



UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO

CARRERA DE COMPUTACIÓN

**DESARROLLO DE UN APLICATIVO WEB PARA SISTEMAS DE RECOMENDACIÓN
DE PRÁCTICAS PREPROFESIONALES EN LA BOLSA DE TRABAJO PARA LOS
ESTUDIANTES DE LA UNIVERSIDAD POLITÉCNICA SALESIANA.**

Trabajo de titulación previo a la obtención del título de
Ingenieros en Ciencias de la Computación

AUTOR: MICHAEL JAVIER ROBLES LEDESMA

TUTOR: DANIEL GIOVANNY DÍAZ ORTIZ

Quito – Ecuador

2023

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Michael Javier Robles Ledesma con documento de identificación 1727418822 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 14 de agosto del año 2023

Atentamente,



Michael Javier Robles Ledesma

1727418822

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Michael Javier Robles Ledesma con documento de identificación 1727418822, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del PROYECTO TÉCNICO: “Desarrollo de un aplicativo web para sistemas de recomendación de prácticas preprofesionales en la bolsa de trabajo para los estudiantes de la Universidad Politécnica Salesiana”, el cual ha sido desarrollado para optar por el título de: INGENIERO EN CIENCIAS DE LA COMPUTACIÓN, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 14 de agosto del año 2023

Atentamente,



Michael Javier Robles Ledesma

1727418822

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Daniel Giovanni Díaz Ortiz con documento de identificación 1716975501, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN APLICATIVO WEB PARA SISTEMAS DE RECOMENDACIÓN DE PRÁCTICAS PREPROFESIONALES EN LA BOLSA DE TRABAJO PARA LOS ESTUDIANTES DE LA UNIVERSIDAD POLITÉCNICA SALESIANA , realizado por Michael Javier Robles Ledesma con documento de identificación 1727418822, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 14 de agosto del año 2023

Atentamente,



Ing. Daniel Giovanni Díaz Ortiz, MSc

1716975501

DEDICATORIA

Dedico este proyecto a las personas más importantes en mi vida, empezando por Dios, por bendecirme en cada paso que he dado. A mi madre Clara Ercilia Ledesma Pozo, ya que esta tesis no habría sido posible sin tu amor, paciencia y apoyo incondicional.

Desde el primer día en que decidí embarcarme en este desafío, tú estuviste ahí para motivarme y ayudarme a superar todos los obstáculos que se presentaron en el camino, tus palabras de aliento y tus abrazos me dieron la fuerza necesaria para seguir adelante en los momentos de duda y cansancio. Tu dedicación y sacrificio han sido una inspiración para mí y me han enseñado el valor del trabajo duro y la perseverancia.

A través de tu ejemplo, he aprendido que el amor y el apoyo de una madre son fundamentales para alcanzar cualquier meta. Gracias por ser mi pilar, mi confidente y mi guía en esta y en todas las etapas de mi vida.

Esta tesis es un tributo a tu amor incondicional y a todo lo que has hecho por mí. Espero que al leer estas palabras sientas lo orgulloso que estoy de tenerte como madre y de todo lo que hemos logrado juntos, gracias por ayudarme a concluir este sueño, te amo.

Finalmente, agradezco a mis hermanos Andrea López, Hugo López y Mónica Ledesma, por brindarme cariño y felicidad, durante todo este proceso, su presencia en mi vida ha sido una fuente constante de alegría y motivación. Saber que podía contar con ustedes para cualquier cosa me dio la tranquilidad y seguridad necesarias para seguir adelante; sus consejos, sugerencias y palabras de aliento han sido de gran ayuda para mí.

Michael Javier Robles Ledesma

AGRADECIMIENTOS

Agradezco a Dios por bendecirme día a día y mantenerme con buena salud para poder concluir este proyecto. También agradezco a mi tutor de tesis: Daniel Geovanny Díaz Ortiz por su apoyo y orientación durante todo el proceso de elaboración de esta tesis, desde el primer día, su compromiso, profesionalismo y paciencia fueron evidentes. Gracias a su experiencia y conocimientos, pude diseñar y llevar a cabo este proyecto, a la vez que pude ampliar mi comprensión sobre los temas más relevantes en mi campo de estudio.

A mi lector Alberto Duchi, también le quiero dar las gracias por su dedicación y entrega. Siempre estuvo dispuesto a compartir sus conocimientos y a aconsejarme en cada paso del camino.

Y por último a todos los docentes que conforman la carrera de computación ya que, con sus conocimientos, sabiduría y apoyo, me pude desarrollar profesionalmente, con valores éticos y morales como un buen y honrado ciudadano.

Michael Javier Robles Ledesma

ÍNDICE

INTRODUCCIÓN	15
PLANTEAMIENTO DEL PROBLEMA.....	16
Antecedentes	16
Justificación	22
OBJETIVOS.....	23
Objetivo general.....	23
Objetivos específicos	23
ALCANCE	24
Alcance general del aplicativo	24
Roles de usuarios	26
Módulos	27
Módulos generales	27
Módulo del estudiante.....	27
Módulo del docente.....	28
Módulo de la empresa.....	28
Módulo del administrador.....	29
MARCO METODOLÓGICO.....	29
Metodología de desarrollo	29
Metodología Ágil.....	30
Metodología De Investigación Experimental	32
CAPITULO 1: ESTADO DEL ARTE.....	34
1.1. Algoritmos de aprendizaje automático:.....	34
1.2. Big Data:.....	34
1.3. Estándares:.....	34
1.4. Frameworks:	34
1.5. Artillery:	34
1.6. Inteligencia Artificial (IA o AI):	35
1.7. Inteligencia Empresarial (Business Intelligence):	35
1.8. Prácticas preprofesionales:	35
1.9. Protocolos de encriptación:	35

1.10.	Procesamiento de lenguaje natural (NLP):	35
1.11.	ISO/IEC 9126:	35
CAPITULO II: ANALISIS Y DISEÑO.....		36
2.1.	Requerimientos funcionales y no funcionales:.....	36
2.1.1.	Requerimientos Funcionales	36
2.1.2.	Requerimientos No Funcionales	37
2.2.	Historias de usuarios.....	38
2.3.	Diagramas de casos de uso	44
2.3.1.	RF01 Registro de usuarios	44
2.3.2.	RF02 Iniciar Sesión.....	46
2.3.3.	RF03 Actualización de datos.....	47
2.3.4.	RF04 Visualización Publicaciones.....	48
2.3.5.	RF05 Visualización Perfiles.....	49
2.3.6.	RF06 Publicaciones.....	49
2.3.7.	RF07 Reacciones.....	50
2.3.8.	RF08 Hoja de vida.....	51
2.4.	Diagramas de secuencia.....	52
2.4.1.	RF01 Registro de usuarios	52
2.4.2.	RF02 Iniciar Sesión.....	53
2.4.3.	RF03 Actualización de datos.....	54
2.4.4.	RF04 Visualización Publicaciones.....	55
2.4.5.	RF05 Visualización Perfiles.....	55
2.4.6.	RF06 Publicaciones.....	56
2.4.7.	RF07 Reacciones.....	57
2.4.8.	RF08 Hoja de vida.....	59
2.5.	Diccionario de datos	59
CAPITULO III: CONSTRUCCION Y PRUEBAS		64
2.4.	Diagrama de despliegue de la aplicación	64
2.5.	Algoritmos Importantes	66
2.5.1.	Algoritmos Importantes de Python	66
2.5.2.	Algoritmos Importantes de Nodejs	77

2.5.3. Algoritmos Importantes de Angular.....	91
2.6. Pruebas De Carga Y Estrés.....	96
2.7. Pruebas de caja negra	100
2.8. Análisis de resultados	103
2.8.1. Obtención de datos:.....	103
2.8.2. Etiquetas e identificadores cualitativos:.....	104
2.8.3. Coincidencia entre las publicaciones y el currículum:.....	104
2.8.4. Nivel de reconocimiento en la carga de datos y automatización de la hoja de vida del estudiante.	108
Conclusiones	109
Recomendaciones.....	110

Índice de Tablas

Tabla 1	Tabla de stack de desarrollo MEAN	29
Tabla 2	Tabla comparativa de metodologías de desarrollo	30
Tabla 3	Tabla requerimientos funcionales	36
Tabla 4	Historia de usuario registro nuevos usuarios.....	39
Tabla 5	Historia de usuario inicio de sesión.....	40
Tabla 6	Historia de usuario actualización de datos	40
Tabla 7	Historia de usuario visualización de publicaciones recomendadas.....	40
Tabla 8	Historia de usuario visualización de perfiles recomendados	41
Tabla 9	Historia de usuario creación de publicaciones	42
Tabla 10	Historia de usuario reacciones.....	42
Tabla 11	Historia de usuario hoja de vida.....	43
Tabla 12	Tabla diccionario modelo Usuario	59
Tabla 13	Tabla diccionario modelo Sesión.	60
Tabla 14	Tabla diccionario modelo Perfil Empresa.....	60
Tabla 15	Tabla diccionario modelo Perfil Docente.....	61
Tabla 16	Tabla diccionario modelo Perfil Estudiante.	61
Tabla 17	Tabla diccionario modelo Publicación.....	62
Tabla 18	Tabla diccionario modelo Errores.....	63
Tabla 19	Tabla diccionario modelo Académico.....	63
Tabla 20	Prueba caja negra, módulo login	100
Tabla 21	Prueba caja negra, módulo registro de usuarios	101
Tabla 22	Prueba caja negra, módulo estudiante	102
Tabla 23	Prueba caja negra, módulo empleador	102
Tabla 24	Prueba caja negra, módulo docente	102
Tabla 25	Tabla acerca del nivel de reconocimiento en la carga de datos	108

Índice de Figuras

Ilustración 1 Cronograma de actividades en base a la metodología implementada usando JIRA software	32
Ilustración 2 Caso de uso de registro de usuarios	44
Ilustración 3 Caso de uso de inicio de sesión	46
Ilustración 4 Caso de uso de actualización de datos	47
Ilustración 5 Caso de uso de visualización de publicaciones	48
Ilustración 6 Caso de uso de visualización de perfiles	49
Ilustración 7 Caso de uso de publicaciones	49
Ilustración 8 Caso de uso reacciones	50
Ilustración 9 Caso de uso hoja de vida	51
Ilustración 10 Diagrama secuencia de registro de usuarios	52
Ilustración 11 Diagrama de secuencia iniciar sesión	53
Ilustración 12 Diagrama de secuencia de la actualización de datos	54
Ilustración 13 Diagrama de secuencia de publicaciones recomendadas	55
Ilustración 14 Diagrama de secuencia de visualización de perfiles	55
Ilustración 15 Diagrama de secuencia de publicaciones	56
Ilustración 16 Diagrama de secuencia reacciones empresa / docente	57
Ilustración 17 Diagrama de secuencia de reacciones de estudiantes	58
Ilustración 18 Diagrama de secuencia de hoja de vida	59
Ilustración 19 Diagrama de despliegue de la aplicación	64
Ilustración 20 Preprocesamiento de texto	66
Ilustración 21 Algoritmo para eliminación de información personal	67
Ilustración 22 Algoritmo para obtener palabras relevantes	68
Ilustración 23 Algoritmo para eliminar palabras repetidas	69
Ilustración 24 Algoritmo para obtener un archivo de AWS	69
Ilustración 25 Algoritmo para extraer el texto de un currículum	70
Ilustración 26 Algoritmo para extraer el texto de una imagen	71
Ilustración 27 Algoritmo para extraer imágenes de un pdf	72
Ilustración 28 Algoritmo para identificar una imagen de carné en la hoja de vida	74
Ilustración 29 Algoritmo para extraer la similitud entre una publicación y un currículum	75
Ilustración 30 Recepción y envío de datos desde nodejs a python y de python a nodejs	76
Ilustración 31 Algoritmo para analizar la hoja de vida del estudiante	77
Ilustración 32 Algoritmo para cerrar sesión	79
Ilustración 33 Algoritmo para iniciar sesión	79
Ilustración 34 Algoritmo para enviar códigos de verificación al correo	81
Ilustración 35 Algoritmo para la creación de publicaciones	84
Ilustración 36 Algoritmo para la obtención de publicaciones	87
Ilustración 37 Encontrar publicaciones con filtros	87
Ilustración 38 Algoritmo para la creación de perfiles y usuarios en mongodb	88

Ilustración 39 Algoritmo para obtener los usuarios por un identificador en específico	90
Ilustración 40 Algoritmo para conexión a mongodb.....	90
Ilustración 41 Servicios generales para envío de peticiones a nodejs (Backend)	91
Ilustración 42 Algoritmo para la protección y control de rutas.....	92
Ilustración 43 Algoritmo para la implementación de la información en el panel del estudiante ...	93
Ilustración 44 Algoritmo para control de envío de correos masivos.....	94
Ilustración 45 Algoritmo para la obtención de datos de inicio de sesión para la parte del cliente	95
Ilustración 46 Prueba de carga 200 usuarios	96
Ilustración 47 Prueba de carga 500 usuarios	98
Ilustración 48 Histograma de tiempos de respuesta prueba de carga 500 usuarios	98
Ilustración 49 Prueba de estrés 500 usuarios con 20 peticiones	99
Ilustración 50 Histograma de prueba de estrés 500 usuarios 20 solicitudes cada uno	100
Ilustración 51 Imagen de referencia grupo Bolsa de trabajo pasantía.....	103
Ilustración 52 Imagen de oferta laboral 1.....	104
Ilustración 53 Imagen de oferta laboral 2.....	105
Ilustración 54 Imagen de oferta laboral 3.....	105
Ilustración 55 Imagen de oferta laboral 4.....	106
Ilustración 56 Imagen de oferta laboral 5.....	106

RESUMEN

En este trabajo se busca desarrollar un aplicativo web personalizado para los estudiantes del campos sur de la carrera de ingeniería en ciencias de la computación de la Universidad Politécnica Salesiana de la sede Quito, con el objetivo de mejorar la gestión de la bolsa de trabajo y adaptarla a las demandas individuales de cada alumno dependiendo de su hoja de vida o perfil académico; así pudiendo obtener un resultado cualitativo entre la similitud de todas las publicaciones realizadas en el sistema con la información personal, laboral, académica, etc.; de los estudiantes registrados en el aplicativo.

Para lograr este objetivo, se plantea implementar una metodología experimental utilizando algoritmos de machine learning y técnicas de procesamiento natural del lenguaje para examinar grandes conjuntos de datos de estudiantes y empresas previamente registrados en el sistema. De esta manera, se desarrollará un aplicativo web con un sistema de recomendación altamente personalizado y escalable, que permitirá a estudiantes, docentes y empresas visualizar y postular a ofertas laborales, así como solicitar información de empleo o vinculación basándose en las habilidades y destrezas de cada estudiante.

Esto es especialmente importante, ya que, en el ámbito académico, las prácticas profesionales y la bolsa de trabajo, son un paso fundamental en el inicio de la vida profesional de los estudiantes. Debido a esto, se busca lograr un proceso rápido, cómodo y dinámico que asegure la eficiencia y precisión del sistema, lo que puede impactar significativamente la competitividad y el éxito del negocio.

Palabras clave: Bolsa de trabajo, sistema de recomendación, prácticas preprofesionales, procesamiento natural del lenguaje, inteligencia artificial.

ABSTRACT

This project aims to create a customized web application for students at the Salesian Polytechnic University's Quito campus, specifically the south campus. The objective is to enhance the job board management system and tailor it to individual students' needs, considering their resumes and academic profiles. By doing so, the goal is to generate a qualitative assessment that measures the similarity between all the postings in the system and the personal information, work experience, academic achievements, and more of registered students using the application.

To accomplish this goal, our proposal entails implementing an experimental methodology that leverages collaborative learning and natural language processing. This approach will enable the analysis of extensive datasets containing information about students and companies already registered in the system. Through this process, we will develop a web application equipped with a highly personalized and scalable recommendation system. This system will empower students, teachers, and companies to easily view and apply for job opportunities, as well as request employment information tailored to each student's unique skills and abilities.

It holds particular significance because, in the academic field, professional internships and job placement are a fundamental step in the beginning of students' professional lives. Based on this, we seek to achieve a fast, comfortable, and dynamic process that ensures the efficiency and accuracy of the system, which can significantly impact the competitiveness and success of the business.

Keywords: Job placement, recommendation system, pre-professional internships, natural language processing, artificial intelligence.

INTRODUCCIÓN

En la actualidad, encontrar trabajo presenta un obstáculo importante para los estudiantes universitarios, según indica (INEC, 2023). A nivel nacional, esta encuesta revela que solo el 34.6% de las personas en general logran obtener un empleo adecuado. Este porcentaje refleja la disminución considerable de oportunidades para los estudiantes, ya que carecen de la experiencia necesaria y se enfrentan a una creciente demanda laboral. Esto dificulta considerablemente su búsqueda de empleo remunerado.

Además, según la encuesta realizada por parte de la (FEUPS - Computación, 2023), la plataforma de empleo de la UPS es muy poco usada, teniendo que el 86% de los estudiantes ni siquiera intentan abrir la aplicación debido a que no se ajusta a sus necesidades y preferencias y un 71% de los encuestados creen que una bolsa de trabajo con inteligencia artificial podría mejorar las posibilidades de encontrar un empleo que se ajusten a sus necesidades.

Para continuar, la falta de un espacio centralizado y el aumento de la competitividad del mercado laboral ha llevado a los estudiantes, docentes y empresas a buscar ayuda para este proceso en diversas fuentes, como foros, redes sociales y aplicaciones de terceros como LinkedIn, Telegram, Multitrabajos, etc.

Por esta razón, se han buscado formas innovadoras de ayudar a los estudiantes en la búsqueda de estas prácticas. En este contexto, el uso de un sistema de recomendación con procesamiento natural de lenguaje es una solución para obtener resultados personalizadas para los estudiantes y sugerencias para las empresas basadas en sus perfiles, donde cada estudiante puede incluir sus proyectos personales, certificaciones, habilidades y cualidades destacadas.

PLANTEAMIENTO DEL PROBLEMA

Antecedentes

En el proyecto de tesis de maestría, (Lugo, 2022) buscó diseñar un modelo de negocio para la selección automatizada de talento humano y la publicación de ofertas de trabajo en Colombia. El estudio utilizó una investigación de tipo mixto con el propósito de analizar datos, incluyendo la alta tasa de desempleo del país del 10,6%, lo que corresponde a 2.644.000 personas sin trabajo, según lo informado por (DANE, 2022). El autor estimó que había 2.200.000 personas que buscan trabajo y 81.725 pequeñas y 19.100 empresas medianas como posibles reclutadores que podrían generar ingresos mensuales de \$130.378.000 al publicar al menos 274.498 puestos de trabajo en el sistema. El startup propuesto ofrecería una solución de software como servicio (SaaS) de bajo costo para la publicación de ofertas de trabajo, contribuyendo a la democratización de la tecnología en Colombia. El estudio implementó algoritmos de la aplicación de aprendizaje automático con el objetivo de agilizar la búsqueda de oportunidades de trabajo, lo que hace que el proceso sea más accesible tanto para empresas públicas como privadas.

El proyecto de investigación que llevó a cabo (Rivas, 2022), en su trabajo de fin de grado se tituló "Uso de técnicas de procesamiento del lenguaje natural (NLP) para sugerir hojas de vida durante el proceso de selección de personal" con el fin de modernizar el proceso de selección de personal mediante inteligencia artificial. Al introducir una serie de datos relacionados con empleos y perfiles laborales en el algoritmo del sistema, este utiliza NLP para analizar y comprender los textos de las hojas de vida, y así generar recomendaciones basadas en la compatibilidad entre los requisitos del empleo, las habilidades y experiencia de los candidatos. A partir de esto se obtuvo que de 4 perfiles el que más se adaptó a la vacante de administrador fue el perfil 2 con una similitud de 0.155929.

El autor concluye que la tecnología de recomendación de currículums puede ser una herramienta beneficiosa en el futuro para aumentar la velocidad en el proceso de selección de personal. Esto proporcionaría a los empleadores una manera más eficiente de identificar a los candidatos más adecuados para una oferta de trabajo específica.

En su tesis de maestría (Ovalle & Roso, 2022), titulada "Evaluación del Proceso para Reclutamiento y Selección Implementando Inteligencia Artificial en la Empresa: AF Servicios de Ingeniería", se destacó cómo la inteligencia artificial afecta a diferentes tipos de procesos, en especial a los de selección, se emplearon diferentes estrategias que facilitaron el objeto de estudio, aplicando estructuras (frameworks) y estándares de buenas prácticas internacionales como la Norma ISO 30408 como foco sobre la aplicación de la IA en los procesos. Se descubrió que la incorporación de inteligencia artificial (IA) en el proceso de reclutamiento y selección de personal resultó en beneficios para la organización al maximizar y optimizar la gestión de recursos, así como al reducir los tiempos y costos asociados. Además, se logró una selección de candidatos de manera objetiva y basada en criterios específicos, por lo que se redujeron los costos por uso en un 80%, así como todos los tiempos de ejecución y análisis de entrevistas personalizadas.

En su trabajo de titulación, (Cushicondor & Edison, 2021), los autores abordan la problemática de la ineficiencia en el proceso manual de gestión de la bolsa de empleo destinada a los graduados de la Universidad Central del Ecuador. Esta ineficiencia puede resultar en la pérdida de oportunidades debido a la falta de interacción directa entre empleadores y postulantes. Para resolver esta situación, se diseñó e implementó un sistema flexible que se encarga de gestionar la bolsa de empleo para los graduados, utilizando la metodología ágil Kanban con el objetivo de establecer conexiones entre la universidad, los sectores productivos y las instituciones tanto públicas como privadas. Como

resultado, se logra un proceso de contratación optimizado que requiere una inversión mínima de tiempo y esfuerzo, aunque no cuantificable. Concluyendo que el sistema permite a los graduados anexar sus hojas de vida en formato pdf, lo que permite a los empleadores revisar detalladamente si cumplen con las competencias laborales solicitadas.

(Hernández, 2018), en su estudio titulado "Sistema de recomendación de ofertas de empleo y cursos de formación mediante conocimiento de fuentes sociales y web", se observó que la forma en que las personas interactúan está experimentando cambios significativos, siendo las redes sociales un ejemplo destacado de este cambio. En consecuencia, trabajar directamente con los datos se ha vuelto complicado, ya que desempeñan un papel crucial en la actualidad. Por lo tanto, resulta esencial avanzar en el desarrollo de sistemas informáticos capaces de analizar estos datos y proporcionar recomendaciones para obtener las mejores opciones posibles. Con esto en mente, los investigadores propusieron un sistema de recomendación que sugiere a los candidatos las mejores ofertas de empleo y cursos de formación que se ajusten a sus necesidades y preferencias.

Usando tecnologías relacionadas con Inteligencia Empresarial y Big Data, recopilando información relevante acerca del empleo desde fuentes sociales y sitios web relacionados, procesándola y, finalmente, realizando recomendaciones de las ofertas de empleo. Teniendo como resultado que, al realizar la predicción de un candidato con la oferta de trabajo, el algoritmo User based Collaborative Filtering (UBCF), sus medidas de error son de 2.27583, las cuales son menores que el método IBCF, siendo el indicador más importante, concluyendo que el sistema de recomendaciones propuesto se muestra útil, ya que proporciona recomendaciones coherentes sobre ofertas de trabajo y cursos de capacitación para aquellos interesados en ellos.

El problema de estudio abordado en este tema es la necesidad de implementar una plataforma web mediante un manejo personalizado de la bolsa de trabajo para estudiantes en cuanto a prácticas preprofesionales. Hasta el momento, hay muy pocas plataformas web que ofrezcan una experiencia personalizada, lo que dificulta la búsqueda de prácticas preprofesionales adecuadas. Por ello, (Rosado & Espinoza, 2015), proponen que el uso de diferentes técnicas y herramientas de aprendizaje de máquina en la selección de candidatos para prácticas preprofesionales es fundamental. En apoyo a esto, el estudio de (Varguillas, Guffante, Manzano, & Moreno, 2020), indica que las prácticas preprofesionales son fundamentales para el desarrollo de habilidades y creación de competencias en los estudiantes, lo que es esencial para su formación. Por otro lado, estudios de (Romero & Tatiana, 2018) destacan la importancia de la experiencia en prácticas preprofesionales para la contratación de candidatos en el ámbito laboral, lo que respalda la justificación de llevar a cabo este proyecto.

A nivel mundial, el uso de IA enfocada en los diferentes procesos para la gestión de la bolsa de trabajo personalizada es una tendencia nueva y probablemente imparable. Según (Jaser, Petrakaki, Starr, & Oyarbide-Magaña, 2022), la mediación tecnológica se utiliza en un alto porcentaje de empleadores, llegando hasta un 86%, en el proceso de realización de entrevistas de trabajo. Sin embargo, encontrar prácticas preprofesionales adecuadas puede ser una tarea mucho más difícil sin un título académico, ya que el mercado laboral es muy competitivo y los empleadores suelen buscar perfiles específicos para sus vacantes.

En América Latina, según la evaluación desarrollada por él (Banco de Desarrollo de América Latina, 2022), se buscó medir el nivel de preparación después de examinar la implementación de IA en varios ámbitos entre ellos el público, y se determinó que tres países estudiados cuentan con

un nivel medio al respecto: Chile con un 42%, Colombia con un 57% y Uruguay con un 54%. En cada caso, se encontraron distintas fortalezas y debilidades en su aplicación. Cada uno de ellos ha progresado significativamente en la creación de entornos políticos propicios para la asignación exitosa de la IA en el ámbito del sector público. Además, según (Rosas, 2022) con la ayuda de la inteligencia artificial, estas tecnologías pueden ayudar a resolver el desafío del empleo y fortalecer la intermediación laboral que ofrecen los servicios públicos de empleo.

En Ecuador, según la (UPS, 2020), en los últimos siete años, para el sistema de la bolsa de trabajo implementado actualmente, los resultados fueron alentadores, ya que, a escala nacional, se han evaluado alrededor de 6.829 currículums. En los últimos 4 años, se han producido 768 ofertas que se dividen entre pasantías, prácticas preprofesionales y ofertas laborales. Sin embargo, no existe información actualizada de que el sistema siga siendo eficiente para los estudiantes o que se adapte a las nuevas demandas de la tecnología para poder incrementar la tasa de éxito en cuanto a postulaciones de empleo o vinculación en este 2023.

A partir de lo visto, el presente proyecto plantea responder a la siguiente problemática: ¿Qué tan efectivo sería el aplicativo web para sistemas de recomendación de prácticas preprofesionales en la bolsa de trabajo para los estudiantes de la UPS? Un aplicativo web de sistemas para recomendación de prácticas preprofesionales puede ser muy efectivo para ayudar a los estudiantes a encontrar oportunidades de trabajo personalizadas y relevantes para ellos. Los sistemas de recomendación se basan en algoritmos que utilizan los datos de los usuarios para ofrecer recomendaciones personalizadas. En este caso, los datos personales de los estudiantes, como su experiencia previa, habilidades, intereses y preferencias laborales, se pueden utilizar para ofrecer recomendaciones de prácticas preprofesionales que se ajusten mejor a sus necesidades. Además,

un aplicativo web de sistemas de recomendación con aprendizaje automático para las prácticas preprofesionales puede ser muy conveniente y eficiente para los estudiantes, ya que les permite acceder a una amplia variedad de oportunidades de trabajo desde una sola plataforma, ahorrándoles tiempo, esfuerzo de búsqueda para aplicar manualmente a cada una de ellas.

Existen múltiples causas que contribuyen a la problemática descrita, como: i) Los estudiantes enfrentan dificultades para encontrar prácticas preprofesionales que se ajusten a sus habilidades, intereses y expectativas. ii) Los sistemas guardan información confidencial de los estudiantes y empresas, lo que representa un riesgo de seguridad significativo si estos datos son accesibles para terceros no autorizados.

iii) No existe un sistema interactivo, inteligente, rápido, preciso y enfocado en las necesidades de los estudiantes o de la empresa. iv) Falta de un espacio centralizado, como foros, donde los estudiantes puedan estar al tanto de las novedades en cuanto a noticias y ofertas laborales.

Esta problemática puede tener diversos efectos tanto directos como indirectos en los estudiantes, así como a nivel de empresas o en el ámbito laboral en general. i) Puede darse el caso de que los estudiantes no tengan la posibilidad de adquirir de manera efectiva habilidades y experiencia en su campo, lo que puede limitar su potencial para el éxito profesional en el futuro. ii) La falta de seguridad en los sistemas que almacenan información confidencial de los estudiantes puede tener consecuencias negativas para la privacidad y la seguridad, incluyendo el robo de identidad, el fraude y el daño a la reputación. iii) Esta situación puede generar frustración y desperdicio de tiempo, lo que a su vez puede tener una repercusión desfavorable en la productividad y desempeño tanto de los estudiantes como de las empresas involucradas. iv) Puede limitar las opciones de carrera y afectar la capacidad de los estudiantes para encontrar trabajo en su campo de elección.

Con el fin de abordar este problema, se ha planteado la creación de una aplicación web que funcione como un sistema de recomendación para prácticas preprofesionales en la bolsa de trabajo. Este enfoque está dirigido específicamente a los estudiantes, brindándoles la posibilidad de realizar búsquedas personalizadas y precisas basadas en sus propias habilidades y aptitudes.

Justificación

En la actualidad, muchos estudiantes se enfrentan a varios desafíos al momento de encontrar un trabajo o prácticas preprofesionales que se ajusten a su perfil académico, según (Ministerio de Trabajo Ecuador, 2021), en relación con la calidad del empleo, se evidencia que los jóvenes que tienen empleos, el 67,9% no son de calidad, mientras que el 9,4% se encuentra desempleado. Esta misma tendencia se observó en marzo de 2021, donde el 64,9% de los jóvenes tenían empleos no plenos y el 9,6% se encontraba desempleado.

Todo esto se debe a varios factores como a la falta de una búsqueda eficaz de empleo con base a sus necesidades o perfil académico, la falta de información actualizada, la dificultad para identificar las empresas que ofrecen este tipo de oportunidades y la falta de conocimiento sobre los requisitos y habilidades necesarios para postular a estas prácticas.

El desarrollo de un aplicativo para sistemas de recomendación para prácticas preprofesionales y bolsa de trabajo permitirá a los estudiantes tener acceso a una plataforma que les brinde información actualizada sobre las empresas que ofrecen laborales, así como también les permita seleccionar y postular a aquellas que mejor se adapten a sus habilidades y preferencias.

Por lo tanto, la implementación de un sistema de recomendación personalizado, adaptado al perfil y preferencias individuales de cada estudiante, ofrecerá una experiencia más eficiente y efectiva.

Además, esto permitirá a los estudiantes ahorrar tiempo al buscar empleo o prácticas preprofesionales.

Los estudiantes pueden enfrentar problemas al momento de encontrar opciones que se adecuen a sus habilidades y preferencias laborales. Una aplicación web de sistemas de recomendación de prácticas preprofesionales puede ser una solución efectiva para conectar a los estudiantes con empresas y organizaciones que ofrecen oportunidades de prácticas preprofesionales que se ajusten mejor a sus necesidades y habilidades.

Para terminar, el aplicativo web puede ser muy útil para ayudar a los estudiantes a encontrar oportunidades de prácticas preprofesionales que se ajusten a sus necesidades y habilidades. Esto les permitiría adquirir experiencia laboral relevante y mejorar sus habilidades, lo que a su vez podría aumentar sus posibilidades de conseguir empleo después de graduarse.

OBJETIVOS

Objetivo general

Desarrollar un aplicativo web para sistemas de recomendación de prácticas preprofesionales en la bolsa de trabajo para los estudiantes de la Universidad Politécnica Salesiana.

Objetivos específicos

- Desarrollar un aplicativo web para un sistema de recomendación personalizado, mediante el uso de algoritmos de aprendizaje automático, que analice la información del currículum de cada estudiante y genere publicaciones recomendadas de forma cualitativa en función de sus perfiles.

- Establecer un sistema de autenticación seguro y eficiente que permita a los estudiantes, docentes, empleadores y administradores acceder a la plataforma. Cada rol tendrá sus respectivos permisos y funcionalidades específicas, garantizando la privacidad y la correcta gestión de la información.
- Diseñar una interfaz intuitiva y atractiva que permitan a los estudiantes interactuar con las publicaciones, expresando su interés, compartiéndolas, guardándolas o descargándolas hacia el computador.

ALCANCE

Alcance general del aplicativo

Para el objetivo A: “Desarrollo de aplicativo web para sistemas de recomendación de prácticas preprofesionales para la bolsa de trabajo en los estudiantes de la carrera de computación”

1. El aplicativo web estará dirigido exclusivamente a los estudiantes y docentes de la carrera de computación de la UPS (Ecuador).
2. El aplicativo web estará disponible únicamente en el idioma español.
3. El sistema no contará con opciones de accesibilidad específicas, como el modo de luz nocturna, el aumento de tamaño de texto, etc.

Para el objetivo B: “Desarrollar un aplicativo web para un sistema de recomendación personalizado, mediante el uso de algoritmos de aprendizaje automático, que analice la información del currículum de cada estudiante y genere publicaciones recomendadas de forma cualitativa en función de sus perfiles.”

1. La efectividad del sistema de la recomendación estará condicionada por la cantidad, veracidad y calidad de los datos disponibles que ingresen cada uno de los estudiantes en la plataforma.
2. La carga inicial de información estará sujeta a la disponibilidad y calidad de los datos a procesar, así como a la conexión a internet de cada estudiante.

Para el objetivo C: “Establecer un sistema de autenticación seguro y eficiente que permita a los estudiantes, docentes, empleadores y administradores acceder a la plataforma. Cada rol tendrá sus respectivos permisos y funcionalidades específicas, garantizando la privacidad y la correcta gestión de la información.”.

1. La implementación estos protocolos son medidas para salvaguardar la información principal de los estudiantes como claves de acceso. No obstante, en base a esto no se puede garantizar que el sistema esté protegido en cuanto a ataques de: Inyección de SQL, phishing, secuestro de sesión, ataque de denegación de servicio (DDoS), Cross-site scripting (XSS), etc.
2. Cada rol tendrá permisos y funcionalidades específicas establecidas previamente, sin permitir la personalización o modificación de los permisos por parte de los usuarios.
3. La política de gestión de información y control de datos se centrará en asegurar las pautas de seguridad fundamentales, como el acceso restringido y el cifrado de datos, con el fin de salvaguardar la información de los usuarios contra posibles accesos no autorizados.

Para el objetivo D: “Diseño una interfaz intuitiva y atractiva que permitan a los estudiantes interactuar con las publicaciones, expresando su interés, compartiéndolas, guardándolas o descargándolas hacia el computador”.

1. La interfaz permitirá a los estudiantes expresar su interés en las publicaciones a través de la función de "Me interesa" o "Guardar", pero no incluirá opciones para realizar comentarios o calificaciones detalladas.
2. Los estudiantes podrán compartir las publicaciones en redes sociales seleccionadas previamente, pero no se incluirán opciones para enviar las publicaciones por correo electrónico o mensajes directos.
3. La interfaz estará optimizada para su visualización en dispositivos de escritorio y dispositivos móviles, pero no se garantizará la compatibilidad con todos los tipos de navegadores o sistemas operativos.
4. Se consideró la implementación de una funcionalidad de comunicación entre usuarios y docentes a través de un sistema de mensajería web (chat). Sin embargo, debido a limitaciones de tiempo, esta Característica no pudo ser perfeccionada y no se encuentra completamente desarrollada en la versión actual del proyecto. Se evaluará la viabilidad de incluir esta funcionalidad en futuras etapas de desarrollo.

Roles de usuarios

El sistema contará con 4 roles:

- Administrador
- Docente
- Estudiante
- Empresa

Módulos

El principal objetivo es entregar un prototipo de bolsa de trabajo aplicada inteligencia artificial que sirva de apoyo para que los estudiantes puedan encontrar prácticas preprofesionales o empleo de una manera mucho más eficiente, siendo los entregables de la primera versión:

Módulos generales

- **Inicio de Sesión:** Los usuarios podrán ingresar utilizando su correo electrónico institucional y contraseña, o bien, podrán optar por la opción de inicio de sesión con su cuenta de Microsoft.
- **Registro:** Los usuarios con rol de estudiante tendrán la opción de registrarse en el sistema utilizando tres métodos diferentes: un proceso eficaz, completo y rápido. Por otro lado, los usuarios con roles distintos al de estudiante tendrán una única forma de registro.
- **Recuperación de contraseña:** Los usuarios podrán acceder un mecanismo seguro de recuperación de contraseña a través del correo electrónico, que permitirá a los usuarios restablecer su contraseña en caso de olvido. Esta funcionalidad asegurará la conveniencia y la seguridad de los usuarios al acceder al sistema y gestionar sus credenciales.
- **Perfil:** Los usuarios podrá acceder a la información personal y de contacto de su cuenta y podrá modificar los datos relevantes.

Módulo del estudiante

- **Publicaciones:** El usuario podrá acceder a las publicaciones realizadas en el sistema por los docentes o la empresa y podrá visualizar de manera cualitativa el resultado del algoritmo de recomendación.

- **Interacciones:** El usuario podrá reaccionar (rechazar, descargar, compartir, dar me interesa y guardar) las publicaciones.
- **Filtros de búsqueda:** El usuario podrá filtrar toda la información como pasantías, vinculación, bolsa de empleo en base a la ubicación, jornada de trabajo, etc.
- **Aceptaciones:** El usuario podrá acceder a las aceptaciones (Coincidencias entre la empresa y el estudiante).

Módulo del docente

- **Publicaciones:** El docente podrá publicar avisos de prácticas preprofesionales o bolsa de trabajo mediante texto plano o como imagen.
- **Interacciones:** El docente podrá realizar y visualizar las interacciones con base a las publicaciones realizadas por el o por otros usuarios con rol docente y empleador (compartir, me gusta, guardar).

Módulo de la empresa

- **Perfiles:** La empresa podrá acceder a la información de los perfiles recomendados de los estudiantes en base a las publicaciones realizadas por el mismo.
- **Publicaciones:** La empresa podrá publicar avisos de prácticas preprofesionales o bolsa de trabajo.
- **Aceptaciones:** La empresa podrá acceder a las aceptaciones (Coincidencias entre la empresa y el estudiante).
- **Interacciones:** La empresa podrá realizar interacciones para cada uno de los perfiles de los estudiantes como (descargar, omitir, regresar, aprobar, guardar).

Módulo del administrador

- **Usuarios:** El administrador puede realizar acciones CRUD para los usuarios registrados en el sistema.
- **Publicaciones:** El administrador puede realizar acciones CRUD para las publicaciones en el caso de que alguna infrinja o haga mal uso de las políticas de las publicaciones.

MARCO METODOLÓGICO

Metodología de desarrollo

Tabla 1 Stack de desarrollo MEAN

Stack	Características
MongoDB	Se trata de un sistema de almacenamiento de datos que es compatible con diversas plataformas y tiene una capacidad de escalabilidad notable, lo que implica que puede manejar el almacenamiento de información de cualquier tamaño sin afectar el funcionamiento continuo del sistema.
Express.js	Se encarga de la creación del (backend) de las aplicaciones web, brindando mayor seguridad y velocidad, al tiempo que simplifica el proceso de desarrollo.
Angular	Es una framework de JavaScript ampliamente reconocida que se distingue por su extensa colección de módulos útiles y características destacables, incluyendo el enlace bidireccional de datos, entre otras.
Node.js	Es un framework de JavaScript del lado del servidor que facilita la creación de aplicaciones de software escalables con un rendimiento extremadamente rápido.

Basado en las Características señaladas por (Pathak, 2021), se puede afirmar que el enfoque de creación de un aplicativo web utilizando el stack MEAN, es una excelente elección para el desarrollo de un sistema de recomendación escalable y de alto rendimiento. Al aprovechar el uso de JavaScript para el cliente (frontend) y para el servidor (backend), junto con una base de datos no-sql, se logra una arquitectura flexible y altamente escalable, especialmente en términos de escalabilidad horizontal. Esto permite manejar grandes volúmenes de datos y asegurar un rendimiento eficiente en el sistema de recomendación.

Metodología Ágil

Tabla 2. Tabla comparativa de metodologías de desarrollo

Metodología	Descripción	Ventajas	Fases
Scrum	Se utiliza en equipos que trabajan en proyectos de alta complejidad. Se basa en un proceso estructurado y un marco de trabajo que permite abordar de manera efectiva los desafíos que surgen en proyectos complejos.	<ul style="list-style-type: none"> - Se agiliza el proceso, ya que la entrega del sprint es frecuente. - Menor probabilidad de errores porque se encuentra en plena observación del proyecto. 	<ul style="list-style-type: none"> - Planificación del sprint. - Reunión del equipo. - Revisión del sprint. - Reflexión del sprint.
Kanban	Kanban también pone énfasis en la mejora continua mediante la medición y análisis de métricas importantes, como el tiempo promedio de finalización de tareas o la capacidad de respuesta del equipo.	<ul style="list-style-type: none"> - No se especifica un periodo. - Reduce ampliamente los tiempos de desarrollo. - Mayor calidad. 	<ul style="list-style-type: none"> - Establecimiento del proceso de trabajo del proyecto. - Representación visual de las etapas del ciclo de producción. - Establecimiento de prioridades en las tareas. - Monitoreo y control del flujo de trabajo.
Lean	Centrado en la reducción de desperdicios y el constante mejoramiento.	<ul style="list-style-type: none"> - Identificación de la propuesta de valor. - Mapeo del flujo de valor - Establecimiento de un flujo de trabajo continuo. 	<ul style="list-style-type: none"> - Creación de valor - Flujo con base al valor - Eliminación de residuos. - Retrospectiva de mejora continua
XP	Enfocado en la entrega de software de alta calidad a través de prácticas de desarrollo ágil y en la potencia de relaciones interpersonales.	<ul style="list-style-type: none"> - Planificación siempre organizada. - Errores poco probables. - Solución de errores centrado en el equipo. - Manejo y actualización de versiones. - Dispone una forma de trabajo más eficiente. 	<ul style="list-style-type: none"> - Planeación. - Diseño. - Desarrollo. - Evaluaciones. - Lanzamiento.

Con base a las Características recopiladas de (Studocu, 2021), para la realización del sistema de recomendación se optó por SCRUM, ya que es la metodología ideal que se divide en iteraciones o

sprints de una duración específica, en este caso cada una o dos semanas. En cada sprint, se van a planificar las tareas a realizar y se establecen los objetivos a cumplir. Al final de cada sprint, se realiza una revisión de lo que se ha logrado y se retroalimenta el proceso para mejorar en la siguiente iteración.

Herramienta: Jira Software

Fases:

1. **Planificación del proyecto:** Se define el objetivo del proyecto, se establecen los requerimientos del aplicativo web de bolsa de trabajo con inteligencia artificial y se crea el Product Backlog, que es una lista de todas las funcionalidades y Características del aplicativo.
2. **Sprint Planning:** En esta fase se seleccionan las tareas más importantes del Product Backlog para el sprint y se establece un objetivo para el sprint. En este caso el tutor toma el cargo de SCRUM MASTER donde verifica el tiempo que tomará cada tarea y la asignación de ellas.
3. **Sprint:** Durante el sprint, el equipo de desarrollo se dedica a completar las tareas asignadas para ese período. Cada jueves, el equipo se reúne para revisar el avance del sprint y detectar posibles problemas.
4. **Revisión del Sprint:** Al final del sprint, el equipo de desarrollo se reúne para revisar el trabajo completado durante el sprint y para demostrar las funcionalidades que se han desarrollado. También se discute lo que se podría mejorar para el próximo sprint.
5. **Entrega del producto:** Cuando se hayan completado todos los sprints, el producto se entrega al tutor. Además, se harán pruebas exhaustivas antes de la entrega y se debe

garantizar que el aplicativo cumple con todos los requerimientos especificados en el Product Backlog.

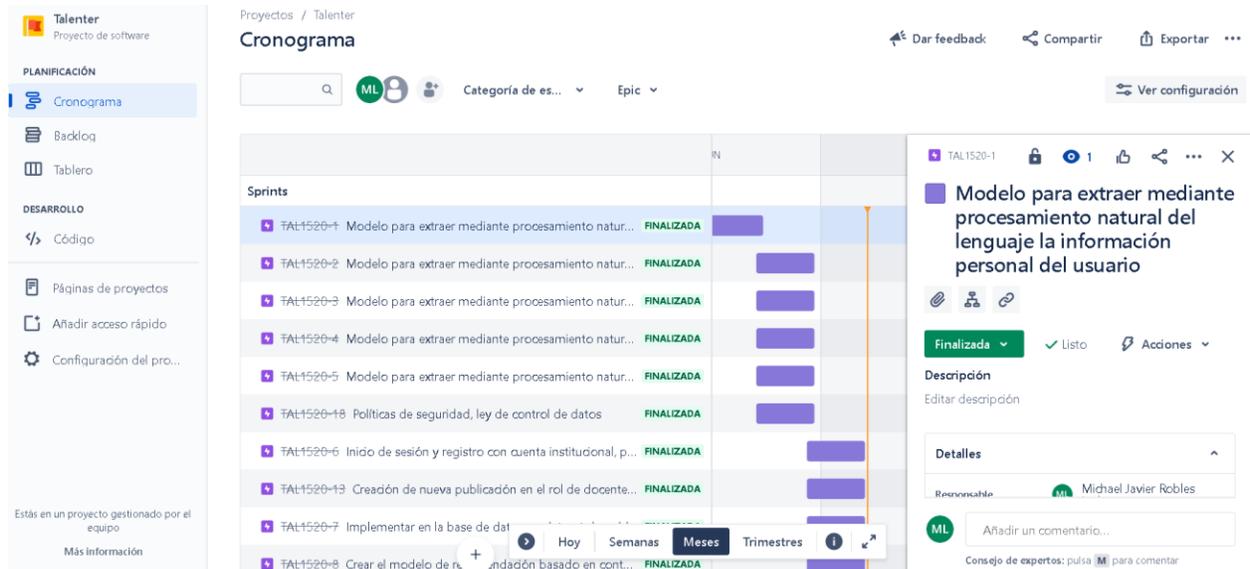


Ilustración 1 Cronograma de actividades en base a la metodología implementada usando JIRA software

Metodología De Investigación Experimental

Para el desarrollo de sistemas de recomendación de empleo altamente personalizados se considera implementar machine learning y nlp con ayuda de Python, por lo que se pretende realizar una metodología experimental:

- 1. Identificación de las necesidades del usuario:** Se definen los requisitos y necesidades de los estudiantes de la carrera de computación como de las empresas que buscan candidatos para prácticas preprofesionales.
- 2. Análisis y recuperación de datos:** Se analizar los datos relevantes de los estudiantes a través de sus cuentas y hojas de vida previamente registradas en el aplicativo web, como su historial académico, habilidades y conocimientos, intereses y preferencias, y se utilizarían

técnicas de análisis de datos para identificar patrones y relaciones que permitan identificar las características más relevantes para la recomendación de prácticas.

- 3. Entrenamiento y diseño del modelo de inteligencia artificial:** Se diseñaría un modelo de inteligencia artificial que pueda procesar y analizar los datos recopilados y proporcionar recomendaciones personalizadas para cada estudiante. El modelo se entrenaría utilizando técnicas de aprendizaje automático y se optimizaría para mejorar su precisión y eficacia mediante técnicas de procesamiento natural del lenguaje.
- 4. Evaluación del modelo:** Para verificar su eficacia y precisión en la recomendación de prácticas preprofesionales. Se utilizarían técnicas de validación cruzada y otras métricas que nos ayudarán para evaluar la calidad de las recomendaciones.
- 5. Implementación y pruebas:** Implementar el modelo en el aplicativo web y realizar pruebas exhaustivas para garantizar su correcto funcionamiento.
- 6. Monitoreo y ajustes:** Monitorear el rendimiento del modelo y hacer ajustes necesarios en cuando se presenten problemas o se requieran mejoras.
- 7. Mantenimiento y actualización:** Mantener el modelo y el aplicativo web actualizado con las últimas técnicas y tecnologías de inteligencia artificial para garantizar su eficacia y eficiencia a lo largo del tiempo.

CAPITULO 1: ESTADO DEL ARTE

Marco teórico o referencial

1.1.Algoritmos de aprendizaje automático: Estos algoritmos son capaces de automejorarse a partir de su experiencia, de manera que la mejora no depende únicamente de las habilidades del programador. Es crucial el proceso de adquisición de experiencia que el algoritmo experimenta al procesar una creciente cantidad de datos, ajustándose automáticamente según las propiedades de dichos datos (Khan Academy, 2023).

1.2.Big Data: Hace referencia aun gran conjunto de datos, ya sea individuales o combinados, que poseen características como tamaño y volumen que dificultan su obtención, manejo o análisis utilizando tecnologías convencionales (Powerdata, 2022).

1.3.Estándares: Según la definición de la ISO, es la documentación que contiene especificaciones técnicas detalladas para ser manipuladas de manera consistente como políticas o directrices. El propósito de las normas es asegurar que los servicios o productos que dispongan se cumplan con su objetivo de acuerdo con las normas establecidas (MINTIC Colombia, 2021).

1.4.Frameworks: Es un conjunto de utilidades diseñadas para simplificar la solución de problemas comunes que pueden surgir durante el proceso de programación. Estas herramientas aceleran el desarrollo al ofrecer facilidades como la organización del código y la colaboración en equipo dentro de un proyecto. Su objetivo principal es aumentar la eficiencia y la optimización en el trabajo de programación (Juan de Assembler Institute, 2023).

1.5.Artillery: Herramienta utilizada para pruebas de carga y estrés que se utiliza para probar la capacidad de respuesta de aplicativos web, API y servicios en la nube. Puede simular cargas de trabajo de alta intensidad y medir el rendimiento del sistema bajo diferentes condiciones de carga (Aguilera, 2018).

1.6. Inteligencia Artificial (IA o AI): Tecnología que busca emular los procesos del desarrollo en base a la creación y desarrollo de algoritmos computacionales dinámicos. En resumen, se trata de un esfuerzo por lograr que las computadoras piensen y actúen de manera similar a los seres humanos (Netapp, 2019).

1.7. Inteligencia Empresarial (Business Intelligence): Conjunto de procedimientos necesarios que proporcionan una solución informática que se encargue de analizar el rendimiento de una empresa. Este análisis proporcionará información clave para optimizar su funcionamiento a través de la toma de decisiones adecuadas (Jiménez, 2023).

1.8. Prácticas preprofesionales: Se refiere a que los estudiantes adquieran las habilidades y competencias necesarias para desenvolverse en el ámbito laboral correspondiente a su campo de estudio, a través de diversas actividades que deben solucionar. (Unir Vive, 2021).

1.9. Protocolos de encriptación: Ofrecen conexiones seguras que permiten a dos partes comunicarse de forma confidencial y aseguran la integridad de los datos transmitidos (IBM, 2023).

1.10. NLP: Es una tecnología que se basa en machine learning y que capacita a las computadoras para comprender el lenguaje humano (Amazon Web Services, Inc., 2023).

1.11. ISO/IEC 9126: El estándar ISO-9126 proporciona un marco para el modelo de calidad del software y consta de un conjunto de características que todos los productos de software deben cumplir (Verity, 2022).

CAPITULO II: ANALISIS Y DISEÑO

2.1.Requerimientos funcionales y no funcionales:

2.1.1. Requerimientos Funcionales

Tabla 3. Requerimientos funcionales

N	CODIGO	DESCRIPCION
1	RF01	Registro de nuevos usuarios El aplicativo web deberá permitir a los estudiantes, registrarse en el sistema y crear un perfil con información relevante sobre sus habilidades, experiencia académica, experiencia laboral, cursos realizados e información personal y de contacto. Para los roles de empleadores, docentes y administradores, deberá permitir registrar su información personal, básica y de interés
2	RF02	Inicio de sesión El aplicativo web deberá permitir a los estudiantes, empleadores, docentes y administradores, iniciar sesión en el sistema para acceder a las publicaciones de la bolsa de trabajo o prácticas preprofesionales de manera segura, utilizando técnicas de encriptación para mantener los datos vulnerables protegidos.
3	RF03	Actualización de datos El aplicativo web deberá permitir a los estudiantes, empleadores, docentes y administradores, actualizar la información de su perfil dependiendo de su rol y de sus permisos.
4	RF04	Visualización de publicaciones recomendadas El sitio web deberá ofrecer a los estudiantes la posibilidad de ver las publicaciones sugeridas y previamente compartidas por los profesores y empresas, de acuerdo con la información de su perfil, junto con una evaluación cualitativa de la recomendación.
5	RF05	Visualización de perfiles recomendados El aplicativo web deberá permitir a las empresas registradas en el sistema visualizar la información de los estudiantes y ver el porcentaje de compatibilidad entre las publicaciones realizadas por la empresa y la hoja de vida.
6	RF06	Publicaciones El aplicativo web deberá permitir a las empresas, docentes registrados en el sistema publicar nuevos avisos de bolsa de trabajo o prácticas preprofesionales, teniendo las opciones de adjuntar una nueva imagen, añadir información relevante como salario, horario, carreras afines a la publicación, descripción, conocimientos requeridos, listado de funciones, requisitos, beneficios, enlaces, etc.
7	RF07	Reacciones El aplicativo web deberá permitir a los estudiantes, empleadores, docentes, reaccionar a las publicaciones, pudiendo colocar "Me interesa", "Compartir", "Guardar", etc.

8	RF08	<p>Hoja de vida</p> <p>El aplicativo web deberá permitir a los estudiantes, en base a la información subida de su hoja de vida, crear de manera rápida y precisa el perfil y crear los modelos para generar predicciones precisas y eficaces.</p>

2.1.2. Requerimientos No Funcionales

Usabilidad: El aplicativo web debe ser fácil de usar e intuitivo para que los estudiantes y empleadores puedan utilizarlo sin dificultades.

Seguridad: El aplicativo web debe ser seguro y proteger la información personal y confidencial de los usuarios.

Responsive: El aplicativo web debe ser en su mayoría adaptable para cualquier pantalla de dispositivo móvil y de escritorio.

Escalable: El aplicativo web debe ser capaz de manejar un gran volumen de usuarios y oportunidades sin perder rendimiento.

2.2.Historias de usuarios

Tabla 4. Historia de usuario registro nuevos usuarios

REQUERIMIENTO	RF01
ACTORES	Invitado
PREREQUISITOS	<ol style="list-style-type: none"> 1. Acceso a internet. 2. El usuario debe tener acceso a su correo electrónico institucional de la UPS o de dominio para recibir el código de confirmación y validar su cuenta. 3. El correo del usuario no debe estar registrado en el sistema. 4. Aceptar los términos y condiciones de la plataforma, que podrían incluir aspectos como la propiedad del contenido, la privacidad de los datos, la moderación de los comentarios, la responsabilidad legal, entre otros.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario ingresa a la opción de Registro. 2. El sistema despliega los componentes de registro. 3. El usuario ingresa el tipo al que pertenece y el correo. 4. El sistema envía un código de verificación de cuenta, al correo ingresado. 5. El usuario ingresa el código de verificación que fue enviado a su correo. 6. El sistema despliega los componentes de actualización de contraseña. 7. El usuario ingresa su contraseña y la confirmación de contraseña. 8. El sistema carga los componentes de información que deben ser completados. 9. El usuario llena todos los campos obligatorios 10. El usuario da clic en finalizar 11. El sistema utiliza procesamiento natural del lenguaje para analizar la hoja de vida 12. El sistema crea un vector con las palabras relevantes o con mayor peso de la hoja de vida. 13. El sistema guarda la hoja de vida en AWS S3 en formato pdf. 14. El sistema crea el usuario en la base de datos mongodb.
FLUJO ALTERNATIVO	<p>3a. Un usuario que intenta registrarse con un rol diferente al suyo desea crear una cuenta, pero el sistema verifica que la dirección de correo electrónico no pertenece al rol indicado. Por ejemplo, un estudiante intenta registrarse como docente, pero el sistema valida el dominio del correo (como "est.ups.edu.ec") y le muestra un mensaje indicando que no puede registrarse, ya que no corresponde a ese rol.</p> <p>3b. El sistema valida que el usuario ingrese datos válidos y complete todos los campos requeridos, caso contrario se despliega un mensaje de error "Completa todos los campos e ingresa datos válidos".</p> <p>3c. El usuario con rol de empresa deberá agregar un RUC válido con lo cual el sistema validará el dígito verificador y validará la longitud del mismo que sea igual a 13 Cadenas. Caso contrario, se desplegará un mensaje de error: "Porfavor completa todos los campos e ingresa datos válidos"</p> <p>3d. El sistema detecta que el correo ingresado ya existe en el sistema y muestra un mensaje de error indicando que ese correo ya está registrado.</p> <p>4a. El usuario no recibe el correo con el código de confirmación, por lo que el sistema muestra un mensaje de error indicando que no se pudo enviar el correo y ofrece opciones para intentar enviar el correo nuevamente o para comunicarse con el soporte técnico.</p>

Tabla 5. Historia de usuario inicio de sesión

REQUERIMIENTO	RF02
ACTORES	Administrador, empleador, estudiante, docente
PREREQUISITOS	<ol style="list-style-type: none"> 1. Acceso a internet. 2. El usuario debe contar con un correo electrónico y una contraseña previamente registrados en el sistema.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario ingresa su correo y contraseña 2. El sistema valida que los datos sean correctos a los que se encuentran en la base de datos. 3. El sistema despliega el panel principal.
FLUJO ALTERNATIVO	<p>2a. El usuario ingresa un nombre de usuario o correo electrónico que no está registrado en el sistema, por lo que el sistema muestra un mensaje de error indicando que no se ha encontrado una cuenta con esa información.</p> <p>2b. El usuario olvida su contraseña, por lo que el sistema muestra una opción para recuperar la contraseña a través de un correo electrónico.</p> <p>2c. Al existir un problema en el servidor o la conexión de red, se despliega un mensaje de error indicando que ha ocurrido un error inesperado.</p>

Tabla 6. Historia de usuario actualización de datos

REQUERIMIENTO	RF03
ACTORES	Administrador, empleador, estudiante, docente
PREREQUISITOS	<ol style="list-style-type: none"> 1. El usuario debe tener una sesión activa en el sistema. 2. El usuario debe tener permisos suficientes para actualizar los datos específicos que desea modificar. 3. El usuario debe conocer los datos que desea actualizar y tener la información necesaria para realizar los cambios en el sistema. 4. El sistema debe contar con una validación de los datos actualizados, para evitar que se ingresen datos incorrectos o incompletos.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario ingresa al perfil de su cuenta 2. El sistema despliega la información personal que se puede modificar y la que no se encuentra disponible. 3. El usuario ingresa los datos que desea actualizar. 4. El usuario da clic en actualizar datos. 5. El sistema actualiza los datos que han sido modificados.
FLUJO ALTERNATIVO	<p>4a. El usuario intenta actualizar un campo con un valor que no cumple con los requisitos establecidos por el sistema, como un formato incorrecto o una longitud inválida. En este caso, el sistema muestra un mensaje de error que indique el problema y solicita al usuario que proporcione un valor válido para el campo.</p> <p>4b. El usuario intenta actualizar un campo que no puede ser modificado debido a restricciones de seguridad o de integridad de los datos. En este caso, el sistema muestra un mensaje de error que no se ha podido actualizar los datos.</p> <p>5a. El sistema experimenta problemas técnicos o de conectividad durante el proceso de actualización de datos, lo que puede provocar la pérdida de información o la interrupción del proceso. En este caso, el sistema muestra un mensaje de error que explica el problema.</p>

Tabla 7. Historia de usuario visualización de publicaciones recomendadas

REQUERIMIENTO	RF04
ACTORES	Estudiantes
PREREQUISITOS	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado e identificado en el sistema para poder acceder a la opción de visualización de publicaciones recomendadas. 2. El sistema debe contar con un conjunto de publicaciones disponibles para mostrar a los estudiantes, ya sea creado por las empresas o docentes.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario ingresa a su cuenta o ingresa a la opción de “inicio” o da clic en el logo de la empresa. 2. El sistema despliega las publicaciones recientes recomendadas para el perfil del usuario.
FLUJO ALTERNATIVO	<p>2a. Si el usuario o el sistema no tiene publicaciones recomendadas, se mostrará un mensaje de que no existen publicaciones disponibles.</p> <p>2b. Si el sistema no puede generar recomendaciones personalizadas para el usuario debido a la falta de información del perfil del usuario, el sistema podría solicitar al usuario que proporcione información adicional sobre sus intereses y preferencias. El sistema mostrará publicaciones populares o relevantes basadas en otros criterios, como la fecha, el tema o la ubicación.</p> <p>2c. Si el sistema detecta un error o problema técnico al mostrar las publicaciones recomendadas, podría mostrar un mensaje de error al usuario.</p> <p>2d. Si el usuario quiere explorar más allá de las publicaciones recomendadas, el sistema proporciona opciones de búsqueda y navegación adicionales, como la búsqueda por ciudad, categorías, horarios, etc.</p>

Tabla 8. Historia de usuario visualización de perfiles recomendados

REQUERIMIENTO	RF05
ACTORES	Empleador, docente
PREREQUISITOS	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado e identificado en el sistema para poder acceder a la opción de visualización de perfiles recomendados de estudiantes. 2. El usuario debe contar con los permisos de acceso necesarios. 3. El sistema debe contar con un algoritmo o método para generar recomendaciones de perfiles de estudiantes que sean relevantes y personalizadas para la empresa u docente. Basado en las publicaciones realizadas y los intereses, habilidades y preferencias, así como en las necesidades y requisitos de las empresas que publican las ofertas de trabajo o prácticas preprofesionales. 4. El sistema debe contar con estudiantes registrados en el sistema.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario con rol empleador o docente ingresa a su cuenta o a la opción de “inicio” o selecciona el botón de “recomendado”. 2. El sistema despliega los perfiles recomendados de los estudiantes en base a las publicaciones realizadas.
FLUJO ALTERNATIVO	<p>2a. Si el usuario no tiene perfiles recomendados, se mostrará un mensaje de que no existen perfiles disponibles.</p> <p>2b. Si el sistema detecta un error o problema técnico al mostrar los perfiles recomendados, mostrará un mensaje de error al usuario.</p> <p>2c. Si el usuario quiere explorar más allá de las publicaciones recomendadas, el sistema proporciona opciones de búsqueda y navegación adicionales, como la búsqueda por ciudad, carreras, categorías o la exploración por perfiles de cada estudiante.</p>

Tabla 9. Historia de usuario creación de publicaciones

REQUERIMIENTO	RF06
ACTORES	Empleador, docente
PREREQUISITOS	<ol style="list-style-type: none"> 1. Conexión a internet 2. El usuario debe estar autenticado e identificado en el sistema con el rol de empleador o docente. 3. El usuario debe contar con los permisos para poder acceder a la opción de crear nuevas publicaciones. 4. Rellenar todos los campos obligatorios con la información que desee añadir a la publicación.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario ingresa a su cuenta o selecciona el botón de inicio. 2. El usuario se dirige a la sección de “Crear nueva publicación” 3. El usuario rellena los datos de la publicación. 4. El usuario selecciona la opción de publicar 5. El sistema se encarga de validar todos los datos ingresados y guarda en la base de datos. 6. El sistema mostrará un mensaje de que se añadió la publicación con éxito.
FLUJO ALTERNATIVO	<p>5a. Si se ha producido un error en los datos ingresados el sistema mostrará un mensaje indicando el motivo del error.</p> <p>5b. Si se ha producido un error al guardar los datos en la base, el sistema mostrará un mensaje de que se ha producido un error inesperado para que lo vuelva a intentar más tarde.</p>

Tabla 10. Historia de usuario reacciones

REQUERIMIENTO	RF07
ACTORES	Empleador, estudiante, docente
PREREQUISITOS	<ol style="list-style-type: none"> 1. Conexión a internet 2. El usuario debe estar autenticado e identificado en el sistema. 3. El usuario debe tener el rol adecuado para poder reaccionar.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario ingresa a su cuenta o selecciona el botón de inicio. 2. El usuario visualiza las publicaciones o perfiles recomendados dependiendo del rol. 3. El usuario quiere compartir, guardar o le interesó el contenido de la publicación. 4. El usuario selecciona “Me interesa”, “Compartir” o “Guardar” para publicaciones o “Cancelar”, “Seleccionar”, “Guardar” para los perfiles. 5. El sistema realizará la acción dependiendo de su selección y cambiará el contador.
FLUJO ALTERNATIVO	<p>2a. El sistema no despliega ninguna publicación o perfil por lo que no se deberá visualizar ninguna reacción.</p> <p>3a. El sistema desplegará un diálogo en el caso de “compartir” donde seleccionará una plataforma externa a la que desee enviar.</p> <p>3b. El sistema desplegará un diálogo en el caso de “guardar” la publicación en una sección dedicada para acceder a ella más tarde.</p> <p>5a. En el caso de que haya ocurrido un error en el sistema y no se haya logrado reaccionar, el botón de la reacción no cambiará el contador +1.</p>

Tabla 11. Historia de usuario hoja de vida

REQUERIMIENTO	RF08
ACTORES	Estudiantes
PREREQUISITOS	<ol style="list-style-type: none"> 1. El usuario puede estar en el proceso de registro o autenticado en el sistema con el rol de estudiante. 2. Documento en formato .pdf de su hoja de vida 3. Deberá tener información relevante acerca de su información personal, académica y laboral.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario se dirige a la opción de “Mi perfil” o se encuentra en la pantalla de “Carga de datos” del proceso de registro. 2. El usuario selecciona la opción de cargar Curriculum 3. El sistema genera un vector con las palabras relevantes en base al análisis de su hoja de vida.
FLUJO ALTERNATIVO	<p>2a. El sistema verifica que el archivo ingresado se encuentre en formato .pdf, caso contrario envía un mensaje de error acerca del tipo de archivo inválido.</p> <p>2b. El sistema verifica el tamaño del archivo y si supera los 3mb, le impedirá subir a la plataforma, por lo que puede optimizar el pdf.</p> <p>3a. El sistema envía un mensaje de error al usuario si es que el proceso de extracción de la información, utilizando procesamiento natural de lenguaje falló al momento de actualizar los datos de la hoja de vida.</p>

2.3. Diagramas de casos de uso

2.3.1. RF01 Registro de usuarios

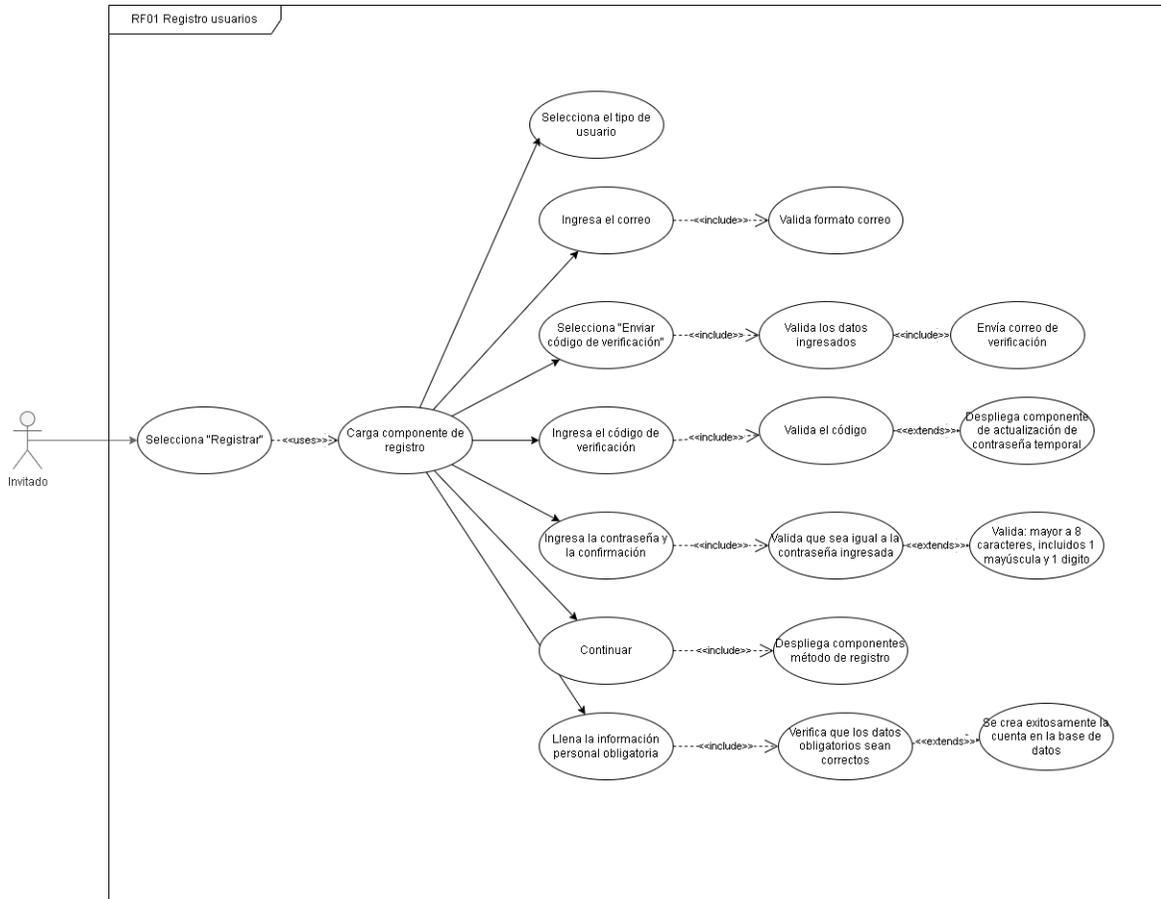


Ilustración 2 Caso de uso de registro de usuarios

El usuario invitado podrá registrarse en talenter.app al momento de seleccionar “**Registro**” en el módulo principal que es la primera ventana que aparece al desplegar la plataforma.

Deberá seleccionar el rol correspondiente al que pertenece (estudiante, docente, empleador), para continuar, deberá ingresar el código de verificación de cuenta que será enviado al correo electrónico ingresado. Si el código coincide con el del correo, deberá realizar la actualización de su contraseña temporal la cual deberá contener al menos 8 Caracteres y máximo 32. Si los valores

son ingresados correctamente y el tipo de rol elegido es “**Estudiante**”, tiene 3 diferentes formas de registro: la eficiente, la completa y la rápida. Cada una varía según el tiempo, precisión y disponibilidad de la información por parte del usuario.

Forma eficiente:

1. Carga automática de datos desde la hoja de vida: El usuario deberá ingresar su hoja de vida en formato pdf y no deberá superar los 3mb de tamaño. Los datos que se encuentran en la hoja de vida como nombres, apellidos, cédula, educación, proyectos, experiencia laboral, habilidades, etc. Se cargará automáticamente en la cuenta del estudiante como información seccionada o agrupada para poder posteriormente usarla para el análisis de similitud con las publicaciones.
2. Tiempo rápido en la configuración y personalización de los datos y preferencias: Al presentar solamente 4 pantallas para que el usuario llene la información importante para una eficiente precisión, el tiempo es rápido a comparación de la carga completa, además podrá seleccionar sus preferencias.
3. Precisión moderada en las recomendaciones iniciales: La información cargada usando procesamiento natural de lenguaje, tiende en un 30% a mostrar errores, por lo que no se visualizará manualmente cada dato al ser subido y el procesamiento usará estos datos para calcular las recomendaciones con base a las publicaciones subidas por lo que la precisión es moderada.

Forma completa:

1. Configuración más exhaustiva y precisa de la información tras el procesamiento sobre su experiencia, habilidades, intereses, etc.a: El usuario podrá ver los datos agrupados y subidos al sistema tras cargar su hoja de vida. Por lo que podrá corregir manualmente los errores si es que se llegara a presentar.

2. Mayor tiempo de registro debido a la cantidad de datos requeridos: Se presentan todas las pantallas de la información que se va a recopilar, por lo que el proceso puede tardar un poco más a comparación de los demás métodos.
3. Mayor nivel de precisión en las recomendaciones posteriores: La información al ser verificada manualmente por un humano tiende a aumentar el nivel de precisión para la comparación de similitud con cada publicación

Forma rápida:

1. Proceso de registro rápido y sencillo: Solamente deberá llenar 3 pantallas que corresponden a los datos personales del usuario y es mucho más rápido a comparación a las demás formas.
2. Información mínima requerida para comenzar a recibir recomendaciones: No se va a recopilar información como habilidades, educación, experiencia laboral, proyectos, etc.
3. No necesita carga automática por hoja de vida: Ninguna pantalla va a requerir que cargue la hoja de vida, pero podrá hacerlo después tras crear su cuenta.
4. Menor nivel de precisión inicial en las recomendaciones, hasta actualizar la información: Tras no recibir muchos datos por parte del usuario, el nivel de precisión va a ser muy bajo hasta que pueda agregar la información adicional recomendada.

2.3.2. RF02 Iniciar Sesión

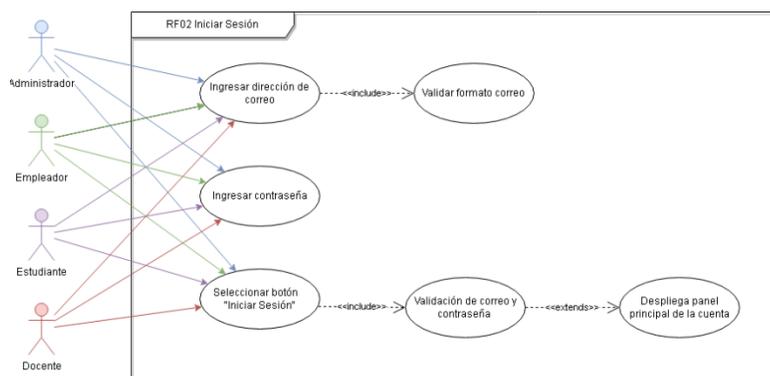


Ilustración 3 Caso de uso de inicio de sesión

El usuario ingresa su correo electrónico y contraseña en los campos correspondientes del formulario de inicio de sesión, tras realizar esto el sistema verifica la validez de los datos ingresados comparándolos con la información almacenada en la base de datos.

Si los datos son correctos y coinciden con los registros de la base de datos, el sistema autentica al usuario y le concede acceso al sistema. Como resultado de la autenticación exitosa, el sistema despliega el panel principal, que es la interfaz desde la cual el usuario puede acceder a las funcionalidades y Características del sistema, dependiendo de su rol de usuario (estudiante, docente, empleador, administrador).

2.3.3. RF03 Actualización de datos



Ilustración 4 Caso de uso de actualización de datos

Para empezar el usuario deberá acceder previamente al sistema e inicia sesión con sus credenciales, una vez autenticado, el usuario ingresa al perfil de su cuenta dentro del sistema. El sistema muestra la información personal asociada a la cuenta del usuario, indicando qué campos se pueden modificar y cuáles no están disponibles para su edición, el usuario selecciona los campos que desea actualizar y proporciona los nuevos valores correspondientes.

Después de ingresar los datos actualizados, el usuario confirma la acción al hacer clic en el botón de "Actualizar datos". El sistema verifica los datos proporcionados por el usuario, realizando validaciones para asegurarse de que cumplan con los requisitos establecidos, como el formato correcto o la longitud adecuada.

Si no hay problemas, el sistema procede a actualizar los datos modificados y guarda los cambios en la base de datos. El sistema confirma la actualización exitosa de los datos al usuario.

2.3.4. RF04 Visualización Publicaciones

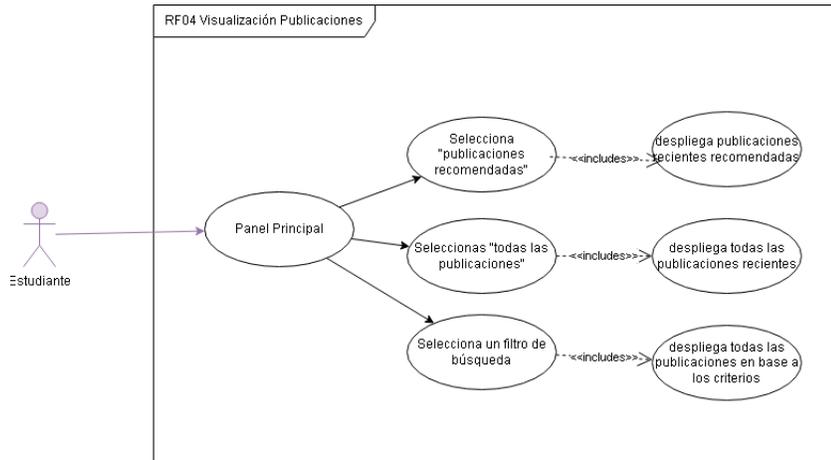


Ilustración 5 Caso de uso de visualización de publicaciones

El usuario autenticado ingresa a su cuenta en el sistema o accede a la opción de "inicio" o selecciona el botón de "recomendado". El sistema muestra las publicaciones más recientes y recomendadas para el perfil del usuario. Si el usuario desea explorar más allá de las publicaciones recomendadas, el sistema proporciona opciones de búsqueda y navegación adicionales. Por ejemplo, el usuario puede buscar publicaciones por ciudad, categorías, horarios, etc.

2.3.5. RF05 Visualización Perfiles

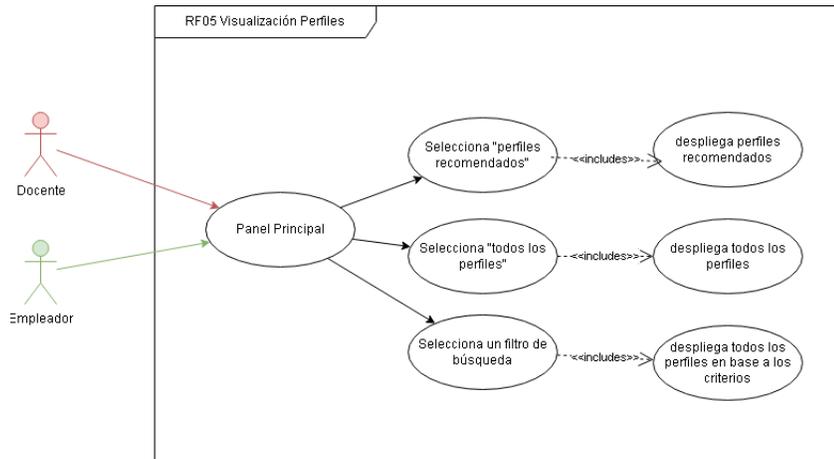


Ilustración 6 Caso de uso de visualización de perfiles

El usuario accede a su cuenta o selecciona la opción de "inicio" o "recomendado". El sistema muestra los perfiles recomendados de estudiantes en base a las publicaciones realizadas y los criterios establecidos. Este caso de uso permite a los empleadores y docentes acceder y explorar perfiles recomendados de estudiantes para satisfacer sus necesidades y requisitos en cuanto a contratación o selección de estudiantes para proyectos académicos.

2.3.6. RF06 Publicaciones

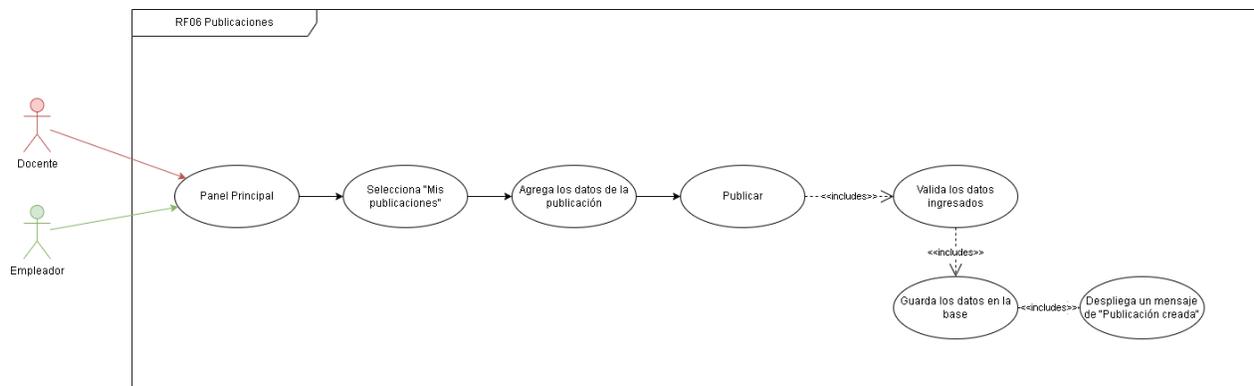


Ilustración 7 Caso de uso de publicaciones

El usuario accede a su cuenta o selecciona el botón de inicio y se dirige a la sección de "Mis publicaciones". Luego, completa los datos de la publicación en los campos correspondientes y selecciona la opción de publicar. El sistema valida los datos ingresados, los guarda en la base de datos y muestra un mensaje de confirmación de la publicación exitosa.

Este caso de uso proporciona a los empleadores y docentes la capacidad de crear y publicar ofertas laborales o publicaciones relevantes en la plataforma teniendo la posibilidad de estilizar el texto y agregar imágenes, facilitando la comunicación con posibles estudiantes interesados.

2.3.7. RF07 Reacciones

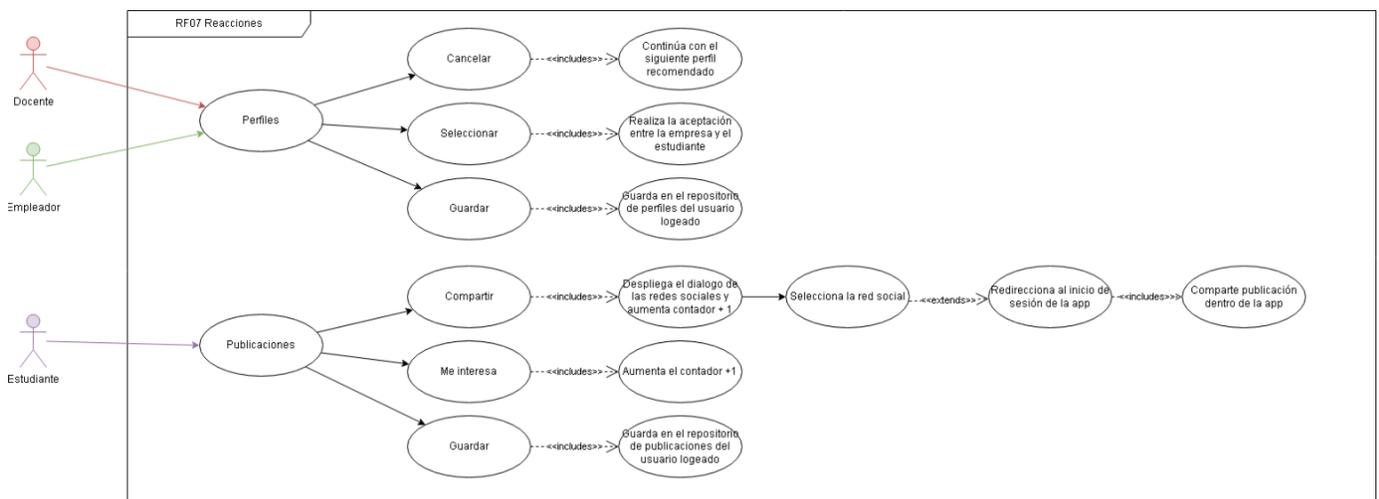


Ilustración 8 Caso de uso reacciones

El usuario accede a su cuenta o selecciona el botón de inicio y visualiza las publicaciones o perfiles recomendados según su rol. Luego, el usuario decide compartir, guardar o mostrar interés en el contenido de la publicación. Para ello, selecciona las opciones de "Me interesa", "Compartir" o "Guardar" para las publicaciones, o "Cancelar", "Seleccionar" o "Guardar" para los perfiles. El sistema realiza la acción seleccionada y actualiza el contador correspondiente.

Este caso de uso permite a los usuarios expresar su interés, compartir y guardar contenido relevante, proporcionando una interacción más enriquecedora en el sistema y mejorando la experiencia del usuario.

2.3.8. RF08 Hoja de vida

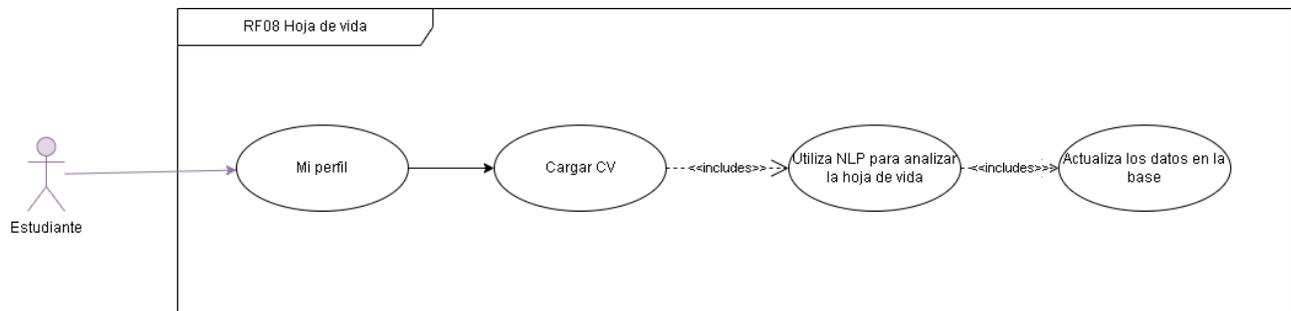


Ilustración 9 Caso de uso hoja de vida

El usuario accede a la opción "Mi perfil" y selecciona la opción de cargar la hoja de vida. El sistema realiza un análisis del PDF y genera un vector con las palabras relevantes extraídas de la hoja de vida, permite a los estudiantes agilizar el proceso de carga de su hoja de vida, utilizando la tecnología para extraer automáticamente información relevante y completar su perfil de manera más eficiente y por último actualiza la información en la base de datos.

2.4. Diagramas de secuencia

2.4.1. RF01 Registro de usuarios

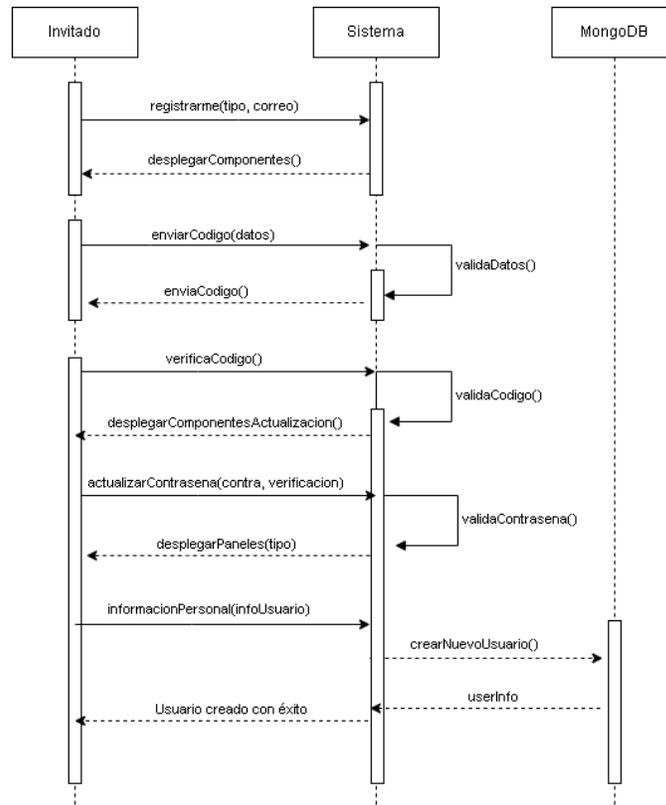


Ilustración 10 Diagrama secuencia de registro de usuarios

El usuario invitado ingresa el tipo de usuario al que corresponde, en este caso puede ser estudiante, docente o empleador. A partir de esto, ingresa el correo y el sistema despliega los componentes necesarios para el registro. El usuario deberá dar clic en enviar código, enviando los datos como argumento, el sistema validará todo acerca de la información de los datos, cómo que los campos se encuentren completados, de que el usuario no exista en la base de datos, que el correo se encuentre en un formato adecuado, etc. Si todos los datos se encuentran correctamente el sistema enviará el

código de verificación al correo ingresado y el usuario deberá ingresar y llamar al método `verificaCodigo()` donde se validará que el código ingresado sea igual al de confirmación.

Si el código coincide se desplegarán los componentes para la actualización de la contraseña, en donde el usuario deberá llenar la contraseña nueva y la verificación de esta. El sistema validará la contraseña ingresada y por último desplegará los paneles por el tipo. El usuario deberá llenar todos los campos de información presentes y al finalizar se guardará toda la información personal ingresada por parte del usuario en `mongodb` y por último se desplegará un mensaje de creación exitosa de la cuenta, caso contrario un mensaje de error.

2.4.2. RF02 Iniciar Sesión

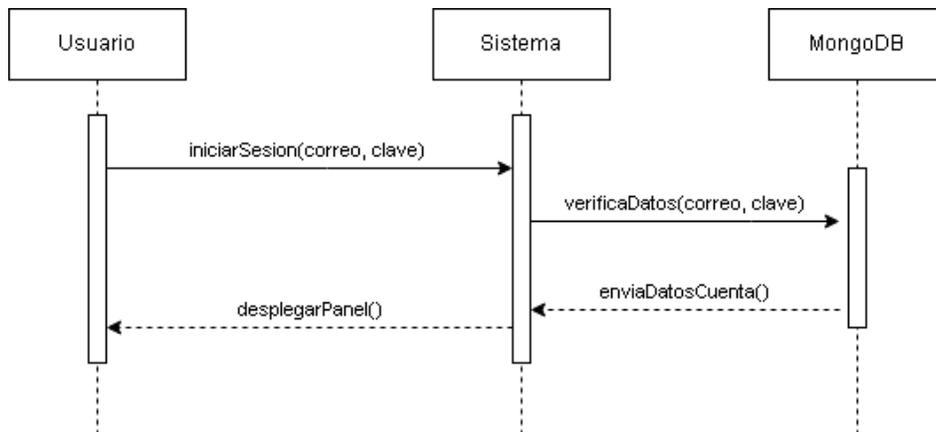


Ilustración 11 Diagrama de secuencia iniciar sesión

El usuario inicia sesión agregando su correo y contraseña, el sistema verifica sus datos y compara con los datos guardados en `mongodb`, el sistema valida que los datos sean correctos y despliega el panel de acuerdo con el rol del usuario.

2.4.3. RF03 Actualización de datos

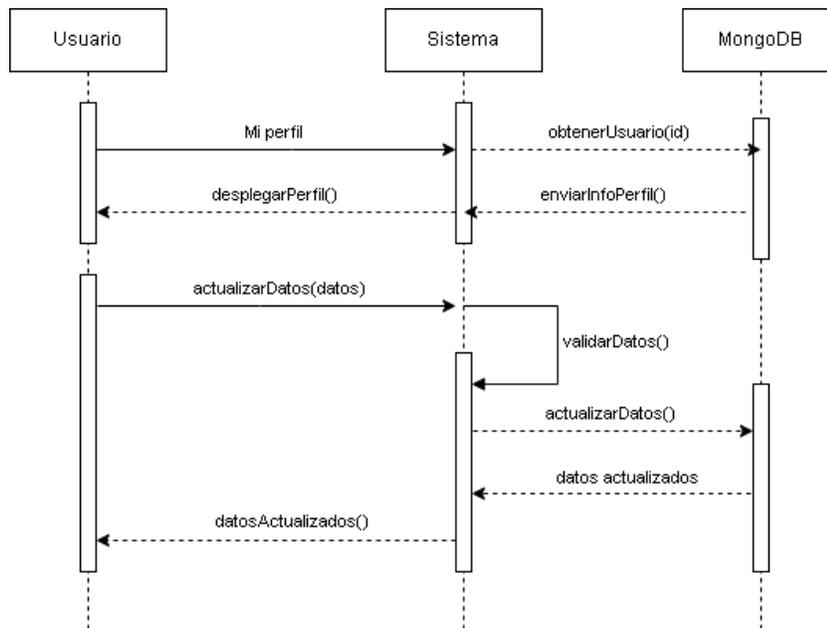


Ilustración 12 Diagrama de secuencia de la actualización de datos

El usuario ingresa a su perfil, el sistema enviará una petición para obtener los datos a mongodb de acuerdo con el identificador, si encuentra los datos de acuerdo con el identificador, envía la información personal del usuario hacia el sistema y el sistema envía al cliente.

Una vez llenado el dato a actualizar el sistema validará el mismo y procede a modificar los datos en mongodb , luego envía los datos actualizadas y el sistema manda la información nueva al cliente.

2.4.4. RF04 Visualización Publicaciones

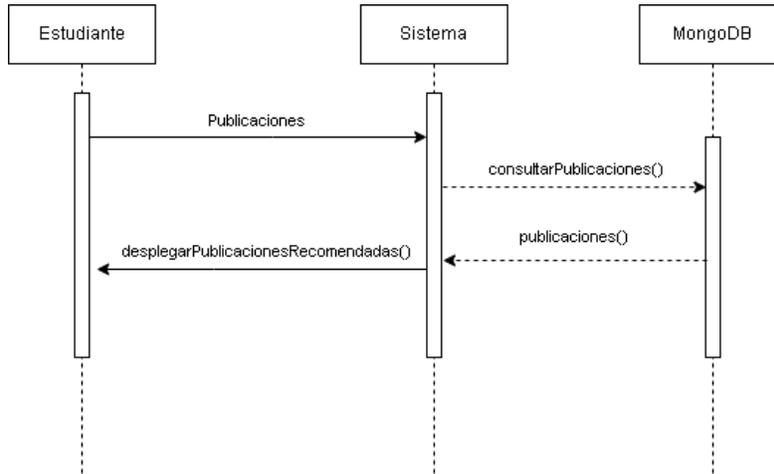


Ilustración 13 Diagrama de secuencia de publicaciones recomendadas

El usuario con el rol de estudiante entra al panel principal donde solicitará al sistema y a su vez a la base de datos todas las publicaciones que se encuentran almacenadas que han sido realizadas por los docentes y empleadores, el sistema obtendrá las palabras relevantes de cada publicación y las analizará con base a la hoja de vida del estudiante, por último, se obtendrá un resultado cualitativo de la similitud que se genera por el método desplegarPublicacionesRecomendadas.

2.4.5. RF05 Visualización Perfiles

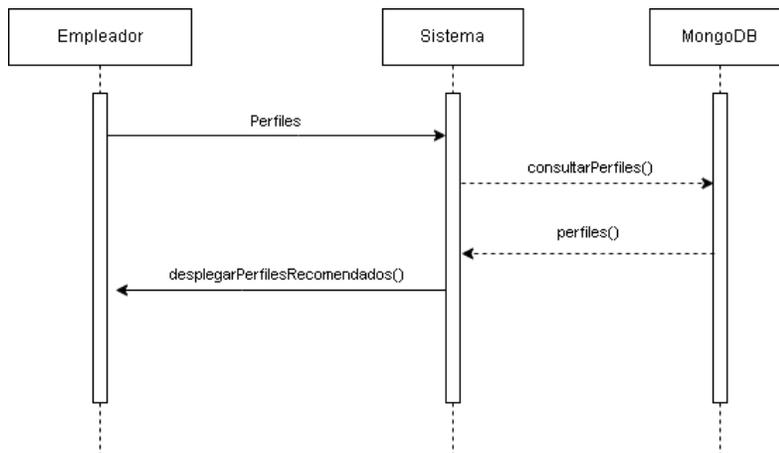


Ilustración 14 Diagrama de secuencia de visualización de perfiles

El usuario con el rol de empleador entra al panel principal donde solicitará al sistema y a su vez a la base de datos todas las hojas de vida de los estudiantes que han postulado para la publicación realizada por el empleador, el sistema obtendrá las palabras relevantes de cada publicación y las analizará con base a la hoja de vida del estudiante, por último, se obtendrá un resultado cualitativo de la similitud que se genera por el método desplegarPerfilesRecomendados.

2.4.6. RF06 Publicaciones

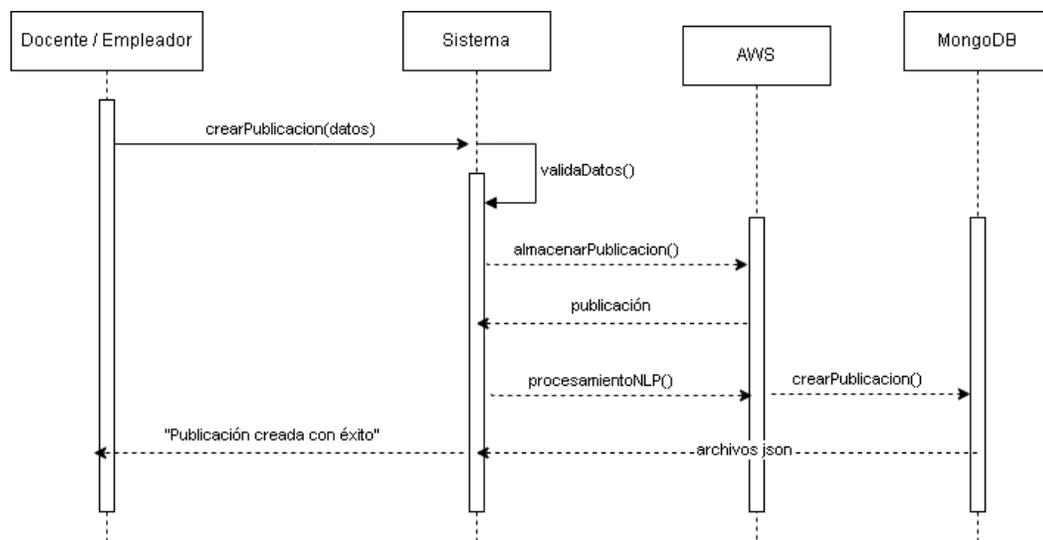


Ilustración 15 Diagrama de secuencia de publicaciones

Los usuarios con el rol de docente y empleador podrán realizar publicaciones hacia el sistema, llenando previamente la información como el título, descripción e imagen si es el caso. El sistema validará que los datos sean los correctos y guardará la publicación en AWS para luego utilizar en el sistema para procesar el texto incluido en la imagen y posteriormente guardar en la base de datos mongodb. Si todos los procesos fueron correctos el sistema enviará el mensaje de confirmación hacia el cliente.

2.4.7. RF07 Reacciones

Empresa

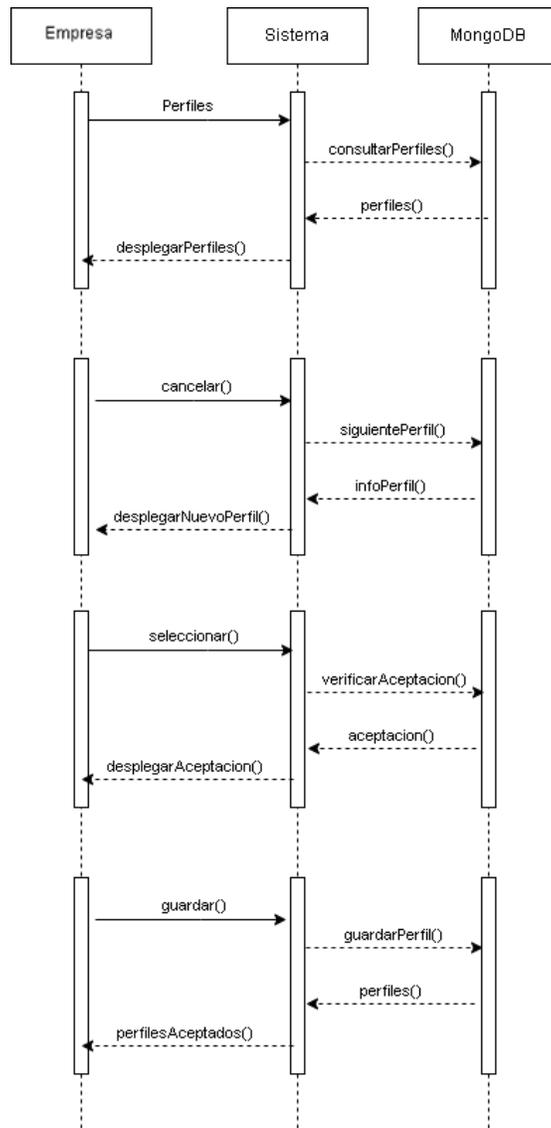


Ilustración 16 Diagrama de secuencia reacciones empresa / docente

Los usuarios con rol de empresa consultan los perfiles al sistema de los estudiantes que han decidido postular para sus ofertas laborales **RF05**, a partir de esto pueden reaccionar a cada perfil, ya sea seleccionándolo, guardando o cancelando hasta poder visualizar otro candidato.

Estudiante

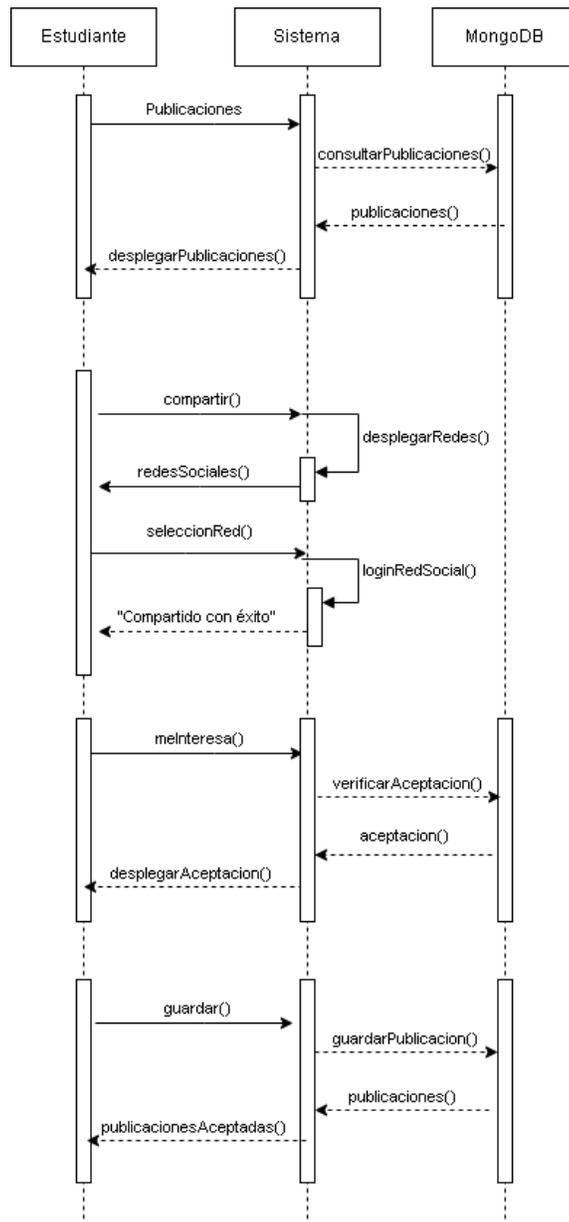


Ilustración 17 Diagrama de secuencia de reacciones de estudiantes

Los usuarios con rol de estudiante consultan las publicaciones disponibles en el sistema, el sistema solicita a mongodb y los devuelve para la visualización al usuario **RF06**. A partir de esto, puede reaccionar a cada publicación, ya sea “Me interesa”, “Guardar”, “Compartir”, etc.

2.4.8. RF08 Hoja de vida

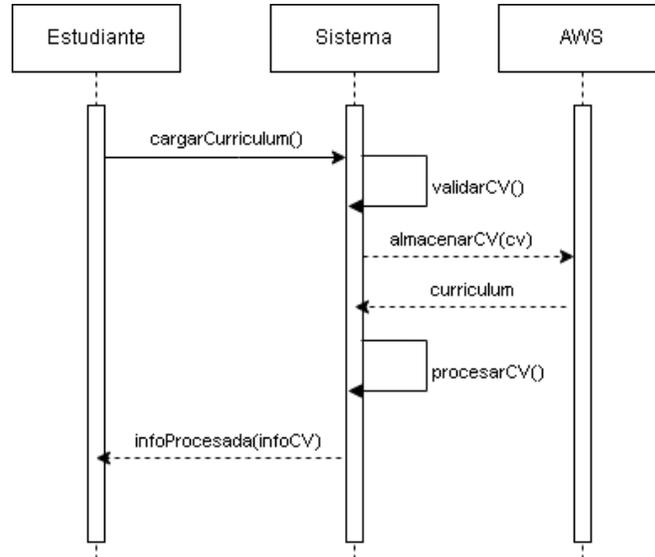


Ilustración 18 Diagrama de secuencia de hoja de vida

El estudiante carga su hoja de vida y el sistema validará que cumpla con los parámetros requeridos, a partir de ello almacenará el archivo en AWS y retornará el archivo para que el sistema lo procese y envíe al cliente la información acerca de sus habilidades, historial académico, laboral, etc.

2.5. Diccionario de datos

Tabla 12. Tabla diccionario modelo Usuario

User			
Campo	Tamaño	Tipo de dato	Descripción
email	100	Cadena	Almacena la dirección de correo electrónico del usuario.
type	50	Cadena	Clasifica los usuarios por su respectivo rol “Administrador”, “Estudiante”, “Docente”, “Empleador”.
password	32	Cadena	Almacena la contraseña del usuario de forma cifrada y segura.

id_profile	50	Cadena	Identificador único para cada perfil de usuario en la base de datos
names	150	Cadena	Nombres del usuario.
lastnames	150	Cadena	Apellidos del usuario.

Tabla 13. Tabla diccionario modelo Sesión.

Session			
Campo	Tamaño	Tipo de dato	Descripción
token	50	Cadena	Llave secreta para identificar al usuario durante su sesión, y para asegurar que sólo él pueda acceder a su información
creation	30	Fecha	Almacena la fecha y hora en que se generó un token o cuando se creó una nueva sesión de usuario.
id_user	40	Cadena	Identificador único para cada usuario.

Tabla 14. Tabla diccionario modelo Perfil Empresa.

InstitutionProfile			
Campo	Tamaño	Tipo de dato	Descripción
badge	-	Objeto	Almacena la imagen o avatar del logo de la empresa o marca.
type	30	Cadena	Clasifica el tipo de empresa ya sea pública, privada, etc.
ruc	13	Cadena	Almacena el Registro Único de Contribuyentes de la empresa
contacts	-	Objeto	Almacena las direcciones de correo electrónico y celulares a los cuales se le puede contactar.
business_name	250	Cadena	Almacena el nombre o razón social de la empresa.
address	150	Cadena	Almacena la dirección física de la empresa
city	100	Cadena	Almacena la ciudad en la que se encuentra la empresa

id_save_users	50	Vector	Identificador único para visualizar la cantidad de estudiantes guardados por el usuario
id_user	50	Cadena	Identificador del usuario.
position	150	Cadena	Cargo que ocupa el usuario dentro de la empresa.

Tabla 15. Tabla diccionario modelo Perfil Docente.

TeacherProfile			
Campo	Tamaño	Tipo de dato	Descripción
badge	-	Objeto	Almacena la foto de perfil o avatar del usuario.
identification	10	Cadena	Identificación única del usuario.
contacts	150	Objeto	Almacena las direcciones de correo electrónico y celulares a los cuales se le puede contactar.
Contributions	10	Numérico	Contribuciones o cantidad de publicaciones que ha realizado el usuario.
id_user	50	Cadena	Identificador del usuario.
grade	150	Cadena	Cargo que ocupa el docente dentro de la institución.
gender	50	Cadena	Género con el cual se identifica el usuario.

Tabla 16. Tabla diccionario modelo Perfil Estudiante.

StudentProfile			
Campo	Tamaño	Tipo de dato	Descripción
badge	-	Objeto	Almacena la foto o avatar del usuario.
identification	10	Cadena	Identificación única del usuario.
career	75	Cadena	Almacenar información sobre la carrera o profesión del usuario.
cities	-	Vector de cadenas	Almacena las ciudades en las que se encuentra interesado para recibir las ofertas.
contacts	-	Objeto	Almacena las direcciones de correo electrónico y celulares a los cuales se le puede contactar.
gender	150	Cadena	Almacena la información sobre el género del usuario.
civil_status	150	Cadena	Almacena la información sobre el estado civil del usuario.

birthdate	15	Fecha	Almacena la información sobre la fecha de nacimiento del usuario.
description	-	Objeto	Almacena una breve descripción del perfil del usuario y la visión dentro del empleo.
years	-	Numérico	Almacena la edad del usuario.
academic	-	Objeto	Almacena los campos de primaria, secundaria, universidad y otros.
projects	-	Vector de Objetos	Almacena los proyectos realizados mediante el título.
skills	-	Vector de Objetos	Almacena las habilidades que posee el usuario dentro del campo laboral mediante el título.
work	-	Vector de Objetos	Almacena los lugares en donde ha trabajado mediante el título.
courses	-	Vector de Objetos	Almacena los cursos realizados mediante el título.
cv	-	Vector de cadenas	Guarda las palabras relevantes encontradas después del procesamiento de la hoja de vida del estudiante
types	-	Vector de cadenas	Almacena el interés por tipo de publicaciones requeridas "Pasantías, bolsa de trabajo, vinculación".
benefits	-	Vector de cadenas	Almacena el salario "Remunerado, no remunerado".
assist	-	Vector de cadenas	Almacena el tipo de asistencia requerida: "Presencial, remoto".
time	-	Vector de cadenas	Almacena el tipo de tiempo requerido: "Completo, medio tiempo, parcial, etc."
urlcv	-	Cadena	Almacena el enlace del currículum en AWS.
deserted	-	Vector de Objetos	Almacena la información de las publicaciones que ha rechazado el usuario.
Languages	-	Vector de Objetos	Almacena la información de los lenguajes o idiomas que dispone el usuario "Español, inglés, etc."

Tabla 17. Tabla diccionario modelo Publicación.

Post			
Campo	Tamaño	Tipo de dato	Descripción
title	100	Cadena	Título de la oferta de trabajo.
img	-	Cadena	Enlace de la imagen relacionada con la oferta de trabajo.
date_publication	10	Fecha	Almacena la fecha en que se publicó la oferta de trabajo.
type	50	Cadena	Almacena el tipo de oferta de trabajo, como "tiempo completo", "medio tiempo" o "contrato por proyecto"

time_start	10	Fecha	Fecha que comienza la postulación
time_end	10	Fecha	Fecha que termina la postulación
salary	10	Numérico	Salario ofrecido en la oferta de trabajo
description	250	Cadena	Descripción detallada de la oferta de trabajo
address	150	Cadena	Lugar en la que se va a ofrecer la oferta de trabajo.
like_count	-	Numérico	Número de veces que la oferta de trabajo ha sido "interesada"
save_count	-	Numérico	Número de veces que la oferta de trabajo ha sido "guardada"
exit_count	-	Numérico	Número de veces que la oferta de trabajo ha sido "rechazada"
transcription	-	Cadena	Resultado de las palabras más relevantes en la publicación tras el procesamiento de texto en la imagen.
time	50	Cadena	Tiempo en el que se va a ofrecer la oferta "Completo, medio tiempo, parcial" etc.
mode	100	Cadena	Asistencia en el que se va a ofrecer la oferta "Presencial, remoto" etc.
experience	-	Numérico	Experiencia requerida para la oferta.
id_user	-	Cadena	Identificador único del usuario.

Tabla 18. Tabla diccionario modelo Errores.

Errors			
Campo	Tamaño	Tipo de dato	Descripción
code	15	Cadena	Código de identificación del error.
severity	15	Cadena	Nivel de gravedad del error al sistema o al usuario en una escala del 1-10.
description	100	Cadena	Breve descripción del error.
module	50	Cadena	Módulo en el que se produjo el error.
id_user	50	Cadena	Identificador único del usuario.
date	10	Fecha	Fecha en la que se produjo el error.

Tabla 19. Tabla diccionario modelo Académico.

Academic			
Campo	Tamaño	Tipo de dato	Descripción
primary	250	Cadena	Estudios de la primaria del estudiante.

secondary	250	Cadena	Estudios de la secundaria del estudiante.
university	250	Cadena	Estudios de la universidad del estudiante.
additional	500	Cadena	Almacena cualquier información adicional relevante sobre la educación y formación, como honores o premios recibidos.

CAPITULO III: CONSTRUCCION Y PRUEBAS

2.4. Diagrama de despliegue de la aplicación

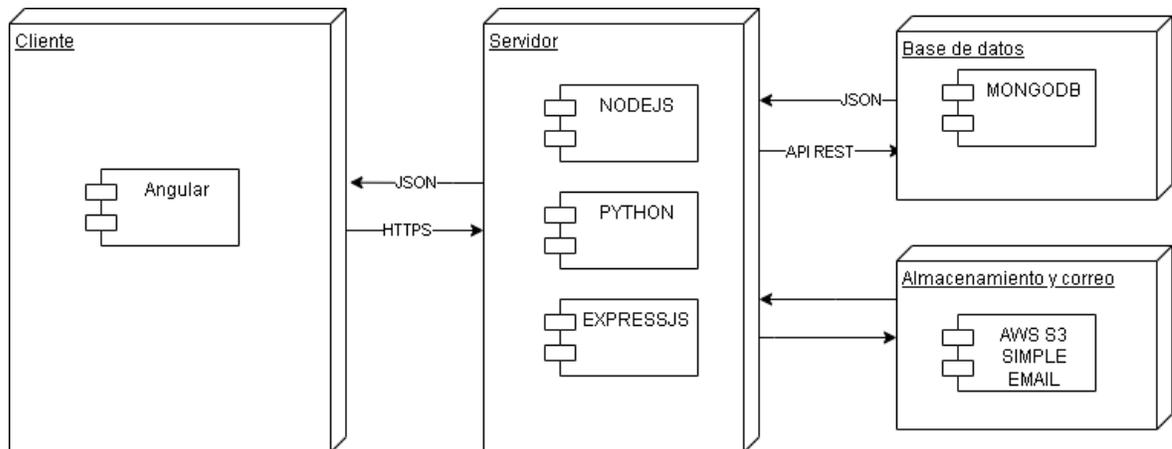


Ilustración 19 Diagrama de despliegue de la aplicación

1. **Cliente (Angular):** En este caso representa la interfaz de usuario con la que los usuarios interactúan con la aplicación ya sea los componentes para mostrar las publicaciones de las ofertas laborales, el perfil del estudiante, el panel de autenticación o de registro, etc. Este a su vez puede accederse a través de un navegador web o móvil que ejecuta la aplicación en Angular y realiza las respectivas comunicaciones con la API del backend. La estructura del proyecto está constituida por componentes en los cuales contiene cada uno hojas de estilo en cascada

(CSS), lenguaje de marcado (HTML) y el lenguaje de programación de JavaScript con sintaxis para tipos (Typescript).

2. Servidor de aplicaciones (NodeJS): Ejecuta la lógica de negocio de la aplicación y proporcionar una API para que el cliente interactúe con el servidor, pudiendo ejecutar funciones mediante las peticiones GET, POST, PUT, DELETE. Este servidor de aplicaciones se comunica con la base de datos mongodb y el servidor web.
3. Servidor web (ExpressJS): Se utiliza como framework de aplicaciones web para Node.js que actúa como un servidor web que maneja las solicitudes HTTP del cliente y las enruta al servidor, manejando así el enrutamiento de archivos estáticos y otros recursos.
4. Base de datos (MongoDB): Base de datos NoSQL orientada a documentos, se realizó el API REST mediante la librería de mongoose la cual soporta Node.JS, para interactuar entre el servidor de aplicaciones para recopilar los datos solicitados por el cliente.
5. API REST: Es usada a través del servidor de aplicaciones y permite que el cliente realice operaciones CRUD (crear, leer, actualizar, eliminar) en la base de datos. Utiliza el protocolo HTTP para la comunicación y se basa en estructuras de datos JSON.
6. AWS: Se utiliza como repositorio de almacenamiento y cómo gestor de correos electrónicos, en donde se conecta con el servidor para recibir los archivos en este caso las hojas de vida de los estudiantes, la foto de perfil y las imágenes de las publicaciones.

2.5. Algoritmos Importantes

2.5.1. Algoritmos Importantes de Python

```
57 def preprocess_text(text):
58     ... # Tokenización y cambio todo a minúscula
59     ... tokens = word_tokenize(text.lower())
60     ...
61     ... # Eliminar palabras vacías (stop words)
62     ... stop_words = set(stopwords.words('spanish'))
63     ... filtered_tokens = [token for token in tokens if token not in stop_words]
64     ...
65     ... # Lematización
66     ... lemmatizer = WordNetLemmatizer()
67     ... lemmas = [lemmatizer.lemmatize(token) for token in filtered_tokens]
68     ...
69     ... return lemmas
```

Ilustración 20 Preprocesamiento de texto

En la ilustración 20, se realiza el procesamiento de texto para preparar el texto de entrada para su posterior análisis o procesamiento. En la **línea 59** se realiza el proceso de tokenización que consiste separar una cadena de caracteres en tokens o palabras individuales dentro de un arreglo, separando en este caso mediante espacios en blanco y convirtiendo estas palabras en minúscula para normalizar el texto o ignorar si al comparar las palabras estas se encuentran con mayúsculas o minúsculas.

Luego, en la **línea 62**, elimina las palabras vacías, que son palabras comunes y no informativas, y finalmente en la **línea 66** aplica la lematización para reducir las palabras a su forma base. Al combinar estos pasos, la función devuelve una lista de lemas, que son las formas simplificadas y normalizadas de las palabras originales en el texto. Este preprocesamiento es esencial para mejorar la calidad y la eficacia de las tareas de análisis o procesamiento del lenguaje natural para las publicaciones y para las hojas de vida de los estudiantes.

```

96 def eliminar_informacion_personal(texto):
97     ... # Eliminar correos electrónicos
98     ... patron_correo = re.compile(r'\S+@\S+')
99     ... texto_sin_correos = [re.sub(patron_correo, '', t) for t in texto]
100
101     ... # Eliminar nombres propios
102     ... texto_sin_nombres = []
103     ... for t in texto_sin_correos:
104         ... tokens = word_tokenize(t)
105         ... etiquetas_pos = pos_tag(tokens)
106         ... texto_sin_nombres.append(' '.join([token for token, pos in etiquetas_pos if pos != 'NNP']))
107
108     ... # Eliminar símbolos
109     ... texto_sin_simbolos = [re.sub(r'^\W\s', '', t) for t in texto_sin_nombres]
110
111     ... # Eliminar números
112     ... texto_sin_numeros = [re.sub(r'\d+', '', t) for t in texto_sin_simbolos]
113
114     ... return texto_sin_numeros

```

Ilustración 21 Algoritmo para eliminación de información personal

En la ilustración 21, la función se encarga de eliminar información personal sensible del texto de entrada. En la **línea 99**, se busca y elimina cualquier ocurrencia de direcciones de correo electrónico en el texto utilizando una expresión regular. Cada correo electrónico encontrado se reemplaza por una cadena vacía. En la **línea 103**, se procede a eliminar nombres propios del texto. Para ello, se tokeniza cada texto y en la **línea 105** se etiquetan las partes de los tokens utilizando pos_tag de NLTK. En la **línea 106**, se filtran los tokens que sean nombres propios (NNP) y se unen nuevamente en un texto sin nombres propios.

En la **línea 109**, se eliminan los símbolos del texto utilizando una expresión regular que conserva solo caracteres alfanuméricos y espacios en blanco.

En la **línea 112**, se termina eliminando todos los números presentes en el texto utilizando otra expresión regular, ya que no aportan valor en la hoja de vida ni en las publicaciones para el momento de hacer la similitud.

La función devuelve el texto resultante después de pasar por todos estos pasos de eliminación de información personal, con el objetivo de proteger la privacidad y confidencialidad de los datos sensibles en el texto original.

```
76 def palabras_relevantes(texto, palabras_clave):
77     # Unir los tokens en un solo texto separados por espacios
78     texto = ' '.join(texto)
79     # Crear una instancia del vectorizador TF-IDF (Palabras frecuentes)
80     # Frecuencia inversa de documento (Calcula la relevancia de la palabra)
81     vectorizador = TfidfVectorizer(vocabulary=palabras_clave)
82     # Ajustar el vectorizador al texto
83     matriz_tfidf = vectorizador.fit_transform([texto])
84     # Obtener las características y sus pesos TF-IDF
85     características = vectorizador.get_feature_names_out()
86
87     pesos_tfidf = matriz_tfidf.toarray()[0]
88
89     # Combinar las características y sus pesos en una lista de tuplas
90     características_con_pesos = list(zip(características, pesos_tfidf))
91
92     # Ordenar las características por sus pesos TF-IDF en orden descendente
93     características_ordenadas = sorted(características_con_pesos, key=lambda x: x[1], reverse=True)
94     características_mayores_que_cero = [característica for característica, peso in características_con_pesos if peso > 0]
95     return características_mayores_que_cero
```

Ilustración 22 Algoritmo para obtener palabras relevantes

En la ilustración 22, se realiza un proceso para identificar las palabras relevantes en un texto utilizando el enfoque de vectorización TF-IDF (Frecuencia de Término-Inversa de Frecuencia de Documento). En la **línea 78**, primero se unen los tokens del texto en una única cadena de texto. Desde la **línea 81**, se crea una instancia del vectorizador TF-IDF y se ajusta al texto para obtener la representación vectorial TF-IDF. A partir de esto, se obtienen las características (palabras) y sus pesos TF-IDF correspondientes. Estas características se ordenan en función de sus pesos de forma descendente. Finalmente, se filtran y devuelven solo las características con pesos TF-IDF mayores que cero, lo que indica su importancia en el texto original. Esta función es útil para identificar las

palabras más relevantes de las hojas de vida y de las publicaciones a través de una bolsa de palabras clave para el análisis de texto.

```
71 # Función para eliminar palabras repetidas
72 def eliminar_repetidas(tokens):
73     tokens_sin_repetidas = list(set(tokens))
74     return tokens_sin_repetidas
```

Ilustración 23 Algoritmo para eliminar palabras repetidas

En la ilustración 23, se realiza un proceso para eliminar las palabras duplicadas dentro de un conjunto de tokens o una lista de palabras.

```
40 def get_file_aws(file_name):
41     # Nombre del bucket y del archivo
42     bucket_name = 'feupsimagefiles'
43     # Obtener el objeto del bucket
44     response = s3.get_object(Bucket=bucket_name, Key=file_name)
45     # Leer el contenido del archivo PDF en un objeto de tipo BytesIO
46     file_content = response['Body'].read()
47     file = io.BytesIO(file_content)
48     return file
```

Ilustración 24 Algoritmo para obtener un archivo de AWS

En la ilustración 24, la función se encarga de obtener un archivo almacenado en un bucket de Amazon S3. En la **línea 42**, se especifica el nombre del bucket donde se encuentra almacenado el archivo en la variable `bucket_name`. En la **línea 44**, se utiliza la función `get_object` de la biblioteca `s3` para obtener el objeto correspondiente al archivo especificado por su nombre. Esto implica realizar una solicitud al servicio de Amazon S3 para recuperar el archivo del bucket.

En la **línea 46**, se obtiene la respuesta del servicio de S3, se lee el contenido del archivo PDF en un objeto de tipo `BytesIO` utilizando el atributo `Body` de la respuesta.

Finalmente, en la **línea 47**, se crea un objeto BytesIO a partir del contenido del archivo leído, que se asigna a la variable file, y se devuelve dicho objeto. El objeto BytesIO nos permite manipular y trabajar con el contenido del archivo en memoria. Esto con el objetivo de obtener archivos almacenados en Amazon S3 y leer su contenido para su posterior procesamiento o análisis.

```
def get_cv(email):  
    ... # Crear un objeto de tipo PdfFileReader para extraer el texto  
    ... file_name = 'curriculums/' + str(email) + '/Curriculum.pdf'  
    ... file = get_file_aws(file_name)  
    ... pdf_reader = PyPDF2.PdfReader(file)  
    ... num_pages = len(pdf_reader.pages)  
    ... # Iterar sobre las páginas y mostrar el texto  
    ... text = ''  
    ... for page in range(num_pages):  
    ...     pdf_page = pdf_reader.pages[page]  
    ...     text = text + pdf_page.extract_text()  
    ... return text
```

Ilustración 25 Algoritmo para extraer el texto de un currículum

En la ilustración 25, la función tiene como objetivo obtener el contenido de un currículum vitae (CV) en formato PDF asociado a una dirección de correo electrónico específica que es la que se genera automáticamente al registrarse y para clasificar la información en AWS S3.

En la **línea 29**, se comienza construyendo el nombre del archivo PDF del currículum utilizando la dirección de correo electrónico proporcionada por el usuario. El nombre del archivo se forma concatenando la carpeta "curriculums/" con la dirección de correo electrónico y la extensión ".pdf".

En la **línea 30**, se utiliza la función "get_file_aws" para obtener el archivo PDF del bucket de Amazon S3. Se crea un objeto de tipo "PdfFileReader" a partir del archivo obtenido para poder extraer el texto.

En la **línea 32**, se obtiene el número total de páginas del currículum PDF utilizando la propiedad “pages” del objeto “PdfFileReader”.

En la **línea 35**, se itera sobre cada página del currículum y se extrae el texto de cada una de ellas. El texto extraído se concatena en la variable “text”.

Finalmente, se devuelve el texto completo del currículum vitae, que representa la concatenación de todo el texto extraído de las páginas del PDF. Esto será usado para realizar tareas de análisis de texto, extracción de información o búsqueda de términos clave en el currículum.

```
50 def get_ocr_from_image(file_name):
51     ... file = get_file_aws(file_name)
52     ... image = Image.open(file)
53     ... text = pytesseract.image_to_string(image, lang='spa')
54     ... return text
```

Ilustración 26 Algoritmo para extraer el texto de una imagen

En la ilustración 26, la función tiene como objetivo realizar reconocimiento óptico de caracteres (OCR, por sus siglas en inglés) en una imagen específica.

En la **línea 51**, llamamos a la función mencionada en la **ilustración 24**, que recupera el archivo desde un bucket de Amazon S3. El nombre del archivo se pasa como parámetro a la función.

Y en la **línea 52**, se utiliza la biblioteca Pillow para abrir la imagen utilizando “Image.open(file)”, donde “file” es el objeto de archivo obtenido previamente.

En la **línea 53**, se utiliza la biblioteca pytesseract para realizar el proceso de OCR en la imagen. La función “image_to_string” de pytesseract se utiliza para extraer el texto de la imagen. Se especifica el parámetro “lang='spa'” para indicar que se espera que el texto esté en español.

Finalmente, se devuelve el texto extraído de la imagen. Esto nos permite extraer el texto de las imágenes de las publicaciones realizadas.

```
41 | def extract_images_from_pdf(pdf_url, bucket_name, email):
42 |     s3 = boto3.client('s3', aws_access_key_id=aws_access_key_id,
43 |                       aws_secret_access_key=aws_secret_access_key)
44 |     response = requests.get(pdf_url)
45 |     pdf_data = io.BytesIO(response.content)
46 |     reader = PyPDF2.PdfReader(pdf_data)
47 |     full_pages = len(reader.pages)
48 |     url = None
49 |     for no_page in range(full_pages):
50 |         page = reader.pages[no_page]
51 |         for image_file_object in page.images:
52 |             # Convertir la imagen a un formato que pueda ser leído por OpenCV
53 |             image_data = np.frombuffer(image_file_object.data, np.uint8)
54 |             image = cv2.imdecode(image_data, cv2.IMREAD_UNCHANGED)
55 |
56 |             # Analizar la imagen para determinar si es una imagen tipo carnet
57 |             if is_id_card(image):
58 |                 key = f'info/photos/{email}/{str(no_page)}_{image_file_object.name}'
59 |                 s3.upload_fileobj(io.BytesIO(image_file_object.data), bucket_name, key)
60 |                 url = f'https://{bucket_name}.s3.amazonaws.com/{key}'
61 |                 break
62 |             if url is not None:
63 |                 break
64 |     return url
```

Ilustración 27 Algoritmo para extraer imágenes de un pdf

En la ilustración 27, la función se encarga de extraer imágenes de un archivo PDF ubicado en una URL específica que luego toma los parámetros que el usuario envía desde nodejs para luego subir las imágenes a un bucket de Amazon S3.

En la **línea 42**, la función comienza configurando el cliente de Amazon S3 mediante la biblioteca boto3. Se utilizan las claves de acceso proporcionadas (aws_access_key_id y aws_secret_access_key) para autenticar la conexión con el servicio de S3.

En la **línea 44**, se utiliza la biblioteca requests para realizar una solicitud a la URL del archivo PDF y obtener los datos del PDF como un objeto de tipo BytesIO (pdf_data).

En la **línea 46**, Se crea un objeto PdfReader de la biblioteca PyPDF2 utilizando los datos del PDF. Se obtiene el número total de páginas del PDF mediante la propiedad pages del objeto reader.

En la **línea 49**, se itera sobre cada página del PDF y se examinan las imágenes presentes en cada una. Para cada imagen, se realiza la conversión a un formato legible por OpenCV y se almacena en un objeto image.

En la **línea 56**, se analiza cada imagen para determinar si es una imagen de tipo carné utilizando la función is_id_card. Si se detecta que es una imagen de carné, se procede a subir la imagen al bucket de Amazon S3 utilizando el cliente s3.upload_fileobj. En la **línea 58**, se genera una clave única para el objeto de imagen en el bucket y se construye una URL de acceso al objeto en S3.

En la **línea 59**, una vez que se ha encontrado y subido una imagen de carné, se almacena la URL de acceso al objeto en la variable url y se rompen los ciclos de iteración para finalizar el proceso.

Finalmente, se devuelve la URL de la imagen de carné en caso de haber encontrado una, o None en caso contrario.

Esto ayudará a extraer y almacenar imágenes específicas de un archivo PDF en un bucket de Amazon S3, lo que permite extraer las imágenes de la hoja de vida del estudiante y posteriormente usarla como perfil de su cuenta.

```

66 def is_id_card(image):
67     # Verificando el tamaño y la relación de aspecto de la imagen
68     height, width, _ = image.shape
69     aspect_ratio = width / height
70     if 0.6 < aspect_ratio < 0.7 and width > 300:
71         return True
72     else:
73         gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
74         # Detectar bordes en la imagen
75         edges = cv2.Canny(gray, 50, 150)
76         # Buscar contornos en la imagen
77         contours, _ = cv2.findContours(edges, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
78         # Verificar si hay un contorno que se parezca a un rectángulo con bordes redondeados
79         for contour in contours:
80             approx = cv2.approxPolyDP(contour, 0.01*cv2.arcLength(contour, True), True)
81             if len(approx) == 4:
82                 x, y, w, h = cv2.boundingRect(approx)
83                 aspect_ratio = w / h
84                 if 0.6 < aspect_ratio < 0.7:
85                     return True
86     return False

```

Ilustración 28 Algoritmo para identificar una imagen de carné en la hoja de vida

En la ilustración 28, la función se encarga de determinar si una imagen dada se asemeja a una imagen de un carné de identidad (ID).

En la **línea 68**, se verifica el tamaño y la relación de aspecto de la imagen. Se obtienen las dimensiones de la imagen (height y width) y se calcula la relación de aspecto dividiendo el ancho entre la altura. Si la relación de aspecto está entre 0.6 y 0.7, y el ancho de la imagen es mayor a 300 píxeles, se considera que la imagen es un carné de identidad.

En la **línea 73**, caso contrario se realiza un análisis más detallado de la imagen. Primero, se convierte la imagen a escala de grises utilizando `cv2.cvtColor`. Luego, se detectan los bordes en la imagen utilizando el algoritmo de detección de bordes Canny mediante `cv2.Canny`. A continuación, se buscan los contornos en la imagen utilizando `cv2.findContours`.

En la **línea 79**, se itera sobre cada contorno detectado y se aproxima su forma mediante `cv2.approxPolyDP` para obtener un polígono que lo represente. Si el polígono tiene exactamente cuatro lados, se calcula la relación de aspecto del rectángulo delimitado por el contorno. Si la relación de aspecto del rectángulo está entre 0.6 y 0.7, se considera que la imagen es un carné de identidad. Esta función nos ayuda a verificar si una imagen se parece a un carné de identidad para extraerlo y agregarlo como posible foto de perfil.

```
4 def get_similarity(post, curriculum):
5     score = fuzz.token_set_ratio(post, curriculum)
6     return score
```

Ilustración 29 Algoritmo para extraer la similitud entre una publicación y un currículum

En la ilustración 29, se utiliza una potente librería para extraer la similitud de los tokens relevantes de las publicaciones como del currículum, una vez procesado previamente todo el texto para realizarlo de una forma más eficiente. En la **línea 5**, calcula la similitud entre la publicación y el currículum del estudiante, utilizando el algoritmo de coincidencia de tokens "token set ratio" de la biblioteca FuzzyWuzzy.

El algoritmo tokeniza los textos en tokens individuales y luego calcula el puntaje de similitud basado en la cantidad y la posición de los tokens compartidos entre los textos. El puntaje de

similitud, denominado score, se calcula como un valor entre 0 y 100, donde un valor más alto indica una mayor similitud entre los textos.

```
118 file = sys.argv[1]
137 print(json.dumps(data_info))
138 sys.stdout.flush()
```

Ilustración 30 Recepción y envío de datos desde nodejs a python y de python a nodejs

En la figura 30, la **línea 118**, permite recibir los parámetros que enviamos desde nodejs a Python y en la **línea 137**, nos permite convertir las variables en un objeto para que no se altere la calidad de los datos y por último en la **línea 138**, se limpia el búfer del flujo de salida estándar (stdout) y escribe los datos a nodejs.

2.5.2. Algoritmos Importantes de Nodejs

```
6 exports.getPersonalInformation = async (req, res) => {
7   try {
8     const pdf = req.body.curriculum;
9     const email = req.body.email;
10    if (pdf) {
11      var buf = Buffer.from(pdf.replace(/data:application\/pdf;base64/, ''), 'base64');
12      AWS.config.update({ region: 'us-east-1' });
13      const SESConfig = {
14        apiVersion: "latest",
15        accessKeyId: process.env.AWS_SECURITY_TOKEN_ID,
16        secretAccessKey: process.env.AWS_SECURITY_TOKEN,
17        region: process.env.AWS_REGION
18      };
19      const s3 = new AWS.S3(SESConfig);
20      const params = {
21        Bucket: 'feupsimagefiles/curriculums/' + email,
22        Key: 'Curriculum.pdf',
23        Body: buf,
24        ContentEncoding: 'base64',
25        ContentType: 'application/pdf'
26      };
27      s3.upload(params, async (err, data) => {
28        if (data) {
29          const location = data.Location;
30          if (location) {
31            const pythonProcess = spawn('python', ["personalInformation.py", location, email]);
32            pythonProcess.stdout.on('data', (data) => {
33              const parsedData = JSON.parse(data.toString());
34              res.json({ status: 1, data: parsedData, msg: 'Archivo subido con éxito' });
35            });
36          }
37          pythonProcess.stderr.on('data', (data) => {
38            res.json({ status: 0, msg: data.toString() });
39          });
40        }
41      } else {
42        console.log('No existe la localización');
43        res.json({ status: 0, msg: 'Ha ocurrido un error, por favor vuelve a intentarlo' });
44      }
45    } else {
46      console.log('No se ha podido subir el archivo');
47    }
48  }
49 }
```

Ilustración 31 Algoritmo para analizar la hoja de vida del estudiante

En la ilustración 31, se visualiza una función asíncrona que maneja una solicitud HTTP para obtener información personal de un currículum.

En la **línea 8 y 9**, se comienza obteniendo el currículum en formato PDF y el correo electrónico de la solicitud por parte del cliente.

En la **línea 10**, se verifica si se proporcionó un currículum en formato PDF (pdf) en la solicitud. Si se cumple esta condición, se procede a realizar el procesamiento. En la **línea 11**, se crea un objeto Buffer a partir de los datos del currículum en base64, que se utiliza para almacenar temporalmente el currículum en memoria.

En la **línea 12**, se configuran las credenciales y la región de Amazon Web Services (AWS) para acceder a los servicios de AWS utilizando las variables de entorno correspondientes.

En la **línea 20**, se definen los parámetros necesarios para subir el currículum al bucket de S3, incluyendo el nombre del bucket, la clave del archivo, el cuerpo del archivo en formato Buffer, y la codificación y tipo de contenido del archivo.

En la **línea 27**, se utiliza el método upload del cliente S3 para cargar el currículum al bucket especificado. Se proporciona una función de devolución de llamada que se ejecutará una vez que se complete la carga.

En la **línea 32**, se ejecuta un proceso de Python utilizando el módulo spawn de la biblioteca child_process de Node.js. Se pasa como argumentos el script personalInformation.py, la ubicación del currículum y el correo electrónico.

En la **línea 33**, se configuran manejadores de eventos para capturar la salida estándar (stdout) y la salida de error (stderr) del proceso de Python. Si se recibe una salida estándar, se analiza y se envía como respuesta en formato JSON. Si se recibe una salida de error, se envía como respuesta un mensaje de error en formato JSON. Esto nos ayuda a recibir una solicitud HTTP para obtener información personal de un currículum, cargar el currículum en un bucket de S3 y procesarlo utilizando un script de Python externo.

```

13 exports.logout = async (req, res) => {
14   try {
15     const token = req.params.token;
16     const deleteSession = await Sessions.findOneAndDelete({ token: token })
17     .then(isDelete => {
18       if (isDelete) {
19         res.json({ status: 1 })
20       } else {
21         res.json({ status: 0, msg: "La sesión ya ha sido cerrada" })
22       }
23     })
24   } catch (error) {
25     res.json({ status: 0, msg: "Ha ocurrido un error inesperado, vuelve a intentarlo" })
26   }
27 }

```

Ilustración 32 Algoritmo para cerrar sesión

En la figura 32, se recibe el token de acceso de la sesión activa. En la **línea 16**, se compara el token en la base de datos para verificar si existe, si lo es; se elimina caso contrario envía un mensaje de error.

```

30 exports.signin = async (req, res) => {
31   try {
32     const { email, password } = req.body;
33     const user = await ViewUsers.findOne({ email: email });
34     if (user) {
35       const cmp = await bcrypt.compare(password, user.password);
36       if (cmp) {
37         const token = jwt.sign({ _id: user._id }, 'secretkey')
38         await Sessions.create({
39           token: token,
40           id_user: user._id
41         });
42         res.json({ status: 1, user, token: token })
43       } else {
44         var description = "Correo y/o contraseña incorrectos";
45         var code = "ING-001";
46         await Errors.create({
47           code: code,
48           severity: 5,
49           description: description,
50           module: 'Login',
51           id_user: user._id
52         });
53         res.json({ status: 0, msg: description, code: code });
54       }
55     } else {
56       var description = "El usuario que has ingresado no existe en nuestro sistema";
57       var code = "ING-002";
58       await Errors.create({
59         code: code,
60         severity: 5,
61         description: description,
62         module: 'Login',
63         id_user: null
64       });
65       res.json({ status: 0, msg: description, code: code });

```

Ilustración 33 Algoritmo para iniciar sesión

En la figura 33, en la **línea 32** se obtiene el correo electrónico y la contraseña proporcionados en la solicitud por parte del cliente. En la **línea 33**, se busca un usuario en la base de datos utilizando el correo electrónico proporcionado mediante la función `findOne` del modelo `ViewUsers`. Si se encuentra un usuario con el correo electrónico proporcionado, se continúa con la verificación de la contraseña.

En la **línea 35**, se utiliza la función `bcrypt.compare` para comparar la contraseña proporcionada con la contraseña almacenada del usuario. Si la comparación es exitosa, se procede a generar un token de autenticación utilizando la biblioteca `jsonwebtoken`. El token se firma con un secreto ('secretkey' en este caso) y contiene el ID del usuario.

En la **línea 38**, se crea un registro en la colección `Sessions` para almacenar el token de sesión y el ID del usuario. Se envía una respuesta exitosa en formato JSON que incluye el estado (`status: 1`), los datos del usuario (`user`), y el token de autenticación (`token`).

En caso de que la contraseña no coincida, se registra un error relacionado con las credenciales ingresadas y se envía una respuesta de error con un mensaje correspondiente.

Si no se encuentra un usuario con el correo electrónico proporcionado, se registra un error y se envía una respuesta de error con un mensaje correspondiente.

En caso de producirse un error durante el proceso, se registra un error genérico y se envía una respuesta de error con un mensaje correspondiente.

```

97 exports.sendCode = async (req, res) => {
98   try {
99     const correo = req.params.email;
100    const type = req.params.type;
101    const user = await Usuario.findOne({ email: correo });
102
103    var name = correo;
104    var code = makeid();
105    AWS.config.update({ region: 'us-east-1' });
106
107    const SESConfig = {
108      apiVersion: "latest",
109      accessKeyId: process.env.AWS_SECURITY_TOKEN_ID,
110      secretAccessKey: process.env.AWS_SECURITY_TOKEN,
111      region: process.env.AWS_REGION
112    }
113    let params = {
114      Destination: {
115        /* required */
116        ToAddresses: [correo, /* more items */]
117      },
118      Message: {
119        /* required */
120        Body: {
121          /* required */
122          Html: {
123            Data: `...`
124          },
125          Text: {
126            Charset: "UTF-8",
127            Data: `NOTIFICACIÓN DE SEGURIDAD`
128          }
129        },
130        Subject: {
131          Data: `...`
132        }
133      }
134    }
135
136    if (type == 'signup') {
137      if (user) {
138        new AWS.SES(SESConfig)
139          .sendEmail(params)
140          .promise()
141          .then(function (data) {
142            res.json({ status: 1, code: code, msg: 'Cuenta registrada exitosamente, porfavor inicia sesión' });
143          }).catch(
144            function (err) {
145              console.log(err);
146              res.json({ status: 0, msg: 'Ha ocurrido un error inesperado con el servidor de correos, porfavor reinicie y vuelva a intentarlo' });
147            }
148          );
149      } else {
150        res.json({ status: 0, msg: 'El correo electrónico ingresado ya se encuentra registrado en el sistema' });
151      }
152    } else {
153      if (user) {
154        new AWS.SES(SESConfig)
155          .sendEmail(params)
156          .promise()
157          .then(function (data) {
158            res.json({ status: 1, code: code, msg: 'Código generado exitosamente, porfavor revisa tu bandeja de entrada' });
159          }).catch(
160            function (err) {
161              console.log(err);
162              res.json({ status: 0, msg: 'Ha ocurrido un error inesperado con el servidor de correos, porfavor reinicie y vuelva a intentarlo' });
163            }
164          );
165      } else {
166        res.json({ status: 0, msg: 'El correo electrónico ingresado no se encuentra registrado en el sistema' });
167      }
168    }
169  }
170 }

```

Ilustración 34 Algoritmo para enviar códigos de verificación al correo

En la ilustración 34, la función se encarga de enviar un código de seguridad a través de correo electrónico.

En la **línea 99**, se obtiene el correo electrónico y el tipo de solicitud (signup o código de seguridad) de los parámetros de la solicitud enviados por el cliente.

En la **línea 101**, se busca un usuario en la base de datos utilizando el correo electrónico proporcionado mediante la función `findOne` del modelo `Usuario`.

En la **línea 104**, se generan un nombre y un código de seguridad utilizando la función `makeid()` que hace referencia a la creación del código de 6. Estos datos se utilizarán más adelante en el envío del correo electrónico.

En la **línea 107**, se configuran las credenciales y la región de Amazon Web Services (AWS) para acceder al servicio SES (Simple Email Service) utilizando las variables de entorno correspondientes.

En la **línea 113**, Se definen los parámetros necesarios para enviar el correo electrónico a través de SES, incluyendo el destino (correo electrónico), el contenido del mensaje (HTML y texto plano), el asunto del correo y la dirección de correo del remitente y de respuesta.

Se verifica el tipo de solicitud (signup o código de seguridad) y se realiza la lógica correspondiente en función de eso.

Si el tipo de solicitud es "signup", se verifica si existe un usuario con el correo electrónico proporcionado. Si no se encuentra un usuario, se utiliza el cliente de SES para enviar el correo electrónico con los parámetros definidos. Se envía una respuesta exitosa en formato JSON

indicando que la cuenta se registró correctamente. Todo esto para identificar cuando se trata de un registro y cuando olvidó su contraseña.

En caso de que no se pueda enviar el correo electrónico, se registra un error y se envía una respuesta de error correspondiente.

Si no se encuentra un usuario con el correo electrónico proporcionado, se envía una respuesta de error indicando que el correo electrónico no se encuentra registrado en el sistema.

En caso de producirse un error durante el proceso, se registra un error genérico y se envía una respuesta de error correspondiente.

```

19 exports.createPost = async (req, res) => {
20   try {
21     const infoPost = req.body;
22     const data = infoPost.data;
23     const file = data.img;
24     const token = infoPost.token;
25     let post = new Post(data);
26     const session = await Session.findOne({ token: token });
27     const type = session.id_user.type.type;
28     const id_user = session.id_user.id_profile_id;
29     let model_user = type == 'Docente' ? TeacherProfile : InstitutionProfile;
30     if (session) {
31       if (file.length > 0) {
32         var buf = Buffer.from(file.replace(/^data:image\/\w+;base64/, ''), 'base64');
33         AWS.config.update({ region: 'us-east-1' });
34         const SESConfig = {
35           apiVersion: "latest",
36           accessKeyId: process.env.AWS_SECURITY_TOKEN_ID,
37           secretAccessKey: process.env.AWS_SECURITY_TOKEN,
38           region: process.env.AWS_REGION
39         };
40         const s3 = new AWS.S3(SESConfig);
41         const params = {
42           Bucket: 'feupsimagefiles/posts/Computación',
43           Key: infoPost.name,
44           Body: buf,
45           ContentEncoding: 'base64',
46           ContentType: 'image/*'
47         };
48         s3.upload(params, async (err, data) => {
49           if (data) {
50             post.img = data.Location;
51             const url = new URL(post.img);
52             const filename = 'posts/Computación/' + url.pathname.split('/').pop();
53             const pythonProcess = spawn('python', ["naturalprocessing_system.py", filename, 'post']);
54             pythonProcess.stdout.on('data', async (data) => {
55               post.transcription = data.toString();
56             });
57             let postsave = await post.save();
58             if (postsave) {
59               const contributions = await model_user.findByIdAndUpdate(
60                 id_user,
61                 { $inc: { contributions: 1 } },
62                 { new: true }
63               );
64               if (contributions) {
65                 res.json({ status: 1, msg: 'La publicación ha sido realizada' });
66               } else {
67                 res.json({ status: 1, msg: 'Publicado pero no contribuido' });
68               }
69             } else {
70               res.json({ status: 0, msg: 'Ha ocurrido un error, porfavor vuelve a intentarlo' });
71             }
72           }
73         });

```

Ilustración 35 Algoritmo para la creación de publicaciones

En la ilustración 35, la función comienza obteniendo la información de la publicación a través de `req.body`, donde `infoPost` contiene los datos de la publicación y `data` contiene los detalles específicos de la misma. Además, se obtiene el token de autenticación del usuario a través de `infoPost.token`.

En la **línea 25**, se crea una nueva instancia del modelo `Post` utilizando los datos de la publicación.

En la **línea 26**, se busca la sesión correspondiente al token de autenticación proporcionado mediante `Session.findOne`. A partir de la sesión, se obtiene el tipo de usuario y el ID del perfil de usuario relacionado.

Se determina el modelo correspondiente al tipo de usuario (`TeacherProfile` para docentes y `InstitutionProfile` para instituciones) y se guarda en la variable `model_user`.

Si se encuentra la sesión correspondiente al token, se verifica si la publicación contiene una imagen adjunta (`file`). Si hay una imagen adjunta, se procede a cargarla en el bucket de S3 utilizando las credenciales y la configuración de región de AWS. Se crea un objeto `Buffer` a partir de la imagen en base64 y se definen los parámetros necesarios para la carga en S3.

Se utiliza el método `upload` del cliente S3 para cargar la imagen al bucket especificado. Se proporciona una función de devolución de llamada que se ejecutará una vez que se complete la carga.

Dentro de la función de devolución de llamada, se verifica si la carga fue exitosa y se obtiene la URL de la imagen cargada. Se extrae el nombre del archivo de la URL y se utiliza para realizar un proceso de procesamiento de lenguaje natural utilizando un script de Python externo

(`naturalprocessing_system.py`). El resultado del procesamiento se asigna a la propiedad `transcription` del objeto `post`.

En la **línea 57**, se guarda la publicación en la base de datos. Si el guardado es exitoso, se incrementa el contador de contribuciones del usuario correspondiente mediante `model_user.findByIdAndUpdate`. Se envía una respuesta exitosa en formato JSON indicando que la publicación se ha realizado correctamente.

En caso de que no se adjunte una imagen a la publicación, se asigna el contenido de la descripción a la propiedad `transcription` del objeto `post`.

Se guarda la publicación en la base de datos y se realiza el mismo proceso de incrementar el contador de contribuciones del usuario correspondiente.

Finalmente, se envía una respuesta exitosa en formato JSON indicando que la publicación se ha realizado correctamente.

Si no se encuentra la sesión correspondiente al token de autenticación, se envía una respuesta de error indicando que no se puede realizar la publicación.

En caso de producirse un error durante el proceso, se registra un error genérico y se envía una respuesta de error correspondiente.

```

111 exports.getPosts = async (req, res) => {
112   ... try {
113     ... const num_skip = req.params.docs;
114     ... const num_limit = 10;
115     ... const posts = await ViewPosts.find().skip(num_skip).limit(num_limit)
116     ... res.json({ status: 1, data: posts });
117   } catch (error) {
118     ... console.log(error);
119     ... var description = "Ha ocurrido un error al obtener las publicaciones";
120     ... var code = "PST-001";
121     ... await Errors.create({
122     ...     code: code,
123     ...     severity: 5,
124     ...     description: description,
125     ...     module: 'Posts',
126     ...     id_user: null
127     ... });
128     ... res.json({ status: 0, msg: description, code: code });
129   }
130 }

```

Ilustración 36 Algoritmo para la obtención de publicaciones

En la ilustración 36, se obtienen los parámetros de la solicitud, donde req.params.docs indica la cantidad de documentos a omitir en la consulta y num_limit se establece en 10 como límite de documentos a obtener. Todo esto para escalar la traída de datos y limitar 10 documentos por petición.

```

... const posts = await ViewPosts.aggregate(
...   [
...     {
...       $match:
...       {
...         '_id': { $nin: information_student.id_profile.deserted },
...         'experience': information_student.id_profile.experience,
...         'time': { $in: information_student.id_profile.time },
...         'assist': { $in: information_student.id_profile.assist },
...         'benefits': { $in: information_student.id_profile.benefits }
...       },
...     }, {
...       $limit: 10
...     },
...   ],
... )

```

Ilustración 37 Encontrar publicaciones con filtros

En la figura 37, se pueden filtrar las publicaciones dependiendo de los criterios de búsqueda en base al perfil del usuario en este la caso la experiencia, el tiempo (completo, medio tiempo), assist (presencial, remoto), benefits (Remunerado, no remunerado). Limitando siempre a 10 la traída de los datos.

```
22 .....if (body && user && profile) {
23 .....const userType = await TypeUsers.findOne({ type: user.type });
24 .....if (userType) {
25 .....let userInfo = new User(user);
26 .....let profileInfo = user.type == 'Estudiante' ? new StudentProfile(profile) : user.type == 'Docente' ? new TeacherProfile(profile) : user.type == 'Profesor' ? new ProfessorProfile(profile) : new Profile();
27 .....if (userInfo && profileInfo) {
28 .....const hashedPwd = await bcrypt.hash(user.password, 10);
29 .....userInfo.type = new ObjectId(userType._id);
30 .....userInfo.id_profile = new ObjectId(profileInfo._id);
31 .....userInfo.password = hashedPwd;
32 .....
33 .....let accountUser = await userInfo.save();
34 .....
35 .....if (user.type == 'Estudiante') {
36 .....const pythonProcess = spawn('python', ["naturalprocessing_system.py", user.email, 'curriculum']);
37 .....pythonProcess.stdout.on('data', async (data) => {
38 .....const parsedData = JSON.parse(data.toString());
39 .....profileInfo.cv = parsedData.content;
40 .....let accountInfo = await profileInfo.save();
41 .....if (accountUser && accountInfo) {
42 .....res.json({ status: 1, msg: 'Cuenta registrada con éxito' });
43 .....} else {
44 .....res.json({ status: 0, msg: data.toString() });
45 .....}
46 .....});
47 .....
48 .....pythonProcess.stderr.on('data', async (data) => {
49 .....{
50 .....res.json({ status: 0, msg: data.toString() });
51 .....})
52 .....} else {
53 .....let accountInfo = await profileInfo.save();
54 .....if (accountUser && accountInfo) {
```

Ilustración 38 Algoritmo para la creación de perfiles y usuarios en mongodb

La ilustración 38, se encarga de guardar la información de un usuario en el sistema.

En la **línea 18**, se obtienen los datos proporcionados en el cuerpo de la solicitud (`req.body`). Se extraen la información del usuario (`body.userInfo`) y la información personal (`body.personalInfo`) que son traídos a través del cliente.

En la **línea 22**, se verifica si se proporcionaron los datos necesarios (`body`, `user` y `profile`). Si se cumplen las condiciones, se continúa con el proceso de registro del usuario.

En la **línea 23**, se busca el tipo de usuario correspondiente en la base de datos utilizando el valor del campo `type` del objeto `user` mediante la función `findOne` del modelo `TypeUsers`. Si se encuentra el tipo de usuario, se continúa con el proceso de registro.

En la **línea 26**, se crean instancias de los modelos `User` y `Profile` correspondientes según el tipo de usuario. Se asignan los datos correspondientes a cada instancia, como el tipo de usuario, el perfil de usuario y la contraseña. La contraseña se cifra utilizando la función `bcrypt.hash` para almacenarla de forma segura en la base de datos.

En la **línea 33**, se guarda el objeto `userInfo` en la base de datos mediante la función `save()`. Si el tipo de usuario es "Estudiante", se realiza un procesamiento adicional de información mediante un script de Python externo (`naturalprocessing_system.py`) para obtener el contenido del currículum del estudiante. El resultado se asigna a la propiedad `cv` del objeto `profileInfo`.

Si el proceso de guardado es exitoso para ambos objetos, se envía una respuesta exitosa en formato JSON indicando que la cuenta se ha registrado correctamente.

En caso de que ocurra un error durante el proceso, se registra un error y se envía una respuesta de error correspondiente con un mensaje descriptivo.

La función `saveUser` es útil para manejar el registro de usuarios en el sistema, almacenando la información del usuario y el perfil correspondiente en la base de datos. Además, realiza un procesamiento adicional en caso de usuarios del tipo "Estudiante" para obtener el contenido del currículum.

```

92  exports.getUserById = async (req, res) => {
93    try {
94      let user = await ViewUsers.findById({ _id: req.params.id });
95      if (user) {
96        res.status(404).json({ msg: "No existe un Registrado" });
97      }
98      res.json(user)
99    }
100   catch (error) {
101     console.log(error);
102     res.status(500).send("Ha ocurrido un error");
103   }
104 }

```

Ilustración 39 Algoritmo para obtener los usuarios por un identificador en específico

En la ilustración 39, se realiza una consulta a la base de datos utilizando el modelo ViewUsers y el método findById. Se busca un usuario cuyo _id coincide con el ID proporcionado. El cual es traído a través del cliente y envía la información en el caso de que exista.

```

1  const mongoose = require('mongoose');
2
3  mongoose.connect(process.env.MONGODB_UI, {
4    useNewUrlParser: true,
5    useUnifiedTopology: true
6  })
7    .then(db => console.log('Base conectada'))
8    .catch(err => console.log(err));

```

Ilustración 40 Algoritmo para conexión a mongodb

En la ilustración 40, se utiliza el método connect() de mongoose para establecer la conexión a la base de datos. Se proporciona la URL de conexión a MongoDB como primer argumento (process.env.MONGODB_UI), que se obtiene de una variable de entorno. Además, se pasan algunas opciones de configuración, como useNewUrlParser: true para evitar advertencias de deprecación y useUnifiedTopology: true para utilizar el nuevo motor de topología unificada de MongoDB.

2.5.3. Algoritmos Importantes de Angular

```
15 | *signIn(email, password): Observable<any> -{
16 | * *const user = -{
17 | * * *email: email,
18 | * * *password: password
19 | * * *}
20 | * *return this.http.post<any>(this.URL + '/signin', user)
21 | * *}
22 |
23 | *getRole(id: string): Observable<any> -{
24 | * *return this.http.get(this.URL + '/role/' + id);
25 | * *}
26 |
27 | *sendEmailCode(email: string, type: string): Observable<any> -{
28 | * *return this.http.get<any>(this.URL + '/code/' + type + '/' + email)
29 | * *}
30 | *updatePassword(user: -{email: string, password: string}): Observable<any> -{
31 | * *return this.http.put(this.URL, user);
32 | * *}
33 | *logout(id: string): Observable<any> -{
34 | * *return this.http.delete(this.URL + '/' + id);
35 | * *}
13 | *getPersonalInformationCV(pdfdata, email): Observable<any> -{
14 | * *const information = -{
15 | * * *curriculum: pdfdata,
16 | * * *email: email
17 | * * *}
18 | * *return this.http.post(this.url + 'personal', information);
19 | * *}

16 | *createPost(post: Post, filename, token): Observable<any> -{
17 | * *const formFile = -{
18 | * * *data: post,
19 | * * *name: filename ? filename : '',
20 | * * *token: token
21 | * * *}
22 | * *return this.http.post(this.URL + '/teacher', formFile)
23 | * *}
24 |
25 |
26 | *getPosts(docs): Observable<any> -{
27 | * *return this.http.get(this.URL + '/teacher/' + docs)
28 | * *}
29 | *getPostsById(id): Observable<any> -{
30 | * *return this.http.get(this.URL + '/student/' + id)
31 | * *}
32 |
33 | *updatePost(id, array): Observable<any> -{
34 | * *return this.http.put(this.URL + '/student/' + id, array);
35 | * *}
```

Ilustración 41 Servicios generales para envío de peticiones a nodejs (Backend)

En la ilustración 41, para recibir los datos de los usuarios y enviar hacia el backend se utilizan servicios de angular para cada tipo de función dependiendo de su tipo en este caso (PUT, GET, POST, DELETE). Para hacer uso en publicaciones, usuarios y hojas de vida.

```
14  ... [{
15  ...   path: '',
16  ...   redirectTo: 'login'
17  ... }, {
18  ...   path: 'login',
19  ...   component: LoginComponent,
20  ...   canActivate: [AuthGuard],
21  ...   data: {
22  ...     role: ['Administrador', 'Estudiante', 'Docente', 'Empresa', null]
23  ...   }
24  ... },
25  ... {
26  ...   path: 'panel-admin',
27  ...   pathMatch: 'full',
28  ...   component: AdminPanelComponent,
29  ...   canActivate: [AuthGuard],
30  ...   data: {
31  ...     role: ['Administrador']
32  ...   }
33  ... },
34  ... {
35  ...   path: 'panel-empresa',
36  ...   pathMatch: 'full',
37  ...   component: BusinessPanelComponent,
38  ...   canActivate: [AuthGuard],
39  ...   data: {
40  ...     role: ['Empresa']
41  ...   }
42  ... },
43  ... {
44  ...   path: 'panel',
45  ...   pathMatch: 'full',
46  ...   component: PanelComponent,
47  ...   canActivate: [AuthGuard],
48  ...   data: {
49  ...     role: ['Estudiante']
```

Ilustración 42 Algoritmo para la protección y control de rutas

En la ilustración 42, se especifican las rutas a los que pueden acceder cada tipo de usuario dependiendo de su rol, si es que no pertenece a ningún rol se redireccionará a una página de 404.

```

76 ..... <!-- Todas las publicaciones -->
77 ..... <div *ngIf="lst_posts.length > 0 && !loader" class="row mb-3 w-100"
78 ..... style="border-radius: 8px !important;">
79 ..... <div class="row w-100">
80 ..... <div class="card shadow-sm w-100" style="border-radius: 8px !important;">
81 ..... <div class="align-items-center d-flex p-3">
82 ..... <div class="me-2">
83 ..... <p-avatar
84 ..... *ngIf="lst_posts[0]?.id_user[0]?.id_profile?.badge?.type === 'image'"
85 ..... image="/assets/images/{{lst_posts[0]?.id_user[0]?.id_profile?.badge?.content}}"
86 ..... shape="circle"></p-avatar>
87 ..... <p-avatar
88 ..... *ngIf="lst_posts[0]?.id_user[0]?.id_profile?.badge?.type === 'letters'"
89 ..... [label]="initials" styleClass="mr-2"
90 ..... [style]="{'background-color': lst_posts[0]?.id_user[0]?.id_profile?.badge?.content, 'color': '#f
91 ..... shape="circle"></p-avatar>
92 ..... </div>
93 ..... <div class="w-100">
94 ..... <h5 class="mb-0" style="font-size: 13px;font-weight: 700;color: #3e4253">
95 ..... {{lst_posts[0]?.id_user[0]?.names + ' ' +
96 ..... lst_posts[0]?.id_user[0]?.lastnames}}</h5>
97 ..... <p class="mb-0 mt-1"
98 ..... style="font-size: 12px;color: #73778b; line-height: 16px;">
99 ..... {{lst_posts[0]?.date_publication | date: 'dd/MM/yyyy hh:mm'}}</p>
100 ..... </div>
101 .....
102 .....
103 ..... <div class="text-end justify-content-end d-flex m-auto w-75">
104 ..... <p-tag
105 ..... [severity]="lst_posts[0]?.score >= 70 ? 'success': lst_posts[0]?.score >= 50 && lst_posts[0]?.sc
106 ..... class="me-4 mt-1"
107 ..... [icon]="lst_posts[0]?.score >= 70 ? 'fa-solid fa-hands-clapping': lst_posts[0]?.score >= 50 && 1
108 ..... [value]="lst_posts[0]?.score >= 70 ? 'Super compatibles': lst_posts[0]?.score >= 50 && lst_posts
109 ..... [rounded]="true"></p-tag>

```

Ilustración 43 Algoritmo para la implementación de la información en el panel del estudiante

En la ilustración 43 se verifica el contenido HTML en el que se imprimen los datos para que el cliente pueda visualizar en el navegador.

```

1630     .getTimer() -{
1631     . . .const timer = localStorage.getItem('timerRunning');
1632     . . .if (timer) -{
1633     . . . .this.timerRunning = localStorage.getItem('timerRunning') !== 'false' ? false : true;
1634     . . . .if(this.timerRunning == true){
1635     . . . . .this.UserForm.email = localStorage.getItem('email');
1636     . . . . .this.UserForm.type = localStorage.getItem('type');
1637     . . . . .this.InstitutionForm.ruc = localStorage?.getItem('ruc');
1638     . . . . .this.timer();
1639     . . . . }
1640     . . . }
1641     . }
1642
1643     .timer() -{
1644     . . .this.countdown = Number.parseInt(localStorage.getItem('countdown'));
1645     . . .const timer = setInterval(() => -{
1646     . . . .this.countdown -= 1;
1647     . . . . localStorage.setItem('countdown', this.countdown.toString());
1648     . . . . if (this.countdown === 0) -{
1649     . . . . .clearInterval(timer);
1650     . . . . .this.timerRunning = false;
1651     . . . . .this.countdown = 60;
1652     . . . . .localStorage.setItem('countdown', '60');
1653     . . . . .localStorage.setItem('timerRunning', 'false');
1654     . . . . }
1655     . . . }, 1000);
1656
1657

```

Ilustración 44 Algoritmo para control de envío de correos masivos

En la ilustración 44, esta función se implementó para crear un contador de 1 minuto al momento de enviar un correo electrónico. Todo esto como medida de seguridad para el envío de correos masivos.

```

1668     signIn() {
1669         this.ldg_login = true;
1670         const email = this.loginForm.get('email')?.value;
1671         const password = this.loginForm.get('password')?.value;
1672
1673         if (email && password) {
1674             this.authService.signIn(email, password).subscribe(
1675                 (res) => {
1676                     if (res.status === 1) {
1677                         const data = res.user;
1678                         const type = data.type.type;
1679                         this.ldg_login = false;
1680                         if (type === 'Estudiante') {
1681                             this.cacheStudent.id = data._id;
1682                             this.cacheStudent.token = res.token;
1683                             this.cacheStudent.type = type;
1684                             this.cacheStudent.fullname = data.names + '.' + data.lastnames;
1685                             this.cacheStudent.badge.type = data.id_profile.badge.type;
1686                             this.cacheStudent.career = data.id_profile.career;
1687                             this.cacheStudent.badge.content = data.id_profile.badge.content;
1688                             this.cacheStudent.years = data.id_profile?.years;
1689                             localStorage.setItem('user_session_data', JSON.stringify(this.cacheStudent))
1690                             this.router.navigate(['/panel']);
1691                         } else {
1692                             if (type === 'Docente') {
1693                                 this.cacheTeacher.id = data._id;
1694                                 this.cacheTeacher.token = res.token;
1695                                 this.cacheTeacher.type = type;
1696                                 this.cacheTeacher.fullname = data.names + '.' + data.lastnames;
1697                                 this.cacheTeacher.email = data.email;
1698                                 this.cacheTeacher.badge.type = data.id_profile.badge.type;
1699                                 this.cacheTeacher.badge.content = data.id_profile.badge.content;
1700                                 this.cacheTeacher.career = data.id_profile.career;
1701                                 this.cacheTeacher.contributions = data.id_profile.contributions;
1702                                 localStorage.setItem('user_session_data', JSON.stringify(this.cacheTeacher))
1703                             }

```

Ilustración 45 Algoritmo para la obtención de datos de inicio de sesión para la parte del cliente

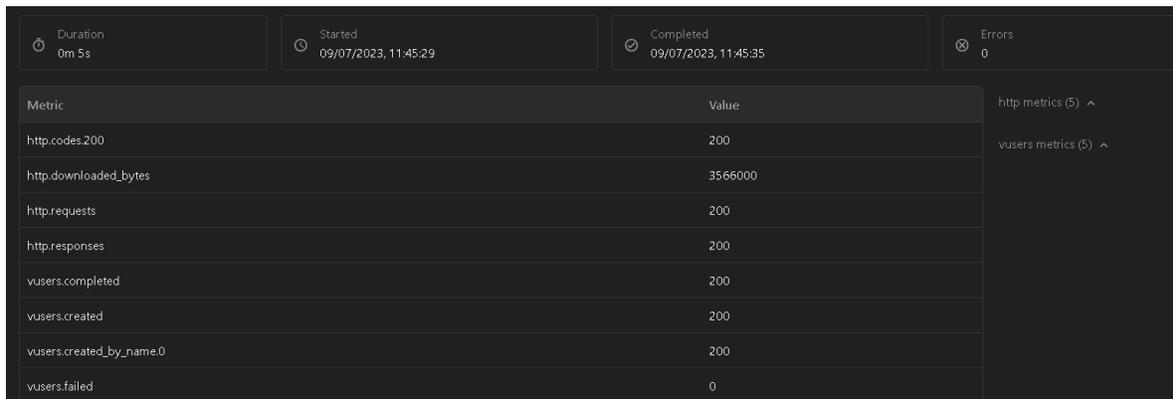
En la ilustración 45, el algoritmo permite enviar una petición al backend para la traída de información de inicio de sesión del usuario dependiendo del rol a través de nodejs, express y mongodb. Para luego setear las variables e imprimir en el navegador tal cual la **ilustración 45**

.Pruebas De Funcionalidad

2.6.Pruebas De Carga Y Estrés

Herramienta: **Artillery**

1. Prueba de carga 200 usuarios accediendo al sistema simultáneamente.



The screenshot displays the Artillery test results interface. At the top, there are four summary boxes: Duration (0m 5s), Started (09/07/2023, 11:45:29), Completed (09/07/2023, 11:45:35), and Errors (0). Below these is a table of metrics with columns for Metric and Value. To the right of the table, there are expandable sections for 'http metrics (5)' and 'users metrics (5)'. The table data is as follows:

Metric	Value
http.codes.200	200
http.downloaded_bytes	3566000
http.requests	200
http.responses	200
users.completed	200
users.created	200
users.created_by_name.0	200
users.failed	0

Ilustración 46 Prueba de carga 200 usuarios

Resultados:

1. Contadores:

- Se crearon 200 usuarios virtuales.
- Se realizaron un total de 200 solicitudes HTTP.
- Hubo 200 respuestas exitosas con el código de estado 200.
- Se descargaron 3,566,000 bytes en total.
- No hubo usuarios virtuales que fallaran.

- Se completaron 200 usuarios virtuales.

2. Tasas:

- La tasa de solicitud HTTP promedio fue de 100 solicitudes por segundo.

3. Resumen de tiempos de respuesta:

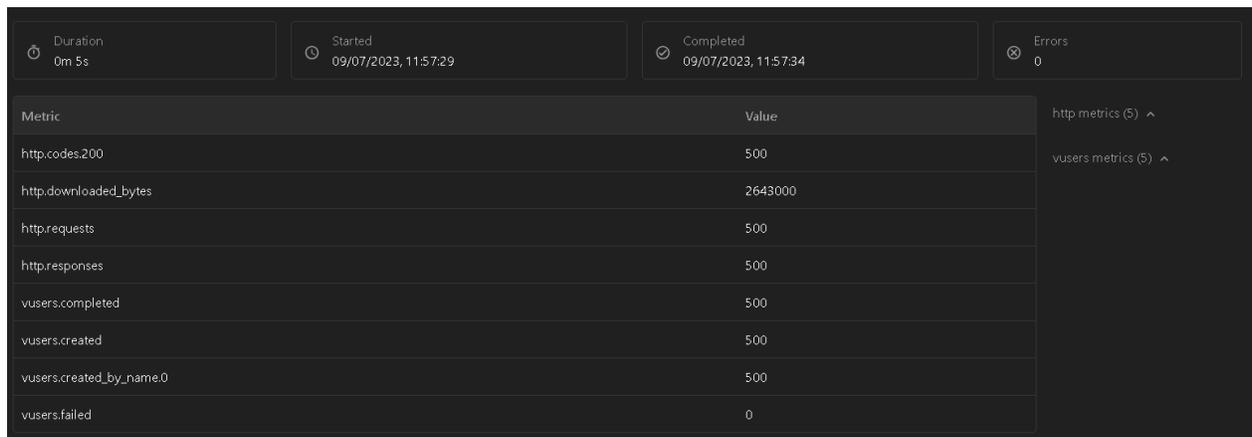
- El tiempo de respuesta mínimo fue de 347 ms.
- El tiempo de respuesta máximo fue de 4,787 ms.
- La mediana (p50) del tiempo de respuesta fue de 2,725 ms.
- El percentil 95 (p95) del tiempo de respuesta fue de 3,752.7 ms.
- El percentil 99 (p99) del tiempo de respuesta fue de 4,231.1 ms.

4. Resumen de duración de sesión de usuarios virtuales:

- La duración mínima de la sesión de un usuario virtual fue de 357 segundos.
- La duración máxima de la sesión de un usuario virtual fue de 4,789.7 segundos.
- La mediana (p50) de la duración de la sesión fue de 2,725 segundos.
- El percentil 99 (p99) de la duración de la sesión fue de 4,231.1 segundos.

Para concluir, se observa que los 200 usuarios virtuales creados lograron completar sus sesiones correctamente sin errores. Los tiempos de respuesta y la duración de la sesión tienen valores razonables y consistentes, con la mediana en torno a 2,725 segundos. Esto indica un rendimiento satisfactorio del sistema durante la prueba.

2. Prueba de carga 500 usuarios accediendo al sistema simultáneamente.



The screenshot shows a performance testing dashboard with the following data:

Metric	Value
http.codes.200	500
http.downloaded_bytes	2643000
http.requests	500
http.responses	500
vusers.completed	500
vusers.created	500
vusers.created_by_name.0	500
vusers.failed	0

Additional dashboard information: Duration: 0m 5s, Started: 09/07/2023, 11:57:29, Completed: 09/07/2023, 11:57:34, Errors: 0.

Ilustración 47 Prueba de carga 500 usuarios

Se crearon un total de 500 usuarios virtuales y se realizaron 500 solicitudes HTTP. Todos los usuarios virtuales completaron sus sesiones sin ninguna falla, lo que indica un rendimiento exitoso.

Los contadores muestran que se obtuvieron 500 respuestas exitosas con el código de estado 200, y el tamaño total de bytes descargados fue de 2,643,000.

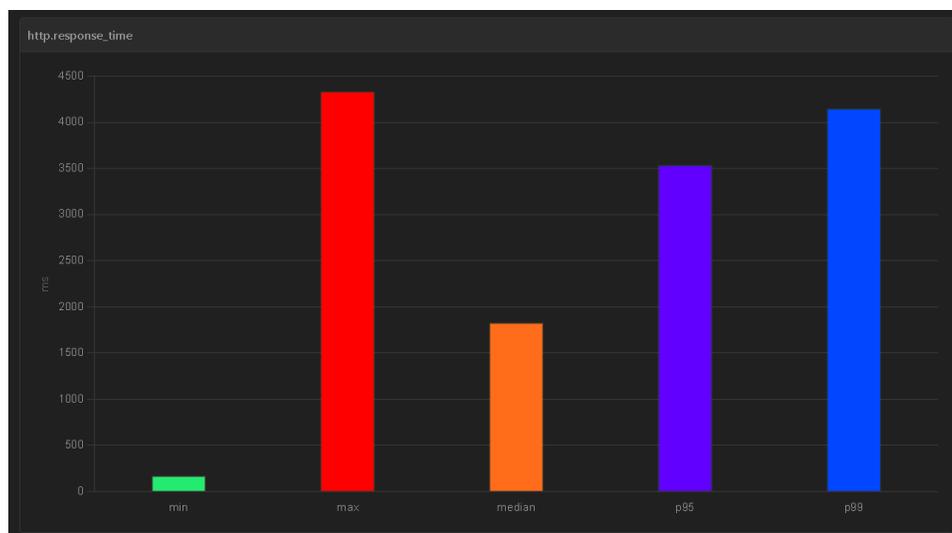


Ilustración 48 Histograma de tiempos de respuesta prueba de carga 500 usuarios

Los resúmenes e histogramas indican que los tiempos de respuesta de las solicitudes HTTP variaron entre 168 y 4,333 milisegundos. El tiempo promedio de respuesta (p50) fue de aproximadamente 1,826.6 milisegundos. La duración de la sesión de los usuarios virtuales también varió entre 173.5 y 4,335.7 milisegundos, con una mediana de 1,826.6 milisegundos.

Para concluir, el informe muestra que el sistema pudo manejar con éxito la carga de 500 usuarios virtuales, completando todas las solicitudes sin errores. Los tiempos de respuesta de las solicitudes y las duraciones de sesión de los usuarios virtuales se encuentran dentro de los rangos esperados. Esto sugiere un rendimiento satisfactorio del sistema durante las pruebas realizadas.

3. Prueba de estrés 500 usuarios accediendo al sistema simultáneamente con 20 peticiones cada uno.

Metric	Value
errors.ETIMEDOUT	1
http.codes.200	9988
http.downloaded_bytes	52796568
http.requests	9989
http.responses	9988
vusers.completed	499
vusers.created	500
vusers.created_by_name.0	500
vusers.failed	1

Ilustración 49 Prueba de estrés 500 usuarios con 20 peticiones

Durante el período de la prueba, se crearon 500 usuarios virtuales y se realizaron un total de 500 solicitudes HTTP. Todas las solicitudes fueron exitosas, excepto una que arrojó un error de tiempo de espera (ETIMEDOUT). Además, se descargaron un total de 52,796,568 bytes.

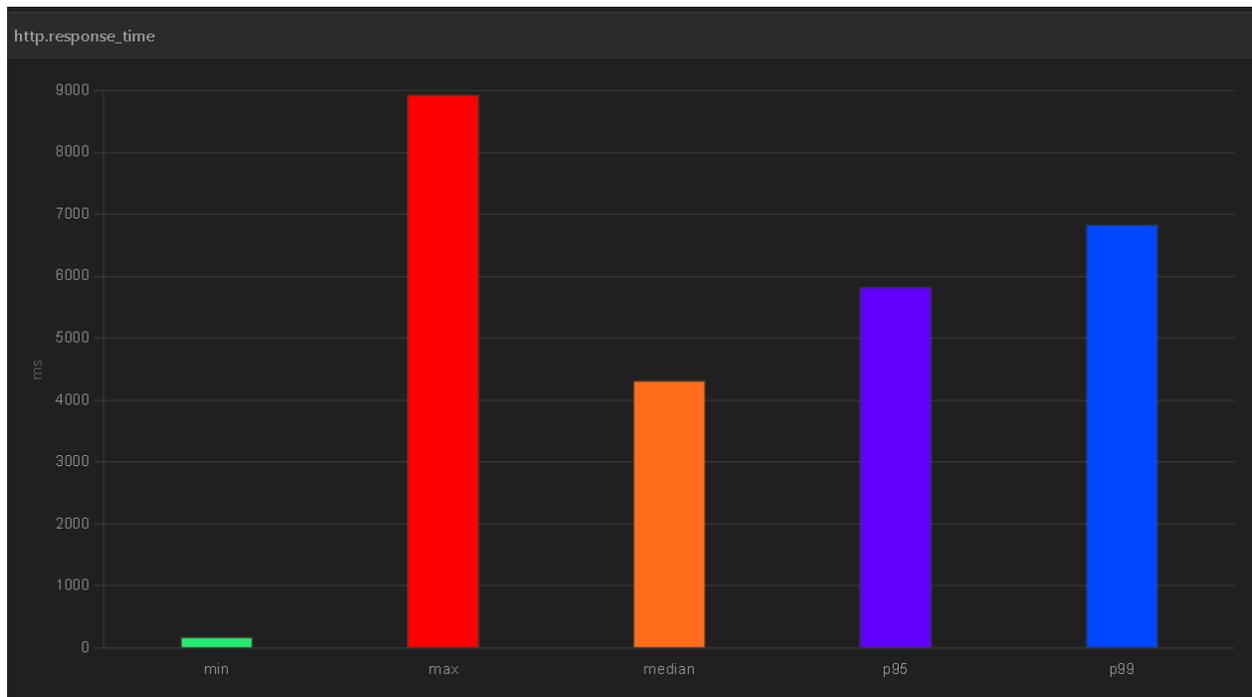


Ilustración 50 Histograma de prueba de estrés 500 usuarios 20 solicitudes cada uno

Además, se muestra que el tiempo mínimo fue de 176 ms y el máximo de 8,932 ms, concluyendo que la prueba de rendimiento fue exitosa con una alta tasa de éxito en las solicitudes y un tiempo de respuesta aceptable. Sin embargo, se identificó un error de tiempo de espera en una de las solicitudes por lo cual no se logró completar.

2.7. Pruebas de caja negra

N	Módulo	Acción	Resultado
1	Inicio Sesión	Ingreso de correo incorrecto	Correo y/o contraseña incorrectos
2		Ingreso de clave incorrecta	Correo y/o contraseña incorrectos
3		Ingreso de usuario no registrado	El usuario que has ingresado no existe en nuestro sistema
4		Ingreso de datos correctos	Ha ocurrido un error inesperado en el sistema, por favor reinicie y vuelva a intentarlo
5		Ingreso de datos correctos	Inicio de sesión

Tabla 20. Prueba caja negra, módulo login

N	Módulo	Acción	Resultado
1	Registro de usuarios	Ingreso de correo	El correo electrónico ingresado ya se encuentra registrado en el sistema
2		Ingreso de correo	Código de verificación enviado al correo electrónico
3		Ingreso de correo	El correo electrónico no corresponde al de un estudiante.
4		Ingreso de correo	El correo electrónico no corresponde al de un docente.
5		Ingreso de correo	El correo electrónico no corresponde al de una empresa.
6		Ingreso de correo	Por favor completa todos los campos e ingresa datos válidos.
7		Ingreso de correo	Ha ocurrido un error inesperado con el servidor de correos, por favor reinicie y vuelva a intentarlo
8		Ingreso de correo	Despliegue panel de ingreso de código
9		Ingreso de código	Código inválido
10		Ingreso de código	Despliegue formulario de actualización de contraseña
11		Ingreso de contraseña	Los campos no pueden estar vacíos
12		Ingreso de contraseña y verificación	Los campos deben contener al menos 8 caracteres y máximo 32.
13		Ingreso de contraseña y verificación	Por favor, asegúrate de ingresar la misma contraseña en ambos campos.
14		Ingreso de contraseña y verificación	Despliegue formulario de información personal
15		Continuar	Por favor, elige una imagen como tu avatar antes de continuar.
16		Continuar	Por favor, elige al menos un idioma antes de continuar.
17		Continuar	Despliegue formulario carga de datos
18		Método de carga eficiente	Despliegue formulario de carga del curriculum vitae.
19		Selecciona curriculum	Tamaño de archivo inválido
20		Selecciona curriculum	Tamaño máximo de carga 3mb
21		Selecciona curriculum	Tipo de archivo inválido
22		Selecciona curriculum	Solo se aceptan archivos .pdf

Tabla 21. Prueba caja negra, módulo registro de usuarios

N	Módulo	Acción	Resultado
1	Estudiante	Ingreso al panel	Ha ocurrido un error al obtener las publicaciones
2		Ingreso al panel	No hay publicaciones disponibles por el momento
3		Ingreso al panel	Carga de publicaciones
4		Actualización de datos	Ha ocurrido un error
5		Actualización de datos	Usuario actualizado con éxito

Tabla 22. Prueba caja negra, módulo estudiante

N	Módulo	Acción	Resultado
1	Empleador	Ingreso al panel	Ha ocurrido un error al obtener los perfiles
2		Ingreso al panel	No hay perfiles disponibles por el momento
3		Ingreso al panel	Carga de perfiles
4		Actualización de datos	Ha ocurrido un error
5		Actualización de datos	Usuario actualizado con éxito
6		Ver publicaciones	Ha ocurrido un error al obtener las publicaciones
7		Crear publicaciones	Ha ocurrido un error, porfavor vuelve a intentarlo
8		Crear publicaciones	Publicado, pero no contribuido
9		Crear publicaciones	Ha ocurrido un error al subir el archivo, intenta de nuevo
10		Crear publicaciones	La publicación ha sido realizada

Tabla 23. Prueba caja negra, módulo empleador

N	Módulo	Acción	Resultado
1	Docente	Actualización de datos	Ha ocurrido un error
2		Actualización de datos	Usuario actualizado con éxito
3		Ver publicaciones	Ha ocurrido un error al obtener las publicaciones
4		Crear publicaciones	Ha ocurrido un error, porfavor vuelve a intentarlo
5		Crear publicaciones	Publicado, pero no contribuido
6		Crear publicaciones	Ha ocurrido un error al subir el archivo, intenta de nuevo
7		Crear publicaciones	La publicación ha sido realizada

Tabla 24. Prueba caja negra, módulo docente

2.8. Análisis de resultados

En este aspecto se busca evaluar la efectividad y el impacto del aplicativo web de la bolsa de trabajo en la búsqueda de empleo por parte de los estudiantes. Se busca determinar si el sistema ha logrado agilizar el proceso de identificación de una oferta laboral de acuerdo con sus necesidades, además de que los estudiantes accedan a este contenido de una manera más rápida y eficiente.

2.8.1. Obtención de datos:

Se utilizó un dataset de 111 imágenes de publicaciones de ofertas de empleo y pasantías del grupo de “bolsa de trabajo pasantía” de la carrera de computación. A partir de esto se realizó el procesamiento de texto para obtener las palabras relevantes de cada una de las publicaciones, así como el tipo al que pertenece “Bolsa de trabajo, pasantías”, la ubicación, el salario y la experiencia.



Ilustración 51 Imagen de referencia grupo Bolsa de trabajo pasantía

2.8.2. Etiquetas e identificadores cualitativos:

Número	Etiqueta	Porcentaje
1	Súper compatibles	70% - 100%
2	Buena coincidencia	50% – 69.99%
3	Puede ser buena opción	25% - 49.99%
4	Baja coincidencia	0% - 24.99%

2.8.3. Coincidencia entre las publicaciones y el currículum:


Alberto Rusbel Duchi Bastidas
 31/12/1969 07:00
 Puede ser una buena opción!

"Se busca talento excepcional en computación: ¡Únete a nuestra familia tecnológica!"

Estudiante #1: Paúl Morales

Publicación: Ilustración 52

Intereses: Pasantías, Bolsa de empleo

Coincidencia Talenter: 39%

Coincidencia Manual: 30%

Identificador: #3



Gobierno del Ecuador
SE REQUIERE ESTUDIANTES PARA PASANTÍAS EN:

- MATERNIDAD ISIDRO AYORA
- REDES
- SOPORTE
- DESARROLLO
COMUNICARSE CON MGS. JIMMY MURILLO 0993922393
- HOSPITAL BACA ORTIZ
- REDES
- SOPORTE
- DESARROLLO
COMUNICARSE CON ING. SONIA BUÑAY 0961142001
- UNIDADES ZONA 9
- REDES
- SOPORTE
- DESARROLLO
COMUNICARSE CON MGS. RICHARD MURILLO 0999718624

ATENTAMENTE
ING. RICHARD MURILLO MIA MGS
 ANALISTA DE GESTIÓN DE SOPORTE TÉCNICO Y REDES

Ministerio de Salud Pública
 Coordinación Zonal 9 - Salud
 Dirección de Salud Laboral y Seguridad Social
 Código postal: 17032 - Quito - Ecuador
 Teléfono: 09912 2 200 0000
 saludlaboral@guia.gub.ec

Pasantía Computación Finaliza el: 31/12/1969 07:00

Ilustración 52 Imagen de oferta laboral 1

AD Alberto Rusbel Duchí Bastidas 10/12/2023 12:00 Buena coincidencia ***

"¡Haz historia en la industria de la tecnología! Únete a nuestro equipo."

Buscamos:
FULL STACK DEVELOPER (COBIS DEVELOPER)

Titulado en Ingeniería en Sistemas o Similares.

Experiencia:

- Mínimo 1 año en proyectos de tecnología para productos COBIS Posivos

Conocimientos

- Framework COBIS (VB6, VBScript, SQL)
- Conocimiento de base de datos (Sybase i6, SqlServer)
- Microservicios con framework spring boot y quarkus
- Framework angular para desarrollo typescript, javascript

¡Trabaja con nosotros!

talentohumano@gizlocorp.com 096 834 8054

Bolsa Guayaquil Computación Finaliza el: 31/12/1969 07:00 1 año(s) experiencia

Ilustración 53 Imagen de oferta laboral 2

Estudiante #2: Michael Robles

Publicación: Ilustración 53

Intereses: Bolsa de empleo

Coincidencia Talenter: 54%

Coincidencia Manual: 50%

Identificador: #2

AD Alberto Rusbel Duchí Bastidas 08/07/2023 11:21 Baja coincidencia ***

"Necesito un chef"

OFERTA LABORAL CENTRO DE EGRESADOS

COMERCIALIZADORA AGROSIGO S.A.S N°118-2018

TÉCNICOS EN: GASTRONOMÍA INTERNACIONAL

DESCRIPCIÓN DEL CARGO: Se solicita Chef ejecutivo comercial, La persona debe tener conocimientos en gastronomía y buen servicio al cliente, será el encargado de dirigirse a los hoteles, restaurantes, casino a ofrecer nuestros productos alimenticios (cárnicos) explicar al cliente (jefe de cocina, chefs, gerentes) todos los beneficios de consumir nuestros productos, cual es la mejor manera de preparar nuestros productos. Debe tener un interés por el área comercial, responsable, con buena presentación personal.

Fecha: 26/09/2018
Salario: \$1300000
Contrato: Término Fijo
Horario: Lunes a Viernes 7:00 am a 5:00 pm
Contacto: Lizeth Ramirez Serma - 4484830
Correo: gh@agrosigo.com.ec

¡IMPORTANTE para aplicar a esta vacante debe enviar su hoja de vida con el ASUNTO: GASTRONOMÍA INTERNACIONAL.

Finaliza el: 08/07/2023 11:21

Ilustración 54 Imagen de oferta laboral 3

Estudiante #3: Bryan Gualoto

Publicación: Ilustración 54

Intereses: Bolsa de empleo

Coincidencia Talenter: 9%

Coincidencia Manual: 0%

Identificador: Baja coincidencia

AD Alberto Rusbel Duchi Bastidas
10/03/2023 12:00 Buena coincidencia

"¡Impulsa tu creatividad y únete a nuestro equipo tecnológico de vanguardia!"

Guayaquil
Talentos TI

Nos encontramos en la búsqueda de incorporar a nuestro equipo de Tecnología las siguientes posiciones:

- Administrador de Base de datos Jr.
- Ingeniero de Software.

Si te apasiona la tecnología, eres graduado de carreras a fines a Ing. en Computación, y tienes conocimiento y experiencia alineada al perfil, te invitamos a revisar la descripción de las vacantes.

¡Esta es tu oportunidad de ser parte de la familia de TIA S.A!

*Aplica hasta el 31 de mayo del 2023

Bolsa Guayaquil Computación Finaliza el: 12/04/2023 12:00

Estudiante #4: Jefferson Romero

Publicación: Ilustración 55

Intereses: Bolsa de empleo

Coincidencia Talenter: 50%

Coincidencia Manual: 60%

Identificador: #2

Ilustración 55 Imagen de oferta laboral 4

AD Alberto Rusbel Duchi Bastidas
31/12/1969 07:00 Buena coincidencia

"Únete a nosotros y lidera la próxima era de la tecnología computacional."

SAGGA
RECURSOS HUMANOS EN LÍNEA

¡NECESITAMOS DE TU EXPERIENCIA!

INGENIERO DE SOPORTE

Requisitos

- Título de Tercer nivel en Ingeniería en Sistemas.
- Experiencia mínima 11 o 2 años en manejo de herramientas de soporte a usuario y aseguramiento de calidad.
- Conocimientos en seguridad informática.
- Idiomas de programación.
- Mantenimiento y actualización de programas informáticos.

Funciones

- Dar soporte de primer nivel.
- Instalación de Software Especializado.
- Configuración de soluciones técnicas de los diferentes fabricantes.
- Manejo de troubleshooting de soluciones.

VACANTE LABORAL APLICA SOLO PARA QUITO, ECUADOR

Envía tu CV actualizada a
empleos@saggaac.com

Bolsa Quito Computación Finaliza el: 12/01/2023 12:00 1 año(s) experiencia

Estudiante #5: Bryan Chacón

Publicación: Ilustración 56

Intereses: Bolsa de empleo, pasantías

Coincidencia Talenter: 52%

Coincidencia Manual: 40%

Identificador: #2

Ilustración 56 Imagen de oferta laboral 5

Se puede concluir que mediante cada publicación el margen de error de talenter es muy bajo comparado a la coincidencia que realizaron los estudiantes manualmente por lo que su precisión es muy alta en cuanto al nivel de similitud entre las publicaciones con la hoja de vida de los estudiantes.

2.8.4. Nivel de reconocimiento en la carga de datos y automatización de la hoja de vida del estudiante.

	Bryan Gualoto	Bryan Chacón	Jefferson Romero	Paúl Morales	Michael Robles	
Información Personal	100%	85%	100%	100%	100%	97%
Foto de perfil	100%	100%	100%	100%	100%	100%
Idiomas	100%	100%	100%	100%	100%	100%
Estudios	95%	80%	86%	80%	85%	85.2%
Habilidades	95%	90%	100%	100%	80%	93%
Cursos	-	92%	95%	-	90%	92.33%
Proyectos	-	85%	80%	-	86%	83.67%
Experiencia laboral	90%	90%	97%	80%	90%	89.4%
Acerca de mí, visión, contactos	90%	90%	67%	85%	80%	82.4%
	95,71%	90.22%	97,67%	92.14%	90.11%	

Tabla 25. Tabla acerca del nivel de reconocimiento en la carga de datos

Conclusiones

El diseño y creación de un aplicativo web con un sistema de recomendación de prácticas preprofesionales y bolsa de trabajo para los estudiantes y docentes de carrera de computación de la Universidad Politécnica Salesiana ha demostrado tener alta precisión en cuanto al nivel de reconocimiento en la carga de datos y entre la comparación de similitudes entre una publicación realizada por el docente y la hoja de vida de un estudiante previamente cargado en el sistema.

Para continuar, con la aplicación de técnicas de procesamiento natural del lenguaje y aprendizaje automático se ha logrado proporcionar recomendaciones personalizadas mediante un valor cualitativo hacia el estudiante que se adapta a las preferencias con base al perfil creado de manera rápida y eficiente al momento de su registro en el aplicativo.

Por lo tanto, el sistema de recomendación ha creado un espacio en el que los docentes pueden agregar publicaciones de diferentes tipos de contenido, tanto textual como a través de imágenes. Esto permite a los estudiantes visualizar las publicaciones realizadas por los docentes y las empresas registradas en el sistema, proporcionando así un recurso de fácil acceso tanto para los estudiantes como para los docentes.

Asimismo, se estableció un sistema de autenticación seguro y eficiente que ha permitido a los diferentes usuarios, incluyendo estudiantes, docentes, empleadores y administradores, acceder a la plataforma de manera segura. Cada rol cuenta con permisos y funcionalidades específicas, garantizando la privacidad de la información y una correcta gestión de los datos debido a la utilización de herramientas de encriptación y controles de ruta de acceso para que terceras personas no puedan acceder a la información sin los privilegios o permisos respectivos.

Además, se diseñó una interfaz intuitiva y atractiva que ha facilitado la interacción de los estudiantes con las publicaciones. A través de la plataforma, los estudiantes pueden expresar su interés en las oportunidades, compartirlas, guardarlas o descargarlas a sus computadoras, brindando así una experiencia interactiva y práctica.

Para continuar, el sistema fue diseñado de manera que no discrimine a los estudiantes en base a su género, origen étnico u otras características. Con esto, se pudo establecer un enfoque equitativo y transparente en la generación de perfil del usuario.

Por otro lado, mediante las pruebas realizadas en el sistema, se ha logrado identificar que es altamente escalable para cualquier tipo de usuario y se limita hasta 500 usuarios con 20 peticiones de manera simultánea, que para la carrera de computación es suficiente en el término de delimitación de usuarios disponibles.

Para concluir, el aplicativo web desarrollado ha cumplido con los objetivos establecidos, brindando a los estudiantes de la Universidad Politécnica Salesiana de la carrera de ciencias de la computación un sistema de recomendación de prácticas preprofesionales en la bolsa de trabajo que se ajusta a sus perfiles y necesidades.

Recomendaciones

La integración de múltiples publicaciones del grupo de bolsa de trabajo ha enriquecido la calidad de las recomendaciones con base al currículo de cada estudiante. Sin embargo, se reconoce la importancia de mejorar continuamente la calidad de los datos y por ello se necesita tener más audiencia de prueba en la siguiente versión para implementar mecanismos de retroalimentación por parte de los usuarios para refinar y mejorar el sistema a lo largo del tiempo.

Con base a lo anterior, se recomienda explorar la implementación de algoritmos de aprendizaje automático más avanzados, como el filtrado colaborativo para mejorar aún más la precisión y personalización de las recomendaciones. Estas técnicas pueden proporcionar recomendaciones más relevantes y acertadas para los estudiantes.

Por ello, para garantizar el acceso a una amplia gama de oportunidades, es importante mantener una base de datos actualizada y diversa. Esto implica continuar estableciendo colaboraciones con empresas y organizaciones para agregar nuevas oportunidades y eliminar aquellas que ya no estén disponibles. Además, se puede considerar la inclusión de información adicional, como valoraciones de empresas y opiniones de estudiantes que hayan participado en prácticas preprofesionales anteriores.

Además, se podría considerar la incorporación de funcionalidades que faciliten la interacción entre estudiantes, docentes y empleadores. Por ejemplo, la posibilidad de enviar mensajes, programar reuniones o participar en foros de discusión relacionados con las prácticas preprofesionales. Esto fomentaría el networking y brindaría a los estudiantes la oportunidad de establecer contactos profesionales relevantes para la carrera.

REFERENCIAS

- Agichtein, E., Brill, E., & Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. *International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Aguilera, R. (22 de 02 de 2018). *Adictos al trabajo*. Obtenido de Tests de rendimiento con artillery: <https://www.adictosaltrabajo.com/2018/02/22/tests-de-rendimiento-con-artillery/>
- Amazon Web Services, Inc. (2023). *¿Qué es el procesamiento de lenguaje natural? - Explicación del procesamiento de lenguaje natural - AWS*. Obtenido de <https://aws.amazon.com/es/what-is/nlp/>
- Banco de Desarrollo de América Latina. (1 de Mayo de 2022). *CAF*. Obtenido de <https://www.caf.com/es/actualidad/noticias/2022/05/inteligencia-artificial-oportunidad-para-el-empleo-publico-en-america-latina/>
- Bolles, R. (2018). *What color is your parachute?*
- Cushicondor, H., & Edison, V. (2021). Desarrollo e implementación de un sistema para la gestión de una Bolsa de Empleos para Graduados de la Universidad Central del Ecuador. Universidad Central del Ecuador.
- DANE. (Mayo de 2022). *Principales indicadores del mercado laboral*. Obtenido de <https://www.dane.gov.co>
- FEUPS - Computación. (14 de 06 de 2023). *Encuesta acerca de la bolsa de trabajo UPS*. Quito.
- Hernández, S. (Julio de 2018). Sistema de recomendación de ofertas de empleo y cursos de formación mediante conocimiento de fuentes sociales y web. Madrid, España: E.T.S. de Ingenieros Informáticos (UPM).
- IBM. (2023). Protocolos de seguridad de cifrado: TLS.
- INEC. (2023). Encuesta Nacional de Empleo, Desempleo y Subempleo -ENEMDU.
- Jaser, Z., Petrakaki, D., Starr, R., & Oyarbide-Magaña, E. (28 de Enero de 2022). *Harvard Business Review*. Obtenido de <https://hbr.org/2022/01/where-automated-job-interviews-fall-short>
- Jiménez, P. (15 de Febrero de 2023). *Thinking for Innovation*. Obtenido de <https://www.iebschool.com/blog/business-intelligence-ventajas-digital-business/>
- Juan de Assembler Institute. (27 de Febrero de 2023). *¿Qué es un Framework en programación y sus principales usos?* Obtenido de <https://assemblerinstitute.com/blog/framework-programacion/>

- Khan Academy*. (2023). Obtenido de <https://es.khanacademy.org/computing/ap-computer-science-principles/data-analysis-101/x2d2f703b37b450a3:machine-learning-and-bias/a/machine-learning-algorithms>
- Kluemper, D., & Rosen, P. (2009). Future employment selection methods: evaluating social networking web sites. *Journal of Managerial Psychology*, 567-580.
- Lugo, N. (2022). *MODELO DE NEGOCIO PARA LA PUBLICACIÓN DE OFERTAS DE EMPLEO Y SELECCIÓN AUTOMATIZADA DE TALENTO HUMANO EN COLOMBIA, APOYADO CON EL DISEÑO DE UNA APLICACIÓN INFORMÁTICA QUE IMPLEMENTE ALGORITMOS DE RECOMENDACIÓN CON MACHINE LEARNING*. UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB.
- Ministerio de Trabajo Ecuador. (2021). Resumen Ejecutivo del Estudio de empleabilidad juvenil.
- MINTIC Colombia. (2021). *Estándares y Tecnologías*. Obtenido de <https://mintic.gov.co/portal/inicio/Atencion-y-Servicio-a-la-Ciudadania/Preguntas-frecuentes/5236:Estandares-y-Tecnologias>
- Netapp. (2019). *Netapp*. Obtenido de <https://www.netapp.com/es/artificial-intelligence/what-is-artificial-intelligence/>
- Ovalle, N., & Roso, A. (2022). Evaluación del Proceso de Reclutamiento y Selección Implementando Inteligencia Artificial. UNIVERSIDAD SANTO TOMÁS .
- Pathak, A. (25 de 04 de 2021). *Geekflare*. Obtenido de 5 plataformas fiables de alojamiento MEAN Stack: <https://geekflare.com/es/mean-hosting-platforms/>
- Powerdata. (2022). Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad. 1.
- Rivas, S. (2022). Aplicación de métodos NLU en la recomendación de CVs para la selección de personal. Universidad de Valladolid.
- Romero, S., & Tatiana, M. (2018). LA PRÁCTICA PRE PROFESIONAL: UN PRIMER ACERCAMIENTO AL CAMPO LABORAL. *eumed*.
- Rosado, M., & Espinoza, Á. (2015). MPORTANCIA DE LA INFORMÁTICA EN PRÁCTICAS PROFESIONALES O. *eumed.net*, 1.
- Rosas, D. (27 de Octubre de 2022). *Factor Trabajo*. Obtenido de <https://blogs.iadb.org/trabajo/es/inteligencia-artificial-clave-para-los-servicios-de-empleo/>
- Rubio, N. (6 de Noviembre de 2020). Obtenido de Redes neuronales profundas: qué son y cómo funcionan: <https://psicologiyamente.com/cultura/redes-neuronales-profundas>

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*.

Studocu. (2021). *Cuadro comparativo Metodologías Ágiles*. Obtenido de <https://www.studocu.com/es-mx/document/instituto-tecnologico-de-pachuca/ingenieria-de-software/cuadro-comparativo-metodologias-agiles/23713846>

Unir Vive. (29 de Diciembre de 2021). *¿Qué son las prácticas preprofesionales? ¿Cuál es la legislación al respecto?* Obtenido de <https://ecuador.unir.net/actualidad-unir/practicas-pre-profesionales/>

Universidad Politécnica Salesiana. (17 de Febrero de 2020). *UPS*. Obtenido de <https://www.ups.edu.ec/noticias?articleId=15900382>

Varguillas, C., Guffante, T., Manzano, M., & Moreno, P. (2020). Develaciones significantes de la práctica pre profesional desde la vivencia estudiantil. *revistaespacios*.

Verity. (29 de 07 de 2022). *Servicios y soluciones innovadoras en el ámbito de las tecnologías de información*. Obtenido de La ISO/IEC 9126: 2001: Características de la calidad de software: <https://www.verity.cl/que-es-norma-iso-iec-9126-2001/>