UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACK OVERFLOW (ESPAÑOL E INGLÉS)

PROYECTO DE TITULACIÓN

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTORES:

CHICA MIRANDA KERLY MICHELL

MOREIRA PINCAY BRYAN ROLANDO

TUTOR(A):

ING. MIGUEL ÁNGEL BOTTO TOBAR, M. Sc.

GUAYAQUIL – ECUADOR

2020

REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍAS		
FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN		
TÍTULO: "DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACK OVERFLOW (ESPAÑOL E INGLÉS)"		
AUTOR(ES): Kerly Michell Chica Miranda Bryan Rolando Moreira Pincay	**REVISOR(A):** Ing. Diana Minda Gilces, M. Sc.	
INSTITUCIÓN: Universidad de Guayaquil	**FACULTAD:** Ciencias Matemáticas y Físicas	
CARRERA: Ingeniería en Sistemas Computacionales		
FECHA DE PUBLICACIÓN:	**N° DE PAGS:** 188	
AREA TEMÁTICA: Investigación		
PALABRAS CLAVES: Stack Overflow, preguntas duplicadas, aprendizaje automático, procesamiento de lenguaje natural, multilingüe.		
RESUMEN: Stack Overflow es una comunidad de preguntas y respuestas y son las preferidas de los programadores para resolver sus dudas. El sitio Stack Overflow en español se inició como alternativa al sitio en inglés pensado para ser utilizado por personas hispanohablantes. Sin embargo, muchas veces sus usuarios prefieren realizar sus preguntas también en el sitio inglés con el fin de obtener una respuesta de manera más rápida creando preguntas duplicadas en ambos sitios. La tarea de detectar estas preguntas duplicadas no se realiza ni siquiera por los moderadores de los sitios por lo que algunos investigadores han intentado abordar el problema utilizando técnicas de procesamiento qué lenguaje natural y aprendizaje automático. En este proyecto se realizará un análisis de la literatura mediante una revisión sistemática para determinar cuáles son las herramientas y técnicas más utilizadas por los investigadores. Luego se extraerán los datos de los sitios a fin de crear un *dataset* con pares de preguntas que serán utilizadas para los experimentos. Como experimentos se utilizarán las técnicas y herramientas analizadas en la revisión sistemática para desarrollar algoritmos cuyos resultados serán contrastados mediante un juicio externo para determinar si el rendimiento del mismo es suficiente para comprobar la hipótesis planteada, es decir, si la aplicación de técnicas de aprendizaje automático y procesamiento del lenguaje natural ayuda en la detección de preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español.		
N° DE REGISTRO:	**N° DE CLASIFICACIÓN:**	
DIRECCIÓN URL: (PROYECTO DE TITULACION EN LA WEB)		
ADJUNTO PDF	**SI** [x]	**NO** [ ]
CONTACTO CON AUTOR(ES): Kerly Michell Chica Miranda Bryan Rolando Moreira Pincay	**Teléfono:** 0939779917 0978897858	**Email:** kerly.chicam@outlook.com brmp96@gmail.com
CONTACTO DE LA INSTITUCIÓN Dirección: Víctor Manuel Rendón 429 y Baquerizo Moreno, Guayaquil.	**Nombre:** Ab. Juan Chávez Atocha	
	Teléfono: 2307729	
	Email: juan.chaveza@ug.edu.ec	

APROBACIÓN DEL TUTOR

En mi calidad de Tutor(a) del Trabajo de Titulación, “Detección de preguntas duplicadas en sitios Stack Overflow (español e inglés)” elaborado por los Sres. Chica Miranda Kerly Michell y Moreira Pincay Bryan Rolando, **estudiantes no titulados** de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero(a) en Sistemas Computacionales, me permito declarar que luego de haber orientado, estudiado y revisado, la **apruebo** en todas sus partes.

Atentamente,

Ing. Miguel Ángel Botto Tobar, M. Sc.

TUTOR

DEDICATORIA

El presente proyecto de titulación se lo dedico a mis abuelos, a mis padres y a mis hermanos por el amor, esfuerzo y apoyo brindado en todos estos años. En especial a mi padre Cristóbal por ser mi pilar fundamental en el desarrollo de mi carrera profesional.

Kerly Michell Chica Miranda

Dedico el presente trabajo de titulación a mis padres, hermanos y abuela que han depositado en mi la confianza necesaria para culminar mi carrera y enorgullecerlos con mi desarrollo académico.

Bryan Rolando Moreira Pincay

AGRADECIMIENTO

Agradezco la confianza, apoyo incondicional de mis padres en especial que con su esfuerzo y dedicación me ayudaron a culminar mi carrera universitaria.

Kerly Michell Chica Miranda

Agradezco principalmente a la familia de mi compañera Michell Chica ya que durante el desarrollo de este proyecto y de la mayor parte de mi carrera me han apoyado de manera desinteresada buscando mi desarrollo académico y profesional.

Bryan Rolando Moreira Pincay

TRIBUNAL PROYECTO DE TITULACIÓN

Ing. José González Ruiz, M. Sc.
DECANO DE LA FACULTAD
CIENCIAS MATEMÁTICAS Y FÍSICAS

Ing. Gary Reyes Zambrano, Mgs.
DIRECTOR DE LA CARRERA DE
INGENIERÍA EN SISTEMAS
COMPUTACIONALES

Ing. Miguel Ángel Botto Tobar, M. Sc.
PROFESOR TUTOR DEL PROYECTO
DE TITULACIÓN

Ing. Diana Minda Gilces, M. Sc.
PROFESOR(A) REVISOR(A) DEL
PROYECTO
DE TITULACIÓN

Ab. Juan Chávez Atocha, Esp.
SECRETARIO

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Proyecto de Titulación, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la UNIVERSIDAD DE GUAYAQUIL".

KERLY MICHELL CHICA MIRANDA

BRYAN ROLANDO MOREIRA PINCAY

CESIÓN DE DERECHOS DE AUTOR

Ingeniero

José González Ruiz, M. Sc.

DECANO DE LA FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

Presente.

A través de este medio indico a usted que procedo a realizar la entrega de la cesión de derechos de autor en forma libre y voluntaria del trabajo de titulación “Detección de preguntas duplicadas en sitios Stack Overflow (español e inglés)”, realizado como requisito previo para la obtención del Título de Ingeniero(a) en Sistemas Computacionales de la Universidad de Guayaquil.

Guayaquil, Marzo de 2021.

Kerly Michell Chica Miranda
C.I. N° 0931745434

Bryan Rolando Moreira Pincay
C.I. N° 0927230144

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACK OVERFLOW (ESPAÑOL E INGLÉS)

Proyecto de Titulación que se presenta como requisito para optar por el título de

INGENIERO(A) EN SISTEMAS COMPUTACIONALES

Autores: Kerly Michell Chica Miranda

C.I. N° 0931745434

Bryan Rolando Moreira Pincay

C.I. N° 0927230144

Tutor: Ing. Miguel Ángel Botto Tobar, M. Sc.

Guayaquil, Marzo de 2021

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Proyecto de Titulación, nombrado por el Consejo Directivo de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

CERTIFICO:

Que he analizado el Proyecto de Titulación presentado por los estudiantes **KERLY MICHELL CHICA MIRANDA, BRYAN ROLANDO MOREIRA PINCAY**, como requisito previo para optar por el Título de Ingeniero(a) en Sistemas Computacionales cuyo proyecto es:

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACK OVERFLOW (ESPAÑOL E INGLÉS)

Considero aprobado el trabajo en su totalidad.

Presentado por:

Chica Miranda Kerly Michell	C.I. N° 0931745434
Moreira Pincay Bryan Rolando	C.I. N° 0927230144

Tutor: Ing. Miguel Ángel Botto Tobar, M.Sc.

Guayaquil, Marzo de 2021

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

AUTORIZACIÓN PARA PUBLICACIÓN DE PROYECTO DE TITULACIÓN EN FORMATO DIGITAL

1. Identificación del Proyecto de Titulación

Nombre del Estudiante: Kerly Michell Chica Miranda	
Dirección: Coop. Sergio Toral 1era Etapa Mz.3479 SL. 7B	
Teléfono: 0939779917	**Email:** kerly.chicam@outlook.com

Nombre del Estudiante: Bryan Rolando Moreira Pincay	
Dirección: Urb. Cataluña Mz. 17 V. 26 Km. 12.5 Vía Samborondón	
Teléfono: 0978897858	**Email:** brmp96@gmail.com

Facultad: Ciencias Matemáticas y Físicas
Carrera: Ingeniería en Sistemas Computacionales
Proyecto de Titulación al que opta: Investigación
Profesor(a) Tutor(a): Ing. Miguel Ángel Botto Tobar, M. Sc.

Título del Proyecto de Titulación: Detección de Preguntas Duplicadas en Sitios Stack Overflow (Español e Inglés)

Palabras Claves: Stack Overflow, preguntas duplicadas, aprendizaje automático, procesamiento de lenguaje natural, multilingüe.

2. Autorización de Publicación de Versión Electrónica del Proyecto de Titulación
A través de este medio autorizo a la Biblioteca de la Universidad de Guayaquil y a la Facultad de Ciencias Matemáticas y Físicas a publicar la versión electrónica de este Proyecto de Titulación.

Publicación Electrónica:

Inmediata	**X**	Después de 1 año	

Firma Estudiante:

Chica Miranda Kerly Michell — C.I. N° 0931745434

Moreira Pincay Bryan Rolando — C.I. N° 0927230144

3. Forma de envío:

El texto del Proyecto de Titulación debe ser enviado en formato Word, como archivo .docx, .RTF o .Puf para PC. Las imágenes que la acompañen pueden ser: .gif, .jpg o .TIFF.

DVDROM [ ] CDROM [X]

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ABREVIATURAS

SO	Stack Overflow
SOES	Stack Overflow en español
Ing.	Ingeniero
M.Sc.	Máster
ML	*Machine Learning*
IA	Inteligencia Artificial
PLN	Procesamiento de Lenguaje Natural
WMD	*Word Mover Distance*
TAN	Traducción Automática Neuronal
TA	Traducción Automática
TAE	Traducción Automática Estadística
W2V	Word2Vec
RNA	Redes Neuronales Artificiales
CNN	*Convolutional Neural Networks*
RNN	*Recurrent Neural Networks*
LSTM	*Long Short- Term Memory*
API	*Application Programming Interface*
DL	*Deep Learning*

SIMBOLOGÍA

p	Probabilidad
X^2	Estadístico Chi-cuadrado
k	Número de resultados

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACK OVERFLOW

(ESPANOL E INGLÉS)

Autores: Kerly Michell Chica Miranda
C.I. N° 0931745434
Bryan Rolando Moreira Pincay
C.I. N° 0927230144

Tutor: Ing. Miguel Ángel Botto Tobar, M.Sc.

RESUMEN

Stack Overflow es una comunidad de preguntas y respuestas y son las preferidas de los programadores para resolver sus dudas. El sitio Stack Overflow en español se inició como alternativa al sitio en inglés pensado para ser utilizado por personas hispanohablantes. Sin embargo, muchas veces sus usuarios prefieren realizar sus preguntas también en el sitio inglés con el fin de obtener una respuesta de manera más rápida creando preguntas duplicadas en ambos sitios. La tarea de detectar estas preguntas duplicadas no se realiza ni siquiera por los moderadores de los sitios por lo que algunos investigadores han intentado abordar el problema utilizando técnicas de procesamiento qué lenguaje natural y aprendizaje automático. En este proyecto se realizará un análisis de la literatura mediante una revisión sistemática para determinar cuáles son las herramientas y técnicas más utilizadas por los investigadores. Luego se extraerán los datos de los sitios a fin de crear un *dataset* con pares de preguntas que serán utilizadas para los experimentos. Como experimentos se utilizarán las técnicas y herramientas analizadas en la revisión sistemática para desarrollar algoritmos cuyos resultados serán contrastados mediante un juicio externo para determinar si el rendimiento del mismo es suficiente para comprobar la hipótesis planteada, es decir, si la aplicación de técnicas de aprendizaje automático y procesamiento del lenguaje natural ayuda en la detección de preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español.

Palabras clave: Stack Overflow, preguntas duplicadas, aprendizaje automatico, procesamiento de lenguaje natural, multilingüe.

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DUPLICATE QUESTION DETECTION ON STACK OVERFLOW SITES

(SPANISH AND ENGLISH)

Autor(a)(es): Kerly Michell Chica Miranda
C.I. N° 0931745434
Bryan Rolando Moreira Pincay
C.I. N° 0927230144
Tutor(a): Ing. Miguel Ángel Botto Tobar, M.Sc.

ABSTRACT

Stack Overflow is a community of questions and answers and is a favorite among programmers to solve their doubts. The Stack Overflow in spanish site was started as an alternative to the english site intended to be used by spanish speakers. However, many times its users prefer to ask their questions also in the english site in order to get an answer in a faster way creating duplicate questions between both sites. The task of detecting these duplicate questions is not performed even by the moderators of the sites so some researchers have tried to address the problem using natural language processing and machine learning techniques. In this project, a literature analysis will be performed through a systematic review to determine which tools and techniques are most commonly used by researchers. Then, data will be extracted from the sites in order to create a dataset with pairs of questions that will be used for the experiments. As experiments, the techniques and tools analyzed in the systematic review will be used to develop algorithms whose results will be contrasted through an external trial to determine whether the performance of the algorithm is enough to test the hypothesis, that is, whether the application of machine learning, and natural language processing techniques helps in the detection of duplicate questions between the Stack Overflow and Stack Overflow in spanish sites.

Key words: Stack Overflow, duplicate questions, machine learning, natural language processing, multilingual.

INTRODUCCIÓN

El sitio Stack Overflow ha sido un éxito total entre los programadores de habla inglesa desde su anuncio en el año 2008. Sin embargo, los sitios alternativos no han logrado alcanzar la popularidad del sitio original debido a que muchos usuarios se plantean si es realmente necesario dividir la comunidad en varios sitios. Lo cierto es que muchos de los usuarios de los sitios alternativos al final terminan buscando respuestas en el sitio inglés de modo que una misma pregunta realizada por un usuario se puede encontrar en dos sitios en distintos idiomas con el objetivo de obtener una respuesta más rápidamente.

Investigadores de diferentes partes del mundo han abordado el problema de detectar la duplicidad de preguntas obteniendo resultados bastante prometedores. Sin embargo, en general no abordan el problema de la duplicidad entre sitios de distintos idiomas. Pocos son los investigadores que incursionado en este campo debido a la complejidad que representa realizar un estudio de este tipo. Aun así, los resultados que han obtenido no distan de aquellos obtenidos por los investigadores enfocados en el sitio en inglés. Esto que resulta alentador ya que son incluso menos los estudios que tratan la duplicidad entre preguntas formuladas en español e inglés

En este proyecto se buscará abordar este problema realizando una investigación para determinar si las herramientas de aprendizaje automático resultan útiles al momento de detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español.

El primer capítulo consistirá en el planteamiento y descripción del problema además de la definición de los objetivos que guiarán el desarrollo del proyecto. También se definirá el respectivo alcance del proyecto tomando en cuenta las limitaciones encontradas en el mismo.

El segundo capítulo presenta el marco teórico de esta investigación en el cual se definirá la fundamentación teórica y los antecedentes del problema planteado. También se incluirá un

análisis de la literatura realizado a través de una revisión sistemática cuyo proceso también será descrito en este capítulo como cumplimiento al primer objetivo. Al final de este capítulo se definirá la hipótesis a comprobar además de las variables de investigación que se van a utilizar en la metodología. Estas son la variable dependiente y la variable independiente.

En el tercer capítulo se detallará el proceso realizado para llevar a cabo la investigación empezando por la definición de la modalidad y el tipo de investigación elegida para este estudio. Luego serán descritos los pasos llevados a cabo para cumplir con el resto de los objetivos empezando con la extracción de los datos que serán utilizados para realizar los experimentos y contrastar los resultados de los mismos. Una vez extraídos los datos se detallará de qué manera fueron utilizadas las herramientas y técnicas que se obtuvieron como resultado de la revisión sistemática conducida en el capítulo anterior. Como resultado se obtendrán una serie de experimentos realizados con los algoritmos desarrollados a partir de estas técnicas y herramientas. Por último, se realizará un contraste de los resultados obtenidos con los algoritmos y los resultados del juicio externo obtenidos mediante la técnica de recolección de datos elegida para este estudio a fin de comprobar la validez de la hipótesis planteada anteriormente.

En el cuarto y último capítulo se redactarán las conclusiones a las que se llegó con el desarrollo de este proyecto y además se darán recomendaciones que serán de ayuda para quienes deseen realizar una investigación similar a esta. Al final del capítulo también se incluirán trabajos futuros que podrán ser realizados por los interesados a fin de complementar el trabajo presentado en este estudio llegando así a ampliar el alcance que se definió inicialmente.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Descripción de la situación problemática

Stack Overflow es uno de los sitios más conocidos de preguntas y respuestas que son beneficiosos al momento de intercambiar conocimientos tanto entre programadores aficionados como profesionales. En la actualidad hay un aproximado de 21 millones de preguntas en Stack Overflow y alrededor de 142 mil preguntas en Stack Overflow en español.

Como se puede apreciar la cantidad de preguntas en español representa menos del 1% de la cantidad de preguntas en inglés por lo que se hace notoria la escasez de información que existe en el habla hispana. Según el Índice de Proficiencia en inglés de la compañía Education First (EF EPI) para el año 2020, Ecuador se encuentra en el puesto 93 de 100 países siendo este el país latinoamericano con el nivel más bajo de conocimiento en el idioma inglés (Education First, 2020).

Esto es importante en el contexto ecuatoriano ya que está directamente relacionado con la velocidad con que las personas dedicadas a la programación logran solucionar un problema o implementar alguna funcionalidad en sus productos. El tener dominio sobre el idioma inglés puede hacer que una tarea que unos logran resolver en 3 semanas pueda ser realizada en 3 horas gracias al aporte de los usuarios de comunidades como Stack Overflow. Es cierto que en la actualidad existen varios sitios y servicios de traducción automática que funcionan de manera decente para propósitos generales pero que al ser utilizados para traducir texto en lenguaje más técnico suelen perder eficacia. Es por todo esto que resulta complicado para los estudiantes y profesionales ecuatorianos, y de habla hispana en general, formular por sí mismos preguntas técnicas en inglés de modo que puedan encontrar una respuesta satisfactoria en Stack Overflow.

Sin embargo, por el gran volumen de preguntas que existen entre los sitios Stack Overflow, se genera una gran cantidad de preguntas duplicadas o similares que son publicadas por los usuarios, por lo que resulta conveniente tener algún mecanismo que permita detectar la duplicidad entre las preguntas en inglés y español y así encontrar la respuesta deseada de manera rápida incluso si la pregunta es realizada en el sitio en español.

Ubicación del problema en un contexto

Tanto los programadores aficionados como los profesionales de la informática utilizan comúnmente la comunidad Stack Overflow para realizar preguntas y obtener respuestas relacionadas con programación, sin embargo, la eficacia de este sitio depende del nivel de experiencia del usuario y de la información utilizable que encuentren en el idioma utilizado.

Debido a esto varios usuarios de habla hispana se inclinan en buscar en sitios de habla inglesa ya que no encuentran mucha información en español, pero para algunos usuarios esto resulta una dificultad debido a que el dominio que tienen inglés no les permite formular preguntas complejas de modo que obtengan las respuestas adecuadas. Resulta especialmente complicado para usuarios ecuatorianos cuyo nivel de inglés está catalogado como el más bajo en Latinoamérica (Education First, 2020).

Por lo tanto, desarrollar un algoritmo que permita detectar las preguntas duplicadas entre Stack Overflow y Stack Overflow en español, permitirá a los desarrolladores revisar la información de las publicaciones en español y también la de similares en ingles facilitando la obtención de respuestas sin que su pregunta tenga que ser publicada en ambos sitios y por lo tanto convertirse en duplicadas.

Situación conflicto nudos críticos

En Stack Overflow las preguntas duplicadas son marcadas como similares por los usuarios que tienen una gran reputación en el sitio o por sus moderadores, es decir que este se realiza manualmente. Aun así, esto no retorna los resultados esperados debido a que es complicado evitar que los usuarios realicen preguntas duplicadas en sitios como Stack Overflow donde el volumen de información es tan elevado, llegando a tener un aproximado de 21 millones de preguntas actualmente.

Siendo lo anteriormente mencionado un problema, cabe mencionar que esta comunidad no realiza una detección de preguntas similares entre los sitios Stack Overflow y Stack Overflow en español, lo cual es una desventaja ya que hay usuarios que no cuentan con el nivel adecuado de inglés y por lo cual no pueden encontrar la información deseada ya que se encuentra en ese idioma como a su vez usuarios de habla hispana que buscan información en habla inglesa ya que comúnmente hay más información.

Este trabajo de investigación tiene como objetivo el desarrollo un algoritmo para detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español para así obtener una respuesta satisfactoria de manera ágil mediante el uso de técnicas de PLN y aprendizaje automático.

Delimitación del problema

Stack Overflow es considerado uno de los sitios más populares de preguntas y respuestas en línea en donde los desarrolladores de software intercambian sus conocimientos relacionadas con programación, sin embargo, como indica Botto-Tobar et al. “por las numerosas preguntas publicadas en el sitio da como resultado que estas preguntas expresen el mismo punto es decir estén duplicadas, lo que es un problema ya que se desperdician fuentes que pueden usarse para

responder otras preguntas" (Botto-Tobar, et al., 2018), esto causa también que los desarrolladores tengan que esperar por una respuesta cuando está ya disponible, incluso en otro idioma.

Stack Overflow permite que estas preguntas sean señaladas manualmente como duplicadas de otras preguntas, esto es realizado por sus moderadores, pero por la gran cantidad de preguntas, el identificar manualmente las preguntas duplicadas o similares resulta un trabajo complejo. El presente estudio se limitará a la investigación tecnológica ya que no existe un mecanismo para detectar las preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español con técnicas de PLN y aprendizaje autónomo.

Tabla 1

Delimitación del problema

Delimitador	Descripción
Campo	Ciencias básicas, Bioconocimiento y Desarrollo industrial
Área	Tecnologías de la información y Telecomunicaciones
Aspecto	Informático: Aplicación de técnicas de PLN y Aprendizaje Autónomo
Tema	Detección de Preguntas Duplicadas en sitios Stack Overflow (español e inglés)

Nota: En esta tabla se plantean los términos de análisis aplicados para la delimitación del problema conforme al contexto en donde se desarrolla la problemática. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Evaluación del Problema

Los aspectos generales para considerar la evaluación del problema son:

- **Delimitado:** Las preguntas duplicadas son un tema que se dificulta a pesar del mantenimiento realizado por los moderadores de la comunidad de Stack Overflow, debido a las numerosas preguntas que son publicadas resulta un trabajo difícil, por lo que el desarrollo de un algoritmo que detecte las preguntas duplicadas entre Stack Overflow y Stack Overflow en español mediante el uso de técnicas de aprendizaje autónomo lo que facilitaría la búsqueda de preguntas similares a los desarrolladores.

- **Claro:** En Stack Overflow a pesar de que los usuarios con gran reputación y los moderadores marcan manualmente las preguntas duplicadas esto no evita que se realice un esfuerzo adicional sino también respuestas retrasadas, incluso no existe un mecanismo que detecte las preguntas similares entre Stack Overflow y Stack Overflow en español lo que resultaría ágil al momento de obtener información sobre temas de programación.
- **Relevante:** Detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español es beneficioso ya que facilitará la búsqueda y reduciría el tiempo para encontrar tanto las preguntas similares como en obtener la información deseada.
- **Original:** Mediante la investigación realizada se llegará a la conclusión de que existe poca o nada de información disponible acerca de estudios para detectar preguntas duplicadas entre un sitio en línea de preguntas y respuestas como Stack Overflow en dos idiomas, en este caso Stack Overflow (español e inglés).
- **Factible:** Al realizar la investigación de diversos estudios se puede destacar varias técnicas para detectar preguntas duplicadas como varias técnicas de PLN y aprendizaje autónomo, que son una base para dar solución a temas relacionados con preguntas duplicadas. En conclusión, existen varias de fuentes de donde se puede obtener información que son seguras con datos reales.
- **Identifica los productos esperados:** Se desarrollará un algoritmo utilizando técnicas de aprendizaje automático para detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español, solucionando los problemas de los desarrolladores al momento de obtener la información esperada de forma rápida sin que los moderadores tengan que etiquetar manualmente las preguntas como duplicadas.

Causas y consecuencias del problema

Tabla 2

Matriz de causas y consecuencias del problema

Causas	Consecuencias
C1. Gran volumen de información.	E1. Demasiada información dificulta encontrar las preguntas similares con agilidad.
C2. Gran cantidad de preguntas duplicadas.	E2. Tiempo de espera innecesario para obtener la respuesta cuando está ya se encuentra disponible.
C3. Incorrecta redacción al formular una pregunta en Stack Overflow.	E3. Se obtiene como resultado que las preguntas sean similares y se marquen como duplicadas.
C4. Marcación manual por los moderadores de las preguntas duplicadas en Stack Overflow.	E4. Se desperdician los recursos que pueden utilizarse para responder otras preguntas.
C5. Falta de búsqueda de preguntas duplicadas entre Stack Overflow (español e inglés).	E5. Exclusión a la posibilidad de encontrar la respuesta a la pregunta realizada que está en otro idioma.

Nota: Esta tabla se muestran las causas y consecuencias del problema, análisis que se realizó en base a la recopilación inicial de información de la situación problemática que genera el proyecto. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Formulación del problema

¿El desarrollo de un algoritmo utilizando técnicas de aprendizaje automático permite la detección de preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español?

Objetivos del proyecto

Objetivo general

Desarrollar un algoritmo utilizando técnicas de aprendizaje automático para la detección de preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español facilitando la búsqueda de respuestas a usuarios de los sitios.

Objetivos específicos

1. **Analizar** la bibliografía reciente para la identificación de técnicas de procesamiento de lenguaje natural y aprendizaje automático aplicables al proyecto.
2. **Construir** un conjunto de datos con las preguntas extraídas de los sitios Stack Overflow y Stack Overflow en español para la detección de preguntas duplicadas.
3. **Desarrollar** un algoritmo que detecte la duplicidad de preguntas entre los sitios Stack Overflow y Stack Overflow en español.
4. **Evaluar** los resultados obtenidos que determinen el desempeño del algoritmo.

Alcance del proyecto

Para asegurar el cumplimiento del objetivo del proyecto se ha descompuesto el mismo en cuatro objetivos específicos a cumplir de manera secuencial. Primero se analizará la bibliografía más reciente mediante un mapeo sistemático de la literatura disponible en Google Académico (Google Scholar) de modo que se pueda determinar el estado del arte del tema y así definir los

recursos a utilizar en el desarrollo del algoritmo. Después de haber determinado las técnicas y herramientas se elaborará un conjunto de datos mediante la extracción y procesamiento de las publicaciones existentes, filtrando las preguntas por una etiqueta en común definida en ambos sitios.

Posteriormente, se desarrollará un algoritmo para la detección de preguntas duplicadas mediante la aplicación de técnicas de PLN y aprendizaje autónomo obtenidas de fuentes bibliográficas. Por último, se evaluarán los resultados obtenidos mediante el cálculo de métricas de rendimiento para comprobar que el algoritmo cumple con los requerido mediante el análisis de los valores de dichas métricas.

Justificación e importancia

Stack Overflow es un sitio del tipo Q&A (Preguntas y Respuestas) en el que los programadores pueden ayudarse unos a otros respondiendo las preguntas y solucionando problemas planteados por ellos mismos mediante publicaciones. Sin embargo, este sitio cuenta con una política de “solo inglés” que limita el idioma que se puede utilizar para interactuar con el sitio. En un estudio realizado por Wang Y., se menciona que dentro de estas comunidades existen dos grupos. Están quienes dominan el idioma inglés y pueden aprovechar todos los beneficios y la información del sitio. Y por otro lado están quienes tienen problemas utilizando el idioma inglés y no pueden utilizar el sitio con libertad. El segundo grupo, por lo tanto, se ve obligado a acudir a variantes del sitio en su respectivo idioma, pero se encuentran con que la cantidad y calidad de información es mucho menor con respecto al sitio principal (Wang Y. , 2019).

Aunque la solución aparentemente más fácil es remover la política de “solo inglés” y unificar los sitios, puede resultar complicado debido a lo costoso que sería desarrollar y mantener

un sitio multilingüe (Wang Y. , 2019). Debido a esta separación de las variantes del sitio resulta mucho más complicado descubrir y vincular preguntas duplicadas ya que los sitios de Stack Overflow se encargan solamente de las preguntas duplicadas en el idioma definido del sitio.

Tabla 3

Preguntas en comunidades Stack Overflow

Sitio	Usuarios	Intersección usuarios con SOEN	Preguntas
Stack Overflow en inglés	9,321,924	/	9,293,483
Stack Overflow en ruso	128,016	10,706	256,042
Stack Overflow en portugués	111,102	8,506	121,460
Stack Overflow en japonés	54,492	1,293	49,203

Nota: Esta tabla refleja la cantidad de preguntas duplicadas en las variantes del sitio que no se compara a la cantidad del sitio principal. La elaboración es propia y la fuente corresponde a datos tomados de (Wang Y. , 2019).

Como se puede observar en la Tabla 3 la cantidad de preguntas publicadas en las variantes del sitio no se compara a la cantidad del sitio principal. Incluso Stack Overflow en portugués, que es la variante más antigua, no llega a tener ni el 2% de la cantidad de preguntas que tiene el sitio principal.

Por lo expuesto anteriormente, el presente proyecto tiene como finalidad desarrollar un algoritmo utilizando técnicas de aprendizaje autónomo para detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español facilitando la búsqueda de respuestas a los usuarios de los sitios. Este trabajo investigativo será de beneficio tanto para la comunidad de Stack Overflow en español como para la comunidad de desarrolladores hispano-hablantes en general ya que permitirá que estos obtengan respuestas con mayor facilidad puesto que se ampliará el rango de búsqueda a las publicaciones en el idioma inglés el cuál es uno de los idiomas con más información disponible.

Limitaciones del estudio

El presente trabajo de titulación puede limitarse en determinados apartados debido a las dificultades que se presentarían las cuales se detallan a continuación:

D1: Para la extracción de datos se consideran solo las preguntas con una etiqueta definida, en este caso “Kivy”. Esto debido a que trabajar y procesar la cantidad total de preguntas en los ambos sitios requiere un procesamiento y tiempo mucho mayor que con el que se cuenta para el desarrollo del proyecto.

D2: Para la construcción del *dataset* se omitirán los ciertos metadatos correspondientes a las preguntas que no resultan útiles para el desarrollo y análisis del algoritmo con el fin de no consumir recursos computacionales innecesariamente y aprovecharlos para tareas más prioritarias.

D3: El etiquetado que se realizará para el entrenamiento de los modelos en el algoritmo se limitará a 5 coincidencias entre las aproximadamente ochenta preguntas en español y las aproximadamente diez mil preguntas en inglés ya que realizar una comparación manual de aproximadamente ochocientos mil pares de preguntas no resulta viable considerando el tiempo otorgado por la institución para el llevar a cabo el proyecto.

D4: Se utilizarán modelos de aprendizaje automático previamente entrenados dentro del algoritmo debido a la dificultad que representa recolectar cantidades masivas de datos y entrenar un modelo con ellas. Dependiendo del rendimiento que presenten en el desarrollo serán incluidos en la versión final del algoritmo.

D5: Dentro del algoritmo se implementarán servicios de traducción de máquinas neuronales como el provisto por Google y su servicio Translate ya que el desarrollo de un algoritmo de traducción no forma parte del alcance de este proyecto.

CAPÍTULO II

MARCO TEÓRICO

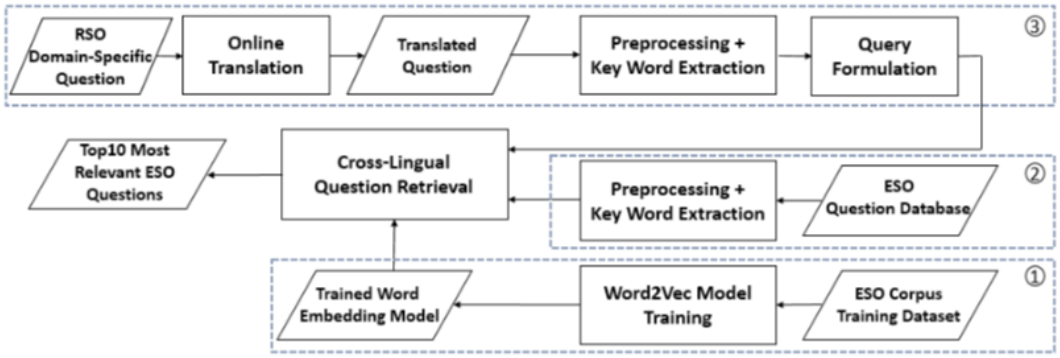
Antecedentes del estudio

El procesamiento del lenguaje natural es una rama de la IA que ha ganado relevancia con el desarrollo de otros campos como la Big Data. Han sido varias las aplicaciones que se han encontrado para el procesamiento del lenguaje natural. Entre ellas se encuentra la búsqueda de similitud semántica entre textos la cual puede ser utilizada para determinar la duplicidad entre textos con una sintaxis distinta pero una semántica parecida. Esta aplicación se hace presente en varios sitios de preguntas y respuestas (Q&A) entre los cuales se encuentra Stack Overflow, el cual tiene la comunidad de desarrolladores más activa en la actualidad contando con un aproximado de 20 millones 600 mil preguntas a la fecha (Stack Overflow en español, 2020).

Varias han sido las estrategias que se han desarrollado con el fin de detectar la duplicidad entre preguntas publicadas en sitios de Q&A. Sin embargo, son muy pocos los estudios que han intentado abordar el problema de la duplicidad entre preguntas de sitios con diferentes idiomas. Entre estos pocos estudios se encuentra el realizado por Qiu, en el que propone una herramienta para extraer preguntas duplicadas o relacionadas tomando como base las preguntas en los sitios de Stack Overflow y Stack Overflow en ruso (Qiu, 2018). En la Figura 1 se puede observar las distintas fases del proceso realizado para obtener preguntas duplicadas entre los sitios. El autor plantea utilizar la traducción automática para introducir el resultado en una función de preprocesamiento y extracción de palabras clave con el fin de formular una consulta que a través de incorporaciones de palabras será comparada con una base de preguntas en inglés de modo que se pueda determinar qué preguntas se relacionan con la de la consulta.

Figura 1

Proceso para obtener preguntas duplicadas



Nota: Aquí se muestran los pasos para obtener preguntas duplicadas entre los sitios. La elaboración y la fuente corresponde a datos tomados de (Qiu, 2018).

Otro estudio relacionado con la recuperación de preguntas relacionadas entre sitios Stack Overflow de distintos idiomas es el realizado por (Xu, Xing, Xia, Lo, & Le, 2017). Aquí los autores proponen una herramienta llamada XSearch para realizar la búsqueda relacionada de preguntas entre los sitios en idioma chino e inglés. Si bien no pretenden identificar explícitamente preguntas duplicadas, proponen estrategias interesantes para establecer la relevancia de una pregunta con respecto a otra sorteando de cierto modo la barrera del idioma. De manera parecida al estudio anterior utilizan la traducción automática, pero desde un enfoque distinto ya que antes de utilizarla los autores realizan una extracción de palabras de dominio especifico de la pregunta en chino que obtiene palabras tanto en chino y en inglés. Luego realizan la traducción de las palabras o términos de dominio especifico en chino tomando como base un vocabulario de palabras y términos de dominio especifico generado a partir de preguntas de Stack Overflow en inglés. Una vez obtenida la traducción formulan una consulta al algoritmo de recuperación de preguntas que utiliza las incorporaciones de palabras para encontrar la similitud semántica de las palabras y obtener las 10 preguntas más relevantes.

A pesar de que el tratamiento de preguntas duplicadas en distintos idiomas es un tema que se ha abordado con poca frecuencia existen estudios que buscan detectar preguntas duplicadas de un mismo idioma y cuyas estrategias pueden ser de mucha utilidad al ser adaptadas un enfoque multilingüe. Uno de los primeros y más reconocidos es el estudio realizado por Zhang, Lo, Xia, & Sun, en el que propone una herramienta para la detección de preguntas duplicadas a la que llamó DupPredictor (Zhang, Lo, Xia, & Sun, 2015). En esta herramienta se extraen partes esenciales de una pregunta como su título, descripción y etiquetas. Además, obtiene el tópico de la pregunta a través de un modelo de tópicos para así utilizarlo como una característica adicional en la comparación. Una vez se obtienen estas características de cada pregunta, se comparan con las características de la pregunta a comparar de modo que se obtiene su similitud para cada característica. Por último, calcula un valor de similitud general utilizando las similitudes de las características multiplicadas por pesos que son encontrados a través de un algoritmo voraz.

Por parte de (Wang, Zhang, & Jing, 2020) se realizaron pruebas utilizando tres estrategias de aprendizaje profundo (*Deep Learning*) basadas en incorporaciones de palabras (*Word Embeddings*), específicamente representaciones obtenidas con el método de Word2Vec. Una vez obtenido el vector se utilizan 3 tipos de redes neuronales para detectar la duplicidad entre preguntas. Los resultados de este estudio demuestran que el uso de las redes LTSM y CNN en conjunto con Word2Vec presentan métricas significativamente mejores a las que muestran técnicas de aprendizaje automático tradicionales como las máquinas de soporte de vectores o el algoritmo de aumento extremo de gradiente (XGBoost). Esto demuestra que para una comparación monolingüistica los algoritmos de *Deep Learning* y las redes neuronales presentan resultados bastante favorables siendo necesario que se incluyan capas extras para tratar con comparaciones multilingües.

Fundamentación teórica

Stack Exchange

Es una red que consiste en 176 comunidades de preguntas y respuestas, entre esas comunidades incluye Stack Overflow la cual es la comunidad en línea más grande en la que los desarrolladores aprenden y comparten sus conocimientos. Fue lanzada en 2010, desde entonces se ha convertido en uno de los 50 destinos en línea los cuales prestan servicios a más de 100 millones de desarrolladores mensualmente (Stack Exchange, 2021).

Stack Exchange fue fundada por Jeff Atwood y Joel Spolsky en el año 2008, fomenta el concepto de reputación entre los usuarios promoviendo el compromiso con la comunidad, esta es premiada, cada usuario puede perder como ganar puntos en el sitio. Respecto a los votos se maneja una estricta reglamentación, lo que resulta provechoso a la hora de extraer datos ya que tienen cierto nivel de filtro que no permite que afecte la calidad de una pregunta o una respuesta (Berón Abou-Nigm & Jardim Godoy, 2017).

Stack Overflow

Es la comunidad en línea principal del sitio Stack Exchange que permite a los desarrolladores de software realizar preguntas y obtener respuestas, su inicio se dio en el año 2008 y se ha convertido en un sitio muy popular para hacer preguntas y obtener respuestas relacionadas con problemas de programación. Los usuarios pueden ganar reputación e insignias según su contribución, el sitio recomienda a los usuarios buscar anteriores publicaciones antes de realizar una nueva pregunta para evitar preguntas duplicadas o similares (Ahasanuzzaman, Asaduzzaman, Roy, & Schneider, 2016).

Stack Overflow en español

Pertenece a la red de comunidades de Stack Exchange al igual que Stack Overflow utilizada por los desarrolladores y profesionales de la informática para desarrollar preguntas y respuestas mediante *posts*, su única diferencia es que esta es una comunidad para programadores y desarrolladores de software de habla hispana (Stack Overflow en español, 2020).

Stack Exchange API

Es un contenedor de la documentación de la API de Stack Exchange con compatibilidad de autenticación y escritura. Su uso está sujeto a una serie de aceleradores que en general las aplicaciones deben esforzarse por hacer solicitudes en un menor número para satisfacer su función (Stack Exchange, Inc., 2021).

Aprendizaje Automático

Aprendizaje automático, aprendizaje autónomo o Machine Learning es el procedimiento de métodos matemáticos de datos con la finalidad de enseñarle a un grupo sin necesidad de recibir un aprendizaje directo ya que utiliza algoritmos para reconocer patrones de datos que se utilizan en la creación del modelo de datos para efectuar predicciones. Estos serán más precisos entre más experiencia y datos se obtengan como resultado, es decir que al igual que los humanos estos prosperan con la experiencia que obtienen con la práctica (Microsoft Azure, 2021).

El aprendizaje automático para Mohri, Rostamizadeh, & Talwalkar, se puede puntualizar como los procedimientos computacionales que manejan la experiencia para realizar predicciones exactas o aumentar la rentabilidad de las mismas. Con experiencia nos referimos al conocimiento o información pasada que se pueden utilizar para su análisis (Mohri, Rostamizadeh, & Talwalkar, 2018).

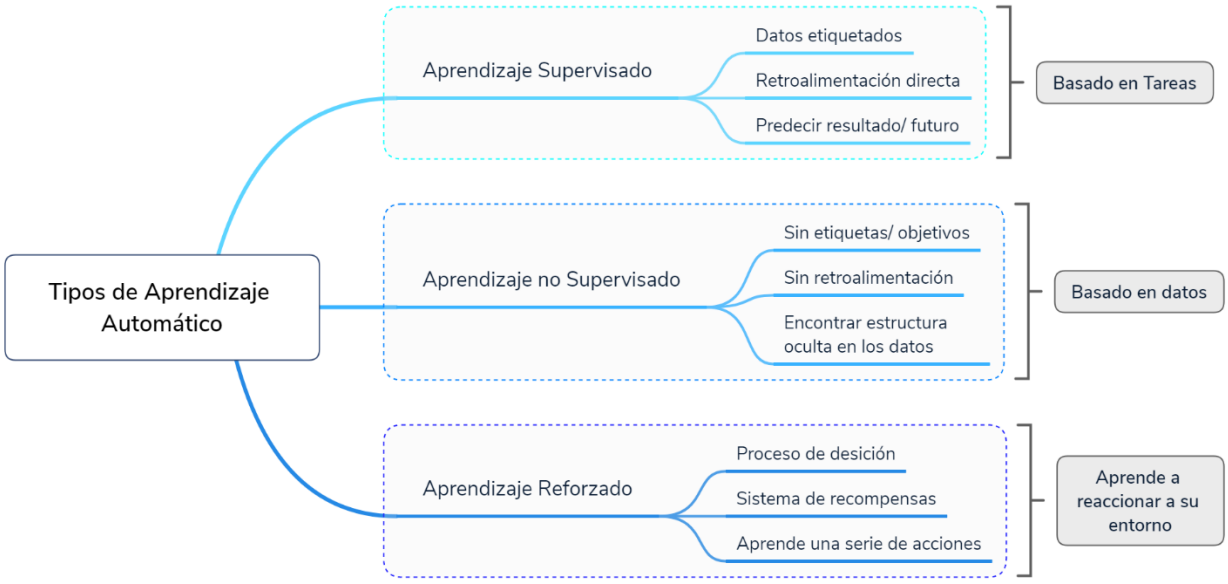
El aprendizaje automático también se puede definir como el área del conocimiento computacional el cual se concentra en el estudio y la apreciación de estructuras de datos y de patrones que hacen factible la enseñanza, el razonamiento y la toma de decisiones sin necesidad de la interacción humana (NetApp, 2020).

Técnicas de aprendizaje automático

Sus técnicas se clasifican en tres tipos de algoritmos como se puede visualizar en la Figura 2 a continuación, se detallará cada tipo de algoritmos.

Figura 2

Tipos de Técnicas de Aprendizaje Automático



Nota: La figura muestra los tres tipos de algoritmos de aprendizaje automático. Los cuales a continuacion, se detallará con un breve concepto de cada tipo siguiendo el orden la figura 2. La elaboración es propia y la fuente corresponde a datos tomados de Raschka, S. & Mirjalili, V. (2017).

Aprendizaje Supervisado

Para Iberdrola, el algoritmo de aprendizaje supervisado: “Cuenta con un aprendizaje ya previo el cual está fundamentado en un método de etiquetas asociadas a unos datos que les proporciona la toma de decisiones o hacer predicciones” (Iberdrola, 2020).

El aprendizaje supervisado es la técnica que aborda los grupos o equipos de datos con etiquetas o estructura que se ocupa como un maestro y prepara al grupo, lo que incrementa su capacidad para ejecutar una predicción o tomar una decisión” (Microsoft Azure, 2021).

Aprendizaje no Supervisado

El aprendizaje supervisado es un algoritmo que basa su desarrollo de entrenamiento en un equipo de datos sin etiqueta o clases ya anticipadamente definidas, es decir que el valor objetivo o de clase no se conoce, este valor puede ser numérico o categórico. Este aprendizaje este empleado a las tareas de agrupamiento o llamadas clustering el cual tiene como objetivo hallar grupos semejantes en el conjunto de datos (Vallalta Rueda, 2019).

Aprendizaje Reforzado

Se utiliza comúnmente para desarrollar un agente o sistema el cual mejore el desempeño de las interacciones con el entorno, el estado actual del entorno incluye normalmente una llamada en señal de recompensa, mediante su interacción con el entorno el agente puede utilizar el aprendizaje reforzado para aprender una serie de acciones que maximizan la recompensa a través de un proceso de prueba y error (Raschka & Mirjalili, 2017).

Inteligencia Artificial

La IA es la capacidad que presentan los ordenadores para realizar funciones que habitualmente requieren la inteligencia humana, es la habilidad que tiene la máquina para manejar algoritmos, instruirse de los datos y usar lo aprendido para tomar decisiones como lo haría un

humano. La diferencia es que no necesitan descansar los dispositivos basados en IA y pueden comparar a la vez considerables cantidades de información por lo que el tamaño de errores es relativamente menor (Rouhiainen, 2018).

Procesamiento de Lenguaje Natural

El PLN (del inglés, Natural Language Processing) pertenece al campo de conocimiento de la Inteligencia Artificial, es un área de investigación y aplicación que analiza como las computadoras se pueden usar para comprender y manipular el texto o el habla en lenguaje natural. El objetivo es recopilar los conocimientos sobre como los humanos usan y comprenden el lenguaje para desarrollar técnicas y herramientas para realizar sistemas informáticos (Chowdhury, 2020).

Para poder definir el PLN se debe entender que es el lenguaje natural. Para Ramos & Velez hace alusión al lenguaje oral y escrito por el cual se intenta comunicar, se necesita de procesos corporales de captación y millones de conexiones neuronales para comprender la estructura de menasajes en lenguaje humano. Una característica es que entre las personas se manifiesta de forma espontánea mientras que los ordenadores se comunican por un riguroso protocolo matemático (Ramos & Velez, 2016).

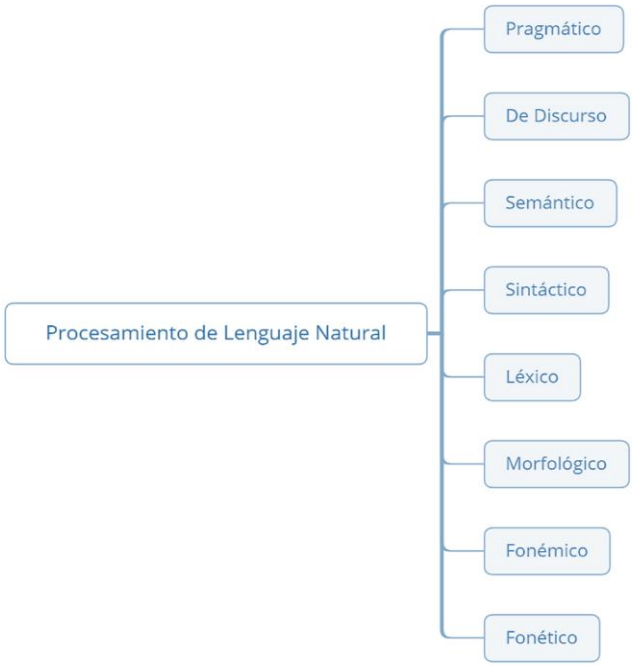
Tal y como expresa Ramos & Velez el PLN se utiliza en una gran cantidad de tareas y aplicaciones tales como la minería de datos, traducción de texto, generación de resúmenes, métodos de búsqueda de respuestas y muchas más (Ramos & Velez, 2016).

Niveles de Procesamiento de Lenguaje Natural

Cada nivel de Procesamiento de Lenguaje Natural interpreta un tipo de estudio que se debe realizar al texto de entrada para extraer información concreta. Los niveles del PLN se muestran en la Figura 3 y se describirán desde el más alto al más bajo.

Figura 3

Niveles de Procesamiento de Lenguaje Natural



Nota: La figura muestra los niveles del procesamiento de lenguaje natural. A continuación se dará un breve concepto siguiendo el orden que muestra la figura 3. La elaboración es propia y la fuente corresponde a datos tomados de Ramos, F. & Velez, J. (2016).

Nivel Pragmático

El nivel pragmático es aquel que examina las diferentes variantes relevantes para la compresión de un escrito o para demostrar en función de las causas contextuales la opción de determinadas maneras de llevarlo a cabo y usa el contexto sobre los contenidos del texto para la percepción (Ramos & Velez, 2016).

Nivel de Discurso

El nivel de discurso a diferencia de otros niveles se ocupa de trabajar con unidades de texto más grande, se enfoca en el texto como un todo para lograr obtener el significado estableciendo conexiones entre las oraciones. En este nivel se realizan dos métodos, el primero que es resolución de anáforas que se basa en sustituir pronombres con la entidad a la cual hacen referencia, mientras

que el segundo método que es el reconocimiento de la estructura del texto intenta reconocer la función que tiene cada oración en el texto (Ramos & Velez, 2016).

Nivel Semántico

El nivel semántico se encarga de conseguir en una oración el sentido desde la interacción a través de las palabras que la constituyen, el procesamiento semántico acepta un solo sentido a las palabras con diferentes significados de esta manera incorporar el sentido en la función semántica de la oración (Ramos & Velez, 2016).

Nivel Sintáctico

El nivel sintáctico analiza la representación de cada palabra dentro de una oración manifestando de esta manera la organización o estructura. Como resultado de este proceso se obtiene la representación de la oración examinada que presentara la relación entre las palabras que la forman (Ramos & Velez, 2016).

Nivel Léxico

El nivel léxico se hace cargo de forma individual del significado de cada palabra, para realizarlo cada palabra debe tratarse por si misma y dependiendo del contexto en el que se encuentra etiquetarla con parte del discurso. El objetivo del nivel léxico es observar cada una de las palabras para conocer su significado y la función que cumple dentro de una oración (Ramos & Velez, 2016).

Nivel Morfológico

El nivel morfológico estudia la composición de las palabras. Teniendo en cuenta este concepto el PLN es apto de separar una palabra y conseguir el significado por medio del significado de cada uno de sus morfemas (Ramos & Velez, 2016).

Nivel Fonémico

El nivel fonémico es conocido como fonemas que son las unidades teóricas fundamentales para aprender el nivel fonológico de la lengua humana. Es decir que estudian la variación en la pronunciación cuando las palabras están enlazadas, este análisis fonológico se emplea en casos en donde la entrada es verbal (Ramos & Velez, 2016).

Nivel Fonético

El nivel fonético se encarga del estudio o interpretación del sonido dentro de las palabras.

Modelos del Procesamiento de Lenguaje Natural

Modelo Lógico

Estos modelos recogen su esencia para aplicarla a la comunicación entre maquina y persona, estos son los creados por los lingüistas especializados basándose en determinadas formas gramaticales. Esto se combina con la información que ya existe en los diccionarios de computación para definir los modelos de resolución de la tarea concreta a través del proceso de automatización de los procesos conversacionales, de esta manera es posible que las maquinas puedan reconocer diferentes patrones de estructuras lingüísticas (IMF Business School, 2020).

Modelo Probabilístico

Este modelo se basa en los datos como eje principal del análisis, los lingüistas recopilan grandes volúmenes de información para analizar y establecer la frecuencia de aparición de todas las unidades que forman la lengua de forma los algoritmos establecen la probabilidad de que aparezcan en un contexto como posibles respuestas. Se lo denomina Aprendizaje Automático en Inteligencia Artificial y en este modelo el proceso se realiza sin tener que valorar los requisitos establecidos en las reglas gramaticales (IMF Business School, 2020).

Word Mover Distance

WMD mide la distancia total que las incorporaciones de palabras (*Word Embeddings*) de dos textos deben pasar por un proceso para convertirse en idénticas, aprovecha los resultados de las técnicas de incrustación avanzada como Word2Vec que aprende de representaciones semánticamente significativas de palabras a partir de coocurrencias locales en oraciones (Medrano, 2020).

Utiliza *Word Embeddings* de las palabras en dos textos para medir la distancia mínima que las palabras de un texto necesitan para viajar en el espacio semántico para llegar a las palabras del otro texto. El WMD se obtiene midiendo el mínimo de *Earth Mover Distance* (Distancia del movimiento de la tierra) entre cada palabra en los dos documentos en el espacio word2vec, si la distancia es pequeña las palabras de los dos documentos están cerca una de otras (Tarek, 2020).

Por ejemplo, como se muestra en la Figura 4 se aprecian dos documentos con una oración:

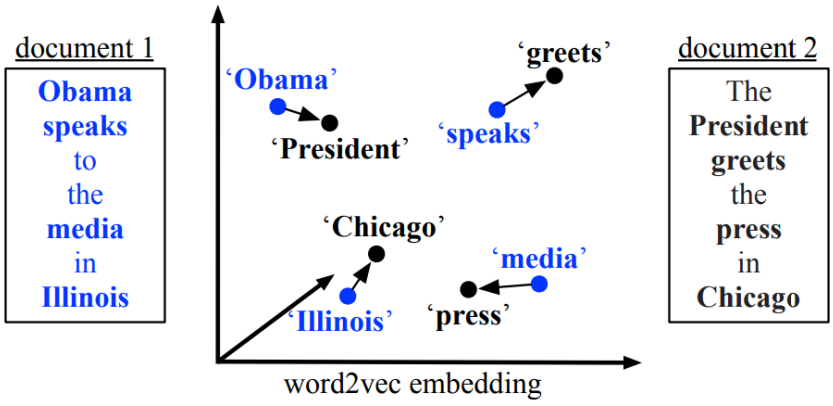
Documento 1: “Obama speaks to the media in Illinois” en español “Obama habla con los medios de comunicación en Illinois”

Documento 2: “The president greets the press in Chicago” en español “El presidente saluda a la prensa en Chicago”

Luego de eliminar las palabras stop, la distancia de movimiento de la palabra es 1.2 el cual es un valor pequeño el cual indica que son similares.

Figura 4

Word Mover Distance incrustado en un espacio Word2Vec para dos oraciones



Nota: Ejemplo de Word Mover Distance. Todas las palabras en negritas de ambos documentos estan incrustadas en un espacio Word2Vec, la distancia entre ambos documentos es la distancia acumulada minima en que las palabras del documento viajan para coincidir con las palabras del otro documento. Tomado de (Tarek, 2020).

Similitud Semántica

Dos términos semánticamente similares son aquellos cuyo significado semántico es aproximado, por ejemplo “coche” y “vehículo” que son semánticamente similares. La similitud semántica se entiende de manera generalizada como el grado de proximidad taxonómica entre dos términos, es decir que esta se establece por la semejanza entre conceptos (Alonso Martínez, 2016).

Se basa en la idea de que las distancias entre palabras en un espacio de incrustación se pueden evaluar mediante juicios heurísticos humanos sobre las distancias semánticas reales entre estas palabras, el evaluador recibe un conjunto de pares de palabras y se le pide evaluar el grado de similitud para cada par. La distancia de estos se recoge en un espacio de inserción de palabras y se comparan los dos conjuntos de distancias obtenidos, entre más similares sean mejor serán las incorporaciones (Bakarov, 2018).

Traducción Automática Neuronal

Traducción Automática Neuronal (del inglés, *Neuronal Machine Translation*) es una aproximación a la TA basada en corpus que en muchos casos provee mejores resultados que la TAE. Una particularidad de la TAN es que las palabras y las frases son representadas de forma numérica mediante vectores, mientras que en las otras aproximaciones la representación era discreta, lo que proporciona el uso de robustas técnicas de aprendizaje automático como las redes neuronales (Casacuberta Nolla & Peris Abril, 2017).

Generalmente la arquitectura de los modelos de TAN se basa en un codificador y decodificador, las cuales presentan las RNN que permiten predecir la palabra que aparecerá en la siguiente secuencia mediante la asociación de la palabra previa, respecto a la traducción la RNN toma el idioma de origen como la secuencia de entrada y el idioma objetivo como la secuencia de salida dando como solución una traducción holística contrario a una traducción de palabra por palabra (Huarcaya Taquiri, 2020).

Aprendizaje Supervisado

El aprendizaje supervisado es un modelo en donde se estudian las funciones y relaciones que asocian entradas con salidas, dentro de esta hay una subcategoría de modelos que son: modelo de clasificación que es cuando la salida es una enumeración o un conjunto de clases, es decir que su salida es un valor categórico. Mientras que el modelo de regresión es cuando la salida es un valor de un espacio continuo (Caparrini, 2020).

Existen varios algoritmos de aprendizaje supervisado clasificados en dos tipos: Clasificación y Regresión. Clasificación es cuando la salida es uno de los valores de un conjunto finito de datos como por ejemplo el conjunto: “soleado, nublado, lluvioso”, mientras que

Regresión es cuando la salida es un número como por ejemplo el valor de la temperatura del día de mañana (Páez Juka, 2019).

Algoritmo de Aprendizaje Supervisado (Clasificación)

En los algoritmos de clasificación las categorías corresponden a los valores de los conjuntos recurrentes. Al igual que el algoritmo de regresión es imprescindible reconocer las variables independientes, la fórmula y los parámetros con los que se va a predecir la variable independiente la cual en este caso solo puede aceptar valores discretos (Rodríguez, 2018).

Algoritmo de Aprendizaje Supervisado (Regresión)

En los algoritmos de regresión los valores de los conjuntos a reproducir son un valor continuo, se escoge las particularidades con las que se desarrollarán las predicciones las cuales se designan como variables independientes, una vez ejecutado se define la fórmula matemática y se calcula los parámetros de forma que al incluir las características se obtenga el valor esperado (Rodríguez, 2018).

Incorporaciones de Palabras

Incorporaciones de palabras (*Word Embeddings*, en inglés) son una serie técnicas de modelado de lenguaje y aprendizaje de características en PLN donde las palabras o frases del vocabulario se establecen a vectores de números reales, las palabras que se emiten del mismo contexto o de uno similar puede agruparse entre sí. Los algoritmos de aprendizaje automático necesitan la transformación de palabras a números ya que piden en su entrada vectores de valores continuos en lugar de cadenas de texto sin formato. Un método de modelado de PLN de *Word Embedding* es word2vec (Calibar , Holleger, & Klenzi, 2018).

Los modelos de *Word Embedding* en donde las palabras se incrustan en vectores de bajo valor dimensional con valores reales, semánticamente estas palabras similares tienden a estar cerca

en ese espacio vectorial, las incorporaciones de palabras entrenadas se pueden utilizar directamente para resolver tareas intrínsecas como la similitud de palabras y la analogía de palabras (Bofang, et al., 2019).

Word2vec

Es una técnica predictiva de incrustación de palabras (Word Embeddings) que convierte una palabra en un vector de números basado en el contexto de la palabra de destino. Los vectores de las palabras se generan utilizando las palabras circundantes para representar las palabras de destino, utiliza la red neuronal cuya capa oculta codifica la representación de la palabra (Saxena, 2019).

W2V son modelos de redes neuronales que se hallan capacitados para inferir el significado de las palabras según su contexto. Este modelo se entrena a partir de un gran conjunto de textos y genera un espacio vectorial, asignándole a cada palabra de los textos su respectivo vector en el espacio (Calibar , Holleger, & Klenzi, 2018).

Redes Neuronales Artificiales

Las RNA (ANN del inglés, *Artificial Neural Networks*) es un esquema de computación distribuida o computación en malla, que se inspira en la estructura del sistema nervioso de un ser humano. Una red neuronal contiene múltiples procesadores elementales conectados conocido como neuronas que forman el sistema adaptativo el cual mediante un algoritmo de aprendizaje es capaz de ajustar los pesos sinápticos para alcanzar los requerimientos de desempeño de un problema dado este proceso se denomina entrenamiento o aprendizaje (Jarrín Rodríguez, 2019).

Aprendizaje Profundo

Es un subcampo del *Machine Learning* y es una de las aplicaciones con mayor crecimiento de la IA, se usa para solucionar problemas que comprometen cantidades muy considerables de datos o que son muy complejos. Se realiza utilizando redes neuronales artificiales. Estas se estructuran en capas de modo que se pueda observar relaciones y reconocer patrones de datos. Para su uso se necesita de una gran cantidad de información y una gran capacidad de procesamiento (Rouhiainen, 2018).

Es una técnica popular y se aplica en tareas de procesamiento de lenguaje natural e ingeniería de software, mediante el procesamiento de autoaprendizaje se puede identificar patrones ocultos como dinámicas subyacentes e información semántica de datos. Hay tres modelos populares de aprendizaje profundo que son: Redes Neuronales Convolucionales, Redes Neuronales Recurrentes y Redes de Memoria a corto y largo plazo. Estos modelos son utilizados para resolver las tareas de PLN como clasificación de texto y análisis de opiniones (Wang, Zhang, & Jing, 2020).

Redes Neuronales Convolucionales (*Convolutional Neural Networks*)

CNN es una red neuronal que trata de conllevar de manera similar a como actúan las neuronas de la corteza visual del cerebro, se determinan por ser redes multicapa cuyas capas radican en matrices bidimensionales que del mismo modo en que el cerebro es capaz mediante un entrenamiento previo y apropiado reconocer desde simples formas la texturas o colores para así definir a que corresponde la imagen y realizar una clasificación. Las capas convolucionales pueden comprender como filtros para las imágenes de entrada cuya convolución con la misma producen una activación probable de la neurona y un conjunto de activaciones origina una salida conveniente a una activación (Suárez Lamadrid, 2018).

Redes Neuronales Recurrentes (*Recurrent Neural Networks*)

RNN es una red neuronal dinámica ya que el cálculo de una entrada en un paso determinado depende del paso anterior y en ciertos casos dependen de un paso futuro. Este tipo de redes neuronales son empleadas para el análisis de trayectorias, tratamiento de secuencias, predicciones no lineales y modelación de sistemas dinámicos (Jarrín Rodríguez, 2019).

Esta se clasifica en dos según el grado de conexión de las neuronas que conforman las redes recurrentes:

- Parcialmente recurrentes, estas son fijas y sus conexiones por lo general son hacia adelante, pero incluye un conjunto de conexiones de retroalimentación.
- Totalmente recurrentes donde cada neurona puede estar conectada a cualquier otra neurona y las conexiones recurrentes son variables

Redes Neuronales Recurrentes de memoria a corto y largo plazo (LSTM)

LSTM es una arquitectura RNN, diseñado para modelar secuencias temporales en dependencias de largo alcance, surge para solventar algunas dificultades presentadas en las redes totalmente recurrentes. Las redes LSTM tienen unidades especiales llamadas bloques de memoria en la capa recurrente oculta, cada bloque contiene una celda de memoria con auto conexiones que almacena el estado temporal de la red, también cuenta con unidades multiplicativas (*gates*) que controlan el flujo de información de la red y cada bloque tiene un *gate* de entrada y salida (Jarrín Rodríguez, 2019).

Python

Python es un lenguaje de programación de alto nivel el cual es legible y sencillo de aprender, también es un lenguaje multiplataforma que puede utilizarse en varios sistemas operativos. Es un lenguaje que soporta paradigmas de programación funcional y programación

imperativa. Dispone de herramientas para aprovechar los medios concretos que le brinda cada plataforma por medio del uso de bibliotecas. Es un software libre y se asigna bajo licencia “Python Software Foundation License” lo que indica que se puede obtener de manera gratuita y no necesita para su uso pago de licencias ya sea este privado o comercial (Hinojosa Gutiérrez, 2016).

StackAPI

Son los contenedores de Python para la API de Stack Exchange, esta biblioteca tiene soporte para el parámetro de retroceso de Stack Exchange, puede recuperar varias páginas de resultados con una sola llamada y combina todos los resultados en una sola respuesta. Tiene también funcionalidad de lectura y escritura a través de la API y produce excepciones devueltas para facilitar la solución de problemas (Wegner, 2019).

Numpy

Es una librería de Python especialista en el cálculo numérico y el análisis de datos en especial para grandes volúmenes de dato, incorpora una clase de arrays que permiten representar colecciones de datos de un mismo tipo en varias dimensiones y funciones eficientes para su manipulación. Las dimensiones de un array se conocen también como ejes y un array es una estructura de datos de un mismo tipo que están organizada en forma de tabla en distintas dimensiones (Sánchez Alberca, La librería Numpy, 2020).

Numpy facilita una estructura de datos multidimensional, varios objetos obtenidos y una diversidad de procedimientos rápidos en matrices en los que se tienen: matemáticas, lógicas, manipulación de formas, clasificación, selección, transformaciones discretas de Fourier entre muchos otros. El principal componente Numpy es el objeto “ndarray” el cual contiene “n” matrices

de datos con varias operaciones que se realizan en código compilado para el rendimiento (NumPy, 2020).

Pandas

Es una librería encargada del manejo y análisis de datos, Pandas concede estructuras de datos flexibles y permite trabajar con ellos de manera eficiente. Pandas dispone de tres estructuras de datos que son: Series, DataFrame y Panel (Panel4D y PanelIND). Para trabajar con la librería Pandas es también indispensable instalar la librería Numpy desde el repositorio de paquetes (Moya, 2020).

Series: son arreglos indexados de una sola dimensión (arrays con índice o etiquetados) parecidos a los diccionarios y se pueden generar a partir de listas.

DataFrame: son estructuras muy parecidas a las tablas de bases de datos relacionales como SQL.

Panel: son estructuras de datos y trabajan con más de dos dimensiones.

Las particularidades de esta librería es que determina nuevas estructuras de datos basados en los arrays de la librería Numpy, pero con nuevos funcionamientos. Proporciona leer y escribir sencillamente ficheros en formato CSV, Excel y bases de datos SQL. Propone métodos para reordenar, dividir y combinar conjuntos de datos. Permite acceder a los datos mediante índices, también permite trabajar con series temporales y elabora estas operaciones de forma eficiente (Sánchez Alberca, La librería Pandas, 2020).

NLTK

NLTK o Natural Language Toolkit es una librería de Python para PLN. Es una plataforma para creación de programas de análisis de texto, esta librería fue creada con fines educativos por lo que permite realizar proyectos con diversos objetivos y alcances. Su instalación en Python es sencilla se instala con una simple instrucción lo indispensable para emplearlo.

NLTK integra más de 50 corpus y un grupo de librerías para segmentación, tokenización, etiquetado del habla, análisis sintáctico, semántico y proporciona demostraciones paso a paso de distintos algoritmos, sin embrago muchas de esas opciones no están disponibles en español (Talamé, Cardoso, & Amor, 2019).

Tokenización: es una cadena de caracteres (signo de puntuación o palabra) que tiene algún significado en contexto de un texto o el procedimiento de distribuir un documento de texto en diferentes componentes eliminando los espacios en blancos y los saltos de línea. Un ejemplo de tokenización: “El niño corre” tiene tres tokens: “El”, “niño”, “corre” (Talamé, Cardoso, & Amor, 2019).

Segmentación: se basa en clasificar el texto en fracciones que se puedan tratar de manera independiente, la forma común es distribuir el texto en párrafos u oraciones.

Keras

Es una librería de redes neuronales en Python entre sus características esta qué es un prototipado sencillo y ágil debido a su modularidad, minimalismo y extensibilidad. Keras resiste tanto redes neuronales convolucionales como recurrentes, asimismo también una composición de ambas. Keras soporta bosquejos de conectividad arbitrarios incluyendo preparación multi-entrada y multi-salida, puede correr en CPU y GPU (Antona Cortés, 2017).

Para Gasca Meza, Keras: “permite evolucionar y determinar modelos de *Deep Learning*, soporta librerías de computación numérica como Theano y TensorFlow, permite definir y preparar modelos de redes neuronales en pocas líneas de código” (Gasca Meza, 2018).

Gensim

Es una librería de Python para el PLN. Esta contiene herramientas de modelado de temas y espacio vectorial, es decir qué puede reconocer de manera autónoma de qué se trata un conjunto de documentos. Es apropiado al momento de elaborar o importar representaciones de vectores como W2V. Además, se puede utilizar para examinar el parecido entre documentos lo que resulta rentable cuando se realizan búsquedas (Martinez Heras, 2020).

Servicios de Google

GoogleTranslate

Google Translate permite traducir desde frases hasta textos enteros de un idioma a otro, actualmente permite traducir textos en 103 idiomas, en el transcurso del tiempo ha mejorado su funcionamiento obteniendo mejor calidad en las traducciones. Google Translate utiliza un tipo de traducción llamado traducción automática basada en redes neuronales (Restrepo Klinge, 2019).

Google Colab

Google Colab o Colaboratory es un ambiente de máquinas virtuales fundamentado en Jupyter Notebooks, se ejecuta en la nube y es posible elegir nuestro notebook en una CPU, GPU o en una TPU de forma gratuita. Son recomendables para principiantes que quieran trabajar con *machine learning* y *Deep learning*, pero sin incidir en costos de proceso cloud. Viene con varias librerías instaladas como Tensorflow (Martinez, 2019).

En Google Colab se trabaja con Python que permite su uso para diferentes paradigmas de programación como programación funcional u orientada a objetos, se utiliza como lenguaje de scripting y es un lenguaje interpretado. Su funcionalidad se quedó corta para los desarrolladores por lo que apareció IPython que evoluciono en Jupyter el cual es un ambiente interactivo que posibilita desarrollar código Python de forma dinámica, se efectúa como una aplicación cliente-servidor y facilita tanto la realización del código como la escritura de texto (de la Fuente Sanz, 2019).

Google Drive

Es una plataforma de almacenamiento en la nube creada por Google la cual integra una serie de herramientas de productividad como editor de texto, hoja de cálculo, presentaciones y calendario. Sobresale por proponer diversas funciones en un mismo paquete ya que facilita la compartición de archivos de modo que permite el trabajo colaborativo. Tiene la función de sincronización, su contenido se almacena en servidores sin necesidad de la intervención del usuario. Google Drive se integra con otras herramientas como Gmail, Drive, Docs, Sheets, Hangouts, Agenda, Contactos, Google Sites, herramienta de anotación (Keep), editor y administrador de formularios, entre otros (Pereira, 2020).

Revisiones sistemáticas

Como metodología para realizar la revisión sistemática se ha elegido el estudio de mapeo sistemático. Este al igual que la revisión sistemática de la literatura se considera efectivo al momento de realizar la búsqueda de herramientas, tecnologías, conceptos y métodos existentes de modo que se puedan tomar decisiones racionales y científicas al momento de realizar una investigación de ingeniería de Software (Barn, Barat, & Clark, 2017).

Para realizar el estudio de mapeo sistemático se tomará como referencia el trabajo realizado por Botto-Tobar et al. Según el autor un mapeo sistemático permite la categorización y resumen de la información disponible con respecto a una pregunta de investigación. Entre las fases que se tiene para llevar a cabo un estudio de este tipo tenemos: Planeación, dirección y reporte. (Botto-Tobar, Ramirez Anormaliza, Cevallos Torres, & Cevallos Ayon, 2017).

Pregunta de investigación

Con el fin de encontrar y analizar los avances más importantes y recientes en los campos relacionados al desarrollo del proyecto y para enfocar la revisión sistemática hacia el objetivo general del proyecto se ha definido la siguiente pregunta de investigación:

¿Cómo los investigadores y profesionales utilizan el aprendizaje automático para detectar la duplicidad entre preguntas de programación realizadas en lenguaje natural y diferentes idiomas?

Para que la pregunta de investigación se pueda abordar y responder de manera más detallada y precisa se ha descompuesto en las siguientes sub-preguntas:

S-PI1: ¿Qué métodos de aprendizaje automático se utilizan para detectar la duplicidad entre preguntas en lenguaje natural?

S-PI2: ¿Qué herramientas existen para detectar la duplicidad entre preguntas realizadas en sitios de preguntas y respuestas?

S-PI3: ¿Qué artefactos se utilizan en el campo del aprendizaje autónomo para detectar preguntas duplicadas en lenguaje natural?

Estrategia de búsqueda

Para la búsqueda de estudios primarios se seleccionó la plataforma Google Académico debido a que indexa de manera efectiva una gran cantidad de publicaciones científicas obtenidas de diversas librerías digitales como IEEExplore, ACM Digital Library, Research Gate, Springer Link y varias más. Google Académico además indexa literatura de otras fuentes como sociedades de profesionales, repositorios, instituciones académicas y otros sitios web (Google) por lo que resulta una herramienta bastante apropiada para realizar una búsqueda contando con una amplia base de información.

Aprovechando las capacidades del motor de búsqueda y su función de "Búsqueda avanzada" se definieron las cadenas de búsqueda presentadas en la Tabla 4 con el fin de realizar una búsqueda automática de la literatura disponible cuyas coincidencias con las cadenas de búsqueda utilizadas estén presentes solo en el título. La búsqueda fue realizada en febrero de 2021 tomando en consideración solo estudios realizados a partir del año 2008, ya que este fue el año de lanzamiento de Stack Overflow, hasta el año 2020 que al momento de realizar la búsqueda ya ha culminado.

Tabla 4

Cadena de Búsqueda

Código	Cadena de búsqueda	Idioma de búsqueda
CB-1	allintitle: "stack overflow"	Español
CB-2	allintitle: duplicate question OR questions "stack overflow"	Inglés
CB-3	allintitle: duplicate question OR questions	Inglés
CB-4	allintitle: question duplicate multilingual OR "dual language" OR "cross language" OR "cross lingual" OR multilanguage OR multilingual	Inglés

CB-5	allintitle: question multilingual OR dual-language OR cross-language OR cross-lingual OR multilanguage OR multilingual	Inglés

Nota: La tabla refleja la definición de la cadena de búsqueda para la búsqueda automática de la literatura disponible. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Selección de estudios primarios

Luego de realizar la búsqueda en Google Académico utilizando las cadenas presentadas previamente se obtienen distintas cantidades de resultados, y por lo tanto estudios, por cada cadena de búsqueda. Para decidir cuál de los estudios debe ingresar al grupo de estudios primarios estos deben cumplir por lo menos uno de los criterios de inclusión definidos a continuación:

-CI1: Estudios que mencionen métodos de aprendizaje autónomo para detectar preguntas duplicadas en sitios de Q&A.

-CI2: Estudios que presenten herramientas utilizadas en la detección de preguntas duplicadas.

-CI3: Estudios que presenten técnicas para encontrar la similitud entre preguntas de diferentes idiomas.

Del mismo modo se definieron criterios de exclusión con el objetivo de descartar estudios que no proporcionen información relevante a la revisión. Estos criterios se definen a continuación:

-CE1: Estudios que no estén enfocados a la detección de preguntas duplicadas exclusivamente en sitios Q&A de programación.

-CE2: Estudios presentando el mismo estudio que uno previamente seleccionado, pero en una fuente distinta.

-CE3: Estudios cuya visualización es limitada o no está disponible de manera gratuita.

En cuanto a la evaluación de la calidad, se decidió no realizarla puesto a que no aportaba una mejora a los resultados obtenidos en el estudio de mapeo. Esta decisión fue tomada

considerando que los criterios de inclusión y exclusión pueden llegar a ser muy estrictos. Debido a esto no es necesario utilizar criterios de calidad para excluir estudios ya que esto podría reducir aún más los estudios en los que se basarán los resultados de modo que no se podrían generalizar como lo menciona Tebes et al. (2020).

Estrategia de extracción de datos

Tomando como referencia la estrategia de Botto-Tobar et al. (2017, pp. 6-8) se definieron 6 criterios de extracción basados en las posibles respuestas que se puedan obtener para cada sub-pregunta de investigación. Cada uno de estos criterios se explican en detalle a continuación de modo que se establezca su correspondencia con las sub-preguntas de investigación que se buscan responder como se muestra en la Tabla 5.

Tabla 5

Sub-Preguntas y Criterios considerados en la estrategia de extracción de datos

Sub-pregunta de Investigación	**Criterios**	**Opciones**
S-PI1: ¿Qué métodos de aprendizaje automático se utilizan para detectar la duplicidad entre preguntas en lenguaje natural?	Cext1: Métodos de aprendizaje automático	Clasificación Procesamiento de lenguaje Natural Redes Neuronales Incorporaciones de palabras
	Cext2: Tipos de aprendizaje automático	Supervisado No supervisado Profundo
	Cext3: Algoritmos de aprendizaje automático	Regresión logística Greedy Word2Vec Bosques Aleatorios Traducción Automática

		Análisis de componentes principales
S-PI2: ¿Qué herramientas existen para detectar la duplicidad entre preguntas realizadas en sitios de preguntas y respuestas?	Cext4: Herramientas de detección de duplicados	Dupe DupPredictor WVTool BM25 XSearch
S-PI3: ¿Qué artefactos se utilizan en el campo del aprendizaje automático para realizar la detección de preguntas duplicadas en lenguaje natural?	Cext5: Características (Features) de entrenamiento	Similitud por coseno Asignación latente de Dirichlet Representaciones vectoriales de palabras
	Cext6: Componentes de entrada	Titulo Cuerpo Etiquetas Tópico

Nota: La tabla refleja cada sub-pregunta considerada para la elaboración la cual tiene relación con la pregunta de investigación. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Con respecto a la sub-pregunta de investigación 1 (S-PI1) se definieron los criterios de extracción Cext1 - Cext3 detallados a continuación:

-Cext1: Métodos de aprendizaje automático: Debido a la enorme ganancia de interés que ha tenido IA en los años recientes como menciona Hozinger et al. (2018), existe una considerable cantidad de métodos que se han desarrollado dentro de este campo. Debido esto, se decidió considerar solo 4 métodos de aprendizaje automático para clasificar los estudios ya que estos tienen una mayor probabilidad de ser utilizados en una tarea de detección de preguntas duplicadas.

1. Clasificación: Si el estudio presenta su enfoque como un problema de clasificación.
2. Procesamiento de lenguaje natural: Si el estudio presenta un enfoque que hace uso de técnicas de procesamiento del lenguaje natural en el proceso.
3. Redes neuronales: Si el enfoque del estudio incluye el uso redes neuronales.

4. Incorporaciones de palabras: Si el enfoque del estudio se basa en la utilización de incorporaciones de palabras.

-Cext2: Tipos de aprendizaje automático: A pesar de que existen numerosos tipos de aprendizaje como los mencionados por Brownlee (2019), para este criterio de extracción solo se incluirán 3 tipos de aprendizaje ya que pueden considerarse los más conocidos e importantes. Estos son:

1. Supervisado: Si el estudio utiliza algoritmos que requieren entrenamiento con datos etiquetados.
2. No supervisado: Si el estudio utiliza algoritmos que realizan la extracción de relaciones entre los datos.
3. Profundo: Si el paper estudio algoritmos que detecta patrones en los datos a través de varias capas de procesamiento.

-Cext3: Algoritmos de aprendizaje automático: Dependiendo del problema que se desee abordar existen varios algoritmos de aprendizaje automático que se pueden aplicar. Para la tarea de detección de preguntas duplicadas se consideraron 6 posibles algoritmos para la categorización de los estudios.

1. Regresión logística: Estimación de valores discretos basada en conjuntos de variables independientes (Ray, 2017).
2. Greedy (Voraz): Generalización de soluciones optimas locales a globales.
3. Word2Vec: Incorporaciones de palabras mediante una red neuronal.
4. Bosques Aleatorios: Clasificación mediante conjuntos de árboles de decisión.
5. Traducción automática: Traducción de textos a través de modelos estadísticos o redes neuronales.

6. Análisis de componentes principales: Descripción de un conjunto de datos en términos de nuevos componentes o variables.

Para la pregunta S-PI2 se definió solo un criterio correspondiente a las herramientas utilizadas en la detección de preguntas duplicadas en sitios Q&A de programación. El criterio Cext4 se describe a continuación:

-Cext4: Herramientas de detección de duplicados: Este criterio se refiere a las herramientas propuestas utilizadas de manera activa en el estudio con el fin de determinar la duplicidad entre pares de preguntas. Las herramientas seleccionadas como opciones para la categorización son:

1. Dupe: Herramienta basada en caracterización y clasificación con regresión logística.
2. DupPredictor: Herramienta basada en caracterización y clasificación con algoritmo Greedy.
3. WVTool: Librería para modelado estadístico de lenguaje.
4. BM25: Algoritmo de jerarquización de documentos relevantes.
5. XSearch: Herramienta de recuperación de preguntas multilingües y de dominio específico.

La pregunta de investigación S-PI3 pretende ser respondida a través de los criterios de extracción Cext5 y Cext6 en los cuales se hace referencia a los artefactos utilizados para la tarea de detección de preguntas duplicadas. Los detalles de cada criterio de extracción se presentan a continuación:

-Cext5: Características (*Features*) de entrenamiento: En el contexto actual las características se definen como los valores o campos generados con el fin de representar el

estado de una observación dentro de un conjunto de datos. Las características pueden ser tan variadas como el problema necesite. En este caso se seleccionaron 3 opciones descritas a continuación:

1. Similitud por coseno: Valor calculado a través del coseno de dos textos representados en un espacio vectorial.
2. Asignación latente de Dirichlet: Representación del tópico de un texto a través de un modelo de generación estadístico.
3. Representaciones vectoriales de palabras: Conjunto de valores que representan las características de un texto a través de sus palabras.

-Cext6: Componentes de entrada: En la detección de preguntas duplicadas es inevitable no hablar de los componentes de las preguntas definidas en los sitios Q&A a manera de publicaciones. Desde puntajes de relevancia hasta las propias respuestas, los componentes pueden ser utilizados de diversas maneras para entender y procesar una pregunta. Los componentes seleccionados para la categorización son:

1. Título: Resumen de la pregunta definido por el autor de esta.
2. Cuerpo: Descripción de la pregunta definido por el autor.
3. Etiquetas: Palabras clave de la pregunta definidas por el autor.
4. Tópico: Representación del tópico generada a través de algoritmos.

Método de síntesis

Para sintetizar la información obtenida se eligió un método cuantitativo que consiste en realizar un conteo de los estudios primarios que clasifiquen dentro de cada pregunta de investigación. También se realizará un conteo de los estudios primarios agrupados por su año de publicación.

Conducción

Siguiendo el protocolo definido en las secciones anteriores se realizó la búsqueda de estudios potenciales en la plataforma de Google Académico con las cadenas de búsqueda e intervalos de tiempo definidos. Como se puede evidenciar en la Tabla 6, la cantidad total de estudios potenciales obtenidos como luego de la ejecución de la estrategia de búsqueda es de 175.

De igual manera, la Tabla 6 muestra que luego de considerar aquellos estudios que cumplían los criterios de inclusión y exclusión establecidos, se obtuvieron 10 estudios a los cuales se les aplicará la categorización respectiva mediante los criterios de extracción de datos definidos previamente.

Tabla 6

Total de estudios obtenidos

Fuente	Cadena de búsqueda	Estudios potenciales	Estudios seleccionados
Google Académico	CB-1	3	0
	CB-2	8	5
	CB-3	51	2
	CB-4	1	1
	CB-5	112	2
Total		175	10

Nota: La tabla refleja la cantidad de estudios seleccionados del resultado obtenido de la cadena de búsqueda y la evaluación. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

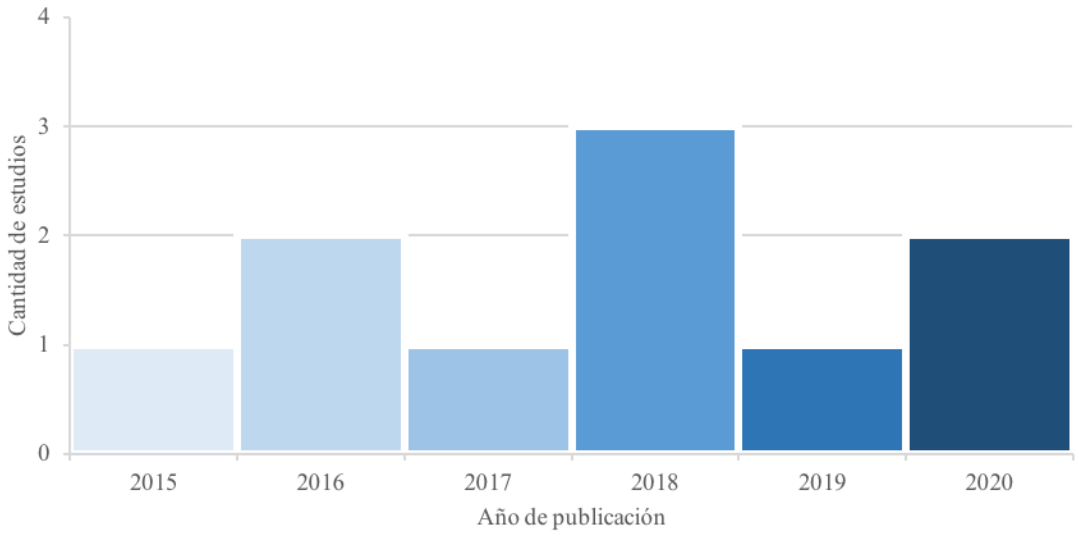
Resultados

Luego de considerar los criterios de extracción correspondientes y realizar la categorización de los estudios se obtuvieron los resultados presentados en la Tabla 7. Los estudios que han sido incluido y citados en esta sección como [E-X] pueden ser revisados en la Tabla 11 de la sección correspondiente.

Al realizar el conteo de estudios por año de publicación que está representado por la Figura 5 se puede observar que todos los estudios seleccionados son relativamente recientes. A pesar de que en la estrategia de búsqueda se decidió tomar en consideración estudios desde el año 2008, al considerar los criterios de inclusión y exclusión correspondientes se observa que el 90% de los estudios fueron publicados en los últimos 5 años del intervalo de tiempo definido para la búsqueda. De esto se puede concluir que el problema de detectar preguntas duplicadas en los sitos Q&A no se había abordado en los años cercanos al lanzamiento de Stack Overflow sino después del año 2012 que según Ahasanuzzaman, Asaduzzaman, Roy, & Schneider, fue en el año en que se evidenció un incremento significativo en el número de preguntas duplicadas en el mes de septiembre (Ahasanuzzaman, Asaduzzaman, Roy, & Schneider, 2016).

Figura 5

Gráfica de estudios por año de publicación



Nota: La figura 5 muestra los estudios por año de publicación. Se puede visualizar que los estudios seleccionados fueron publicados en los últimos 5 años. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

A continuación, se presentarán los detalles de los resultados obtenidos en cada uno de los criterios de extracción de datos que se definieron previamente.

Métodos de aprendizaje automático

Los resultados del criterio Cext1 muestran que el 70% de los estudios hicieron uso de las incorporaciones de palabras. Debido a la flexibilidad y a la variedad de aplicaciones de las incorporaciones de palabras estas se pueden combinar con otros métodos como Wang et al. [E-3] hizo al aprovechar los beneficios de las incorporaciones de palabras para mejorar el rendimiento de las redes neuronales que, dicho sea de paso, solo se hicieron presentes en el 20% de los estudios. Por otro lado, los métodos de procesamiento de lenguaje natural fueron utilizados en el 60% de los estudios y del mismo modo que con las incorporaciones de palabras, estos fueron combinados con otros métodos como en el caso de Zang et al. [E-6]. Por último, los métodos de clasificación fueron presentados en el 40% de los estudios. En algunos casos estos fueron utilizados en solitario como demostró Ahasanuzzaman [E-1] pero también se usaron en conjunción con otros métodos como hizo Siu [E-7].

Tipos de aprendizaje automático

En cuanto al criterio Cext2, los resultados muestran que el tipo de aprendizaje que predomina es el no supervisado ya que está presente en el 80% de los estudios. Esto probablemente sea debido a que sus aplicaciones sirven como un excelente complemento para mejorar el rendimiento de las propuestas como en el caso de Zhang [E-2]. El tipo de aprendizaje supervisado fue utilizado en el 70% de las propuestas, ya sea como complemento [E-7] o como principal [E-1]. Por otro lado, el tipo de aprendizaje profundo solo se utilizó en el estudio realizado por Wang et al. [E-3] representando así el 10% de los estudios.

Algoritmos de aprendizaje automático

El criterio Cext3 es el que más opciones de clasificación tiene dentro del conjunto. Sin embargo, en los resultados se puede evidenciar que el algoritmo predominante es Word2Vec ya que es utilizado en el 70% de los estudios debido a su utilidad para aprender las incorporaciones de palabras que pueden ser utilizadas como entrada para otros algoritmos como en [E-7] o ser utilizadas de manera independiente como en la propuesta de Babu et al. [E-5]. La regresión logística es el segundo algoritmo más utilizado con una presencia en el 40% de los estudios debido a su eficiencia en tareas de clasificación binaria [E-1]. Los algoritmos de traducción automática se utilizaron en el 30% de los estudios lo que concuerda con la poca cantidad de estudios encontrados que abordaban la tarea de detección de preguntas duplicadas desde un enfoque multilingüe [E-8][E-9][E-10]. Por otro lado, el algoritmo greedy, el de bosques aleatorios y el de análisis de componentes principales comparten un 20% en cuanto a su presencia en los estudios seleccionados.

Herramientas de detección de duplicados

Los resultados del criterio Cext4 demuestran que DupPredictor [E-2] es la herramienta a la que más se hace referencia en los estudios, probablemente por ser una de las primeras en su tipo y haber servido como línea base para estudios posteriores. Dupe y BM25 aparecen en el 30% de los estudios. WVTool fue utilizado en el 20% de los estudios mientras que XSearch solo aparece en el artículo que lo propone [E-9] representando el 10% y concordando con ser la única herramienta de detección de duplicidad multilingüe dentro del conjunto.

Características (Features) de entrenamiento

En los resultados del criterio Cext5 se puede evidenciar que la similitud por coseno es una de las características más utilizadas en los estudios ya que es una forma eficiente de asignar un valor de similitud a un par de preguntas. La similitud por coseno es utilizada como característica en el 70% de los estudios. Por otro lado, la asignación latente de Dirichlet y las representaciones de palabras aparecen en el 40% de los estudios ya que permiten obtener la representación del tema [E-2] y de la semántica palabras respectivamente [E-5].

Componentes de entrada

Como observación notable de resultados del criterio Cext6Entre se aprecia que el título y el cuerpo se utilizan en todos los estudios, llegando a alcanzar el 100% en cuanto a su presencia en los mismos. Esto debido a que son las principales fuentes de información relevante de las preguntas. Las etiquetas son utilizadas en el 70% de los estudios ya que suelen utilizarse como complementos del título y el cuerpo para el aporte de información [E-1]. El tópico aparece en el 30% ya que es un componente generado automáticamente [E-2] por lo que probablemente no aporte mucha información al ser derivado del título y el cuerpo.

Tabla 7

Resultados del Mapeo Sistemático

Sub-pregunta de Investigación	**Criterios**	**Opciones**	**Resultados**	
			#Estudios	**Porcentaje**
S-PI1	Cext1: Métodos de aprendizaje automático	Clasificación	4	40%
		Procesamiento de lenguaje Natural	6	60%
		Redes Neuronales	2	20%
		Incorporaciones de palabras	**7**	**70%**

	Cext2: Tipos de aprendizaje automático	Supervisado	7	70%
		No supervisado	**8**	**80%**
		Profundo	1	10%
	Cext3: Algoritmos de aprendizaje automático	Regresión logística	4	40%
		Greedy	2	20%
		Word2Vec	**6**	**60%**
		Bosques Aleatorios	2	20%
		Traducción Automática	3	30%
		Análisis de componentes principales	2	20%
S-PI2	Cext4: Herramientas de detección de duplicados	Dupe	3	30%
		DupPredictor	**4**	**40%**
		WVTool	2	20%
		BM25	3	30%
		XSearch	1	10%
S-PI3	Cext5: Características (Features) de entrenamiento	Similitud por coseno	**7**	**70%**
		Asignación latente de Dirichlet	4	40%
		Representaciones vectoriales de palabras	4	40%
	Cext6: Componentes de entrada	Titulo	**10**	**100%**
		Cuerpo	**10**	**100%**
		Etiquetas	7	70%
		Tópico	3	30%

Nota: La tabla refleja los resultados del mapeo sistemático, en la columna porcentaje se indica el valor relativo de cada opción en relación con la suma de los estudios por cada uno de los criterios. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Hipótesis

¿Pueden las técnicas de aprendizaje automático ayudar a detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español?

Variables de la investigación

Una variable de investigación no puede definirse como algo en específico ya que tiene un concepto bastante y complejo. Esta más bien logra definirse en un momento dado cuando el investigador la asume de acuerdo con el objetivo de la investigación y sus propios intereses (Carballo & Guelmes, 2016).

Las variables de investigación pueden ser clasificadas por diversos criterios. Para Carballo & Guelmes, las variables de investigación se pueden clasificar según su naturaleza, complejidad, función o relación y su nivel de medición. Dentro la clasificación según su función se encuentran las variables independientes y dependientes que son del tipo que será utilizado en este estudio (Carballo & Guelmes, 2016).

Variable independiente

Una variable independiente se define como aquella que es manipulada por el investigador con el objetivo de transformar el objeto de estudio (Carballo & Guelmes, 2016). En esta investigación la variable independiente corresponde a las *técnicas de aprendizaje automático*.

Variable dependiente

La variable dependiente de este estudio se ha definido como la *detección de preguntas duplicadas entre sitios.* Esto coincide con la afirmación de Carballo & Guelmes, que indica que una variable también puede definirse como el resultado de un proceso (Carballo & Guelmes, 2016).

Definiciones conceptuales

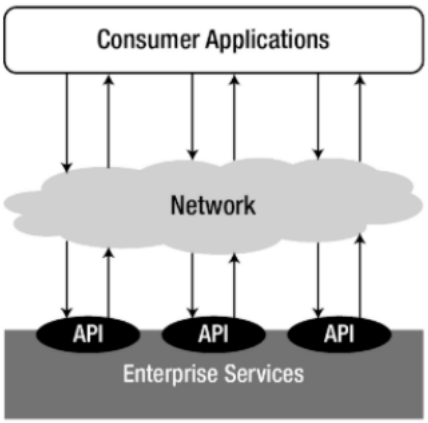
API (Application Programming Interface)

Una API es el término con que se conoce a las interfaces de programación de aplicaciones. Estas sirven como el medio mediante el cual distintas aplicaciones se puedan comunicar entre sí y se establece mediante una especie de contrato o reglas a seguir por parte de ambos extremos (Amengual Bauza, 2019).

Mientras que para Fernández la API se trata de las distintas definiciones y pasos a seguir con el fin de desarrollar software que se puede comunicar e integrar fácilmente con otros sin la intervención del usuario (Fernández, 2019).

Figura 6

Diagrama uso de APIs



Nota: la Figura muestra el diagrama de uso de API, en donde una API proporciona una interfaz para las aplicaciones que quieran consumirlas de esta manera las aplicaciones pueden interactuar con la información de los servicios que se encuentran detrás de estas APIs. La elaboración y la fuente corresponde a datos tomados de (Amengual Bauza, 2019).

Algoritmo

Un algoritmo se define según Robledano como ordenes que se ejecutan secuencialmente de modo que resulten en la solución de un problema o la ejecución de una tarea. Un algoritmo se puede presentar en diversos campos como en las matemáticas como el proceso para resolver una operación o en el caso de la programación que es la aplicación de algoritmos en la informática (Robledano, 2019). Mientras que para Marker, un algoritmo es una serie de instrucciones estructuradas, ordenadas y finitas que permiten solucionar un problema o llevar a cabo una tarea (Marker, 2020).

Dataframe

Se determina como un marco de datos ordenados, se podría pensar que un *dataset* y *Dataframe* son lo mismo ya que ambos son un conjunto de datos almacenados, pero la diferencia está en que un *Dataframe* a parte de tener la información ordenada por filas también lo hace por columnas (Barrera Bolivar, 2020).

Dataset

Un *Dataset* es un anglicismo que para Barrera Bolivar, se incorpora en el lenguaje español para denotar un conjunto ordenado de datos los cuales están tabulados e incluye también las relaciones entre entidades del modelo de datos (Barrera Bolivar, 2020).

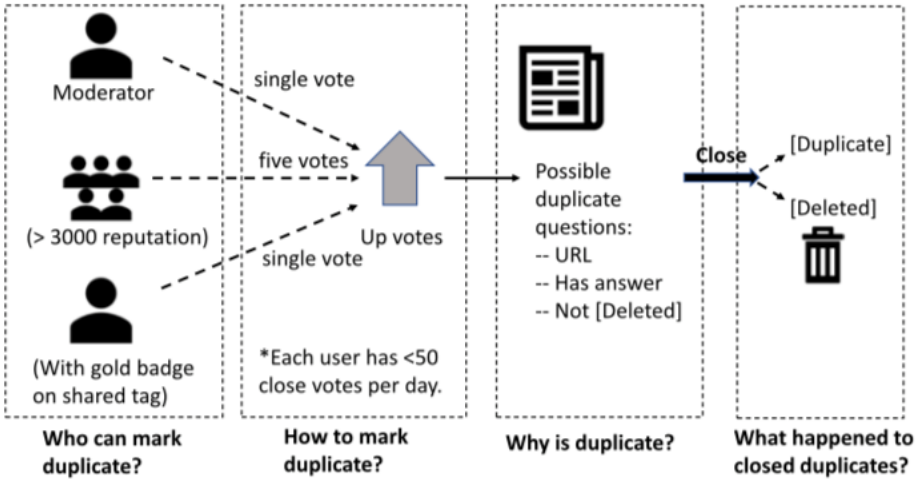
Duplicidad de preguntas en sitios Stack Overflow

Se dice que la duplicidad entre dos preguntas de Stack Overflow existe cuando ambas describen el mismo problema y por lo tanto una solución o respuesta válida sería suficiente para satisfacer las necesidades de ambos autores de las preguntas.

En la actualidad las preguntas duplicadas en los sitios son tratadas con un proceso manual que se describe en la Figura 7. Este proceso requiere que usuarios con una alta reputación en la comunidad voten a favor del cierre de la pregunta especificando el motivo del cierre. En este caso las preguntas que se cierran como duplicadas deben incluir una referencia a la pregunta original a la que algunos autores se refieren como la pregunta "maestra" (Zhang, Lo, Xia, & Sun, 2015).

Figura 7

Proceso para detectar duplicidad entre preguntas de Stack Overflow



Nota: La figura muestra el proceso manual para detectar duplicidad entre dos preguntas de Stack Overflow. La elaboración y la fuente corresponde a datos tomados de (Zhang, Sheng, Lau, Abebe, & Ruan, 2018)

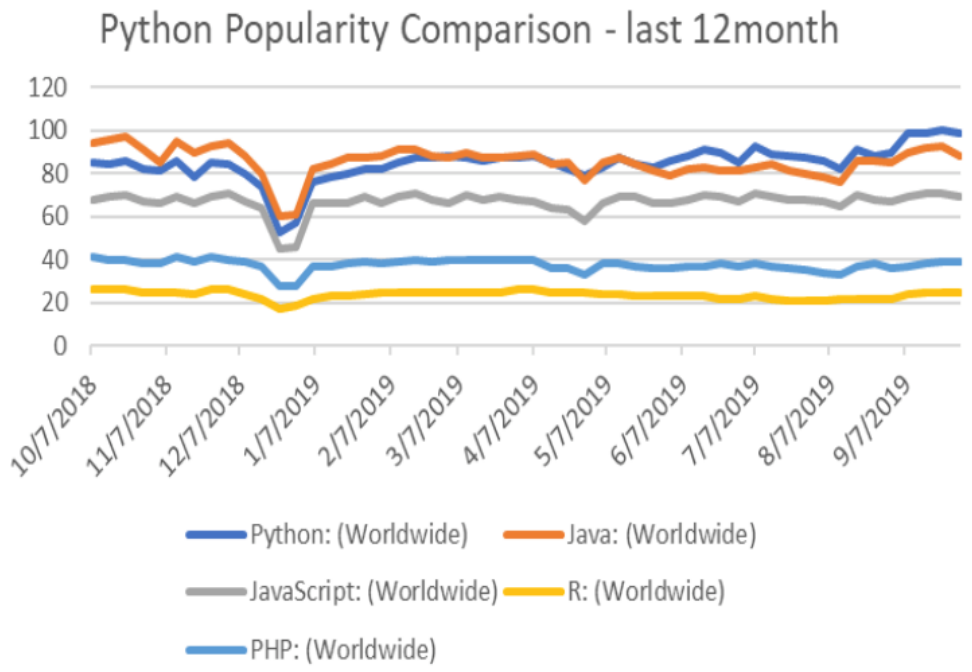
Lenguaje de programación

En este estudio el lenguaje se refiere a la herramienta básica para la construcción de el algoritmo a desarrollar. En este caso se utilizará el lenguaje Python. Como se puede aprecia en la Figura 8 este lenguaje ha ganado mucha popularidad en los últimos años siendo uno de los lenguajes que más prefieren los científicos de datos, estadísticos y desarrolladores de inteligencia artificial. Según Challenger-Pérez et al. el incremento en la popularidad de Python se debe a que existen varias librerías de visualización, procesamiento de señales, estadística y más que son

también fáciles de utilizar gracias también a su completa documentación (Challenger-Pérez, Díaz-Ricardo, & Becerra-García, 2014). Este lenguaje fue elegido debido a su simplicidad y eficiencia ya que permitirá enfocarse en el diseño del algoritmo y no en su desarrollo como tal.

Figura 8

Comparación de popularidad de Python



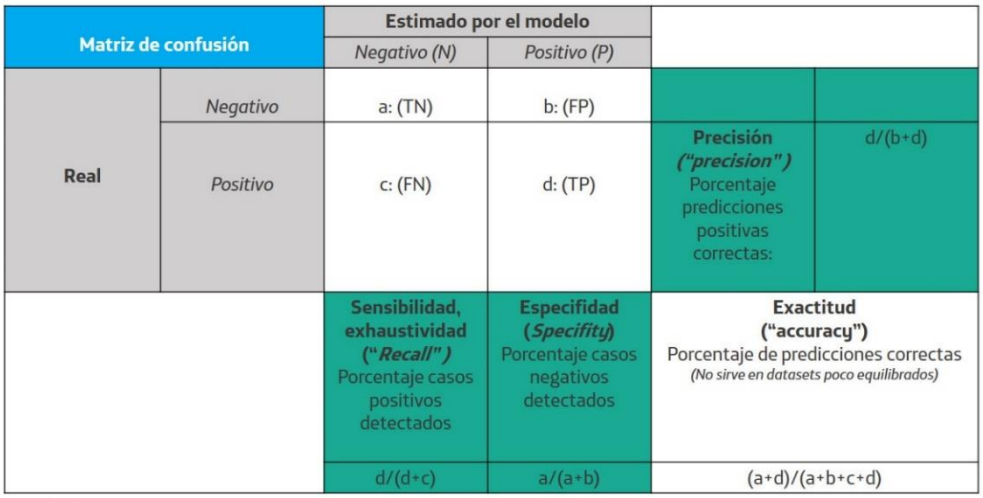
Nota: La Figura muestra la comparación de la popularidad de Python frente a otros lenguajes, en los últimos años ha ganado mucha popularidad ya que es la mejor opción para los desarrolladores de inteligencia artificial. La elaboración y la fuente corresponde a datos tomados de (Saabith, Fareez, & Vinothraj, 2019).

Matriz de confusión

Dentro del campo de la inteligencia artificial y del aprendizaje automático la matriz de confusión se trata de una herramienta que consiste en un arreglo de valores que representa en un eje la cantidad de valores reales para las clases definidas de un problema y en otro eje representa los valores predichos para esas clases obtenidos de un algoritmo de aprendizaje supervisado con el fin de evaluar su desempeño. Por medio de esta se pueden visualizar la cantidad de aciertos y de errores de predicción del algoritmo (Barrios Arce, 2019).

Figura 9

Ejemplo de matriz de confusión con otras métricas de evaluación



Nota: La Figura muestra la matriz de confunsión con otras métricas de evaluación a continuación se detallara el significado de cada uno de los términos. La elaboración y la fuente corresponde a datos tomados de (Recuero de los Santos, 2020).

Exactitud (*Accuracy*): Esta métrica representa el porcentaje de predicciones correctas sobre la cantidad total de observaciones de predicción. Matemáticamente es la suma de a,d dividido para la suma de a,b,c y d (Recuero de los Santos, 2020).

Precisión (*Precision*): Esta métrica indica el porcentaje de predicciones positivas reales obtenidas del total de predicciones positivas obtenidas por el modelo. Es el resultado de la operación d dividido para la suma de d y b.

Sensibilidad, Exhaustividad (*Recall*): Este es el porcentaje de casos positivos que se lograron predecir correctamente por el modelo sobre el total de casos positivos reales.

Especifidad (*Specifity*): Esta métrica representa el porcentaje de casos negativos que se lograron predecir correctamente por el modelo sobre el total de casos negativos reales (Recuero de los Santos, 2020).

Modelos predictivos

Los modelos predictivos realizan el análisis predictivo construyendo un modelo estadístico que utiliza los datos existentes para predecir datos de los cuales no se dispone y su objetivo es analizar la probabilidad de que una unidad de valor parecido en una muestra distinta presente un comportamiento determinado. (Espino Timón, 2017).

Preguntas en sitios Stack Overflow

Las preguntas en Stack Overflow son un tipo de publicación (post) conformadas por varias características y campos generados automáticamente o definidos por el autor de la pregunta. Estas constituyen el principal elemento de interacción entre los usuarios de las comunidades. Entre los campos (Stack Exchange, Inc.) más importantes que puede incluir una pregunta a nivel lógico se encuentran:

- **Question_id.** Código identificador de la pregunta dentro de un sitio determinado
- **Owner.** Detalles del usuario que creó la pregunta.
- **Title.** Título de la pregunta.
- **Body.** Cuerpo o descripción de la pregunta.
- **Tags.** Etiquetas o palabras clave de la pregunta.
- **Link.** URL de la pregunta.
- **Accepted_answer_id.** Código identificador de la mejor respuesta si es que llegara a existir.

- **View_count.** Conteo de vistas de la pregunta.
- **Up_vote_count.** Conteo de votos a favor de la pregunta.
- **Down_vote_count.** Conteo de votos en contra de la pregunta.
- **Delete_vote_count.** Conteo de votos de eliminación de la pregunta.
- **Closed_reason.** Razón del cierre de la pregunta.
- **Closed_details.** Detalles del cierre de la pregunta.

Preprocesamiento

El preprocesamiento de datos implica toda técnica de análisis que resulte en la mejora de la calidad de estos de modo que los métodos de extracción logren recuperar mayor y mejor información. Tiene como objetivo la obtención de un *dataset* final que sea de calidad y utilidad para la fase de extracción de conocimiento. Los procesos que se incluyen en la fase son recopilación de datos, limpieza (valores ausentes y ruido), transformación y reducción (Bello García, 2019).

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

En las secciones a continuación se procederá a describir el proceso realizado en este estudio. Primero se describirá la modalidad que corresponde al presente estudio explicando los motivos por los cuales se eligió esta. El tipo de investigación seleccionado será descrito de igual manera justificando también por qué se eligió este tipo en específico. En el apartado de diseño de la investigación se incluirán todos los detalles acerca del proceso de realización del estudio empezando por la metodología de investigación a fin de llevar a cabo proyecto. También se analizan los resultados obtenidos de la herramienta de recolección de datos y se realizará el respectivo contraste a fin de negar o afirmar la hipótesis planteada.

Modalidad de la investigación

El actual proyecto se realizó utilizando la modalidad de investigación de campo. Esto debido a que fue necesario llevar a cabo la recolección de los datos en el lugar en el que se produce el fenómeno, es decir, los sitios Stack Overflow y Stack Overflow en español. Una investigación de campo podría definirse cómo los pasos que se realizan para recopilar datos de un fenómeno con el fin de estudiarlos tal y como se presentan en la realidad sin ningún tipo de intervención o manipulación externa. Debido a esto las investigaciones de campo se conducen en el lugar donde ocurre el fenómeno y no dentro de un laboratorio o entorno controlado.

Dependiendo del objetivo las investigaciones de campo pueden explorar un fenómeno nuevo o con poca cobertura o también para confirmar si el fenómeno se ajusta a un paradigma establecido. También se pueden conducir investigaciones de campo para describir variables o compararlas. (Significados, 2020).

Tipo de investigación

El tipo de investigación seleccionado para este proyecto fue la cuasi experimental. Como mencionan Portell & Vives, en su libro, las investigaciones de tipo cuasi experimental se aplican en estudios dirigidos al contraste de una hipótesis causal implicando modificaciones en el valor o estado de la variable independiente en el proceso (Portell & Vives, 2019). Además, este tipo de investigación de diferencia de la experimental por el criterio de aleatorización ya que en la investigación cuasi experimental se toman grupos previamente definidos y en vez de tomar las muestras al azar.

Este tipo de investigación se eligió ya que se intentará contrastar la hipótesis formulada en el capítulo 2 realizando un experimento, es decir, el algoritmo desarrollado y el contraste de los resultados con un juicio externo obtenido mediante la herramienta de recolección de datos. Además, resulta conveniente plantear un enfoque no aleatorio ya que existen limitaciones en cuanto al desarrollo del experimento que se verían sobrepasadas si se tomaran muestras al azar.

Diseño metodológico de la investigación

Metodología de investigación

Para el desarrollo del actual proyecto de investigación propuesto se buscará encontrar la respuesta del problema formulado en el capítulo 1 por lo que se buscará cumplir los objetivos especificados para así comprobar la hipótesis planteada.

Análisis de la bibliografía

Para responder el primer objetivo específico se procedió a realizar el análisis de la literatura documentado en el capítulo 2 bajo el apartado de Revisiones sistemáticas. Como resultado se observó que las incorporaciones de palabras son unas de las técnicas más utilizadas para la tarea

de detectar preguntas duplicadas en sitios Q&A de programación. Esto debido a su flexibilidad y variedad de aplicaciones. También se encontraron herramientas cuyas estrategias definieron un punto de partida para el desarrollo del experimento a realizar. Se hizo evidente la necesidad de implementar una función para detectar la similitud de texto a nivel semántico y de utilizar distintos componentes de las preguntas para realizar una comparación más efectiva entre los pares de preguntas. Esta revisión sistemática fue fundamental para definir las estrategias a seguir en el resto del proceso de desarrollo del experimento y del algoritmo en específico. Los artículos seleccionados como resultado de esta revisión sistemática pueden encontrarse en la Tabla 11 en la sección de anexos.

Construcción del dataset

En cumplimiento del segundo objetivo se realizó la construcción del *dataset* a utilizar en las pruebas del algoritmo luego de realizar la extracción y pre procesado de los datos en los sitios Stack Overflow. Considerando el objetivo general del proyecto se pudo establecer la población objetivo de la investigación, es decir, los usuarios que tengan en común los sitios Stack Overflow y Stack Overflow en español. Sin embargo, debido a limitaciones en cuanto a tiempo y recursos previstas en el desarrollo del algoritmo se decidió limitar el alcance a aquellos usuarios que hayan interactuado con la etiqueta Kivy lo que modificó drásticamente el tamaño de la población.

Debido a lo expuesto anteriormente, el *dataset* a construir de igual manera estaría limitado a publicaciones tomadas solamente de la etiqueta Kivy dentro de ambos sitios. Al principio se planteó llevar a cabo la misma estrategia utilizada en los estudios encontrados durante la revisión sistemática. Esta consistía en buscar y descargar un *data dump* (volcado de datos) de los sitios requeridos. Sin embargo, esta estrategia tenía un gran inconveniente para el sitio de Stack Overflow el cual es sitio con el mayor volumen de datos. El inconveniente consistía en el tamaño

de la *data dump* que en sus versiones más recientes y con información actualizada, podían alcanzar pesos de aproximadamente 15 Gb. Estos data dumps contienen información extraída a manera de respaldos de las bases de datos de sus respectivos sitios y aunque resulta ser información bastante completa, muchas veces se incluye información irrelevante para el estudio.

Para sortear estos inconvenientes se realizó una búsqueda adicional de información para encontrar un método de extracción dinámica que permita obtener solo los datos requeridos por el usuario. El método fue encontrado dentro de los mismos foros de la comunidad Stack Exchange y se trataba de una API del sitio. Esta poderosa API tenía la capacidad extraer y hasta cierto punto modificar ciertos atributos en las entidades de las bases de datos de la comunidad Stack Exchange.

Dado que esta API funciona como la mayoría de APIs disponibles en la web, esta se puede utilizar de varias maneras que depende del tipo de aplicación en las que son implementadas. Básicamente todo se resume a realizar una petición a la API con parámetros definidos por el usuario y esperar una respuesta comúnmente obtenida en formato JSON. Para el lenguaje de programación elegido en este experimento se utilizaron dos estrategias con el fin de utilizar la API de Stack Exchange.

La primera estrategia fue la de utilizar un *wrapper* de Python para la API en cuestión. Un *wrapper* se traduce textualmente como envoltura y a nivel de programación se suele referir a la adaptación de un código, función o algoritmo de determinado lenguaje de modo que pueda ser utilizado con la sintaxis de otro lenguaje. El *wrapper* seleccionado fue *StackAPI* el cual es capaz de funcionar con la última versión de la API al momento de redactar este informe, es decir, la versión 2.2. El proceso básico para utilizar esta API consiste en instalar e importar la respectiva librería a el entorno de desarrollo. Luego se puede realizar una consulta con unas pocas líneas de código como se puede observar en la Figura 10. En ella se puede apreciar el proceso de consulta

que consiste en instanciar la clase StackAPI enviando como parámetro el nombre del sitio del cual se requiere extraer los datos en formato *string*. Luego se define la cantidad de elementos a extraer mediante el atributo *page_size* cuyo valor máximo es de 100 elementos. Además, se puede definir la cantidad de páginas a extraer, que no es otra cosa que la cantidad de peticiones que se realizará a la API en caso de que la cantidad de resultados de la consulta sea mayor a 100 elementos. Esto se puede realizar a través del atributo *max_pages* cuyo valor máximo es de 30.

Figura 10

Proceso de Consulta

```
1 SITE = StackAPI('es.stackoverflow')
2 SITE.page_size = 100
3 SITE.max_pages = 1
4 questions = SITE.fetch('questions', tagged='kivy', filter='withbody')
```

Nota: Proceso de consulta que consiste en instanciar la clase StackAPI. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

El uso de este *wrapper* limita la cantidad de elementos a extraer a 3000 debido al poco control que se tiene sobre la propia URL de llamada a la API. Los límites de los atributos *page_size* y *max_pages* responden a varios factores. Con respecto a la máxima cantidad de elementos por página esta se define debido a que dentro de la URL de la API se pueden incluir máximo 100 criterios de búsqueda en ciertos métodos de la API. Por ejemplo, si se desea buscar usuarios por su ID solo se podrán incluir 100 IDs como parámetro de búsqueda. Sobre el límite de cantidad de páginas por extracción esta se debe a que la API implementa un sistema de control de flujos que no permite que un usuario no pueda hacer más de 30 peticiones a la API por segundo. Esto se hace por motivos de seguridad con el fin de evitar ataques DDoS.

Para el método *fetch* se pueden utilizar varios parámetros de los cuales el principal es el *endpoint* (extremo) de la API. Aquí se define el tipo de dato que se va a extraer de la lista incluida

en la documentación de la API (Stack Exchange, Inc., 2021). También se pueden incluir otros parámetros dependiendo del *endpoint* que se haya definido para refinar más la consulta. Existen etiquetas, filtros, IDs que se pueden enviar como parámetros siendo los filtros los más complejos ya que estos deben ser generados en la propia página de la API para definir con más exactitud qué datos deseamos que se incluyan en la respuesta a nuestra petición.

Como segunda estrategia de extracción se siguió utilizando la API de Stack Exchange como base, pero se decidió utilizarla de una manera más convencional a través de peticiones directas a utilizando la librería *request* que precisamente sirve para hacer peticiones HTTP en Python. Este método resulta un poco más complicado en cuanto a implementación, pero como beneficio tiene que se pueden definir más parámetros y tener un mayor control sobre la URL que se utilizará para realizar la petición. Un ejemplo de cómo utilizar este método se observa en la Figura 11.

Figura 11

Método de extracción

```
url = 'https://api.stackexchange.com/2.2/questions?key=' + key
response = requests.get(url)
results = json.loads(response.text)
```

Nota: En esta figura muestra el método de estrategia de extracción. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Como se puede observar en la figura 11 se tiene que definir la URL a la que se va a realizar la solicitud con sus respectivos parámetros y luego utilizar el método *request* para hacer la petición. Una desventaja en cuanto al uso del *wrapper StackAPI* es que con este método no habrá ningún tipo de control de excepciones por lo que se debe conocer muy bien las limitaciones y capacidades de la API, de lo contrario es probable que se obtengas respuestas no deseadas se complique la solución de los errores. Como ventaja se tiene el mayor control sobre la URL de la petición lo que

fue de mucha ayuda para extraer datos que con el *wrapper* no se pueden extraer como los datos de la comunidad Stack Exchange como tal. Debido a que la comunidad Stack Exchange es la organización que agrupa todos los sitios, esta no constaba dentro de la lista de sitios accesibles mediante el *wrapper* por lo que resultaba imposible extraer cualquier tipo de información.

A nivel de presentación y formato de las respuestas por parte de la API, cualquiera de los dos métodos planteados retorna un objeto de tipo diccionario a través del cual podremos acceder a los elementos de la consulta y a otros atributos propios de la respuesta como banderas indicando si existen más resultados que no se incluyeron en esa página o la cantidad de peticiones restantes asignadas a nuestra IP o proyecto por parte de la API.

Utilizando la API se extrajeron primero todas las preguntas realizadas bajo la etiqueta Kivy en Stack Overflow en español. Esta etiqueta se eligió debido a su manejable volumen de datos en el sitio en español siendo aproximadamente 80 la cantidad de preguntas obtenidas. Se decidió realizar el experimento con una cantidad pequeña de preguntas debido a que Stack Exchange no realiza un registro de preguntas duplicadas entre bases de diferentes sitios. Dado que el objetivo de nuestro experimento era realizar la comparación entre preguntas de sitios con distintos idiomas era necesario obtener preguntas de manera independiente para ambos sitios y realizar la comparación entre ellas de manera manual a fin de entrenar el modelo utilizado para la detección de duplicidad.

Aunque dentro de la respuesta de la API se incluían varios datos que para efectos del experimento resultaban irrelevantes, se extrajeron cuatro campos que se consideraron importantes para el desarrollo del algoritmo, es decir, el ID de la pregunta en su respectivo sitio, el título de la pregunta, el cuerpo o descripción de la pregunta y las etiquetas de la pregunta. Este proceso se realizó tanto para el sitio en español como para el sitio en inglés.

Una corrección adicional que se tuvo que hacer a los datos extraídos fue el tratamiento de los caracteres latinos en el título de las preguntas en español. En estos títulos los caracteres latinos, como aquellos que tienen tilde, diéresis o virgulillas, se extraían de la API como caracteres en código decimal por lo que fue necesario utilizar expresiones regulares para reemplazar estos valores por el carácter correspondiente a través de un proceso que se describe en la Figura 12.

Figura 12

Algoritmo de conversión de decimales a caracteres

```
def dec2char(txt):
  chars = list(filter(None,re.findall("(?!&#)\d*(?=;)", txt)))
  for char in chars:
    matches = list(re.finditer("&#\d*;", txt))
    begin = matches[0].start(0)
    end = matches[0].end(0)
    txt = txt[:begin] + chr(int(char)) + txt[end:]
  return txt
```

Nota: La Figura muestra el proceso de conversión de decimales a caracteres. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Preprocesamiento

Entre las acciones realizadas en la fase de preprocesamiento se encuentran la conversión del texto a minúsculas, la eliminación de etiquetas HTML, la eliminación de signos de puntuación, la eliminación de palabras vacías y en el caso de las preguntas en español se tradujeron sus componentes al idioma inglés.

Uno de los puntos más importantes en la etapa de preprocesamiento fue decidir qué datos debían mantenerse como entrada para el algoritmo. Esto debido a que en el cuerpo de la pregunta se incluían muchos datos que podían causar ruido en el análisis. Por ejemplo, las preguntas de algunos usuarios contenían enlaces a repositorios de imágenes donde ellos subían una captura del problema que estaban teniendo o de lo que querían llegar a desarrollar mediante su pregunta. Ya

que este estudio está basado en la utilización de herramientas de aprendizaje automático y procesamiento del lenguaje natural, se descartaba el implementar o desarrollar algún sistema de reconocimiento óptico de caracteres o visión por computadora para lograr extraer información relevante de las imágenes compartidas mediante estos enlaces. Es por esto por lo que todo tipo de enlace en la pregunta fue eliminado junto con su respectiva etiqueta HTML.

Otro tipo de etiqueta HTML cuya eliminación fue analizada es la etiqueta que se utiliza para escribir fragmentos de código fuente <*code*>. Inicialmente esta etiqueta fue eliminada junto con su contenido debido a que ciertos usuarios incluían porciones de código y registros de errores con demasiada información irrelevante al punto de llegar confundir incluso a los miembros de la comunidad que intentaban responder sus preguntas. Luego se determinó que el código fuente incluido en las preguntas muchas veces era la clave para entender a qué se refería el usuario con su pregunta y por ende entender el sentido de esta. Esto es útil tanto para quienes buscan responder la pregunta como para quienes buscan determinar si la pregunta es duplicada de otra. Debido a esto en vez de eliminar por completo el contenido de la etiqueta, se intentó realizar una limpieza de los fragmentos de código eliminando términos duplicados y caracteres especiales.

Luego de realizar el tratamiento de las etiquetas HTML y su contenido además de los signos de puntuación se procedió a realizar la traducción del texto en el caso de las preguntas en español. Para esta tarea se utilizó nuevamente un *wrapper* que permitía utilizar la API de Google Traductor dentro de un entorno de desarrollo en Python. Actualmente existen dos traductores automáticos reconocidos por sus excelentes resultados. Estos son Google Traductor y DeepL. El motivo de la selección de Google Traductor para este estudio fue la cantidad de información disponible sobre su utilización y el soporte que recibe por parte de la comunidad lo cual fue un

factor determinante. Aun así, se determinó qué DeepL presenta resultados muy buenos y sería una opción viable para realizar la misma tarea.

Un problema que se encontró con Google Traductor se debía a un bug de la API qué hacía que devolviera un mensaje de error cuando encontrara determinados caracteres especiales. Realizando una investigación de las causas y soluciones de este problema se encontró que la comunidad de desarrolladores lo había solucionado y había compartido la librería para su uso público. Aunque los nombres de las clases, funciones y parámetros eran diferentes al de la librería original no resultaban confusos.

Otra limitación que se encontró al utilizar Google Traductor y que compartía incluso con DeepL era la cantidad de caracteres máxima que se podía incluir en una cadena de texto a traducir. Estos servicios de traducción permitían incluir cadenas de máximo 5000 caracteres. A simple vista puede no parecer un problema, pero existían preguntas qué incluyendo los bloques de código superaban este límite. Con unas cuantas líneas de código se logró dividir las cadenas que superaban los 5000 caracteres para traducir el texto en partes de modo que se superara esta limitación.

Tabla 8

Fase de Preprocesamiento

Proceso	**Titulo**	**Cuerpo**
Contenido original	Excepción java.lang NoClassDefFoundError. al tratar de usar pyjnius	" Estoy desarrollando una aplicación android con python e intento utilizar pyjnius para implementar los módulos de JavaCuando le doy un import a jnius no tengo problemas El detalle viene cuando intento importar cualquier parte de la api de android, pues obtengo el

		siguiente error He leído varias publicaciones con un error similar, sin embargo no encuentro la solución. Muchas gracias. "
Eliminación de signos y etiquetas HTML	Excepción java.lang.NoClassDefFoundError. al tratar de usar pyjnius	Estoy desarrollando una aplicación android con python e intento utilizar pyjnius para implementar los módulos de Java Cuando le doy un import a jnius no tengo problemas El detalle viene cuando intento importar cualquier parte de la api de android, pues obtengo el siguiente error He leído varias publicaciones con un error similar sin embargo no encuentro la solución Muchas gracias.
Traducción	Java.lang.NoClassDefFoundError Exception. when trying to use pyjnius	I am developing an android application with python and trying to use pyjnius to implement the java modules When I give an import to jnius I have no problems The detail comes when I try to import any part of the android api, because I get the following error I have read several posts with a similar error, however I cannot find the solution. Thanks a lot.

Nota: La tabla muestra la fase de preprocesamiento hasta el proceso de traducción . La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 13 se puede observar el proceso básico para la instalación, importación y utilización de la librería para traducir una cadena de texto de un determinado idioma a otro.

Figura 13

Proceso básico para la instalación, importación y utilización de la librería

```
!pip install google_trans_new
from google_trans_new import google_translator
translator = google_translator()
text_translate = translator.translate(text, lang_tgt='en', lang_src='es')
```

Nota: La Figura muestra el proceso básico para la instalación, importación y utilización de la librería para traducir la cadena de texto de un idioma a otro. La elaboración es propia y la fuente corresponde a datos propios de la

investigación.

Como se puede observar el proceso consiste en la instalación mediante el gestor PIP, la importación de la clase para realizar la traducción y por último la utilización del método de traducción de esta clase. Como parámetros de este método tenemos la cadena de texto a traducir en formato string y el idioma en que está escrito esta cadena cuyo nombre es *lang_src* (idioma fuente). También se debe especificar el idioma al que se bebe traducir esta cadena de texto mediante el parámetro *lang_tgt* (idioma objetivo).

Luego de realizar todos estos pasos se incluyeron parámetros para determinar ciertos aspectos del preprocesamiento como la inclusión de términos duplicados o la conversión de las cadenas de texto a una lista contenedora de los términos de cada elemento. Esto se hizo debido a que los experimentos realizados posteriormente requerían entradas dispuestas en distintos formatos. En la Tabla 8 se puede observar el resultado de la fase de preprocesamiento hasta el proceso de traducción.

Etiquetado

Para realizar la detección de preguntas duplicadas mediante técnicas de aprendizaje automático era necesario tener un dataset en el cual se incluyeran las preguntas en español con sus posibles duplicadas en inglés a fin de utilizarlo para entrenar los modelos. Este proceso de comparación se tuvo que realizar manualmente debido a las limitaciones de la plataforma Stack Exchange en general.

Como primer paso para el proceso de etiquetado debían buscarse coincidencias de búsqueda de las preguntas en español. Para esto se decidió utilizar el motor de búsquedas personalizadas de Google (Custom Search Engine CSE). Es común que Google ofrezca sugerencias bastante acertadas al momento de realizar una búsqueda por lo que se utilizaron las

preguntas del sitio en español ya traducidas al idioma inglés para realizar búsquedas mediante el motor personalizado de Google el cual se configuró con parámetros como la página de búsqueda que en este caso sería la de Stack Overflow en inglés. Este motor de búsquedas se utiliza de manera bastante parecida a las APIs ya mencionadas debido a que en principio es una API perteneciente al grupo de APIs proporcionadas al público por Google. Una vez se realiza la llamada a la API con su respectiva consulta, dentro de los resultados obtenidos en la respuesta se utilizaron expresiones regulares para extraer los IDs de las preguntas que aparecían como parte de los resultados.

Para cada pregunta en español se decidió tomar las cinco primeras coincidencias proporcionadas por el motor de búsquedas personalizadas de Google con lo que se obtuvo un aproximado de 326 pares de preguntas para realizar el entrenamiento del algoritmo.

Una vez obtenidos los IDs de las coincidencias se procedió a utilizar nuevamente la API de Stack Exchange para extraer la información de estas preguntas como su título, cuerpo y etiquetas. De este modo pudo generar una versión más completa del *dataset* de etiquetado en la cual se incluían los datos completos de cada par de preguntas incluyendo su URL que serviría para comparar manualmente las preguntas desde sus respectivos sitios y determinar su duplicidad.

Luego de haber realizado las comparaciones sea añadió una columna adicional al *dataset* para indicar si el par de preguntas de determinada fila eran consideradas como preguntas duplicadas o no.

Técnicas de aprendizaje automático

Con el fin de cumplir con el tercer objetivo se emplearon diversas técnicas de aprendizaje automático a fin de encontrar preguntas duplicadas entre los elementos del *dataset* luego de haber realizado la construcción, preprocesamiento y etiquetado de los datos. Para los experimentos se realizaron pruebas con 3 estrategias diferentes utilizando técnicas de aprendizaje

autónomo de diversas maneras de modo que se lograra determinar cuál tiene un mejor rendimiento frente a las otras.

Como punto a considerar se tiene la utilización de un modelo pre entrenado de incorporaciones de palabras tomado del estudio conducido por (Efstathiou, Chatzilenas, & Spinellis, 2018). Este modelo fue realizado debido a la falta de modelos pre entrenados cuyo dominio específico sea la ingeniería de software. Se decidió utilizar este modelo pre entrenado debido al costo en cuanto a tiempo y recursos que implica realizar un entrenamiento con un volumen de datos igual al utilizado por los autores. Este modelo se entrenó utilizando un data dump de Stack Overflow que abarcaba información desde su lanzamiento en el año 2008 hasta el año 2017 y el tiempo que tomó realizar el entrenamiento fue de 11 horas. El modelo fue entrenado durante 3 épocas y como resultado se obtuvieron vectores de 200 dimensiones para representar cada palabra en el vocabulario. Los resultados del estudio muestran que el modelo puede diferenciar con bastante precisión las palabras poli semánticas y de igual manera encontrar relaciones bastante cercanas entre las palabras que componen el vocabulario.

Una de las desventajas de utilizar este modelo es que los autores lo hicieron público como un objeto de tipo *keyedVector* Propio de la librería *Gensim* con la cual se entrenó el modelo. La limitación de este tipo de objetos es que no permiten un entrenamiento posterior haciendo que no se pueda actualizar el vocabulario ni realizar ningún cambio o mejora en el modelo. Por otra parte, tiene varias ventajas de entre las cuales resalta su eficiencia en cuanto al consumo de almacenamiento llegando a hacer que el modelo tenga un peso mucho menor al de un modelo almacenado en el formato predeterminado. Por medio de este modelo se utilizarán varios métodos y se obtendrán datos muy importantes al momento de ser aplicados en los experimentos.

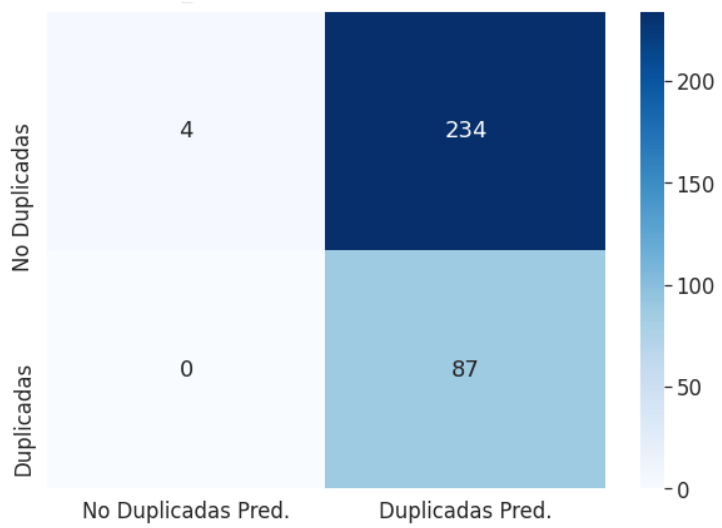
Word Mover Distance

La primera estrategia pensada para detectar la duplicidad entre preguntas fue la de utilizar la técnica de *Word Mover Distance* para calcular la similitud entre grupos de palabras. Para esto se realizó la conversión del título y el cuerpo de cada pregunta a listas de palabras de modo que se pudieran unir el título y el cuerpo en una sola lista que representará todas las palabras presentes en la pregunta. Una vez obtenidas las listas de palabras se procedió a ejecutar el método *wmdistance* incluido en el objeto contenedor del modelo. Este método recibe como parámetro dos listas de palabras y retorna el valor de la distancia entre estos vectores (listas). Este valor puede tomarse como relativo ya que por más que sus valores mínimos y máximos sean distantes siempre representarán la similitud con respecto al vocabulario del modelo entrenado.

El experimento realizado con esta estrategia consistió en calcular la similitud entre cada uno de los pares de preguntas del *dataset* etiquetado. Una vez se calcule el valor se intentará encontrar el umbral óptimo para el algoritmo. El umbral se refiere al valor a partir del cual un par de preguntas se podrían considerar como duplicada considerando su valor de similitud con respecto a una métrica específica. En este caso se intentó encontrar el umbral óptimo de modo que la métrica de exhaustividad sea la mejor. El valor obtenido como umbral óptimo de 1.162 produciendo un puntaje de exhaustividad de 1.0. Esto quiere decir que el 100% de las preguntas duplicadas lograron ser detectadas mediante la similitud entre su texto. Sin embargo, al analizar la matriz de confusión para el umbral óptimo para exhaustividad presentada en la Figura 14 se puede observar que la cantidad de falsos positivos es bastante elevada debido a que muchas preguntas parecen tener una similitud de texto bastante parecida sin llegar a ser un par duplicado necesariamente.

Figura 14

Matriz de confusión de similitud con Word Mover Distance con umbral óptimo para exhaustividad

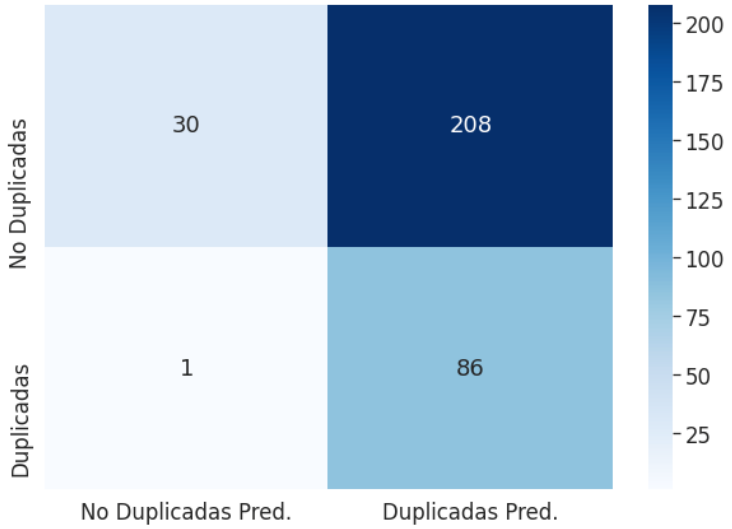


Nota: La figura muestra la matriz de consunsión de similitud con *Word Mover Distance* con el umbral óptimo para exhaustividad, se puede visualizar la cantidad de falsos positivos. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Debido a los problemas presentados al utilizar la exhaustividad como métrica de referencia se decidió cambiar esta métrica por el puntaje F1 el cual se calcula a partir de los valores de precisión y exhaustividad. Al realizar esto se obtuvieron mejores resultados obteniendo un puntaje de exhaustividad de 0.99 y disminuyendo la cantidad de falsos positivos predichos lo cual se ve en la matriz de confusión de umbral óptimo para el puntaje F1 presentada en la Figura 15.

Figura 15

Matriz de confusión de similitud con Word Mover Distance con umbral óptimo para el puntaje F1



Nota: La Figura muestra la matriz de confusión de similitud con *Word Mover Distance* para el umbral óptimo para el puntaje F1 el que se considera como el balance entre la exhaustividad y la precisión. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Caracterización con Word Mover Distance

En la segunda estrategia formulada se determinó qué las comparaciones que hacen los humanos entre preguntas no consisten necesariamente en revisar si sus palabras se parecen. Más bien al momento de realizar una comparación entre preguntas es común que se revisen la similitud de los componentes por separados, es decir, los títulos con títulos, cuerpos con cuerpos y etiquetas con etiquetas. Se pensó que estos valores de similitud podrían constituir las características para una observación que se pudiera utilizar como entrada de un algoritmo más convencional y ampliamente utilizado, es decir, la regresión logística. Incluso estos valores pueden ser tomados como entradas para modelos de redes neuronales cuyo rendimiento, bajo determinadas condiciones, suelen tener mejores resultados que la regresión logística.

En este caso la observación equivale al par de preguntas comparadas con sus respectivos valores de similitud de modo que el modelo intente hallar los pesos adecuados para clasificar cada observación como un par de preguntas duplicadas o no duplicadas haciendo de éste un problema de clasificación binaria. El valor de similitud que se utiliza en cada componente sigue siendo el valor de distancia obtenido mediante el método *wmdistance* utilizado en la estrategia anterior pero aplicado a cada par de componentes como se describió anteriormente.

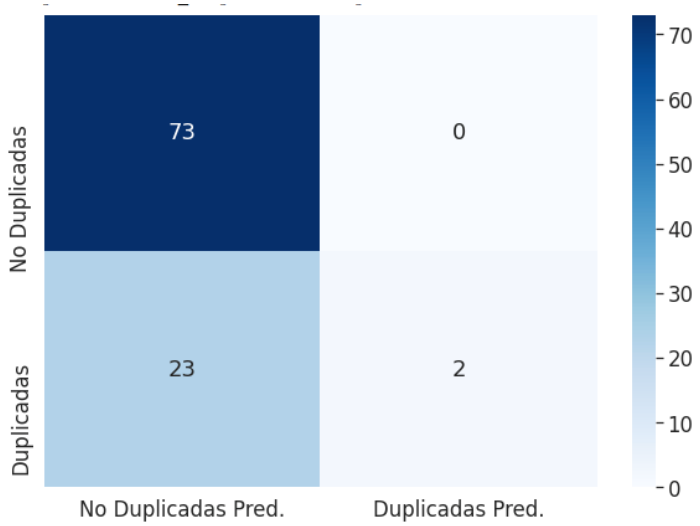
Para entrenar los modelos se procedió a realizar la división del *dataset* de entrenamiento fue de un 70% dejando así un 30% disponible para realizar las respectivas pruebas. Como métricas para evaluar los modelos se utilizaron los métodos provistos por la clase *metrics* de la librería *sklearn.*

Regresión logística

Utilizando el modelo de regresión logística se obtuvo un puntaje de exactitud de 0.76531. En principio este valor parece indicar que el modelo tiene un rendimiento bastante bueno. Sin embargo, al revisar la matriz de confusión presentada en la Figura 16 se hizo evidente que el modelo tiene en realidad un rendimiento pésimo al momento de detectar preguntas duplicadas llegando a obtener un puntaje de exhaustividad de 0,08. Esto significa que solo pudo reconocer 2 preguntas duplicadas de las 25 incluidas en el *dataset* de prueba.

Figura 16

Matriz de Confusión de duplicida con Regresión Logística

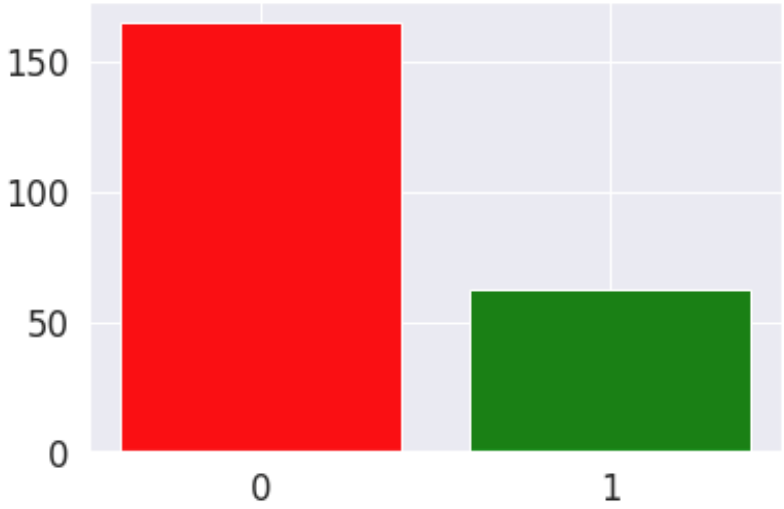


Nota: En la matriz se muestran los valores reales y predichos obtenidos al utilizar el modelo de regresión logística. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Al intentar determinar las causas de este bajo rendimiento se encontró que en efecto el *dataset* estaba bastante desbalanceado. Esto quiere decir que la cantidad de pares de preguntas duplicadas era mucho mayor a la cantidad de pares duplicados como se puede observar en la Figura 17. Debido a esto era más probable que el modelo detectara un par de preguntas como no duplicada.

Figura 17

Cantidad de preguntas duplicadas en el dataset



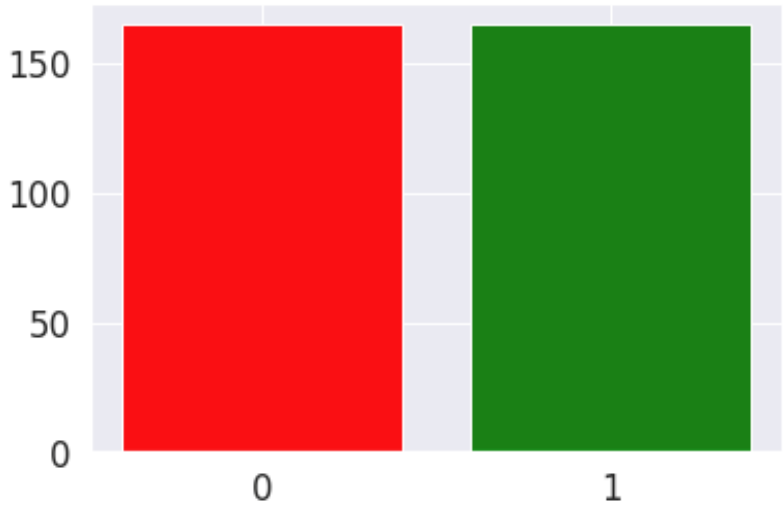
Nota: La Figura muestra la cantidad de preguntas duplicadas en el *dataset*. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Para solucionar este problema de desbalanceo existen varias opciones. Por un lado, se puede realizar un *subsampling*. Esto quiere decir que se reducirá el número de muestras negativas en el *dataset* de modo que su cantidad coincida con la cantidad de muestras positivas. Este método no se consideró apropiado ya que el *dataset* de por sí contiene una cantidad muy pequeña de preguntas por lo que al eliminar pares influirá negativamente en el aprendizaje del modelo. Por otro lado, se puede realizar un *supersampling*, es decir, complementar el *dataset* con una mayor cantidad de preguntas duplicadas. Sin embargo, esto no resultaba viable ya que implicaba tener que considerar un nuevo grupo de preguntas que debe pasar nuevamente por el proceso de extracción preprocesamiento y etiquetado haciendo que se desfasaran los tiempos para el desarrollo del proyecto. Por lo tanto, se buscaron otras alternativas a fin de solucionar el problema de desbalanceo del *dataset*.

Como resultado de esta búsqueda se determinó utilizar el método *SMOTE (Synthetic Minority Oversampling Technique).* Este método es una forma de *supersampling* que incrementa el número de muestras seleccionando ejemplos que se encuentran cerca en el espacio de características. Luego traza una línea entre estos ejemplos en su espacio de características de modo que se obtiene una nueva muestra trazada a lo largo de esta línea de características. Luego de aplicar el *SMOTE* el *dataset* quedó balanceado como se muestra en la Figura 18. A pesar de los beneficios de utilizar este método, no es recomendable que se utilice en los *dataset* de prueba por lo que se aplicó solamente al dataset de entrenamiento obtenido luego de la división.

Figura 18

Cantidad de preguntas duplicadas en el dataset balanceado con SMOTE



Nota: La Figura muestra la cantidad de preguntas duplicadas en el *dataset* balanceado a tráves del método *SMOTE*. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

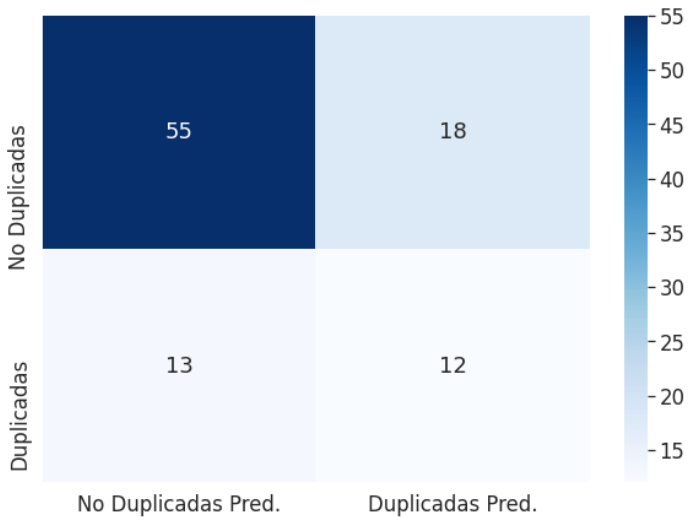
Al utilizar el *dataset* balanceado a través del *SMOTE* se obtuvieron resultados significativamente mejores a los anteriores. El problema del desbalanceo resultó ser determinante en el entrenamiento y por lo tanto en el rendimiento del algoritmo. Esto también hizo evidente que la cantidad de muestras para el entrenamiento debe incrementarse para obtener mejores resultados

como en cualquier modelo. Sin embargo, el modelo de regresión logística se eligió debido a que es uno de los que menos datos consumen para producir buenos resultados.

Los resultados de la regresión logística utilizando el *dataset* balanceado se pueden observar en la Figura 19. Se puede observar cómo se pudieron detectar 12 de los 25 pares de preguntas duplicadas obteniendo así un puntaje de exhaustividad de 0,48 el cual representa una mejora sustancial al 0,08 obtenido usando el *dataset* desbalanceado. Resulta curioso que el puntaje de exactitud de este modelo fue de 0,68 llegando a ser incluso menor al del modelo con el *dataset* desbalanceado. Esto se debe a que no se recomienda utilizar el puntaje de exactitud para medir el desempeño de un modelo entrenado con un *dataset* desbalanceado debido a la alta probabilidad de predecir las clases sesgado por la cantidad de la clase con mayor cantidad de muestra.

Figura 19

Matriz de Confusión de similitud con Regresión Logística con el dataset balanceado



Nota: Aquí se mustran los valores de predicción obtenidos con regresión logística y el *dataset* balanceado. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Red Neuronal

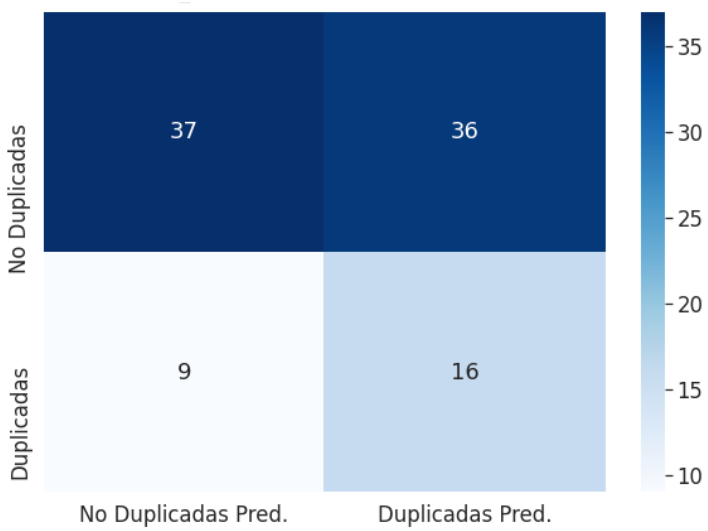
Además de la regresión logística se hicieron experimentos utilizando la caracterización con *wmdistance* como entrada de una red neuronal. La red neuronal se creó utilizando la clase *sequential* provista por la librería *Keras* perteneciente a Google.

En cuanto a la arquitectura de la red neuronal se configuraron 3 capas. La primera capa es la capa visible de entrada que recibirá las características de cada observación para la cual se debe definir el parámetro *input_dim* para especificar la cantidad de características de cada observación, en otras palabras, la dimensión del vector de características. Esta capa se definió como una capa densa de 12 nodos cuya función de activación elegida fue *ReLU*. Cuando una capa se define como densa quiere decir que todos sus nodos están conectados con todos los nodos de la siguiente capa. La segunda capa es oculta y está conformada por 8 nodos y al igual que la primera capa esta es densa y también utiliza *ReLU* para la activación de sus nodos. La última capa es de salida y consta de un solo nodo debido a que la predicción de un par de preguntas solo puede tener un valor que corresponda ya sea a la clase “no duplicada” o “duplicada”. Debido a esto la función de activación de esta capa fue Sigmoide debido al buen rendimiento que tiene esta función en problemas de clasificación binaria. El modelo se entrenó por 100 épocas con un *batch_size* de 2.

Los resultados obtenidos por el modelo superaron a los obtenidos por el modelo de regresión logística por un margen significativo presentando un puntaje de exhaustividad de 0,64 en comparación al 0,48 del modelo anterior. Cabe recalcar que estos resultados se obtuvieron entrenando el modelo con el *dataset* balanceado ya que los resultados del *dataset* desbalanceado fueron pésimos al igual sucedió que con el modelo de regresión logística. En la matriz presentada en la Figura 20 se puede observar cómo se redujo la cantidad de predicciones erróneas en el *dataset* de prueba.

Figura 20

Matriz de Confusión de similitud con Red Neuronal



Nota: La Figura muestra la matriz de confusión de similitud utilizando la arquitectura de la red neuronal. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

A pesar de ser una red neuronal sencilla cuyo proceso de ajuste no se optimizó en lo absoluto se puede determinar esta estrategia de detección como la mejor de entre las demás analizada mediante estos experimentos. Considerando la poca optimización y la poca cantidad disponible para el entrenamiento este modelo proporcionó resultados significativamente superiores a los de los modelos y métodos anteriores.

Población y muestra

Para cumplir con el cuarto objetivo se realizó la evaluación del algoritmo a través de un juicio externo obtenido mediante encuestas a la población definida para este estudio. La población que se tomó en el presente trabajo de investigación fue extraída del total de los usuarios del sitio Stack Overflow en español e inglés que participan en preguntas con la etiqueta Kivy, en donde el criterio de selección que se utilizo fue que los usuarios formen parte de las comunidades Stack

Overflow en español y Stack Overflow de manera simultánea obteniendo así un total de 79 usuarios para el estudio. Un criterio de exclusión que se utilizó fue que los usuarios que no contaban con una manera de ser contactados en su perfil no serían incluidos ya que no existía ninguna manera formal de comunicarse con estos e invitarlos a participar en la encuesta. Por lo expuesto anteriormente fueron seleccionados 17 usuarios que tenían en su perfil disponible una dirección de correo. Debido a que los criterios y limitaciones de la selección redujeron tanto la cantidad de usuarios seleccionados se determinó que la muestra sería igual a la población ya qué es lo recomendado cuando se trabaja con poblaciones muy pequeñas.

Procesamiento y análisis

El procesamiento y análisis que se realizó fue el mecanismo de recolección de datos mediante una encuesta online que nos permitió obtener resultados al instante y por medio de la cual se pudo recabar datos estadísticos que permitieron validar el funcionamiento del algoritmo y detectar las preguntas duplicadas entre los sitios Stack Overflow en español e inglés.

Técnicas de recolección de datos.

Encuesta.

La técnica elegida como parte del diseño de investigación fue la encuesta debido a que resulta útil para recopilar datos sobre las opiniones de los encuestados mediante el uso de preguntas formuladas en base a una metodología definida. Resulta una forma práctica para obtener datos representativos acerca de una hipótesis o problema con el fin de evaluar sus resultados y definir conclusiones en concordancia con la información recopilada (Cabezas Mejía, Andrade Naranjo, & Torres Santamaría, 2018).

La encuesta se realizó en la plataforma online Microsoft Forms en la cual se creó dos versiones de la encuesta una en español y otra en inglés debido a que los usuarios encuestados pertenecen a las comunidades de Stack Overflow en español e inglés. De esta forma se logró un mayor alcance ya que con esta herramienta los usuarios contaban con acceso desde cualquier lugar. La encuesta está compuesta por un total de 25 preguntas y esta se separó en 3 secciones con el fin de separar los datos específicos según los factores que componen las preguntas presentadas en la Tabla 9 en donde se muestra cada una de las secciones, las preguntas que conforman cada sección y el propósito que tiene cada sección.

Tabla 9

Estructura de la encuesta y sus secciones

N.	Sección	Preguntas	Propósito
1	Consentimiento	1. Consentimiento Electrónico	Informar al usuario encuestado los detalles del proyecto y obtener el consentimiento del usuario para trabajar sobre los datos que nos proporcione.
2	Antecedentes	2. ¿Cuál es su ocupación? 3. ¿Qué tiempo lleva participando en las comunidades Stack Overflow? 4. ¿Durante qué tipo de proyectos con Kivy ha utilizado Stack Overflow? 5. ¿Qué tan frecuentemente realiza las siguientes actividades en los sitios Stack Overflow?	Conocer el perfil del usuario encuestado, el tiempo que lleva formando parte de las comunidades Stack Overflow en español e inglés y su experiencia.
3	Evaluación del Modelo	6. Duplicidad de Preguntas: 1 Fallo en Buildozer **vs.** Android App created with Kivy (Buildozer) crashes on pone, but why?	Evaluar si existe o no duplicidad de preguntas entre los sitios de Stack Overflow en español e inglés para la evaluación del modelo. Al final de

7. Duplicidad de Preguntas: 2 Utilizar fichero .kv **vs.** Using external .kv file vs doing things internally? 8. Duplicidad de Preguntas: 3 Error al usar la Camara en Android con kivy **vs.** Python kivy with plyer app crashes on Android (camera) 9. Duplicidad de Preguntas: 4 llamar función con un botón usando Python-kivy **vs.** Calling function in a different class through Kivy button 10. Duplicidad de Preguntas: 5 Poner botón para ejecutar la camera **vs.** Python Kivy: ¿how to call a function on button click? 11. Duplicidad de Preguntas: 6 TypeError: 'NoneType' object is not callable **vs**. Kivy popup Filechooser pass variable (selection) 12. Duplicidad de Preguntas: 7 Congelamiento de interface grafica kivy **vs.** Kivy app freezes when using threading 13. Duplicidad de Preguntas: 8 Enviar valor entre ventanas en Kivy **vs.** Pass values between classes (screens) in kivy 14. Duplicidad de Preguntas: 9 Error de Kivy en Visual Studio Code **vs.** Problem with kv file in visual studio code 15. Duplicidad de Preguntas: 10 Como gestionar hilos en kivy **vs.** How to run two process at once using kivy	cada pregunta de duplicidad se busca conocer las opiniones de las mismas, estas son opcionales.

Nota: Esta tabla plantea las 15 preguntas realizadas en la encuesta y sus 3 secciones excluyendo las preguntas opcionales y/o casillas de comentarios. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Beneficiarios directos e indirectos del proyecto

Los beneficiarios del presente proyecto se detallarán a continuación:

Beneficiarios directos: Los beneficiarios directos son los administradores de las comunidades de SO y SOES, ya que con el resultado del estudio se podrá determinar si las técnicas de aprendizaje automático permiten detectar preguntas duplicadas entre los sitios SO y SOES. Si bien es cierto los dos sitios son parte de una red de comunidades, la integración de estos a nivel de información es muy limitada. Actualmente los sitios no manejan ningún protocolo que permita a los usuarios a marcar preguntas duplicadas entre sitios con diferentes idiomas. En caso de comprobarse que la hipótesis, los administradores pueden considerar que al existir una herramienta que ejecute este proceso de manera automática no tendrán que preocuparse por los costos de mantener un sitio multilingüe ya que los sitios seguirían siendo independientes, pero con una pequeña capa adicional de integración.

Beneficiarios indirectos: Los beneficiarios indirectos son los desarrolladores profesionales, estudiantes sean o no usuarios registrados en las comunidades de SO y SOES, incluidos quienes realizan consultas de las preguntas ya publicadas en los sitios, sin necesidad de formular una pregunta nueva, ni de crear una cuenta. Estos se podrán beneficiar de una búsqueda de información más amplia y por lo tanto de una probabilidad mayor de encontrar respuestas satisfactorias en un tiempo determinado. Aunque la calidad de este beneficio depende de las decisiones tomadas por los desarrolladores y administradores de Stack Overflow también podría hacerse efectivo por parte de los mismos beneficiarios ya que podrían adaptar las estrategias propuestas en el algoritmo para realizar una implementación externa del mismo.

Criterios de validación del estudio

Análisis de datos

Para la validación de la propuesta del presente proyecto de investigación realizó el contraste de la hipótesis calculando el estadístico X^2 mediante la utilización de tablas de contingencia para realizar el respectivo análisis.

La hipótesis se evaluará de acuerdo con el siguiente criterio de expertos:

- Si el *p-value* es menor o igual al valor de significancia entonces se acepta H_1y se rechaza H_0. Cuando se cumple esta condición se dice que el *p-value* es significativo para el valor de significancia.
- Si el *p-value* es mayor al valor de significancia entonces se acepta H_0 y se rechaza H_1. Cuando se cumple esta condición se dice que el *p-value* no es significativo para el valor de significancia.

Donde:

- p: es el valor de probabilidad
- α: es el nivel de significancia, con un valor de 0.05
- $H_{0:}$ El modelo realizado con técnicas de aprendizaje autónomo ayuda a detectar preguntas duplicadas en SO y SOES. Hipótesis nula.
- $H_{1:}$ El modelo realizado con técnicas de aprendizaje autónomo no ayuda a detectar preguntas duplicadas en SO y SOES. Hipótesis alternativa.

Para realizar el cálculo de Chi-cuadrado de todas las hipótesis se escogió las siguientes secciones: la sección 2 Antecedentes (la pregunta ¿Cuál es su ocupación?) y la sección 3 Evaluación del Modelo (las 10 preguntas del modelo).

Los cálculos de Chi-cuadrado fueron realizados en la calculadora del sitio web "*Social Science Statistics"*.

Contraste 1

Figura 21

Tabla de contingencia. Pregunta #6 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	4 (4.67) [0.1]	3 (2.33) [0.19]	7
Ingenierio / Desarrollador de Software	4 (3.33) [0.13]	1 (1.67) [0.27]	5
Marginal Column Totals	8	4	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.6857. El valor *p-value* es 0.407626. No significativo en p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 21 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.407626 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 2

Figura 22

Tabla de contingencia. Pregunta #8 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	6 (5.83) [0]	1 (1.17) [0.02]	7
Ingeniero / Desarrollador de Software	4 (4.17) [0.01]	1 (0.83) [0.03]	5
Marginal Column Totals	10	2	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.0686. El valor *p-value* es 0.793428. No significativo en p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 22 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.793428 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 3

Figura 23

Tabla de contingencia. Pregunta #10 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	3 (2.33) [0.19]	4 (4.67) [0.1]	7
Ingeniero / Desarrollador de Software	1 (1.67) [0.27]	4 (3.33) [0.13]	5
Marginal Column Totals	4	8	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.6857. El valor *p-value* es 0.407626. No significativo en p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 23 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.407626 > 0,05) que de acuerdo con los criterios de significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 4

Figura 24

Tabla de contingencia. Pregunta #12 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	3 (2.92) [0]	4 (4.08) [0]	7
Ingenierio / Desarrollador de Software	2 (2.08) [0]	3 (2.92) [0]	5
Marginal Column Totals	5	7	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.0098. El valor *p-value* es 0.921159. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 24 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.921159 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 5

Figura 25

Tabla de contingencia. Pregunta #14 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	4 (3.5) [0.07]	3 (3.5) [0.07]	7
Ingenierio / Desarrollador de Software	2 (2.5) [0.1]	3 (2.5) [0.1]	5
Marginal Column Totals	6	6	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.3429. El valor *p-value* es 0.558185. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 25 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.558185 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 6

Figura 26

Tabla de contingencia. Pregunta #16 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	3 (2.92) [0]	4 (4.08) [0]	7
Ingenierio / Desarrollador de Software	2 (2.08) [0]	3 (2.92) [0]	5
Marginal Column Totals	5	7	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.0098. El valor *p-value* es 0.921159. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 26 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.9221159> 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 7

Figura 27

Tabla de contingencia. Pregunta #18 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	2 (1.75) [0.04]	5 (5.25) [0.01]	7
Ingenierio / Desarrollador de Software	1 (1.25) [0.05]	4 (3.75) [0.02]	5
Marginal Column Totals	3	9	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.1143. El valor *p-value* es 0.735317. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 27 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.735317 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 8

Figura 28

Tabla de contingencia. Pregunta #20 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	2 (1.75) [0.04]	5 (5.25) [0.01]	7
Ingeniero / Desarrollador de Software	1 (1.25) [0.05]	4 (3.75) [0.02]	5
Marginal Column Totals	3	9	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.1143. El valor *p-value* es 0.735317. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 28 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.735317 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 9

Figura 29

Tabla de contingencia. Pregunta #22 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	1 (1.17) [0.02]	6 (5.83) [0]	7
Ingeniero / Desarrollador de Software	1 (0.83) [0.03]	4 (4.17) [0.01]	5
Marginal Column Totals	2	10	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 0.0686. El valor *p-value* es 0.793428. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 29 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.793428 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Contraste 10

Figura 30

Tabla de contingencia. Pregunta #24 de la encuesta sección #3

	No Duplicada	Duplicada	*Marginal Row Totals*
Estudiante	5 (4.08) [0.21]	2 (2.92) [0.29]	7
Ingenierio / Desarrollador de Software	2 (2.92) [0.29]	3 (2.08) [0.4]	5
Marginal Column Totals	7	5	12 (Grand Total)

Nota: La estadística Chi-cuadrado es 1.1853. El valor *p-value* es 0.276278. No significativo a p<0.05. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

En la Figura 30 muestra la tabla de contingencia de la primera hipótesis en donde se calcula el *p-value*, que detalla que es mayor (0.276278 > 0,05) que de acuerdo con los criterios significancia se acepta la hipótesis nula (H_0), obteniendo como resultado que el modelo realizado con técnicas de aprendizaje automático ayuda a reconocer la duplicidad de preguntas entre SO y SOES.

Resultados

Como resultado de los experimentos realizados con técnicas de aprendizaje automático se determinó que el modelo con rendimiento más equilibrado es la red neuronal entrenada con el *dataset* balanceado con SMOTE.

Según se puede observar en la Tabla 10 a continuación, el modelo con mayor puntaje de exhaustividad es el de similitud de texto con WMD con umbral óptimo para la exhaustividad. Sin embargo, también se hace notable la baja puntuación de exactitud debido a que el modelo tiende a predecir muchos falsos positivos debido al elevado umbral de similitud que se necesita para obtener una exhaustividad de 1.

El segundo mejor puntaje de exhaustividad lo obtiene el modelo de similitud de texto con WMD con umbral óptimo para puntaje F1 el cual es la media harmónica entre la exhaustividad y la precisión. Como podemos observar en la tabla la exhaustividad se redujo a un 0,99 y se incrementó la exactitud a 0,35 de modo que se evidencia una mejora en el rendimiento, pero no es lo suficientemente significativa para ser considerado un modelo con resultados óptimos.

En cuanto al modelo de regresión logística se realizaron experimentos con el *dataset* desbalanceado y luego balanceado a través de SMOTE. Si bien es cierto la exhaustividad con respecto a los modelos con WMD se vio reducida significativamente, también se notó un incremento significativo en la exactitud. El desbalanceo del *dataset* resultó ser un gran problema en el entrenamiento por lo que el modelo entrenado con el *dataset* fue el que mejores resultados proporcionó.

Por último, se utilizó una red neuronal entrenada con el mismo *dataset* balanceado debido a que se descartó el realizar pruebas con el *dataset* original debido a sus problemas de balanceo. Como se muestra en la Tabla 10 se puede notar que la exhaustividad tuvo un incremento sustancial

con respecto al modelo anterior. Esto lo logra sin sacrificar demasiado su rendimiento en cuanto a exactitud por lo que debido a este equilibrio en rendimiento se eligió el mejor modelo de los experimentos realizados.

Tabla 10

Resultados de las distintas técnicas probadas en el desarrollo del algoritmo.

Técnicas utilizadas	**Exactitud**	**Exhaustividad**
WMD con umbral óptimo para exhaustividad	0,28	1,00
WMD con umbral óptimo para puntaje F1	0,35	0,99
Regresión Logística con *dataset* desbalanceado	0,76	0,08
Regresión Logística con *dataset* balanceado	0,68	0,48
Red neuronal con *dataset* balanceado	0,54	0,64

Nota: Esta tabla de las distintas técnicas probadas en el desarrollo del algoritmo. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Luego de haber contrastado los resultados de los experimentos con un juicio externo utilizando el estadístico chi-cuadrado se puede determinar que la hipótesis planteada es correcta. Es decir, que las herramientas y técnicas de aprendizaje automático en efecto ayudan a detectar la presencia de preguntas duplicadas entre los sitios de Stack Overflow y Stack Overflow en español. Esto se hace evidente al analizar los resultados individuales de las diez preguntas que se incluyeron en la encuesta. Se pudo apreciar que en cada una de estas preguntas el *p-value* resulta ser mayor al nivel de significancia determinado, en este caso de .05. Al ser mayor qué nivel de significancia el *p-value* indica que la hipótesis nula se acepta. Dado que éste fue el caso para las 10 preguntas propuestas en la encuesta se puede concluir que la hipótesis es verdadera.

Puesto que la complejidad de establecer la duplicidad entre dos preguntas de los sitios Stack Overflow es bastante elevada no se puede determinar con exactitud el rendimiento de los algoritmos propuestos en el experimento ya que como se muestra en las tablas de contingencia no

todos los encuestados estuvieron de acuerdo en cuanto a la duplicidad de una pregunta determinada por lo que se hace evidente que resulta complicado incluso para los humanos determinar cuándo una pregunta es duplicada. Esto puede suceder debido a varios factores que pueden dificultar la comparación. Entre estos tenemos el mal uso del lenguaje, la no inclusión de código fuente descriptivo o por el contrario la inclusión de código fuente no descriptivo que puede llegar a crear confusión y no dejar que se haga evidente la intención de la pregunta.

Ante lo expuesto se puede reconocer que la aplicación de técnicas de aprendizaje automático logra detectar un porcentaje de las preguntas duplicadas con lo cual se cumple el objetivo del estudio sentando las bases para próximos estudios que deseen ahondar en este tema y de alguna manera mejorar y complementar el trabajo realizado.

CAPÍTULO IV

En este capítulo se detallan los hallazgos producto de la realización de este proyecto investigación. En el apartado de conclusiones y recomendaciones se describirán las formas en que se han cumplido los objetivos específicos y cómo se pueden realizar mejoras en una revisión futura del proyecto. En la sección de conclusiones se encontrarán párrafos que corresponden a los resultados del proyecto con respecto a cada uno de los objetivos específicos definidos anteriormente. Por su parte, en la sección de recomendaciones se incluirá información que sea de utilidad para la mejora de la estrategia propuesta en este proyecto. Esto se hace con el fin de que se logren sortear o evitar los problemas encontrados en el estudio realizado y se pueda definir un alcance más amplio. La sección de trabajos futuros incluirá pautas sobre los trabajos que se pueden realizar a partir de los hallazgos y resultados de este proyecto de investigación.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Como resultado de la revisión sistemática conducida como cumplimiento del primer objetivo se llegó a la conclusión de que las incorporaciones de palabras son de los métodos más utilizados al momento de detectar y recuperar preguntas duplicadas. Cuando se combinan con técnicas de PLN pueden mejorar considerablemente el rendimiento de las soluciones propuestas. Es interesante también que las incorporaciones de palabras puedan usar utilizadas en conjunto con redes neuronales de modo que se obtienen resultados interesantes al momento de realizar las tareas de detección. También se encontró que el aprendizaje de tipo no supervisado era el más utilizado pero la mayoría de las veces se utilizaba como

herramienta para la caracterización y no para la clasificación por lo que es recomendable utilizarlo en conjunto con el aprendizaje supervisado. El aprendizaje profundo no fue utilizado en muchos estudios, pero si demostró su eficacia al combinarlo con incorporaciones de palabras. De las herramientas encontradas se tomaron varias ideas y estrategias sin llegar a utilizarlas directamente para el algoritmo final. Como resultado de este análisis literario se plantearon varias estrategias para el desarrollo del algoritmo.

- Con el fin de cumplir con el segundo objetivo se logró realizar una extracción de los datos de los sitios Stack Overflow y Stack Overflow en español de modo que se pudo construir un *dataset* que se ajustaba a las necesidades del estudio. Es importante recalcar la utilización de la API de Stack Exchange ya que es una estrategia que no se pudo apreciar en los estudios encontrados en la revisión sistemática pero que sin embargo resultó bastante útil y novedosa en la tarea de extracción de datos para construir el *dataset*. Utilizar la API resolvió el problema de tener que descargar data dumps con tamaños bastante elevados que muchas veces contiene información innecesaria para el estudio ya que con la API se pueden generar consultas para información específica que cumpla con parámetros definidos por el propio usuario.
- Para cumplir con el objetivo tres se desarrolló un algoritmo tomando varias ideas de los estudios encontrados en la revisión sistemática. Sin embargo, este contiene una estrategia de detección distinta a la de los estudios revisados como se explicó en el capítulo 3. El rendimiento del algoritmo en términos generales podría parecer bajo, pero al comparar sus resultados con el contraste realizado a través de las

encuestas se puede notar que los resultados y las métricas se asemejan bastante por lo que se puede concluir que funciona lo suficientemente bien y los errores de predicción se encuentran dentro de lo aceptable. Cabe mencionar que son muy pocos los estudios que se han realizado sobre la detección de preguntas duplicadas multilingües en Stack Overflow llegando incluso a no haberse encontrado ningún estudio en la revisión sistemática que comparara preguntas en inglés y español. Debido a esto es difícil comparar los resultados del algoritmo y determinar si su rendimiento es superior o inferior ya que para efectos de este estudio no se encontraron herramientas similares.

- Para la evaluación del algoritmo y como cumplimiento del cuarto objetivo se recolectaron datos a través de encuestas bajo los parámetros definidos en el capítulo 3. Como se mencionó anteriormente, resulta difícil evaluar el algoritmo presentado debido a la falta de elementos para realizar la comparación. Utilizando las encuestas como técnica de recolección de datos se logró obtener una evaluación asistida por desarrolladores de la comunidad Stack Overflow. Sin embargo, debido a una interrupción en el proceso de recolección ocasionada por un bloqueo por parte de un moderador de uno de los sitios no se logró recolectar la cantidad de datos esperada para satisfacer la muestra. Fueron 12 respuestas las que se obtuvieron de las 17 esperadas lo que representa un 70% de datos recolectados por lo cual se decidió continuar con el análisis ya que en términos relativos se logró recolectar una cantidad significativa de datos para realizar la evaluación. En cuanto al análisis de las hipótesis realizado a través del cálculo del estadístico chi-cuadrado para cada una de las preguntas de la encuesta, como se pudo observar en el capítulo 3, se

logró determinar que en efecto el algoritmo propuesto permite detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español.

Recomendaciones

- Si se desea obtener una cantidad mayor de estudios en la revisión sistemática se recomienda establecer criterios menos estrictos de inclusión y exclusión. Esto depende directamente del tema a analizar y de la pregunta de investigación a contestar.
- Utilizar alguna herramienta informática como StArt Tool para asistir en el proceso de revisión sistemática. Al facilitar ciertos aspectos del proceso el investigador puede enfocarse más en realizar un análisis de la literatura más completo.
- Para la extracción de datos mediante la API de Stack Exchange se recomienda crear una aplicación a través del sitio Stack Apps ya que así el usuario obtendrá una cuota de peticiones mucho mayor a la ofrecida a los usuarios en general pasando de 300 consultas diarias a 10000. Esto permitirá extraer más datos y más importante aún realizar más pruebas al momento de aprender a usar la API por primera vez.
- Si se desea realizar el entrenamiento de un modelo con un gran volumen de datos se podría aprovechar la funcionalidad de Google Colab de utilizar un GPU en sus máquinas virtuales y así reducir el tiempo de ejecución de operaciones complejas.
- Cuando se vaya a recolectar información dentro de los sitios Stack Overflow mediante una encuesta se recomienda contactar primero varios moderadores de las comunidades con el fin de explicar el objetivo del estudio y obtener recomendaciones y/o ayuda para obtener los datos de las encuestas.

Trabajos futuros

- Como trabajo relacionado a este estudio se puede considerar la utilización de un *dataset* de entrenamiento con un tamaño mucho mayor ya que el tamaño del *dataset* utilizado en este estudio influyó en el rendimiento de los algoritmos debido a que la cantidad de datos para realizar el entrenamiento no se podía equilibrar adecuadamente con la cantidad necesaria para realizar las pruebas. Dado a que el trabajo de etiquetado entre las preguntas de los dos sitios es manual es necesario tener el tiempo adecuado para realizar un trabajo como el descrito anteriormente.
- Debido a las limitaciones de tiempo y recursos no se pudo realizar el entrenamiento de un modelo de incorporaciones de palabras propio de este estudio teniendo que utilizar un modelo pre entrenado. Como trabajo futuro se podía plantear el entrenamiento de un modelo utilizando los datos de ambos sitios. Esto para que el modelo resultante se ajuste mucho más a las necesidades del estudio y por lo tanto los experimentos tengan la posibilidad de ofrecer mejores resultados ya que al usar modelos pre entrenados muchas veces se pierden ciertas funcionalidades importantes propias de un modelo de incorporación de palabras.

REFERENCIAS BIBLIOGRÁFICAS

Ahasanuzzaman, M., Asaduzzaman, M., Roy, C. K., & Schneider, K. A. (2016). Mining Duplicate Questions in Stack Overflow. *IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, (págs. 402-412). Austin, TX, USA.

Alonso Martínez, I. (2016). Análisis y optimización del recurso UMLS en la recuperación de información biomédica mediante métricas de similitud semántica.

Amengual Bauza, M. (Junio de 2019). Security in API and API managers.

Antona Cortés, C. (Enero de 2017). Herramientas modernas en redes neuronales: la librería Keras. Obtenido de https://repositorio.uam.es/bitstream/handle/10486/677854/antona_cortes_carlos_tfg.pdf?sequence=1

Bakarov, A. (2018). A Survey of Word Embeddings Evaluation Methods. *arXiv preprint arXiv: 1801.09536*.

Barn, B., Barat, S., & Clark, T. (2017). Conducting systematic literature reviews and systematic mapping studies. *Proceedings of the 10th Innovations in Software Engineering Conference*, (págs. 212-213). Jaipur.

Barrera Bolivar, A. C. (2020). Diseño e implementación de un modelo analítico predictivo para el apoyo en la toma de decisiones enfocado en las empresas de telecomunicaciones. Universidad Santo Tomás.

Barrios Arce, J. I. (26 de Julio de 2019). *La matriz de confusión y sus métricas.* Obtenido de Big Data: https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Bello García, B. (Junio de 2019). Biblioteca para el preprocesamiento de datos usando conjuntos aproximados. Doctoral dissertation, Universidad Central “Marta Abreu” de Las Villas.

Berón Abou-Nigm, R., & Jardim Godoy, E. (28 de Diciembre de 2017). SCARR Sistema Clasificador Automático de Respuestas según Relevancia. Obtenido de https://www.fing.edu.uy/inco/grupos/pln/prygrado/SCARR_Documento_Principal.pdf

Bofang, L., Aleksandr, D., Yuhe, G., Tao, L., Satoshi, M., & Xiaoyong, D. (2019). Scaling word2vec on big corpus. *Data Science and Engineering*, 157-175.

Botto-Tobar, M., Ramirez Anormaliza, R., Cevallos Torres, L. J., & Cevallos Ayon, E. (2017). Migrating SOA applications to cloud: a systematic mapping study. *International Conference on Technologies and Innovation* (págs. 3-16). Springer, Cham.

Botto-Tobar, M., van den Brand, M. G., Torres, W., Vasilescu, B., Lozano, A., & Serebrenik, A. (2018). Is Stack Overflow in Portuguse attractive for Brazilian Users? *Proceedings of the 13th International Conference on Global Software Engineering*, (págs. 21-29).

Brownlee, J. (11 de Noviembre de 2019). *14 Different Types of Learning in Machine Learning*. Recuperado el Febrero de 2021, de Machine Learning Mastery: https://web.archive.org/web/20201130084741/https://machinelearningmastery.com/types-of-learning-in-machine-learning/

Cabezas Mejía, E. D., Andrade Naranjo, D., & Torres Santamaría, J. (2018). *Introducción a la metodología de la investigación científica.* Universidad de las Fuerzas Armadas ESPE.

Calibar , A. B., Holleger, J., & Klenzi, R. O. (2018). Análisis de similitud en documentos de texto mediante técnicas de ciencia de datos basadas en aprendizaje profundo (deep learning). *XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste)*, (págs. 246-250).

Caparrini, F. S. (14 de Diciembre de 2020). *Aprendizaje Supervisado y No Supervisado.* Obtenido de http://www.cs.us.es/~fsancho/?e=77

Carballo, M., & Guelmes, E. (2016). Algunas consideraciones acerca de las variables en las investigaciones que se desarrollan en educación. *Revista Universidad y Sociedad, 1*(8), 140-150.

Casacuberta Nolla, F., & Peris Abril, Á. (2017). Traducción automática neuronal. *Tradumática: tecnologies de la traducció*, 66-74.

Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 1-13.

Chowdhury, G. (2020). Natural Language Processing. *Fundamentals of Artificial Intelligence* (págs. 603-649). Springer, New Delhi.

Código Orgánico de la Economia Social de los Conocimientos, Creatividad e Innovación. (25 de Noviembre de 2020). Obtenido de https://lotaip.ikiam.edu.ec/ikiam2019/abril/anexos/Mat%20A2-Base_Legal/codigo_organico_de_la_economia%20social_de_los_conocimientos_creatividad_e_innovacion.pdf

Constitución de la República del Ecuador. (27 de Octubre de 2020). Obtenido de https://www.wipo.int/edocs/lexdocs/laws/es/ec/ec030es.pdf

de la Fuente Sanz, Ó. M. (04 de Junio de 2019). *Google Colab: Python y Machine Learning en la nube*. Obtenido de Adictos al trabajo: https://www.adictosaltrabajo.com/2019/06/04/google-colab-python-y-machine-learning-en-la-nube/

Education First. (2020). *índice del EF English Proficiency*. Obtenido de EF: https://www.ef.com.ec/epi/regions/latin-america/ecuador/#

Efstathiou, V., Chatzilenas, C., & Spinellis, D. (2018). Word Embeddings for the Software Engineering Domain. *Proceedings of the 15th International Conference on Mining Software Repositories.* ACM.

Espino Timón, C. (16 de Enero de 2017). Análisis predictivo: técnicas y modelos utilizados y aplicaciones del mismo-herramientas Open Source que permiten su uso.

Fernández, Y. (23 de Agosto de 2019). *API: qué es y para qué sirve.* Obtenido de Xataka Basics: https://www.xataka.com/basics/api-que-sirve

Gasca Meza, G. (05 de Julio de 2018). *Tu primer red neuronal usando Keras*. Obtenido de Medium: https://medium.com/@gogasca_/tu-primer-red-neuronal-usando-keras-72d36130ee6c

Google. (s.f.). *Acerca de Google Académico*. Recuperado el Febrero de 2021, de Google Académico: https://scholar.google.com/intl/es/scholar/about.html

Hinojosa Gutiérrez, Á. P. (2016). *Python paso a paso.* Grupo Editorial RA-MA.

Holzinger, A., Kieseberg, P., Weippl, E., & Tjoa, A. M. (2018). Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable AI . *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (págs. 1-8). Springer, Cham.

Huarcaya Taquiri, D. (Octubre de 2020). Traducción automática neuronal para lengua nativa peruana.

Iberdrola. (24 de Noviembre de 2020). *Iberdrola S.A. Web site.* Obtenido de https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico

Icart Isern, M. T., & Canela Soler, J. (1998). El uso de hipótesis en la investigación científica. *Atención Primaria, 21*(3), 172–178.

IMF Business School. (08 de Junio de 2020). *El procesamiento del lenguaje natural: sus modelos y sus usos prácticos.* Obtenido de IMF Business School: https://blogs.imf-formacion.com/blog/tecnologia/procesamiento-lenguaje-natural-modelos-usos-practicos-202006/

Jarrín Rodríguez, D. A. (06 de Septiembre de 2019). Estudio comparativo entre modelos de aprendizaje profundo, desarrollados a partir de redes neuronales recurrentes a redes neuronales convolucionales, para la detección de intrusos de red. Universidad Internacional SEK.

Khurana, D., H., V., & K., S. (22 de Noviembre de 2017). Semantic Text Similarity - Detecting Duplicate Questions.

Ley Orgánica de Educación Superior. (29 de Enero de 2021). *Ley Orgánica de Educación Superior (LOES)* . Obtenido de https://siteal.iiep.unesco.org/sites/default/files/sit_accion_files/ec_6011.pdf

Marker, G. (2020 de Septiembre de 2020). *¿Qué es un algoritmo? Definición, características y tipos. Algoritmo computacional.* Obtenido de Tecnologia Informática: https://www.tecnologia-informatica.com/algoritmo-definicion/

Martinez Heras, J. (10 de Octubre de 2020). *15 Librerías de Python para Machine Learning*. Obtenido de IArtificial.net: https://www.iartificial.net/librerias-de-python-para-machine-learning/#gensim

Martinez, P. (13 de Febrero de 2019). *Google Colab: Tips para principiantes*. Obtenido de Medium: https://medium.com/marvik/google-colab-tips-para-principiantes-e39d6e7051d4

Medrano, J. F. (2020). Enfoque Combinado de Word2Vec y 2-grams para la Recuperación de Avisos Clasificados Inmobiliarios Semánticamente Relacionados. *Revista Tecnología y Ciencia*, 195-206.

Microsoft Azure. (2021). *¿Qué es el aprendizaje automático?* Obtenido de Microsoft Web site: https://azure.microsoft.com/es-es/overview/what-is-machine-learning-platform/

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning.* MIT press.

Moya, R. (28 de Noviembre de 2020). *Pandas en Python, con ejemplos -Parte I- Introducción*. Obtenido de Jarroba: https://jarroba.com/pandas-python-ejemplos-parte-i-introduccion/

NetApp. (30 de Octubre de 2020). *¿Qué es el aprendizaje automático?* Obtenido de NetApp Web site: https://www.netapp.com/es/artificial-intelligence/what-is-machine-learning/

NumPy. (28 de Noviembre de 2020). *What is NumPy?* Obtenido de NumPy: https://numpy.org/doc/stable/user/whatisnumpy.html

Páez Juka, S. D. (18 de Junio de 2019). *Análisis comparativo de herramientas Open Source para Data Mining sobre datos públicos del Ministerio de Educación de la República del Ecuador.* Obtenido de Repositorio de Tesis de Grado y Posgrado: http://repositorio.puce.edu.ec/bitstream/handle/22000/17060/Tesis%20Sergio%20P%C3%A1ez%20Doc.pdf?sequence=1&isAllowed=y

Pereira, M. (13 de Noviembre de 2020). *Cómo funciona Google Drive*. Obtenido de Hotmart: https://blog.hotmart.com/es/google-drive/

Portell, M., & Vives, J. (2019). *Investigación en Psicología y Logopedia: Introducxción a los diseños experimentales, cuasi-experimentales y ex post facto* (Vol. 60). Barcelona: Servei de Publicacions de la Universitat Autònoma de Barcelona.

Qiu, Z. (2018). Multilingual Stack Overflow Empirical Study and Question Retrieval Tool.

Ramos, F. M., & Velez, J. I. (Mayo de 2016). Integración de técnicas de procesamiento de lenguaje natural a través de servicios web.

Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning.* Packt Publishing Ltd. Obtenido de https://books.google.es/books?hl=es&lr=&id=_plGDwAAQBAJ&oi=fnd&pg=PP1&ots=8tDGhSfoBE&sig=0patv4fjTrVzfFYoyK2yA4d1eaA#v=onepage&q&f=false

Ray, S. (9 de Septiembre de 2017). *Commonly used Machine Learning Algorithms (with Python and R Codes).* Recuperado el Febrero de 2021, de Analytics Vidhya: https://web.archive.org/web/20210205230753/https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

Recuero de los Santos, P. (09 de Septiembre de 2020). *Cómo interpretar la matriz de confusión: ejemplo práctico.* Obtenido de Empresas BlogThinkBig: https://empresas.blogthinkbig.com/como-interpretar-la-matriz-de-confusion-ejemplo-practico/

Restrepo Klinge, S. (2019). Google Translate vs Traducción Humana: percepciones de ocho traductores en torno al papel de este traductor automático en su labor. Pontificia Universidad Javeriana.

Robledano, Á. (18 de Junio de 2019). *Qué es un algoritmo informático.* Obtenido de Open Webinars: https://openwebinars.net/blog/que-es-un-algoritmo-informatico/

Rodríguez, D. (26 de Noviembre de 2018). *Diferencias entre regresión y clasificación en aprendizaje automático.* Obtenido de Analytics Lane:

https://www.analyticslane.com/2018/11/26/diferencias-entre-regresion-y-clasificacion-en-aprendizaje-automatico/

Rouhiainen, L. (2018). *Inteligencia Artificial.* Madrid: Alienta Editorial.

Saabith, A. S., Fareez, M. M., & Vinothraj, T. (2019). Python current trend applications-an overview. *International Journal of Advance Engineering and Research Development, 6*(10).

Sánchez Alberca, A. (04 de Octubre de 2020). *La librería Numpy*. Obtenido de Aprende con Alf: https://aprendeconalf.es/docencia/python/manual/numpy/

Sánchez Alberca, A. (Octubre de 2020). *La librería Pandas*. Obtenido de Aprende con Alf: https://aprendeconalf.es/docencia/python/manual/pandas/

Saxena, N. (26 de Agosto de 2019). *Word Mover's Distance for Text Similarity.* Obtenido de Towards Data Science: https://towardsdatascience.com/word-movers-distance-for-text-similarity-7492aeca71b0#:~:text=Word%20Mover's%20Distance%20targets%20both,embedded%20words%20of%20another%20document.

Significados. (28 de Octubre de 2020). *Investigación de campo*. Recuperado el 2021, de Significados: https://www.significados.com/investigacion-de-campo/

Stack Exchange. (11 de Enero de 2021). *The world's largest programming.* Obtenido de StackExchange: https://stackexchange.com/about

Stack Exchange, Inc. (26 de Enero de 2021). *StackExchange API*. Obtenido de StackExchange: https://api.stackexchange.com/docs

Stack Exchange, Inc. (s.f.). *question Type*. Recuperado el Febrero de 2021, de Stack Exchange API: https://api.stackexchange.com/docs/types/question

Stack Overflow en español. (27 de Septiembre de 2020). *Bienvenido a Stack Overflow en español*. Obtenido de Stack Overflow en español: https://es.stackoverflow.com/tour

Suárez Lamadrid, A. (2018). Aplicación de técnicas de aprendizaje profundo (deep learning) a clasificación de imágenes histológicas. *(Tesis de Grado)*.

Talamé, L., Cardoso, A., & Amor, M. (2019). Comparación de herramientas de procesamiento de textos en español extraídos de una red social para Python. *XX Simposio Argentino de Inteligencia Artificial (ASAI 2019)-JAIIO 48 (Salta)* (págs. 53-67). ASAI.

Tarek, S. (11 de Junio de 2020). *Document Similarity using Word Mover's Distance and Cosine similarity.* Obtenido de Medium: https://medium.com/@tarekseif0/document-similarity-using-word-movers-distance-and-cosine-similarity-d698ad435422

Tebes, G., Peppino, D., Becker, P., & Olsina, L. (2020). Proceso para Revisión Sistemática de Literatura y Mapeo Sistemático. , 19(2), . *Electronic Journal of SADIO (EJS)*, 94-118.

Vallalta Rueda, J. F. (04 de Agosto de 2019). *Aprendizaje supervisado y no supervisado.* Obtenido de Health Data Miner: https://healthdataminer.com/data-mining/aprendizaje-supervisado-y-no-supervisado/

Wang, L., Zhang, L., & Jing, J. (2020). Duplicate Question Detection With Deep Learning in Stack Overflow. *IEEE Access*, 25964 - 25975.

Wang, Y. (2019). An Empirical Study on Multilingual Q&A Communities and Crosssite Question Retrieval Method.

Wegner, A. (18 de Enero de 2019). *StackAPI*. Obtenido de PyPi: https://pypi.org/project/StackAPI/

Xu, B., Xing, Z., Xia, X., Lo, D., & Le, X. B. (2017). Xsearch: a domain-specific cross-language relevant question retrieval tool. *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, (págs. 1009-1013).

Zhang, W. E., Sheng, Q., Lau, J., Abebe, E., & Ruan, W. (2018). Duplicate Detection in Programming Question Answering Communities. *ACM Transactions on Internet Technology*, 1-21.

Zhang, Y., Lo, D., Xia, X., & Sun, J. L. (2015). Multi-factor duplicate question detection in stack overflow. . *Journal of Computer Science and Technology*, 981-997.

ANEXOS

Anexo 1. Planificación de actividades del proyecto

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
1		**Proyecto: DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS)**	**93 días**	**lun 9/11/20**	**mar 16/3/21**		
2		**Inicio del proyecto**	**51 días**	**lun 9/11/20**	**lun 18/1/21**		
3		Reunión Tutor: Socializar tema de Proyecto	1 día	lun 9/11/20	lun 9/11/20		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
4		Propuesta de Trabajo de Titulación	6 días	mar 10/11/2(	mar 17/11/20		Kerly Chica;Bryan Moreira
5		Planeación del Mapeo Sistemático de la literatura	6 días	lun 23/11/20	dom 29/11/20		Kerly Chica;Bryan Moreira
6		Reunión Tutor: Revisión Propuesta de Trabajo de Titulación	1 día	mar 24/11/20	mar 24/11/20		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
7		Corrección de Propuesta de Trabajo de Titulación	11 días	mar 24/11/20	mar 8/12/20		Kerly Chica;Bryan Moreira
8		Reunión Tutor: Revisión de Preguntas del Mapeo Sistemático	1 día	sáb 26/12/20	sáb 26/12/20		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
9		Revisión de la planeación del mapeo sistemático de la literatura	6 días	lun 30/11/20	dom 6/12/20		Bryan Moreira
10		Entrega de Anexo, Anexo I, Anexo II, Cronograma	9 días	lun 30/11/20	jue 10/12/20		Kerly Chica;Bryan Moreira
11		Corrección de preguntas de investigación y cadenas de búsqueda. Pruebas de cadenas de búsqueda	6 días	lun 7/12/20	dom 13/12/20		Kerly Chica;Bryan Moreira
12		Reunión Tutor: Revision de preguntas y cadena de búsqueda de mapeo	1 día	mar 8/12/20	mar 8/12/20		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
13		Mapeo sistemático de la literatura	6 días	lun 14/12/20	dom 20/12/2(		Bryan Moreira
14		Reunión Tutor: Avanses de la revisión sistemática	1 día	mié 16/12/20	mié 16/12/20		
15		Extracción de datos de publicaciones, construcción del conjunto de datos	6 días	lun 21/12/20	dom 27/12/20		Bryan Moreira
16		Desarrollo del Algoritmo	6 días	lun 28/12/20	dom 3/1/21		Bryan Moreira
17		Evaluación de desempeño del algoritmo	6 días	lun 4/1/21	dom 10/1/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
18		Anexo IV 1er informe de avance de la gestión tutorial	3 días	lun 4/1/21	mié 6/1/21		Kerly Chica
19		Reunión Tutor: Revisión del algoritmo, extracción de datos de los sitios StackOverflow	1 día	mar 5/1/21	mar 5/1/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
20		**CAPITULO I**	**6 días**	**lun 11/1/21**	**dom 17/1/21**		
21		**Planteamiento del problema**	**6 días**	**lun 11/1/21**	**dom 17/1/21**		Kerly Chica;Bryan Moreira
22		Descripción de la situación problemáti(	1 día	lun 11/1/21	lun 11/1/21		Kerly Chica
23		Causas y Consecuencias del Problema	1 día	mar 12/1/21	mar 12/1/21	22	Kerly Chica
24		Formulación del problema	1 día	mié 13/1/21	mié 13/1/21	23	Kerly Chica
25		Objetivos del proyecto	1 día	jue 14/1/21	jue 14/1/21	24	Kerly Chica
26		Alcance del Proyecto	1 día	vie 15/1/21	vie 15/1/21	24	Kerly Chica
27		Justificación e importancia	1 día	sáb 16/1/21	sáb 16/1/21	24	Kerly Chica
28		limitaciones del estudio	1 día	sáb 16/1/21	sáb 16/1/21	24	Bryan Moreira

Elaboración: Investigadores.
Fuente: Propia.

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
29		Reunión Tutor: Revisión Dataset de las preguntas similares	1 día	mar 12/1/21	mar 12/1/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
30		Revisión y corrección del Capítulo I	6 días	lun 18/1/21	dom 24/1/21		Kerly Chica;Bryan Moreira
31		**CAPITULO II**	**6 días**	**lun 25/1/21**	**dom 31/1/21**	**20**	
32		**Marco teórico**	**6 días**	**lun 25/1/21**	**dom 31/1/21**		Kerly Chica;Bryan Moreira
33		Antecedentes del estudio	1 día	lun 25/1/21	lun 25/1/21		Bryan Moreira
34		Fundamentación teórica	5 días	mar 26/1/21	sáb 30/1/21		Kerly Chica
35		Revisiones Sistemáticas, hipótesis, variables, definiciones conceptuales	5 días	mar 26/1/21	dom 31/1/21		Bryan Moreira
36		Reunión Tutor: Algoritmo, revisión del etiquetado de las preguntas	1 día	mar 26/1/21	mar 26/1/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
37		Revisión y corrección del Capítulo II	6 días	lun 1/2/21	dom 7/2/21		Kerly Chica;Bryan Moreira
38		Reunión Tutor: Revisión de recolección de datos de contacto de usuario para	1 día	mar 2/2/21	mar 2/2/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
39		Anexo IV 2do informe de avance de la gestión tutorial	3 días	jue 4/2/21	lun 8/2/21		Kerly Chica
40		Reunión Tutor: Revisión del desempeño del algoritmo, propuesta de técnicas	1 día	sáb 6/2/21	sáb 6/2/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
41		**CAPITULO III**	**6 días**	**lun 8/2/21**	**dom 14/2/21**	**31**	
42		**Metodología de la investigación**	**6 días**	**lun 8/2/21**	**dom 14/2/21**		**Kerly Chica**
43		Tipo de investigacíon, Diseño metodológico de investigación	4 días	lun 8/2/21	jue 11/2/21		Kerly Chica
44		Entregables, Propuesta, Criterios de validación de la propuesta y resultado	3 días	jue 11/2/21	dom 14/2/21		Kerly Chica;Bryan Moreira
45		Reunión Tutor: Revisión de formato de encuesta	1 día	mar 9/2/21	mar 9/2/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
46		Revisión y correción del Capítulo III	6 días	lun 15/2/21	dom 21/2/21		Kerly Chica;Bryan Moreira
47		**CAPITULO IV**	**7 días**	**lun 22/2/21**	**dom 28/2/21**	**41**	
48		**Conclusiones y Recomendaciones**	**7 días**	**lun 22/2/21**	**dom 28/2/21**		Kerly Chica;Bryan Moreira
49		Conclusiones	3 días	lun 22/2/21	mié 24/2/21		Kerly Chica;Bryan Moreira
50		Recomendaciones	2 días	jue 25/2/21	vie 26/2/21	49	Kerly Chica;Bryan Moreira
51		Trabajos Futuros	2 días	sáb 27/2/21	dom 28/2/21	49;50	Kerly Chica;Bryan Moreira
52		Revisión y corrección del Capítulo IV	6 días	lun 1/3/21	dom 7/3/21		Kerly Chica;Bryan Moreira
53		Anexo V, Anexo VI, Anexo VIII y documento del Proyecto de Titulación	10 días	mié 3/3/21	mar 16/3/21		Kerly Chica;Bryan Moreira
54		Revisión final y aprobación del tutor	2 días	lun 8/3/21	mar 9/3/21		Kerly Chica;Bryan Moreira;Ing. Miguel Botto
55		**Fin del Proyecto**					

Elaboración: Investigadores.
Fuente: Propia**.**

Anexo 2. Fundamentación Legal

El presente proyecto de titulación se fundamenta en documentos tales como la: Constitución de la República del Ecuador, Ley Orgánica de Educación Superior y Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación. Estos documentos serán la base legal de este proyecto y se detallan a continuación:

CONSTITUCIÓN DE LA REPUBLICA DEL ECUADOR

TITULO I

ELEMENTOS CONSTITUTIVOS DEL ESTADO

Capítulo segundo

Derechos del buen vivir

Sección tercera

Comunicación e Información

Art. 16.- Todas las personas, en forma individual o colectiva, tienen derecho a:

1. Una comunicación libre, intercultural, incluyente, diversa y participativa, en todos los ámbitos de la interacción social, por cualquier medio y forma, en su propia lengua y con sus propios símbolos.
2. El acceso universal a las tecnologías de información y comunicación.
3. La creación de medios de comunicación social, y al acceso en igualdad de condiciones al uso de las frecuencias del espectro radioelectrónico para la gestión de estaciones de radio y televisión públicas, privadas y comunitarias, y a bandas libres para la explotación de redes inalámbricas.
4. El acceso y uso de todas las formas de comunicación visual, auditiva, sensorial y a otras que permitan la inclusión de personas con discapacidad.

5. Integrar los espacios de participación previstos en la Constitución en el campo de la comunicación.

Sección cuarta

Cultura y ciencia

Art. 22.- las personas tienen derecho a desarrollar su capacidad creativa, al ejercicio digno y sostenido de las actividades culturales y artísticas, y a beneficiarse de la protección de los derechos morales y patrimoniales que les correspondan por las producciones científicas, literarias o artísticas de su autoría.

Sección quinta

Educación

Art. 26.- La educación es un derecho de las personas a lo largo de su vida y un deber ineludible e inexcusable del Estado. Constituye un área prioritaria de la política pública y de la inversión estatal, garantía de la igualdad e inclusión social y condición indispensable para el buen vivir. Las personas, las familias y la sociedad tienen el derecho y la responsabilidad de participar en el proceso educativo.

Art. 28.- la educación responderá al interés público y no estará al servicio de intereses individuales y corporativos. Se garantizará el acceso universal, permanencia, movilidad y egreso sin discriminación alguna y la obligatoriedad en el nivel inicial, básico y bachillerato o su equivalente.

Es derecho de toda persona y comunidad interactuar entre culturas y participar en una sociedad que aprende. El estado promoverá el dialogo intercultural en sus múltiples dimensiones.

El aprendizaje se desarrollará de forma escolarizada y no escolarizada.

La educación pública será universal y laica en todos sus niveles, y gratuita hasta el tercer nivel de educación superior inclusive.

TÍTULO VI

RÉGIMEN DE DESARROLLO

Capítulo sexto

Trabajo y Producción

Sección segunda

Tipos de propiedad

Art. 322.- Se reconoce la propiedad intelectual de acuerdo con las condiciones que señale la ley. Se prohíbe toda forma de aprobación de conocimientos colectivos, en el ámbito de las ciencias, tecnologías y saberes ancestrales. Se prohíbe también la apropiación sobre los recursos genéticos que contienen la diversidad biológica y la agrobiodiversidad.

TÍTULO VII

RÉGIMEN DEL BUEN VIVIR

Capítulo primero

Inclusión y Equidad

Sección primera

Educación

Art. 350.- El sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnología; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo.

Art. 355.- El Estado reconocerá a las universidades y escuelas politécnicas autonomía académica, administrativa, financiera y orgánica, acorde con los objetivos del régimen de desarrollo y los principios establecidos en la Constitución.

Se reconoce a las universidades y escuelas politécnicas el derecho a la autonomía, ejercida y comprendida de manera solidaria y responsable.

Dicha autonomía garantiza el ejercicio de la libertad académica y el derecho a la búsqueda de la verdad, sin restricciones; el gobierno y gestión de sí mismas, en consonancia con los principios de alternancia, transparencia y los derechos políticos; y la producción de ciencia, tecnología, cultura y arte.

Sección octava

Ciencia, tecnología, innovación y saberes ancestrales

Art. 385.- El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad:

1. Generar, adaptar y difundir conocimientos científicos y tecnológicos.
2. Recuperar, fortalecer y potenciar los saberes ancestrales.
3. Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir.

Art. 388.- El Estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del conocimiento. Un porcentaje de estos recursos se destinará a

financiar proyectos mediante fondos concursables. Las organizaciones que reciban fondos públicos estarán sujetas a la rendición de cuentas y al control estatal respectivo.

Elaboración: Investigadores.
Fuente: Constitución de la República del Ecuador (2020).

LEY ORGÁNICA DE EDUCACIÓN SUPERIOR

TÍTULO I

AMBITO, OBJETO, FINES Y PRINCIPIOS DEL SISTEMA DE EDUCACIÓN SUPERIOR

CAPÍTULO 1

AMBITO Y OBJETO

Art. 1.- Ámbito. - Esta Ley regula el sistema de educación superior en el país, a los organismos e instituciones que lo integran; determina derechos, deberes y obligaciones de las personas naturales y jurídicas, y establece las respectivas sanciones por el incumplimiento de las disposiciones contenidas en la constitución y la presente ley.

Art. 2.- Objeto. - Esta Ley tiene como objeto definir sus principios, garantizar el derecho a la educación superior de calidad que propenda a la excelencia, al acceso universal, permanencia, movilidad y egreso sin discriminación alguna.

CAPÍTULO 2

FINES DE LA EDUCACIÓN SUPERIOR

Art. 3.- Derecho a la Educación Superior. - El derecho a la educación superior consiste en el ejercicio efectivo de la igualdad de oportunidades, en función de los méritos respectivos, a fin de acceder a una formación académica y profesional con producción de conocimiento pertinente y de excelencia.

Las ciudadanas y los ciudadanos en forma individual y colectiva, las comunidades, pueblos y nacionalidades tienen el derecho y la responsabilidad de participar en el proceso educativo superior, a través de los mecanismos establecidos en la constitución y esta ley.

TÍTULO IV

IGUALDAD DE OPORTUNIDADES

CAPÍTULO 1

DEL PRINCIPIO DE IGUALDAD DE OPORTUNIDADES

Art. 87.- Requisitos previos a la obtención del título. – El principio de igualdad de oportunidades consiste en garantizar a todos los actores del Sistema de Educación Superior las mismas posibilidades en el acceso, permanencia, movilidad y egreso del sistema, sin discriminación de género, credo, orientación sexual, etnia, cultura, preferencia política, condición socioeconómica o discapacidad.

CAPÍTULO 2

DE LA GARANTÍA DE LA IGUALDAD DE OPORTUNIDADES

Art. 87.- Requisitos previos a la obtención del título. - Como requisito previo a la obtención del título, los y las estudiantes deberán acreditar servicios a la comunidad mediante practicas o pasantías preprofesionales, debidamente monitoreadas, en los campos de su especialidad, de conformidad con los lineamientos generales definidos por el Consejo de Educación Superior.

Dichas actividades se realizarán en coordinación con organizaciones comunitarias, empresas e instituciones públicas y privadas relacionadas con la respectiva especialidad.

TÍTULO V

CALIDAD DE LA EDUCACIÓN SUPERIOR

CAPÍTULO 1

DEL PRINCIPIO DE CALIDAD

Art. 93.- Principio de calidad. - El principio de calidad consiste en la búsqueda constante y sistemática de la excelencia, la pertinencia, producción optima, transmisión del conocimiento y desarrollo del pensamiento mediante la autocrítica, la crítica externa y el mejoramiento permanente.

TÍTULO VII

INTEGRIDAD

CAPÍTULO 2

DE LA TIPOLOGÍA DE INSTITUCIONES, Y RÉGIMEN ACADÉMICO

Sección Tercera

Del Funcionamiento de las Instituciones de Educación Superior

Art. 144.- Tesis Digitalizadas. – Todas las instituciones de educación superior estarán obligadas a entregar las tesis que se elaboren para la obtención de títulos académicos de grado y posgrado en formato digital para ser integradas al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

Elaboración: Investigadores.
Fuente: Ley Orgánica de Educación Superior (2021).

.

CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN

TÍTULO II

DE LOS DERECHOS DE AUTOR Y LOS DERECHOS CONEXOS

CAPÍTULO II

DE LOS DERECHOS DE AUTOR

Sección II

Objeto

Artículo 104.- Obras susceptibles de protección. - La protección reconocida por el presente Título recae sobre todas las obras literarias, artísticas y científicas, que sean originales y que puedan reproducirse o divulgarse por cualquier forma o medio conocido o por conocerse.

Las obras, las siguientes:

1. Las obras expresadas en libros, folletos, impresos, epistolarios, artículos, novelas, cuentos, poemas, crónicas, críticas, ensayos, misivas, guiones para teatro, cinematografía, televisión, conferencias, discursos, lecciones, sermones, alegatos en derecho, memorias y otras obras de similar naturaleza, expresadas en cualquier forma;
2. Colecciones de obras, tales como enciclopedias, antologías o compilaciones y bases de datos de toda clase, que por la selección o disposición de las materias constituyan creaciones intelectuales originales, sin perjuicio de los derechos que subsistan sobre las obras, materiales, información o datos;
3. Obras dramáticas y dramático musicales, las coreografías, las pantomimas y, en general las obras teatrales;
4. Composiciones musicales con o sin letra;
5. Obras cinematográficas y otras obras audiovisuales;
6. Las esculturas y las obras de pintura, dibujo, grabado, litografía y las historietas gráficas, tebeos, comics, así como sus ensayos o bocetos y las demás obras plásticas;
7. Proyectos, planos, maquetas y diseños de obras arquitectónicas y de ingeniería;

8. Ilustraciones, gráficos, mapas, croquis y diseños relativos a la geografía, la topografía y, en general, a la ciencia;
9. Obras fotográficas y las expresadas por procedimientos análogos a la fotografía;
10. Obras de arte aplicado, en la medida en que su valor artístico pueda ser disociado del carácter industrial de los objetos a los cuales estén incorporadas;
11. Obras remezcladas, siempre que, por la combinación de sus elementos, constituyan una creación intelectual original; y
12. Software.

Sección V

Disposiciones especiales sobre ciertas obras

Parágrafo Primero

Del software y bases de datos

Apartado Primero

Del software de código cerrado y base de datos

Artículo 131.- Protección de software. - El software se protege como obra literaria. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea como código fuente; es decir, en forma legible por máquina, ya sea sistemas operativos o sistemas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.

Elaboración: Investigadores.
Fuente: Código Orgánico de la Economia Social de los Conocimientos, Creatividad e Innovación (2020).

Anexo 3. Formatos de técnicas de recolección de datos aplicadas para variables cuantitativas o cualitativas.

Formato de Encuesta en español

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

* Obligatorio

stackoverflow en español

Encuesta de duplicidad en sitios StackOverflow (Español e Inglés).

stackoverflow

Bienvenido a la encuesta para detectar preguntas duplicadas en sitios StackOverflow (español e inglés).

En este estudio el objetivo es desarrollar un algoritmo de detección de preguntas duplicadas entre los sitios StackOverflow y StackOverflow en español utilizando técnicas de aprendizaje automático de modo que facilite la búsqueda de respuestas a los usuarios de los sitios.

Su participación en la encuesta es voluntaria y confidencial, no se solicitan datos personales ni información de contacto. Sin embargo, si desea participar en el sorteo de la gift card deberá ingresar un correo válido al final para enviar el código digital en caso de ser el ganador ya que la encuesta es anónima y Microsoft Forms no recopila información personal sobre los participantes. El tiempo estimado para completar la encuesta es de aproximadamente 15 minutos.

De antemano agradecemos su participación.

1

Consentimiento Electrónico. *

Seleccione una opción.

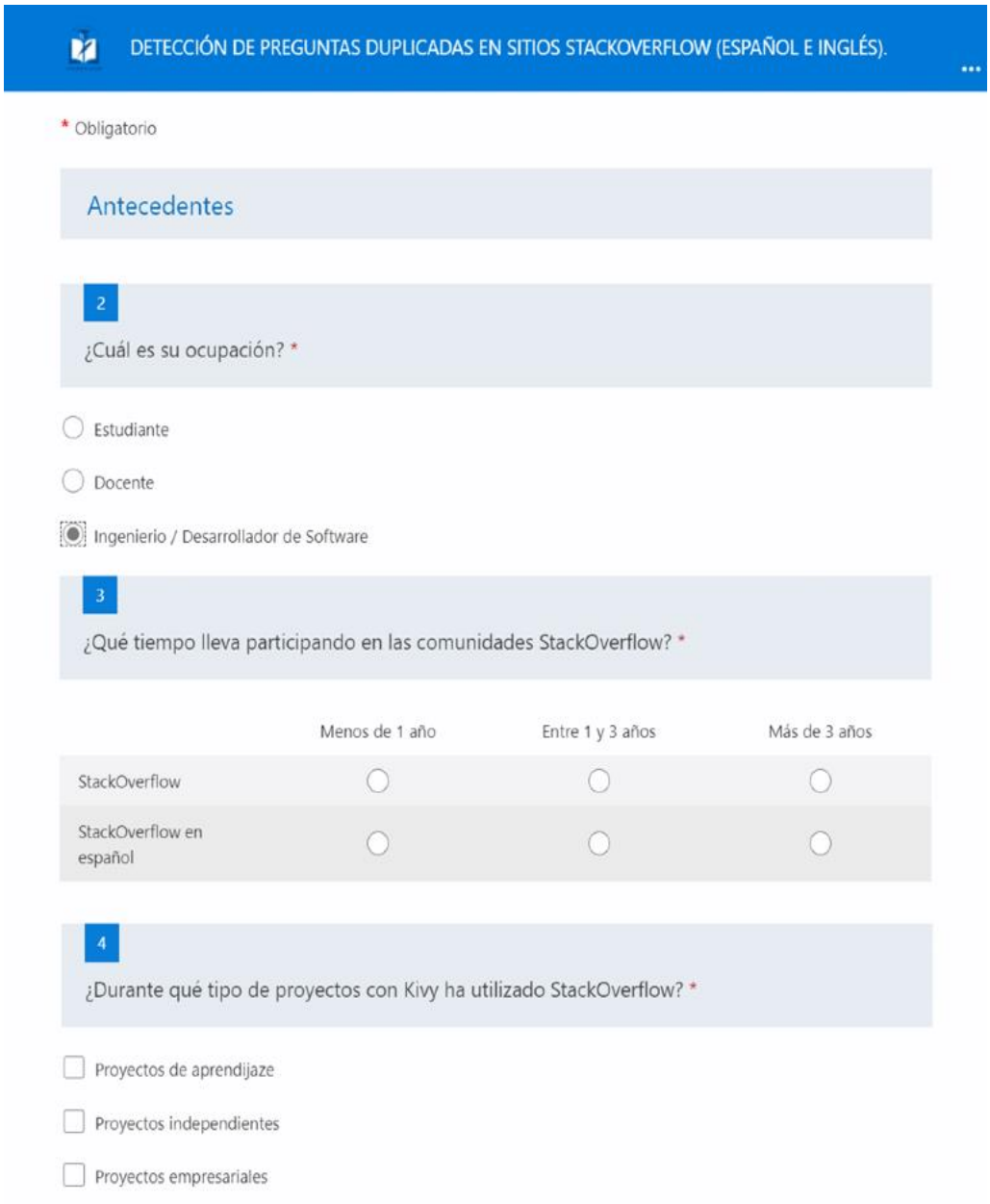
Si selecciona la opción "SI" indica que:
**Ha leído la información anterior y acepta participar voluntariamente.*
**Es mayor de 18 años.*

Si no desea participar en el estudio de investigación seleccione la opción "NO".

- ○ SI
- ○ NO

Siguiente

Página 1 de 14



DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

* Obligatorio

Antecedentes

2

¿Cuál es su ocupación? *

- ○ Estudiante
- ○ Docente
- ◉ Ingeniero / Desarrollador de Software

3

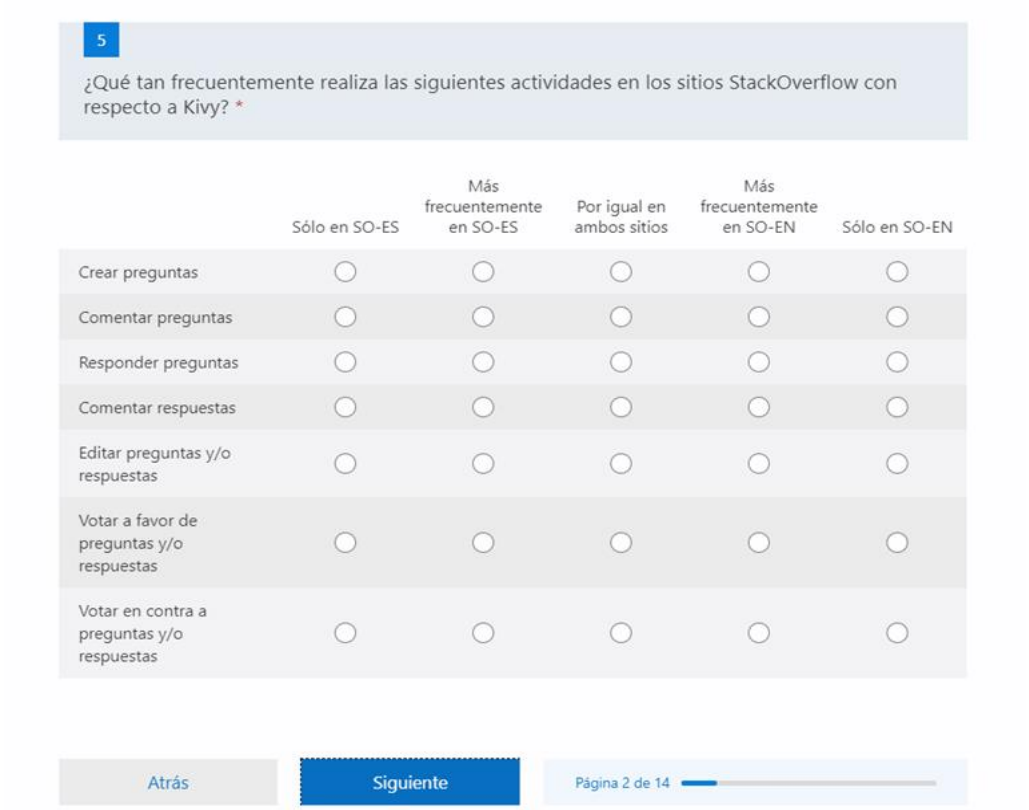
¿Qué tiempo lleva participando en las comunidades StackOverflow? *

	Menos de 1 año	Entre 1 y 3 años	Más de 3 años
StackOverflow	○	○	○
StackOverflow en español	○	○	○

4

¿Durante qué tipo de proyectos con Kivy ha utilizado StackOverflow? *

- ☐ Proyectos de aprendijaze
- ☐ Proyectos independientes
- ☐ Proyectos empresariales

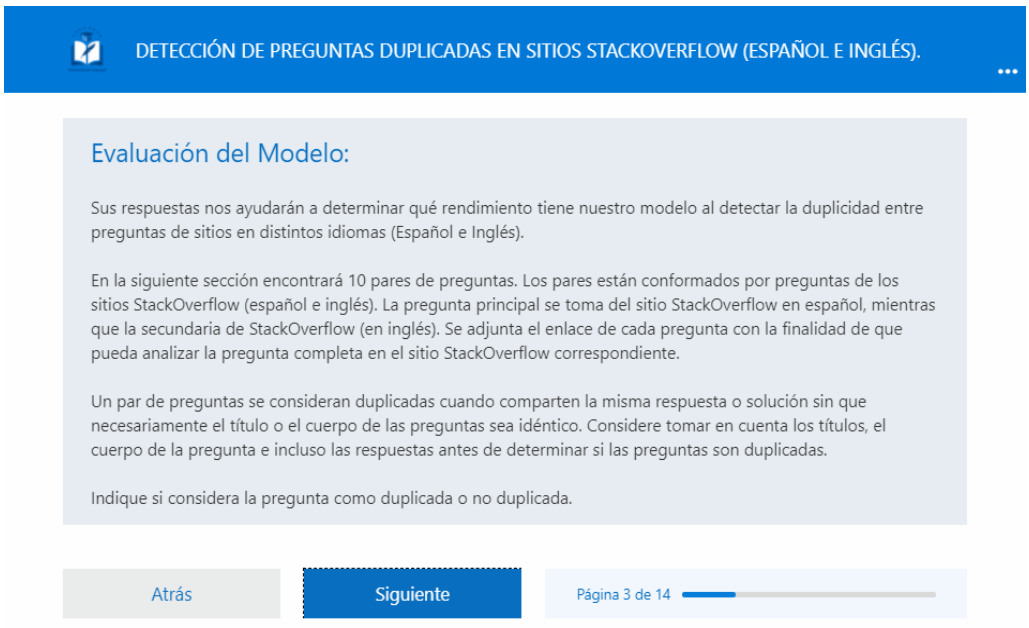


5

¿Qué tan frecuentemente realiza las siguientes actividades en los sitios StackOverflow con respecto a Kivy? *

	Sólo en SO-ES	Más frecuentemente en SO-ES	Por igual en ambos sitios	Más frecuentemente en SO-EN	Sólo en SO-EN
Crear preguntas	○	○	○	○	○
Comentar preguntas	○	○	○	○	○
Responder preguntas	○	○	○	○	○
Comentar respuestas	○	○	○	○	○
Editar preguntas y/o respuestas	○	○	○	○	○
Votar a favor de preguntas y/o respuestas	○	○	○	○	○
Votar en contra a preguntas y/o respuestas	○	○	○	○	○

Atrás Siguiente Página 2 de 14



DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

Evaluación del Modelo:

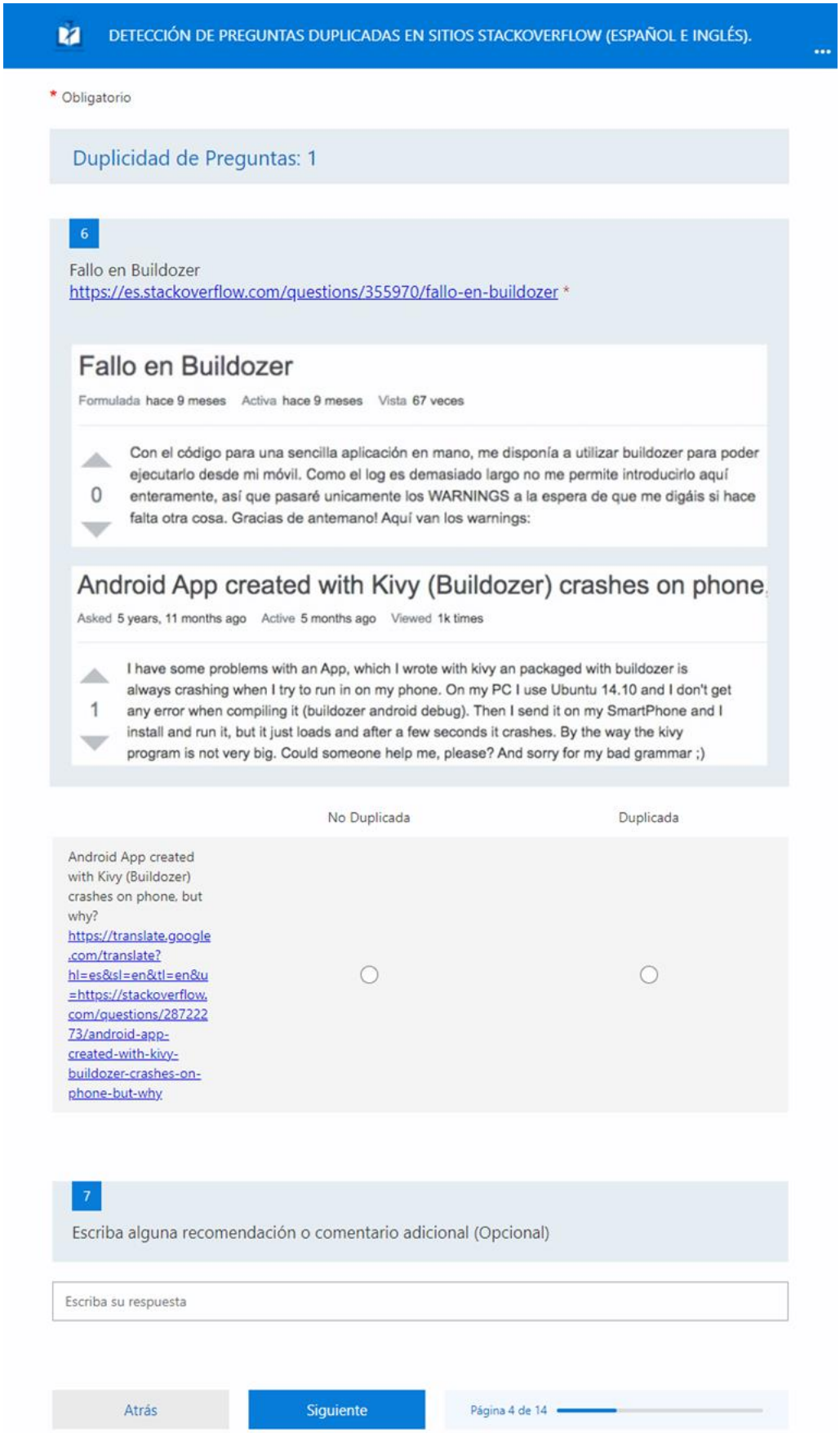
Sus respuestas nos ayudarán a determinar qué rendimiento tiene nuestro modelo al detectar la duplicidad entre preguntas de sitios en distintos idiomas (Español e Inglés).

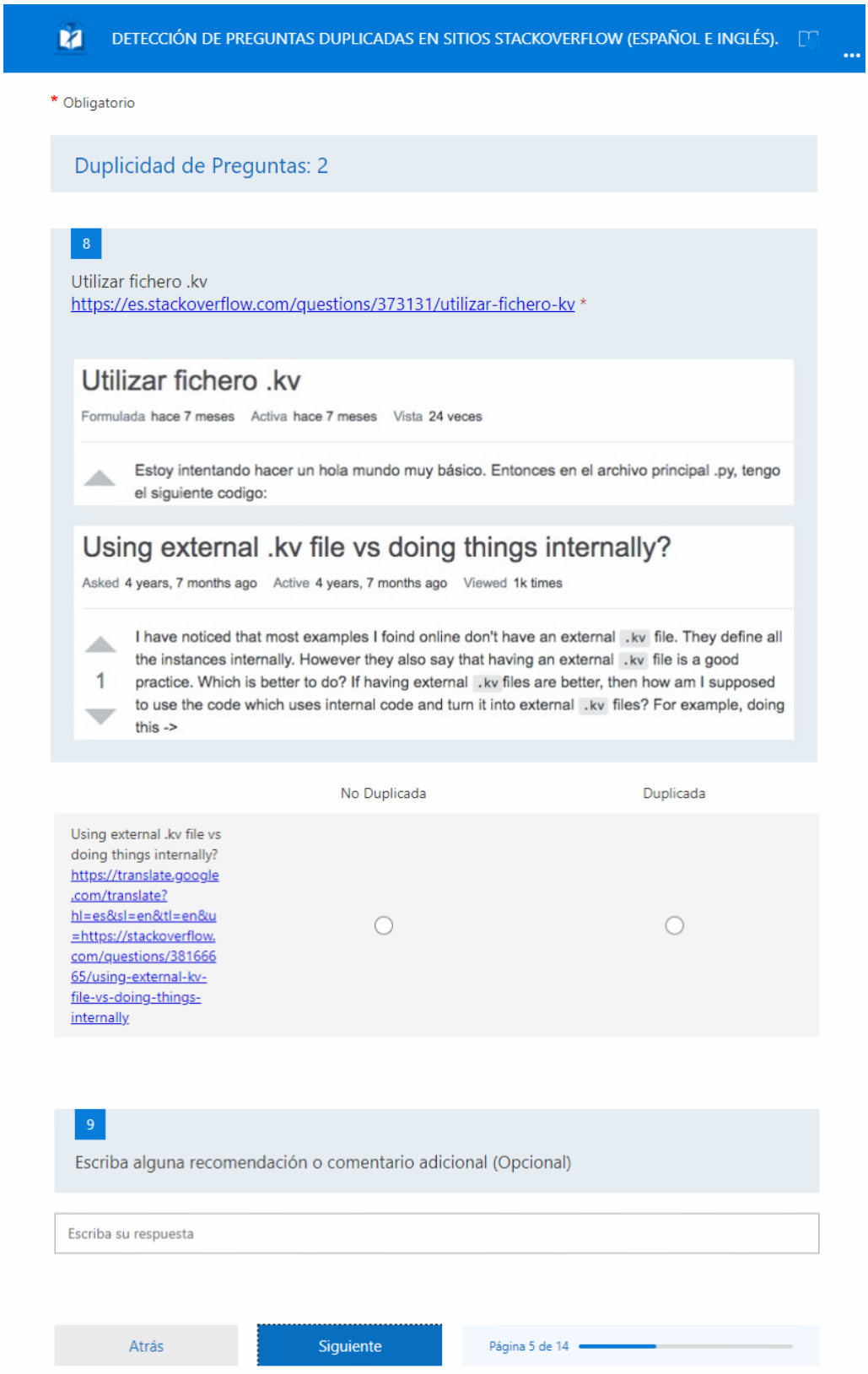
En la siguiente sección encontrará 10 pares de preguntas. Los pares están conformados por preguntas de los sitios StackOverflow (español e inglés). La pregunta principal se toma del sitio StackOverflow en español, mientras que la secundaria de StackOverflow (en inglés). Se adjunta el enlace de cada pregunta con la finalidad de que pueda analizar la pregunta completa en el sitio StackOverflow correspondiente.

Un par de preguntas se consideran duplicadas cuando comparten la misma respuesta o solución sin que necesariamente el título o el cuerpo de las preguntas sea idéntico. Considere tomar en cuenta los títulos, el cuerpo de la pregunta e incluso las respuestas antes de determinar si las preguntas son duplicadas.

Indique si considera la pregunta como duplicada o no duplicada.

Atrás Siguiente Página 3 de 14


DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).
* Obligatorio
Duplicidad de Preguntas: 1
6
Fallo en Buildozer
https://es.stackoverflow.com/questions/355970/fallo-en-buildozer *
Fallo en Buildozer
Formulada hace 9 meses Activa hace 9 meses Vista 67 veces
0
Con el código para una sencilla aplicación en mano, me disponía a utilizar buildozer para poder ejecutarlo desde mi móvil. Como el log es demasiado largo no me permite introducirlo aquí enteramente, así que pasaré unicamente los WARNINGS a la espera de que me digáis si hace falta otra cosa. Gracias de antemano! Aquí van los warnings:
Android App created with Kivy (Buildozer) crashes on phone
Asked 5 years, 11 months ago Active 5 months ago Viewed 1k times
1
I have some problems with an App, which I wrote with kivy an packaged with buildozer is always crashing when I try to run in on my phone. On my PC I use Ubuntu 14.10 and I don't get any error when compiling it (buildozer android debug). Then I send it on my SmartPhone and I install and run it, but it just loads and after a few seconds it crashes. By the way the kivy program is not very big. Could someone help me, please? And sorry for my bad grammar ;)
No Duplicada
Duplicada
Android App created with Kivy (Buildozer) crashes on phone, but why? https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/28722273/android-app-created-with-kivy-buildozer-crashes-on-phone-but-why
7
Escriba alguna recomendación o comentario adicional (Opcional)
Escriba su respuesta
Atrás
Siguiente
Página 4 de 14


DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).
* Obligatorio
Duplicidad de Preguntas: 2
8
Utilizar fichero .kv
https://es.stackoverflow.com/questions/373131/utilizar-fichero-kv *
Utilizar fichero .kv
Formulada hace 7 meses Activa hace 7 meses Vista 24 veces
Estoy intentando hacer un hola mundo muy básico. Entonces en el archivo principal .py, tengo el siguiente codigo:
Using external .kv file vs doing things internally?
Asked 4 years, 7 months ago Active 4 years, 7 months ago Viewed 1k times
1
I have noticed that most examples I foind online don't have an external .kv file. They define all the instances internally. However they also say that having an external .kv file is a good practice. Which is better to do? If having external .kv files are better, then how am I supposed to use the code which uses internal code and turn it into external .kv files? For example, doing this ->
No Duplicada
Duplicada
Using external .kv file vs doing things internally? https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/38166665/using-external-kv-file-vs-doing-things-internally
9
Escriba alguna recomendación o comentario adicional (Opcional)
Escriba su respuesta
Atrás
Siguiente
Página 5 de 14

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

* Obligatorio

Duplicidad de Preguntas: 3

10

Error al usar la Camara en Android con kivy
https://es.stackoverflow.com/questions/370009/error-al-usar-la-camara-en-android-con-kivy *

Error al usar la Camara en Android con kivy

Formulada hace 7 meses Activa hace 7 meses Vista 77 veces

0

Version python: 3.6.9

Descripción del problema: Estoy usando la libreria plyer de python para acceder a las API de android en una app kivy, la de la cámara en este caso. Se genera el .apk lo más bien pero cuando ejecuto la aplicación, se cierra de golpe la app. Depurando con el comando adb logcat -s "python", obtengo lo siguiente:

python kivy with plyer app crashes on android (camera)

Asked 4 years, 1 month ago Active 4 years, 1 month ago Viewed 1k times

3

I develop kivy application using plyer. Build by buildozer and starting app on android succeeded, but application crashes when I push the button starting a camera, and nothing is output in logcat. I work without a problem when I carry out the same code in kivylauncher. It was similar even if I tested it with an accelerometer.

I think that necessary setting may be short when I build application using plyer.

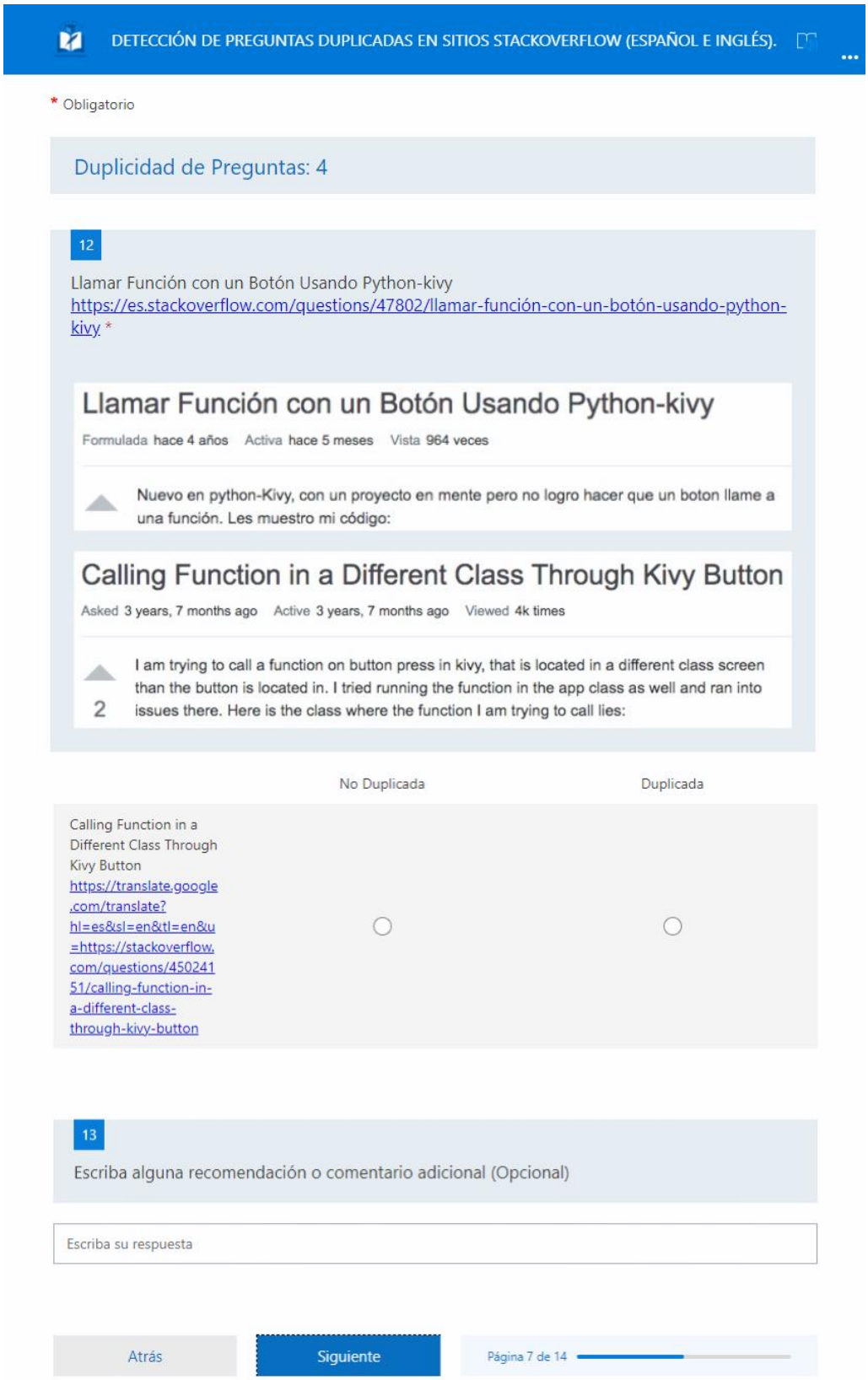
	No Duplicada	Duplicada
python kivy with plyer app crashes on android (camera) https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/41156449/python-kivy-with-plyer-app-crashes-on-android-camera	○	○

11

Escriba alguna recomendación o comentario adicional (Opcional)

Escriba su respuesta

Atrás Siguiente Página 6 de 14


DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).
* Obligatorio
Duplicidad de Preguntas: 4
12
Llamar Función con un Botón Usando Python-kivy
https://es.stackoverflow.com/questions/47802/llamar-función-con-un-botón-usando-python-kivy *
Llamar Función con un Botón Usando Python-kivy
Formulada hace 4 años Activa hace 5 meses Vista 964 veces
Nuevo en python-Kivy, con un proyecto en mente pero no logro hacer que un boton llame a una función. Les muestro mi código:
Calling Function in a Different Class Through Kivy Button
Asked 3 years, 7 months ago Active 3 years, 7 months ago Viewed 4k times
2
I am trying to call a function on button press in kivy, that is located in a different class screen than the button is located in. I tried running the function in the app class as well and ran into issues there. Here is the class where the function I am trying to call lies:
No Duplicada
Duplicada
Calling Function in a Different Class Through Kivy Button https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/45024151/calling-function-in-a-different-class-through-kivy-button
13
Escriba alguna recomendación o comentario adicional (Opcional)
Escriba su respuesta
Atrás
Siguiente
Página 7 de 14

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

* Obligatorio

Duplicidad de Preguntas: 5

14

Poner botón para ejecutar la camera
https://es.stackoverflow.com/questions/353918/poner-botón-para-ejecutar-la-camera *

Poner botón para ejecutar la camera

Formulada hace 9 meses Activa hace 9 meses Vista 41 veces

2

Amigos tengo un código en python, resulta me ayudaron a escribirlo soy nuevo en python, y más en kivy, tengo un pequeño dilema y no sé cómo hacer tengo un código que apenas se ejecuta, se abre la cámara web pero no quiero que se ejecute apenas ejecute el codigo, lo que me sirve para entender mejor es poner un botón que cuando le dé click, si me active la cámara espero me puedan ayudar gracias amigos, este es el código que tengo:

Python Kivy: how to call a function on button click?

Asked 3 years, 4 months ago Active 8 months ago Viewed 16k times

6

i'm pretty new at using kivy library.

I have an app.py file and an app.kv file , my problem is that I can't call a function on button press.

	No Duplicada	Duplicada
Python Kivy: how to call a function on button click? https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/46351997/python-kivy-how-to-call-a-function-on-button-click	○	○

15

Escriba alguna recomendación o comentario adicional (Opcional)

Escriba su respuesta

Atrás Siguiente Página 8 de 14

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

* Obligatorio

Duplicidad de Preguntas: 6

16

TypeError: 'NoneType' object is not callable
https://es.stackoverflow.com/questions/357088/typeerror-nonetype-object-is-not-callable *

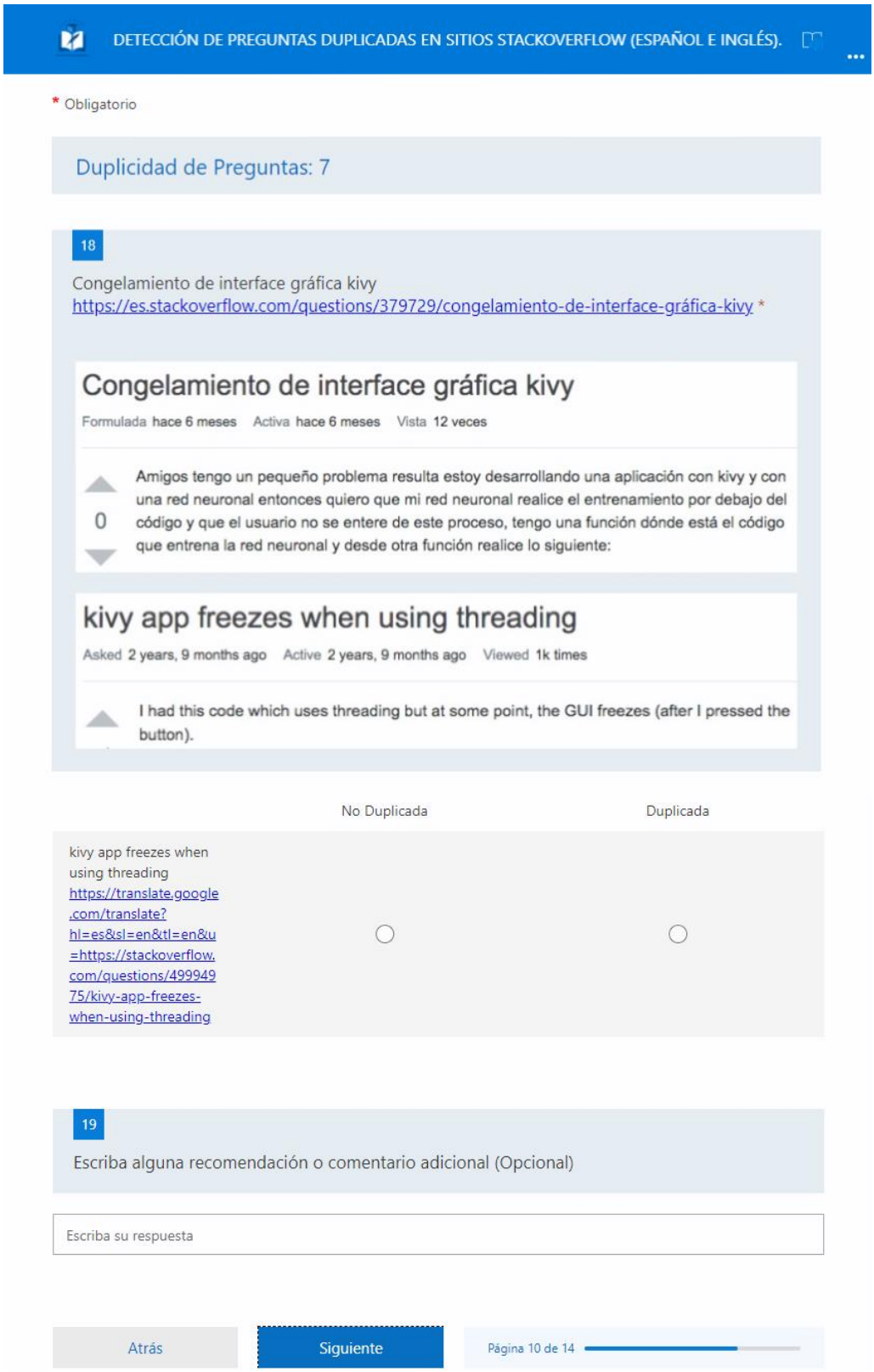
	No Duplicada	Duplicada
Kivy popup Filechooser pass variable (selection) https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/48386125/kivy-popup-filechooser-pass-variable-selection	○	○

17

Escriba alguna recomendación o comentario adicional (Opcional)

Escriba su respuesta

Atrás | Siguiente | Página 9 de 14


DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).
* Obligatorio
Duplicidad de Preguntas: 7
18
Congelamiento de interface gráfica kivy
https://es.stackoverflow.com/questions/379729/congelamiento-de-interface-gráfica-kivy *
Congelamiento de interface gráfica kivy
Formulada hace 6 meses Activa hace 6 meses Vista 12 veces
0
Amigos tengo un pequeño problema resulta estoy desarrollando una aplicación con kivy y con una red neuronal entonces quiero que mi red neuronal realice el entrenamiento por debajo del código y que el usuario no se entere de este proceso, tengo una función dónde está el código que entrena la red neuronal y desde otra función realice lo siguiente:
kivy app freezes when using threading
Asked 2 years, 9 months ago Active 2 years, 9 months ago Viewed 1k times
I had this code which uses threading but at some point, the GUI freezes (after I pressed the button).
No Duplicada
Duplicada
kivy app freezes when using threading https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/49994975/kivy-app-freezes-when-using-threading
19
Escriba alguna recomendación o comentario adicional (Opcional)
Escriba su respuesta
Atrás
Siguiente
Página 10 de 14

DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).

* Obligatorio

Duplicidad de Preguntas: 8

20

Enviar valor entre ventanas en Kivy
https://es.stackoverflow.com/questions/354296/enviar-valor-entre-ventanas-en-kivy *

Enviar valor entre ventanas en Kivy

Formulada hace 9 meses Activa hace 9 meses Vista 71 veces

1

Tengo una ScreenView que contiene una serie de botones generados automáticamente de una lista. Quiero que al pulsar un botón determinado se cambie la ventana actual por otra Screen, y que en esta se muestre el número del boton que le envio. El objetivo de esta ventana va a ser realmente que tras haber pulsado un botón quede regristrado el id pulsado para posteriormente emplear este valor para buscar datos pero con saber pasar el número me valdría. Lo que tengo hasta ahora es lo siguiente:

Pass values between classes (screens) in kivy

Asked 2 years, 7 months ago Active 6 months ago Viewed 2k times

1

I know that generally similar questions have been answered before. I have read them all of them. I have tried them all. Nothing seems to work for my particular case.

I am working in kivy with python 3. Not sure if that is the reason (maybe prior answers work only on python 2?).

I simply want to pass the text input from screen1_textinput (screen1_textinput.text) and the text input from screen1_textinput2 (screen1_textinput2.text), [the last one being the input from the slider of screen 1] into the text input of screen2_textinput (screen2_textinput.text).

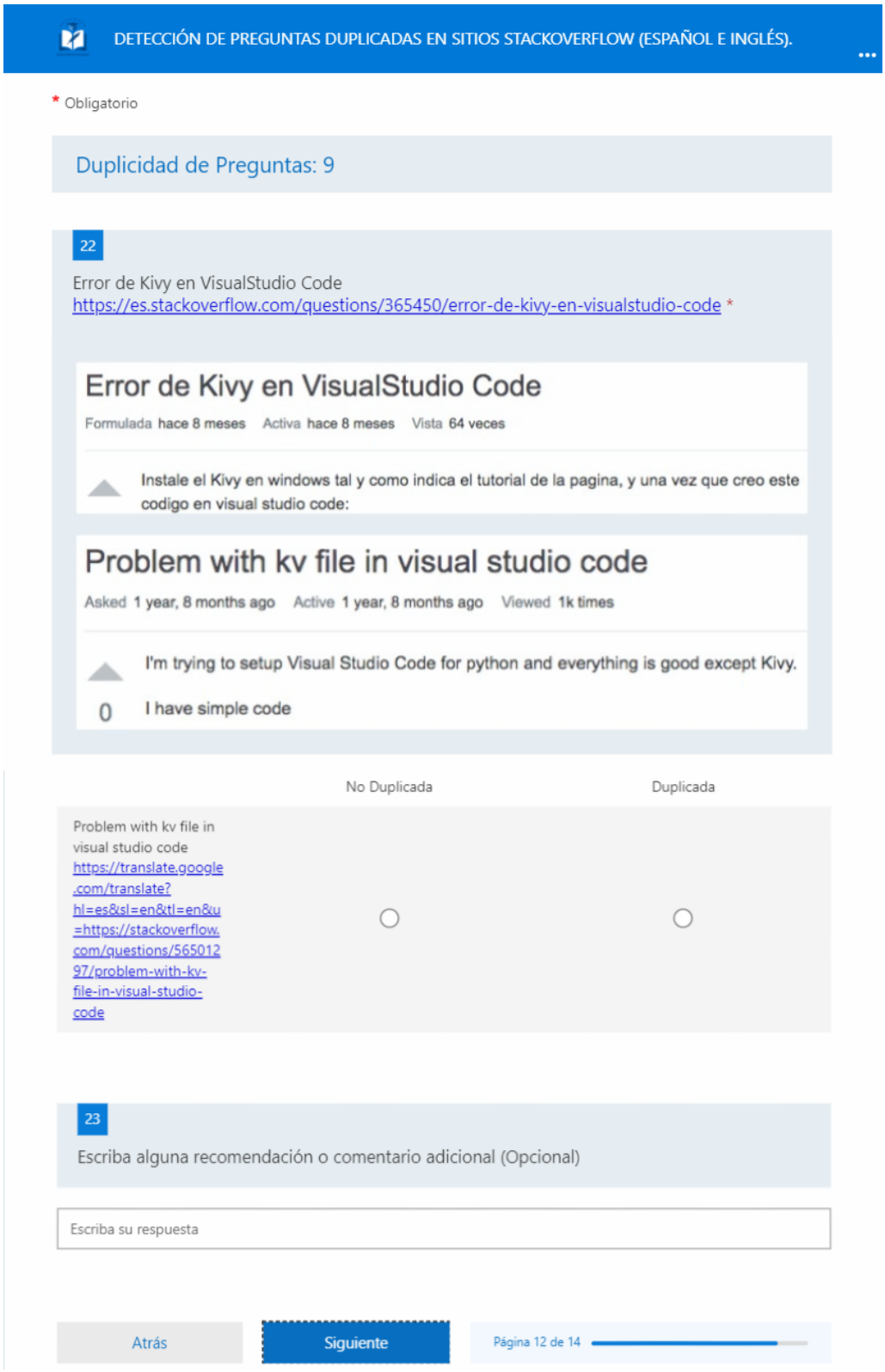
	No Duplicada	Duplicada
Pass values between classes (screens) in kivy https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/50991371/pass-values-between-classes-screens-in-kivy	○	○

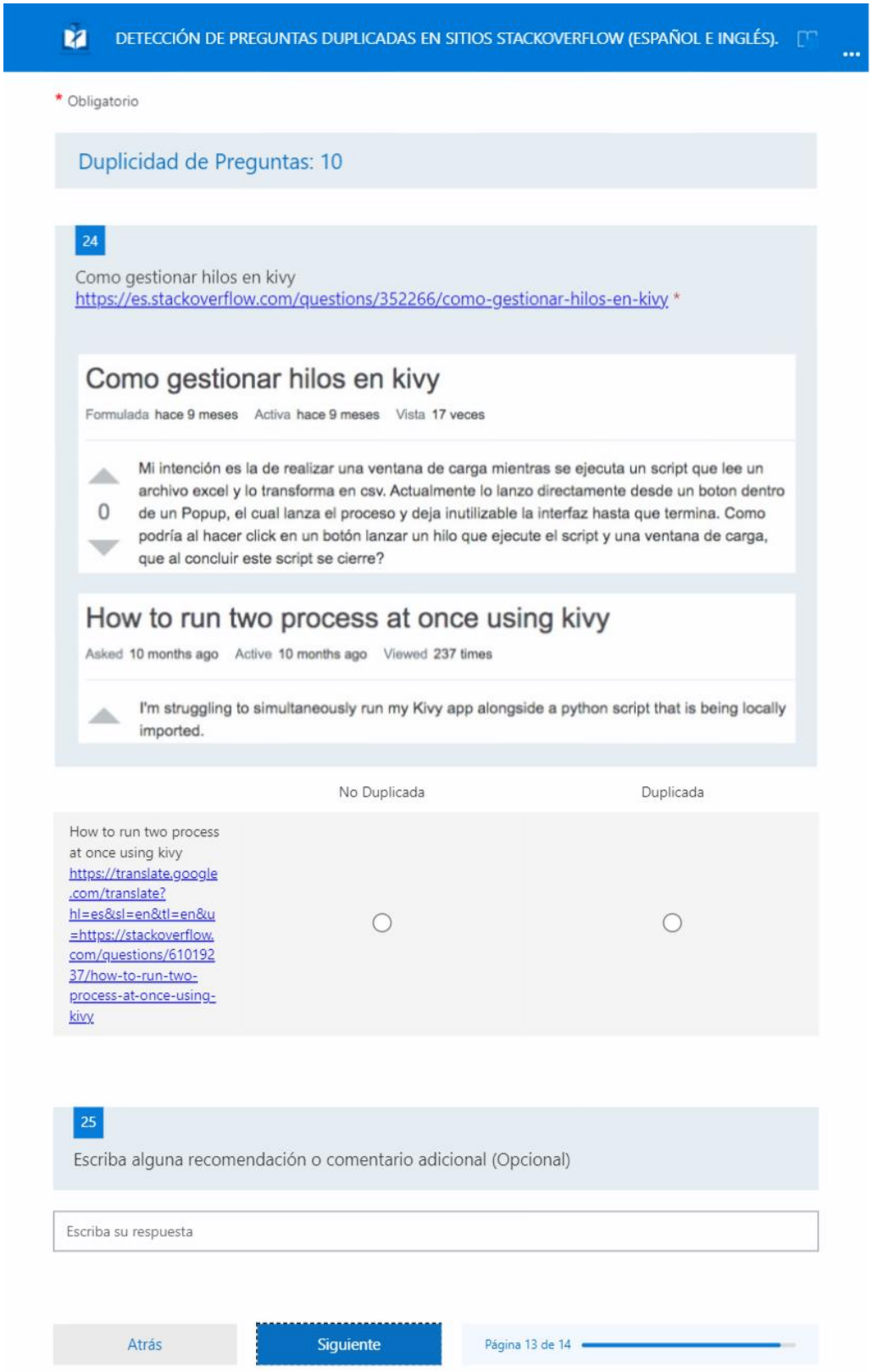
21

Escriba alguna recomendación o comentario adicional (Opcional)

Escriba su respuesta

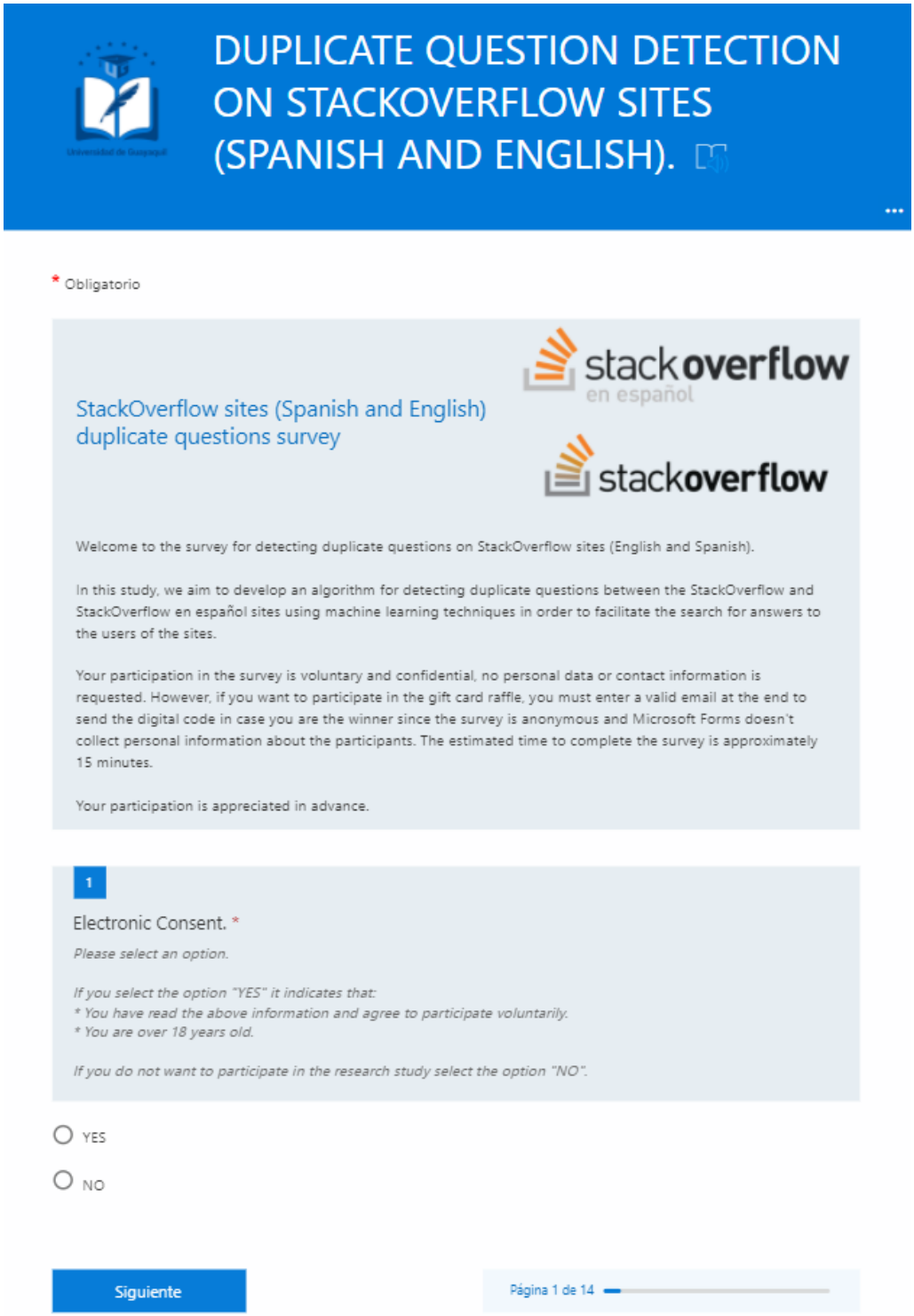
Atrás Siguiente Página 11 de 14


DETECCIÓN DE PREGUNTAS DUPLICADAS EN SITIOS STACKOVERFLOW (ESPAÑOL E INGLÉS).
* Obligatorio
Duplicidad de Preguntas: 9
22
Error de Kivy en VisualStudio Code
https://es.stackoverflow.com/questions/365450/error-de-kivy-en-visualstudio-code *
Error de Kivy en VisualStudio Code
Formulada hace 8 meses Activa hace 8 meses Vista 64 veces
Instale el Kivy en windows tal y como indica el tutorial de la pagina, y una vez que creo este codigo en visual studio code:
Problem with kv file in visual studio code
Asked 1 year, 8 months ago Active 1 year, 8 months ago Viewed 1k times
I'm trying to setup Visual Studio Code for python and everything is good except Kivy.
0
I have simple code
No Duplicada
Duplicada
Problem with kv file in visual studio code https://translate.google.com/translate?hl=es&sl=en&tl=en&u=https://stackoverflow.com/questions/56501297/problem-with-kv-file-in-visual-studio-code
23
Escriba alguna recomendación o comentario adicional (Opcional)
Escriba su respuesta
Atrás
Siguiente
Página 12 de 14



Elaboración: Investigadores.
Fuente: Propia.

Formato de Encuesta en inglés



DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).

* Obligatorio

StackOverflow sites (Spanish and English) duplicate questions survey

Welcome to the survey for detecting duplicate questions on StackOverflow sites (English and Spanish).

In this study, we aim to develop an algorithm for detecting duplicate questions between the StackOverflow and StackOverflow en español sites using machine learning techniques in order to facilitate the search for answers to the users of the sites.

Your participation in the survey is voluntary and confidential, no personal data or contact information is requested. However, if you want to participate in the gift card raffle, you must enter a valid email at the end to send the digital code in case you are the winner since the survey is anonymous and Microsoft Forms doesn't collect personal information about the participants. The estimated time to complete the survey is approximately 15 minutes.

Your participation is appreciated in advance.

1

Electronic Consent. *

Please select an option.

If you select the option "YES" it indicates that:
** You have read the above information and agree to participate voluntarily.*
** You are over 18 years old.*

If you do not want to participate in the research study select the option "NO".

YES

NO

Siguiente

Página 1 de 14

DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).

* Obligatorio

Background

2

What is your occupation? *

- ○ Student
- ○ Academic
- ○ Software Developer / Engineer

3

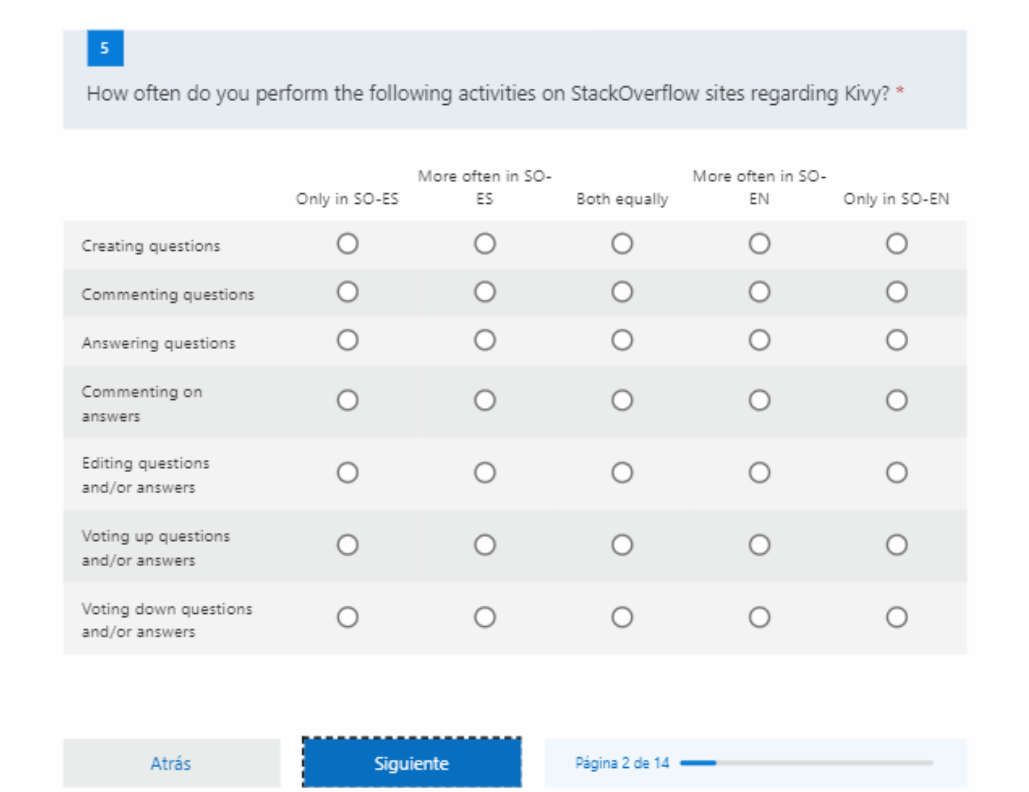
How long have you been participating in StackOverflow communities? *

	Less than 1 year	Between 1 and 3 years	More than 3 years
StackOverflow	○	○	○
StackOverflow en español	○	○	○

4

During which kind of projects using Kivy have you used StackOverflow? *

- ☐ Learning projects
- ☐ Freelance projects
- ☐ Business projects

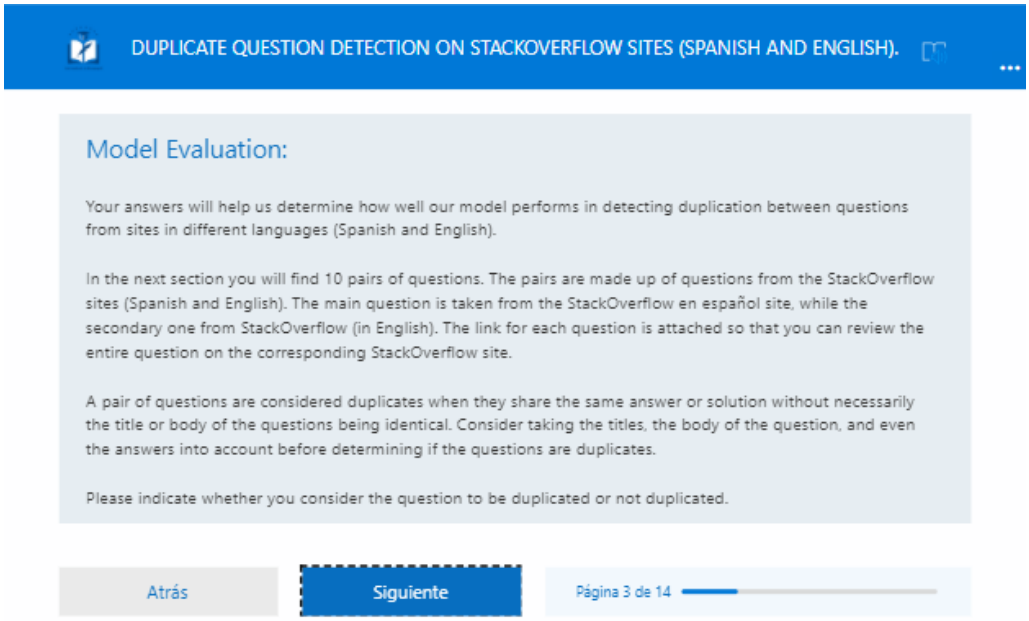


5

How often do you perform the following activities on StackOverflow sites regarding Kivy? *

	Only in SO-ES	More often in SO-ES	Both equally	More often in SO-EN	Only in SO-EN
Creating questions	○	○	○	○	○
Commenting questions	○	○	○	○	○
Answering questions	○	○	○	○	○
Commenting on answers	○	○	○	○	○
Editing questions and/or answers	○	○	○	○	○
Voting up questions and/or answers	○	○	○	○	○
Voting down questions and/or answers	○	○	○	○	○

Atrás Siguiente Página 2 de 14



DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).

Model Evaluation:

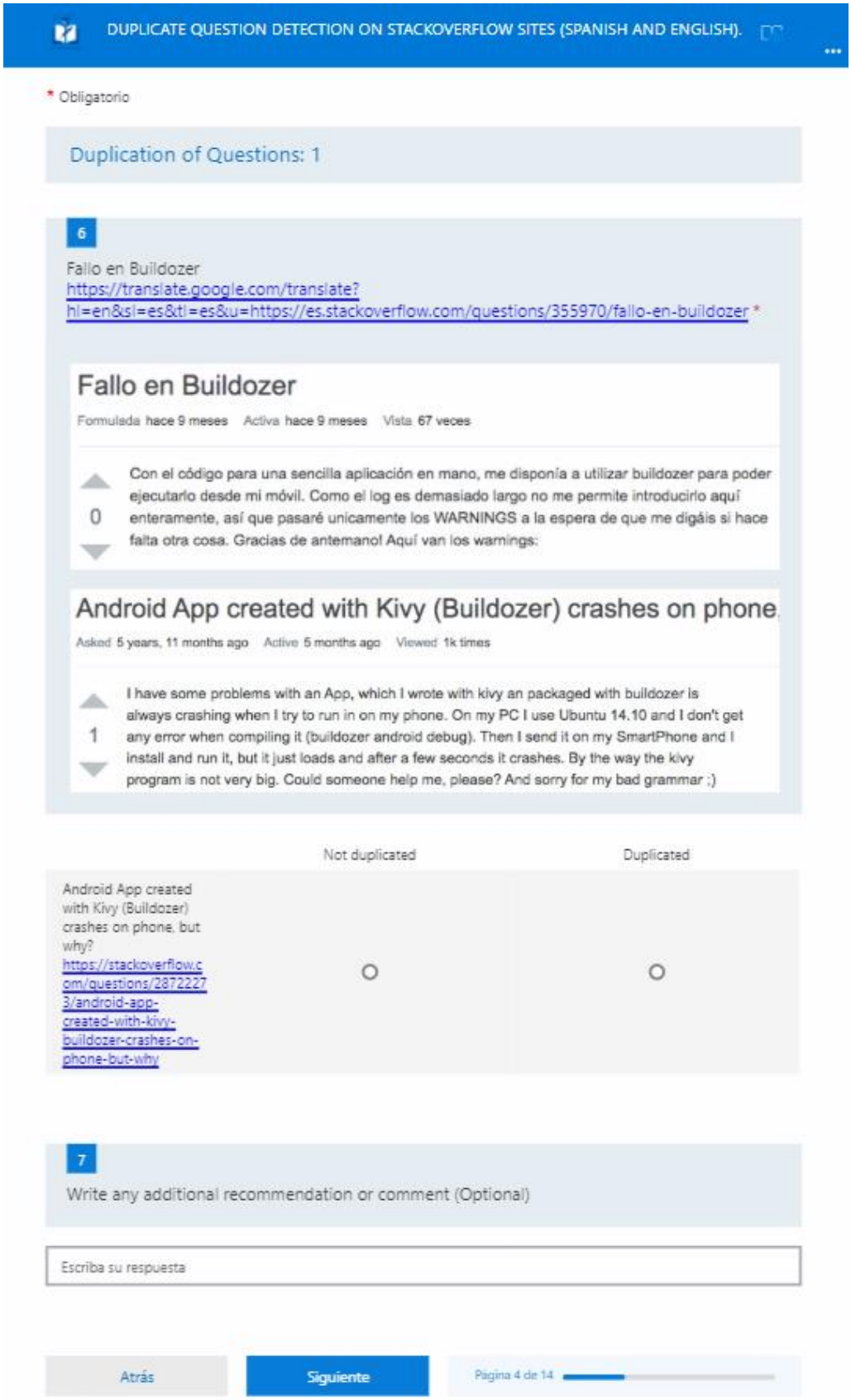
Your answers will help us determine how well our model performs in detecting duplication between questions from sites in different languages (Spanish and English).

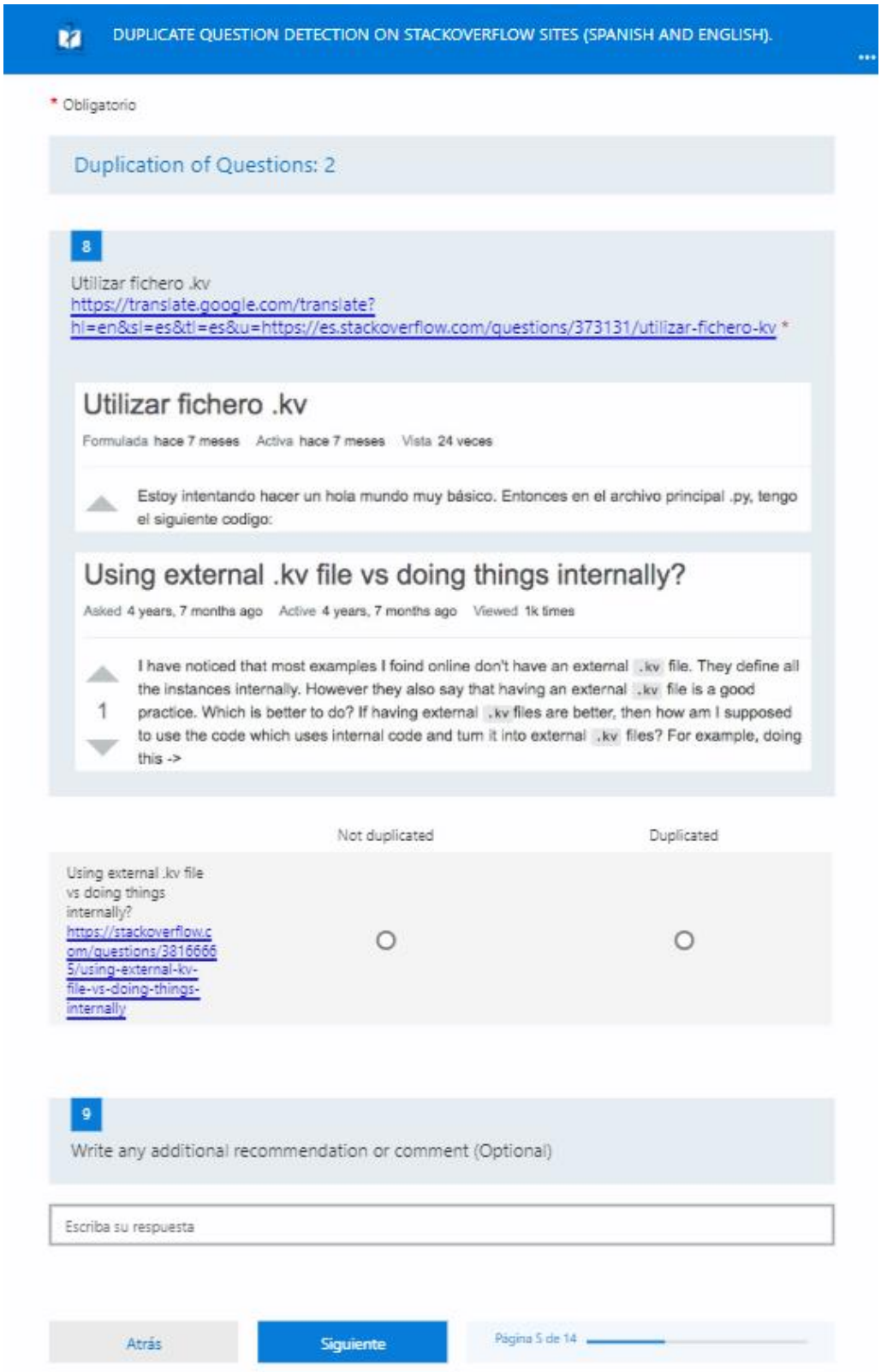
In the next section you will find 10 pairs of questions. The pairs are made up of questions from the StackOverflow sites (Spanish and English). The main question is taken from the StackOverflow en español site, while the secondary one from StackOverflow (in English). The link for each question is attached so that you can review the entire question on the corresponding StackOverflow site.

A pair of questions are considered duplicates when they share the same answer or solution without necessarily the title or body of the questions being identical. Consider taking the titles, the body of the question, and even the answers into account before determining if the questions are duplicates.

Please indicate whether you consider the question to be duplicated or not duplicated.

Atrás Siguiente Página 3 de 14


DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 1
6
Fallo en Buildozer
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/355970/fallo-en-buildozer *
Fallo en Buildozer
Formulada hace 9 meses Activa hace 9 meses Vista 67 veces
0
Con el código para una sencilla aplicación en mano, me disponía a utilizar buildozer para poder ejecutarlo desde mi móvil. Como el log es demasiado largo no me permite introducirlo aquí enteramente, así que pasaré unicamente los WARNINGS a la espera de que me digáis si hace falta otra cosa. Gracias de antemano! Aquí van los warnings:
Android App created with Kivy (Buildozer) crashes on phone
Asked 5 years, 11 months ago Active 5 months ago Viewed 1k times
1
I have some problems with an App, which I wrote with kivy an packaged with buildozer is always crashing when I try to run in on my phone. On my PC I use Ubuntu 14.10 and I don't get any error when compiling it (buildozer android debug). Then I send it on my SmartPhone and I install and run it, but it just loads and after a few seconds it crashes. By the way the kivy program is not very big. Could someone help me, please? And sorry for my bad grammar ;)
Not duplicated
Duplicated
Android App created with Kivy (Buildozer) crashes on phone, but why? https://stackoverflow.com/questions/28722273/android-app-created-with-kivy-buildozer-crashes-on-phone-but-why
7
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 4 de 14



DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).

* Obligatorio

Duplication of Questions: 2

8

Utilizar fichero .kv
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/373131/utilizar-fichero-kv *

Utilizar fichero .kv

Formulada hace 7 meses Activa hace 7 meses Vista 24 veces

Estoy intentando hacer un hola mundo muy básico. Entonces en el archivo principal .py, tengo el siguiente codigo:

Using external .kv file vs doing things internally?

Asked 4 years, 7 months ago Active 4 years, 7 months ago Viewed 1k times

1

I have noticed that most examples I foind online don't have an external .kv file. They define all the instances internally. However they also say that having an external .kv file is a good practice. Which is better to do? If having external .kv files are better, then how am I supposed to use the code which uses internal code and turn it into external .kv files? For example, doing this ->

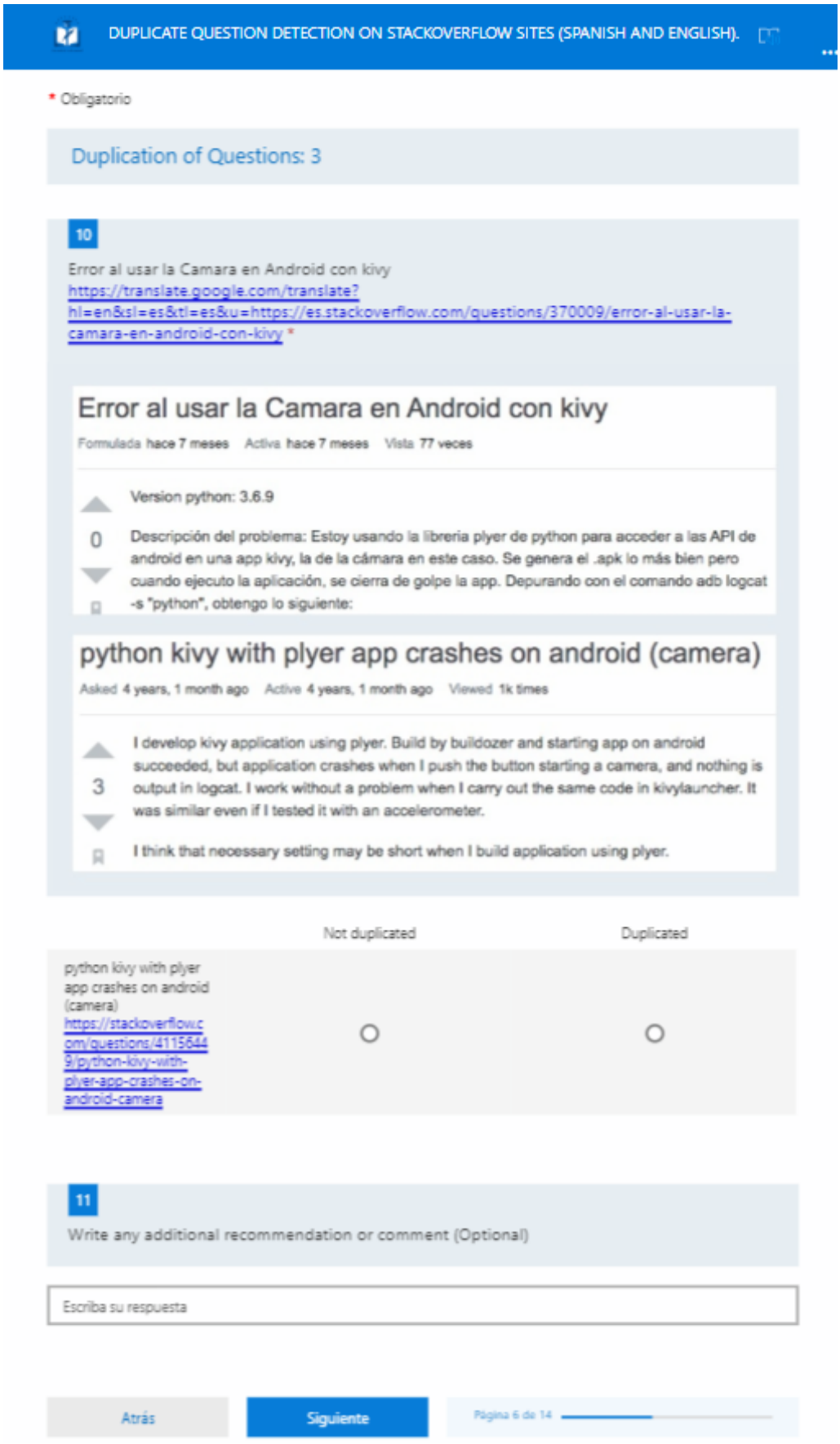
	Not duplicated	Duplicated
Using external .kv file vs doing things internally? https://stackoverflow.com/questions/38166665/using-external-kv-file-vs-doing-things-internally	○	○

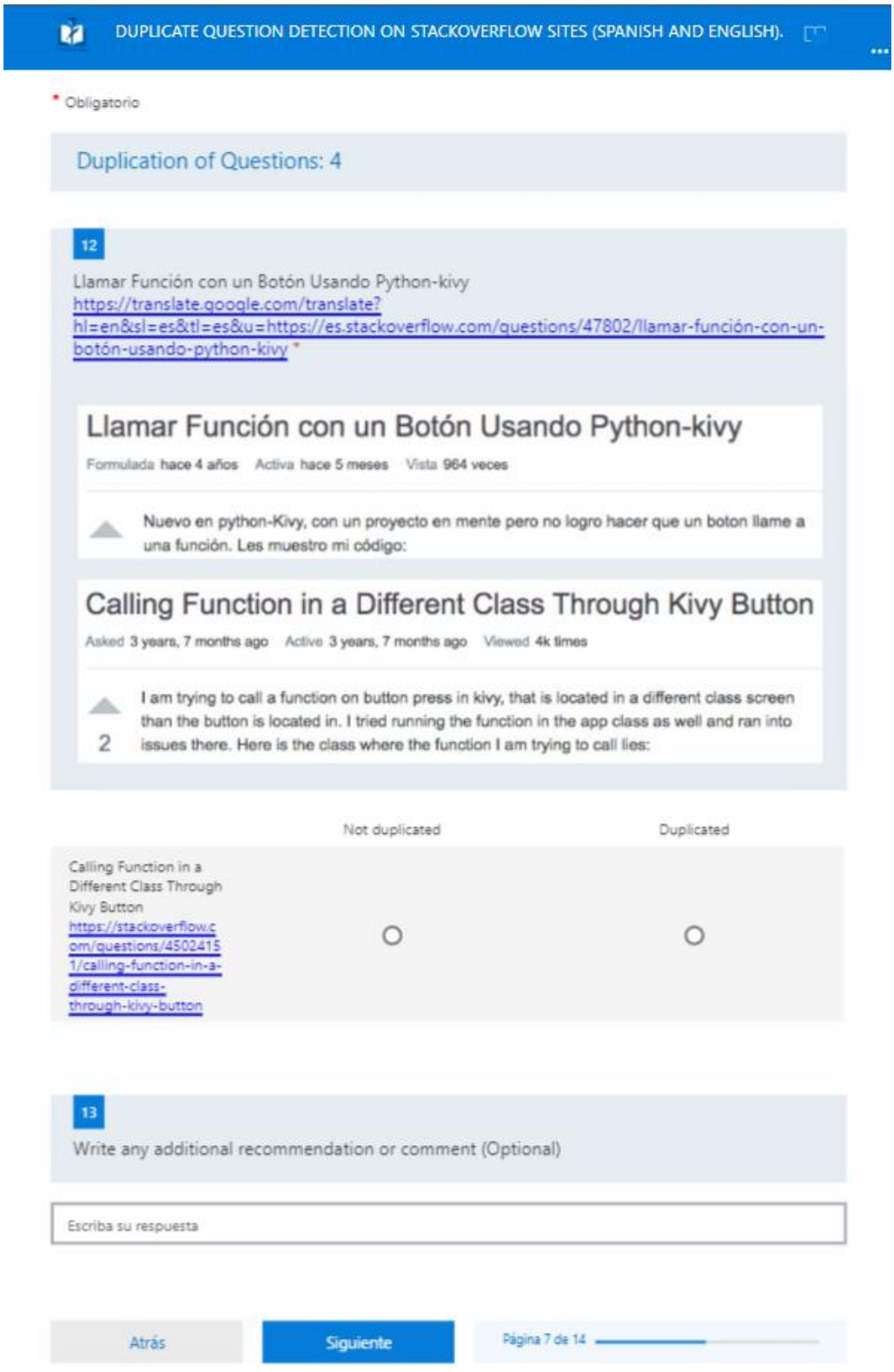
9

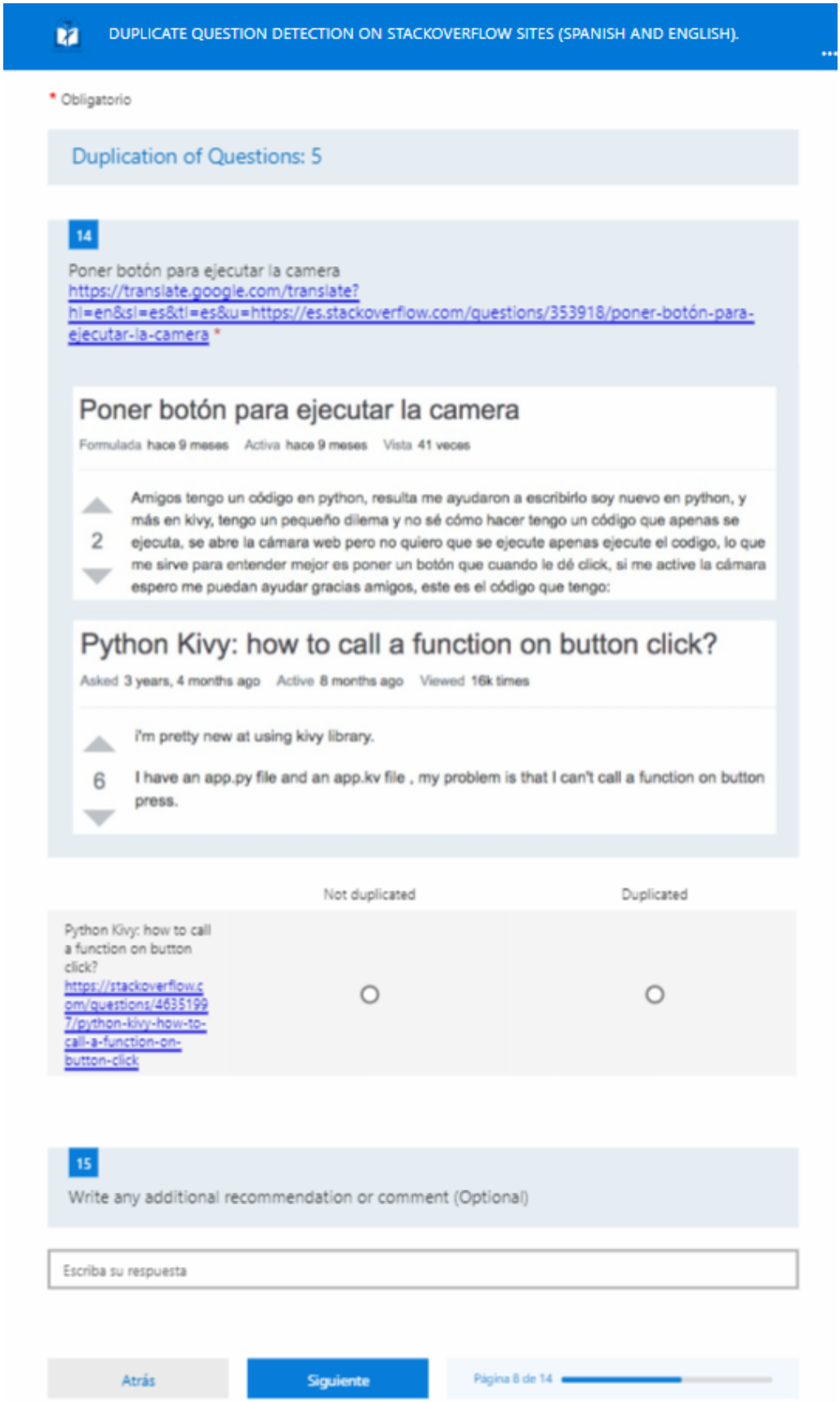
Write any additional recommendation or comment (Optional)

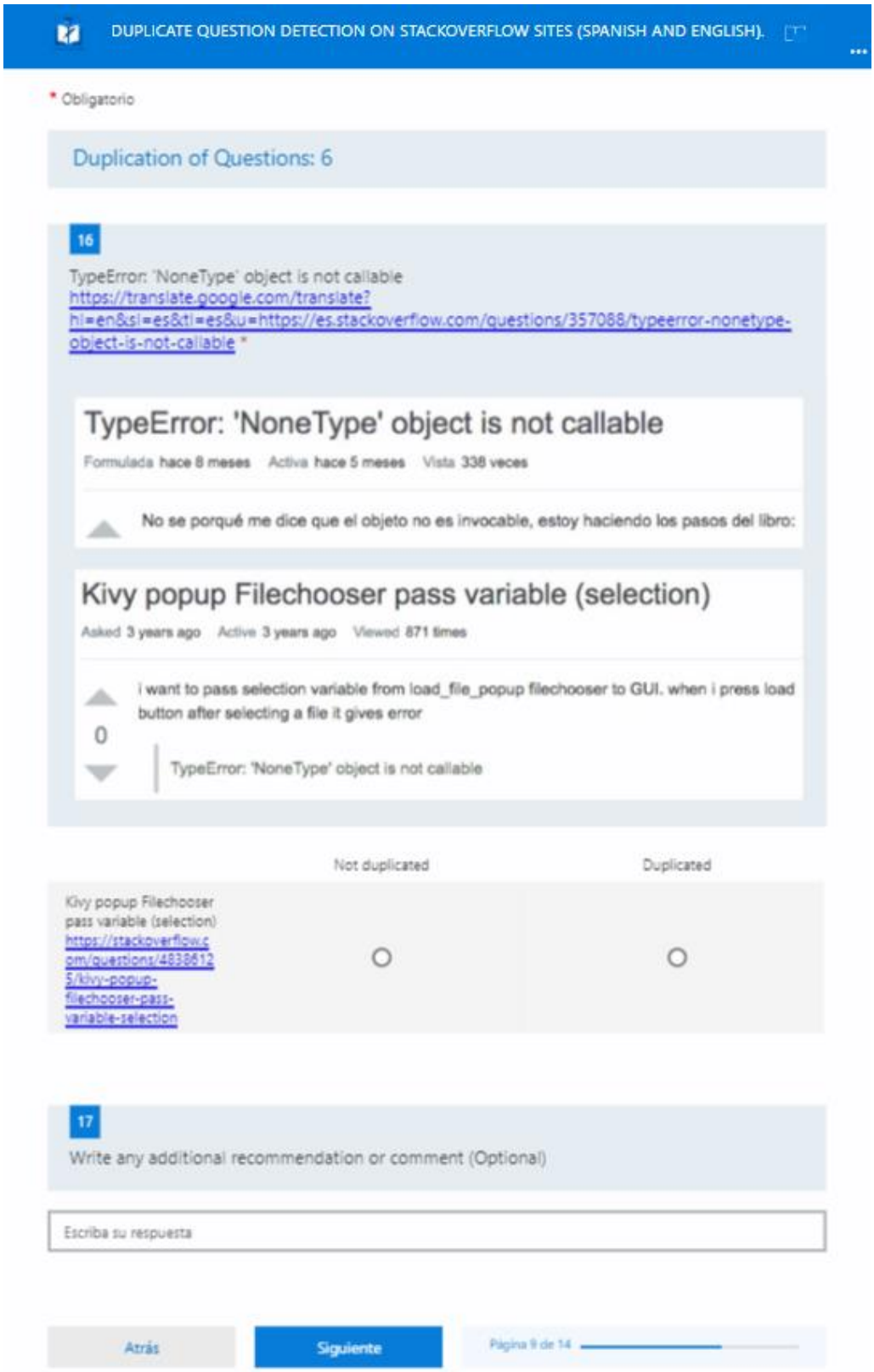
Escriba su respuesta

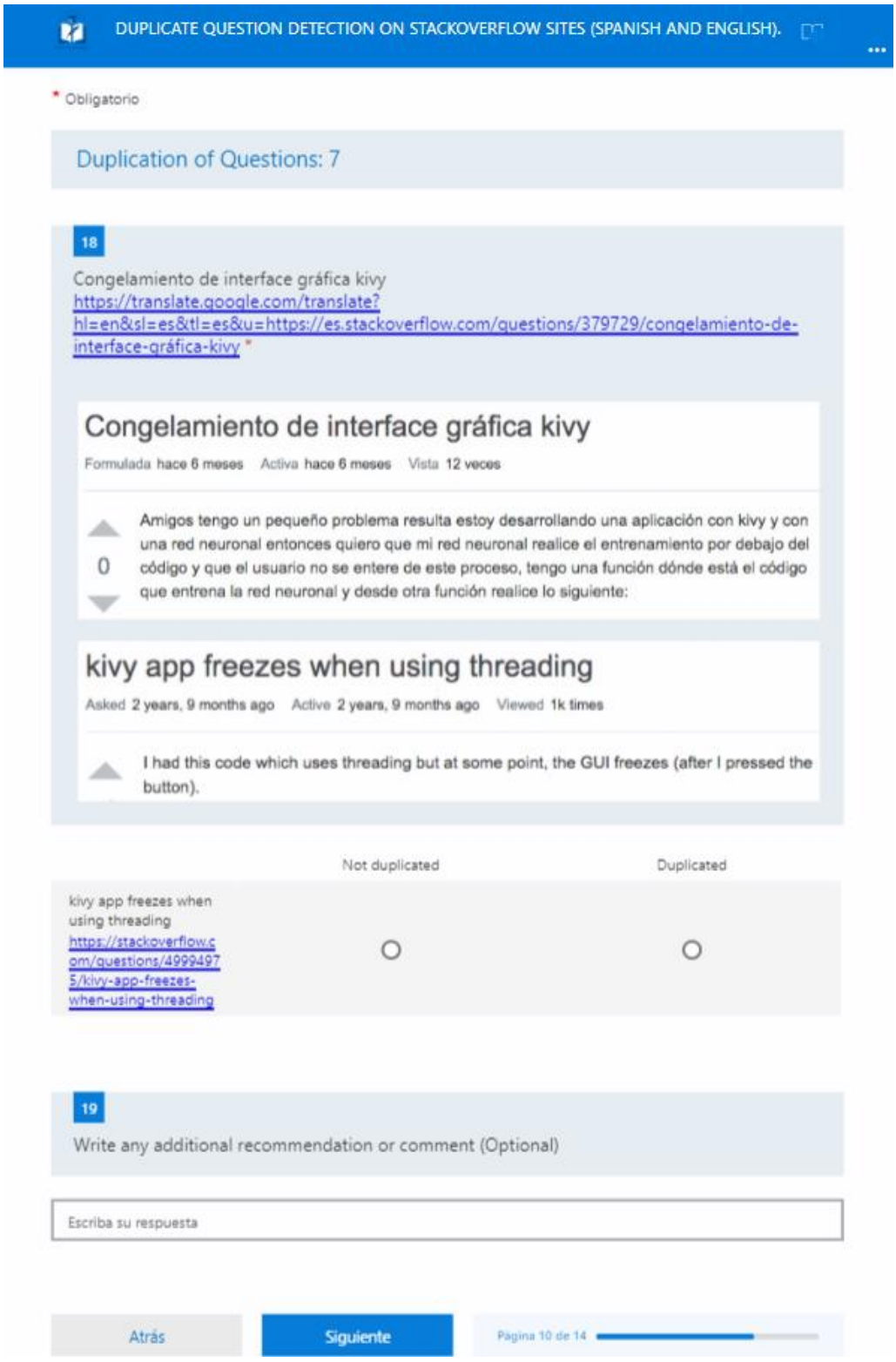
Atrás Siguiente Página 5 de 14

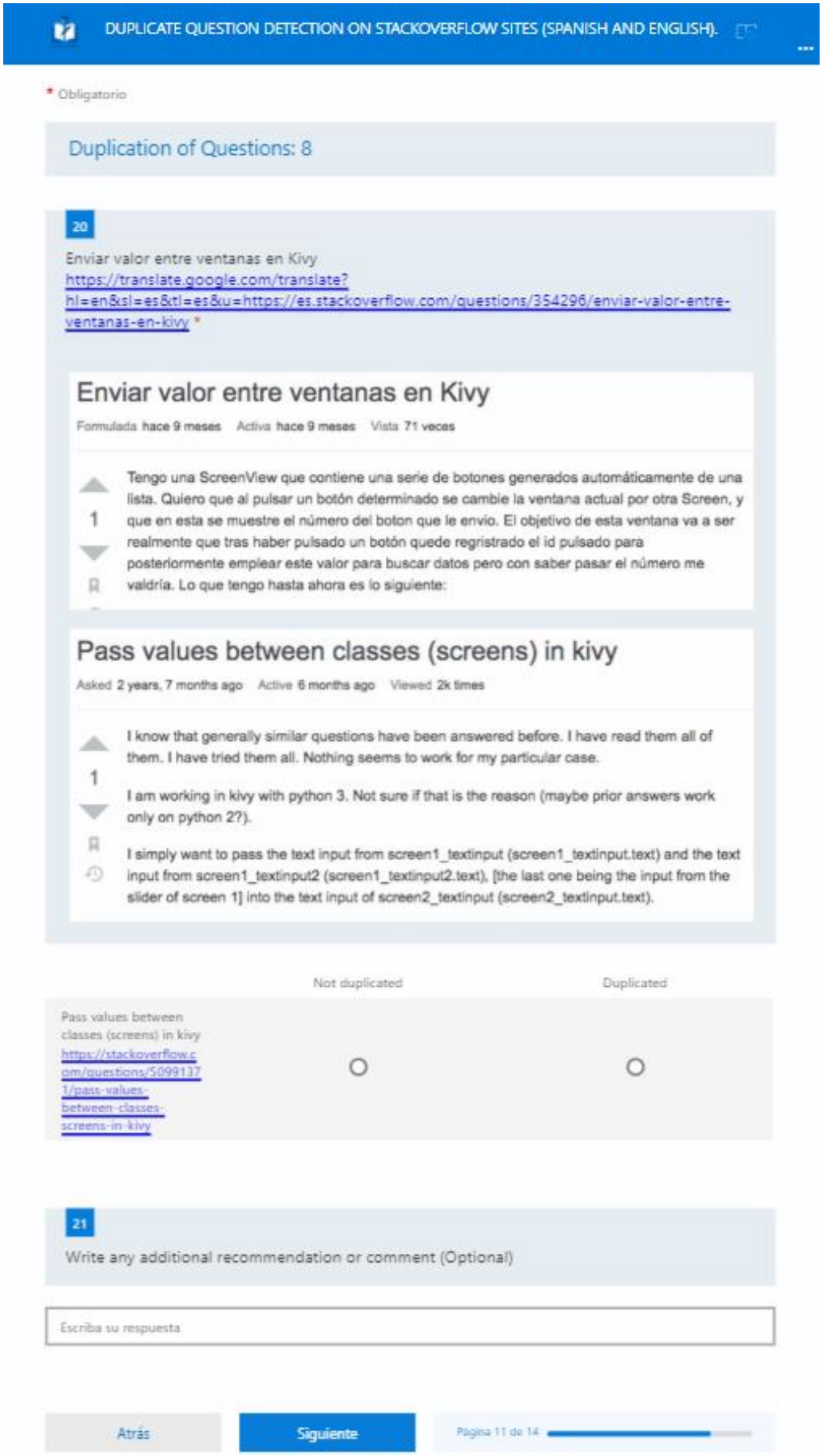

DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 3
10
Error al usar la Camara en Android con kivy
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/370009/error-al-usar-la-camara-en-android-con-kivy *
Error al usar la Camara en Android con kivy
Formulada hace 7 meses Activa hace 7 meses Vista 77 veces
Version python: 3.6.9
0
Descripción del problema: Estoy usando la libreria plyer de python para acceder a las API de android en una app kivy, la de la cámara en este caso. Se genera el .apk lo más bien pero cuando ejecuto la aplicación, se cierra de golpe la app. Depurando con el comando adb logcat -s "python", obtengo lo siguiente:
python kivy with plyer app crashes on android (camera)
Asked 4 years, 1 month ago Active 4 years, 1 month ago Viewed 1k times
3
I develop kivy application using plyer. Build by buildozer and starting app on android succeeded, but application crashes when I push the button starting a camera, and nothing is output in logcat. I work without a problem when I carry out the same code in kivylauncher. It was similar even if I tested it with an accelerometer.
I think that necessary setting may be short when I build application using plyer.
Not duplicated
Duplicated
python kivy with plyer app crashes on android (camera) https://stackoverflow.com/questions/41156449/python-kivy-with-plyer-app-crashes-on-android-camera
11
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 6 de 14

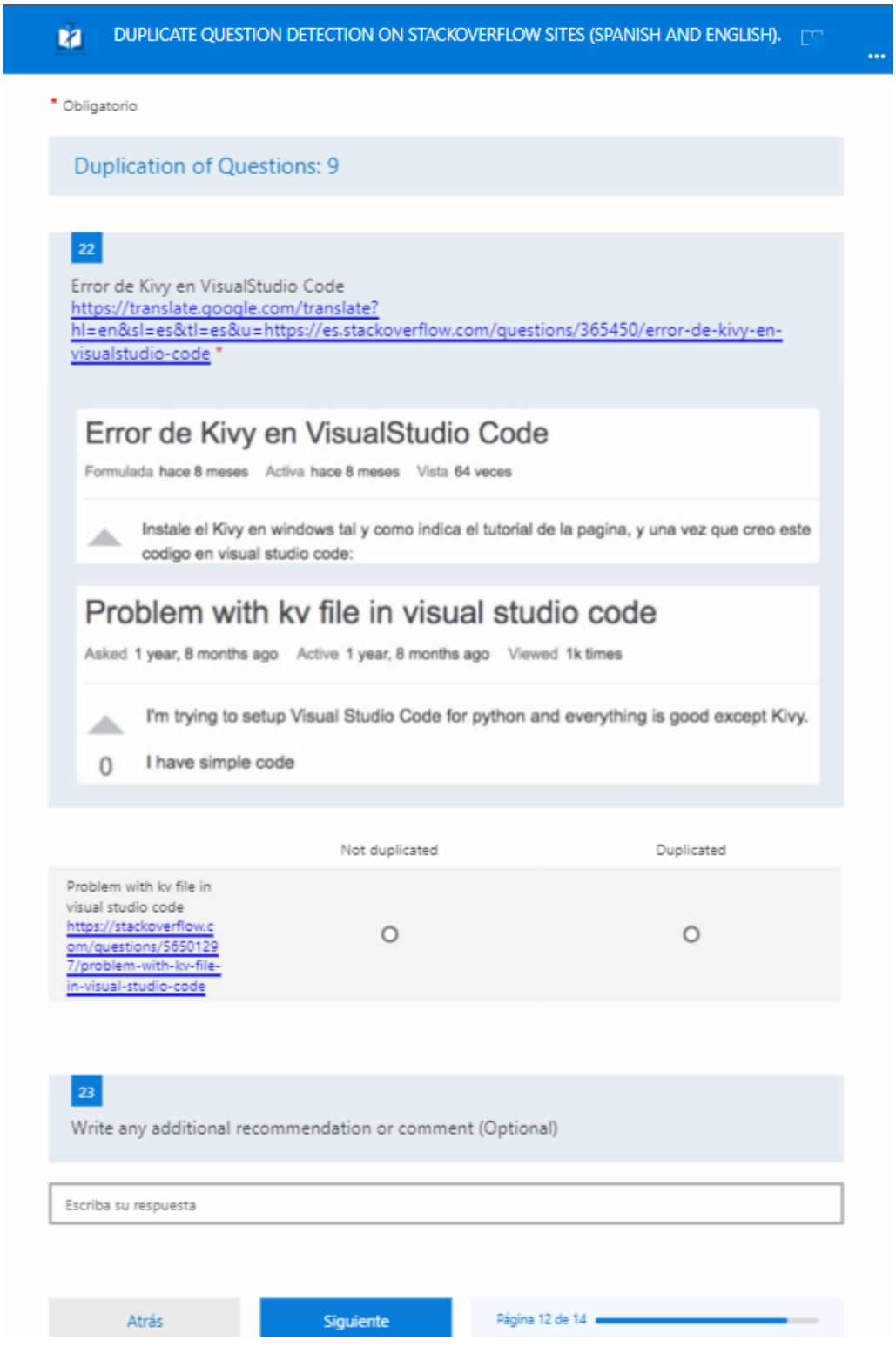

DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 4
12
Llamar Función con un Botón Usando Python-kivy
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/47802/llamar-función-con-un-botón-usando-python-kivy *
Llamar Función con un Botón Usando Python-kivy
Formulada hace 4 años Activa hace 5 meses Vista 964 veces
Nuevo en python-Kivy, con un proyecto en mente pero no logro hacer que un boton llame a una función. Les muestro mi código:
Calling Function in a Different Class Through Kivy Button
Asked 3 years, 7 months ago Active 3 years, 7 months ago Viewed 4k times
2
I am trying to call a function on button press in kivy, that is located in a different class screen than the button is located in. I tried running the function in the app class as well and ran into issues there. Here is the class where the function I am trying to call lies:
Not duplicated
Duplicated
Calling Function in a Different Class Through Kivy Button https://stackoverflow.com/questions/45024151/calling-function-in-a-different-class-through-kivy-button
13
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 7 de 14

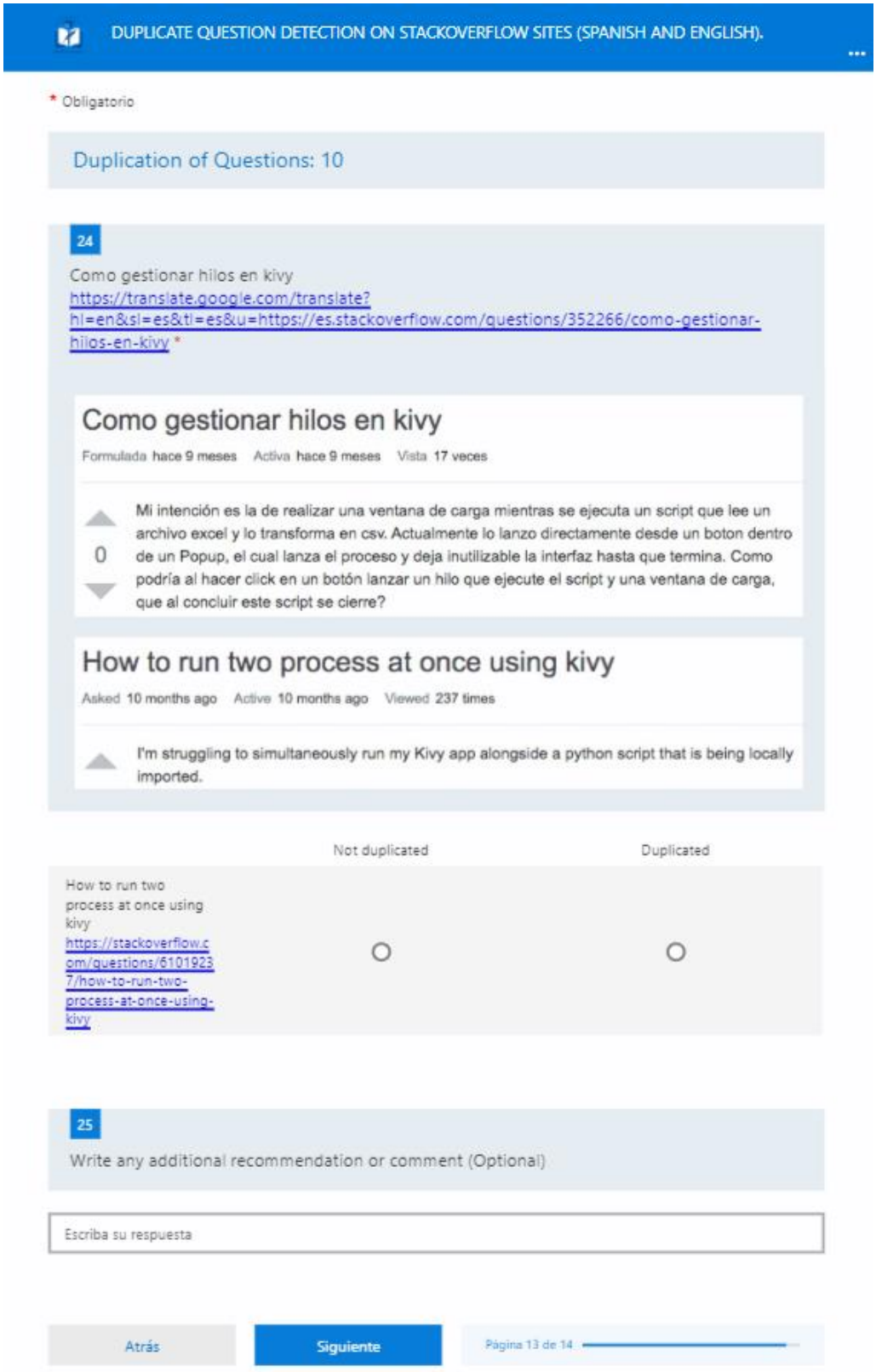

DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 5
14
Poner botón para ejecutar la camera
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/353918/poner-botón-para-ejecutar-la-camera *
Poner botón para ejecutar la camera
Formulada hace 9 meses Activa hace 9 meses Vista 41 veces
2
Amigos tengo un código en python, resulta me ayudaron a escribirlo soy nuevo en python, y más en kivy, tengo un pequeño dilema y no sé cómo hacer tengo un código que apenas se ejecuta, se abre la cámara web pero no quiero que se ejecute apenas ejecute el codigo, lo que me sirve para entender mejor es poner un botón que cuando le dé click, si me active la cámara espero me puedan ayudar gracias amigos, este es el código que tengo:
Python Kivy: how to call a function on button click?
Asked 3 years, 4 months ago Active 8 months ago Viewed 16k times
6
i'm pretty new at using kivy library.
I have an app.py file and an app.kv file , my problem is that I can't call a function on button press.
Not duplicated
Duplicated
Python Kivy: how to call a function on button click? https://stackoverflow.com/questions/46351997/python-kivy-how-to-call-a-function-on-button-click
15
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 8 de 14


DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 6
16
TypeError: 'NoneType' object is not callable
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/357088/typeerror-nonetype-object-is-not-callable *
TypeError: 'NoneType' object is not callable
Formulada hace 8 meses Activa hace 5 meses Vista 338 veces
No se porqué me dice que el objeto no es invocable, estoy haciendo los pasos del libro:
Kivy popup Filechooser pass variable (selection)
Asked 3 years ago Active 3 years ago Viewed 871 times
I want to pass selection variable from load_file_popup filechooser to GUI. when i press load button after selecting a file it gives error
0
TypeError: 'NoneType' object is not callable
Not duplicated
Duplicated
Kivy popup Filechooser pass variable (selection) https://stackoverflow.com/questions/48386125/kivy-popup-filechooser-pass-variable-selection
17
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 9 de 14


DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 7
18
Congelamiento de interface gráfica kivy
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/379729/congelamiento-de-interface-gráfica-kivy *
Congelamiento de interface gráfica kivy
Formulada hace 6 meses Activa hace 6 meses Vista 12 veces
0
Amigos tengo un pequeño problema resulta estoy desarrollando una aplicación con kivy y con una red neuronal entonces quiero que mi red neuronal realice el entrenamiento por debajo del código y que el usuario no se entere de este proceso, tengo una función dónde está el código que entrena la red neuronal y desde otra función realice lo siguiente:
kivy app freezes when using threading
Asked 2 years, 9 months ago Active 2 years, 9 months ago Viewed 1k times
I had this code which uses threading but at some point, the GUI freezes (after I pressed the button).
Not duplicated
Duplicated
kivy app freezes when using threading https://stackoverflow.com/questions/49994975/kivy-app-freezes-when-using-threading
19
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 10 de 14


DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 8
20
Enviar valor entre ventanas en Kivy
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/354296/enviar-valor-entre-ventanas-en-kivy *
Enviar valor entre ventanas en Kivy
Formulada hace 9 meses Activa hace 9 meses Vista 71 veces
1
Tengo una ScreenView que contiene una serie de botones generados automáticamente de una lista. Quiero que al pulsar un botón determinado se cambie la ventana actual por otra Screen, y que en esta se muestre el número del boton que le envio. El objetivo de esta ventana va a ser realmente que tras haber pulsado un botón quede regristrado el id pulsado para posteriormente emplear este valor para buscar datos pero con saber pasar el número me valdría. Lo que tengo hasta ahora es lo siguiente:
Pass values between classes (screens) in kivy
Asked 2 years, 7 months ago Active 6 months ago Viewed 2k times
1
I know that generally similar questions have been answered before. I have read them all of them. I have tried them all. Nothing seems to work for my particular case.
I am working in kivy with python 3. Not sure if that is the reason (maybe prior answers work only on python 2?).
I simply want to pass the text input from screen1_textinput (screen1_textinput.text) and the text input from screen1_textinput2 (screen1_textinput2.text), [the last one being the input from the slider of screen 1] into the text input of screen2_textinput (screen2_textinput.text).
Not duplicated
Duplicated
Pass values between classes (screens) in kivy
https://stackoverflow.com/questions/50991371/pass-values-between-classes-screens-in-kivy
21
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 11 de 14


DUPLICATE QUESTION DETECTION ON STACKOVERFLOW SITES (SPANISH AND ENGLISH).
* Obligatorio
Duplication of Questions: 9
22
Error de Kivy en VisualStudio Code
https://translate.google.com/translate?hl=en&sl=es&tl=es&u=https://es.stackoverflow.com/questions/365450/error-de-kivy-en-visualstudio-code *
Error de Kivy en VisualStudio Code
Formulada hace 8 meses Activa hace 8 meses Vista 64 veces
Instale el Kivy en windows tal y como indica el tutorial de la pagina, y una vez que creo este codigo en visual studio code:
Problem with kv file in visual studio code
Asked 1 year, 8 months ago Active 1 year, 8 months ago Viewed 1k times
I'm trying to setup Visual Studio Code for python and everything is good except Kivy.
0
I have simple code
Not duplicated
Duplicated
Problem with kv file in visual studio code https://stackoverflow.com/questions/56501297/problem-with-kv-file-in-visual-studio-code
23
Write any additional recommendation or comment (Optional)
Escriba su respuesta
Atrás
Siguiente
Página 12 de 14



Elaboración: Investigadores.
Fuente: Propia.

Anexo 4. Estudios seleccionados del Mapeo Sistemático

Tabla 11

Estudios seleccionados del mapeo sistemático

Código	Estudios
E-1	Ahasanuzzaman, M., Asaduzzaman, M., Roy, C. K., & Schneider, K. A. (2016, May). Mining duplicate questions of stack overflow. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)* (pp. 402-412). IEEE.
E-2	Zhang, Y., Lo, D., Xia, X., & Sun, J. L. (2015). Multi-factor duplicate question detection in stack overflow. *Journal of Computer Science and Technology*, *30*(5), 981-997.
E-3	Wang, L., Zhang, L., & Jiang, J. (2020). Duplicate question detection with deep learning in stack overflow. *IEEE Access*, *8*, 25964-25975.
E-4	Silva, R. F., Paixão, K., & de Almeida Maia, M. (2018, March). Duplicate question detection in stack overflow: A reproducibility study. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 572-581). IEEE.
E-5	Babu, J., & Thara, S. (2020). Finding the Duplicate Questions in Stack Overflow using Word Embeddings. *Procedia Computer Science*, *171*, 2729-2733.

E-6	Zhang, W. E., Sheng, Q. Z., Lau, J. H., Abebe, E., & Ruan, W. (2018). Duplicate detection in programming question answering communities. *ACM Transactions on Internet Technology (TOIT), 18*(3), 1-21.
E-7	Siu, C. (2016). Duplicate Question Detection using Online Learning.
E-8	Qiu, Z. (2018). Multilingual Stack Overflow Empirical Study and Question Retrieval Tool.
E-9	Xu, B., Xing, Z., Xia, X., Lo, D., & Le, X. B. D. (2017, August). Xsearch: a domain-specific cross-language relevant question retrieval tool. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 1009-1013).
E-10	Wang, Y. (2019). An Empirical Study on Multilingual Q&A Communities and Crosssite Question Retrieval Method.

Nota: En esta tabla se presentan los estudios que se seleccionaron en el proceso de revisión sistemática. Cada estudio es representado por un código. La elaboración es propia y la fuente corresponde a datos propios de la investigación.

Anexo 5. Artículo científico

Detección de Preguntas Duplicadas en Sitios Stack Overflow (Español e Inglés)

Bryan Moreira Pincay[1], Kerly Chica Miranda[1]
[1]Facultad de ciencias matemáticas y físicas, Universidad de Guayaquil,
Victor Manuel Rendón y Baquerizo Moreno
Guayaquil, Ecuador
{bryan.moreirap, kerlychicam}@ug.edu.ec

Resumen. Stack Overflow es una comunidad de preguntas y respuestas y son las preferidas de los programadores para resolver sus dudas. El sitio Stack Overflow en español se inició como alternativa al sitio en inglés pensado para ser utilizado por personas hispanohablantes. Sin embargo, muchas veces sus usuarios prefieren realizar sus preguntas también en el sitio inglés con el fin de obtener una respuesta de manera más rápida creando preguntas duplicadas en ambos sitios. La tarea de detectar estas preguntas duplicadas no se realiza ni siquiera por los moderadores de los sitios por lo que algunos investigadores han intentado abordar el problema utilizando técnicas de procesamiento qué lenguaje natural y aprendizaje automático. En este proyecto se realizará un análisis de la literatura mediante una revisión sistemática para determinar cuáles son las herramientas y técnicas más utilizadas por los investigadores. Luego se extraerán los datos de los sitios a fin de crear un *dataset* con pares de preguntas que serán utilizadas para los experimentos. Como experimentos se utilizarán las técnicas y herramientas analizadas en la revisión sistemática para desarrollar algoritmos cuyos resultados serán contrastados mediante un juicio externo para determinar si el rendimiento del mismo es suficiente para comprobar la hipótesis planteada, es decir, si la aplicación de técnicas de aprendizaje automático y procesamiento del lenguaje natural ayuda en la detección de preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español.

Palabras Claves. Stack Overflow, preguntas duplicadas, aprendizaje automático, procesamiento de lenguaje natural, multilingüe.

1 Introducción

Stack Overflow es uno de los sitios más conocidos de preguntas y respuestas que son beneficiosos al momento de intercambiar conocimientos tanto entre programadores aficionados como profesionales. En la actualidad hay un aproximado de 21 millones de preguntas en Stack Overflow y alrededor de 142 mil preguntas en Stack Overflow en español.

Según el Índice de Proficiencia en inglés de la compañía Education First (EF EPI) para el año 2020, Ecuador se encuentra en el puesto 93 de 100 países siendo este el país latinoamericano con el nivel más

bajo de conocimiento en el idioma inglés (Education First, 2020). Es por esto que resulta complicado para los estudiantes y profesionales ecuatorianos, y de habla hispana en general, formular por sí mismos preguntas técnicas en inglés de modo que puedan encontrar una respuesta satisfactoria en Stack Overflow. Lo cierto es que muchos de los usuarios de los sitios alternativos, como el sitio en español, al final terminan buscando respuestas en el sitio inglés de modo que una misma pregunta realizada por un usuario se puede encontrar en dos sitios en distintos idiomas con el objetivo de obtener una respuesta más rápidamente.

Qiu propone una herramienta para extraer preguntas duplicadas o relacionadas en la que utiliza la traducción automática para introducir el resultado en una función de preprocesamiento y extracción de palabras clave con el fin de formular una consulta que a través de incorporaciones de palabras será comparada con una base de preguntas en inglés de modo que se pueda determinar qué preguntas se relacionan con la de la consulta (Qiu, 2018). Otro estudio relacionado con la recuperación de preguntas relacionadas entre sitios Stack Overflow de distintos idiomas es el realizado por Xu et al. Aquí los autores introducen una herramienta llamada XSearch para realizar la búsqueda relacionada de preguntas entre los sitios en idioma chino e inglés (Xu, Xing, Xia, Lo, & Le, 2017). Uno de los primeros y más reconocidos es el estudio realizado por Zhang, Lo, Xia, & Sun, en el que propone una herramienta para la detección de preguntas duplicadas a la que llamó DupPredictor (Zhang, Lo, Xia, & Sun, 2015). En esta herramienta se extraen partes esenciales de una pregunta como su título, descripción y etiquetas. Además, obtiene el tópico de la pregunta a través de un modelo de tópicos para así utilizarlo como una característica adicional en la comparación. Por parte de (Wang, Zhang, & Jing, 2020) se realizaron pruebas utilizando tres estrategias de aprendizaje profundo (*Deep Learning*) basadas en incorporaciones de palabras

En este proyecto se buscará abordar este problema realizando una investigación para determinar si las herramientas de aprendizaje automático resultan útiles al momento de detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español por medio de la búsqueda de técnicas y herramientas de aprendizaje automático, la creación de un datasert y desarrollo de un algoritmo con su respectiva evaluación a través de un juicio externo.

2 Metodología

Considerando el objetivo del estudio se pudo establecer la población objetivo de la investigación, es decir, los usuarios que tengan en común los sitios Stack Overflow y Stack Overflow en español. Sin embargo, debido a limitaciones en cuanto a tiempo y recursos previstas en el desarrollo del algoritmo se decidió limitar el alcance a aquellos usuarios que hayan interactuado con la etiqueta Kivy lo que modificó drásticamente el tamaño de la población.

Extracción de datos

. El método utilizado para la extracción de datos que se utilizó fue la API de Stack Exchange. Esta poderosa API tiene la capacidad extraer y hasta cierto punto modificar el contenido de las bases de datos de la comunidad Stack Exchange.

Dado que esta API funciona como la mayoría de APIs disponibles en la web se la puede utilizar de varias maneras que depende del tipo de aplicación en las que son implementadas. Básicamente todo se resume a realizar una petición a la API con parámetros definidos por el usuario y esperar una respuesta comúnmente obtenida en formato JSON.

Para el estudio se decidió utilizarla de una manera convencional a través de peticiones directas a utilizando la librería *request* que precisamente sirve para hacer peticiones HTTP en Python, el lenguaje utilizado para el desarrollo del algoritmo. Este método resulta un poco complicado en cuanto a implementación, pero como beneficio tiene que se pueden definir más parámetros y tener un mayor control sobre la URL que se utilizará para realizar la petición. Utilizando la API se extrajeron todas las preguntas realizadas bajo la etiqueta Kivy tanto en Stack Overflow en español como en Stack Overflow.

Preprocesamiento

Entre las acciones realizadas en la fase de preprocesamiento se encuentran la conversión del texto a minúsculas, la eliminación de etiquetas HTML, la eliminación de signos de puntuación, la eliminación de palabras vacías y en el caso de las preguntas en español se tradujeron sus componentes al idioma inglés.

Uno de los puntos más importantes en la etapa de preprocesamiento fue decidir qué datos debían mantenerse como entrada para el algoritmo. Esto debido a que en el cuerpo de la pregunta se incluían muchos datos que podían causar ruido en el análisis. Por ejemplo, las preguntas de algunos usuarios contenían enlaces a repositorios de imágenes donde ellos subían una captura del problema que estaban teniendo o de lo que querían llegar a desarrollar mediante su pregunta.

También se determinó que el código fuente incluido en las preguntas muchas veces era la clave para entender a qué se refería el usuario con su pregunta y por ende entender el sentido de esta. Esto es útil tanto para quienes buscan responder la pregunta como para quienes buscan determinar si la pregunta es duplicada de otra. Debido a esto en vez de eliminar por completo el contenido de la etiqueta, se intentó realizar una limpieza de los fragmentos de código eliminando términos duplicados y caracteres especiales.

Luego de realizar el tratamiento de las etiquetas HTML y su contenido además de los signos de puntuación se procedió a realizar la traducción del texto en el caso de las preguntas en español. Para esta tarea se utilizó nuevamente un *wrapper* que permitía utilizar la API de Google Traductor dentro de un entorno de desarrollo en Python.

En la tabla 2.1 se muestra la comparación entre el cuerpo de una pregunta antes del preprocesado y el cuerpo de la misma pregunta luego del preprocesado.

Versión	Cuerpo
Contenido original	Estoy desarrollando una aplicación android con python e intento utilizar pyjnius para implementar los módulos de JavaCuando le doy un import a jnius no tengo problemas El detalle viene cuando intento importar cualquier parte de la api de android, pues obtengo el siguiente error He leído varias publicaciones con un error similar, sin embargo no encuentro la solución. Muchas gracias.
Preprocesado	I am developing an android application with python and trying to use pyjnius to implement the java modules When I give an import to jnius I have no problems The detail comes when I try to import any part of the android api, because I get the following error I have read several posts with a similar error, however I cannot find the solution. Thanks a lot.

Tabla 2.1. Comparación de contenido original y preprocesado

Etiquetado

Para realizar la detección de preguntas duplicadas mediante técnicas de aprendizaje automático era necesario tener un dataset en el cual se incluyeran las preguntas en español con sus posibles duplicadas en inglés a fin de utilizarlo para entrenar los modelos.

Como primer paso para el proceso de etiquetado debían buscarse coincidencias de búsqueda de las preguntas en español. Para esto se decidió utilizar el motor de búsquedas personalizadas de Google (Custom Search Engine CSE). Para cada pregunta en español se decidió tomar las cinco primeras coincidencias proporcionadas por el motor de búsquedas personalizadas de Google que además coincidieran con la base de preguntas en inglés bajo la etiqueta Kivy. Con esto se obtuvo un aproximado de 326 pares de preguntas para realizar el entrenamiento del algoritmo. Luego de haber realizado las comparaciones sea añadió una columna adicional al *dataset* para indicar si el par de preguntas de determinada fila eran consideradas como preguntas duplicadas o no.

Algoritmo

Se emplearon diversas técnicas de aprendizaje automático a fin de encontrar preguntas duplicadas entre los elementos del *dataset* luego de haber realizado la construcción, preprocesamiento y etiquetado de los datos. Para los experimentos se realizaron pruebas con 3 estrategias diferentes utilizando técnicas de aprendizaje automático de diversas maneras de modo que se lograra determinar cuál tiene un mejor rendimiento frente a las otras.

Como punto a considerar se tiene la utilización de un modelo pre entrenado de incorporaciones de palabras tomado del estudio conducido por (Efstathiou, Chatzilenas, & Spinellis, 2018). Este modelo fue realizado debido a la falta de modelos pre entrenados cuyo dominio específico sea la ingeniería de software. Se decidió utilizar este modelo pre entrenado debido al costo en cuanto a tiempo y recursos que implica realizar un entrenamiento con un volumen de datos igual al utilizado por los autores. Este modelo se entrenó utilizando un data dump de Stack Overflow que abarcaba información desde su lanzamiento en el año 2008 hasta el año 2017.

Word Mover Distance

La primera estrategia pensada para detectar la duplicidad entre preguntas fue la de utilizar la técnica de *Word Mover Distance* para calcular la similitud entre grupos de palabras. Para esto se realizó la conversión del título y el cuerpo de cada pregunta a listas de palabras de modo que se pudieran unir el título y el cuerpo en una sola lista que representará todas las palabras presentes en la pregunta. Una vez obtenidas las listas de palabras se procedió a ejecutar el método *wmdistance* incluido en el objeto contenedor del modelo.

Una vez se calcule el valor de similitud de cada par de preguntas se intentará encontrar el umbral óptimo para el algoritmo. En este caso se intentó encontrar el umbral óptimo de modo que la métrica de exhaustividad sea la mejor. El valor obtenido como umbral óptimo de 1.162 produciendo un puntaje de exhaustividad de 1.0. Esto quiere decir que el 100% de las preguntas duplicadas lograron ser detectadas mediante la similitud entre su texto. Sin embargo, al analizar la matriz de confusión para el umbral óptimo para exhaustividad presentada en la figura 3.1 se puede observar que la cantidad de falsos positivos es bastante elevada debido a que muchas preguntas parecen tener una similitud de texto bastante parecida sin llegar a ser un par duplicado necesariamente.

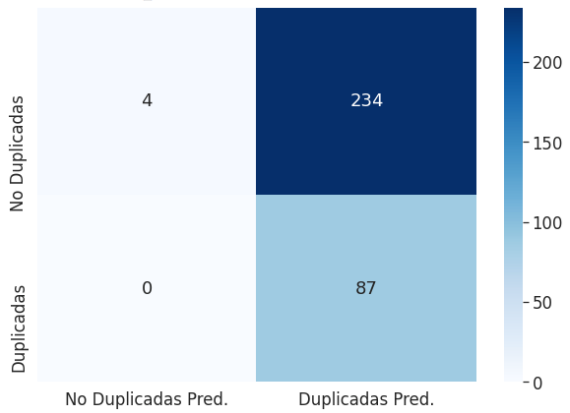


Figura 2.1. Matriz de confusión del método WMD con umbral óptimo para exhaustividad

Debido a los problemas presentados al utilizar la exhaustividad como métrica de referencia se decidió cambiar esta métrica por el puntaje F1 el cual equivale a la media armónica entre la precisión y la

exhaustividad. Al realizar esto se obtuvieron mejores resultados obteniendo un puntaje de exhaustividad de 0.99 y disminuyendo la cantidad de falsos positivos predichos como se muestra en la matriz de confusión de umbral óptimo para el puntaje F1 presentada en la figura 2.2.

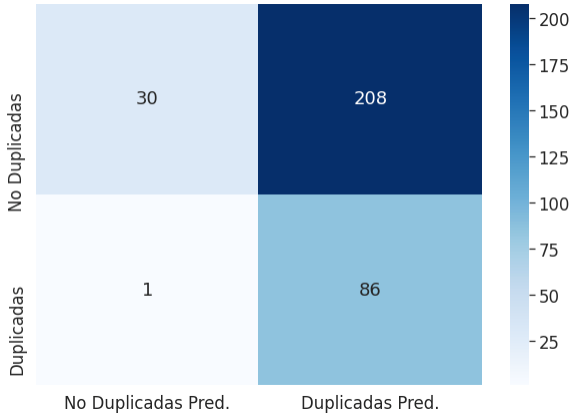


Figura 2.2. Matriz de confusión del método WMD con umbral óptimo para el puntaje F1

Caracterización con Word Mover Distance

En la segunda estrategia formulada se determinó qué las comparaciones que hacen los humanos entre preguntas no consisten necesariamente en revisar si sus palabras se parecen. Más bien al momento de realizar una comparación entre preguntas es común que se revisen la similitud de los componentes por separados, es decir, los títulos con títulos, cuerpos con cuerpos y etiquetas con etiquetas. Se pensó que estos valores de similitud podrían constituir las características de observación que se pudieran utilizar para entrenar modelos de aprendizaje automático. Para entrenar los modelos se procedió a realizar la división del *dataset* de entrenamiento fue de un 70% dejando así un 30% disponible para realizar las respectivas pruebas. Como métricas para evaluar los modelos se utilizaron los métodos provistos por la clase *metrics* de la librería *sklearn.*

Regresión logística

Utilizando el modelo de regresión logística se obtuvo un puntaje de exactitud de 0.76531. En principio este valor parece indicar que el modelo tiene un rendimiento bastante bueno. Sin embargo, al revisar la matriz de confusión presentada en la figura 2.3 se hizo evidente que el modelo tiene en realidad un rendimiento pésimo al momento de detectar preguntas duplicadas llegando a obtener un puntaje de exhaustividad de 0,08.

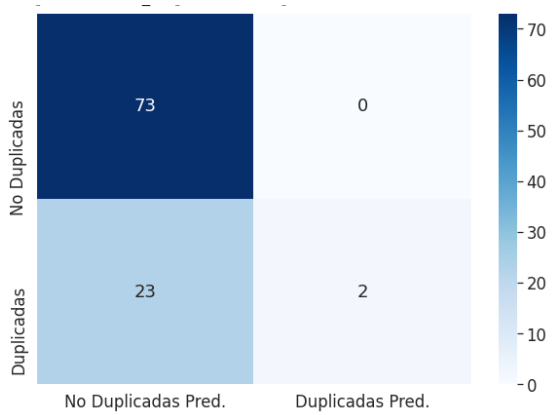


Figura 2.3. Matriz de Confusión de duplicidad con Regresión Logística

Al intentar determinar las causas de este bajo rendimiento se encontró que en efecto el *dataset* estaba bastante desbalanceado. Esto quiere decir que la cantidad de pares de preguntas duplicadas era mucho mayor a la cantidad de pares duplicados como se muestra en la figura 2.4. Debido a esto era más probable que el modelo detectara un par de preguntas como no duplicada.

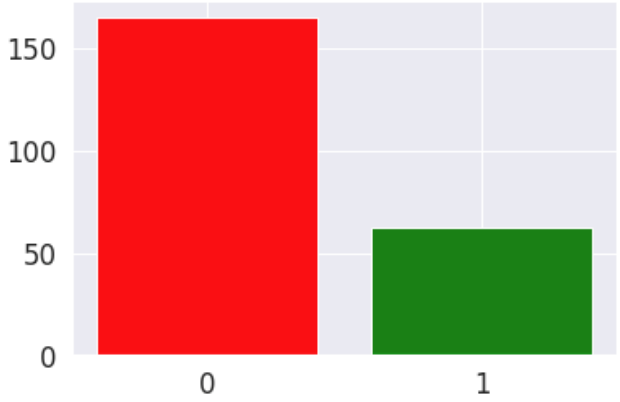


Figura 2.4. Histograma de cantidad de preguntas duplicadas y no duplicadas en el dataset

Para solucionar este problema se decidió utilizar el método *SMOTE (Synthetic Minority Oversampling Technique)*. Este método es una forma de *supersampling* que incrementa el número de muestras seleccionando ejemplos que se encuentran cerca en el espacio de características. Luego traza una línea entre estos ejemplos en su espacio de características de modo que se obtiene una nueva muestra trazada a lo largo de esta línea de características. Cómo se puede observar en la figura 2.5 la cantidad de preguntas duplicadas es igual a la de las no duplicadas después de aplicar el método SMOTE para balancear el dataset.

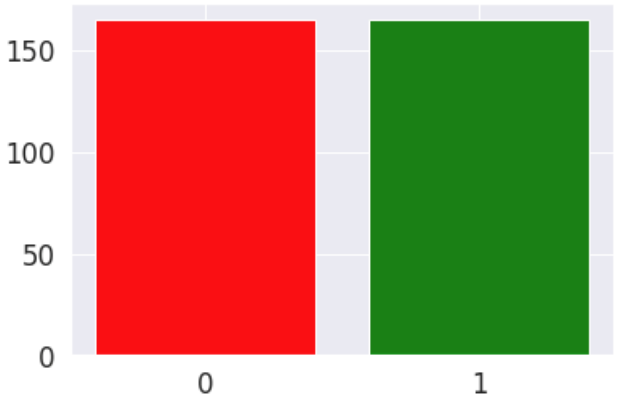


Figura 2.5. Histograma de cantidad de preguntas en el dataset balanceado con SMOTE

Los resultados de la regresión logística utilizando el *dataset* balanceado se pueden observar en la figura 2.6. Se puede observar cómo se pudieron detectar 12 de los 25 pares de preguntas duplicadas obteniendo así un puntaje de exhaustividad de 0,48 el cual representa una mejora sustancial al 0,08 obtenido usando el *dataset* desbalanceado. Resulta curioso que el puntaje de exactitud de este modelo fue de 0,68 llegando a ser incluso menor al del modelo con el *dataset* desbalanceado.

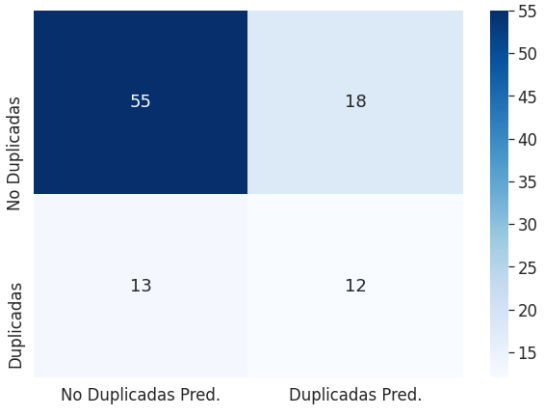


Figura 2.6. Matriz de Confusión de duplicidad con Regresión Logística y dataset balanceado

Red Neuronal

Además de la regresión logística se hicieron experimentos utilizando la caracterización con *wmdistance* como entrada de una red neuronal. La red neuronal se creó utilizando la clase *sequential* provista por la librería *Keras* perteneciente a Google.

En cuanto a la arquitectura de la red neuronal se configuraron 3 capas. La primera capa es la capa visible de entrada que recibirá las características de cada observación y se definió como una capa densa de 12 nodos cuya función de activación elegida fue *ReLU*. La segunda capa es oculta y está conformada por 8 nodos y al igual que la primera capa esta es densa y utiliza la misma función de activación. Por último, la capa de salida consta de un solo nodo y la función de activación de esta capa fue Sigmoide debido al buen rendimiento que tiene esta función en problemas de clasificación binaria. El modelo se entrenó por 100 épocas con un *batch_size* de 2.

Los resultados obtenidos por el modelo superaron a los obtenidos por el modelo de regresión logística por un margen significativo presentando un puntaje de exhaustividad de 0,64 en comparación al 0,48 del modelo anterior. Cabe recalcar que estos resultados se obtuvieron entrenando el modelo con el *dataset* balanceado ya que los resultados del *dataset* desbalanceado fueron pésimos al igual sucedió que con el modelo de regresión logística. En la matriz de confusión presentada en la figura 2.7 se puede observar cómo se redujo la cantidad de predicciones erróneas en el *dataset* de prueba.

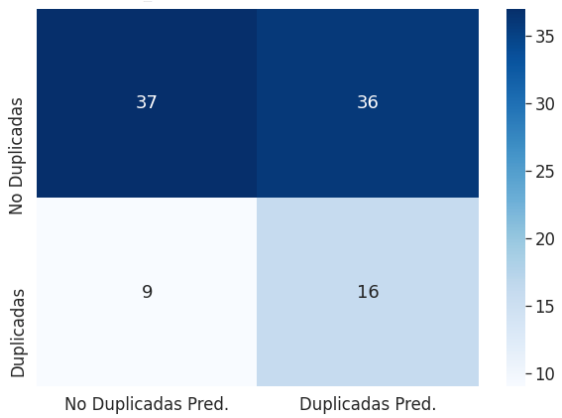


Figura 2.7. Matriz de Confusión de duplicidad con red neuronal

A pesar de ser una red neuronal sencilla cuyo proceso de ajuste no se optimizó en lo absoluto se puede determinar esta estrategia de detección como la mejor de entre las demás analizada mediante estos experimentos. Considerando la poca optimización y la poca cantidad disponible para el entrenamiento este modelo proporcionó resultados significativamente superiores a los de los modelos y métodos anteriores.

Evaluación del algoritmo

se realizó la evaluación del algoritmo a través de un juicio externo obtenido mediante encuestas a la población definida para este estudio. La población que se tomó en el presente trabajo de investigación fue extraída del total de los usuarios del sitio Stack Overflow en español e inglés que participan en preguntas con la etiqueta Kivy, en donde el criterio de selección que se utilizo fue que los usuarios formen parte de las comunidades Stack Overflow en español y Stack Overflow de manera simultánea obteniendo así un total de 79 usuarios para el estudio. Un criterio de exclusión que se utilizó fue que los usuarios que no contaban con una manera de ser contactados en su perfil no serían incluidos ya que no existía ninguna manera formal de comunicarse con estos e invitarlos a participar en la encuesta. Por lo expuesto anteriormente fueron seleccionados 17 usuarios que tenían en su perfil disponible una dirección de correo. Debido a que los criterios y limitaciones de la selección redujeron tanto la cantidad de usuarios seleccionados se determinó que la muestra sería igual a la población ya qué es lo recomendado cuando se trabaja con poblaciones muy pequeñas.

La encuesta estuvo compuesta por un total de 25 preguntas y esta se separó en 3 secciones con el fin de separar los datos específicos según los factores que componen las preguntas. En cuanto a las preguntas más importante se tuvo la ocupación del encuestado y 10 pares de preguntas seleccionadas al azar de los resultados del algoritmo. Estos 10 pares de preguntas debían ser comparados por el encuestado y ser marcados como duplicadas o no duplicadas.

Para la validación de las encuestas la técnica de análisis de contingencia (Estadístico de prueba Chi-cuadrado) para el contraste de hipótesis.

La hipótesis se evaluará de acuerdo con el siguiente criterio de expertos:

- Si $p < 0.05$ entonces se rechaza H0 y se acepta H1 y se dice significativo (Si la probabilidad correspondiente al valor calculado por la prueba estadística es menor o igual que su respectivo valor crítico al nivel de 0.05, entonces se rechaza H0 y se dice significativo).
- Si $p > 0.05$ entonces se acepta H0 y se dice no significativo (Si la probabilidad correspondiente al valor calculado por la prueba estadística es mayor que su respectivo valor crítico al nivel de 0.05, entonces se acepta H0 y se dice no significativo).

Donde:

p: es el valor de probabilidad

α: es el nivel de significancia, con un valor de 0.05

H_0: El modelo realizado con técnicas de aprendizaje automático ayuda en la detección de preguntas duplicadas en SO y SOES. Hipótesis nula.

H_1: El modelo realizado con técnicas de aprendizaje automático no ayuda en la detección de preguntas duplicadas en SO y SOES. Hipótesis alternativa.

En la tabla 2.2 se puede observar los valores del estadístico chi-cuadrado para cada una de las preguntas de la encuesta. Además se muestra el respectivo *p-value* que al ser comparado con el valor de significancia permitirá determinar si se acepta o se rechaza la hipótesis nula H_0.

Contraste	**Chi-cuadrado**	***p-value***
Par de preguntas 1	0,6857	0,4076
Par de preguntas 2	0,0686	0,7934
Par de preguntas 3	0,6857	0, 4076
Par de preguntas 4	0,0098	0,9211
Par de preguntas 5	0,3429	0,5581
Par de preguntas 6	0,0098	0,9211
Par de preguntas 7	0,1143	0,7353
Par de preguntas 8	0,1143	0,7353
Par de preguntas 9	0,0686	0,7934
Par de preguntas 10	1,1853	0,2762

Tabla 2.2. Resultados del contraste para los 10 pares de preguntas con chi-cuadrado

3 Resultados y discusión

Como resultado de los experimentos realizados con técnicas de aprendizaje automático se determinó que el modelo con rendimiento más equilibrado es la red neuronal entrenada con el *dataset* balanceado con SMOTE. Como se muestra en la tabla 2.3 se puede notar que la exhaustividad tuvo un incremento sustancial con respecto al modelo anterior. Esto lo logra sin sacrificar demasiado su rendimiento en cuanto a exactitud por lo que debido a este equilibrio en rendimiento se elegió como el mejor modelo de los experimentos realizados.

Técnicas utilizadas	**Exactitud**	**Exhaustividad**
WMD con umbrál óptimo para exhaustividad	0,28	1,00
WMD con umbrál óptimo para puntaje F1	0,35	0,99
Regresión Logística con *dataset* desbalanceado	0,76	0,08
Regresión Logística con *dataset* balanceado	0,68	0,48
Red neuronal con *dataset* balanceado	0,54	0,64

Tabla 2.2. Resultados de las distintas técnicas probadas en el desarrollo del algoritmo

Luego de haber contrastado los resultados de los experimentos con un juicio externo utilizando el estadístico chi-cuadrado se puede determinar que la hipótesis planteada es correcta. Es decir, que las herramientas y técnicas de aprendizaje automático en efecto ayudan a detectar la presencia de preguntas duplicadas entre los sitios de Stack Overflow y Stack Overflow en español. Esto se hace evidente al analizar los resultados individuales de las diez preguntas que se incluyeron en la encuesta. Se pudo apreciar que en cada una de estas preguntas el *p-value* resulta ser mayor al nivel de significancia determinado, en este caso de .05. Al ser mayor qué nivel de significancia el *p-value* indica que la hipótesis nula se acepta. Dado que éste fue el caso para las 10 preguntas propuestas en la encuesta se puede concluir que la hipótesis es verdadera.

Puesto que la complejidad de establecer la duplicidad entre dos preguntas de los sitios Stack Overflow es bastante elevada no se puede determinar con exactitud el rendimiento de los algoritmos propuestos en el experimento ya que como se muestra en las tablas de contingencia no todos los encuestados estuvieron de acuerdo en cuanto a la duplicidad de una pregunta determinada por lo que se hace evidente que resulta complicado incluso para los humanos determinar cuándo una pregunta es duplicada. Esto puede suceder debido a varios factores que pueden dificultar la comparación. Entre estos tenemos el mal uso del lenguaje, la no inclusión de código fuente descriptivo o por el contrario la inclusión de código fuente no descriptivo que puede llegar a crear confusión y no dejar que se haga evidente la intención de la pregunta.

Ante lo expuesto se puede reconocer que la aplicación de técnicas de aprendizaje automático logra detectar un porcentaje de las preguntas duplicadas con lo cual se cumple el objetivo del estudio sentando

las bases para próximos estudios que deseen ahondar en este tema y de alguna manera mejorar y complementar el trabajo realizado.

4 Conclusiones

El rendimiento del algoritmo en términos generales podría parecer bajo, pero al comparar sus resultados con los de las herramientas presentadas en la revisión sistemática se puede notar que los resultados y las métricas se asemejan bastante por lo que se puede concluir que funciona lo suficientemente bien y los errores de predicción se encuentran dentro de lo aceptable. Cabe mencionar que son muy pocos los estudios que se han realizado sobre la detección de preguntas duplicadas multilingües en Stack Overflow llegando incluso a no haberse encontrado ningún estudio en la revisión sistemática que comparara preguntas en inglés y español. Debido a esto es difícil comparar los resultados del algoritmo y determinar si su rendimiento es superior o inferior ya que para efectos de este estudio no se encontraron herramientas similares.

Utilizando las encuestas como técnica de recolección de datos se logró obtener una evaluación asistida por desarrolladores de la comunidad Stack Overflow. Sin embargo, debido a una interrupción en el proceso de recolección ocasionada por un bloqueo por parte de un moderador de uno de los sitios no se logró recolectar la cantidad de datos esperada para satisfacer la muestra. Fueron 12 respuestas las que se obtuvieron de las 17 esperadas lo que representa un 70% de datos recolectados por lo cual se decidió continuar con el análisis ya que en términos relativos se logró recolectar una cantidad significativa de datos para realizar la evaluación. En cuanto al análisis de las hipótesis realizado a través del cálculo del estadístico chi-cuadrado para cada una de las preguntas de la encuesta, como se pudo observar en la sección de resultados, se logró determinar que en efecto el algoritmo propuesto permite detectar preguntas duplicadas entre los sitios Stack Overflow y Stack Overflow en español.

Referencias bibliográficas

Education First. (2020). *índice del EF English Proficiency*. Obtenido de EF: https://www.ef.com.ec/epi/regions/latin-america/ecuador/#

Qiu, Z. (2018). Multilingual Stack Overflow Empirical Study and Question Retrieval Tool.

Zhang, Y., Lo, D., Xia, X., & Sun, J. L. (2015). Multi-factor duplicate question detection in stack overflow. . *Journal of Computer Science and Technology*, 981-997.

Wang, L., Zhang, L., & Jing, J. (2020). Duplicate Question Detection With Deep Learning in Stack Overflow. *IEEE Access*, 25964 - 25975.

Xu, B., Xing, Z., Xia, X., Lo, D., & Le, X. B. (2017). Xsearch: a domain-specific cross-language relevant question retrieval tool. *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, (págs. 1009-1013).

Efstathiou, V., Chatzilenas, C., & Spinellis, D. (2018). Word Embeddings for the Software Engineering Domain. *Proceedings of the 15th International Conference on Mining Software Repositories.* ACM.