripción de las categorías de las noticias.	59
Tabla 22-4: Resumen de tiempos noticias cortas.	60
Tabla 23-4: Tabla de cuartiles, noticias cortas.	61
Tabla 24-4: Resumen de tiempos noticias medias.	64
Tabla 25-4: Resumen de cuartiles noticias medias.	65
Tabla 26-4: Resumen de tiempos noticias largas.	68
Tabla 27-4: Resumen de cuartiles noticias largas.	69
Tabla 28-4: Tiempos de respuesta sin datos atípicos noticias cortas.	72
Tabla 29-4: Tiempos de respuesta sin datos atípicos noticias medias.	73
Tabla 30-4: Tiempos de respuesta sin datos atípicos noticias largas.	73
Tabla 31-4: Resumen de tiempos finales de respuesta.	76
Tabla 32-4: Límites de tiempo de respuesta.	76

INDICE DE ILUSTRACIONES

Ilustración 1-2.	Neurona de la regresión logística.....	10
Ilustración 2-2	Modelo de neurona artificial estándar.....	14
Ilustración 3-2	Arquitectura neuronal.....	14
Ilustración 4-2	Red neuronal profunda completamente conectada.....	15
Ilustración 5-2	Estructura básica de una red neuronal recurrente.....	16
Ilustración 6-2	Estructura de una celda LSTM.....	17
Ilustración 7-2	Arquitectura de red neuronal convolucional para la clasificación de texto ...	18
Ilustración 8-2	Fases de la metodología CRISP-DM.....	23
Ilustración 9-3.	Características de la ISO 25000.....	28
Ilustración 10-3.	Arquitectura de red neuronal MLP.....	40
Ilustración 11-3.	Arquitectura de red neuronal convolucional.....	41
Ilustración 12-3.	Arquitectura de red neuronal LSTM.....	42
Ilustración 13-3.	Arquitectura del servicio web.....	53
Ilustración 14-3.	Boceto de la interfaz de usuario.....	54
Ilustración 15-3:	Repositorio en GitHub.....	56
Ilustración 16-3:	Despliegue del servicio web.....	57

INDICE DE GRÁFICOS

Gráfico 1-3.	Puntuación F1 MLP.	43
Gráfico 2-3.	Función de pérdida MLP.....	44
Gráfico 3-3.	Puntuación F1 CNN.....	45
Gráfico 4-3.	Función de pérdida CNN.....	45
Gráfico 5-3.	Función de pérdida LSTM.	46
Gráfico 6-3.	Puntuación F1 LSTM.....	46
Gráfico 7-4.	Tiempos de respuesta noticias longitud corta.	60
Gráfico 8-4.	Diagrama de cajas, tiempos de respuesta noticias de longitud corta.....	61
Gráfico 9-4.	Tiempos de respuestas atípicos de noticias longitud corta.....	62
Gráfico 10-4.	Tiempos de clasificación atípicos de noticias longitud corta.	63
Gráfico 11-4.	Tiempos de preprocesamiento atípicos de noticias longitud corta.....	63
Gráfico 12-4.	Tiempos de respuesta noticias longitud media.	64
Gráfico 13-4.	Diagrama de cajas, tiempos de respuesta de longitud media.	65
Gráfico 14-4.	Tiempos de respuestas atípicos de noticias longitud media.	66
Gráfico 15-4.	Tiempos de clasificación atípicos de noticias longitud media.	66
Gráfico 16-4.	Tiempos de preprocesamiento atípicos de noticias longitud media.	67
Gráfico 17-4.	Tiempos de respuesta noticias longitud larga.	68
Gráfico 18-4.	Diagrama de cajas, tiempos de respuesta noticias de longitud larga.....	69
Gráfico 19-4.	Tiempos de respuesta atípicos de noticias longitud larga.	70
Gráfico 20-4.	Tiempos de clasificación atípicos de noticias longitud larga.	70
Gráfico 21-4.	Tiempos de preprocesamiento atípicos de noticias longitud larga.....	71
Gráfico 22-4.	Tiempos de respuesta sin datos atípicos, noticias longitud corta.	72
Gráfico 23-4.	Tiempos de respuesta sin datos atípicos, noticias longitud media.	72
Gráfico 24-4.	Tiempos de respuesta sin datos atípicos, noticias longitud larga.	73
Gráfico 25-4.	Comparación de tiempos mínimos de respuesta.	74
Gráfico 26-4.	Comparación de tiempos promedios de respuesta.	75
Gráfico 27-4.	Comparación de tiempos máximos de respuesta.	75

INDICE DE ANEXOS

Anexo A: Manual de usuario

Anexo B: Encuestas realizadas

Anexo C: Historias técnicas, tareas de ingeniería y pruebas de aceptación

Anexo D: Manual técnico

RESUMEN

El presente Trabajo de Integración Curricular planteó como objetivo el desarrollo de un servicio web para la clasificación de notificaciones falsas en Español, integrando un modelo de aprendizaje automático para la clasificación. En el desarrollo se utilizaron los métodos analítico y deductivo, para determinar los fundamentos teóricos, además se aplicó la metodología SCRUM para el servicio web, donde se establecieron requerimientos, personas y roles involucrados en el proyecto, también se definieron 7 historias técnicas y 2 historias de usuarios, además se realizó el sprint backlog para planificar el periodo de realización de las historias. Por otro lado, se utilizó la metodología CRISP-DM, cubriendo tareas de recolección del conjunto de datos, preprocesamiento, selección de las técnicas a implementar, finalmente se evaluó y seleccionó la técnica con mejor rendimiento para su integración en el servicio web. Tanto el servicio web como la implementación del modelo, fueron realizados mediante el lenguaje de programación Python, con el uso de librerías de machine learning. Para evaluar la eficiencia de desempeño en el servicio web, se basó en la norma ISO/IEC 25010, tomando como métrica el tiempo de respuesta, para evaluar el comportamiento temporal. En el análisis se determinó que el servicio web tiene un tiempo de respuesta promedio de 1.08 segundos. Por otra parte, para la evaluación del rendimiento de los modelos de aprendizaje automático se utilizó la métrica F1, obteniendo que la arquitectura LSTM tiene el mejor rendimiento con una puntuación de 0.746. Finalmente, se recomienda que, para el entrenamiento de modelos de aprendizaje automático, se tenga en cuenta el tamaño del conjunto de datos optando por conjuntos de datos balanceados en el número de datos existentes por clase, de tal forma que no se afecte los resultados de la clasificación.

Palabras clave: <INGENIERÍA DE SOFTWARE>, <INTELIGENCIA ARTIFICIAL>, <API REST>, <METODOLOGÍA SCRUM>, <REDES NEURONALES>, <ISO/IEC 25010>



A handwritten signature in blue ink, appearing to read "Ana María".

0238-DBRA-UPT-2023

SUMMARY

The objective of this Curricular Integration Project was the development of a web service for the classification of false notifications in Spanish, integrating a machine learning model for the classification. We utilized the analytical and deductive methods in the development in order to determine the theoretical foundations, and we applied the SCRUM methodology for the web service, where we established the requirements, people and roles involved in the project. Seven technical stories and two user stories were defined, too and a sprint backlog was created to plan the period for the completion of the stories. On the other hand, we utilized the CRISP-DM methodology, covering tasks of data set collection, preprocessing, selection of the techniques to be implemented. Finally, we evaluated and selected the best performing technique for integration into the web service. Both the web service and the implementation of the model were carried out using the Python programming language, with the use of machine learning libraries. We based ourselves on the ISO/IEC 25010 standard to evaluate the performance efficiency of the web service, taking response time as a metric to evaluate the temporal behavior. The analysis determined that the web service has an average response time of 1.08 seconds. On the other hand, the F1 metric was utilized for the evaluation of the performance of the machine learning models, obtaining that the LSTM architecture has the best performance with a score of 0.746. Finally, we recommend that for the training of machine learning models, the size of the data set be taken into account, selecting data sets balanced in the number of existing data per class, so that the classification results will not be affected.

Key words: <SOFTWARE ENGINEERING>, <ARTIFICIAL INTELLIGENCE>, <API REST>, <SCRUM METHODOLOGY>, <NEURONALS NETWORKS>, <ISO/IEC 25010 STANDARD>.

Translated by:



Lic. Carolina Campaña D. Mgs.

ID number: 1804191482

EFL Professor

0238-DBRA-UPT-2023

INTRODUCCIÓN

Las noticias falsas siempre han existido, y en la actualidad por uso de medios electrónicos y la tecnología, se ha incrementado, y debido a la pandemia las personas han optado por usar medios digitales como fuentes de información, tales como las redes sociales, sin embargo, existe gran cantidad de noticias no verificadas, resultando ser falsas. Además, se ha observado una mayor irrupción de estas noticias en las plataformas de acceso abierto y gratuito, la circulación de estas noticias puede provocar desde robo de información, estafas, histeria colectiva, desprestigio del honor de una persona o negocio, hasta ocasionar desestabilización de gobiernos. Con el fin contrarrestar la propagación de noticias falsas, las nuevas tecnologías utilizan técnicas de inteligencia artificial para su detección automática, un campo clave es el aprendizaje automático conocido en inglés como Machine Learning. Por otro lado, la mayoría de los modelos de aprendizaje automático que permiten detectar noticias falsas solo lo realizan en textos en el idioma inglés. Por tal motivo, se ve la necesidad de la implementación de un servicio web que integre un modelo de aprendizaje automático para la clasificación de noticias falsas en español.

Agudelo et al. (2018), propone el uso de Naive Bayes para la clasificación de noticias falsas, donde se utiliza como conjunto de datos Fake_real_news_dataset¹, y se obtiene una certeza del 84%. Por otro lado, en Prada (2019), se aplica la técnica de web scrapping para obtener tweets y así crear el conjunto de datos para entrenar los modelos de clasificación: Árbol de decisión, Máquina de soporte vectorial, Naive Bayes, Regresión logística. Además, usando la puntuación F1 como métrica de evaluación determinan el mejor modelo, siendo la Regresión Logística el modelo con mayor F1. Así mismo, en Martínez-Gallego et al. (2021), se utiliza el conjunto de datos FakesNewsCorpusSpanish² correspondiente a una recopilación de noticias del 2018 desde varios medios, por otro lado, los modelos evaluados son: Máquina de soporte vectorial, Regresión logística, Bosque aleatorio, y Boosting. En este estudio usan la precisión como métrica para comprobar el rendimiento de los modelos, se determina que el modelo de Bosque aleatorio es el mejor con una precisión de 76,94%.

Entre las herramientas utilizadas para la definir la arquitectura y entrenar los modelos de Machine Learning se encuentran las librerías como keras, scikit-learn, spacy, y tensorflow. Por otro lado, para al área del software, en el desarrollo del servicio web, está el framework Flask escrito en el lenguaje Python, el mismo que es utilizado para la implementación del modelo, además de otros lenguajes como HTML, CSS y Javascript.

¹ https://github.com/GeorgeMcIntire/fake_real_news_dataset

² <https://github.com/jpposadas/FakeNewsCorpusSpanish>

El presente documento se encuentra dividido en cuatro capítulos, estos se detallan a continuación:

Capítulo I: Presenta el planteamiento del problema describiendo los antecedentes de la problemática, justificación teórica y aplicativa, también se menciona el objetivo general del proyecto y objetivos específicos.

Capítulo II: Contiene la revisión de literatura y la conceptualización teórica, mediante la revisión de trabajos relacionados al tema del proyecto, además se definen las herramientas para el desarrollo del servicio web.

Capítulo III: Detalla el tipo de estudio, los métodos y técnicas utilizadas para el desarrollo y evaluación del servicio web, mediante la aplicación de las metodologías CRISP-DM y SCRUM, para la implementación del modelo entrenado y el servicio web respectivamente.

Capítulo IV: Presenta los resultados obtenidos en función de los objetivos planteados, además se especifican las conclusiones y recomendaciones.

CAPÍTULO I

1 DIAGNÓSTICO DEL PROBLEMA

En esta sección se describe los antecedentes, justificación y objetivos relacionados a la problemática a resolver mediante el desarrollo del presente trabajo.

1.1 Antecedentes

En la actualidad, el uso de los medios electrónicos y la tecnología en el país se ha incrementado, debido a la pandemia, obligando a personas a usar medios digitales, como redes sociales para informarse, sin embargo, hay gran cantidad de noticias no verificadas, que pueden resultar ser falsas (Rashid et al., 2020). Por otro lado, el aprendizaje automático conocido en inglés como *Machine Learning*, es una subárea de la Inteligencia Artificial. Estos son sistemas que aprenden automáticamente, basados en un conjunto de algoritmos construidos para resolver problemas y tomar decisiones mediante la experiencia acumulada en los casos resueltos con anterioridad de tal manera que se mejorara su respuesta ante nuevas tareas (Moreno et al., 1994).

Los algoritmos o modelos de aprendizaje automático son aplicados para resolver distintas problemáticas, como por ejemplo la clasificación automática de texto. El uso de estos clasificadores en la actualidad está presente en varias áreas, en las cuales se clasifica el texto para determinar si corresponde a un mensaje racista, habla sobre política, un comentario positivo o negativo, el idioma en el que se encuentra escrito, entre otros. Además, se ha utilizado clasificadores para determinar si una noticia es verdadera o falsa basado en el análisis del texto (Guardiola, 2020).

Existen varios modelos de clasificación de noticias para determinar si son verdaderas o falsas, estos modelos suelen ser supervisados lo cual requiere tener un conjunto de datos etiquetados para permitir el aprendizaje. Por lo que muchas veces, además de la técnica de clasificación utilizada, el desempeño del modelo se puede ver afectado por la calidad del conjunto de datos usados durante el entrenamiento del modelo. Actualmente, la mayoría de los conjuntos de datos se encuentran en inglés, por lo que los modelos para la detección de noticias falsas están enfocados a este lenguaje, debido a esto existe la necesidad de construir un modelo que permita realizar la clasificación de noticias falsas en español. Por otro lado, las noticias falsas siempre han estado presentes, pero en la actualidad se observa una mayor irrupción en las plataformas de acceso abierto y gratuito. La circulación de noticias falsas puede provocar desde el robo de información,

estafas, histeria colectiva, desprestigio del honor de una persona o negocio, hasta ocasionar la desestabilización de los gobiernos (Durán y Tusa, 2019).

Dados los inconvenientes en los modelos y la problemática existente en la sociedad por la presencia de noticias falsas. En el presente trabajo se propone entrenar un modelo que permita la clasificación de noticias, basado en un conjunto de datos en español, complementando también dicho conjunto de datos con noticias identificadas en Ecuador, para minimizar y mitigar la desinformación que existe actualmente en medios digitales. Además, integrando la inteligencia artificial con la ingeniería de software, también se propone disponibilizar este clasificador a través de un servicio web para ser consumido por los usuarios. Este servicio web desarrollado en esta investigación permitirá verificar si el texto correspondiente a una noticia es clasificado como una noticia falsa o verdadera.

1.2 **Formulación del problema**

¿Cómo el desarrollo de un servicio web permite la clasificación de noticias falsas mediante el uso de técnicas de aprendizaje automático?

1.3 **Sistematización del problema**

¿Qué técnicas de aprendizaje automático son utilizados para la clasificación de textos de noticias falsas?

¿Qué técnica del estado del arte del aprendizaje automático tiene el mejor desempeño para la clasificación de noticias falsas?

¿Qué servicio web es el más adecuado para la integración con modelos de aprendizaje automático?

¿Qué modelo entrenado tiene la mejor puntuación F1?

¿Cuál es el comportamiento temporal del servicio web desarrollado?

1.4 **Justificación**

1.4.1 *Justificación teórica*

El lenguaje de programación Python fue publicado en 1991, creado por Guido Van Rossum, este es un lenguaje interpretado que permite el desarrollo más rápido al no requerir compilación y construcción. Según Sarkar et al. (2018), establece varias ventajas en cuanto a este lenguaje como, la facilidad de aprendizaje, la curva de aprendizaje es baja, permitiendo al desarrollador entender y aprender fácilmente. Tiene soporte a múltiples paradigmas de programación como, orientada a objetos, estructurada, funcional, y orientada a eventos. Una de las más importantes características es la extensibilidad, que tiene mediante la instalación de módulos y la importación de librerías facilitando la resolución de distintos problemas en diferentes áreas. Una de las aplicaciones que tiene este lenguaje, es el entrenamiento y definición de modelos de aprendizaje automático utilizando las librerías disponibles.

Para Dineva y Atanasova (2020), el aprendizaje automático es el estudio y el uso de algoritmos matemáticos, que permiten mejorar el rendimiento sin la necesidad de una interacción humana, estos algoritmos de aprendizaje automático son considerados como un subconjunto de la inteligencia artificial. También considera que la aplicación de algoritmos de aprendizaje automático se ha realizado en muchas áreas para la resolución de diferentes tareas, pero aclara que para obtener la máxima precisión de los resultados objetivos es necesario la aplicación de algoritmos correctos.

Keras, es una librería basada en aprendizaje automático, utilizadas para la creación de modelos de redes neuronales artificiales, esta herramienta ofrece API's consistentes y simples, además, provee de una retroalimentación en casos de error que permiten su resolución, con respecto a la documentación de esta herramienta es extensa, favoreciendo a los desarrolladores para su uso (Keras: the Python deep learning API, 2022) .

Según la plataforma oficial, Tensorflow es una librería de código abierto que permite el desarrollo y entrenamiento de modelos, su funcionamiento con Keras permite una interacción de modelos instantáneamente y con una fácil depuración, la implementación de esta librería es utilizada por muchos desarrolladores que tienen como objetivo resolver problemas complejos del mundo real, como la creación de modelos, y debido a los diversos beneficios que supone, es usado por compañías como Google, Intel, DeepMind, Twitter, entre otros (TensorFlow, 2022).

Google Colab, también llamado Collaboratory, es un entorno basado en Jupyter notebook que lleva a cabo su ejecución en los servidores de Google, este entorno trae consigo la configuración necesaria para el uso de Python, los beneficios que provee esta plataforma son los recursos gratuitos como, GPU Nvidia K80, TPU y CPU, también permite la integración con Google Drive para el almacenamiento de proyectos o recursos (G.L, 2021).

Según la “World Wide Web Consortium”, define a un Servicio Web como “Un sistema de software diseñado para soportar la interacción interoperable máquina a máquina sobre una red. (Web Services Architecture, 2004). Facilita un servicio a través de internet, donde el cliente envía una solicitud a un servidor, desencadenando una acción por parte de este, una vez procesada la información, se devuelve la respuesta al cliente (Web services, 2020).

El comportamiento temporal, forma parte de la característica Eficiencia de desempeño, según la ISO 25000 (2019), define como los tiempos de respuesta, procesamiento y los ratios de rendimiento de un sistema, cuando se lleva a cabo sus funciones bajo condiciones determinadas, en relación con un banco de pruebas establecido.

Basado en los beneficios de las diferentes herramientas descritas anteriormente, estas serán utilizadas en el presente Trabajo de Integración Curricular, para el desarrollo del servicio web con integración al modelo entrenado para la clasificación de noticias falsas.

1.4.2 *Justificación aplicativa*

En el contexto actual de pandemia, los medios electrónicos como las redes sociales son plataformas convenientes para la difusión de noticias falsas, uno de los más medios más grandes por su tamaño y alcance es Facebook, a fines de 2020 tenía alrededor de 2.800 millones de usuarios en todo el mundo (RT en Español, 2021). Según estadísticas del informe Digital 2021, el 56.4% de los usuarios de internet, manifiestan preocupación por no saber si las noticias publicadas en medios digitales son verdaderas o falsas (Digital 2021, 2021).

Según las estadísticas de la Arab Youth Survey muestran que, en el año 2020, el 79% de ellos recibía las noticias a través de las redes sociales, en comparación con el 25% en el 2015. En contexto de pandemia, según la Revista Panamericana de salud, se encontraron 329 noticias falsas relacionadas con el Covid-19, estas noticias fueron divulgadas por medios como, WhatsApp y Facebook.

Debido a la gran cantidad de información, determinar si una noticia es verdadera o falsa mediante el uso de medios tradicionales no es factible, por el tiempo y recursos que ocupa, por este motivo, la aplicación de técnicas de aprendizaje automático para realizar clasificaciones de texto ha ido aumentando en los últimos años, sin embargo, debido a que las noticias falsas varían continuamente, se requiere de la selección adecuada la técnica de aprendizaje automático para obtener los resultados correctos, además, para el entrenamiento del modelo, se debe disponer de un conjunto de datos adecuado.

Por otro lado, exponer los servicios de clasificación de noticias falsas mediante el desarrollo de un servicio web, permite promover la interoperabilidad con aplicaciones desarrolladas anteriormente o por desarrollar. Con el desarrollo de este servicio web para la clasificación de noticias falsas se pretende proveer de un servicio para otros sistemas que quieran consumir las funciones.

De acuerdo con el eje y lineamientos de investigación de la Escuela superior politécnica de Chimborazo, la presente propuesta de proyecto de integración curricular, se encuentra dentro de la línea transversal de tecnologías de la información y comunicación (TICS), y en cuanto a las líneas de investigación de la carrera de software, se encuentra en los programas de ingeniería de software e inteligencia artificial.

1.5 **Objetivos**

1.5.1 *Objetivo general*

Desarrollar un servicio web para la clasificación de noticias falsas, aplicando técnicas de aprendizaje automático.

1.5.2 *Objetivos específicos*

- Analizar las principales técnicas que permiten la clasificación de textos de noticias falsas.
- Seleccionar la técnica con mejor desempeño para la clasificación de texto de noticias falsas.
- Desarrollar el servicio web adecuado para la integración con modelos de aprendizaje automático.
- Evaluar los modelos entrenados en función de la puntuación F1.
- Evaluar el comportamiento temporal del servicio web desarrollado mediante el uso de la norma ISO/IEC 25010.

CAPÍTULO II

2 MARCO TEÓRICO

En este apartado se revisan fuentes bibliográficas relacionadas a los temas del presente proyecto, con la finalidad de fundamentar teóricamente, los temas a tratar se encuentran divididos en seis secciones, estas son: Aprendizaje automático, Servicio web, metodología de minería de datos CRISP-DM, metodología ágil de desarrollo SCRUM, métodos de evaluación, y trabajos relacionados.

2.1 Aprendizaje automático

Según Mitchell (1997), define al aprendizaje automático como un subconjunto de la Inteligencia Artificial (IA), este construye un modelo matemático basado en datos de muestra, estos se encuentran divididos en un conjunto de entrenamiento. Y mediante la aplicación de algoritmos, permite realizar predicciones y aprender patrones que pueden presentarse en los datos, posibilitando a realizar tareas sin la necesidad de tener una programación explícita.

2.1.1 Tipos de aprendizaje

Banda (2014), menciona los diferentes tipos de aprendizaje, los cuales se muestran a continuación:

2.1.1.1 *Aprendizaje supervisado*

En este tipo de aprendizaje, los algoritmos usan datos ya etiquetados, categorizados u organizados con anterioridad, con el fin de especificar cómo será la categorización de la nueva información. Por este motivo, se requiere de una persona que conozca del entorno, la cual está encargada de configurar el conjunto de vectores descriptivos que representan el conocimiento, de este modo la persona involucrada puede proveer a la red neuronal la respuesta para cada vector utilizado en el entrenamiento (Rouhiainen, 2018).

2.1.1.2 *Aprendizaje por refuerzo*

El tipo de aprendizaje por refuerzo, tiene una interacción continua con el entorno, este es un aprendizaje del mapeo entrada-salida. El proceso de refuerzo permite una interacción con el entorno para desarrollar habilidades, permitiendo desempeñar una tarea basándose en los resultados de estas interacciones, los resultados pueden considerarse como “un refuerzo positivo” cuando se realiza un acierto (Rouhiainen, 2018).

2.1.1.3 *Aprendizaje no supervisado*

El aprendizaje no supervisado, también llamado auto-organizativo, se encuentra la ausencia de una persona y hace uso de una regla de aprendizaje competitivo. Este tipo de aprendizaje trabaja

con datos no etiquetados, es decir, no se tiene un conocimiento a priori del tipo de datos. Con la implementación, se intenta encontrar una forma de reconocer las similitudes entre los datos para su posterior agrupación, en base a las características (Guardiola, 2020).

2.1.2 Técnicas de aprendizaje automático para la clasificación de texto

Existen diferentes tipos de técnicas para distintos campos de aplicación, sin embargo, algunos son usados para la clasificación de texto, por tal motivo, en base a la revisión de investigaciones de diferentes autores, se han tomado algunas de las técnicas, las cuales se especifican a continuación.

2.1.2.1 Naive Bayes

Esta técnica se basa en el teorema de probabilidad posterior condicionada de Bayes, utilizando la información conocida como variables explicativas denominadas priors, para la clasificación de las observaciones (Pranckevičius y Marcinkevičius, 2017). De este modo, el clasificador Bayesiano asume la presencia o ausencia de características, asignando una probabilidad, y define una variable en función a la relación existente.

En los problemas de clasificación se tiene una variable clase C y el conjunto de variables predictoras, o atributos $\{a_1, a_2, a_3, \dots, a_n\}$. Para el conjunto de variables se obtiene una red Bayesiana B , que define la distribución de probabilidad conjunta $P(C, a_1, a_2, \dots, a_n)$. De este modo, utilizando el conjunto de atributos, el clasificador basado en B retorna una etiqueta, maximizando la probabilidad a posteriori $P(C=c/a_1, a_2, \dots, a_n)$. La clase C toma “ m ” posibles valores c_1, c_2, \dots, c_m (Jaimes, 2015).

El proceso de clasificación, asumiendo que los atributos son condicionalmente independientes, la probabilidad de que una observación pertenezca a la clase “ C ” está dada por.

$$P(C = c/a_1, a_2, \dots, a_n) = \frac{P(C = c)P(a_1/C = c)P(a_2/C = c) \dots P(a_n/C = c)}{P(a_1, a_2, \dots, a_n)}$$

La ecuación factorizada es:

$$P(C = c/a_1, a_2, \dots, a_n) = \frac{P(C = c) \prod_{i=1}^n P(a_i/C = c)}{P(a_1, a_2, \dots, a_n)}$$

Esta técnica es ampliamente utilizada para la clasificación de textos debido a su simplicidad, aprendizaje rápido, a un desempeño razonable con el uso de grandes cantidades de datos y la capacidad de aprender de forma incremental.

2.1.2.2 Regresión logística

La regresión logística en inglés conocido como logistic regression, es un algoritmo de clasificación que se utiliza para conocer la probabilidad de, que una variable pertenezca a una categoría, en este tipo de técnica la variable es binaria y tiene datos codificados como 0 y 1. A pesar de que es utilizada para la clasificación binaria, existen librerías como scikit-learn, que permite utilizar el algoritmo para la clasificación de múltiples clases, mediante el uso de la estrategia One vs Rest (Guardiola González, 2020).

La estructura de la neurona en la regresión se representa en la siguiente **Ilustración 1-2**.

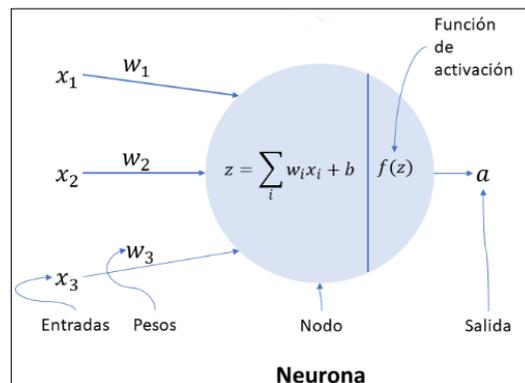


Ilustración 1-2. Neurona de la regresión logística

Fuente: Ponce, 2021

Los valores de X , corresponden a los atributos del problema (Entradas), cada uno se multiplica por su respectivo peso W , el valor resultante de la suma ponderada es representado por Z . La regresión está compuesta de dos partes, la combinación lineal y la aplicación de la función logística, de forma matemática la función es representada como (Martin y Jurafsky, 2021).:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

La función logística se encuentra acotada entre 0 y 1, representándose los resultados como probabilidades, el límite de decisión de un resultado es 0.5, es decir, si la probabilidad es menor, corresponde a la clase 0, caso contrario, será de la clase 1. Este límite se encuentra representado a continuación (Regresión Logística para Clasificación - IArtificial.net, 2019):

$$desición(x) = \begin{cases} 1 & \text{si } P(y = 1|x) > 0.5 \\ 0 & \text{de lo contrario} \end{cases}$$

2.1.2.3 Máquinas de soporte vectorial

La máquina de soporte vectorial, en inglés Support Vector Machine (SVM), es un algoritmo de aprendizaje supervisado que alcanza altos grados de clasificación correcta en diferentes tipos de

aplicaciones. Funciona como un clasificador lineal que permite separar los datos en dos clases utilizando diversos hiperplanos. Los vectores son considerados un subconjunto de los datos más próximos a las clases que se definen en la posición del hiperplano ideal. Si las clases no son separables linealmente, se utiliza la función kernel para proyectar los datos en un espacio dimensional superior, con el propósito de hacer que los puntos de los datos sean linealmente separables (Pranckevičius y Marcinkevičius, 2017).

Entre las funciones kernel utilizadas son: lineal, polinomio, base radial y sigmoideal, varios autores han señalado que el kernel lineal tiene un mejor desempeño que las demás. (Wanjun y Xiaoguang, 2010). La representación matemática es la siguiente:

$$K(X_i, X_j) = \left\{ \begin{array}{l} X_i \cdot X_j \text{ **Lineal**} \\ (yX_i \cdot X_j + C)^d \text{ **Polinomial**} \\ \exp(-\gamma |X_i - X_j|^2) \text{ **RBF**} \\ \tanh(yX_i \cdot X_j + C) \text{ **Sigmoidal**} \end{array} \right\}$$

Donde X_i son las variables predictoras del conjunto no vacío X , los Y_i se denominan objetivos, d es el grado del polinomio, y C corresponde a una constante.

Según Chapelle et al. (2002), señala que las dos principales ventajas de utilizar esta técnica para la clasificación supervisada son:

- El desempeño de SVM es bueno cuando se tiene un gran número de características
- El SVM ayuda mucho cuando se tiene pocos ejemplos de entrenamiento en tareas de clasificación de múltiples clases

2.1.2.4 *K-vecino cercano*

El clasificador K -vecino cercano, conocido en inglés como K -nearest neighbor (KNN), es un método de aprendizaje perezoso que se basa en ejemplos, consiste en que, cuando una nueva instancia necesita ser clasificada, se le compara con los ejemplos existentes usando una métrica de distancia, los más cercanos son utilizados para asignarle la clase a la nueva instancia (Introduction to Information Retrieval, 2009).

Este algoritmo, se encuentra basado en el espacio vectorial, que representa los documentos en forma de vector con la frecuencia de términos existentes. Una vez generado el vector de características para el texto desconocido, se procura por todos los ejemplos de entrenamiento comparar la similitud entre sus vectores de características, para encontrar los ejemplos más cercanos, y el documento desconocido es asignado a los k vecinos más próximos con un valor de clasificación mayor (Wanjun y Xiaoguang, 2010).

Las distancias entre las clases se calculan mediante las siguientes fórmulas matemáticas:

- Distancia euclidiana $d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$
- Distancia Manhattan $d(p, q) = \sum_i |p_i - q_i|$

Donde p_i y q_i son puntos correspondientes a los conjuntos a clasificar y entrenamiento respectivamente.

La regla de decisión de KNN matemáticamente se representa por:

$$\begin{aligned} f(x) &= \arg \max S(x, C_j) \\ &= \sum_{d_i \in KNN} \text{sim}(x, d_i) y(d_i, C_j) \end{aligned}$$

Donde S es la distancia entre x (documento del conjunto de prueba) y C_j (categoría del conjunto de entrenamiento), y $f(x)$ es una etiqueta para el documento del conjunto de prueba (Kowsari et al., 2019).

2.1.2.5 Árboles de decisión

El árbol de decisión, en inglés decision tree (DT), representa una estructura en árbol donde los atributos a ser probados son representados por cada nodo, las ramas representan la salida de prueba y los nodos finales (hojas) representa la clasificación. Es una técnica de análisis, que permite predecir la asignación de muestras a grupos predefinidos en función de variables predictivas, estos árboles a diferencia de las redes neuronales son fácilmente interpretables, sin embargo, se debe tener en cuenta que puede tender al sobre entrenamiento, la técnica es utilizada cuando la variable objetivo es categórica (Pranckevičius y Marcinkevičius, 2017).

Este algoritmo posee 2 fases principales, en la primera fase denominada crecimiento del árbol, donde se inicia con todo el conjunto de los datos como nodos raíz, estos datos serán divididos en subconjuntos utilizando algún criterio de división. Para la segunda fase, llamada la etapa de podada del árbol, todo el árbol se poda para prevenir el exceso de ajuste denominado en inglés overfitting, sobre los datos de entrenamiento (Guardiola, 2020).

Algunos de los algoritmos utilizados para construir árboles de decisión son, ID3, C4.5, SPRINT, SLIQ y PUBLIC, que utilizan medidas de decisión para la clasificación. Las medidas más utilizadas son: índice de Gini y ganancia de información. De manera matemática se representan a continuación (Rokach y Maimon, 2005) .

$$Gini(y, S) = 1 - \sum_{c_j \in \text{dom}(y)} \left(\frac{\sigma_{y=c_j} S}{S} \right)^2$$

Ganancia de información(a_i, S)

$$= Entropía(y, S) - \sum_{v_{i,j} \in \text{dom}(a_i)} \frac{|\sigma_{a_i=v_{i,j}} S|}{|S|} \cdot Entropía(y, \sigma_{a_i=v_{i,j}} S)$$

2.1.2.6 Redes neuronales

Las redes neuronales artificiales, conocidas en inglés como Artificial Neural Network (ANN) nacen como un esfuerzo conjunto de profesionales en diferentes campos como: la ingeniería, filosofía, fisiología, psicología, entre otras. Frank Rosenblatt, fue el creador del primer algoritmo que representó una red neuronal simple en 1958 y a pesar de que se plantearon estos fundamentos teóricos en el siglo pasado, ha tenido las aplicaciones prácticas en estos últimos años (Basogain, 1998).

Larranaga, et al., (2022), mencionan que las ANN tratan de imitar los siguientes conceptos claves de las redes neuronales biológicas.

- **Procesamiento paralelo**
Se deriva de que miles de millones de neuronas que intervienen para realizar una tarea.
- **Memoria distribuida**
En las redes biológicas existe una redundancia en el almacenamiento, con el fin de evitar pérdida de información.
- **Adaptabilidad al entorno**
Aprende de la experiencia y generar conceptos partiendo de casos particulares.

El modelo estándar según los principios de Rumelhart, et al. (1986), la *i*ésima neurona artificial que se representa en la **Figura 2-2** y está compuesta por:

- Un conjunto de entradas x_j , y pesos sinápticos w_{ij} , con $j = 1, 2, \dots, n$.
- Una regla de propagación h_i definida a partir del conjunto de entrada y pesos, representada como $h_i(x_1, \dots, x_n, w_{i_1}, \dots, w_{i_n})$

Comúnmente se hace uso de combinaciones lineales de las entradas y los pesos, además se suele añadir al conjunto de pesos un parámetro Θ_i , denominado umbral, esta resta al potencial post-sináptico, donde se obtiene:

$$h_i(x_1, \dots, x_n, w_{i_1}, \dots, w_{i_n}) = \sum_{j=1}^n w_{ij}x_j - \theta_i$$

- Una función de activación, representa la salida de la neurona y su estado de activación, denotándola como y_i se tiene $y_i = f_i(h_i) = f_i(\sum_{j=0}^n w_{ij}x_j)$

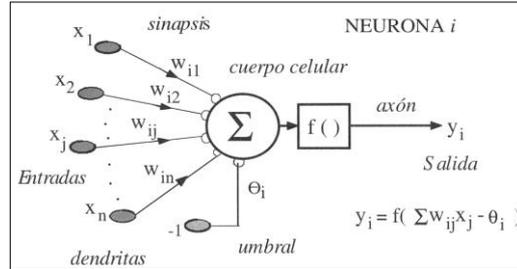


Ilustración 2-2 Modelo de neurona artificial estándar

Fuente: Larranaga, et al., 2022

En las redes neuronales se pueden distinguir 3 tipos de capa como se muestra en la **Ilustración 3-2**.

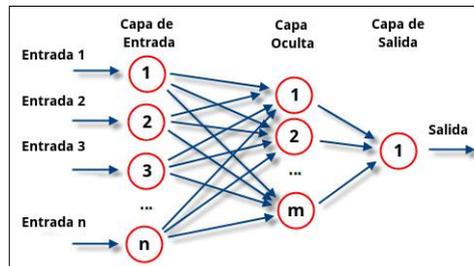


Ilustración 3-2 Arquitectura neuronal

Fuente (Qué son las redes neuronales y sus funciones, 2019)

Larranaga et al., (2022) mencionan las siguientes capas de una red neuronal:

- La capa de entrada se compone por las neuronas que reciben los datos.
- La capa de salida es la encargada de proporcionar la respuesta de la red neuronal.
- La capa oculta no tiene conexiones directas con el entorno, se considera como una probabilidad de procesamiento neuronal del 50%.

A. Redes neuronales profundas

La estructura se basa en una red neuronal básica, es decir, posee 3 capas: entrada, oculta y salida. La diferencia sustancial es que, en la capa oculta, internamente se tienen 2 o más capas. Todas las neuronas de la primera capa se conectan a todas las neuronas de la primera capa oculta, estas a su vez se conectan con las neuronas de la segunda capa oculta, y así sucesivamente se repiten las

conexiones en las n capas ocultas, hasta conectarse con las neuronas de la capa de salida como se muestra en la **Ilustración 4-2**. Esta última capa se encarga de hacer regresión o clasificación. Dependiendo de la activación y parámetros, es importante tener en cuenta que el número de salidas es igual al número de clases para multiclase, o una salida para clasificación binaria (Larranaga et al., 2022).

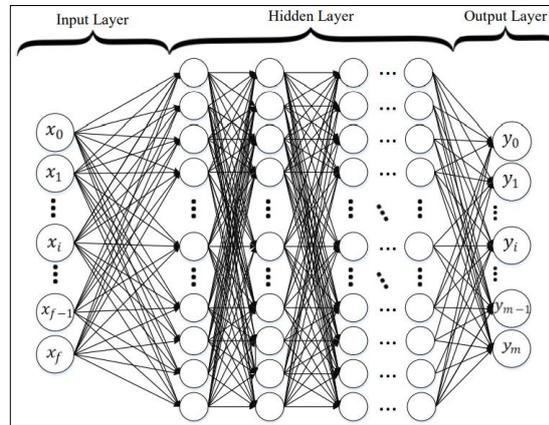


Ilustración 4-2 Red neuronal profunda completamente conectada

Fuente (Kowsari et al., 2019)

Este es un modelo entrenado discriminativo, que utiliza un algoritmo de propagación hacia atrás, en inglés Back-propagation, usando la función sigmoideal, y una función de activación ReLU. En cambio, si se tiene una clasificación multiclase, la función debería ser Softmax, estas funciones se muestran a continuación (Kowsari et al., 2019).

- Sigmoidal: $f(x) = \frac{1}{1+e^{-x}} \in (0,1)$, donde x representa las entradas.
- ReLU: $f(x) = \max(0, x)$
- Softmax: $\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$, donde k corresponde a un vector dimensional y z los valores.

B. Redes neuronales recurrentes

Las redes neuronales recurrentes, en inglés Recurrent Neural Network (RNN), tienen entradas de secuencia variable, y están compuestas de una capa oculta recurrente, la que se activa dependiendo del momento anterior. Por tal motivo se considera que este tipo de redes neuronales es una de las mejores opciones para la información textual (Bahad et al., 2019). A una RNN se puede formalizar como una secuencia de entrada (x_1, x_2, \dots, x_n) , para cada marca de tiempo, el modelo actualiza el estado de la capa oculta (h_1, h_2, \dots, h_n) y salida (y_1, y_2, \dots, y_n) . En la **Ilustración 5-2** se representa la estructura básica de una RNN(Bahad et al., 2019).

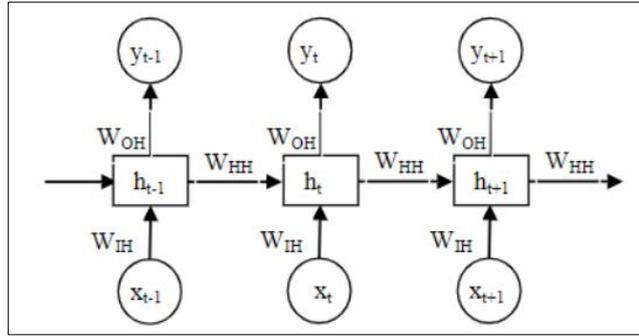


Ilustración 5-2 Estructura básica de una red neuronal recurrente

Fuente (Bahad et al., 2019)

Los elementos presentes en la **Ilustración 5-2**, los vectores x_t y y_t representan las entradas y salidas. Las matrices W_{IH} , W_{HH} y W_{OH} corresponden a los pesos de cada vector de las capas. Este tipo de redes neuronales hacen uso de propagación hacia atrás para realizar el entrenamiento, y su comportamiento puede ser descrito como un sistema dinámico mediante un par de ecuaciones en función de las funciones de activación de la capa oculta (f_h) y salida (f_o), b_h y b_y corresponden a sesgos asociados al vector oculto y de salida (Bahad et al., 2019).

$$h_t = f_h(W_{IH}x_t + W_{hH}h_{t-1} + b_H)$$

$$y_t = f_o(W_{OH}h_t + b_y)$$

Mientras se realiza el entrenamiento de la red neuronal, el mínimo de la función error se establece mediante pasos en la dirección de la derivada del error negativo con respecto al peso de las redes. Un inconveniente con estas redes neuronales es el problema de gradiente de fuga en las capas inferiores de la red profunda.

Como posibles soluciones al error mencionado anteriormente se plantea 3 posibles soluciones, función de activación Rectified Linear Unit (ReLU). Usar el algoritmo RMSProp para optimizar el aprendizaje adaptativo, o usar una arquitectura como la red neuronal Long Short Term Memory (LSTM) (Bahad et al., 2019).

C. Long Short Term Memory

Es una red neuronal que aborda los problemas al preservar la dependencia a largo plazo de manera más efectiva que las RNN básicas. De este modo se logra superar el problema de gradiente de fuga. En la **Ilustración 6-2** se muestra la celda básica de un modelo LSTM, donde está presente el uso de múltiples puertas con el fin de regular la cantidad de información para cada nodo estado (Kowsari et al., 2019).

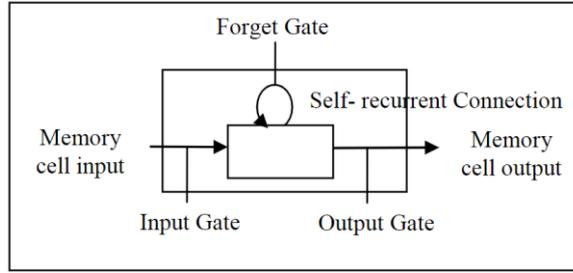


Ilustración 6-2 Estructura de una celda LSTM

Fuente (Bahad et al., 2019)

Las expresiones matemáticas de los elementos de una celda LSTM se muestran a continuación, donde x_t es el estado en el momento, σ denota una función de elemento sabio, b es un vector bias, W representa una matriz de pesos, los índices i, c, f, o corresponde a las entradas, celda de memoria, olvido, y salida, respectivamente.

Puerta de entrada: $i_t = \sigma(W_i[x_t, h_{t-1}] + b_i)$

- Celda de memoria: $\tilde{C}_t = \tanh(W_c[x_t, h_{t-1}] + b_c)$
- Activación de la puerta de olvido: $f_t = \sigma(W_f[x_t, h_{t-1}] + b_f)$
- Calcula el nuevo valor de la celda de memoria: $C_t = i_t * \tilde{C}_t + f_t C_{t-1}$
- Define el valor de la puerta de salida:

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o)$$

$$h_t = o_t \tanh(C_t)$$

De las fórmulas anteriores, b corresponde a un vector bias, W es una matriz de pesos, x_t representa la entrada a la celda de memoria en el tiempo t . Los índices i, c, f, o se refieren a las puertas de entrada, memoria de celda, olvido y salida respectivamente.

D. Redes neuronales convolucionales

A pesar de que se ha utilizado para el procesamiento de texto, el uso más común que se ha dado es para procesamiento de imágenes. La estructura que se muestra en la **Ilustración 7-2** representa una arquitectura para la clasificación de texto.

En la capa intermedia se encuentra la capa de pooling, que reduce el tamaño de salida, desde una capa a la siguiente en la red, pero manteniendo las características más importantes. Esta actividad permite reducir la complejidad computacional, que es una limitante en las redes neuronales profundas (Kowsari et al., 2019).

El principal problema que tiene la aplicación de este tipo de redes, para la clasificación de texto es el número de ‘canales’, cuando es aplicado para la clasificación de imágenes, comúnmente se tiene 3 canales (RGB), en cambio para texto resulta en una dimensionalidad alta (Kowsari et al., 2019).

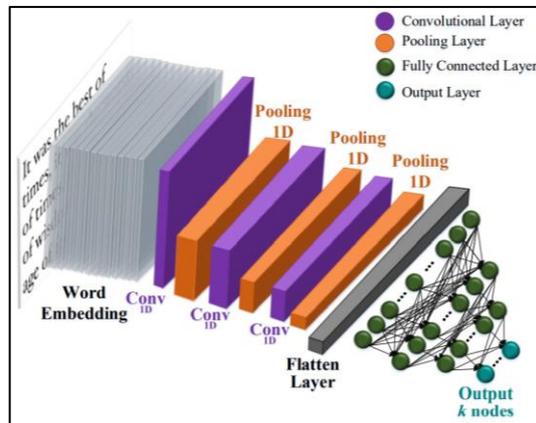


Ilustración 7-2 Arquitectura de red neuronal convolucional para la clasificación de texto

Fuente Kowsari et al., 2019

2.1.3 Aplicaciones del aprendizaje automático

La implementación del aprendizaje automático en la industria es múltiple, en aplicaciones de uso cotidiano se encuentran Twitter o Facebook en los algoritmos de recomendación, sin embargo, fuera del ámbito tecnológico también se encuentran presentes, algunas de estas áreas que Ramírez (2018) menciona, se describen a continuación.

2.1.3.1 Medicina

En la rama de medicina, la aplicación de aprendizaje automático tiene como objetivo la eficiencia de los equipos generadores de diagnósticos médicos, para evitar errores humanos en el análisis de datos y reducir costos en investigaciones.

Ejemplos de esta aplicación en la medicina son, IBM Watson y chatbots para la interacción con el paciente de tal manera que permita formular una hipótesis del estado de salud.

2.1.3.2 Educación

La aplicación en esta rama permitiría a sistemas, evaluar a cada estudiante mediante un plan de trabajo con las necesidades especificadas, de tal manera que el profesor pueda enfocarse en las deficiencias de los estudiantes, con el fin de mejorar la calidad en la educación.

2.1.3.3 *Construcción*

Algunas fábricas realizan actividades de riesgos o que les lleva un tiempo considerable para terminar de desarrollarlas, reduciendo la eficiencia, con la aplicación de aprendizaje automático en conjunto con la robótica, se llega a automatizar los procesos industriales, generando mayor rentabilidad y eficiencia.

2.1.3.4 *Robótica*

A nivel de la robótica el aprendizaje automático ha sido de gran impacto, un claro ejemplo es el robot Asimo que ha demostrado ser capaz de realizar actividades que una persona realiza y son consideradas complejas para las máquinas.

2.1.4 **Lenguajes para el aprendizaje automático**

Existen diferentes lenguajes de programación utilizados para la construcción de Inteligencia artificial y aprendizaje automático. Cada aplicación tiene sus propios requisitos y restricciones, por lo que un lenguaje puede ser mejor en el sentido que satisfacen los requisitos especificados. Entre los diferentes de lenguajes de programación existen algunos que han sido creados teniendo en cuenta satisfacer los requisitos de las aplicaciones de IA y aprendizaje automático (Rojas, 2020).

2.1.4.1 *Lenguaje Python*

El lenguaje Python, creado en la década de los noventa, es un lenguaje orientado a objetos que permite realizar desde aplicaciones de escritorio hasta servidores de red o desarrollo web (Sarkar et al., 2018).

La popularidad de este lenguaje se ha dado por las siguientes razones que Rojas (2020) señala :

- Número de bibliotecas
- La gratuidad del lenguaje, incluso para fines comerciales
- Simplicidad y curva de aprendizaje baja
- Desarrollo más rápido en menos líneas en comparación con otros lenguajes
- Cantidad de plataformas en las que se permite el desarrollo de Python

La relación que tiene el lenguaje Python con la ciencia de datos es debido al número elevado de librerías, y aunque el lenguaje es interpretado causando que el rendimiento sea menor a lenguajes de bajo nivel, se han desarrollado librerías como NumPy y SciPy sobre implementaciones de C y Fortran de capa inferior para realizar las operaciones rápidas y vectorizadas en matrices

multidimensionales. La librería más popular para aprendizaje automático de código abierto y accesible es scikit-learn (Mirjalili y Raschka, 2020).

2.1.4.2 *Lenguaje R*

Es un lenguaje de programación orientado al análisis estadístico y a la representación gráfica de los resultados obtenidos. Forma parte del proyecto GNU por tal razón la modificación y extensión de este lenguaje es de forma libre. R es multiplataforma y se encuentra disponible para los sistemas operativos Linux, Windows y Mac, la comunidad de este lenguaje es activa por lo que ha facilitado el desarrollo de nuevas funcionalidades y versiones actuales. Este lenguaje se basa en comandos, que se ejecutan y representa un flujo de trabajo para la realización de tareas. Las características de este lenguaje se muestran a continuación (Rojas, 2020).

1. Posibilidad de crear gráficos basados en LaTeX
2. Gran cantidad de herramientas estadísticas
3. Posibilidad de crear sus propias funciones
4. Integración con diferentes bases de datos
5. Uso matemático para sustituir MATLAB

El potencial de este lenguaje reside en las librerías que permiten funcionalidades para el análisis estadístico, manipulación de datos, visualización de gráficos y la disponibilidad de algoritmos de aprendizaje automático (Rojas, 2020).

2.1.4.3 *Lenguaje Matlab*

Es una plataforma optimizada para la resolución de problemas científicos y de ingeniería, se encuentra basado en matrices, la forma más natural de representar las expresiones matemáticas computacionales, los gráficos integrados permiten facilitar la visualización de los datos y obtención de la información (Rojas, 2020).

Según Rojas (2020), este lenguaje ha sido utilizado para el desarrollado de aplicaciones de mantenimiento predictivo, análisis de sensores, finanzas y electrónica de comunicaciones, facilita los siguientes procesos de aprendizaje automático mediante:

- Técnicas de procesamiento de señales y extracción de funciones avanzadas
- Ajuste de hiperparámetros y selección de funciones automáticas para la optimización de modelos
- Posibilidad de usar el mismo código para el escalamiento de procesamiento de big data y clústeres.
- Generación automática de código en lenguaje C/C++ para aplicaciones de alto rendimiento.

- Algoritmos de clasificación, regresión y agrupamiento para el aprendizaje supervisado y no supervisado.
- Ejecución más rápida en la mayoría de los cálculos estadísticos y aprendizaje automático.

Matlab provee funciones prediseñadas, caja de herramientas integrales y aplicaciones especializadas para la clasificación, regresión y agrupamiento. Dentro del kit de herramientas se proporcionan algoritmos de aprendizaje automático como, máquinas de soporte vectorial, árboles de decisión potenciados, k-vecinos cercanos, k-medias, modelos de Márkov (Rojas, 2020).

2.1.5 Librerías y Frameworks

2.1.5.1 *Keras*

Es una librería de alto nivel de código abierto escrita en Python, que proporciona una API de aprendizaje automático, además provee otras librerías de bajo nivel, que son utilizadas para el procesamiento y manipulación de tensores, es capaz de ejecutarse sobre la plataforma de Tensorflow. Fue desarrollada enfocándose en permitir una experimentación rápida, de tal manera que permita plasmar las ideas en el menor tiempo posible (StackOverflow Contributors, 2022).

Según el sitio oficial, algunas características en las que se fundamenta Keras, son (Keras documentation, 2022):

- Simple: No simplista, permite que el desarrollador pueda concentrarse en el problema mediante la reducción de la carga cognitiva.
- Flexible: Tiene el principio de revelación progresiva de la complejidad, donde considera que los flujos de trabajo simples deben ser rápidos y fáciles. Por otro lado, los trabajos avanzados son posibles a través de un camino recorrido, en el cual se ha ido aprendiendo.
- Potente: el rendimiento proporcionado y la escalabilidad son sólidos, por tal motivo es utilizada por organizaciones como, La NASA, YouTube, Waymo.

2.1.5.2 *Scikit-learn*

Es una librería de Python, que integra una amplia variedad de algoritmos para aprendizaje automático de última generación, para problemas supervisados y no supervisados, permitiendo realizar tareas de clasificación, regresión, estimación de matriz de covarianza, preprocesamiento de datos, entre otras (Bisong, 2019).

Una de las ventajas que tiene es la facilidad de uso para nuevos usuarios, haciendo hincapié en el rendimiento, la documentación de algoritmos para la clasificación como, modelos lineales, máquinas de soporte vectorial, vecinos cercanos, procesos Gaussianos, Naive Bayes, Árboles de decisión, modelos de redes neuronales, entre otros, cada uno con algunas de sus variantes (Pedregosa et al., 2011).

2.1.5.3 *Tensorflow*

Según el sitio oficial, Tensorflow Federated(TFF), es un framework de código abierto para el aprendizaje automático. Este fue desarrollado por Google y potencia varios de sus servicios, como, Google cloud speech, Google photos y Google Search. Su aplicación se encuentra en una variedad de proyectos que hacen uso de tareas de aprendizaje automático como, la clasificación de imágenes, procesamiento de lenguaje natural, sistemas de recomendación, y demás (TensorFlow, 2022).

Géron (2019), menciona algunas funciones que ofrece Tensorflow, estas se enlistan a continuación:

- Un Kernel similar a NumPy, pero con soporte de GPU.
- Compilador Justo-a-Tiempo para optimizar los cálculos para velocidad y uso de memoria
- Compatibilidad de los modelos entrenados entre plataformas
- Una de las más importantes funciones es tf.keras, además tiene operaciones para el pre-procesamiento y carga de datos, imágenes, señales y más.

2.2 **Servicios Web**

Los servicios web, son un conjunto de aplicaciones auto-contenidas, auto-descriptivas y modulares, que incluyen tecnologías como XML, SOAP, REST, WSDL, UDDI. Estos servicios se pueden encontrar publicados en la web, desde la cual se realizan sus invocaciones. Estos servicios permiten intercambiar datos entre las máquinas conectadas a la red (Gutierrez, 2016).

A los proveedores de servicios se les conoce como servidores, y estos “exponen sus servicios” para que las máquinas denominadas clientes, consuma dicho servicio. Dado que la comunicación utilizada es estándar, permite mejorar la interoperabilidad y extensibilidad de las aplicaciones (Gutierrez, 2016).

2.2.1 **API REST**

Es una interfaz de programación de aplicaciones que permite conectar varios sistemas basados en el protocolo HTTP, posibilitando obtener y generar datos y operaciones, las respuestas devueltas son en formato específicos como XML y JSON. Los métodos usados son GET, POST, PUT y DELETE, siendo una alternativa a los servicios SOAP que tiene más complejidad, haciendo más fácil su desarrollo (¿Qué es REST?, 2018).

Una API es considerada REST si cumple con los siguientes criterios (¿Qué es una API de REST?, 2022):

- Arquitectura cliente-servidor
- Comunicación entre el cliente y el servidor sin estado
- Datos que almacenan en caché, optimizando las respuestas
- Sistemas en capas que organizan en jerarquías cada uno de los servidores

2.3 Metodología CRISP-DM

La metodología CRISP-DM por sus siglas en inglés Cross Industry Standard Process for Data Mining, es descrita en términos de un modelo de proceso jerárquico que se encuentra constituido por tareas descritas en cuatro niveles de abstracción: fase, tareas genéricas, tareas especializadas e instancia de proceso (Cortina, 2015).

En el documento CRISP-DM 1.0 de Chapman et al. (2000), define a CRISP-DM como una metodología que incluye un modelo y una guía que se estructura en 6 fases siendo algunas de estas bidireccionales, por lo que la ordenación de estas fases no necesariamente se encuentran desde la primera hasta la última, a continuación se describen cada fase y también se encuentra representadas en la **Ilustración 8-2**. Sin embargo, todas las fases de esta metodología no son consideradas para el desarrollo del presente trabajo, sino que se realiza una adaptación con las fases que requeridas.

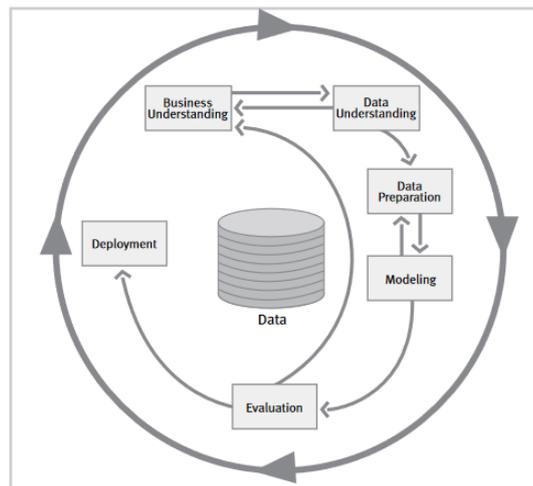


Ilustración 8-2 Fases de la metodología CRISP-DM

Fuente (Chapman et al., 2000)

2.3.1 Comprensión del negocio

En esta fase inicial se enfoca en entender los objetivos del proyecto y los requerimientos desde una perspectiva de negocio, De esta manera se convierte este conocimiento hacia la definición de un problema de minería de datos y se diseña un plan preliminar para alcanzar los objetivos.

2.3.2 **Entendiendo los datos**

Esta fase empieza con la recolección inicial de los datos y se procede como actividades que permiten la familiarización con los datos, identificar problemas de calidad de los datos, descubrir primeras perspectivas en los datos y/o detectar subconjuntos de interés para la formulación de hipótesis.

2.3.3 **Preparación de los datos**

En esta fase se cubren las actividades para construir el conjunto de datos final, a partir de los datos brutos iniciales. Posiblemente estas actividades de preparación de datos se deban repetir varias veces sin seguir un orden específico, entre las tareas está la selección de tablas, registros y atributos, incluyen también la transformación y limpieza de datos.

2.3.4 **Modelado**

En la fase de modelado, varias técnicas son seleccionadas y aplicadas, también se calibran sus parámetros con el objetivo de alcanzar valores óptimos. Cada técnica tiene requisitos específicos, por lo que puede ser necesario regresar a la fase anterior.

2.3.5 **Evaluación**

En esta fase, ya tenemos uno o más modelos construidos, que parecen tener una alta calidad desde la perspectiva de análisis de datos. Sin embargo, es necesario realizar un análisis profundo para garantizar el cumplimiento de los objetivos. Al final de esta fase se debe llegar a una decisión sobre el uso de los resultados.

2.3.6 **Despliegue**

Dependiendo de los requisitos la fase de implementación puede ir desde tareas simples como generar un informe hasta complejas como implementar un proceso de extracción de datos. En muchas ocasiones, el que realiza la actividad de despliegue es el cliente, por tal motivo es importante que conozca las acciones que deberá a llevar a cabo.

2.4 **Metodología SCRUM**

Es una metodología ligera, iterativa e incremental, que genera valor mediante soluciones adaptativas, permitiendo resolver problemas complejos de software. Esta emplea varios procesos, técnicas y métodos. Basada en Sprints, esta metodología enfoca sus procesos de manera general en 3 etapas (Scrum Guide | Scrum Guides, 2020).

Los sprints, son eventos de duración de, entre 1 y 4 semanas. El comienzo de uno empieza inmediatamente cuando el anterior concluye. Todo el trabajo como, la planificación, reuniones diarias, scrums diarios, revisión de sprints, y retrospectiva, se realizan dentro del mismo sprint.

Para garantizar previsibilidad, adaptación del progreso hacia el objetivo, es recomendable que el tiempo del sprint sea en un tiempo corto, por tal motivo a un sprint se considera como un proyecto corto. Un sprint es cancelado cuando el Sprint Goal se vuelve obsoleto (Cohen, 2013).

Según Gabriel (2015), en su artículo describe las siguientes actividades que se llevan a cabo en Scrum:

- Planificación de la iteración, constituida por 2 partes, ambas se realizan con un tiempo promedio máximo de entre 1 y 2 horas.
- Ejecución de la iteración, cada iteración tiene que proporcionar un incremento de producto que sea entregable, de tal manera que, cuando el cliente solicite, solo sea necesario un esfuerzo mínimo para que el producto esté disponible.
- Reuniones diarias de sincronización, facilita la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad.
- Demostración de requisitos completados, es reunión informal donde el equipo presenta al cliente los requisitos completados en la iteración.
- Retrospectiva, el equipo analiza cómo ha sido su manera de trabajar durante la iteración.

2.4.1 Roles Scrum

Scrum define 3 tipos de roles: Product Owner, Scrum Master, y Team, cada uno de estos roles se describen a continuación (Khalane y Tanner, 2013).

- **Product Owner:** Es el responsable en definir, priorizar y cambiar las características del producto, además de especificar las fechas de lanzamiento.
- **Scrum Master:** Considerado como el líder del grupo, sus funciones son: facilitar la productividad, colaboración entre los roles, la resolución de impedimentos también es el encargado de coordinar todo proceso del desarrollo.
- **Team:** Está compuesto por un grupo de entre 5 y 9 miembros, son los encargados de realizar entregas de los objetivos del proyecto.

2.4.2 Eventos scrum

El sprint es el contenedor de todos los otros eventos, cada evento es una oportunidad para inspeccionar y adaptar los artefactos, de este modo se permite una transparencia requerida para alcanzar los objetivos planteados (Scrum Guide | Scrum Guides, 2020).

- **Planificación del sprint**

Se realiza la planificación de cada sprint, el equipo Scrum interviene en esta actividad, sin embargo, pueden estar presentes invitados para el asesoramiento.

- **Reunión diaria**

Se inspecciona el progreso en el alcance del objetivo del sprint. La duración sprint está entre 15 a 20 minutos, cualquiera puede asistir, sin embargo, solo los miembros pueden hablar.

- **Revisión del sprint**

Se realiza la inspección del resultado del sprint presentado por el equipo. También se analiza y determina adaptaciones para los siguientes sprint.

- **Retrospectiva**

Tiene como objetivo aumentar la calidad y mejorar la eficacia mediante la planificación de tareas. La duración está entre 15 y 30 minutos. Los participantes son todos los roles de SCRUM, además de posibles invitados.

2.4.3 Artefactos Scrum

Los artefactos SCRUM son denominados a los elementos físicos que producen un resultado de la aplicación de SCRUM, entre los 3 principales artefactos se detallan en la tabla 4 (Roche, 2022).

- **Product Backlog**

Es un inventario que contiene todo tipo de trabajo propuesto, requerimientos, trabajo de investigación, bugs, funcionalidades, historias de usuarios, casos de uso, tareas y dependencias. Se considera como la fuente principal de información del producto. Es el product owner quien gestiona este artefacto

- **Sprint Backlog**

Es el plan de trabajo que los desarrolladores siguen en cada sprint, está compuesto por tareas técnicas que permite conseguir incrementos del software. Toda actividad seleccionada por el grupo de desarrollo para el siguiente sprint, pasa a Sprint Backlog. Como objetivo de este artefacto es visualizar el trabajo a realizar y es gestionado por el equipo de desarrollo.

- **Incremento**

Es el resultado del sprint, considerado como la suma de todas las tareas y elementos desarrollados durante el sprint.

Para el incremento del producto hay que considerar que:

- Cada incremento debe ser probado para asegurar su funcionamiento de manera integral con incrementos anteriores.
- Al finalizar el sprint, el nuevo incremento debe estar terminado.
- Durante el sprint se puede presentar varios incrementos.
- Un incremento se puede poner en producción antes de concluir el sprint.
- El incremento es presentado en la revisión del sprint.

• **Otros artefactos**

Otros artefactos considerados destacables de SCRUM son:

- Definición de hecho, consiste en un documento donde se establece una serie de criterios comunes para especificar cuando el ítem ha sido terminado.
- Definición de listo, es un documento que define si un requerimiento se considera listo para incluirse en un Sprint Planning.
- Cuadro de incendio, representa un gráfico de trabajo pendiente y permite extrapolar si el equipo podrá completar el trabajo en el tiempo definido.

2.5 Métodos de evaluación

En esta sección se describen los métodos de evaluación utilizados para medir el rendimiento, tanto del modelo como del servicio web, utilizando F1 score y Eficiencia desempeño, respectivamente.

2.5.1 F1 score

Esta métrica también conocida como Medida F1 o Puntuación balanceada F, puede ser interpretada como un medio armónico entre la precisión y recall los intervalos de valores se encuentran entre, 1 y 0 siendo el mejor y el peor valor respectivamente (sklearn.metrics.f1_score, 2022). De forma matemática se describe como:

$$F1 = 2 * \frac{Precisión * recall}{Precisión + recall}$$

Esta métrica es utilizada en los modelos de Machine Learning que realizan clasificación, por este motivo será considerado para la evaluación del modelo a implementar.

2.5.2 Eficiencia de desempeño

La eficiencia de desempeño como característica de calidad de software, ha tenido un valor importante en la aceptación de software, esta forma parte de la ISO 25000, siendo una de las ocho características definidas como se ve en la **Ilustración 9.3**.



Ilustración 9-3. Características de la ISO 25000

Fuente ISO 25000, 2016

Según la ISO 25000(2016) menciona que esta característica representa el rendimiento en relación con la cantidad de recursos utilizados en determinadas condiciones. Esta está dividida en las siguientes subcaracterísticas:

- **Comportamiento temporal**

Tiempos de respuesta y procesamiento, así como el rendimiento del sistema al realizar sus funciones bajo ciertas condiciones, en relación con un valor estándar establecido.

- **Utilización de recursos**

Es el número y tipos de recursos utilizados en el desempeño de la función del software bajo condiciones específicas.

- **Capacidad**

Es el grado en que los límites máximos de un producto de software o parámetro del sistema cumplen con los requisitos.

El comportamiento temporal implica un factor importante desde el desarrollo del sistema, debido a que, si se presenta un mal desempeño del sistema, este repercute en la satisfacción del cliente, disminuyendo la calidad del software (González et al., 2021).

2.6 Trabajos relacionados

Se han realizado varios estudios e investigaciones donde se aplican diferentes técnicas de aprendizaje automático para la detección y clasificación de noticias falsas.

En la facultad de ingeniería de la universidad distrital “Francisco José de Caldas” de Bogotá, DC, Colombia, se ha realizado la implementación de un modelo para la detección de noticias falsas, utilizando técnicas de aprendizaje automático utilizando el lenguaje Python, en este estudio se

aplican el algoritmo Naive Bayes y el paquete Scikit-Learn para su implementación, las fases del proceso seguido son; Preprocesamiento, Ordenado y transformación de los datos, reordenado de los datos, creación del conjunto de prueba y entrenamiento, tokenización y aplicación del aprendizaje automático. Finalmente, como resultado se obtiene un 89.3% de clasificación correcta, siendo inferior al resultado de otros modelos como SVM y DNN. (Agudelo et al., 2018)

Por otro lado en la investigación de Bahad, Saxena y Kamal (2019) presentada en la conferencia internacional sobre tendencias recientes en computación avanzada, se realiza un análisis de la aplicación de redes neuronales LSTM bidireccionales, para la clasificación de noticias falsas, mediante el diseño, la implementación, como resultados obtenidos de dicho modelo en comparación con las redes neuronales convolucionales, y las redes neuronales Long Short Term Memory unidireccional tiene un rendimiento significativo.

En 2019, en la universidad de ICESI, Colombia, en el trabajo presentado por Prada (2019) para la clasificación de tweets escritos en español como verdaderos o falsos, se analizaron los algoritmos: árboles de decisión, Naive Bayes, Regresión logística. Además se tiene en cuenta la métrica F1 para seleccionar el que mejor se ajusta a los datos para la tarea de clasificación, luego del análisis los resultados y la comparación de los mismos, se escoge a la regresión logística con un resultado del 74.37%, siendo superior a SVM y Naive Bayes con puntuación 71.79% y 71.49 respectivamente (Prada, 2019).

CAPITULO III

3 MARCO METODOLÓGICO

En este capítulo se describen los métodos, técnicas de investigación, variables y métricas utilizadas en el desarrollo, también se realiza un análisis para determinar la factibilidad del proyecto. Además, en la práctica se realiza la aplicación de las fases de la metodología ágil de desarrollo SCRUM, para el desarrollo del servicio web, y para la implementación del modelo de aprendizaje automático, se utiliza la metodología de minería de datos CRISP-DM.

3.1 Tipo de estudio

El presente Trabajo de Integración Curricular es de tipo aplicativo, donde se emplea sapiencias de inteligencia artificial en el desarrollo de un modelo entrenado e ingeniería de software para el desarrollo del servicio web, integrándolos conjuntamente, con el fin de obtener un servicio web que permita la clasificación de noticias falsas en español.

3.2 Métodos y técnicas

Para la definición y entrenamiento del modelo de aprendizaje automático se aplica la metodología CRISP-DM, utilizada para minería de datos, que está compuesta de 6 fases, en cada una de estas se especifican las tareas a realizar. Por otro lado, para el servicio web la metodología de desarrollo utilizada es SCRUM, donde se especifican 3 macro fases, planeación, desarrollo y cierre, de la misma manera que la metodología anterior, en cada una de estas se especifican las tareas a realizar. En la **Tabla 1-3** se describen los métodos y técnicas utilizadas en el presente trabajo.

Tabla 1-3: Métodos y técnicas.

Objetivos	Métodos	Descripción	Técnicas	Fuente
Analizar las principales técnicas que permiten la clasificación de textos de noticias falsas.	Analítico	Con la finalidad de conocer cuales técnicas se encuentran disponibles para la clasificación de textos y cuales tienen mejor rendimiento para el conjunto de datos seleccionados.	<ul style="list-style-type: none"> • Revisión de documentos 	<ul style="list-style-type: none"> • Artículos científicos • Tesis • Libros
Seleccionar la técnica con mejor desempeño para la clasificación de texto de noticias falsas.	Deductivo	Para permitir reconocer en base a las investigaciones anteriores cuál es la técnica con mejor desempeño para la clasificación de textos.	<ul style="list-style-type: none"> • Observación • Revisión de documentación 	<ul style="list-style-type: none"> • Código fuente del modelo • Artículos científicos
Desarrollar el servicio web adecuado para la integración con modelos de aprendizaje automático.	Metodología SCRUM	Metodología ágil que permite el proceso de desarrollo para el servicio web a implementar.	<ul style="list-style-type: none"> • Revisión de documentación • Sprint Backlog • Product Backlog • Planificación • Desarrollo • Cierre 	<ul style="list-style-type: none"> • Documentación de requerimientos • Encuesta

<p>Evaluar los modelos entrenados en función de la puntuación F1</p>	<p>Metodología CRISP-DM Experimental</p>	<p>Metodología de minería de datos que provee las fases requeridas para el desarrollo del modelo entrenado.</p>	<ul style="list-style-type: none"> • Comprensión del negocio • Entendiendo los datos • Preparación de los datos • Modelado • Evaluación • Despliegue 	<ul style="list-style-type: none"> • Modelo propuesto • Conjunto de datos
<p>Evaluar el comportamiento temporal del servicio web desarrollado, mediante el uso de la norma ISO/IEC 25010.</p>	<p>Inductivo Experimental</p>	<p>Mediante la realización de pruebas y ponderando los resultados permite llegar a concluir cuál es el comportamiento temporal del servicio web.</p>	<ul style="list-style-type: none"> • Observación • ISO/IEC 25010 	<ul style="list-style-type: none"> • Servicio web desarrollado • Modelo propuesto

Realizado por: Jefferson Chariguamán, 2022

3.3 Determinación de la población y muestra

Con el fin de determinar la muestra, se toma como población a los habitantes del Ecuador, teniendo en cuenta que estos serían los usuarios potenciales del servicio web. La fórmula y los parámetros utilizados para el cálculo de la muestra se detallan a continuación:

Cálculo de la muestra:

$$n = \frac{N * Z^2 * P * Q}{e^2 * (N - 1) + Z^2 * P * Q}$$

Parámetros

N=18000000 (Población de Ecuador según estimaciones del INEC)

Error: 2%

Z:2,58 (Asume un nivel de confianza del 99%)

P=Q=50%

e=2%

n=4159

$$n = \frac{18000000 * (2,58)^2 * 0,5 * 0,5}{(0,2)^2 * (18000000 - 1) + (2,58)^2 * 0,5 * 0,5} = 4160$$

El valor total de la muestra está calculado tomando un error de 2%, con una población de 18000000 (habitantes de Ecuador), y un nivel de confianza del 99%, este valor es utilizado para determinar la cantidad de peticiones que se realizarán en el capítulo de resultados.

3.4 Estudio de factibilidad

En esta sección se determina la factibilidad de desarrollar el proyecto propuesto.

3.4.1 Factibilidad técnica

En el estudio de factibilidad se especifican los recursos hardware y software, también se incluye los recursos humanos para realizar el proyecto técnico.

En la **Tabla 2-3** se especifican los recursos hardware que son utilizados en el desarrollo del Trabajo de Integración Curricular.

Tabla 2-3: Recursos hardware.

Cantidad	Descripción
1	Laptop Dell, Intel i5, 8va generación, 8GB RAM, Windows 10 Pro.

1	Computadora de escritorio, Intel i5, 3ra generación, 8GB RAM, Windows 10 Pro, Nvidia GTX 970.
1	Impresora Epson L210

Realizado por: Jefferson Chariguamán, 2022

En la **Tabla 3-3** se observan los recursos de software utilizados en el desarrollo del Trabajo de Integración Curricular.

Tabla 3-3: Recursos Software.

Nombre	Descripción	Observaciones
Windows 10	Sistema operativo Windows 10 Pro.	Sistema operativo por defecto que permite el uso del computador.
Visual Studio Code	Editor de código con soporte para varios lenguajes de programación.	Editor de código utilizado para el desarrollo del servicio web.
Python	Lenguaje de programación para diferentes aplicaciones como machine learning.	Es el lenguaje para el desarrollo del modelo y del servicio web.
GitHub	Herramienta multiplataforma para el control de versiones.	
Google Colab	Plataforma para el desarrollo con lenguaje en Python, basado en cuadernos estilos Jupyter.	Permite el desarrollo que permite el desarrollo del modelo de machine learning, además provee de recursos hardware como CPU, GPU, TPU y RAM.

Realizado por: Jefferson Chariguamán, 2022

En la **Tabla 4-3** se encuentran los recursos humanos participantes a lo largo del desarrollo del Trabajo de Integración Curricular.

Tabla 4-3: Recursos humanos.

Nombre	Función	Participación
Jefferson Chariguamán	Desarrollador del trabajo de titulación	Desarrollador del proyecto

Realizado por: Jefferson Chariguamán, 2022

En base al análisis de los recursos técnicos de hardware, software y humanos requeridos, y considerando que los recursos actuales que se posee son suficientes para cubrir las necesidades planteadas, se considera factible técnicamente el desarrollo de este proyecto.

3.4.2 Factibilidad operativa

El presente proyecto tiene como objetivo disponer de un servicio web que permita a las personas verificar y/o tener cierta certeza sobre la veracidad de noticias falsas, además debido a que se encontrará como un servicio, la integración y consumo de los servicios permite la interoperabilidad con las diversas plataformas.

3.4.3 Factibilidad económica

Este estudio de factibilidad permite analizar los costos y beneficios del presente proyecto a desarrollar, de este modo se determina la factibilidad de su realización.

Los costos económicos de los recursos hardware, software, humanos y otros para el desarrollo del proyecto se representa en la **Tabla 5-3**.

Tabla 5-3: Costos del proyecto.

Recurso	Cantidad	Tipo de recurso	Fuente de financiamiento	Precio unitario	Precio total
Laptop	1	Hardware	Personal	\$800	\$800
Impresora	1	Hardware	Personal	\$ 300	\$300
Visual Studio	1	Software	Personal	\$ 0	\$0
Google Colab	4	Software	Personal	\$ 9.99	\$39,96
Suministros de oficina	1	Otros	Personal	\$ 50	\$50
Servidor	1	Hardware	ESPOCH		
Desarrollador por 4 meses	1	Humano	Personal	\$ 800	\$3200
<u>Internet y servicios básicos</u>	<u>5</u>	<u>Otros</u>	<u>Personal</u>	<u>\$ 50</u>	<u>\$250</u>
Transporte	5	Otros	Personal	\$ 80	\$400
Costo total del proyecto					\$5039,96

Realizado por: Jefferson Chariguamán, 2022

En base al análisis económico de los recursos necesarios para el desarrollo del proyecto, y teniendo en cuenta que el costo total del proyecto será asumido por el proponente, se considera factible para su realización.

3.5 Metodología de desarrollo CRISP-DM

En esta sección, se realiza la aplicación de cada una de las fases correspondientes a metodología CRISP-DM, con el propósito de obtener el modelo entrenado, encargado en realizar la clasificación de las noticias, este a su vez será integrado dentro del servicio web.

3.5.1 Comprensión del negocio

3.5.1.1 *Objetivo de negocio*

El objetivo del negocio es clasificar noticias en español, de esta forma determinar si es verdadera o falsa.

3.5.1.2 *Evaluar la situación*

El conjunto de datos recolectados es desde diferentes fuentes públicas, por tal motivo se tiene la libertad de uso, sin embargo, la limitación es la cantidad de datos disponibles, debido a que se requiere de noticias en español, y los conjuntos de datos existentes de gran tamaño se encuentra en el idioma inglés.

3.5.1.3 *Objetivo de la minería de datos*

Como principal objetivo de la minería de datos aplicado a este proyecto, es la aplicación de un modelo que permita la clasificación de noticias en español.

3.5.2 Comprensión de los datos

En esta fase se realiza las primeras exploraciones de los conjuntos de datos utilizados para el entrenamiento del modelo implementado.

3.5.2.1 *Recopilar los datos iniciales*

El conjunto de datos a utilizar para el entrenamiento del modelo es el resultado de la recolección de diferentes fuentes que se listan a continuación:

- Noticias falsas en español(Tretiakov, 2021)
- Fake and real news(Zules, 2019)
- Fake News Corpus Spanish(Chaumond, 2022)
- Train-00000(Ayala, 2022)

Además, de los conjuntos de datos mencionados, se agregaron nuevas noticias desde fuentes de información nacionales.

3.5.2.2 *Describir los datos*

Los conjuntos de datos recolectados se encuentran en formato csv, parquet, y otros fueron recolectados en la página web de Ecuador Chequea, donde se publican noticias clasificadas en diferentes tipos según su contenido.

En la **Tabla 6-3** se describen los conjuntos de datos utilizados para el entrenamiento y el desarrollo del proyecto.

Tabla 6-3: Descripción de los conjuntos de datos.

Nombre del conjunto de datos	Nombre del dato	Valor	Descripción
1000 fakes news	class	Booleano	Valor booleano true o false que determina la clase de la noticia
	Text	String	Conjunto de texto que corresponde a una noticia
SpanishFakeNews	texto	String	Texto correspondiente a una noticia
	clase	String	Texto que determina si la noticia es real o falsa, los valores están escritos en inglés, real y fake.
00001.parquet	Article_ID	Integer	Corresponde al número de artículo asignado según la fuente
	Author	String	El autor de la noticia publicada
	Content	String	Es el cuerpo de la noticia, tal como se extrajo desde la fuente
	Corrected_label	String	Es la clasificación verdadera de la noticia, es decir puede ser Fake o True
FakesNewsCorpusSpanish	Category	String	Categoría de la noticia, como Fake o True
	Source	String	Fuente de donde fue extraída la noticia
	Headline	String	Título de la noticia
	Text	String	Cuerpo de la noticia

Realizado por: Jefferson Chariguamán, 2022

3.5.2.3 Explorar los datos

Los conjuntos de datos seleccionados son de diferente longitud, cantidad de columnas presentes, y debido a que algunas fueron recolectadas desde fuentes no oficiales como redes sociales, la longitud no es adecuada para formar parte del conjunto de datos de entrenamiento, además, están presentes símbolos, *stop words* y enlaces que son considerados ruido, los cuales pueden afectar

el entrenamiento. Por tal motivo se requiere una limpieza de esos datos, tomando en cuenta las columnas correspondientes al texto de la noticia y a la clase que corresponde, para posteriormente realizar la extracción de características.

3.5.3 Preparación de los datos

Esta fase se realiza durante la recolección y exploración, implementando las siguientes acciones.

3.5.3.1 *Recolección de los conjuntos de datos desde diferentes fuentes*

Los conjuntos de datos correspondientes textos de noticias falsas, son recolectados desde fuentes como GitHub, Kaggle y Hugging Face, teniendo en cuenta que correspondan al lenguaje en español. Estos a su vez son el resultado de recolección mediante Scraping o simple recolección de sitios web de noticias.

3.5.3.2 *Descartar las noticias correspondientes a tweets*

Entre los conjuntos de datos recolectados, algunas de las noticias pertenecían a tweets, que, debido a su corta longitud, no son de utilidad, por lo que se descartan antes de la integración con los otros conjuntos de datos.

3.5.3.3 *Integración de los conjuntos de datos en uno solo*

Una vez realizado la exploración de los datos, se establecen cuáles son los campos útiles para el entrenamiento del modelo, por tal motivo estos fueron agregados al conjunto de datos final, para su posterior limpieza.

3.5.3.4 *Eliminar documentos con campos nulos o vacíos*

En el preprocesamiento de los datos se eliminan los elementos nulos o vacíos, se toma esta acción debido a que la cantidad de estos documentos en la fase de exploración no representa una cantidad que afecte al conjunto de datos de entrenamiento.

3.5.3.5 *Eliminación de ruido en los documentos*

Esta fase es de gran importancia, porque se realiza la eliminación de palabras que no aportan nada en la extracción de las características como signos de puntuación y enlaces. Otros procesos implicados en la limpieza de los datos son las técnicas de *Stemming* y *tokenización* de palabras, que mediante el uso de librerías para el procesamiento de lenguaje natural permiten la obtención de *embeddings*, dejando listo el conjunto de datos final para el entrenamiento del modelo.

3.5.4 Modelado

En esta fase se implementan varios de los algoritmos de aprendizaje automático para la clasificación de texto, estos son seleccionados tomando en cuenta los mejores rendimientos en estudios realizados previamente.

3.5.4.1 Escoger la técnica de modelado

Luego de realizar una revisión de la literatura y en base a los trabajos de varios autores que presentan análisis de diferentes algoritmos, se realiza un resumen en la **Tabla 17-4**, donde se especifican varias características y se determina que las redes neuronales son las apropiadas para la implementación en el presente trabajo.

Tabla 7-4: Comparativa entre técnicas de aprendizaje automático.

Técnica	Característica			
	Resuelve problemas no lineales	Efectividad en el manejo de datos tipo texto	Tolerante al ruido de los datos	Facilidad de implementación
Naive Bayes	Sí	Sí	No	Sí
Regresión logística	No	No	No	Sí
Máquina de soporte vectorial	Sí	Sí	No	Sí
K-vecino cercano	Sí	No	No	Sí
Árboles de decisión	Sí	No	No	Sí
Redes neuronales	Sí	Sí	Sí	No

Fuente: Prankevičius y Marcinkevičius, 2017

Realizado por: Jefferson Chariguamán, 2022

3.5.4.2 Generar el plan de prueba

La métrica seleccionada para elegir el mejor modelo entrenado, para la clasificación de texto correspondiente a noticias falsas, es la puntuación F1, debido a que esta se calcula mediante los

valores de la precisión y recall. Siendo una métrica más confiable para determinar el rendimiento del modelo, en comparación con otras como la exactitud (accuracy).

3.5.4.3 Construcción del modelo

En este apartado se describen los hiperparámetros utilizados en la implementación de los modelos de redes neuronales, estos han sido seleccionados en base a diferentes pruebas y de acuerdo a modelos implementados para la clasificación binaria de textos.

Para la implementación de los modelos de redes neuronales se utilizan los siguientes tipos; Redes neuronales MLP (Multi Layer Perceptron), redes neuronales convolucionales (Convolutional Neural Network) y redes recurrentes (Long Short Term Memory). La arquitectura de los modelos se describe a continuación.

- **Redes neuronales MLP**

Esta red neuronal es una de las más básicas, comprende de múltiples capas densas interconectadas entre sí, realizando el procesamiento de la información mediante estas, para finalmente en la capa de salida mediante la aplicación de una función de activación se realiza la clasificación. El resumen de la arquitectura de la red neuronal implementada se muestra en la **Ilustración 10-3**.

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
dense (Dense)                (None, 512)               154112
leaky_re_lu (LeakyReLU)     (None, 512)               0
dropout (Dropout)           (None, 512)               0
dense_1 (Dense)              (None, 256)              131328
leaky_re_lu_1 (LeakyReLU)   (None, 256)               0
dropout_1 (Dropout)         (None, 256)               0
dense_2 (Dense)              (None, 128)               32896
leaky_re_lu_2 (LeakyReLU)   (None, 128)               0
dropout_2 (Dropout)         (None, 128)               0
dense_3 (Dense)              (None, 1)                 129
activation (Activation)     (None, 1)                 0
-----
Total params: 318,465
Trainable params: 318,465
Non-trainable params: 0

```

Ilustración 10-3. Arquitectura de red neuronal MLP

Realizado por: Jefferson Chariguamán, 2022

En la **Ilustración 10-3** se puede observar las diversas capas interconectadas para crear la red neuronal MLP, la primera capa densa corresponde a la capa de entrada, con un total de 512 neuronas y seguido a esto se tiene una función de activación LeakyReLU. Las 3 siguientes capas densas corresponden a las capas ocultas, estas son: una capa Dropout para desactivar aleatoriamente el porcentaje de neuronas establecidas, para evitar el sobre entrenamiento, seguido a esto se encuentra una capa densa de 512 neuronas con una función de activación LeakyReLU.

La misma estructura de la capa oculta se repite nuevamente, con diferencia del número de neuronas en la capa densa, que se establece a 256. Finalmente, para la capa de salida se utiliza una capa de Dropout seguido de una capa densa de 1 sola neurona, esta cantidad corresponde al número de salidas, la función de activación en esta capa es sigmoideal, utilizada para la clasificación binaria.

- **Redes neuronales convolucionales**

Las redes neuronales convoluciones se implementan capas convolucionales, para la implementación de este modelo se hace uso de la arquitectura que se muestra en la **Ilustración 11-3**.

```

Model: "sequential_2"
-----
Layer (type)                Output Shape              Param #
-----
leaky_re_lu_2 (LeakyReLU)   (None, 300)              0
embedding_2 (Embedding)     (None, 300, 60)         1200000
dropout_8 (Dropout)         (None, 300, 60)         0
conv1d_6 (Conv1D)           (None, 297, 16)         3856
max_pooling1d_6 (MaxPooling1D) (None, 148, 16)         0
dropout_9 (Dropout)         (None, 148, 16)         0
conv1d_7 (Conv1D)           (None, 148, 16)         784
max_pooling1d_7 (MaxPooling1D) (None, 74, 16)         0
dropout_10 (Dropout)        (None, 74, 16)          0
conv1d_8 (Conv1D)           (None, 71, 16)          1040
max_pooling1d_8 (MaxPooling1D) (None, 35, 16)         0
flatten_2 (Flatten)         (None, 560)              0
dense_4 (Dense)             (None, 256)              143616
dropout_11 (Dropout)        (None, 256)              0
dense_5 (Dense)             (None, 1)                257
-----
Total params: 1,349,553
Trainable params: 1,349,553
Non-trainable params: 0

```

Ilustración 11-3. Arquitectura de red neuronal convolucional

Realizado por: Jefferson Chariguamán, 2022

Como primera capa se tiene una función de activación LeakyRelu para convertir y permitir las entradas negativas del conjunto de datos hacia la capa de embedding, seguida a esta capa se encuentra la capa de embedding con 300 dimensiones de entrada, y 60 como longitud del vector para cada palabra, y un función Glorot Normal para el embedding_initializer.

A continuación, se encuentran 3 capas convolucionales, conformadas por una capa de Dropout con parámetro 0.5, Convolutacional 1D con filtro de 16, tamaño de kernel 4, padding para rellenar el máximo permitido y MaxPooling con tamaño de ventana de agrupación máxima de 16. La función de activación utilizada en estas capas es relu.

La capa de salida está compuesta por una capa Flatten para posteriormente pasar a la capa dropout y una capa densa con una neurona y una función de activación sigmoidal por ser una clasificación binaria.

- **Redes neuronales LSTM**

En las redes neuronales LSTM utilizan internamente compuertas de entrada, olvido y salida, que utilizan funciones de activación softmax o relu, dependiendo del tipo de clasificación.

La estructura general del modelo está compuesta por capas en un modelo secuencial, esta se representa a continuación en la **Ilustración 12-3** con los parámetros como una capa de embedding, de dimensión 300, una capa LSTM con 128 neuronas, seguido de capa Global Max Pooling, además se tiene capas Dropout para evitar el sobre-entrenamiento.

Layer (type)	Output Shape	Param #
leaky_re_lu_4 (LeakyReLU)	(None, 300)	0
embedding_4 (Embedding)	(None, 300, 30)	3000000
dropout_16 (Dropout)	(None, 300, 30)	0
lstm_layer (LSTM)	(None, 300, 128)	81408
global_max_pooling1d_4 (GlobalMaxPooling1D)	(None, 128)	0
dropout_17 (Dropout)	(None, 128)	0
dense_12 (Dense)	(None, 64)	8256
dropout_18 (Dropout)	(None, 64)	0
dense_13 (Dense)	(None, 32)	2080
dropout_19 (Dropout)	(None, 32)	0
dense_14 (Dense)	(None, 1)	33

=====
 Total params: 3,091,777
 Trainable params: 3,091,777
 Non-trainable params: 0

Ilustración 12-3. Arquitectura de red neuronal LSTM

Realizado por: Jefferson Chariguamán, 2022

3.5.5 Evaluación

Una vez entrenado los modelos seleccionados con las arquitecturas e hiperparámetros especificadas anteriormente, se evalúa cada uno, utilizando el conjunto de prueba de noticias, para verificar que el modelo no presente problemas de entrenamiento como el *overfitting* o *undertitting*, además para seleccionar el modelo con mejor puntuación F1. En caso de presentarse problemas de entrenamiento, es necesario volver a la fase anterior para realizar cambios de arquitectura o hiperparámetros.

A continuación, se presentan mediante gráficos los resultados obtenidos tanto en entrenamiento como en prueba, en función del número de épocas, la puntuación F1 y la función de pérdida. La cantidad de épocas son diferentes en cada modelo debido a que alcanzan su entrenamiento óptimo y empiezan a sobre entrenar.

3.5.5.1 Red MLP

En el **Gráfico 1-3** se muestra el resultado de la puntuación F1 obtenido en la evaluación de este modelo, teniendo en cuenta que la red se entrena durante 200 épocas, y el guardado del modelo se realiza automáticamente con el valor mínimo de la función de pérdida.

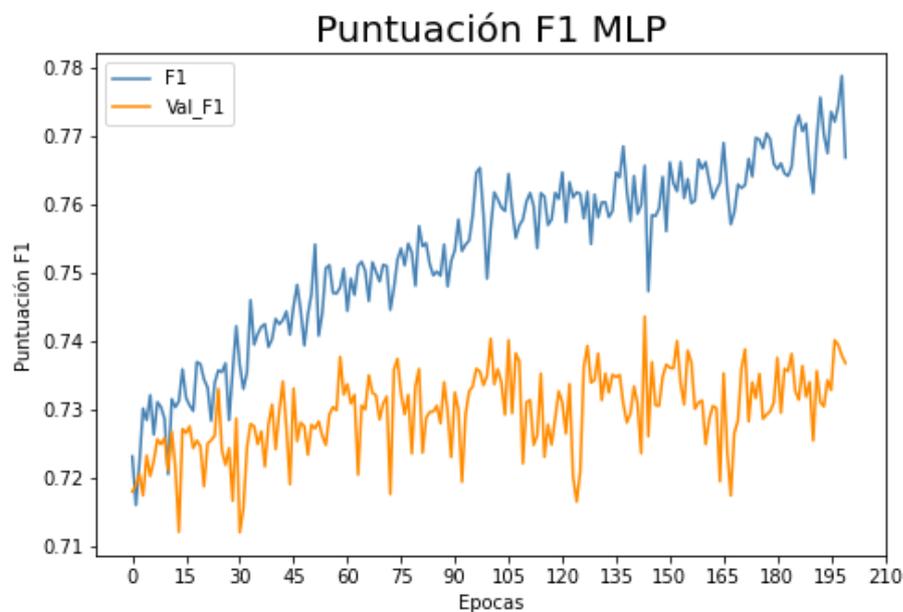


Gráfico 1-3. Puntuación F1 MLP.

Realizado por: Jefferson Chariguamán, 2022

Los valores de la función de pérdida graficados en el **Gráfico 2-3**, se observa que hasta la época 185 alcanza el valor mínimo de 0.522, luego de este el modelo empieza a sobre entrenar, empezando a incrementar sus valores con los datos de evaluación. En consecuencia, el modelo es guardado automáticamente en esta época, con un valor F1 de 0.738

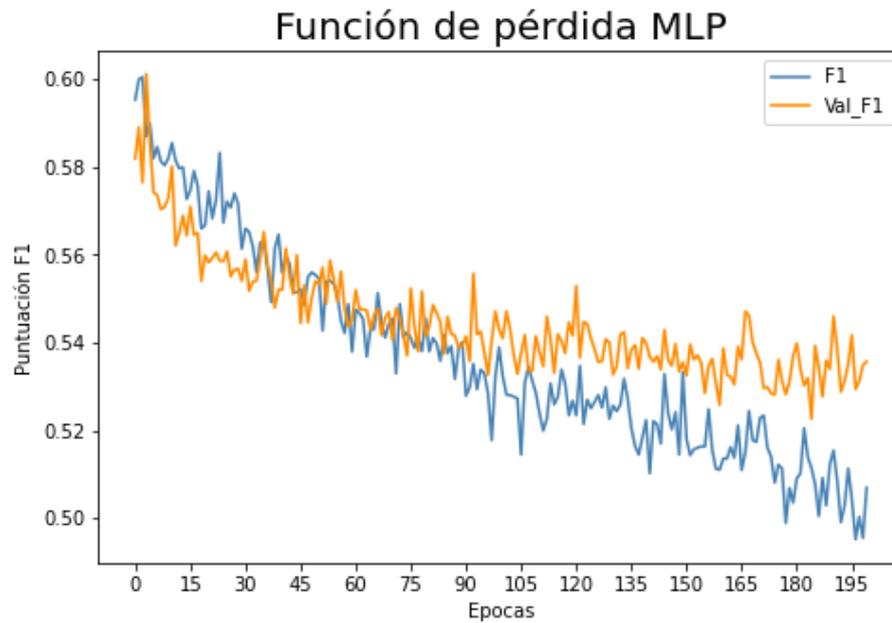


Gráfico 2-3. Función de pérdida MLP.

Realizado por: Jefferson Chariguamán, 2022

3.5.5.2 Red CNN

Para el entrenamiento de este modelo se realiza durante 50 épocas. Para la puntuación F1, se puede observar en el **Gráfico 3-3** dónde se mantiene un nivel variable, pero sin un incremento significativo en cada época, incluso empieza a disminuir la puntuación debido al sobreentrenamiento. Teniendo en cuenta el valor mínimo de la función de pérdida de 0.660 (Ver **Gráfico 4-3**) en la época 37, y con un valor del 0.678 de puntuación F1, el modelo es guardado automáticamente.

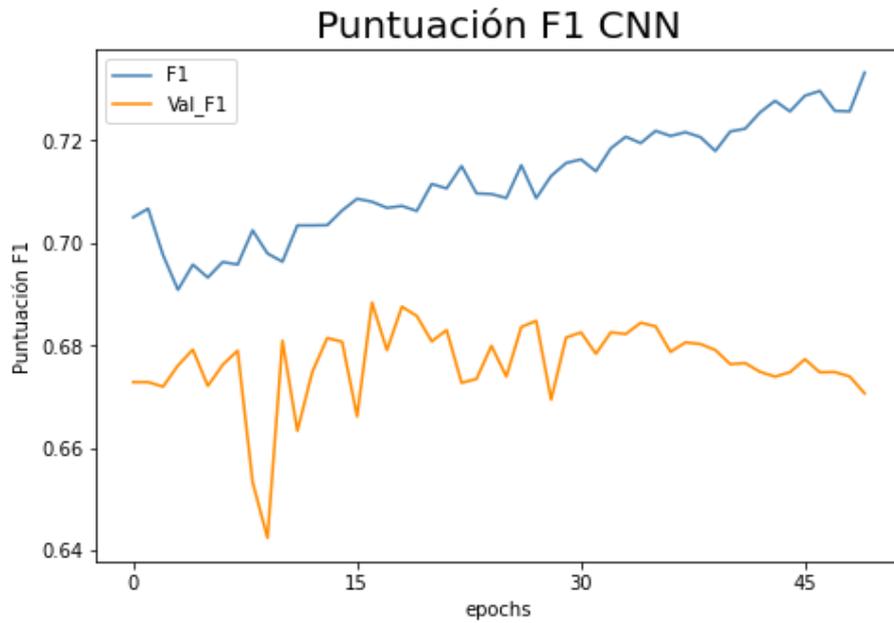


Gráfico 3-3. Puntuación F1 CNN.

Realizado por: Jefferson Chariguamán, 2022

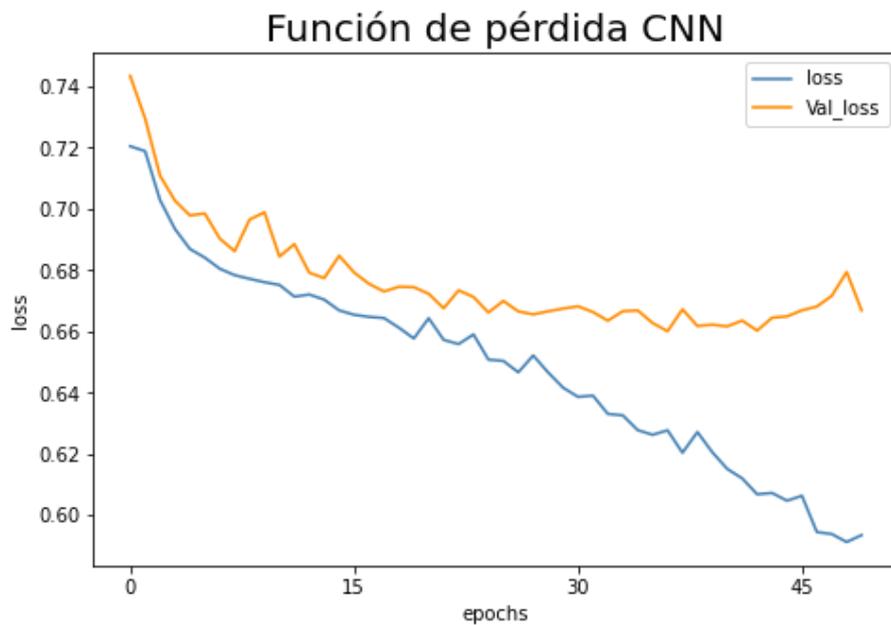


Gráfico 4-3. Función de pérdida CNN.

Realizado por: Jefferson Chariguamán, 2022

3.5.5.3 Red LSTM

Las pruebas de este modelo se realizan durante 200 épocas, los valores de la función de pérdida, se encuentra en el **Gráfico 5-3**, donde se observa que tiene una tendencia de disminución desde el inicio hasta la época 149, con un valor de 0,585, luego de este punto, el valor empieza a aumentar, provocando un sobre entrenamiento.

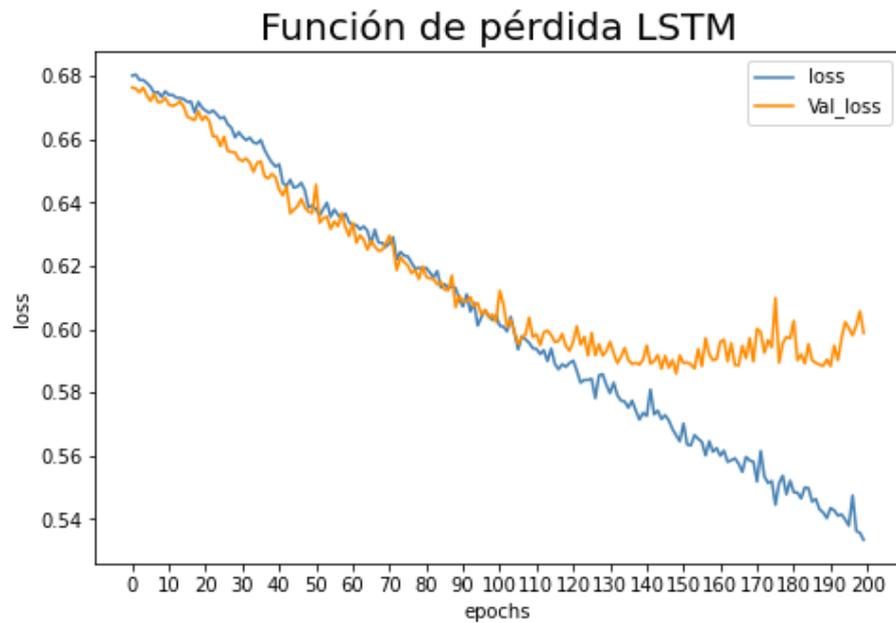


Gráfico 5-3. Función de pérdida LSTM.

Realizado por: Jefferson Chariguamán, 2022

En el **Gráfico 6-3** se muestran los valores de la puntuación F1, que tiende a crecer desde la época 22, con un valor de 0.5690 hasta la época 149, donde se determina que es la época óptima de entrenamiento, con una puntuación F1 de 0,7468, de igual manera, el gráfico corresponde a un entrenamiento de 200 épocas.

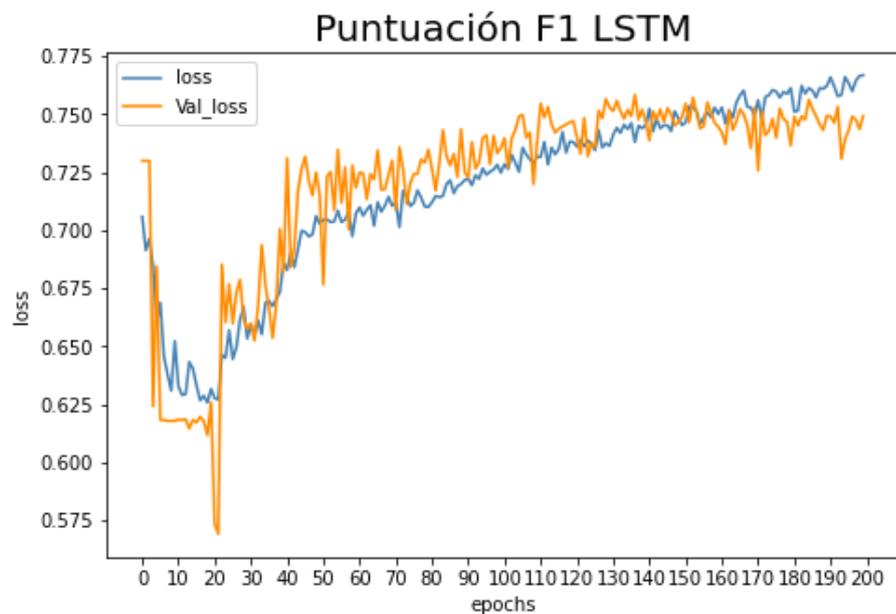


Gráfico 6-3. Puntuación F1 LSTM.

Realizado por: Jefferson Chariguamán, 2022

A continuación, en la **Tabla 8-3** se muestran un resumen de los resultados de los modelos de redes neuronales implementados, y las mejores métricas determinadas en su entrenamiento. Como resultado a las pruebas realizadas la arquitectura de red neuronal con mejor desempeño es la LSTM, con una puntuación F1 de 0,746.

Tabla 8-3: Resultados de entrenamiento de redes neuronales.

Modelo	Puntuación F1	Función de pérdida
MLP	0.738	0.522
CNN	0.678	0.660
LSTM	0.746	0.56

Realizado por: Jefferson Chariguamán, 2022

3.5.6 Despliegue

Esta fase de despliegue se realiza mediante la integración del modelo entrenado, en el servicio web, en el apartado de la metodología SCRUM, se detalla el proceso de integración en el servicio web.

3.6 Metodología de desarrollo SCRUM

Luego de aplicar una encuesta a la muestra planteada, para determinar la necesidad de un artefacto de software que permita la verificación de noticias publicadas en redes sociales, en el ANEXO A se muestran los resultados, donde el 83% de los encuestados desconocen de una herramienta software que les permita la verificación de una noticia, además el 32,6% considera que las noticias publicadas en redes sociales no son confiables, también el 32,9% de los encuestados consideran que es muy importante comprobar si una noticia publicada en internet es verdadera. Por lo tanto, se determina la necesidad del desarrollo de este servicio web.

3.6.1 Fases de planificación

En esta fase se realiza el análisis del Trabajo de Integración Curricular para determinar las funcionalidades que tendrá el servicio web, personas que forman parte del proyecto, producto backlog.

3.6.1.1 *Requerimientos Funcionales*

En el desarrollo del servicio web se establecen los siguientes requisitos presentados en la tabla **Tabla 9-3** y requisitos no funcionales en la **Tabla 10-3**.

Tabla 9-3: Requerimientos funcionales.

ID.	Requerimiento	Descripción
RF1	El sistema permitirá el preprocesamiento del texto de una noticia mediante el uso de librerías para el procesamiento del lenguaje natural.	El sistema recibirá el texto de las noticias mediante peticiones POST para su procesamiento.
RF2	El sistema permitirá la clasificación del texto de una noticia en falso o verdadero	El sistema procesará las peticiones GET para enviar al cliente la interfaz de usuario.
RF3	Se preprocesará el conjunto de datos para el entramiento del modelo	Se debe realizar el preprocesamiento del conjunto de datos para su uso posterior en el entrenamiento del modelo.
RF4	Se codificará el modelo para la clasificación de noticias	Se procederá con la codificación del modelo seleccionado con mejor rendimiento para su posterior implementación en el servicio web.
RF5	Se implementará el modelo entrenado en el servicio web	Se implementará el modelo desarrollado en el servicio web, para la clasificación de noticias.

Realizado por: Jefferson Chariguamán, 2022

Tabla 10-3: Requerimiento no funcional.

ID	Requerimiento	Descripción
RNF1	Eficiencia	El cálculo del tiempo será realizado en el capítulo 4, mediante pruebas realizadas al servicio web desarrollado.

Realizado por: Jefferson Chariguamán, 2022

3.6.1.2 *Personas y roles involucrados en el proyecto*

Dentro del desarrollo del sistema web, se establecen a las personas en los roles involucrados establecidos en la metodología, en la **Tabla 11-3** se especifican la información.

Tabla 11-3: Personas y roles involucrados en el proyecto.

Personas	Rol	Contacto	Institución
Ing. Patricio Moreno	Scrum Master	patricio.moreno@esPOCH.edu.ec	ESPOCH
Ing. Gisel Bastidas	Product Owner	gis.bastidas@esPOCH.edu.ec	ESPOCH
Jefferson Chariguamán	Development Team	Jefferson.chariguaman@esPOCH.edu.ec	ESPOCH

Realizado por: Jefferson Chariguamán, 2022

3.6.1.3 Product Backlog

Los requerimientos planteados para el desarrollo del proyecto, fueron priorizados en conjunto con el product owner, de esta manera, mediante el uso del product backlog se detallan las tareas, sus estimaciones y las priorizaciones de los requerimientos.

El método T-shirt permite registrar cuánto tiempo toma realizar una tarea, mediante la asignación de una talla a cada una de estas, en la **Tabla 12-3** se detallan las horas y puntos asignados a cada talla, las cuales permiten estimar el esfuerzo a cada una de las historias de usuario/técnicas.

Tabla 12-3: Valoraciones de tiempo según T-Shirt.

Talla	Horas de trabajo	Puntos
XL	>20	>20
L	20	20
M	10	10
S	5	5

Realizado por: Jefferson Chariguamán, 2022

Las estimaciones de las tareas, asignándole una prioridad se encuentran en la **Tabla 13-3**.

Tabla 13-3: Estimación de tareas.

ID	Historia de usuario	Prioridad	Estimación
HT1	Definición de una arquitectura para el servicio web.	Alta	M
HT2	Análisis y selección de las herramientas	Alta	M

	para el desarrollo del servicio web.		
HT3	Desarrollo de los endpoints definidos para procesar las peticiones.	Alta	L
HT4	Preprocesamiento del conjunto de datos para el entrenamiento del modelo.	Alta	L
HT5	Codificación y entrenamiento del modelo con la arquitectura seleccionada.	Alta	L
HT6	Integración del modelo entrenado para la clasificación de noticias.	Alta	M
HT7	Realizar el manual de usuario del servicio web	Medio	M
HU1	Crear la interfaz de usuario	Alta	M
HU2	Gestionar el envío y respuesta del texto.	Alta	M

Realizado por: Jefferson Chariguamán, 2022

3.6.1.4 *Sprint Backlog*

En este apartado se detalla en la **Tabla 14-3** las fechas de inicio, fin, y las horas estimadas en las que se realizarán las historias de usuario, considerando la duración del Sprint de 50 horas.

Tabla 14-3: Sprint Backlog.

ID	Actividad	Horas	Fecha de inicio	Fecha de fin
Sprint 1		40	01/05/2022	15/05/2022

HT1	Definición de una arquitectura para el servicio web.	10	01/05/2022	02/05/2022
HT2	Análisis y selección de las herramientas para el desarrollo del servicio web.	10	03/05/2022	05/05/2022
HT3	Desarrollo de los endpoints definidos para procesar peticiones.	20	06/05/2022	15/05/2022
Sprint 2		40	16/05/2022	30/05/2022
HT4	Preprocesamiento del conjunto de datos para el entrenamiento del modelo.	20	16/05/2022	22/05/2022
HT5	Codificación y entrenamiento del modelo con la arquitectura seleccionada.	20	23/05/2022	30/05/2022
Sprint 3		40	31/05/2022	14/06/2022
HT6	Integración del modelo entrenado para la clasificación de noticias.	10	31/05/2022	03/06/2022
HT7	Realizar el manual de usuario del servicio web	10	04/06/2022	07/06/2022
HU1	Crear interfaz de usuario.	10	08/06/2022	11/06/2022
HU2	Gestionar el envío y respuesta del texto.	10	12/06/2022	14/06/2022

Realizado por: Jefferson Chariguamán, 2022

3.6.1.5 Historias de usuario

A continuación, se presenta la **Tabla 15-3.** como ejemplo de documentación de historia de usuario. Estas historias de usuario son empleadas para conocer los requerimientos planteados a ser desarrollados en el software, dentro de estas se detallan datos como las fechas, la descripción, roles de usuarios, además, las tareas de ingeniería se encuentran en la **Tabla 16-3, Tabla 18-3** y

Tabla 15-3: HU1 Crear interfaz de usuario.

Historia de usuario	
Código: HU1	Nombre: Crear interfaz de usuario.
Usuario: Consumidor del servicio web	Sprint Asignado: 3
Fecha de inicio: 08/06/2022	Fecha de fin: 11/06/2022
Puntos estimados: 10	Puntos reales: 10
Descripción: Como usuario del servicio web, requiero presentar resultado de la clasificación en la interfaz de usuario, así como mostrar los diversos componentes para la interacción con el sistema.	
Observaciones	
Pruebas de aceptación	
<ul style="list-style-type: none"> • Verificar que la información enviada desde el servicio web, se visualice en la interfaz de usuario. • Verificar que los componentes para interacción con el servicio web se presenten correctamente. 	

Realizado por: Jefferson Chariguamán, 2022

Tabla 16-3: TI_HU1 Crear interfaz de usuario.

Tarea de ingeniería	
Identificador: TI_HU1	Nombre de la historia: Crear interfaz de usuario.
Nombre de tarea: Desarrollar la interfaz para visualizar resultados	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 08/06/2022	Fecha de fin: 09/06/2022
Descripción: Desarrollar la función que permita extraer los datos de la respuesta del servicio web, y presentar en la interfaz de usuario.	
Pruebas de aceptación:	
<ul style="list-style-type: none"> • Verificar que la respuesta aparece en la interfaz de usuario. 	

Realizado por: Jefferson Chariguamán, 2022

Tabla 17-3: T2_HU1 Crear interfaz de usuario.

Tarea de ingeniería	
Identificador: T2_HU1	Nombre de la historia: Crear interfaz de usuario.
Nombre de tarea: Desarrollar la interfaz para la interacción con el sistema.	

Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 10/06/2022	Fecha de fin: 11/06/2022
Descripción: Desarrollar el documento HTML para la presentación de los componentes con el que el cliente interactuará	
Pruebas de aceptación: <ul style="list-style-type: none"> • Verificar que la respuesta aparece en la interfaz de usuario. 	

Realizado por: Jefferson Chariguamán, 2022

3.6.2 Fase de desarrollo

En esta fase se detallan las actividades que se realizan en el desarrollo del servicio web, integrando un modelo de aprendizaje de máquina para clasificar noticias.

3.6.2.1 Arquitectura del sistema

La arquitectura de este servicio forma parte de la categoría SOA (Service Oriented Architecture), que establece un diseño para que demás aplicaciones puedan integrarse y accedan a funcionalidades a través de la red. En la **Ilustración 13-3** se presenta la arquitectura utilizada en el proyecto para el desarrollo del servicio web.

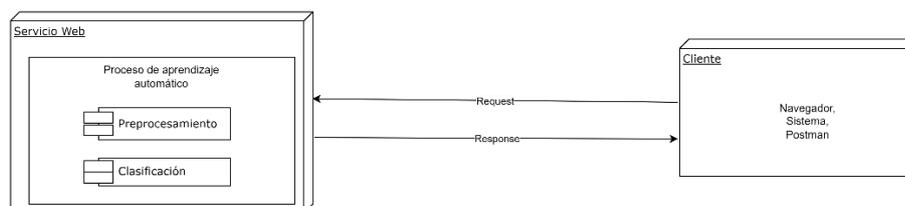


Ilustración 13-3. Arquitectura del servicio web.

Realizado por: Jefferson Chariguamán, 2022

3.6.2.2 Diseño de interfaz

El diseño de interfaces permite que el usuario final conozca de qué manera se presentará el sistema y cómo podrá interactuar mediante esta, para este caso en particular, el diseño es minimalista, donde se tiene apartados de información acerca del funcionamiento del sistema, uno donde se muestra la respuesta del servidor y los parámetros utilizados para el procesamiento. Además, se incluye un área de texto donde se ingresa el contenido de la noticia, en la **Ilustración 14-3** se visualiza un bosquejo de la interfaz.

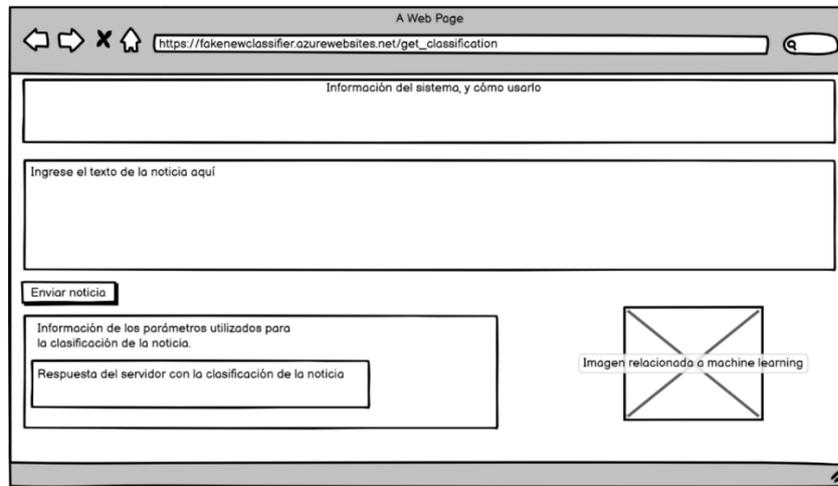


Ilustración 14-3. Boceto de la interfaz de usuario.

Realizado por: Jefferson Chariguamán, 2022

3.6.3 Fase de pruebas

En esta fase se realiza la revisión y retrospectiva que después de cada sprint implementado, además de realizar una revisión general del proyecto para validar y verificar que los requisitos se hayan cumplido de acuerdo a la planificación, y de este modo cumplir con los requisitos funcionales planteados. En la Tabla 18-3 y Tabla 19-3, se muestran las pruebas de aceptación de las historias de usuarios para el requisito de la interfaz de usuario, donde se describen su identificador, el responsable, las condiciones y los pasos para su ejecución. Las demás pruebas realizadas se encuentran en el Anexo C de este documento.

Tabla 18-3: PA_TI_HU Crear interfaz de usuario.

Prueba de aceptación	
Identificador: PA_TI_HU1	Nombre de la historia: Crear interfaz de usuario.
Nombre: Desarrollar la interfaz que permita visualizar los resultados.	
Responsable: Usuario consumidor del servicio	Fecha: 09/06/2022
Descripción: Verificar que luego de haber realizado la clasificación, la respuesta se visualice.	
Condiciones de Ejecución: Es necesario que la respuesta a la petición enviada sea respondida por el servicio web.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar al servicio web• Ingresar el texto de la noticia y presionar en el botón de “enviar noticia”• Verificar que la información se visualice	
Resultado esperado: Los resultados de la clasificación se muestran en la interfaz de usuario.	
Evaluación de la prueba: Exitosa.	

Realizado por: Jefferson Chariguamán, 2022

Tabla 19-3: PA_T2_HU1 Crear interfaz de usuario.

Prueba de aceptación	
Identificador: PA_T2_HU1	Nombre de la historia: Crear interfaz de usuario.
Nombre: Desarrollar la interfaz para la interacción con el sistema.	
Responsable: Usuario consumidor del servicio	Fecha: 11/06/2022
Descripción: Verificar que los componentes para la interacción con el sistema se presenten correctamente se muestren.	
Condiciones de Ejecución: Es necesario que el servicio web esté activo y procesando las peticiones POST/GET.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar al servicio web • Verificar que los componentes se visualicen 	
Resultado esperado: Los componentes se muestran en la interfaz de usuario.	
Evaluación de la prueba: Exitosa.	

Realizado por: Jefferson Chariguamán, 2022

3.6.4 Fase de despliegue

Esta última fase, se envían los entregables del sistema desarrollado, incluyendo la integración del modelo de Machine Learning implementado, el proceso de despliegue se realiza desde Visual Studio Code, como primer paso se sube el código del proyecto a GitHub como se muestra en la Ilustración 15-3, y luego se despliega mediante el plugin de Azure.

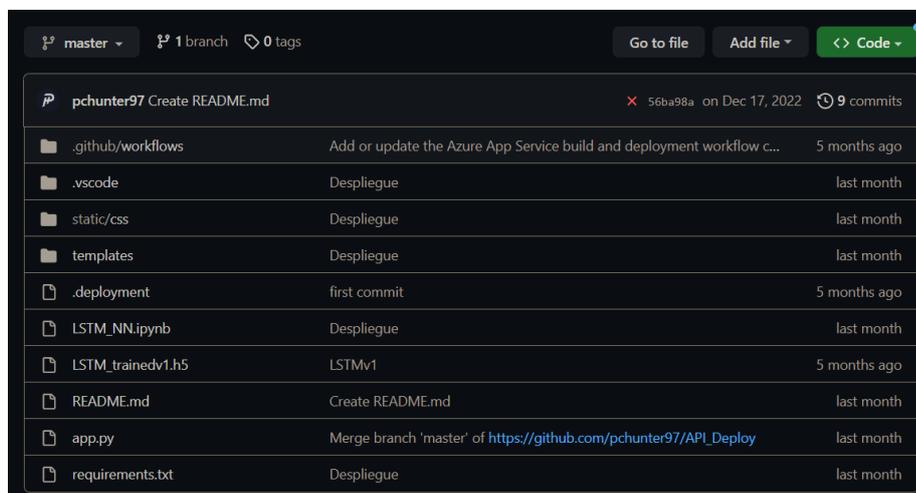


Ilustración 15-3: Repositorio en GitHub.

Realizado por: Jefferson Chariguamán, 2022

Este proyecto es desplegado en la plataforma Microsoft Azure, utilizando App Services, con ubicación en East US 2 sobre un servidor con sistema operativo Linux como se muestra en la Ilustración 16-3.

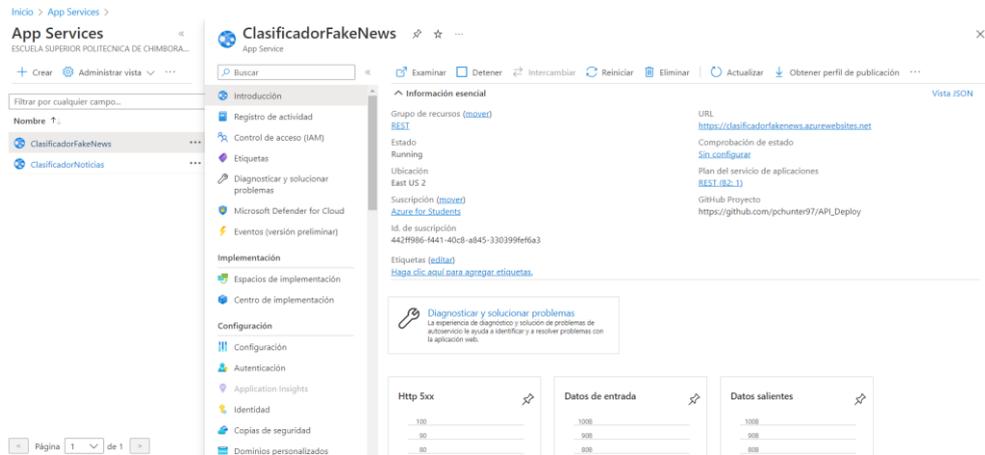


Ilustración 16-3: Despliegue del servicio web.

Realizado por: Jefferson Chariguamán, 2022

CAPÍTULO IV

4 RESULTADOS

En este capítulo se identifican los indicadores de eficiencia de desempeño del servicio web para la clasificación de noticias falsas, para su posterior análisis y evaluación de los resultados, con el fin determinar el límite en el que se encuentra el tiempo promedio de respuesta del servicio web, tomando en cuenta los límites de tiempos mencionados por Nielsen, además se analiza la influencia de la extensión de las noticias en los tiempos de respuesta, preprocesamiento y clasificación.

4.1 Eficiencia de desempeño

Esta característica está especificada en la ISO 25010, de esta se realiza la evaluación de la subcaracterística comportamiento temporal, que representa el desempeño relativo a la cantidad de recursos utilizados bajo ciertas condiciones. Dentro de esta subcaracterística se utiliza como métrica a los tiempos de respuesta, para la evaluación de la eficiencia de desempeño del servicio web desarrollado.

4.1.1 Tiempo de respuesta

Para realizar la medición de la métrica, se considera todo el tiempo desde que el cliente envía la petición al servicio web, internamente se realiza el preprocesamiento de los datos recibidos, y la clasificación mediante el modelo de aprendizaje de máquina, hasta que el cliente recibe la respuesta con la clasificación de la noticia. De este modo permite recolectar los tiempos en que una noticia tardaría en ser clasificada mediante el servicio web, para un análisis posterior. Las métricas definidas para de eficiencia de desempeño se muestran en la **Tabla 20-3**.

Tabla 20-3: Métricas para la característica de calidad Eficiencia en el desempeño.

Métricas para la característica de calidad Eficiencia en el desempeño			
Subcaracterística	Comportamiento temporal		
Métrica	Tiempo de respuesta	Tiempo de espera	Rendimiento
Fase del ciclo de vida de calidad del producto	Interna/externa	Interna/externa	Interna/externa
Propósito de la métrica de calidad	¿Cuál es el tiempo estimado para completar una tarea?	¿Cuál es el tiempo desde que se envía una instrucción, para que inicie un trabajo, hasta que lo completa?	¿Cuántas tareas pueden ser procesadas por unidad de tiempo?
Método de aplicación	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta	Tomar el tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo	Contar el número de tareas completadas en un intervalo de tiempo

Fórmula	X=B-A A=Tiempo de envío de petición. B=Tiempo en recibir la primera respuesta.	X=B-A A=Tiempo cuando se inicia un trabajo. B=Tiempo en completar el trabajo	X=A/T A=Número de tareas T=Intervalo de tiempo Donde: T>0
Valor deseado	0<=X<=1 El más cercano a 0 es el mejor. Donde el peor caso es >=15t.	0<=X<=1 El más cercano a 0 es el mejor. Donde el peor caso es >=15t	X=A/T El más lejano a 0/t es el mejor. Donde el mejor caso es >=10/t
Tipo de medida	X=Tiempo/Tiempo A=Tiempo B=Tiempo	X=Tiempo/Tiempo A=Tiempo B=Tiempo	X=Contable/Tiempo A=Contable T=Tiempo
Recursos utilizados	Especificación de requerimientos, Código fuente, Desarrollador, Tester.	Especificación de requerimientos, Código fuente, Desarrollador, Tester.	Especificación de requerimientos, Código fuente, Desarrollador, Tester.

Fuente: (ISO/IEC 25023, 2016)

Realizado por: Jefferson Chariguamán, 2022

4.2 Despliegue de la aplicación

El servicio web fue desplegado en plataforma de Azure, mediante el servicio de *App Service*, con ubicación del centro de datos East US 2, considerando que tiene la latencia más baja en comparación con los demás centros de datos. Luego de realizar el despliegue de la aplicación, se procede a realizar las pruebas mediante el envío de peticiones al servicio web, utilizando un script desarrollado en el lenguaje Python para consumir el servicio.

4.3 Tiempos de respuesta

La cantidad de peticiones establecidas para realizar las pruebas al servicio web está definida por la muestra, con un valor de 4160, esta es dividida en 3 grupos de 1386 peticiones, considerando la cantidad de caracteres en el contenido de la noticia, categorizadas como: cortas, medias y largas. En la **Tabla 18-4** se visualiza el número de caracteres que tiene cada categoría.

Tabla 21-4: Descripción de las categorías de las noticias.

Categoría de noticia	Rango de caracteres
Cortas	1300 a 2000
Medias	2000 a 3200
Largas	Más de 3200

Realizado por: Jefferson Chariguamán, 2022

4.3.1 Tiempos de respuesta con noticias de longitud corta

En el **Gráfico 7-4** se representan los tiempos de respuestas correspondientes a las peticiones realizadas al servicio web, con noticias de longitud corta que tienen una extensión de entre 1300 y 2000 caracteres, como se observan, existen picos que sobresalen del resto de tiempos, y con el fin de evitar que estos afecten el promedio, se identifican valores atípicos para realizar una segunda prueba con estos (Ver **Gráfico 8-4**). El **Gráfico 7-4** y los valores de mínimo, promedio y máximo de estos tiempos que se encuentran en la **Tabla 19-4** son obtenidos mediante el script utilizado para el análisis de resultados.

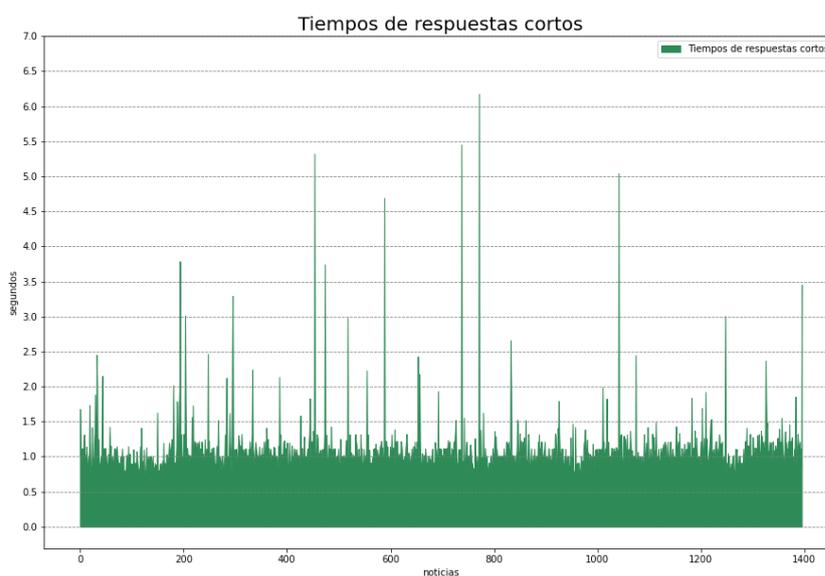


Gráfico 7-4. Tiempos de respuesta noticias longitud corta.

Realizado por: Jefferson Chariguamán, 2022

Tabla 22-4: Resumen de tiempos noticias cortas.

Tiempo mínimo	Tiempo promedio	Tiempo máximo
0.711474	1.0114535	6.176839

Realizado por: Jefferson Chariguamán, 2022

Con la finalidad de identificar los valores atípicos de los tiempos de respuesta representados en el **Gráfico 8-4**, se aplica el test de Tukey, implicando el cálculo de tres cuartiles que se muestran en la **Tabla 20-4**, por otro lado, los tiempos de respuesta son representados gráficamente mediante el diagrama de cajas (Ver **Gráfico 8-4**) con el objetivo de visualizar la presencia de valores atípicos representados por pequeños círculos. El cálculo de los cuartiles y el aislamiento de los valores atípicos se realizan mediante una función declarada en el script utilizado para el análisis de resultados. Estos valores aislados son utilizados en una segunda prueba para analizar los tiempos de respuesta de los procesos internos de preprocesamiento y clasificación.

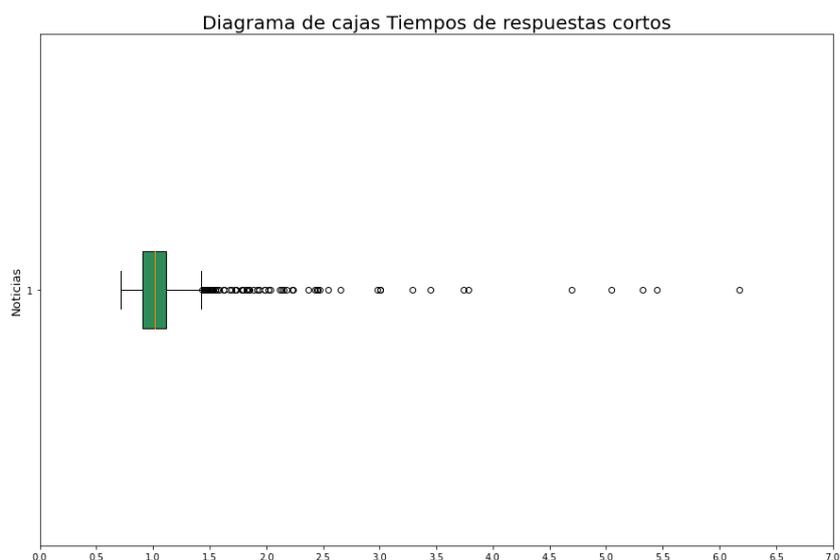


Gráfico 8-4. Diagrama de cajas, tiempos de respuesta noticias de longitud corta.

Realizado por: Jefferson Chariguamán, 2022

Tabla 23-4: Tabla de cuartiles, noticias cortas.

Cuartil	Valor
Q1	0.910059
Q2	1.0114535
Q3	1.11760075

Realizado por: Jefferson Chariguamán, 2022

Los valores atípicos observados en el diagrama de cajas (Ver **Gráfico 8-4**), son utilizados para identificar la noticia a la que corresponden, estas son enviadas al servicio web en una segunda prueba, con el objetivo de analizar y contrastar los tiempos de respuesta, preprocesamiento y clasificación entre la primera y segunda prueba. Mediante el **Gráfico 9-4.**, se representan los tiempos de respuesta de ambas pruebas, la línea roja corresponde a la primera y la verde a la segunda prueba, como se puede observar, en la segunda prueba los tiempos de respuesta tienen una reducción que no sobrepasan el valor de 1 segundo, en comparación con la primera prueba en la que la mayoría de tiempos de respuesta sobrepasan el valor de 1.5 segundos.

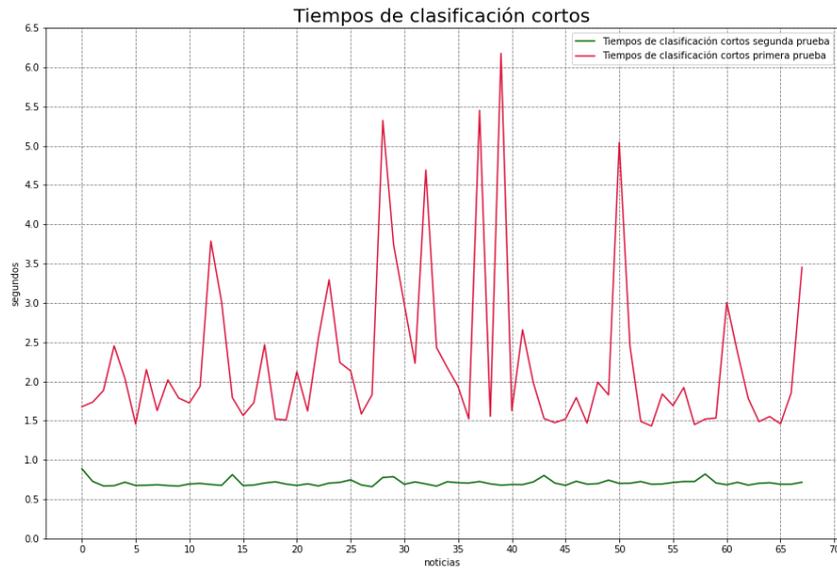


Gráfico 9-4. Tiempos de respuestas atípicos de noticias longitud corta.

Realizado por: Jefferson Chariguamán, 2022

Con el fin de analizar el comportamiento temporal de los procesos internos de clasificación y preprocesamiento del servicio web, utilizando las noticias con tiempos atípicos, en el **Gráfico 10-4** se representan los tiempos de clasificación, donde se puede apreciar que, en ambas pruebas, en su mayoría los valores son semejantes. Por otro lado, en el **Gráfico 11-4** se encuentran representados los tiempos de preprocesamiento, que, de igual manera, tienden a ser similares, excepto por algunos tiempos. De este modo se considera que la reducción en los tiempos de respuesta entre la primera y segunda prueba representados en el **Gráfico 9-4**, no es debido a los procesos internos de preprocesamiento y clasificación.

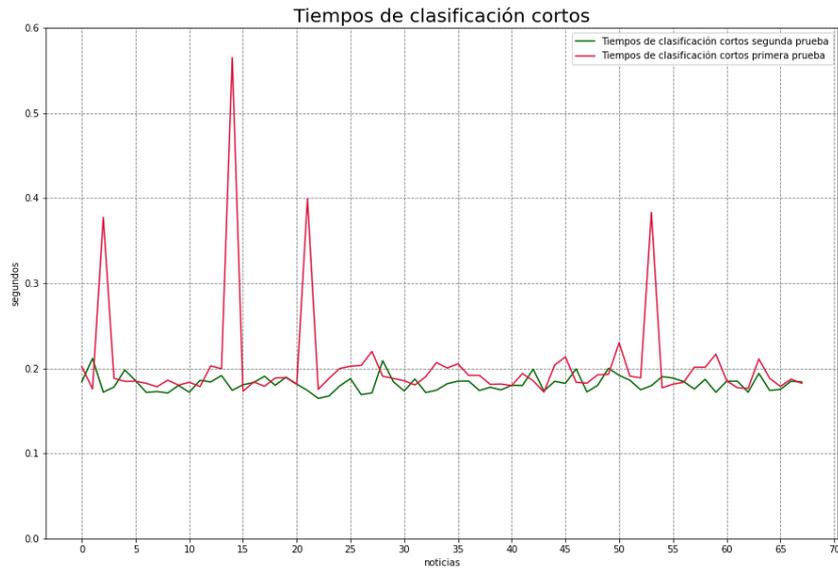


Gráfico 10-4. Tiempos de clasificación atípicos de noticias longitud corta.

Realizado por: Jefferson Chariguamán, 2022

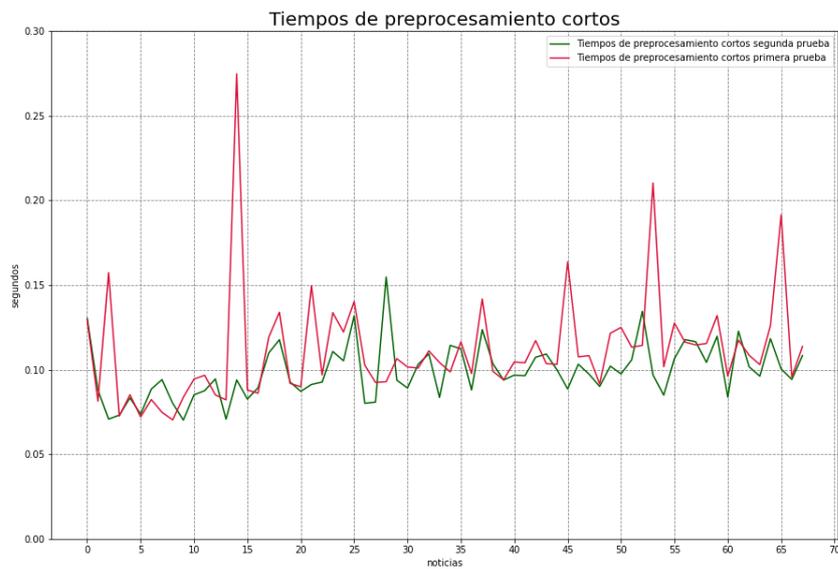


Gráfico 11-4. Tiempos de preprocesamiento atípicos de noticias longitud corta.

Realizado por: Jefferson Chariguamán, 2022

4.3.2 Tiempos de respuesta con noticias de longitud media

Los tiempos de respuesta correspondientes a la categoría media, con una extensión de caracteres entre 2000 y 3200 caracteres, se representan en el **Gráfico 12-4**, en el cual se observan picos que sobresalen del resto, estos valores pueden afectar al tiempo promedio, por lo que se debe

identificar valores atípicos, pero debido a la cantidad de puntos en el gráfico donde no se puede determinar visualmente, por tanto se utiliza el script para el análisis de resultados, para calcular los valores mínimo, promedio y máximo que se encuentran resumidos en la **Tabla 21-4**.

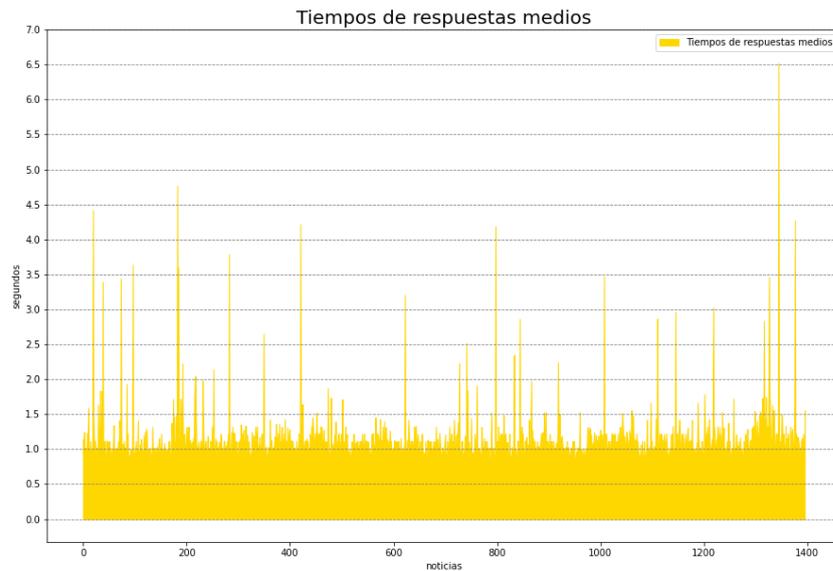


Gráfico 12-4. Tiempos de respuesta noticias longitud media.

Realizado por: Jefferson Chariguamán, 2022

Tabla 24-4: Resumen de tiempos noticias medias.

Tiempo mínimo	Tiempo promedio	Tiempo máximo
0.746878	1.031371	6.523439

Realizado por: Jefferson Chariguamán, 2022

Como se mencionó, los picos de tiempos de respuesta pueden afectar en el cálculo del promedio, por esa razón se aplica el test de Tukey, donde se determinan tres cuartiles (ver **Tabla 22-4**) y los límites superiores e inferiores, para identificar visualmente los valores atípicos presentes en la lista de tiempos de respuesta, se utiliza un diagrama de cajas representado en el **Gráfico 13-4**, estos valores atípicos se muestran como pequeños círculos. Para seleccionar y aislar los valores atípicos de la lista de tiempos de respuesta para una segunda prueba, se utiliza el script de análisis de resultados.

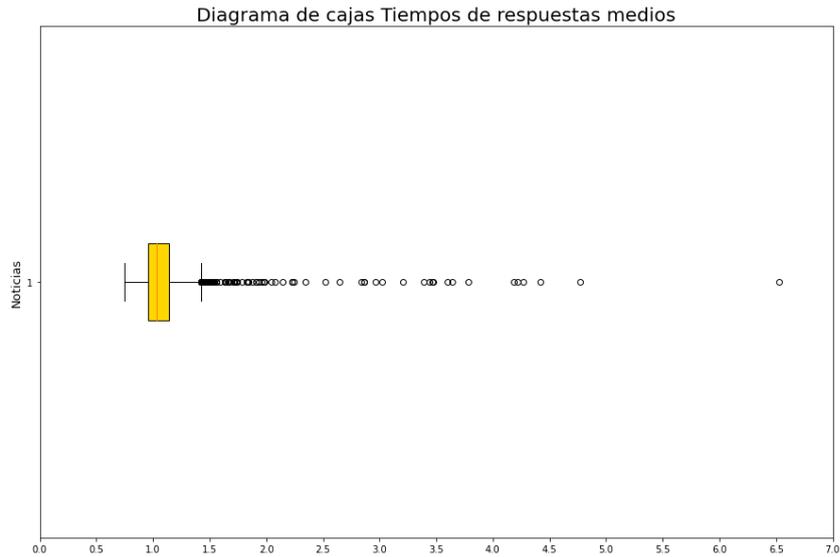


Gráfico 13-4. Diagrama de cajas, tiempos de respuesta de longitud media.

Realizado por: Jefferson Chariguamán, 2022

Tabla 25-4: Resumen de cuartiles noticias medias.

Cuartil	Valor
Q1	0.9570960000000001
Q2	1.031371
Q3	1.1438812500000002

Realizado por: Jefferson Chariguamán, 2022

Las noticias correspondientes a los valores atípicos, son enviadas nuevamente al servicio web para obtener los nuevos tiempos de respuesta, clasificación y preprocesamiento, con el fin de analizar y contrastar los tiempos de ambas pruebas. En el **Gráfico 14-4** se representan los tiempos de respuesta de las dos pruebas, donde se observa que la línea verde correspondiente a los tiempos de respuesta de la segunda prueba, tiene una reducción significativa en comparación con la línea roja, que representa los tiempos de respuesta de la primera prueba.

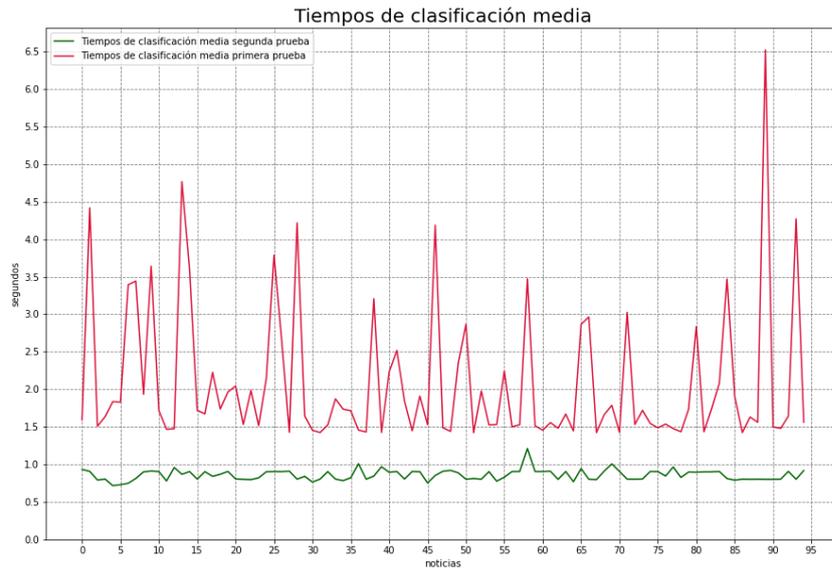


Gráfico 14-4. Tiempos de respuestas atípicos de noticias longitud media.

Realizado por: Jefferson Chariguamán, 2022

Por otro lado, en el **Gráfico 15-4** y **Gráfico 16-4** correspondientes a los tiempos de clasificación y preprocesamiento, mantienen similitud tanto en la primera (línea roja) como en la segunda prueba (línea verde), por lo tanto, se deduce que los tiempos de los procesos internos no afectan la variación de los tiempos de respuesta en ambas pruebas.

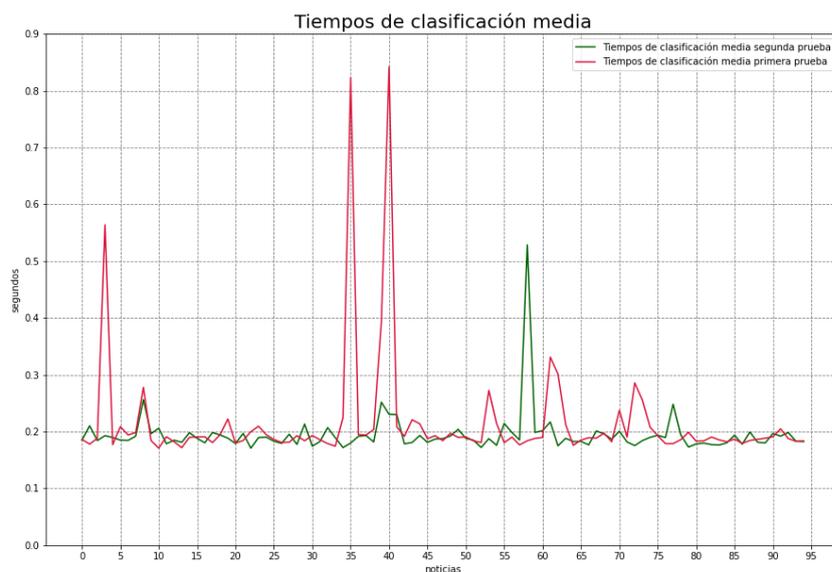


Gráfico 15-4. Tiempos de clasificación atípicos de noticias longitud media.

Realizado por: Jefferson Chariguamán, 2022

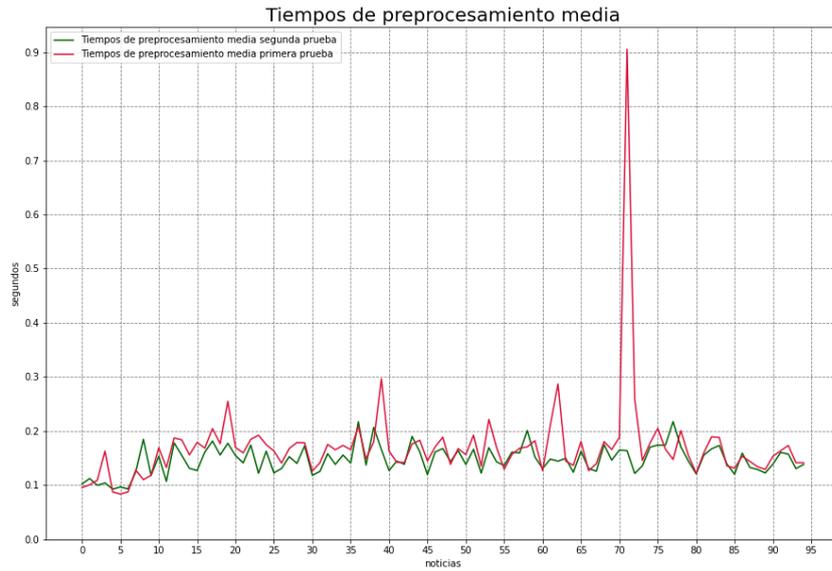


Gráfico 16-4. Tiempos de preprocesamiento atípicos de noticias longitud media.

Realizado por: Jefferson Chariguamán, 2022

4.3.3 Tiempos de respuesta con noticias de longitud larga

Los tiempos de respuesta obtenidas a partir del envío de peticiones al servicio web, utilizando las noticias de longitud larga con una cantidad de caracteres mayor a 3200, se muestran a en el **Gráfico 17-4**, en este se puede apreciar la presencia de tiempos picos que superan por mucho a la mayoría de tiempos, por tal motivo se aplicará el test de Tukey para determinar la presencia de valores atípicos. Los valores de los tiempos mínimo, promedio y máximo que se utilizarán en un análisis posterior, son obtenidos mediante el script para analizar resultados, estos tiempos se muestran en la **Tabla 23-4**.

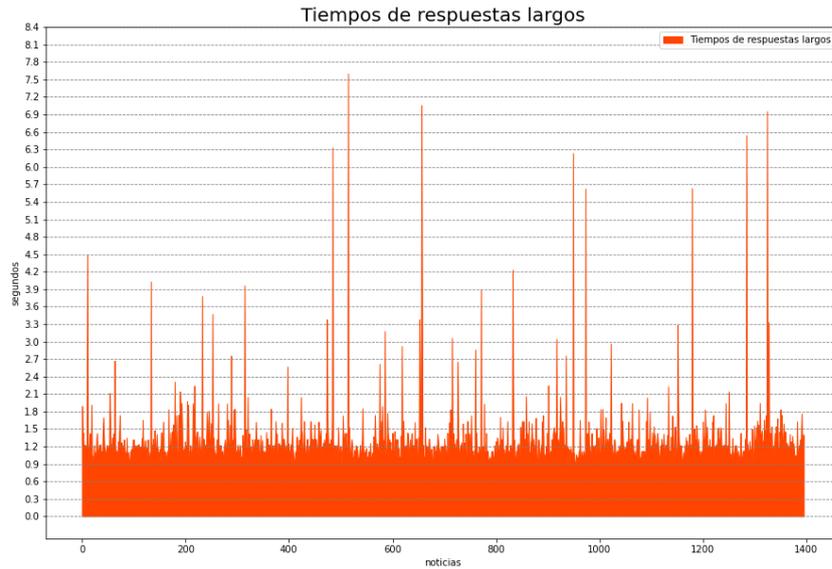


Gráfico 17-4. Tiempos de respuesta noticias longitud larga.

Realizado por: Jefferson Chariguamán, 2022

Tabla 26-4: Resumen de tiempos noticias largas.

Tiempo mínimo	Tiempo promedio	Tiempo máximo
0.815583	1.159528	7.603454

Realizado por: Jefferson Chariguamán, 2022

Como se observó en el **Gráfico 17-4** existen valores sobresalientes, que mediante la aplicación del test de Tukey se determinan tres cuartiles (Ver **Tabla 24-4**) calculados mediante el script para el análisis de resultados, con el objetivo de identificar los valores atípicos correspondientes a los tiempos de respuesta y aislarlos para realizar una segunda prueba con estos, con el propósito de analizar su comportamiento. De manera análoga para determinar la presencia de valores atípicos de forma gráfica, los tiempos de respuesta de noticias de longitud larga, son representados mediante un diagrama de cajas en el **Gráfico 18-4**, donde los tiempos atípicos son los que se encuentran por encima del extremo superior y son simbolizados por círculos.

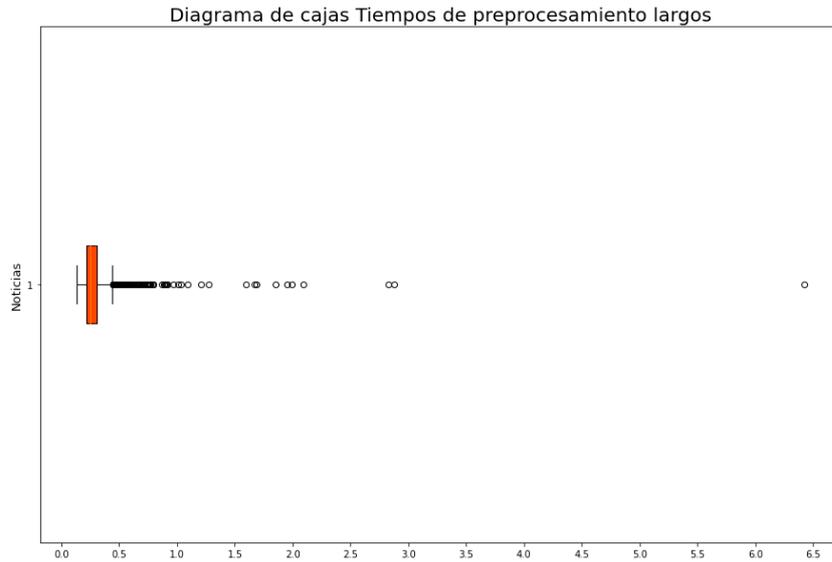


Gráfico 18-4. Diagrama de cajas, tiempos de respuesta noticias de longitud larga.

Realizado por: Jefferson Chariguamán, 2022

Tabla 27-4: Resumen de cuartiles noticias largas.

Cuartil	Valor
Q1	0.21207416057586664
Q2	0.2475438117980957
Q3	0.3049336075782776

Realizado por: Jefferson Chariguamán, 2022

Luego de haber identificado y aislado los tiempos atípicos, las noticias correspondientes a estos, son utilizadas para una segunda prueba, es decir se realizó el envío de peticiones con estas noticias, con el objetivo de obtener nuevamente los tiempos de respuesta, clasificación y preprocesamiento para analizar el comportamiento de estos entre ambas pruebas.

Los tiempos de respuesta de los valores atípicos son representados en el **Gráfico 19-4**, en el que los puntos de color rojo corresponden a la primera prueba y los de color verde a los tiempos de la segunda prueba. Como aprecia en el **Gráfico 19-4**, los tiempos en la segunda prueba son menores en su mayoría, respecto a los de la primera prueba. Para determinar si la variación de los tiempos de respuesta entre la primera y segunda prueba, es debido a los tiempos de los procesos internos de preprocesamiento y clasificación, también se proceden a analizar estos.

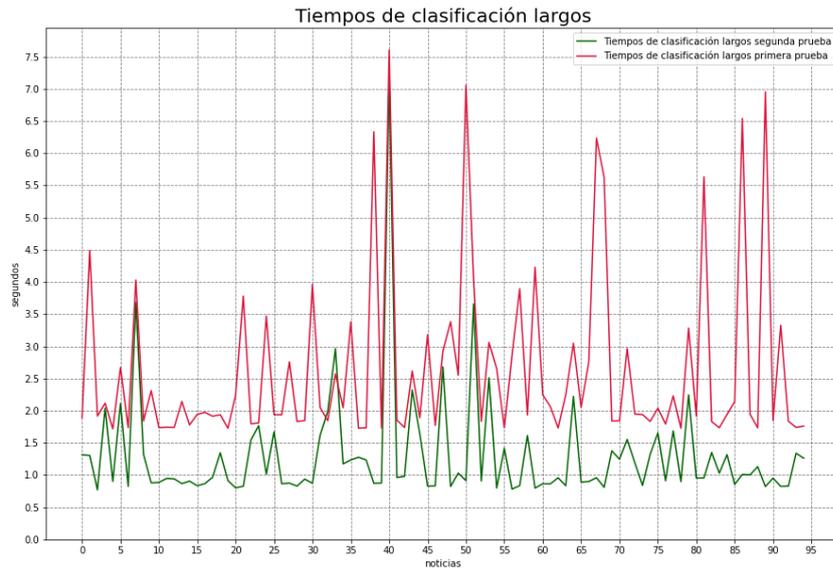


Gráfico 19-4. Tiempos de respuesta atípicos de noticias longitud larga.

Realizado por: Jefferson Chariguamán, 2022

La representación de los tiempos de clasificación y preprocesamiento correspondientes a los procesos internos del servicio web, se encuentran en el **Gráfico 20-4** y **Gráfico 21-4**, donde se observa que la variación entre la primera y segunda prueba, representados con los colores rojo y verde respectivamente, tienden a ser similares en su mayoría, con algunas excepciones. Por consiguiente, se determina que estos tiempos no influyen en la reducción de los tiempos de respuestas visualizados en el **Gráfico 19-4**.

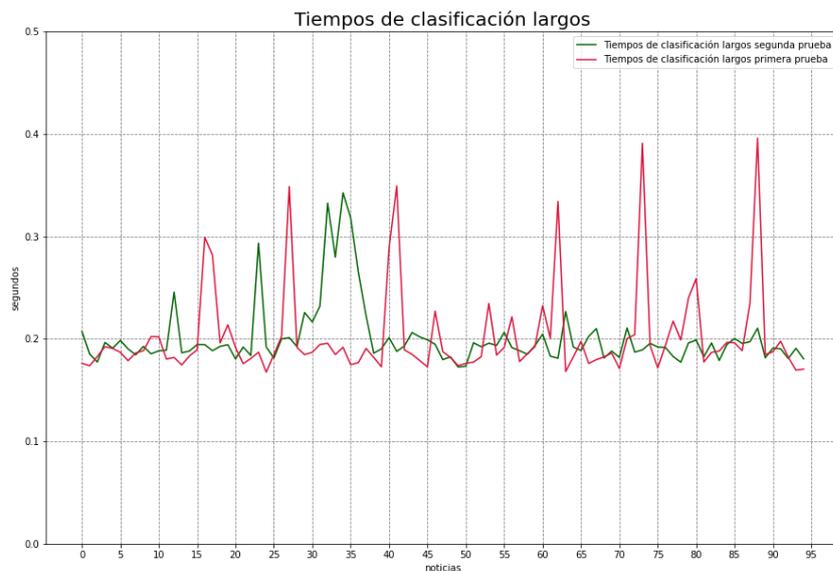


Gráfico 20-4. Tiempos de clasificación atípicos de noticias longitud larga.

Realizado por: Jefferson Chariguamán, 2022

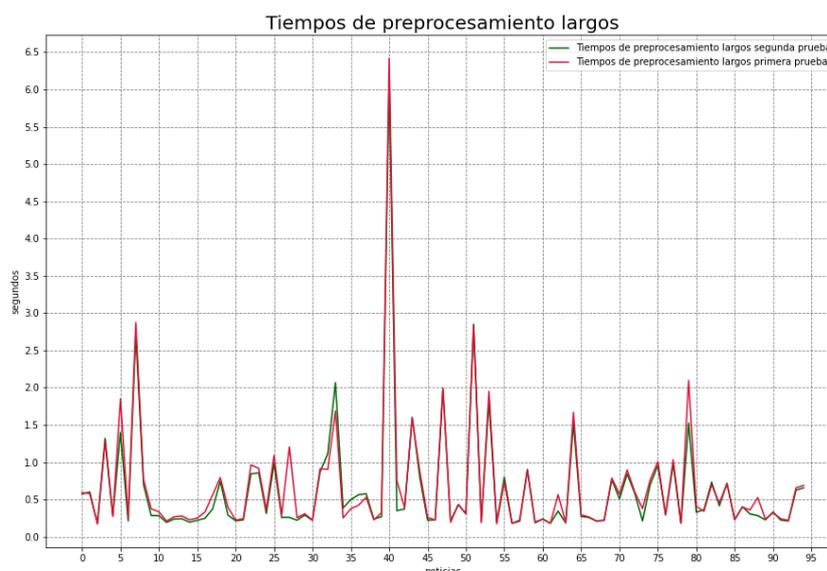


Gráfico 21-4. Tiempos de preprocesamiento atípicos de noticias longitud larga.

Realizado por: Jefferson Chariguamán, 2022

4.3.4 Tiempos de respuestas totales

Una vez identificados los valores atípicos en los tiempos de respuesta de las tres categorías de noticias (Ver **Tabla 18-4**) y tomando en cuenta que estos afectan el resultado del promedio final, como estrategia en el tratamiento de este tipo de datos, se realiza la eliminación de estos valores de las listas de tiempos de respuesta, mediante el script para el análisis de resultados se compara cada uno de los tiempos de la lista de tiempos de respuesta, con la lista de tiempos atípicos para la categoría respectiva, si el valor se encuentra en la lista de tiempos atípicos, es retirado de la lista de tiempos de respuesta, dando como resultado, listas de tiempos de respuesta sin valores atípicos. En el **Gráfico 22-4**, **Gráfico 23-4** y **Gráfico 24-4** se representan los tiempos de respuesta para las noticias de longitud, corta, media y larga respectivamente.

Debido a la cantidad de valores presentes en los gráficos, se dificulta visualizar los tiempos mínimo, promedio y máximo que son utilizados para el análisis, por lo tanto, utilizando el script para el análisis de resultados, se logra obtener estos valores a partir de las listas de tiempos de respuesta pertenecientes a noticias cortas, medias y largas. Estos valores se encuentran en la **Tabla 25-4**, **Tabla 26-4** y **Tabla 27-4**, para cada categoría de noticias respectivas.

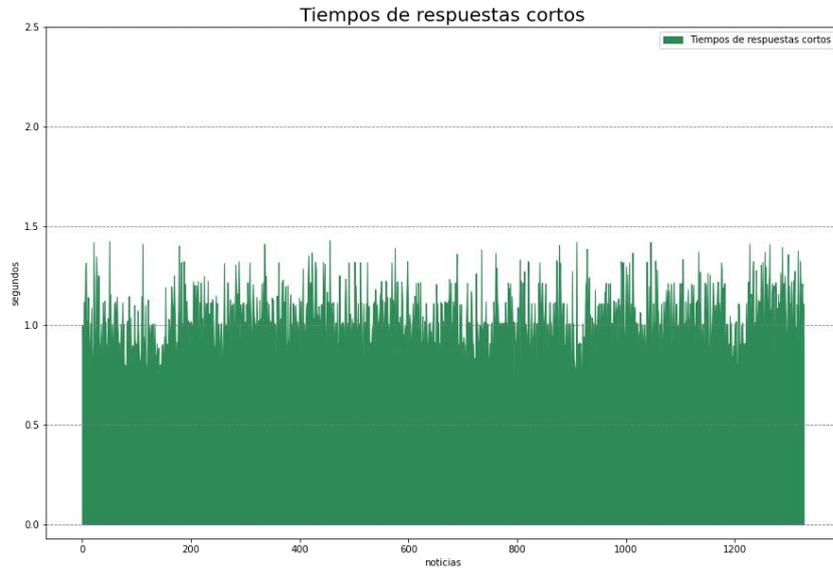


Gráfico 22-4. Tiempos de respuesta sin datos atípicos, noticias longitud corta.

Realizado por: Jefferson Chariguamán, 2022

Tabla 28-4: Tiempos de respuesta sin datos atípicos noticias cortas.

Tiempo mínimo	Tiempo promedio	Tiempo máximo
0.711474	1.0099205	1.427149

Realizado por: Jefferson Chariguamán, 2022

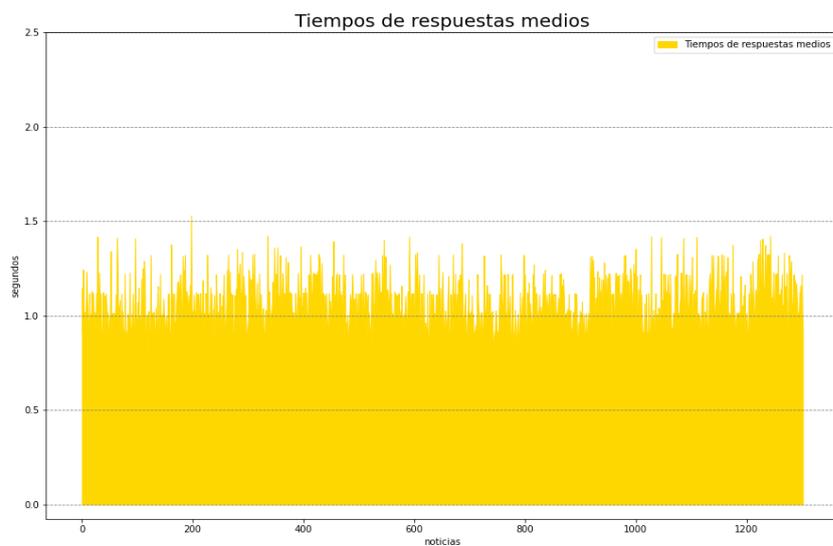


Gráfico 23-4. Tiempos de respuesta sin datos atípicos, noticias longitud media.

Realizado por: Jefferson Chariguamán, 2022

Tabla 29-4: Tiempos de respuesta sin datos atípicos noticias medias.

Tiempo mínimo	Tiempo promedio	Tiempo máximo
0.746878	1.018528	1.530575

Realizado por: Jefferson Chariguamán, 2022

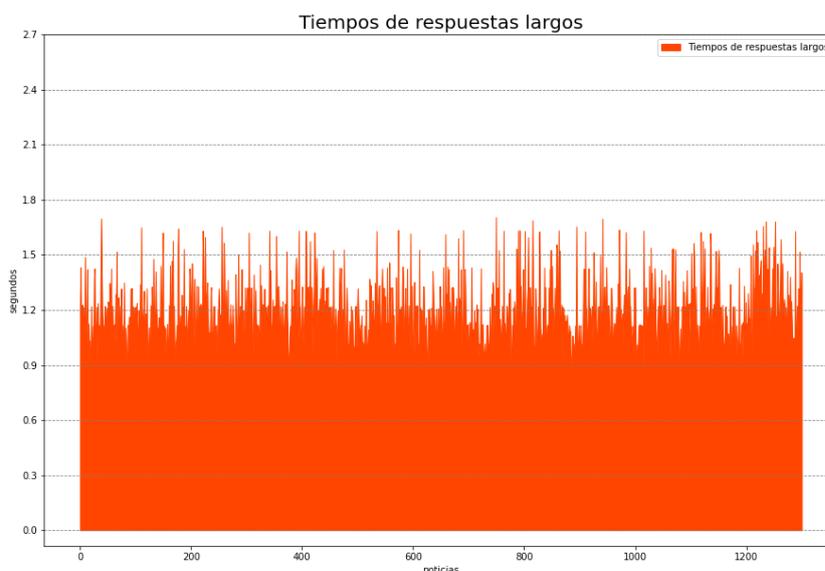


Gráfico 24-4. Tiempos de respuesta sin datos atípicos, noticias longitud larga.

Realizado por: Jefferson Chariguamán, 2022

Tabla 30-4: Tiempos de respuesta sin datos atípicos noticias largas.

Tiempo mínimo	Tiempo promedio	Tiempo máximo
0.815583	1.12832	1.705028

Realizado por: Jefferson Chariguamán, 2022

Después de calcular los tiempos de respuesta mínimo, promedio y máximo sin datos atípicos (Ver **Tabla 25-4**, **Tabla 26-4**, **Tabla 27-4**) para cada una de las categorías de noticias cortas, medias y largas, los tiempos mínimos se representan en el **Gráfico 25-4** mediante un diagrama de barras, donde se aprecia el incremento de los tiempos de respuesta en función de la categoría de la noticia. De la misma manera, los tiempos promedios sin datos atípicos, son representados mediante un diagrama de barras en el **Gráfico 26-4**, que tiene el mismo comportamiento que los tiempos mínimos, es decir se nota un incremento progresivo en el tiempo desde la categoría de noticias cortas hasta la de longitud larga. Por otro lado, en el **Gráfico 27-4** se grafican los resultados de tiempos máximos sin datos atípicos, donde también se muestra un incremento de los tiempos, dependiendo de la longitud del texto de la noticia.

Tomando en cuenta el comportamiento de los tiempos atípicos en los procesos internos de clasificación (Ver **Gráfico10-4**, **Gráfico 15-4**, **Gráfico 20-4**) y preprocesamiento (Ver **Gráfico 11-4**, **Gráfico 16-4**, **Gráfico 21-4**) de las categorías de noticias cortas, medias y largas, se llega a concluir que estos tiempos no alteran significativamente en los tiempos de respuestas(Ver **Gráfico 9-4**, **Gráfico 14-4**, **Gráfico 19-4**), sino que puede ser causado por tiempo de transferencia o latencia, es decir al tiempo que tarda en enviarse la petición desde el cliente hasta el servidor, y el de respuesta por parte del servidor hasta el cliente, sin tomar en cuenta el tiempo de los procesos de clasificación y preprocesamiento.

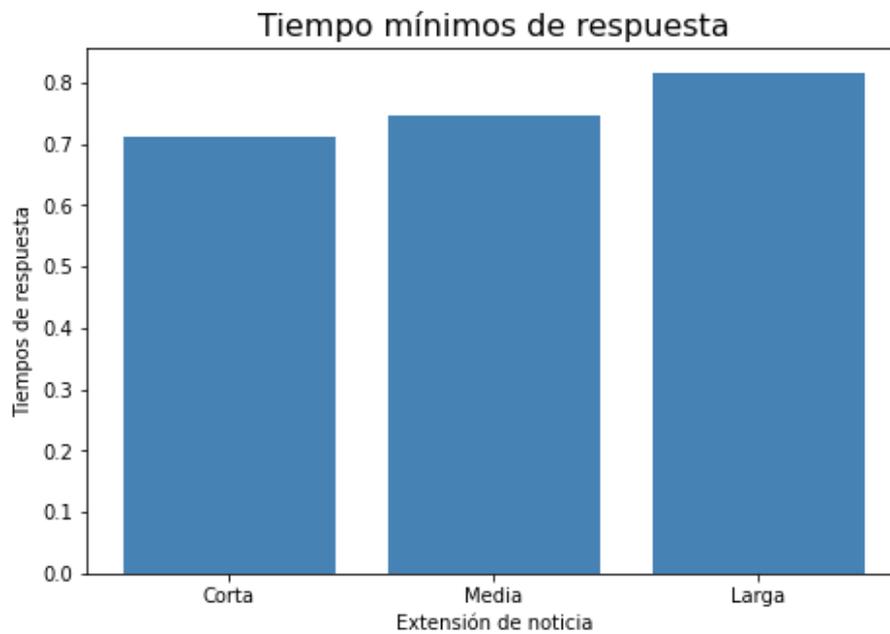


Gráfico 25-4. Comparación de tiempos mínimos de respuesta.

Realizado por: Jefferson Chariguamán, 2022

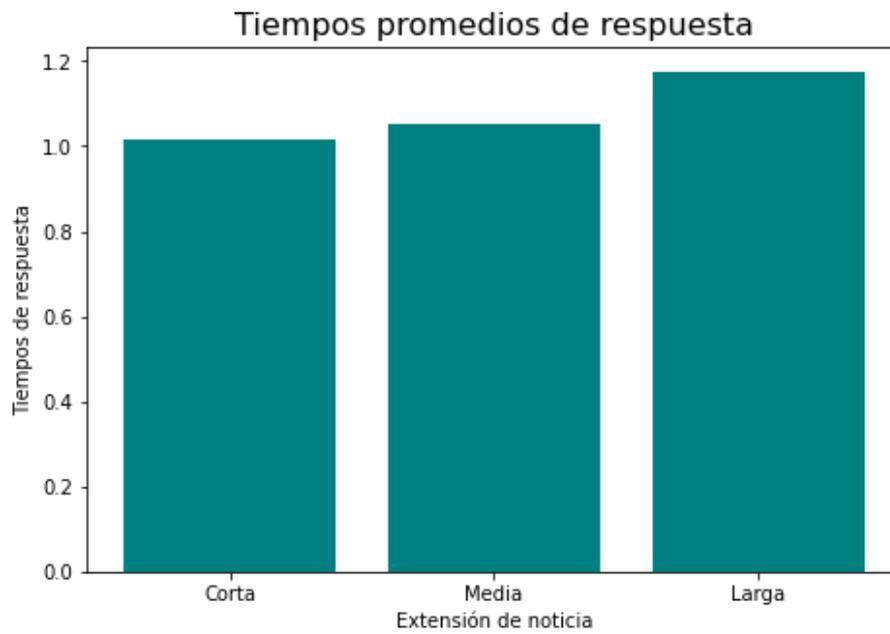


Gráfico 26-4. Comparación de tiempos promedios de respuesta.

Realizado por: Jefferson Chariguamán, 2022

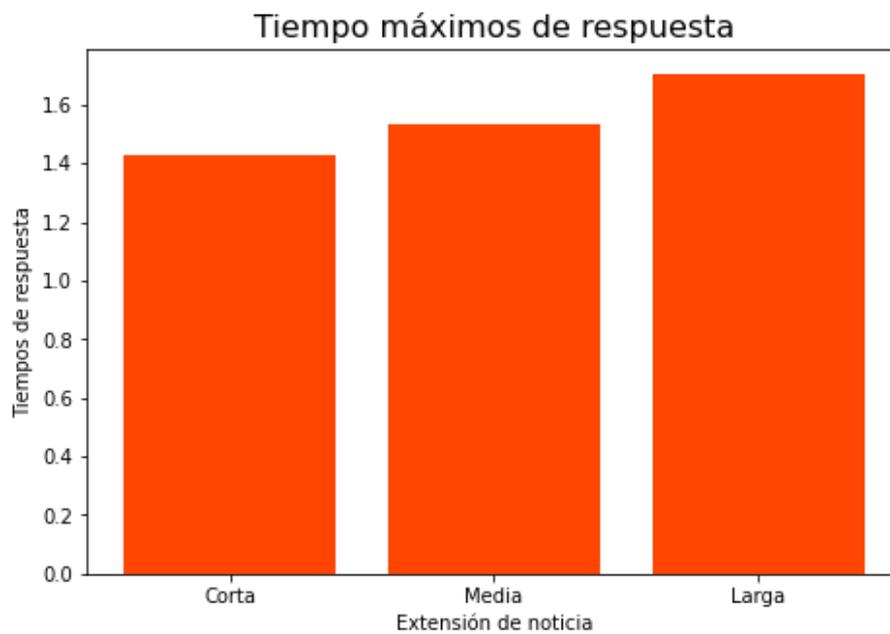


Gráfico 27-4. Comparación de tiempos máximos de respuesta.

Realizado por: Jefferson Chariguamán, 2022

Los tiempos de respuesta mínimos, promedios y máximos sin tiempos atípicos de las tres categorías de noticias, se resumen en **Tabla 28-4**, también se procede a calcular los tiempos promedios finales en base a los tiempos mencionados anteriormente, con el fin de determinar el

tiempo de respuesta promedio general que tiene el servicio web, para a continuación poder determinar el límite en el que se encuentra, según los tiempos mencionados por Nielsen.

Tabla 31-4: Resumen de tiempos finales de respuesta.

Categoría	Tiempos de respuesta		
	Mínimo	Promedio	Máximo
Corta	0.7114	1.0172	1.4271
Media	0.7468	1.0507	1.5305
Larga	0.8155	1.1749	1.7050
Total	0.7560	1.0809	1.5542

Realizado por: Jefferson Chariguamán, 2022

Según la(ISO/IEC 25023, 2016), en el enfoque de la medición de calidad, nota 3, menciona que algunas medidas de calidad como el tiempo de respuesta puede resultar difíciles de interpretar, por lo que proporciona 3 formas para aplicar las medidas de calidad, estas son:

- a) Conformidad: Comparar medidas con un negocio en específico. (Ejemplo, el tiempo de respuesta máximo aceptable es 0.5 segundos.)
- b) Benchmarks: Comparar medidas con un punto de referencia para un producto similar.
- c) Series de tiempo: Comparar tendencias a lo largo del tiempo.

Para este proyecto, se escoge la conformidad como forma de aplicar esta medida, considerando el tiempo disponible y la falta de un producto similar. Los límites de tiempos de referencia son los mencionados por Nielsen (2010) que divididos en 3 límites, el detalle de estos se muestran en la **Tabla 29-4**.

Tabla 32-4: Límites de tiempo de respuesta.

Límite	Tiempo (segundos)	Descripción
1	0.1	Da la sensación de respuesta instantánea y no sostienen ninguna interrupción.
2	0.2 -1	Se define como tiempo de respuesta aceptable, si excede este tiempo se puede interrumpir la experiencia del usuario.
3	1-10	Es el límite máximo de tiempo de respuesta aceptable, pasado de este tiempo el usuario abandonaría el sitio.

Fuente: Nielsen ,2010

Realizado por: Jefferson Chariguamán, 2022

Utilizando el tiempo de respuesta promedio general del servicio web de 1.080 descrito en la **Tabla 28-4** y tomando como referencia los límites de tiempos de respuesta de la **Tabla 29-4**, se realiza una comparación y se determina que el servicio web tiene un tiempo de respuesta de límite 3, sin embargo, mantiene una tendencia al límite 2, debido a que se supera por 0.08 segundos.

CONCLUSIONES

- Mediante la revisión de literatura referente a las técnicas de aprendizaje automático, se determina que los algoritmos tienen diferentes aplicaciones, sin embargo, solo algunos de estos son utilizados para la clasificación de textos. Por lo que se pudo determinar que las técnicas supervisadas más utilizadas en la clasificación de noticias falsas son: Máquinas de Soporte Vectorial, Árboles de decisión, K-vecino cercano, Regresión Logística, Naive Bayes y Redes Neuronales.
- Después del análisis de las técnicas seleccionadas para la clasificación de textos, se determina que los modelos de aprendizaje automático con mejor rendimiento utilizan Redes Neuronales, teniendo un rendimiento superior a técnicas como: Naive Bayes, Regresión logística, Máquinas de soporte vectorial, K-vecino cercano y Árboles de decisión. Además, mediante la revisión de trabajos relacionados se evidencia que en la actualidad las redes neuronales son ampliamente utilizadas para la clasificación de texto, debido a que arquitecturas como la LSTM, permiten tener un estado de memoria, manteniendo relación entre las secuencias de palabras del texto, lo que implica un mejor rendimiento al momento de realizar las clasificaciones.
- El sistema web desarrollado para la clasificación de noticias falsas en español, se realiza aplicando la metodología SCRUM, mediante 3 Sprints de duración de 40 horas cada uno, dentro de las cuales se plantearon 7 historias técnicas, y 2 historias de usuario, permitiendo cumplir los requerimientos establecidos. Por otro lado, el modelo de aprendizaje de máquina integrado en el servicio web es desarrollado mediante la metodología CRISP-DM, utilizando como técnica a las redes neuronales LSTM para la clasificación de noticias.
- Las métricas permiten evaluar el rendimiento de modelos entrenados de aprendizaje automático, la puntuación F1 es una de las métricas utilizadas para evaluar el rendimiento de modelos clasificadores, debido a que su cálculo implica los valores de la precisión y el recall. Se utilizó la puntuación F1 para evaluar modelos de redes neuronales con arquitectura MLP, CNN y LSTM para clasificar noticias falsas, resultando que la arquitectura LSTM posee un mejor rendimiento respecto a las demás con un valor de F1 de 0.746. Por lo tanto, este modelo entrenado fue integrado en el sistema web clasificador.
- El tiempo promedio de respuesta del servicio web desarrollado es de 1.080 segundos el cual se obtuvo después de realizar las pruebas con las noticias contenidas en el conjunto de datos de diferentes longitudes de texto, este tiempo se encuentra dentro del límite 3 según los límites

mencionados por Nielsen, en consecuencia, se logra determinar que la eficiencia del servicio web es buena y permite que el usuario interactúe con el sistema de forma correcta.

RECOMENDACIONES

- En el desarrollo de modelos de aprendizaje automático, tener en cuenta el tamaño del conjunto de datos para el entrenamiento de modelos, debido a que puede afectar el rendimiento de estos, además, para un clasificador binario, se recomienda que en el conjunto de datos sea balanceado, es decir, que exista un número similar de datos entre ambas clases.
- Se debe evaluar el comportamiento temporal como parte de la eficiencia en trabajos futuros que impliquen la clasificación de noticias, debido a que, en trabajos desarrollados, la evaluación de la eficiencia únicamente se enfoca al modelo de clasificación.
- Debido al avance constante en el área de aprendizaje automático, utilizar las nuevas arquitecturas para permitir un mejor rendimiento en la implementación de modelos para la clasificación de noticias falsas.

GLOSARIO

SVM: del inglés Support Vector Machine, algoritmo de machine Learning.

RF: del inglés Random Forest, algoritmo de machine Learning.

NB: del inglés Naive Bayes, algoritmo de machine Learning.

MLP: del inglés Multi Layer Perceptron, algoritmo de machine Learning correspondiente a redes neuronales.

CNN: del inglés Convolutional Neural Network, algoritmo de machine Learning correspondiente a redes neuronales

LSTM: del inglés Long Short Term Memory, algoritmo de machine Learning correspondiente a redes neuronales

AI: del inglés artificial intelligence, en español inteligencia artificial.

CPU: del inglés Central Unit Process, en español unidad central de procesamiento

RAM: del inglés Random Access Memory, en español Memoria de acceso aleatorio

URL: del inglés Uniform Resource Locator

API: del inglés Application Programming Interface

HTTP: del inglés HyperText Transfer Protocol, protocolo que permite la transferencia de hipermedia en la red.

Endpoint: es un extremo de un canal de comunicación, este puede incluir una URL.(What Is an API Endpoint?, 2022)

BIBLIOGRAFÍA

- AGUDELO, G.E.R., PARRA, O.J.S. y VELANDIA, J.B., 2018. Raising a Model for Fake News Detection Using Machine Learning in Python. En: S.A. AL-SHARHAN, A.C. SIMINTIRAS, Y.K. DWIVEDI, M. JANSSEN, M. MÄNTYMÄKI, L. TAHAT, I. MOUGHRABI, T.M. ALI y N.P. RANA (eds.), *Challenges and Opportunities in the Digital Era*. Cham: Springer International Publishing, pp. 596-604. ISBN 978-3-030-02131-3. DOI 10.1007/978-3-030-02131-3_52.
- AYALA, S., 2022. data/train-00000-of-00001.parquet · julien-c/impressionists at main. [en línea]. [Consulta: 4 enero 2023]. Disponible en: <https://huggingface.co/datasets/julien-c/impressionists/blob/main/data/train-00000-of-00001.parquet>.
- BAHAD, P., SAXENA, P. y KAMAL, R., 2019. Fake News Detection using Bi-directional LSTM-Recurrent Neural Network. *Procedia Computer Science*, vol. 165, pp. 74-82. ISSN 1877-0509. DOI 10.1016/j.procs.2020.01.072.
- BANDA, H., 2014. *Inteligencia Artificial: Principios y Aplicaciones*. S.l.: s.n.
- BASOGAIN, X., 1998. *REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES* [en línea]. 1998. S.l.: s.n. [Consulta: 29 abril 2022]. Disponible en: https://ocw.ehu.es/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf.
- BISONG, E., 2019. Introduction to Scikit-learn. En: E. BISONG (ed.), *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* [en línea]. Berkeley, CA: Apress, pp. 215-229. [Consulta: 20 abril 2022]. ISBN 978-1-4842-4470-8. Disponible en: https://doi.org/10.1007/978-1-4842-4470-8_18.
- CHAPELLE, O., VAPNIK, V., BOUSQUET, O. y MUKHERJEE, S., 2002. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, vol. 46, no. 1, pp. 131-159. ISSN 1573-0565. DOI 10.1023/A:1012450327387.
- CHAPMAN, P., CLINTON, J., KERBER, R., KHABAZA, T., REINARTZ, T., SHEARER, C. y WIRTH, R., 2000. CRISP-DM 1.0: Step-by-step data mining guide. *undefined* [en línea], [Consulta: 23 enero 2022]. Disponible en: <https://www.semanticscholar.org/paper/CRISP-DM-1.0%3A-Step-by-step-data-mining-guide-Chapman-Clinton/54bad20bbc7938991bf34f86dde0babfbd2d5a72>.
- CHAUMOND, J., 2022. sayalaruano/FakeNewsCorpusSpanish · Datasets at Hugging Face. [en línea]. [Consulta: 4 enero 2023]. Disponible en: <https://huggingface.co/datasets/sayalaruano/FakeNewsCorpusSpanish>.
- COHEN, M., 2013. Una Introducción a Scrum. , pp. 53.
- CORTINA, V.G., 2015. Aplicación de la Metodología CRISP-DM a un Proyecto de Minería de Datos en el Entorno Universitario. , pp. 120.
- Digital 2021. *We Are Social UK* [en línea], 2021. [Consulta: 10 febrero 2022]. Disponible en: <https://wearesocial.com/uk/blog/2021/01/digital-2021-uk/>.
- DINEVA, K. y ATANASOVA, T., 2020. SYSTEMATIC LOOK AT MACHINE LEARNING ALGORITHMS - ADVANTAGES, DISADVANTAGES AND PRACTICAL APPLICATIONS. . S.l.: s.n., DOI 10.5593/sgem2020/2.1/s07.041.

- DURÁN, M.B. y TUSA, F., 2019. La era de la desinformación y de las noticias falsas en el ambiente político ecuatoriano de transición: *#PerDebate*, vol. 3, pp. 18-41. ISSN 2697-3294. DOI 10.18272/pd.v3i1.1550.
- GABRIEL, E., 2015. Metodologías de desarrollo de software. , pp. 117.
- GÉRON, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd edition. Beijing China ; Sebastopol, CA: O'Reilly Media. ISBN 978-1-4920-3264-9.
- G.L, B., 2021. Google Colaboratory. [en línea]. [Consulta: 14 enero 2022]. Disponible en: https://colab.research.google.com/drive/1YKHLSIG-B9Ez2-zf-YFxXTVgfC_Aqt#scrollTo=pmECBCuVdu8N.
- GONZÁLEZ, M.P., DIAZ, A.M., CASAÑOLA, Y.T. y HIDALGO, D.B., 2021. Buenas prácticas para prevenir los riesgos de la eficiencia del desempeño en los productos de software. *Revista Cubana de Ciencias Informáticas*, vol. 15, no. 1, pp. 89-113.
- GUARDIOLA, C., 2020. *Clasificador de textos mediante técnicas de aprendizaje automático*. PhD Thesis. S.l.: Universitat Politècnica de València.
- GUTIERREZ, E.L., 2016. *IMPLEMENTACIÓN DE UN SERVICIO WEB PARA LA COMUNIDAD DE PROFESIONALES UNIVERSITARIOS DE COLOMBIA* [en línea]. Colombia: UNIVERSIDAD AUTÓNOMA DE OCCIDENTE. [Consulta: 20 abril 2022]. Disponible en: <https://red.uao.edu.co/bitstream/handle/10614/9477/T07147.pdf?sequence=1&isAllowed=y>.
- Introduction to Information Retrieval. [en línea], 2009. [Consulta: 23 enero 2022]. Disponible en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>.
- ISO 25000, 2016. NORMAS ISO 25000. [en línea]. [Consulta: 18 enero 2022]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000>.
- ISO/IEC 25023. [en línea], 2016. [Consulta: 24 mayo 2022]. Disponible en: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25023:ed-1:v1:en>.
- JAIMES, C.A., 2015. *OPTIMIZACIÓN DEL CLASIFICADOR "NAIVE BAYES" USANDO ÁRBOL DE DECISIÓN C4.5*. S.l.: s.n.
- Keras documentation: About Keras. *Team Keras* [en línea], 2022. [Consulta: 20 abril 2022]. Disponible en: <https://keras.io/about/>.
- Keras: the Python deep learning API. [en línea], 2022. [Consulta: 14 enero 2022]. Disponible en: <https://keras.io/>.
- KHALANE, T. y TANNER, M., 2013. Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. *2013 International Conference on Adaptive Science and Technology* [en línea]. S.l.: s.n., pp. 1-6. DOI 10.1109/ICASTech.2013.6707499. Disponible en: <https://ieeexplore.ieee.org/abstract/document/6707499>.
- KOWSARI, K., JAFARI MEIMANDI, K., HEIDARYSAFA, M., MENDU, S., BARNES, L., BROWN, D., ID, L., y BARNES, 2019. Text Classification Algorithms: A Survey. *Information (Switzerland)*, vol. 10. DOI 10.3390/info10040150.

- LARRANAGA, P., INZA, I. y MOUJAHID, A., 2022. Tema 8. Redes Neuronales. ,
- MARTIN, J. y JURAFSKY, D., 2021. *Speech and Language Processing* [en línea]. 3. S.l.: s.n. [Consulta: 13 abril 2022]. Disponible en: https://web.stanford.edu/~jurafsky/slp3/old_sep21/ed3book_sep212021.pdf.
- MARTÍNEZ-GALLEGO, K., ÁLVAREZ-ORTIZ, A.M. y ARIAS-LONDOÑO, J.D., 2021. *Fake News Detection in Spanish Using Deep Learning Techniques* [en línea]. 12 octubre 2021. S.l.: arXiv. [Consulta: 23 mayo 2022]. arXiv:2110.06461. Disponible en: <http://arxiv.org/abs/2110.06461>.
- MIRJALILI, V. y RASCHKA, S., 2020. *Python Machine Learning*. S.l.: Marcombo. ISBN 978-84-267-2772-5.
- MITCHELL, T.M., 1997. *Machine Learning*. New York: McGraw-Hill. McGraw-Hill series in computer science. ISBN 978-0-07-042807-2. Q325.5 .M58 1997
- MORENO, A., ARMENGOL, E., BÉJAR, A., BELANCHE, L., CORTÉS, C., GAVALDÁ, R., GIMENO, J., MARTÍN, M. y SÁNCHEZ, M., 1994. *Aprendizaje automático* [en línea]. Barcelona: Edicions UPC. [Consulta: 9 febrero 2022]. ISBN 978-84-7653-460-1. Disponible en: <http://hdl.handle.net/2099.3/36157>.
- NIELSEN, J., 2010. Website Response Times. *Nielsen Norman Group* [en línea]. [Consulta: 26 julio 2022]. Disponible en: <https://www.nngroup.com/articles/website-response-times/>.
- PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A. y COURNAPEAU, D., 2011. Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON*, vol. 12, pp. 6.
- PONCE, J., 2021. Conoce qué son las funciones de activación y cómo puedes crear tu función de activación usando Python, R y Tensorflow. *Jahaziel Ponce* [en línea]. [Consulta: 30 abril 2022]. Disponible en: <https://jahazielponce.com/funciones-de-activacion-y-como-puedes-crear-la-tuya-usando-python-r-y-tensorflow/>.
- PRADA, J.D., 2019. *NATURAL LANGUAGE PROCESSING COMO HERRAMIENTA PARA DETECTAR FAKE NEWS: APLICADO A LAS ELECCIONES DE ALCALDÍA DE CALI PARA EL PERIODO 2020-2023* [en línea]. 2019. S.l.: s.n. [Consulta: 9 abril 2022]. Disponible en: https://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/85367/1/TG02611.pdf.
- PRANCKEVIČIUS, T. y MARCINKEVIČIUS, V., 2017. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, vol. 5, no. 2, pp. 221.
- ¿Qué es REST? Conoce su potencia. *OpenWebinars.net* [en línea], 2018. [Consulta: 2 mayo 2022]. Disponible en: <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>.
- ¿Qué es una API de REST? [en línea], 2022. [Consulta: 2 mayo 2022]. Disponible en: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.
- Qué son las redes neuronales y sus funciones. *ATRIA Innovation* [en línea], 2019. [Consulta: 29 abril 2022]. Disponible en: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>.

- RAMÍREZ, D., 2018. EL MACHINE LEARNING A TRAVÉS DE LOS TIEMPOS, Y LOS APORTES A LA HUMANIDAD. , pp. 17.
- RASHID, J., MAHMOOD, T., NISAR, M.W. y NAZIR, T., 2020. Phishing Detection Using Machine Learning Technique. *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*. S.l.: s.n., pp. 43-46. DOI 10.1109/SMART-TECH49988.2020.00026.
- Regresión Logística para Clasificación - IArtificial.net. [en línea], 2019. [Consulta: 13 abril 2022]. Disponible en: <https://www.iartificial.net/regresion-logistica-para-clasificacion/>.
- ROCHE, J., 2022. Artefactos Scrum: las 3 herramientas clave de gestión. *Deloitte Spain* [en línea]. [Consulta: 11 febrero 2022]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>.
- ROJAS, E.M. 1 1 F. de C. y A., 2020. Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. , pp. 586-599. ISSN 16469895.
- ROKACH, L. y MAIMON, O., 2005. Decision Trees. En: O. MAIMON y L. ROKACH (eds.), *Data Mining and Knowledge Discovery Handbook* [en línea]. Boston, MA: Springer US, pp. 165-192. [Consulta: 19 abril 2022]. ISBN 978-0-387-25465-4. Disponible en: https://doi.org/10.1007/0-387-25465-X_9.
- ROUHIAINEN, L., 2018. *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. S.l.: s.n. ISBN 978-84-17568-08-5.
- RT EN ESPAÑOL, 2021. Facebook conocía la campaña de manipulación del presidente de Honduras y permitió que continuara casi un año. *RT en Español* [en línea]. [Consulta: 2 agosto 2022]. Disponible en: <https://actualidad.rt.com/actualidad/389280-facebook-permitir-campana-manipulacion-presidente-honduras>.
- RUMELHART, D.E., MCCLELLAND, J.L. y GROUP, T.P.R., 1986. *Psychological and Biological Models*. Cambridge, Mass: s.n. ISBN 978-0-262-13218-3.
- SARKAR, D., BALI, R. y SHARMA, T., 2018. *Practical Machine Learning with Python* [en línea]. Berkeley, CA: Apress. [Consulta: 14 enero 2022]. ISBN 978-1-4842-3206-4. Disponible en: <http://link.springer.com/10.1007/978-1-4842-3207-1>.
- Scrum Guide | Scrum Guides. [en línea], 2020. [Consulta: 21 abril 2022]. Disponible en: <https://scrumguides.org/scrum-guide.html>.
- sklearn.metrics.f1_score. *scikit-learn* [en línea], 2022. [Consulta: 2 mayo 2022]. Disponible en: https://scikit-learn/stable/modules/generated/sklearn.metrics.f1_score.html.
- STACKOVERFLOW CONTRIBUTORS, 2022. keras eBook. *StackOverflow Contributors* [en línea]. [Consulta: 20 abril 2022]. Disponible en: <https://riptutorial.com/ebook/keras>.
- TensorFlow. *TensorFlow* [en línea], 2022. [Consulta: 14 enero 2022]. Disponible en: <https://www.tensorflow.org/?hl=es-419>.
- TRETIAKOV, A., 2021. noticias falsas en español. [en línea]. [Consulta: 4 enero 2023]. Disponible en: <https://www.kaggle.com/datasets/arseniitretiakov/noticias-falsas-en-espaol>.

WANJUN, Y. y XIAOGUANG, S., 2010. Research on text categorization based on machine learning. *2010 IEEE International Conference on Advanced Management Science(ICAMS 2010)*. S.l.: s.n., pp. 253-255. DOI 10.1109/ICAMS.2010.5552917.

Web Services Architecture. [en línea], 2004. [Consulta: 9 febrero 2022]. Disponible en: <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>.

Web services: servicios de máquina a máquina. *IONOS Digitalguide* [en línea], 2020. [Consulta: 9 abril 2022]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/web-services/>.

What Is an API Endpoint? *Kinsta®* [en línea], 2022. [Consulta: 2 agosto 2022]. Disponible en: <https://kinsta.com/knowledgebase/api-endpoint/>.

ZULES, F., 2019. Spanish Fake and Real News. [en línea]. [Consulta: 4 enero 2023]. Disponible en: <https://www.kaggle.com/datasets/zulanac/fake-and-real-news>.

ANEXOS

ANEXO A: Manual de Usuario

Introducción

Este documento tiene como finalidad brindar una guía de uso para el usuario, dando las pautas generales para el uso correcto y la forma de interpretar los resultados.

Usuarios del sistema

Los usuarios a los cuales está destinado el servicio web, son las personas con la necesidad de verificar si una noticia es verdadera o falsa.

1. Funcionalidad del sistema

1.1. Funcionamiento mediante la interfaz web



ANEXO 1. Interfaz web del servicio.

Realizado por: Jefferson Chariguamán, 2022

A continuación, se describe cada apartado de la interfaz de usuario del servicio web, donde se le asignan un número, estos se detallan en la siguiente tabla.

Punto	Descripción
1	En esta área se permite el ingreso del texto de la noticia a ser clasificada.
2	Botón para enviar la noticia a ser clasificada, una vez que el texto haya sido ingresado.
3	En este apartado se muestran los resultados de la clasificación de la noticia.
4	Aquí se indican la manera de interpretar los resultados devueltos por el servicio web.

5	En este apartado se indica la información acerca del sitio web.
---	---

Los pasos para utilizar el servicio web mediante la interfaz son:

1. Ingresar el texto de la noticia
2. Presionar el botón de “Enviar noticia”
3. Interpretar los resultados devueltos

1.2. Funcionamiento mediante los endpoint

Para el uso del servicio web mediante el endpoint de la API, se debe enviar una petición POST al enlace https://clasificadorfakenews.azurewebsites.net/get_classification, donde se envía en el body un json, con la siguiente estructura.

Clave	Valor
texto	Texto de la noticia
Ejemplo: 'Texto': 'Gran sorpresa se llevó la hija de Donald Trump, la socialité y empresaria Ivanka Trump, luego de que una persona le enviara solicitud y cientos de mensajes a su cuenta ultra personal de Facebook, la cual solo se la da a personas conocidas.'	

Como respuesta, el servicio web devuelve un json con la siguiente información.

Clave	Valor
clase	En texto la categoría <i>Real</i> o <i>Falso</i>
predicted	El valor de la predicción del modelo, donde la tendencia a 0 se considera falso y a 1 es verdadero.
time_preprocess	El tiempo de preprocesamiento realizado al texto antes de que el modelo realice la clasificación.
time_predict	El tiempo que demora el modelo en realizar la clasificación de la noticia.

ANEXO B: Encuestas realizadas.

ANEXO C: Historias técnicas, tareas de ingeniería y pruebas de aceptación.

Historia de técnica	
Código: HT1	Nombre: Definición de una arquitectura para el servicio web.
Usuario: Programador	Sprint Asignado: 1
Fecha de inicio: 01/05/2022	Fecha de fin: 02/05/2022
Puntos estimados: 10	Puntos reales: 10
Descripción: Como programador se requiere determinar la arquitectura del servicio web, considerando que se realizará la integración con un modelo entrenado.	
Observaciones: La arquitectura del sistema tiene que tener un enfoque donde se expongan los servicios para la clasificación de noticias.	
Pruebas de aceptación Verificar que la arquitectura permite exponer sus servicios	

Tarea de ingeniería	
Identificador: TI_HT1	Nombre de la historia: Definición de una arquitectura para el servicio web.
Nombre de tarea: Análisis y selección de arquitectura para servicio web.	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 01/05/2022	Fecha de fin: 02/05/2022
Descripción: Análisis y selección de una arquitectura para el servicio web.	
Pruebas de aceptación: Verificar que la arquitectura seleccionada permita exponer el servicio de clasificación	

Prueba de aceptación	
Identificador: PA1_TI_HT1	Nombre de la historia: Definición de una arquitectura para el servicio web.
Nombre la tarea: Análisis y selección de arquitectura para servicio web	
Responsable: Jefferson Chariguamán	Fecha: 2/05/2022
Descripción: Se verifica que la arquitectura seleccionada luego del análisis y revisión de arquitecturas, sea adecuada y permita el desarrollo del servicio web.	
Condiciones de Ejecución: Conocer los requerimientos del servicio web	

Pasos de ejecución:
<ul style="list-style-type: none"> • Revisión de literatura • Selección de la arquitectura • Verificar si la arquitectura permite cubrir los requerimientos del servicio web
Resultado esperado: Arquitectura adecuada para el desarrollo del servicio web
Evaluación de la prueba: Exitosa.

Historia de técnica	
Código: HT2	Nombre: Análisis y selección de las herramientas para el desarrollo del servicio web.
Usuario: Programador	Sprint Asignado: 1
Fecha de inicio: 03/05/2022	Fecha de fin: 05/05/2022
Puntos estimados: 10	Puntos reales: 10
Descripción: Como programador se requiere realizar un análisis de las herramientas disponibles en el mercado, y la selección de estas para el desarrollo del servicio web.	
Observaciones: Hay que considerar las herramientas que en trabajos anteriores se hayan utilizado para la integración con modelos de aprendizaje automáticos.	
Pruebas de aceptación: Verificar que las herramientas seleccionadas permitan el desarrollo del servicio web e incorporar el modelo entrenado.	

Tarea de ingeniería	
Identificador: TI_HT2	Nombre de la historia: Análisis y selección de las herramientas para el desarrollo del servicio web.
Nombre de tarea: Analizar las herramientas de desarrollo	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 03/05/2022	Fecha de fin: 05/05/2022
Descripción: Analizar las herramientas disponibles para el desarrollo de un servicio web, considerando que permita la implementación de un modelo de aprendizaje automático.	
Pruebas de aceptación:	
Verificar que las herramientas permitan desarrollar la arquitectura seleccionada.	

Prueba de aceptación	
Identificador: PA1_TI_HT2	Nombre de la historia: Análisis y selección de las herramientas para el desarrollo del servicio web.
Nombre la tarea: Analizar las herramientas de desarrollo	
Responsable: Jefferson Chariguamán	Fecha: 05/05/2022
Descripción: Se verificará que las herramientas seleccionadas permitan exponer el servicio de clasificación y además de permitir la integración de un modelo entrenado.	
Condiciones de Ejecución: Tener un modelo entrenado para encontrar herramientas compatibles y tener en cuenta la arquitectura del servicio.	
Pasos de ejecución: <ul style="list-style-type: none"> • Realizar revisión de literatura y búsqueda en internet al fin de encontrar herramientas adecuadas • Verificar que las herramientas seleccionadas sean compatibles con la arquitectura y el modelo 	
Resultado esperado: Conjunto de herramientas que permitan el desarrollo del servicio web	
Evaluación de la prueba: Exitosa.	

Historia de técnica	
Código: HT3	Nombre: Desarrollo de los endpoints definidos para procesar las peticiones.
Usuario: Programador	Sprint Asignado: 1
Fecha de inicio: 06/05/2022	Fecha de fin: 15/05/2022
Puntos estimados: 20	Puntos reales: 20
Descripción: Como fase de desarrollo de la arquitectura seleccionada, se requiere la programación de los endpoints para exponer los servicios del sistema.	
Observaciones:	
Pruebas de aceptación: Los endpoints desarrollados deben responder las peticiones GET para responder la interfaz de usuario y POST para exponer el servicio de clasificación.	

Tarea de ingeniería	
Identificador: TI_HT3	Nombre de la historia: Desarrollo de los endpoints definidos para procesar las peticiones.
Nombre de tarea: Programación del endpoint para peticiones GET	

Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 06/05/2022	Fecha de fin: 11/05/2022
Descripción: Realizar la programación del endpoint para procesar peticiones GET, enviando un documento HTML con la interfaz de usuario para la interacción con el usuario.	
Pruebas de aceptación: Verificar que el endpoint devuelva el archivo con la interfaz de usuario en las peticiones GET.	

Prueba de aceptación	
Identificador: PA1_TI_HT3	Nombre de la historia: Desarrollo de los endpoints definidos para procesar las peticiones.
Nombre la tarea: Programación del endpoint para peticiones GET	
Responsable: Jefferson Chariguamán	Fecha: 11/05/2022
Descripción: Se verificará que el servicio web procese y responda las peticiones GET.	
Condiciones de Ejecución: Tener el servidor en ejecución y habilitado para recibir las peticiones además de tener desarrollado el endpoint para peticiones GET.	
Pasos de ejecución: <ul style="list-style-type: none"> • Enviar una petición mediante el método GET mediante el navegador accediendo a la URL/IP del servidor. • Verificar en el navegador la carga de la interfaz de usuario. 	
Resultado esperado: Interfaz de usuario cargada en el navegador	
Evaluación de la prueba: Exitosa.	

Tarea de ingeniería	
Identificador: TI2_HT3	Nombre de la historia: Desarrollo de los endpoints definidos para procesar las peticiones.
Nombre de tarea: Programación del endpoint para peticiones POST	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 12/05/2022	Fecha de fin: 15/05/2022
Descripción: Realizar la programación del endpoint para procesar peticiones POST, enviando la respuesta en formato Json donde se establece la información luego de procesar la noticia.	
Pruebas de aceptación: Verificar que el endpoint devuelva correctamente la respuesta a las peticiones POST	

Prueba de aceptación	
Identificador: PA1_TI2_HT3	Nombre de la historia: Desarrollo de los endpoints definidos para procesar las peticiones.
Nombre la tarea: Programación del endpoint para peticiones POST	
Responsable: Jefferson Chariguamán	Fecha: 15/05/2022
Descripción: Se verificará que el servicio web procese y responda las peticiones POST.	

Condiciones de Ejecución: Tener el servidor en ejecución y habilitado para recibir las peticiones además de tener desarrollado el endpoint para peticiones POST.
Pasos de ejecución: <ul style="list-style-type: none"> • Enviar una petición mediante el método POST mediante POSTMAN. • Verificar la respuesta recibida por parte del servidor en POSTMAN.
Resultado esperado: Respuesta exitosa del servidor al cliente.
Evaluación de la prueba: Exitosa.

Historia de técnica	
Código: HT4	Nombre: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Usuario: Programador	Sprint Asignado: 2
Fecha de inicio: 16/05/2022	Fecha de fin: 22/05/2022
Puntos estimados: 20	Puntos reales: 20
Descripción: Como parte del desarrollo del modelo de aprendizaje automático, es necesario realizar un preprocesamiento del conjunto de datos, para ser utilizados posteriormente en la fase de entrenamiento.	
Observaciones	
Pruebas de aceptación: El conjunto de datos debe contener los textos en un rango determinado de palabras, en los cuales han sido preprocesados aplicando técnicas de lematización, stemming, eliminación de stopwords y de signos, que son considerados ruido.	

Tarea de ingeniería	
Identificador: TI_HT4	Nombre de la historia: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Nombre de tarea: Limpieza de datos	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 16/05/2022	Fecha de fin: 17/05/2022
Descripción: Realizar la limpieza de datos, eliminando las noticias con problemas de extensión y fuente.	
Pruebas de aceptación: Verificar que las noticias se encuentren en el rango establecido en cuanto a su extensión.	

Prueba de aceptación	
Identificador: PA_TI_HT4	Nombre de la historia: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Nombre la tarea: Limpieza de datos	
Responsable: Jefferson Chariguamán	Fecha: 17/05/2022
Descripción: Se verificará que el conjunto de datos mantenga sólo las noticias con extensión adecuada.	
Condiciones de Ejecución: Mantener el conjunto de datos unificado, en formato dataframe para la verificación de cada noticia.	

Pasos de ejecución: <ul style="list-style-type: none"> Recorrer cada una de las noticias del conjunto de datos y extraer su extensión Verificar que la extensión de la noticia sea la establecida
Resultado esperado: Noticias con extensión dentro del rango establecido
Evaluación de la prueba: Exitosa.

Tarea de ingeniería	
Identificador: TI2_HT4	Nombre de la historia: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Nombre de tarea: Eliminar caracteres extraños, símbolos y stopwords	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 18/05/2022	Fecha de fin: 19/05/2022
Descripción: Eliminar los símbolos, caracteres extraños y stopwords considerados como ruidos, de los textos de las noticias del conjunto de datos.	
Pruebas de aceptación: Verificar que los textos se encuentren sin caracteres extraños, stopwords y símbolos.	

Prueba de aceptación	
Identificador: PA2_T2_HT4	Nombre de la historia: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Nombre la tarea: Limpieza de datos	
Responsable: Jefferson Chariguamán	Fecha: 19/05/2022
Descripción: Se verificará que los textos de cada noticia en el conjunto de datos, se encuentren sin caracteres extraños, símbolos y stopwords.	
Condiciones de Ejecución: Mantener el conjunto de datos unificado, en formato dataframe para la verificación de cada noticia	
Pasos de ejecución: <ul style="list-style-type: none"> Recorrer cada una de las noticias del conjunto de datos y verificar que no haya presencia de stopwords, símbolos y caracteres extraños. 	
Resultado esperado: Texto de las noticias del conjunto de datos sin stopwords, símbolos y caracteres extraños.	
Evaluación de la prueba: Exitosa.	

Tarea de ingeniería	
Identificador: TI3_HT4	Nombre de la historia: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Nombre de tarea: Lematización y stemming	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 20/05/2022	Fecha de fin: 22/05/2022

Descripción: Aplicar las técnicas de lematización y stemming a los textos de las noticias, para volver a las palabras a su forma base
Pruebas de aceptación: Verificar que las palabras en los textos del conjunto de datos hayan sido lematizadas y vueltas a su forma base.

Prueba de aceptación	
Identificador: PA3_ T3_HT4	Nombre de la historia: Preprocesamiento del conjunto de datos para el entrenamiento del modelo.
Nombre la tarea: Limpieza de datos	
Responsable: Jefferson Chariguamán	Fecha: 22/05/2022
Descripción: Se verificará que las palabras del texto de las noticias hayan sido lematizadas y vueltas a su forma básica.	
Condiciones de Ejecución: Mantener el conjunto de datos unificado, en formato dataframe para la verificación de cada noticia.	
Pasos de ejecución: <ul style="list-style-type: none"> Recorrer cada una de las noticias del conjunto de datos. Verificar que las palabras hayan sido lematizadas y vueltas a su forma base. 	
Resultado esperado: Las palabras de los textos fueron lematizados y en su forma base, con excepciones.	
Evaluación de la prueba: Exitosa.	

Historia de técnica	
Código: HT5	Nombre: Codificación y entrenamiento del modelo con la arquitectura seleccionada.
Usuario: Programador	Sprint Asignado: 2
Fecha de inicio: 23/05/2022	Fecha de fin: 30/05/2022
Puntos estimados: 20	Puntos reales: 20
Descripción: Como parte del área de aprendizaje automático, se requiere codificar el modelo y realizar el entrenamiento para su posterior implementación dentro del servicio web.	
Observaciones	
Pruebas de aceptación: El modelo debe ser capaz de procesar los datos de entrada y devolver el resultado con la clasificación en base a la información ingresada.	

Tarea de ingeniería	
Identificador: TI_HT5	Nombre de la historia: Codificación y entrenamiento del modelo con la arquitectura seleccionada.
Nombre de tarea: Codificación de la arquitectura seleccionada para el modelo de clasificación	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 23/05/2022	Fecha de fin: 25/05/2022

Descripción: Codificar la arquitectura seleccionada para la clasificación de texto, teniendo en cuenta el uso de embedding.
Pruebas de aceptación: Verificar que la arquitectura sea la correcta para realizar la clasificación de noticias considerando el uso de embedding.

Prueba de aceptación	
Identificador: PA1_TI_HT5	Nombre de la historia: Codificación y entrenamiento del modelo con la arquitectura seleccionada.
Nombre la tarea: Limpieza de datos	
Responsable: Jefferson Chariguamán	Fecha: 25/05/2022
Descripción: Se verificará que el modelo cumpla con la arquitectura seleccionada	
Condiciones de Ejecución: Tener la arquitectura del modelo definida, además de tener el conjunto de datos preprocesado.	
Pasos de ejecución: <ul style="list-style-type: none"> • Verificar que la arquitectura de la red neuronal sea la seleccionada • Verificar su compilación 	
Resultado esperado: Noticias con extensión dentro del rango establecido	
Evaluación de la prueba: Exitosa.	

Tarea de ingeniería	
Identificador: TI2_HT5	Nombre de la historia: Codificación y entrenamiento del modelo con la arquitectura seleccionada.
Nombre de tarea: Entrenamiento del modelo para la clasificación	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 26/05/2022	Fecha de fin: 30/05/2022
Descripción: Utilizar el modelo codificado y compilado, para realizar el entrenamiento con el conjunto de datos de prueba, teniendo en cuenta las métricas de la puntuación F1 y la función de pérdida.	
Pruebas de aceptación: Verificar que el modelo se guarde considerando el valor menor de la función de pérdida.	

Prueba de aceptación	
Identificador: PA2_TI2_HT5	Nombre de la historia: Codificación y entrenamiento del modelo con la arquitectura seleccionada.
Nombre la tarea: Entrenamiento del modelo para la clasificación.	
Responsable: Jefferson Chariguamán	Fecha: 30/05/2022
Descripción: Se verificará que el modelo cumpla con la arquitectura seleccionada	

Condiciones de Ejecución: Tener la arquitectura definida del modelo y compilada, además de tener el conjunto de datos preprocesados para el entrenamiento.
Pasos de ejecución: <ul style="list-style-type: none"> • Separar el conjunto de datos en conjunto de entrenamiento y validación • Utilizar los conjuntos para entrenar la red, programando el guardado automático en cada época, considerando la función de pérdida como parámetro para guardar. • Verificar que el modelo guardado haya sido el que tenga menor función de pérdida.
Resultado esperado: Modelo entrenado para la clasificación.
Evaluación de la prueba: Exitosa.

Historia de técnica	
Código: HT6	Nombre: Integración del modelo entrenado para la clasificación de noticias.
Usuario: Programador	Sprint Asignado: 3
Fecha de inicio: 31/05/2022	Fecha de fin: 03/06/2022
Puntos estimados: 10	Puntos reales: 10
Descripción: Como parte de la arquitectura del servicio web, es necesario incorporar el modelo entrenado para realizar la clasificación de noticias.	
Observaciones	
Pruebas de aceptación: El modelo entrenado debe funcionar correctamente dentro del servicio web y en conjunto con los endpoints para responder las peticiones.	

Tarea de ingeniería	
Identificador: TI_HT6	Nombre de la historia: Integración del modelo entrenado para la clasificación de noticias.
Nombre de tarea: Programación de la función de preprocesamiento	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 31/06/2022	Fecha de fin: 01/06/2022
Descripción: Importar las librerías adecuadas para realizar el preprocesamiento del texto para posteriormente enviar el texto preprocesado al modelo entrenado y desarrollar la función de preprocesamiento.	
Pruebas de aceptación: Verificar que el texto sea preprocesado correctamente para el ingreso al modelo de aprendizaje automático.	

Prueba de aceptación	
Identificador: PA1_HT6	Nombre de la historia: Integración del modelo entrenado para la clasificación de noticias.
Nombre la tarea: Programación de la función de preprocesamiento	
Responsable: Jefferson Chariguamán	Fecha: 01/06/2022
Descripción: Se verificará que el texto recibido mediante las peticiones POST, sea preprocesado correctamente y listo para servir como entrada hacia el modelo entrenado.	

Condiciones de Ejecución: Tener cargadas las librerías para el preprocesamiento de texto y los endpoints funcionando para recibir peticiones
Pasos de ejecución: <ul style="list-style-type: none"> • Enviar una petición mediante el método POST con el texto de la noticia. • Recibir el texto de la noticia y preprocesar. • Verificar que el texto haya sido preprocesado, encontrándose en el formato adecuado para el ingreso hacia el modelo de entrenado.
Resultado esperado: Texto preprocesado
Evaluación de la prueba: Exitosa.

Tarea de ingeniería	
Identificador: TI2_HT6	Nombre de la historia: Integración del modelo entrenado para la clasificación de noticias.
Nombre de tarea: Incorporación del modelo dentro del servicio web	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 02/06/2022	Fecha de fin: 03/06/2022
Descripción: Cargar el modelo de machine Learning, y programar la función donde se interactúa con el modelo para enviarle las entradas y recibir la respuesta de clasificación, para el envío en la respuesta a las peticiones POST.	
Pruebas de aceptación: Verificar que el texto sea preprocesado correctamente para el ingreso al modelo de aprendizaje automático.	

Prueba de aceptación	
Identificador: PA2_TI2_HT6	Nombre de la historia: Integración del modelo entrenado para la clasificación de noticias.
Nombre la tarea: Incorporación del modelo dentro del servicio web	
Responsable: Jefferson Chariguamán	Fecha: 03/06/2022
Descripción: Se carga el modelo entrenado para integrar dentro de una función de clasificación en el servicio web.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Endpoints funcionando correctamente para recibir peticiones. • Función de preprocesamiento ejecutando correctamente. • Librerías para la manipulación de modelos de machine Learning cargadas. 	
Pasos de ejecución: <ul style="list-style-type: none"> • Importar las librerías • Ubicar el modelo entrenado y realizar la carga • Realizar pruebas donde se obtenga la clasificación del texto mediante el modelo 	
Resultado esperado: Respuesta de clasificación del texto por parte del modelo	
Evaluación de la prueba: Exitosa.	

Historia de técnica	
Código: HT7	Nombre: Realizar el manual de usuario del servicio web
Usuario: Programador	Sprint Asignado: 3
Fecha de inicio: 04/06/2022	Fecha de fin: 07/06/2022
Puntos estimados: 10	Puntos reales: 10
Descripción: Como programador se requiere elaborar un manual de usuario para que el usuario pueda utilizar correctamente el servicio web.	
Observaciones: Se documenta la funcionalidad tanto con interfaz de usuario como con los endpoints del servicio web.	
Pruebas de aceptación Verificar que la documentación en el manual de usuario esté completa y consistente.	

Tarea de ingeniería	
Identificador: TI_HT7	Nombre de la historia: Realizar el manual de usuario del servicio web
Nombre de tarea: Documentación del manual de usuario	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 04/06/2022	Fecha de fin: 07/06/2022
Descripción: Se requiere documentar el manual de usuario para el uso correcto del servicio web.	
Pruebas de aceptación: Verificar que la documentación en el manual de usuario sea completa a las funciones del servicio web.	

Prueba de aceptación	
Identificador: PA1_HT7	Nombre de la historia: Realizar el manual de usuario del servicio web
Nombre la tarea: Documentación del manual de usuario	
Responsable: Jefferson Chariguamán	Fecha: 07/06/2022
Descripción: Desarrollar el manual de usuario en base a las funcionalidades del servicio web, especificando su funcionamiento de cada una de ellas.	
Condiciones de Ejecución: El sistema debe estar ya en funcionamiento e integrado con el modelo entrenado.	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir el documento correspondiente al manual de usuario • Describir cada una de las funcionalidades del servicio web 	
Resultado esperado: Documento del manual de usuario completado.	
Evaluación de la prueba: Exitosa.	

Historia de usuario

Código: HU2	Nombre: Gestionar el envío y respuesta del texto
Usuario: Consumidor del servicio web	Sprint Asignado: 3
Fecha de inicio: 12/06/2022	Fecha de fin: 14/06/2022
Puntos estimados: 10	Puntos reales: 10
Descripción: Como usuario del servicio web, requiero enviar el texto de la noticia al sistema, y de la misma manera recibir la respuesta de la clasificación.	
Observaciones	
Pruebas de aceptación	
<ul style="list-style-type: none"> • Verificar que el texto sea enviado al sistema. • Recibir el resultado de la clasificación. 	

Tarea de ingeniería	
Identificador: TI_HU2	Nombre de la historia: Gestionar el envío y respuesta del texto
Nombre de tarea: Gestionar las peticiones	
Tipo de tarea: Desarrollo	Programador responsable: Jefferson Chariguamán
Fecha de inicio: 12/06/2022	Fecha de fin: 14/06/2022
Descripción: Desarrollar las funciones que permitan la gestión recibir el texto y envío de respuestas.	
Pruebas de aceptación:	
<ul style="list-style-type: none"> • Verificar que el servidor reciba y responda a las peticiones del cliente con la información correcta. 	

Prueba de aceptación	
Identificador: PA_TI_HU2	Nombre de la historia: Gestionar el envío y respuesta del texto.
Nombre: Gestionar las peticiones	
Responsable: Usuario consumidor del servicio	Fecha: 14/06/2022
Descripción: Verificar que el sistema gestione los envíos y respuestas de las peticiones.	
Condiciones de Ejecución: Es necesario que el cliente realice una petición al servidor.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar al servicio web • Ingresar el texto de la noticia y presionar en el botón de “enviar noticia” • Verificar que el cliente responda 	
Resultado esperado: El servidor responde a las peticiones del usuario.	
Evaluación de la prueba: Exitosa.	

ANEXO D: Manual técnico

Manual técnico

DESARROLLO DE UN SERVICIO WEB PARA LA CLASIFICACIÓN DE NOTICIAS
FALSAS, APLICANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Jefferson José Chariguamán Cuji

1. Resumen

Este manual permite ayudar al personal técnico para poder hacer uso del servicio web, tomando en cuenta qué tipo de arquitectura, los datos de envío requeridos y las respuestas. De igual manera, para el modelo se especifica el tipo de algoritmo con el que se realiza la clasificación

2. Introducción

Para poder utilizar el servicio de clasificación para texto de noticias falsas, se requiere de conocer las rutas para realizar las peticiones, además del tipo de estructura que se debe enviar como respuesta para el correcto funcionamiento.

3. Objetivo

Describir el soporte técnico del funcionamiento, la estructura desarrollada tanto para el modelo y el servicio web, para después usar correctamente.

1. Requisitos del sistema para el cliente

1.1. Requisitos software

Navegador web para el acceso al servicio web mediante URL.

Conexión a internet

1.2. Requisitos hardware

Procesador: Cualquiera funcional capaz de realizar la tarea de peticiones

Memoria RAM: 500MB

2. Requisitos del sistema para el servidor

2.1. Requisitos software para el desarrollo

Lenguaje de programación Python

Librerías de machine Learning, Spacy, keras, tensorflow, flask, scikit-learn, matplotlib, numpy

IDE para realizar la programación.

2.2. Requisitos software para el despliegue

Servidor compatible con el lenguaje Python

Librerías instaladas como Spacy, keras, tensorflow, flask, scikit-learn, numpy, modelo pre-entrenado core_news_es_lg de Spacy.

2.3. Requisitos hardware para el desarrollo

Arquitectura de CPU x86_64; procesador Intel Core de segunda generación o posterior, o CPU AMD compatible con un hipervisor de Windows.

8 GB de RAM o más.

8 GB de espacio disponible en el disco como mínimo (IDE + SDK de Android + Android Emulator)

Resolución de pantalla mínima de 1280 × 800

2.4. Requisitos hardware para el despliegue

Azure ACU:200

RAM 3.5 GB

3. Herramientas utilizadas para el desarrollo

Para el desarrollo del servicio web se emplearon diferentes herramientas y plataformas, estas se describen a continuación:

3.1. Google Colab

Google Colab, también llamado Collaboratory, es un entorno basado en Jupyter notebook que lleva a cabo su ejecución en los servidores de Google, este entorno trae consigo la configuración necesaria para el uso de Python, los beneficios que provee esta plataforma son los recursos gratuitos como, GPU Nvidia K80, TPU y CPU, también permite la integración con Google Drive para el almacenamiento de proyectos o recursos. (G.L 2021)

3.2. Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft, multiplataforma que incluye Linux, Windows, macOS y Web, permite la edición y plugins para varios lenguajes, además de la integración con Git.

3.3. Postman

Es una plataforma API que permite construir y usar API's, esta simplifica realizar y comprender el ciclo de vida de una API, además permite la colaboración para la creación de mejores API de forma más rápida.

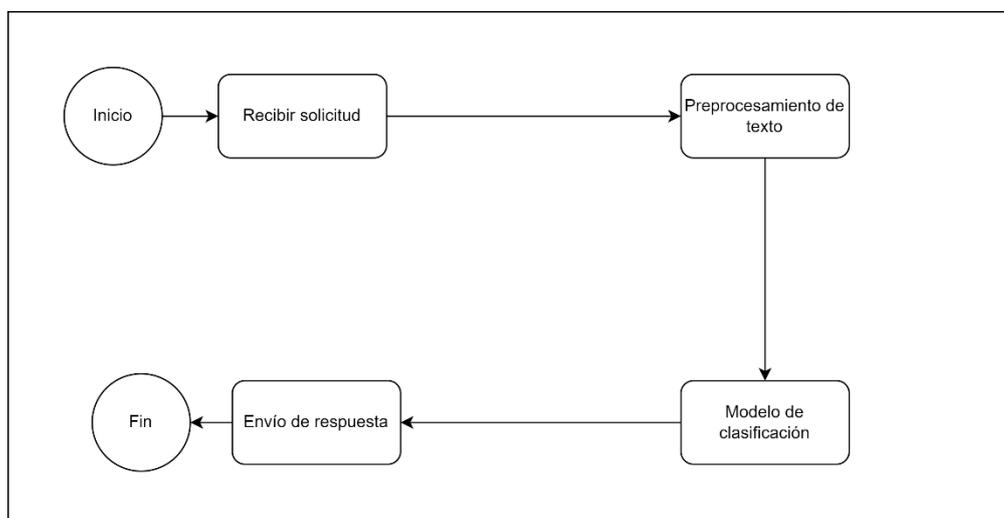
3.4. Python

El lenguaje Python, creado en la década de los noventa, es un lenguaje orientado a objetos que permite realizar desde aplicaciones de escritorio hasta servidores de red o desarrollo web.(Sarkar, Bali y Sharma 2018)

4. Requisitos del sistema

ID.	Requerimiento	Descripción
RF1	El sistema permitirá el preprocesamiento del texto de una noticia mediante el uso de librerías para el procesamiento del lenguaje natural.	El sistema recibirá el texto de las noticias mediante peticiones POST para su procesamiento.
RF2	El sistema permitirá la clasificación del texto de una noticia en falso o verdadero	El sistema procesará las peticiones GET para enviar al cliente la interfaz de usuario.
RF3	Se preprocesará el conjunto de datos para el entramiento del modelo	Se debe realizar el preprocesamiento del conjunto de datos para su uso posterior en el entrenamiento del modelo.
RF4	Se codificará el modelo para la clasificación de noticias	Se procederá con la codificación del modelo seleccionado con mejor rendimiento para su posterior implementación en el servicio web.
RF5	Se implementará el modelo entrenado en el servicio web	Se implementará el modelo desarrollado en el servicio web, para la clasificación de noticias.

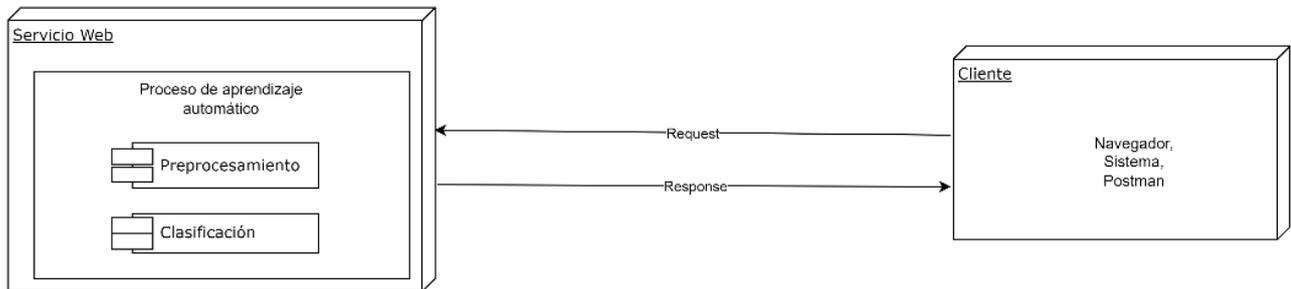
5. Diagrama de procesos



6. Arquitectura del sistema

El sistema web responde a peticiones POST donde recibe en formato json, el texto de la noticia con la clave “texto”, y la respuesta se envía en el mismo formato.

Las peticiones GET son respondidas con la interfaz de usuario, esta es utilizada para visualizar la interfaz en el navegador.



7. Modelo de aprendizaje automático

Para la implementación del modelo de aprendizaje automático para la clasificación de texto, estas 2 fases se describen a continuación.

7.1. Preprocesamiento de datos

Esta es una fase importante en el proyecto, dentro de estas se realizan las tareas de.

- Lemmatización
- Stemming
- Remover los stopwords
- Reemplazo de símbolos extraños

Las librerías utilizadas para el preprocesamiento de datos son:

```
import nltk as nl ## natural language toolkit nltk.org
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from nltk.tokenize import word_tokenize
import es_core_news_lg
esnlp=es_core_news_lg.load()
```

Como función para la limpieza de datos es la siguiente:

```
def limpieza_datos(docs, logging=False):
    texts = []
    counter = 1
    for doc in docs:
        doc = esnlp(doc, disable=['parser', 'ner'])
        tokens = [tok.lemma_.lower().strip() for tok in doc if tok.lemma_ != '-PRON-'
' and not tok.is_punct and not tok.is_stop]
        tokens = ' '.join(tokens)
        texts.append(tokens)
    return pd.Series(texts)
```

```
data_unified_cleaned = limpieza_datos(data_unified['texto'],logging= True)
```

7.2. Entrenamiento del modelo

Una vez realizado el preprocesamiento de los datos, se tiene el conjunto de datos limpios y unificado para utilizar en el entrenamiento del modelo. El siguiente código incluye la importación de librerías y la definición de la arquitectura.

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, GlobalMaxPool1D, Flatten, Concatenate, Max-
Pooling1D, LSTM, LeakyReLU
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
from keras.callbacks import ModelCheckpoint
from os import name
from keras.engine.input_layer import Input
from sklearn import metrics
from keras.backend import dropout
from keras.layers.core.spatial_dropout import SpatialDropout2D, SpatialDropout1D
model = Sequential()
model.add(LeakyReLU(0.1))
model.add(Embedding(100000,30,input_length=300,embeddings_initializer='GlorotNormal'))
model.add(Dropout(0.2))
model.add(LSTM(128, return_sequences=True,name= 'lstm_layer'))
model.add(GlobalMaxPool1D())
model.add(Dropout(0.5))
model.add(Dense(64,activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(32,activation="relu"))
model.add(Dropout(0.1))
model.add(Dense(1,activation='sigmoid'))
model.build(input_shape=(None,300))
model.compile(loss="binary_crossentropy",optimizer="Adam",metrics=["acc",f1_m,precision_m,re-
call_m])
model.summary()
```

Para guardar el modelo entrenado en checkpoints de acuerdo al val_loss se usa el siguiente código:

```
#Definicion del checkpoint (callbacks)
filepath = "/content/drive/MyDrive/FakesNews/LSTM_training.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_loss', ver-
bose=1, save_best_only=True, mode='min')
callbacks_list = [checkpoint]
```

Finalmente se utiliza el siguiente código para realizar el entrenamiento y guardar el historial.

```
history = model.fit(x_train, y_train, epochs=200, batch_size=32, validation_data=(x_test,y_test), verbose=1,callbacks=[callbacks_list,history_logger])
```



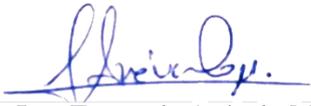
ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 28/02/2023

INFORMACIÓN DEL AUTOR	
Nombres – Apellidos: Jefferson José Chariguamán Cuji	
INFORMACIÓN INSTITUCIONAL	
Facultad: Informática y Electrónica	
Carrera: Software	
Título a optar: Ingeniero de Software	
f. Analista de Biblioteca responsable:	 Ing. Fernanda Arévalo M.



0238-DBRA-UPT-2023