UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

Tema: "Algoritmos de machine learning para la correlación de proyectos de investigación"

Trabajo de Integración Curricular previo a la obtención del título de Ingeniero en Ciencias de la Computación

AUTOR: Cisneros Carlosama Luis David

TUTOR: Ing. Lascano Rivera Samuel Benjamín, MSc.

Tulcán, 2024.

CERTIFICADO DEL TUTOR

Certifico que el estudiante Cisneros Carlosama Luis David con el número de cédula 0401918057 respectivamente ha desarrollado el Trabajo de Integración Curricular: "Algoritmos de machine learning para la correlación de proyectos de investigación"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva.

Ing. Lascano Rivera Samuel Benjamín, MSc.

TUTOR

Tulcán, julio de 2024

AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Cisneros Carlosama Luis David con cédula de identidad número 0401918057 respectivamente declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Cisneros Carlosama Luis David

AUTOR

Tulcán, julio de 2024

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo Cisneros Carlosama Luis David declaro ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Algoritmos de machine learning para la correlación de proyectos de investigación" y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.

Cisneros Carlosama Luis David

AUTOR

Tulcán, julio de 2024

AGRADECIMIENTO

En primer lugar, deseo expresar mi gratitud a Dios y a mi madre por otorgarme la vida. También quiero reconocer el invaluable apoyo de mis hermanos y mis abuelitos. Agradezco profundamente a mi familia y amigos en general, quienes de diversas maneras me han animado a seguir adelante con mis estudios.

Agradezco especialmente a mis profesores por impartirme los conocimientos fundamentales que estoy seguro serán de gran ayuda en mi desarrollo profesional. Quisiera destacar la dedicación y paciencia de mi tutor, MSc. Samuel Lascano, cuyo apoyo incondicional y valiosos consejos me han permitido alcanzar la meta que tanto anhelaba.

Por último, quiero expresar mi reconocimiento a nuestra alma mater, la Universidad Politécnica Estatal del Carchi, y a la Carrera de Computación por brindarme la oportunidad de formar parte de una institución tan prestigiosa. Agradezco sinceramente por proporcionarme todas las herramientas necesarias para convertirme en un profesional de excelencia.

DEDICATORIA

Quiero dedicar este trabajo a mi madre, quien ha sido el pilar fundamental en mi vida. Su amor infinito y su constante apoyo han sido la fuerza que me ha impulsado a seguir adelante y alcanzar mis sueños. Agradezco profundamente por darme la oportunidad de continuar con mis estudios y por enseñarme el valor de la lucha y la perseverancia en cada situación que enfrento.

A mis hermanos y abuelitos, les dedico este trabajo como una muestra de gratitud por ser una fuente constante de inspiración y motivación en mi vida. Sus ánimos, consejos y palabras de confianza siempre permanecerán en mi mente y corazón. Aprecio profundamente el ejemplo de dedicación y esfuerzo que me han brindado, así como su incondicional apoyo en cada paso de mi camino.

A mis amigos de la universidad, les agradezco por compartir conmigo momentos inolvidables y por brindarme su sincera amistad y confianza. Su compañía y apoyo han sido fundamentales en mi trayectoria académica y personal, y siempre llevaré conmigo los gratos recuerdos que hemos compartido.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

Este proyecto de investigación presenta una respuesta a las demandas específicas de los estudiantes con respecto a los trabajos de integración curricular, el mismo que tributa a la investigación del SMART DATA LAB UPEC, en un entorno donde la excelencia en el análisis y la comprensión de la literatura académica es crucial. La singularidad de la propuesta radica en la experiencia de usuario que esta ofrece. En lugar de someter a los investigadores a largas y monótonas listas de resultados, la aplicación web brinda una interfaz visualmente atractiva y dinámica. Para la extracción de datos se utilizó web scraping o raspado web que consiste en analizar el código HTML de una página, en este caso a repositorios universitarios entre ellos la Universidad Politécnica Estatal del Carchi, la Universidad del Pacifico, Universidad de Azuay, la Universidad Internacional del Ecuador, Universidad Nacional de Loja y la Universidad Espíritu Santo, de lo cual se extrajo datos relevantes, tales como la fecha, tema de investigación, los autores y enlace de referencia, este proceso se llevó a cabo mediante el desarrollo de scripts personalizados utilizando el lenguaje de programación Python. También se aplicó en procesamiento del lenguaje natural (PNL), ya que es crucial para la extracción, limpieza y trasformación de texto no estructurado en datos estructurados que pueden ser utilizados por algoritmos machine learning. Para la aplicación de algoritmos machine learning se incluyeron algoritmos tales como el algoritmo k-Means para la agrupación de datos, así como la Logistic Regression para la clasificación de información. En última instancia se aplica el concepto de un coeficiente de correlación en el contexto de relaciones numéricas entre las variables, donde se busca medir la fuerza y dirección de la relación entre ellas.

Palabras Claves: SMART DATA LAB UPEC, web scraping, machine learning, coeficiente de correlación.

ABSTRACT

This research project presents a response to the specific demands of students in relation to curricular integration work, thus contributing to the research of the SMART DATA LAB UPEC. This effort is framed in an environment where excellence in the analysis and understanding of academic literature is crucial. The uniqueness of the proposal lies in the user experience it offers. Instead of subjecting researchers to long, monotonous lists of results, the web application provides a visually appealing and dynamic interface. For data extraction, web scraping was used, which consists of analysing the HTML code of a page. In this case, it was applied to university repositories such as the State Polytechnic University of Carchi, the University of the Pacific, the University of Azuay, the International University of Ecuador, the National University of Loja and the University of Espiritu Santo. From these repositories, relevant data such as date, research topic, authors, and reference link were extracted. This process was carried out by developing custom scripts using the Python programming language. In addition, natural language processing (NLP) techniques were applied, crucial for the extraction, cleaning and transformation of unstructured text into structured data that can be used by machine learning algorithms. Algorithms used included k-Means for data grouping and Logistic Regression for information classification. Finally, the concept of correlation coefficient was applied in the context of numerical relationships between variables, seeking to measure the strength and direction of the relationship between them.

Keywords: SMART DATA LAB UPEC, web scraping, machine learning, correlation coefficient.

INTRODUCCIÓN

Vivimos inmersos en una era digital donde la presencia abrumadora de la tecnología es palpable en cada aspecto de nuestra vida. En este escenario, la informática y la tecnología de la información ocupan un lugar central, siendo motores impulsores de la innovación y el progreso en nuestra sociedad contemporánea. Estas disciplinas no solo se limitan al estudio teórico y práctico de la información, sino que abarcan la creación de algoritmos, el diseño meticuloso de hardware y software, así como la aplicación de sistemas computacionales para abordar una amplia gama de problemas que enfrentamos en el día a día. Desde la automatización de procesos hasta la resolución de desafíos complejos en campos tan diversos como la medicina, la economía o la ingeniería, la informática y la tecnología de la información se han convertido en cimientos sólidos sobre los cuales se erige la estructura de nuestra sociedad moderna.

En medio de los avances tecnológicos continuos, existe la necesidad de proveer una herramienta que simplifique la búsqueda de proyectos de investigación de manera más fácil. Esto a su vez, podría estimular la generación de nuevos temas de investigación entre estudiantes y docentes universitarios.

Este proyecto de investigación responde de manera precisa a las necesidades particulares de los estudiantes en lo que respecta a los trabajos de integración curricular. Además, contribuye al avance de la investigación llevada a cabo por el SMART DATA LAB de la UPEC, y tiene como objetivo el desarrollo una herramienta informática que permita la búsqueda y clasificación de proyectos de investigación, bajo este concepto se creó una aplicación web desarrollada en el framework Flask y en el lenguaje de Python, apoyada con la metodología CRISP-DM, facilitando la entrega de un producto de calidad. Proponiendo el uso de la aplicación para los estudiantes, docentes de la Universidad Politécnica Estatal del Carchi.

I. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

A Nivel mundial la mayoría de las veces los datos disponibles sobre proyectos de investigación provienen de fuentes diversas y no están estandarizados, lo que dificulta su procesamiento y análisis, esta calidad de datos puede afectar a la precisión y confiabilidad de los resultados obtenidos mediante algoritmos de machine learning, esto puede tener un impacto negativo en la identificación de proyectos relacionados y en la toma de decisiones basadas en los resultados de la correlación (CAF, 2022).

Uno de los limitantes del desarrollo científico y tecnológico del Ecuador es la deficiencia de una coordinación efectiva entre los proyectos de investigación a nivel nacional. Esto se debe en parte a la dificultad de identificar patrones y relaciones entre los diferentes proyectos, lo que a su vez dificulta la planificación y gestión de futuras investigaciones. (Consejo de Educación Superior , 2024).

En el contexto de la investigación científica en el Ecuador, la correlación de proyectos de investigación es una tarea clave en el proceso de identificación de oportunidades de investigación y en la asignación de recursos. Sin embargo, este proceso puede ser difícil y costoso de realizar manualmente debido a la gran cantidad de datos y la complejidad de los mismos.

Además, existen dos problemas importantes que afectan la capacidad del Ecuador para implementar algoritmos de machine learning para la correlación de proyectos de investigación. En primer lugar, la carencia de talento humano calificado en el área de la ciencia de datos y el machine learning puede limitar la capacidad del país para desarrollar y aplicar estas tecnologías de manera útil. En segundo lugar, el presupuesto escaso para pagar por este tipo de funciones puede dificultar la inversión en tecnologías avanzadas de machine learning y reducir la capacidad del Ecuador para competir a nivel internacional en la investigación científica (Consejo de Educación Superior , 2024).

La ausencia de participación entre las instituciones académicas y los organismos de financiamiento puede disminuir la capacidad del Ecuador para identificar oportunidades de investigación y asignar recursos de manera eficiente. Además, la reducida cantidad y calidad de proyectos de investigación que se realizan afectan a los desafíos que se beben tomar en cuenta para un progreso en el ámbito tecnológico y científico, a esto se le aumenta la limitación del tiempo, debido a la dificultad para anteponer tareas esto significa que al tener múltiples tareas y proyectos en curso puede obstaculizar la priorización y dedicación de tiempo para la correlación de los proyectos de investigación.

En la Universidad Politécnica Estatal del Carchi se desarrollan diversos proyectos de investigación. Sin embargo, cuando son evaluados por otros investigadores, estos proyectos suelen quedar aislados de otras investigaciones que se llevan a cabo simultáneamente o que se realizaron. Este problema no es exclusivo de nuestra universidad, sino que es un inconveniente a nivel nacional. En los portales universitarios, por ejemplo, existen repositorios de proyectos de investigación y tesis que proporcionan información, pero que no establecen relaciones entre los distintos trabajos. Esta situación ha sido identificada como un problema que requiere atención en el planteamiento del estudio.

1.2. FORMULACIÓN DEL PROBLEMA

¿Qué algoritmos de machine learning son efectivos para identificar la correlación entre proyectos de investigación y facilitar la clasificación de los mismos?

1.3. JUSTIFICACIÓN

La clasificación automática de proyectos de investigación es un aspecto crítico en el análisis de información científica. Actualmente, la cantidad de proyectos de investigación generados a nivel mundial es enorme y la tarea de identificar y seleccionar los proyectos más relevantes y significativos para un área específica de estudio puede resultar abrumadora para los investigadores. La clasificación automática puede simplificar esta tarea permitiendo categorizar y agrupar proyectos de investigación en diferentes categorías.

La investigación sobre algoritmos de machine learning para la correlación de proyectos de investigación es altamente conveniente debido a su capacidad para mejorar la eficiencia y la precisión de la correlación de los mismo. El uso de algoritmos

de machine learning puede reducir significativamente el tiempo y el esfuerzo requeridos para correlación de tales proyectos, lo que resulta en un proceso más rápido y más acerado en la identificación de proyectos relevantes y relacionados.

Además, esta investigación tiene una relevancia social significativa, ya que los resultados pueden beneficiar a una amplia gama de actores sociales, incluyendo a la comunidad académica, a los responsables de la toma de decisiones, a los financiadores y a los miembros de la universidad, lo que puede tener implicaciones prácticas importantes para resolver problemas reales y potenciar la vida de los investigadores.

Los principales beneficiarios de esta investigación son los estudiantes de la UPEC, ya que les brinda la posibilidad de mejorar la capacidad de investigar e identificar proyectos significativos en diversas áreas y acelerar el progreso de la investigación que estén realizando.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Aplicar algoritmos de Machine Learning a través de una interfaz gráfica para la correlación automática en proyectos de investigación.

1.4.2. Objetivos Específicos

- Fundamentar teóricamente las variables de estudio para el desarrollo de la investigación.
- Comparar la precisión de algoritmos machine learning con técnicas tradicionales de clasificación para determinar su aplicación.
- Evaluar la precisión de diferentes algoritmos machine learning para la correlación entre proyectos de investigación.
- Proponer un algoritmo machine learning para la correlación entre proyectos de investigación.

1.4.3. Preguntas de Investigación

- ¿Cómo contribuiría la revisión bibliográfica en la ejecución de la aplicación?
- ¿Qué algoritmos machine learning se puede aplicar para identificar patrones significativos en los datos de proyectos de investigación?

- ¿Cuáles son los factores clave que influyen en la precisión de algoritmos machine learning?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

(Valero Cajahuanca, Navarro Raymundo, Larios Franco, & Julca Flores, 2022), en su trabajo: "Deserción universitaria: Evaluación de diferentes algoritmos de Machine Learning para su predicción", realizo el algoritmo K-Nearest-Neightbor este modelo resultante ayuda a predecir qué estudiantes tienen más probabilidades de abandonar los estudios en el primer ciclo y ayuda a alertar al Departamento de Servicios Sociales sobre la necesidad y atención de clases individuales y grupales. Los resultados muestran que el algoritmo K-Nearest-Neighbor tiene un rendimiento superior en la predicción de la deserción universitaria con una precisión de 0.91, utilizando las variables académicas y socioeconómicas de los estudiantes.

(Rojas Gonzáles & Vargas Gonzáles, 2022), su trabajo se centra en la comparación y evaluación de varios algoritmos de machine learning para predecir la maliciosidad de URLs. Utilizando un dataset extenso de alrededor de 50 mil URLs clasificadas, se implementan y evalúan diferentes pipelines de procesamiento en la plataforma colaborativa Google Colab. Los algoritmos evaluados incluyen Decision Tree, Random Forest, Support Vector Machine y redes neuronales, así como un ensamblado de estos. El objetivo principal es identificar el algoritmo que proporciona la mejor precisión en la detección de URLs maliciosas. Esta evaluación exhaustiva no solo contribuye al conocimiento existente en el campo, sino que también ofrece una guía práctica para la aplicación de machine learning en la seguridad cibernética. La evaluación de estos algoritmos de machine learning mostró que el ensamble de algoritmos obtuvo la mejor precisión general (89%), aunque se pueden mejorar sus parámetros. Decision Tree fue el más equilibrado y se eligió para la prueba de concepto. La red neuronal tiene potencial para generalizar sin datos etiquetados. Support Vector Machine fue menos efectivo para categorías maliciosas. En general, los algoritmos evaluados, excepto Support Vector Machine, son prometedores para la predicción de URLs maliciosas y tienen aplicaciones en entornos profesionales.

(Morales et al., 2022), en este estudio se desarrollaron dos clasificadores de aprendizaje automático, una red neuronal multicapa (perceptrón multicapa [MLP]) y un modelo de potenciación del gradiente (GB), con el objetivo de predecir el grado de logro académico en las asignaturas de español y matemáticas para estudiantes de sexto grado de primaria (2008) y tercero de secundaria (2011) en el estado de Tlaxcala, México. Se utilizaron 13 variables contextuales obtenidas de los Exámenes Nacionales del Logro Académico en Centros Escolares (Enlace) para entrenar y probar los clasificadores con un conjunto de datos de 11,036 registros de estudiantes que permanecieron en el sistema escolar de 2008 a 2011. El algoritmo bosque aleatorio (RF) se utilizó para determinar la importancia relativa de las variables de entrada, y se observó que los puntajes en español y matemáticas tuvieron una mayor importancia relativa en comparación con otros factores contextuales analizados, como el sexo, beca y turno de la escuela. En cuanto a los resultados, se encontró que el clasificador MLP fue superior a GB en español con una precisión global de clasificación (PG) del 70.1% en 2008 y del 61.1% en 2011. En matemáticas, GB obtuvo mejores resultados con una PG del 68.8% en 2008 y del 63.5% en 2011. Además, se observó que el puntaje en español tenía una fuerte asociación con el grado de logro académico en matemáticas. En la población de estudiantes analizada, se encontró que en español y matemáticas la proporción de mujeres era mayor que la proporción de hombres en los grados de logro académico elemental, bueno o excelente. En contraste, en ambas asignaturas esta proporción se invierte con el grado de logro insuficiente.

(Rico Páez & Gaytán Ramírez, 2022) El propósito de este estudio es proponer una metodología para crear modelos predictivos del rendimiento académico de los estudiantes de ingeniería en México, utilizando sus características, y comparar los modelos utilizando diferentes métricas de evaluación. La muestra de este estudio incluyó 228 estudiantes de una universidad pública, y se recopilaron los datos al comienzo del curso. Se utilizaron tres técnicas de aprendizaje automático para crear los modelos predictivos, y se analizaron las características de cada modelo. La precisión de las predicciones fue de alrededor del 65%. El modelo construido con la técnica Naive Bayes fue el más adecuado para la mayoría de las métricas empleadas en el estudio, particularmente para identificar a los estudiantes en riesgo de reprobar. Además, se descubrió que la característica más significativa para predecir el rendimiento académico fue el promedio actual del estudiante. La

metodología desarrollada es replicable para otros cursos, y las características de los estudiantes pueden recopilarse antes del comienzo del curso, lo que permite realizar intervenciones estratégicas para los estudiantes en riesgo de reprobar.

En 2020, un equipo de investigadores de la Universidad Nacional de Ingeniería en Perú desarrolló un algoritmo de aprendizaje automático para analizar más de 20,000 artículos científicos publicados por investigadores peruanos. El objetivo era obtener una mejor comprensión de las principales áreas de investigación en el país, así como identificar patrones de colaboración entre investigadores. El algoritmo de aprendizaje automático utilizado en el estudio se basó en técnicas de procesamiento de lenguaje natural y aprendizaje no supervisado. El modelo analizó los títulos, resúmenes y palabras clave de los artículos científicos para identificar las palabras y frases más comunes utilizadas por los investigadores en sus trabajos. El estudio descubrió que las áreas de investigación más activas en Perú son la biología y la medicina, seguidas de cerca por la ingeniería y la física. También se identificaron patrones de colaboración entre investigadores, lo que permitió a los responsables de la política científica identificar áreas prioritarias para la financiación y la promoción de la investigación.

(Hernández, 2020) Esta investigación compara varios modelos de aprendizaje automático para predecir la velocidad de alimentación en el proceso de aserrado en la industria forestal. La predicción precisa de este valor es crucial para mejorar la productividad de las empresas en este sector. Los avances en tecnología digital permiten la recopilación de datos de maquinarias clave para procesos transversales en estas empresas, lo que se combina con técnicas de aprendizaje automático para hacer recomendaciones operativas que garanticen el cumplimiento de los objetivos. La metodología CRIS-DM, ampliamente utilizada en la industria, fue empleada en este proyecto de título, que se enfoca en la minería de datos y análisis. Esta metodología proporciona una estructura sólida y probada para llevar a cabo proyectos analíticos, desde la identificación de los requisitos comerciales hasta la implementación de la solución. En la ciencia de datos, la minería de datos y el aprendizaje automático, se llevan a cabo diversos procesos para extraer información y conocimientos de los datos.

(Cabrera Palacios & Chiluiza Luna , 2018) El objetivo principal de esta investigación es desarrollar un algoritmo de aprendizaje automático para la detección temprana de fallas en vehículos M1 con el fin de mejorar el mantenimiento automotriz. Se utilizó un algoritmo de aprendizaje no supervisado K-means para encontrar agrupaciones en los datos obtenidos a partir de diferentes variables. Esto permitió utilizar herramientas de aprendizaje y clasificación para obtener una mayor precisión y ahorrar tiempo en la detección de fallas. Los resultados del estudio mostraron que el clasificador SVM gaussiano y cubico tuvo una precisión del 99.6%, lo que valida su capacidad para distinguir las categorías de kilometraje actual y kilometraje del último mantenimiento en los vehículos.

2.2. MARCO TEÓRICO

2.2.1. Ciencias de la computación

Es un campo amplio que abarca muchos aspectos de la informática y la tecnología de la información y desempeña un papel central en nuestra sociedad moderna, ya que sustenta gran parte de la tecnología y la innovación del mundo.

La informática es una disciplina que estudia la teoría, el análisis y la práctica de la información, la informática y los sistemas informáticos. Se trata de crear algoritmos, diseñar hardware y software y aplicar computadoras para resolver problemas prácticos que se incitan en la actualidad. (ANEP, 2023, pág. 9)

La ciencia de la computación es una disciplina en constante evolución en la que continuamente se realizan nuevos avances. Los trabajos de investigación son una forma importante de compartir estos progresos y aumentar la tecnología de la información. Aquí algunas definiciones de áreas que componen a las ciencias de la computación.

- Teoría de la Computación: explora los fundamentos matemáticos de la computación, como la complejidad computacional, la teoría de la automatización y la teoría de la información.
- Arquitectura de computadoras: estudia el diseño y construcción del hardware de las computadoras como procesadores, memoria y dispositivos de almacenamiento.
- Sistemas operativos: controlan el funcionamiento de las computadoras y brindan servicios a programas de aplicación.

- Lenguajes de programación: permiten a los programadores escribir instrucciones que las computadoras puedan entender y ejecutar.
- Bases de datos: almacene y recupere grandes cantidades de datos de manera eficiente.
- Inteligencia artificial: Desarrollar sistemas que puedan aprender y razonar como los humanos.
- Gráficos por computadora: cree imágenes y animaciones realistas. Robótica: cree robots que puedan realizar tareas de forma independiente.

2.2.2. Machine learning

(DataScientest, 2021) dice el aprendizaje automático o machine learning es un campo de la ciencia, específicamente una subcategoría de la inteligencia artificial. Incluye habilitar algoritmos para detectar "patrones", es decir, patrones repetidos, en conjuntos de datos. Estos datos pueden ser números, palabras, imágenes, estadísticas, etc.

Los algoritmos de aprendizaje automático aprenden por sí solos a realizar una tarea o hacer predicciones basadas en datos y mejoran su rendimiento con el tiempo. Una vez entrenado, el algoritmo debería poder encontrar patrones en los nuevos datos.

(RECFACES, 2021) dice, el reconocimiento de patrones es un proceso que implica el uso de algoritmos informáticos para clasificar los datos de entrada en objetos, clases o categorías en función de sus características destacadas o invariantes. Un algoritmo siempre define pasos claros para lograr un objetivo, pero un diagrama es una descripción de alto nivel de la solución.

Tabla 1. Comparación entre algoritmos machine learning

Algoritmo	Tipo	Aplicación	Ventajas	Desventajas
Regresión Lineal	Supervisado	Predicción de valores continuos	Simple, interpretable, rápido	Sensible a outliers, asume linealidad
Regresión Logística	Supervisado	Clasificación binaria	Fácil interpretación, probabilidades claras	No adecuado para relaciones no lineales
Árboles de Decisión	Supervisado	Clasificación y regresión	interpretación intuitiva, manejo automático de NA	Propenso al sobreajuste, inestable
Bosques Aleatorios	Supervisado	Clasificación y regresión	Reducción del sobreajuste, manejo de datos faltantes	Menos interpretable que los árboles de decisión

Máquinas de Vectores de Soporte (SVM)	Supervisado	Clasificación y regresión	Eficaz en espacios de alta dimensión, versátil	Ineficiente para conjuntos de datos muy grandes
k-Vecinos Más Cercanos (k-NN)	Supervisado	Clasificación y regresión	Fácil implementación, adaptabilidad	Sensible a la elección de k y a la dimensionalidad
Redes Neuronales Artificiales (ANN)	Supervisado	Clasificación y regresión	Capacidad para capturar relaciones complejas	Requiere grandes conjuntos de datos, lento entrenamiento
Naive Bayes	Supervisado	Clasificación	Eficiente con datos de alta dimensionalidad	Asunción de independencia entre variables
Clustering K-Means	No supervisado	Agrupamiento	Eficiente en grandes conjuntos de datos	Sensible a la elección del número de clusters
Clustering Aglomerativo	No supervisado	Agrupamiento	Capaz de manejar diferentes formas de clusters	Ineficiente para grandes conjuntos de datos
Clustering DBSCAN	No supervisado	Detección de outliers	Identifica clusters de formas arbitrarias	Sensible a la elección de parámetros
Análisis de Componentes Principales (PCA)	No supervisado	Reducción de dimensionalidad	Conserva la varianza, útil para visualización	Interpretación no siempre clara, información perdida
Análisis de Componentes Independientes (ICA)	No supervisado	Separación de fuentes	Útil para separar señales mezcladas	No garantiza la identificación de fuentes reales
Algoritmo Genético	No supervisado	Optimización	Puede encontrar soluciones óptimas en problemas complejos	Tiempo de computación alto, no garantiza la optimalidad
Support Vector Clustering (SVC)	No supervisado	Agrupamiento	Manejo de datos no lineales, eficiente	Sensible a la elección de hiperparámetros
Algoritmo de Asociación (Apriori)	No supervisado	Análisis de asociación	Identifica patrones de co-ocurrencia	Sensible a la elección de umbral y tamaño del itemset
Gradient Boosting Machines (GBM)	Supervisado	Clasificación y regresión	Eficaz en una amplia variedad de datos	Sensible a sobreajuste, tiempo de entrenamiento largo
Redes Neuronales Convolucionales (CNN)	Supervisado	Clasificación	Buen rendimiento en datos con estructura espacial	Requiere grandes conjuntos de datos, computacionalmente intensivo
Redes Neuronales Recurrentes (RNN)	Supervisado	Series temporales, procesamiento de lenguaje natural	Modelado de secuencias, manejo de datos secuenciales	Problemas con gradientes que desaparecen/explodes
Máquinas de Aprendizaje Extremo (ELM)	Supervisado	Clasificación y regresión	Rápido entrenamiento, buena generalización	Sensible a la elección de hiperparámetros

Fuente: IBM, (2024). ¿Qué es un algoritmo de machine learning?

Grupos de algoritmos de aprendizaje automático o machine learning (ML).

Aprendizaje supervisado

El machine learning supervisado, la maquina enseña dado un ejemplo. El operador alimenta al algoritmo de aprendizaje automático con un conjunto de datos que contiene las entradas y salidas deseadas. El algoritmo debe encontrar un método para determinar cómo usar las entradas y salidas para generar resultados precisos. Mientras que el operador conoce la respuesta correcta al problema, el algoritmo detectará patrones en los datos, aprenderá de las observaciones y hará predicciones. El algoritmo hace predicciones y el operador lo ajusta, un proceso que continúa hasta que el algoritmo alcanza un alto nivel de precisión y rendimiento (Redacción APD, 2019).

Aprendizaje no supervisado

El aprendizaje no supervisado es un tipo de aprendizaje automático en el que solo se tienen datos de entrada (X), sin variables de salida. El objetivo es encontrar la estructura o distribución subyacente en esos datos para aprender más sobre ellos.

Un ejemplo de la vida real sería clasificar monedas de diferentes colores en pilas separadas. Nadie te ha enseñado a hacerlo, pero solo observando sus características, como el color, puedes ver qué monedas están relacionadas y agruparlas.

Se llama aprendizaje no supervisado porque, a diferencia del aprendizaje supervisado, no hay respuestas correctas ni un "profesor". Los algoritmos se dejan "a su suerte" para descubrir y mostrar la estructura interesante en los datos.

Además, consiste en entrenar modelos con datos sin procesar y sin etiquetar. Como sugiere el nombre, el aprendizaje automático no supervisado no requiere tanta intervención humana como el aprendizaje supervisado. Los humanos pueden establecer parámetros del modelo, como la cantidad de puntos de clúster, pero el modelo es capaz de manejar grandes conjuntos de datos de manera eficiente sin intervención humana (Universidad Europea, 2022).

El aprendizaje automático no supervisado se utiliza principalmente para:

- Grupos de datos con similitud entre datos de características o segmentos

- Comprenda las relaciones entre diferentes puntos de datos, como las recomendaciones automáticas de música.
- Realizar análisis de datos iniciales

Aprendizaje por refuerzo

El aprendizaje por refuerzo es una forma de aprendizaje automático menos común pero más compleja que genera resultados sorprendentes. A diferencia del aprendizaje supervisado, no utiliza etiquetas, sino que aprende a través de recompensas. Es similar al aprendizaje por refuerzo en psicología, donde se utilizan comentarios positivos y negativos para moldear el comportamiento.

Imagina entrenar a un perro: los buenos comportamientos se recompensan con una golosina y se vuelven más frecuentes, mientras que los malos comportamientos se castigan y disminuyen. Este comportamiento motivado por la recompensa es fundamental en el aprendizaje por refuerzo.

En cierto modo, el aprendizaje por refuerzo se puede ver como "aprender de los errores". Si colocamos un algoritmo de aprendizaje por refuerzo en un entorno, al principio cometerá muchos errores. Pero si le proporcionamos señales que asocien los buenos comportamientos con una señal positiva y los malos comportamientos con una negativa, podemos reforzar el algoritmo para que prefiera los buenos comportamientos. Con el tiempo, el algoritmo aprende a cometer menos errores (MERINO, 2019).

2.2.3. Algoritmos supervisados con Machine Learning (ML)

El aprendizaje automático, en su mayoría, se basa en el aprendizaje supervisado.

Este tipo de aprendizaje funciona así: se le proporcionan al algoritmo variables de entrada (x) y una variable de salida (Y), y este aprende la función de mapeo (las reglas) que relaciona la entrada con la salida. La ecuación sería:

El objetivo es que el algoritmo aprenda la función de mapeo tan bien que, al darle nuevos datos de entrada (x), pueda predecir la variable de salida (Y) para esos datos. Se llama aprendizaje supervisado porque es como si un profesor estuviera supervisando el proceso de aprendizaje del algoritmo. El profesor conoce las respuestas correctas, así que el algoritmo va haciendo predicciones con los datos que tiene, y el profesor las corrige. El aprendizaje termina cuando el algoritmo consigue obtener resultados aceptables.

2.2.3.1. Algoritmos de clasificación

(Heras, 2020) dice que predicen etiquetas o clases con prioridades conocidas. El resultado deseado es una etiqueta separada o especial, en caso de que el modelo de entrenamiento esté entre 2 clases, se define como una clase binaria. Si tenemos que predecir más de 2 clases, se llama clasificación multiclase.

El algoritmo de clasificación se utiliza cuando el resultado es un número infinito de resultados.

El ejemplo más utilizado para entender los algoritmos de clasificación es el spam o detectores de spam. Si queremos saber si un correo es spam o no, el algoritmo de clasificación decidirá qué tipo de correo es. Este método también se conoce como clasificación binaria.

Hay varios métodos de aprendizaje automático que podemos usar en problemas de clasificación. Podemos destacar:

- regresión logística (logistic regression)
- máquinas de vectores de soporte (support vector machines)
- árboles de decisión (decision trees)
- bosques aleatorios (random forests)
- redes neuronales y aprendizaje profundo (deep learning)

2.2.3.2. Algoritmo de regresión lineal

(DATLAS, 2020)para problemas de regresión, un programa de aprendizaje automático necesita evaluar y comprender las relaciones entre las variables. El análisis de regresión se centra en una variable dependiente y varias otras variables, lo que lo hace particularmente útil para realizar pronósticos y predicciones.

Hay varios métodos de aprendizaje automático que podemos usar en problemas de regresión lineal. Podemos destacar:

- regresión lineal y regresión no lineal
- máquinas de vectores de soporte (support vector machines)
- árboles de decisión (decision trees)
- bosques aleatorios (random forests)

- redes neuronales y aprendizaje profundo (deep learning)

2.2.3.3. Arboles de decisión

Los árboles de decisión son herramientas de análisis de datos y machine learning que utilizan algoritmos estadísticos para crear modelos predictivos basados en la clasificación de datos según ciertas características o en la relación entre variables para predecir valores. Estos modelos son utilizados en la analítica de Big Data para hacer predicciones y tomar decisiones informadas (UNIR, 2021).

La estructura del árbol de decisión se compone de ramas y nodos que tienen diferentes funciones.

- Los nodos internos del árbol representan características o propiedades que son importantes para tomar una decisión.
- Por otro lado, las ramas del árbol representan decisiones que se toman en función de condiciones específicas, como la probabilidad de que ocurra algo.
- Finalmente, los nodos finales representan el resultado de la decisión tomada en función de las características y condiciones consideradas.

2.2.3.4. Redes neuronales

Las redes neuronales simuladas (SNN) son un subconjunto del machine learning y se utilizan ampliamente en el deep learning. Están diseñadas para imitar la estructura y el funcionamiento del cerebro humano.

Si la salida de cualquier nodo individual supera el valor de umbral establecido, el nodo se activa y los datos se envían a la siguiente capa de la red. Si la salida es inferior al valor de umbral, los datos no se pasan a la siguiente capa de la red. En resumen, las redes neuronales artificiales imitan la estructura y el comportamiento del cerebro humano mediante la conexión de nodos con pesos y umbrales para procesar y transmitir información (IBM Cloud Education, 2020).

Existen diferentes tipos de redes neuronales, entre los cuales se incluyen las de alimentación directa, las recurrentes y las modulares. Cada tipo de red neuronal tiene sus propias características y se utiliza en diferentes situaciones según las necesidades de la tarea específica que se esté abordando.

La gran ventaja de las redes neuronales es que no requieren la intervención humana para aprender, ya que las capas anidadas dentro de ellas procesan los datos y sacan

conclusiones por sí mismas. Con el tiempo, esto les permite aprender a través de sus propios errores y mejorar su capacidad para hacer predicciones precisas (Tokio School, 2021).

2.2.3.5. Bosques aleatorios (Random Forest)

Random Forest es ampliamente reconocido y utilizado por científicos de datos como uno de los algoritmos más populares. Se trata de un algoritmo de aprendizaje automático supervisado que encuentra aplicación en diversos problemas de clasificación y regresión. La metodología consiste en construir múltiples árboles de decisión utilizando diferentes muestras y combinar sus resultados a través de un voto mayoritario en el caso de clasificación, y mediante el promedio en situaciones de regresión (ER, 2023).

Una característica destacada del algoritmo Random Forest es su capacidad para manejar conjuntos de datos que incluyen tanto variables continuas, como en el caso de la regresión, como variables categóricas, como en la clasificación. Esta versatilidad es una de las cualidades más importantes del algoritmo.

Funcionamiento del algoritmo Random Forest:

1. Embolsado: Este método implica crear diferentes subconjuntos de entrenamiento a partir de muestras de datos con reemplazo. El resultado final se obtiene mediante la combinación de las predicciones de cada subconjunto, utilizando un enfoque de votación por mayoría. Un ejemplo de este enfoque es el algoritmo de Bosque Aleatorio.
2. Impulso: En este método, se combinan aprendices débiles para formar un aprendiz fuerte mediante la creación secuencial de modelos. Cada modelo se enfoca en corregir los errores del modelo anterior, de modo que el modelo final tenga la mayor precisión posible.

El algoritmo de bosque aleatorio sigue los siguientes pasos:

- Paso 1: En el modelo de bosque aleatorio, se elige un subconjunto de puntos de datos y un subconjunto de características para construir cada árbol de decisión. Esto implica seleccionar aleatoriamente n registros y m características del conjunto de datos que originalmente contiene k registros.
- Paso 2: Se construyen árboles de decisión individuales para cada subconjunto de muestra.

- Paso 3: Cada árbol de decisión genera una salida o predicción independiente.
- Paso 4**:** El resultado final se determina mediante la votación mayoritaria en el caso de clasificación, o mediante el promedio en el caso de regresión. Para llegar a la predicción final, se considera la decisión o el valor predicho de cada árbol. El resultado más común o el promedio de los valores se selecciona para la predicción final.

El algoritmo de bosque aleatorio presenta las siguientes características:

1. Diversidad: Cada árbol individual en el bosque aleatorio se construye considerando un subconjunto diferente de atributos o características, lo que hace que cada árbol sea único y diverso.
2. Inmunidad a la maldición de la dimensionalidad: Debido a que cada árbol solo utiliza un subconjunto de características, el espacio de características se reduce, lo que ayuda a evitar los problemas asociados con conjuntos de datos de alta dimensionalidad.
3. Paralelización: La construcción de cada árbol en el bosque aleatorio se realiza de manera independiente, lo que permite aprovechar completamente los recursos de procesamiento paralelo, como la capacidad de la CPU, acelerando así el proceso de construcción del bosque.
4. División de prueba y entrenamiento**:** En el bosque aleatorio, no es necesario dividir los datos en conjuntos separados de entrenamiento y prueba. Esto se debe a que cada árbol se construye utilizando una muestra diferente de datos, por lo que siempre hay un 30% de los datos que no se utilizan en la construcción de un árbol específico, lo que brinda una evaluación interna del rendimiento.
5. Estabilidad: La estabilidad en el bosque aleatorio se logra mediante la combinación de múltiples árboles y la toma de decisiones basada en votación por mayoría o promedio. Esto reduce la sensibilidad a los cambios en los datos de entrada y aumenta la robustez del modelo.

Tabla 2. Diferencias entre árbol de decisión y bosque aleatorio

Árboles de decisión	Bosque aleatorio
1. Los árboles de decisión pueden sufrir de sobreajuste si se les permite crecer sin restricciones.	1. Los bosques aleatorios se construyen a partir de subconjuntos de datos, lo que evita el sobreajuste y el resultado final se basa en una clasificación promedio o mayoritaria.
2. Un solo árbol de decisión es más rápido en términos de cálculo.	2. El proceso de construcción del bosque aleatorio es comparativamente más lento debido a la construcción de múltiples árboles.

3. Un árbol de decisión formula reglas basadas en un conjunto de datos y características de entrada para hacer predicciones.	3. El bosque aleatorio selecciona observaciones al azar, construye múltiples árboles de decisión y toma el resultado promedio. No se basa en un conjunto específico de reglas.

Fuente: Sruthi ER. (2023). Understand Random Forest Algorithms With Examples (Updated 2023)

2.2.3.6. Máquinas de vectores de soporte (Support Vector Machines)

Los SVM es un modelo de aprendizaje automático que se utiliza para clasificar y predecir datos. Trabajan encontrando un hiperplano que maximiza la distancia entre puntos de datos de diferentes clases.

La línea se dibuja de modo que la distancia entre los datos de diferentes conjuntos sea lo más grande posible. Sin embargo, también se puede aplicar a desafíos de regresión (Ray, 2023).

Tabla 3. Características importantes del algoritmo Support Vector Machine

Característica	Descripción
Capacidad de manejar datos linealmente separables y no linealmente separables	El SVM puede encontrar un hiperplano de separación óptimo tanto en problemas de clasificación linealmente separables como en problemas de clasificación no linealmente separables mediante el uso de funciones de kernel.
Margen máximo	El SVM busca maximizar el margen entre las clases al encontrar el hiperplano de separación óptimo, lo que contribuye a una buena generalización y evita el sobreajuste.
Transformación de características	El SVM utiliza funciones de kernel para transformar los datos de entrada en un espacio de características de mayor dimensión donde las clases son más fácilmente separables.
Soporte de vectores	Los vectores de soporte son los puntos de datos más cercanos al hiperplano de separación y desempeñan un papel crítico en la definición del hiperplano óptimo y la toma de decisiones.
Regularización	El SVM cuenta con un parámetro de regularización que equilibra el margen máximo y la tolerancia a errores de clasificación, permitiendo ajustar el modelo para evitar el sobreajuste y garantizar un buen rendimiento en datos no vistos.
Sensibilidad a los datos de soporte	El SVM es menos sensible a los datos lejanos al hiperplano de separación y se enfoca en los vectores de soporte más relevantes para la clasificación, lo que lo hace más resistente a valores atípicos y ruido en los datos.
Eficiencia en espacios de alta dimensión	El SVM puede ser eficiente en espacios de características de alta dimensión, ya que solo requiere los vectores de soporte para definir el hiperplano óptimo, lo que reduce la complejidad computacional.
Interpretación de los resultados	En algunos casos, el SVM proporciona una interpretación clara de los límites de decisión y los factores que influyen en la clasificación, lo cual es útil para el análisis y la comprensión del modelo.
Versatilidad	El SVM puede aplicarse tanto a problemas de clasificación como de regresión, adaptándose a diferentes tipos de datos y problemas.

Fuente: Sunil Ray. (2023). Learn How to Use Support Vector Machines (SVM) for Data Science

2.2.3.7. The Gradient Boosting Algorithm

En el ámbito del aprendizaje automático, el impulso emerge como una técnica de conjunto de gran relevancia, diferenciándose de los modelos tradicionales que operan de manera independiente. A diferencia de estos últimos, el impulso aprovecha la sinergia entre diversos "alumnos débiles" para construir un "alumno fuerte" único y con mayor precisión (Tuychiev, 2023).

En esencia, el impulso funciona como un equipo de expertos: cada alumno débil aporta su perspectiva única al problema, y el impulso combina estas predicciones individuales para generar una predicción final más robusta y confiable. Esta técnica se asemeja a la sabiduría colectiva que surge de la colaboración entre diversos individuos con diferentes conocimientos y experiencias.

Fortalezas del Gradient Boosting:

- Mayor precisión: Al combinar las predicciones de múltiples alumnos, el impulso reduce el sesgo individual y aumenta la precisión general del modelo.
- Robustez frente a errores: Si un alumno débil comete un error, el impulso no se ve tan afectado gracias a las contribuciones de los demás alumnos.
- Flexibilidad: El impulso puede adaptarse a diversos tipos de modelos de aprendizaje automático, desde árboles de decisión hasta redes neuronales.
- Interpretabilidad: Al analizar las predicciones de cada alumno débil, se puede obtener una mejor comprensión del modelo y las decisiones que toma.

2.2.4. Algoritmos no supervisados con Machine Learning (ML)

2.2.4.1. Algoritmos de Clustering

El propósito de un algoritmo de clustering es clasificar los objetos de un conjunto de datos según su grado de similitud, de tal manera que los objetos dentro de un mismo grupo (o cluster) sean más parecidos entre sí que con aquellos que pertenecen a otros grupos (Caparrini, 2020).

En términos generales, este problema consiste en agrupar eficazmente un conjunto de datos sin etiquetar. Aunque parece intuitivo, la noción de "cluster/agrupamiento" no puede definirse de manera precisa, lo que ha llevado a la propuesta de una amplia variedad de algoritmos de clustering.

2.2.4.2. Algoritmo k-means

El método de agrupamiento k-means se utiliza para dividir un conjunto de datos en k grupos o clusters, de manera que los datos en un mismo cluster sean más parecidos entre sí que aquellos en clusters distintos.

Dentro de los algoritmos de aprendizaje no supervisado, K-means es posiblemente el más conocido. Este método fue desarrollado para abordar el aumento exponencial en la cantidad de datos generados, capturados, copiados y consumidos a nivel mundial, que actualmente se estima en alrededor de 100 Zettabytes y sigue en aumento. Con el algoritmo k-means, es posible reunir grandes cantidades de información similar en un solo lugar, lo que facilita la identificación de patrones y la realización de predicciones en grandes conjuntos de datos (RAMÍREZ, 2023).

Antes de utilizar el algoritmo K-means, es necesario establecer el número de clusters deseados (k). Por ejemplo, si se establece «k» como 2, el conjunto de datos se agrupará en 2 grupos, mientras que, si se establece «k» como 4, se agrupará el conjunto de datos en 4 grupos.

Cada grupo es representado por su centro o centroid, que se calcula como la media aritmética de los puntos de datos asignados al grupo. El algoritmo itera hasta que cada punto de datos está más cerca del centro de su grupo que de los centros de otros grupos. En cada paso, el algoritmo minimiza la distancia entre los puntos de datos y el centro de su grupo.

El algoritmo K-means funciona de la siguiente manera:

1. Especificar el número de clústers deseados (k): En primer lugar, se debe determinar cuántos clústeres se quieren crear en el conjunto de datos. Este número se denomina k.
2. El algoritmo k-means comienza seleccionando k puntos al azar del conjunto de datos. Estos puntos se denominan centroides iniciales. Los centroides son el punto central o el promedio de cada clúster.
3. A continuación, el algoritmo asigna cada punto del conjunto de datos al clúster cuyo centroide está más cerca. El punto se asigna al clúster cuyo centroide tiene la menor distancia.

4. Una vez que se han asignado todos los puntos a un clúster, los centroides de cada clúster se vuelven a calcular como la media de todos los puntos del clúster.
5. Repetir los pasos 3 y 4 hasta que los centroides del clúster ya no cambien o hasta que se alcance el número máximo de iteraciones: Los pasos 3 y 4 se repiten hasta que los centroides del clúster ya no cambien o hasta que se alcance el número máximo de iteraciones, lo que indica que se ha alcanzado una solución estable.

2.2.5. Aprendizaje por refuerzo

El aprendizaje por refuerzo, o Reinforcement Learning, es una rama del aprendizaje automático que se distingue por no necesitar grandes cantidades de datos de entrenamiento para funcionar. En su lugar, el sistema aprende a través de ensayo y error, guiado por una serie de indicaciones. A diferencia del aprendizaje supervisado, donde se proporciona un conjunto de datos que indica a la máquina qué hacer, en el aprendizaje por refuerzo se utilizan recompensas para reforzar el comportamiento deseado (García, 2020).

Este tipo de aprendizaje es especialmente útil para situaciones en las que se sabe el resultado deseado, pero no se sabe cómo alcanzarlo. Esto requiere muchas iteraciones y pruebas.

La estrategia consiste en enseñar al agente a través de la experiencia y la retroalimentación, y para lograrlo, se deben seguir los siguientes pasos:

1. Observación del entorno
2. Decidir cómo actuar
3. Actuar de acuerdo a esa decisión lo que modifica el entorno
4. Recibir una recompensa o penalización
5. Aprender de las experiencias y refinar la estrategia
6. Iterar hasta que se encuentre la estrategia óptima

2.2.6. Algoritmo araña o rastreadores web

Un rastreador web, también llamado araña web o crawler, es un software que realiza una inspección sistemática y automatizada de las páginas de Internet. Su objetivo principal es indexar el contenido de los sitios web para que los motores de búsqueda

puedan proporcionar resultados relevantes en respuesta a las consultas de los usuarios (Cardona, 2021).

La función principal de las arañas web es identificar nuevas páginas y enlaces para registrarlos y notificarlos a distintos buscadores o plataformas. El funcionamiento de estas arañas es tanto simple como complejo. Al acceder a un sitio web, verifican el archivo robots.txt para determinar qué URL deben evitar visitar. A partir de esta información, comienzan a analizar cada página siguiendo los enlaces disponibles. Las arañas web realizan un rastreo continuo de Internet y utilizan algoritmos para analizar las páginas web en función de las palabras clave y clasificarlas. La información recopilada se almacena de manera organizada en archivos, lo que permite acceder a estos datos durante cada búsqueda.

Tabla 4. Características de la araña web o rastreador web.

Característica	Descripción
Las arañas web son una parte central de los motores de búsqueda	Son elementos fundamentales en los motores de búsqueda.
Los rastreadores web son operados habitualmente por motores de búsqueda	Los motores de búsqueda, como Google y Bing, utilizan rastreadores web.
Los detalles sobre los algoritmos y arquitectura de las arañas web se mantienen como secretos comerciales	La información sobre los algoritmos y la estructura de las arañas web se considera confidencial.
Los rastreadores web pueden descargar/cargar tantos recursos como sea posible de un sitio web concreto	Los rastreadores web intentan descargar todos los recursos disponibles en un sitio web.
Para rastrear la web, es importante poder hacerlo rápidamente y de manera eficiente. También es importante asegurarse de que los datos sean precisos y actualizados.	Los rastreadores web buscan realizar un seguimiento de la web de manera eficiente y actualizada.
Los rastreadores web más comunes incluyen Googlebot, Bingbot, Baiduspider, YandexBot, entre otros	Algunos de los rastreadores web más conocidos son Googlebot, Bingbot, Baiduspider y YandexBot.
No todos los rastreadores son buenos. Algunos pueden ser maliciosos y dañar tu sitio web	Existen rastreadores webs maliciosos que pueden causar daños a un sitio web.

Fuente: Laia Cardona. (2021). Rastreadores web: ¿Qué es un crawler o araña en SEO?

2.2.7. Coeficientes

2.2.7.1. Coeficiente de correlación de Pearson.

La prueba del coeficiente de correlación de Pearson es un método que evalúa la relación estadística existente entre dos variables continuas. Sin embargo, si la relación entre los elementos no es lineal, el coeficiente no representará correctamente la relación existente. Este coeficiente tiene un rango de valores que va desde +1 a -1. Cuando el valor es 0, esto indica que no hay relación alguna entre las dos variables.

Si el valor es mayor a 0, esto significa que existe una asociación positiva entre las dos variables, lo que implica que a medida que aumenta el valor de una variable, también aumenta el valor de la otra variable. Por el contrario, un valor menor a 0 indica una asociación negativa, lo que implica que a medida que aumenta el valor de una variable, disminuye el valor de la otra (Ortega, 2022).

En la fórmula del coeficiente de correlación de Pearson, se utilizan los siguientes símbolos: "x" representa la primera variable, "y" representa la segunda variable, "zx" representa la desviación estándar de la variable "x", "zy" representa la desviación estándar de la variable "y", y "N" representa el número total de datos.

Interpretación del coeficiente de correlación de Karl Pearson:

- Si la correlación es menor a cero, esto indica que la relación entre las variables es negativa, lo que significa que ambas variables están inversamente relacionadas.
- Por otro lado, si la correlación es mayor a cero y es igual a +1, esto indica que la correlación es perfectamente positiva. En este caso, ambas variables están directamente relacionadas y aumentan juntas.
- Sin embargo, si la correlación es igual a cero, esto indica que no existe una relación lineal entre las variables. No obstante, esto no significa que no pueda existir alguna forma de relación no lineal entre ellas.

2.2.7.2. Coeficiente de correlación de Spearman.

El coeficiente de correlación de Spearman es una medida de cómo se relacionan dos variables que no están en una escala numérica. Este coeficiente es útil para analizar datos en los que se ha establecido un orden o ranking de las variables, y permite determinar la relación existente entre ellas (Parra, 2022).

- La variable **'n'** se refiere al número de puntos de datos que están presentes en las dos variables que se están comparando.
- Por otro lado, **'di'** hace referencia a la diferencia de rango del elemento 'n' en ambas variables.

El coeficiente de correlación de Spearman, representado por ρ, tiene un rango de valores que va desde +1 hasta -1.

- Un valor de +1 indica una correlación de rango perfectamente positiva, mientras que un valor de 0 indica que no hay correlación de rango entre las variables.
- Por otro lado, un valor de -1 en ρ indica una correlación de rango perfectamente negativa.
- Si el valor de ρ se aproxima a 0, esto indica una correlación de rango más débil entre las variables.

2.2.7.3. Coeficiente de correlación de Kendall

El Coeficiente de Concordancia de Kendall (W) es utilizado en pruebas estadísticas para determinar el grado de concordancia entre expertos. El valor de W varía entre 0 y 1, donde 1 indica una concordancia total entre los expertos y 0 indica una falta total de concordancia. Es deseable que el valor de W se acerque a 1, y si no se alcanza una concordancia significativa en la primera ronda, se pueden realizar rondas adicionales para mejorar la concordancia (Benites, 2022).

- Podemos definir Cn como el número total de pares que están en concordancia.
- Mientras que NCn representa el número total de pares que están en discordancia.

El valor de τ varía entre -1 y 1, donde -1 indica una correlación negativa perfecta, 0 indica que no hay correlación y 1 indica una correlación positiva perfecta.

El coeficiente de correlación Kendall es una medida útil para datos ordinales o de rango, ya que no se basa en la distribución de los datos ni en la normalidad. Sin embargo, puede ser menos sensible que otras medidas de correlación, como el coeficiente de correlación de Pearson, para detectar ciertos tipos de relaciones entre variables.

2.2.8. Valor p

El valor p (también conocido como nivel de significancia) es una medida estadística que indica la probabilidad de obtener los resultados observados o resultados más extremos si la hipótesis nula es verdadera. En otras palabras, el valor p es la probabilidad de que el resultado observado sea una coincidencia aleatoria (Rodó, 2020).

- La variable D es estocástica y sigue una distribución específica

- Mientras que d es el valor observado del estadístico de contraste en la muestra de datos.

Si el valor p es menor que el nivel de significancia predefinido (por ejemplo, 0,05), se considera que el resultado es estadísticamente significativo y se rechaza la hipótesis nula.

Si la probabilidad de que se produzca el resultado observado es alta, no hay evidencia suficiente para rechazar la hipótesis nula.

2.2.9. Python

Python es un lenguaje de programación fácil de leer y usar. Se centra en la simplicidad y la claridad, lo que lo hace una buena opción para principiantes y profesionales. Se utiliza ampliamente en la comunidad informática debido a su sintaxis clara y su capacidad para manejar tareas de programación. Python destaca por su versatilidad, lo que lo convierte en una poderosa herramienta para el desarrollo de aplicaciones web, análisis de datos, aprendizaje automático y automatización de procesos. Su comunidad activa y su amplia selección de bibliotecas lo convierten en una opción popular tanto para principiantes como para profesionales.

Las bibliotecas de Python son una interfaz o herramienta que permite a los desarrolladores crear fácilmente modelos de aprendizaje automático sin tener que profundizar en las profundidades de los algoritmos (González, 2022). Los casos de uso más comunes de este lenguaje de programación son el desarrollo web, la automatización, las pruebas de software, el análisis de datos, el aprendizaje automático y el desarrollo de juegos.

Ventajas de Python para machine learning:

- Fácil de aprender y usar: Python es un lenguaje de programación que es fácil de entender y usar. Tiene una sintaxis simple que se basa en el inglés, lo que lo hace una buena opción para aprender a programar. Además, Python no necesita ser compilado antes de ejecutarlo, lo que lo hace rápido y fácil de probar.
- Amplia gama de bibliotecas y marcos: Python tiene una amplia gama de bibliotecas y marcos que facilitan la implementación de algoritmos de aprendizaje automático. Algunas de las bibliotecas más populares son scikit-learn, TensorFlow y PyTorch.

- Multiplataforma: Python es un lenguaje multiplataforma, lo que significa que se puede utilizar en múltiples sistemas operativos, incluidos Windows, macOS y Linux.
- Gran comunidad de desarrolladores: Python tiene una gran comunidad de desarrolladores que aportan código, bibliotecas y documentación. Facilita la búsqueda de ayuda y recursos para aprender y utilizar Python para el aprendizaje automático.

Ejemplos de aplicaciones de machine learning en Python:

- Clasificación: Python se puede utilizar para clasificar datos en categorías como imágenes, texto o audio.
- Regresión: Python se puede utilizar para predecir valores continuos como el precio de un producto o la probabilidad de conversión del cliente.
- Detección de patrones: Python se puede utilizar para detectar patrones en los datos, como tendencias o anomalías.
- Aprendizaje automático: Python se puede utilizar para capacitar a agentes que aprendan a tomar decisiones en entornos complejos.

2.2.10. R

R es un lenguaje de programación y un entorno de software utilizado principalmente para análisis estadístico, visualización de datos y desarrollo de modelos predictivos. Es un lenguaje interpretado y orientado a objetos, con una sintaxis similar a la de otros lenguajes como Python (Vega, 2023).

Tabla 5. Características de R.

Característica	Descripción
Análisis estadístico	Potente conjunto de funciones y paquetes para realizar análisis descriptivos, inferenciales y predictivos.
Visualización de datos	Amplia variedad de herramientas y paquetes para crear visualizaciones de datos altamente personalizables.
Paquetes y comunidades	Gran cantidad de paquetes desarrollados por la comunidad de usuarios y organizaciones especializadas.
Reproducibilidad	Facilita la creación de scripts y documentos que pueden ser compartidos y ejecutados por otros usuarios.

Fuente: (Vega, 2023). R para principiantes.

2.2.11. C#

C# (pronunciado "C Sharp") es un lenguaje de programación desarrollado por Microsoft como parte de su plataforma .NET. Es un lenguaje de programación de

propósito general que se utiliza principalmente para desarrollar aplicaciones de escritorio, aplicaciones web, juegos y aplicaciones móviles en el ecosistema de desarrollo de software de Microsoft (Tokio, 2023).

Tabla 6. Características de C#.

Característica	Descripción
Orientado a objetos	Basado en el concepto de clases y objetos para modelar el mundo real de manera modular.
Sintaxis similar a C++ y Java	Diseñado con una sintaxis familiar a desarrolladores de C++ y Java, facilitando su adopción.
Gestión automática de memoria	Utiliza un recolector de basura para gestionar la memoria, eliminando la necesidad de manejo manual.
Plataforma .NET	Parte del ecosistema de desarrollo de software de Microsoft, permitiendo la integración con otras tecnologías de .NET.
Multiplataforma (con .NET Core)	Esfuerzos recientes para hacer C# multiplataforma, permitiendo su ejecución en sistemas operativos diferentes a Windows.

Fuente: (Tokio, 2023). *¿Sabes qué es C#? ¡Conoce este lenguaje de programación!* Tokio School.

Tabla 7. Comparativa de lenguajes de programación.

Característica	Python	R	C#
Tipo de lenguaje	Multipropósito	Especializado en análisis estadístico	Multipropósito
Orientado a objetos	Sí	Sí	Sí
Sintaxis	Fácil de aprender y leer	Sintaxis especializada para análisis estadístico	Similar a C++ y Java
Análisis estadístico	Funciones y paquetes disponibles	Altamente especializado con paquetes de análisis estadístico	Menos enfocado en análisis estadístico, pero tiene herramientas disponibles
Visualización de datos	Bibliotecas como Matplotlib, Seaborn y Plotly	Bibliotecas como ggplot2 y Plotly	Menos énfasis en visualización de datos, pero hay bibliotecas disponibles
Comunidad y soporte	Gran comunidad de usuarios y abundante documentación	Comunidad activa y numerosos paquetes disponibles	Fuerte respaldo de Microsoft y comunidad de .NET
Multiplataforma	Altamente compatible con múltiples plataformas	Compatible, pero menos común en entornos no-Windows	Iniciativas recientes para hacerlo multiplataforma
Desarrollo de aplicaciones	Utilizado en una amplia gama de aplicaciones	Principalmente para análisis estadístico	Utilizado en una variedad de aplicaciones, incluyendo aplicaciones de escritorio, web y móviles
Gestión de memoria	Gestión automática de memoria (a través del recolector de basura)	Enfoque en la eficiencia y rendimiento en análisis de datos	Gestión automática de memoria (a través del recolector de basura)
Ecosistema	Amplia variedad de bibliotecas y herramientas disponibles	Enfoque especializado en análisis de datos, con numerosos paquetes estadísticos	Integrado en el ecosistema de desarrollo de software de Microsoft, con acceso a tecnologías de .NET

Fuente: (Vega, 2023). Lenguajes de programación.

2.2.12. NumPy

NumPy es un conocido paquete de manipulación de matrices de propósito general. Una extensa colección de funciones matemáticas altamente complejas hace que NumPy sea poderoso para manejar vectores grandes y matrices multidimensionales. NumPy es muy útil para trabajar con álgebra lineal, transformadas de Fourier y números aleatorios.

NumPy le permite definir tipos de datos arbitrarios e integrarlos fácilmente en la mayoría de las bases de datos. NumPy también puede actuar como un potente contenedor multidimensional para cualquier tipo de datos (González, 2022).

Ventajas:

- Intuitivo e interactivo.
- Proporciona transformadas de Fourier, capacidades de números aleatorios y otras herramientas para integrar lenguajes de programación como C, C++ y Fortran.
- Puede manejar fácilmente datos multidimensionales.
- Ayuda a manipular la matriz con datos y operaciones como transposición, matriz de identidad y más.
- Permite un mejor rendimiento y recolección de basura al proporcionar una estructura de datos dinámica.

2.2.13. Pandas

Pandas es la biblioteca de Python más popular para el análisis de datos, ya que admite estructuras de datos rápidas, flexibles y expresivas diseñadas para trabajar con datos relacionales y de identidad. Pandas es ahora una biblioteca indispensable para resolver análisis de datos del mundo real en Python. Es muy estable y ofrece un rendimiento altamente optimizado.

Panda utiliza dos tipos principales de estructuras de datos:

- Series (1 dimensión)
- DataFrame (2 dimensiones)

Ventajas:

- Tiene estructuras de datos descriptivas, rápidas y compatibles.

- Admite operaciones como agrupación de datos, integración, iteración, re-indexación y visualización.
- Es muy flexible de usar con otras bibliotecas de Python.
- Contiene funciones informáticas específicas que se pueden implementar utilizando comandos mínimos.
- Gracias a su rendimiento optimizado, se puede aplicar en diversos campos, especialmente aquellos relacionados con los negocios y la educación.

2.2.14. Matplotlib

Matplotlib es una biblioteca de visualización de datos que se utiliza para el trazado 2D para crear gráficos y figuras con calidad de publicación en varios formatos. La biblioteca le ayuda a crear histogramas, gráficos de error, gráficos de dispersión y de líneas con solo unas pocas líneas de código.

Ventajas:

- Admite shells de Python e IPython, scripts de Python, Jupyter Notebook, servidores de aplicaciones web y muchas herramientas GUI.
- Alternativamente, proporciona una interfaz similar a MATLAB para un trazado sencillo.
- La interfaz de Olio ofrece control total sobre las propiedades de los ejes, las propiedades de las fuentes, los estilos de línea y más.
- Ayuda a crear gráficos personalizables, eficientes y precisos.
- Compatible con muchas tarjetas gráficas y sistemas operativos.

2.2.15. Scikit Learn

Scikit Learn proporciona un marco simple y sólido para ayudar a los modelos de aprendizaje automático a aprender, transformar y predecir datos de usuario. También proporciona funciones para ayudar a crear modelos de clasificación, regresión y agrupación. También ofrece una amplia gama de aplicaciones para procesamiento, análisis estadístico, estimación de modelos y más.

Ventajas:

- Tiene un paquete que incluye todos los métodos para implementar algoritmos estándar de aprendizaje automático.

- Tiene una interfaz simple y consistente para ayudar a adaptar y transformar modelos a cualquier conjunto de datos. Puede extraer información de imágenes y texto.
- Es la biblioteca más adecuada para crear canalizaciones que ayuden a la creación rápida de prototipos.
- Es la mejor biblioteca para implementar de manera confiable modelos de aprendizaje automático.

2.2.16. TensorFlow

TensorFlow para Python tiene un poderoso ecosistema de herramientas y recursos comunitarios. Este tipo de conjunto de herramientas permite que el aprendizaje automático y la investigación de aprendizaje profundo creen aplicaciones potentes. Además, Google continúa agregando varias funciones valiosas a TensorFlow para mantenerse al día con la dura competencia.

Ventajas:

- Admite el aprendizaje por refuerzo y otros algoritmos.
- Le permite visualizar modelos de aprendizaje automático directamente usando TensorBoard.
- Los modelos creados con TensorFlow se pueden implementar tanto en CPU como en GPU.
- Ofrece una comunidad muy grande.
- Ofrece TensorBoard, una herramienta para visualizar modelos de aprendizaje automático directamente en el navegador.
- Está listo para la producción.

2.2.17. Keras

Keras es una biblioteca fácil de usar diseñada para reducir la dificultad de los desarrolladores que crean aplicaciones basadas en aprendizaje automático. También proporciona múltiples backends para ayudar a los desarrolladores a integrar plantillas con backends para garantizar una alta estabilidad de la aplicación.

Ventajas:

- Es la mejor biblioteca para investigación y creación de prototipos eficientes.
- Permite una representación simple de redes neuronales.
- Es muy poderoso para visualización y modelado.

2.2.18. PyTorch

Pytorch es una biblioteca de aprendizaje automático desarrollada por Facebook, ahora Meta. y tiene un conjunto de herramientas y bibliotecas que respaldan el aprendizaje automático, la visión por computadora y el procesamiento del lenguaje natural. La mayor ventaja es su facilidad de aprendizaje y uso.

Ventajas:

- Incluye herramientas y bibliotecas que respaldan el aprendizaje profundo.
- La visión por computadora, el procesamiento del lenguaje natural y muchos otros programas de aprendizaje automático.
- Es popular por su velocidad de ejecución.
- Puede manejar gráficos potentes.
- Ayuda a la integración con varios objetos y bibliotecas de Python.
- El proceso de modelado es simple y transparente.

Tabla 8. Comparativa de herramientas de python para machine learning

Herramienta	Tipo	Características	Ventajas	Desventajas
NumPy	Biblioteca de cálculo científico	Matrices y vectores multidimensionales, álgebra lineal, funciones estadísticas	Fácil de usar, potente, eficiente	No es adecuado para análisis de datos de gran tamaño
Pandas	Biblioteca de análisis de datos	Dataframes, series temporales, análisis estadístico	Extensible, fácil de usar, versátil	Puede ser lento para grandes conjuntos de datos
Matplotlib	Biblioteca de visualización de datos	Gráficos de líneas, barras, dispersión, etc.	Potente, flexible, personalizable	Puede ser difícil de aprender
Scikit-learn	Biblioteca de aprendizaje automático	Algoritmos de clasificación, regresión, clustering, etc.	Fácil de usar, amplia gama de algoritmos	Puede ser difícil de entender el funcionamiento interno de los algoritmos
TensorFlow	Framework de aprendizaje automático	Redes neuronales, aprendizaje automático por refuerzo, etc.	Potente, flexible, escalable	Puede ser difícil de aprender
Keras	API de alto nivel para TensorFlow	Redes neuronales, aprendizaje automático por refuerzo, etc.	Fácil de usar, potente, escalable	Requiere TensorFlow
PyTorch	Framework de aprendizaje automático	Redes neuronales, aprendizaje automático por refuerzo, etc.	Potente, flexible, escalable	Puede ser difícil de aprender

Fuente: (González, 2022). Herramientas de Python para Machine Learning.

2.2.19. Selenium

Una herramienta de código abierto, creada principalmente para automatizar navegadores y probar aplicaciones, posibilita ejecutar diversas acciones que simulan la interacción humana. Esta capacidad ha ampliado sus posibilidades de uso hacia el web scraping, donde se requiere identificar etiquetas, clases e identificadores para extraer información específica (Selenium, 2024). Principalmente empleada en la evaluación de aplicaciones web, esta herramienta también se utiliza para extraer datos de páginas web mediante técnicas de "scraping".

2.2.20. Beautiful Soup

La librería Beautiful Soup facilita la extracción de contenido de páginas web y su conversión en estructuras de datos de Python, como listas, matrices o diccionarios. Esta librería es ampliamente reconocida por su documentación exhaustiva y por la organización clara de sus funcionalidades. Además, cuenta con una comunidad activa que ofrece diversas soluciones para aprovechar al máximo sus capacidades (DataScientest, 2022).

Los sitios web se construyen utilizando lenguajes de marcado como HTML y CSS. Mientras que HTML se encarga de la estructura y organización del contenido, CSS se utiliza para gestionar el aspecto visual de la página, como colores y tamaños de texto. En el ámbito del desarrollo web, el término "tag soup" (sopa de etiquetas) se refiere de forma despectiva a la escritura incorrecta o desordenada del HTML en una página web.

2.2.21. Flask

Flask es un Framework "micro" desarrollado en Python, diseñado para simplificar la creación de Aplicaciones Web siguiendo el patrón MVC, el término "micro" no implica que sea limitado o adecuado solo para páginas web simples, sino que al instalar Flask, se obtienen las herramientas esenciales para construir una aplicación web funcional. Sin embargo, en caso de necesitar nuevas funcionalidades en el futuro, Flask cuenta con un amplio conjunto de extensiones (plugins) que pueden ser instaladas para añadir más características a la aplicación.

2.2.22. Web Scraping

El web scraping, también conocido como raspado de páginas web, permite automatizar la extracción y estructuración de datos de una o varias páginas web

interconectadas a través de enlaces. El propósito es utilizar estos datos en diversas actividades. Este proceso implica varias etapas, como el análisis de los sitios web, la identificación del contenido y la organización de los datos (Vlad & Leigh, 2022).

Cada una de estas etapas requiere el uso de herramientas específicas. Es importante tener en cuenta que, aunque se pueden automatizar muchas partes del proceso, algunas tareas aún necesitan la intervención humana debido a su complejidad o variabilidad.

2.2.23. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos de código abierto diseñado para manejar bases de datos relacionales, las cuales almacenan datos interrelacionados en tablas compuestas por registros y campos. Cada registro posee una identificación única, conocida como clave. Este tipo de bases de datos son ampliamente utilizadas por desarrolladores en la creación de sitios web (UNIR, 2021).

PostgreSQL, también conocido como Postgres, brinda diversas funcionalidades para trabajar con estas bases de datos, incluyendo consultas, inserción, modificación y eliminación de datos. Además, ofrece la capacidad de realizar consultas no relacionales, ampliando así su utilidad y flexibilidad en el manejo de datos.

2.2.24. Visual Studio Code

Se trata de un editor de código abierto, compatible con múltiples plataformas, altamente reconocido y desarrollado por Microsoft. Destaca por su ligereza y potencia en el manejo de diversos lenguajes de programación. Es compatible con sistemas operativos como Windows, Linux, MacOS y entornos web, y ofrece una amplia gama de extensiones personalizables para adaptarse a las necesidades individuales. Además, cuenta con soporte para cualquier lenguaje de programación y puede ejecutarlos mediante el uso de terminal, tecnología web y el entorno Node.js para JavaScript.

2.2.25. Correlación de variables

La correlación de variables es una medida estadística que indica la fuerza y dirección de una relación lineal entre dos variables estadísticas. Se dice que dos variables cuantitativas están correlacionadas si los valores de una varían sistemáticamente con relación al valor homónimo de la otra (Zamorano, 2020).

Algunos coeficientes de correlación tienden a medirse de 1 a -1, donde:

- Un valor de r = 1 indica una correlación positiva perfecta, lo que significa que los valores de las dos variables aumentan o disminuyen juntas.
- Un valor de r = -1 indica una correlación negativa perfecta, lo que significa que los valores de las dos variables aumentan o disminuyen en direcciones opuestas.
- Un valor de r = 0 indica que no existe correlación entre las dos variables.

2.2.26. Correlación de palabras

Un algoritmo de machine learning para establecer conexiones entre palabras emplea un conjunto de datos de texto para comprender la relación que existe entre las palabras. Este conjunto de datos puede consistir en un corpus de texto, como una recopilación de libros o artículos, o en un grupo de datos de texto creado por el usuario (Kaur, 2021, pág. 10).

- En un primer paso, el algoritmo representa cada palabra como un vector de características. Dicho vector puede incluir información sobre la frecuencia de uso de la palabra, su posición en la oración o su relación con otras palabras.
- Una vez que las palabras se han transformado en vectores, el algoritmo puede evaluar la correlación entre ellas. La correlación es una medida de la relación lineal entre dos variables. En el contexto de las palabras, la correlación puede señalar si las palabras suelen aparecer juntas o no.

2.2.27. Text Mining

El Text-Mining tuvo su origen en los años 80 cuando la gestión de textos demandaba un esfuerzo humano considerable. No obstante, los progresos tecnológicos han posibilitado un rápido avance en esta área durante la última década.

El Text-Mining implica el análisis de conjuntos de textos con el propósito de identificar conceptos y temas fundamentales, así como de revelar relaciones y tendencias subyacentes, prescindiendo de la necesidad de conocer las palabras o términos exactos empleados por los autores para expresar dichos conceptos (DataScientest, 2022).

2.2.28. Técnicas de Text-Mining

La Extracción de términos: es una técnica elemental que consiste en identificar los términos esenciales y entidades lógicas de un texto. Esta metodología representa el nivel más básico del Text-Mining, donde la estructura de datos más sencilla se

manifiesta a través del vector de características, que es una lista de palabras ponderadas presentes en el texto (Gomez, 2022).

La Extracción de información: se fundamenta en los términos previamente extraídos del texto para reconocer relaciones fundamentales, como, por ejemplo, las diversas funciones desempeñadas por diferentes empresas durante una fusión. Esta etapa se concentra en recopilar un conjunto de hechos que constituyen un evento específico.

El Análisis relacional: fusiona múltiples enlaces para crear modelos de procesos complejos que pueden extenderse a través de varios pasos. Este análisis se compone de un conjunto de técnicas diseñadas para comprender las relaciones entre diversas entidades que están interconectadas de forma múltiple.

2.2.29. CRISP-DM

CRISP-DM significa Proceso estándar entre industrias para minería de datos y es un método ampliamente utilizado para proyectos de análisis y minería de datos. Proporciona un enfoque estructurado para guiar a las organizaciones a través de los procesos de exploración, modelado e implementación de datos (Arias, 2023).

Una descripción general del proceso CRISP-DM es la siguiente:

1. Comprender el negocio: el primer paso es comprender los objetivos y requisitos del proyecto desde una perspectiva empresarial. Esto incluye definir las metas, los objetivos y los requisitos del proyecto y traducirlos en objetivos de extracción de datos.
2. Análisis de datos: en esta fase, se identifican las fuentes de datos y se realiza la recopilación y exploración de datos. Esto incluye recopilar información sobre los datos disponibles, evaluar su calidad y comprender sus características.
3. Preparación de datos: este paso implica preparar los datos para el análisis. Este paso implica limpiar los datos, manejar los valores faltantes, transformar variables y seleccionar los parámetros apropiados. El objetivo es crear un conjunto de datos limpio y estructurado adecuado para el muestreo.
4. Modelado: en este paso, se seleccionan y aplican varias técnicas de muestreo al conjunto de datos preparado. Esto incluye la construcción y validación de modelos predictivos utilizando algoritmos estadísticos y de aprendizaje automático. Se examinan varios algoritmos y parámetros para encontrar el mejor modelo para los datos.

5. Evaluación: una vez creado el modelo, se evalúa para determinar su utilidad para resolver problemas comerciales. Evalúe la calidad de su modelo calculando medidas de rendimiento como precisión, exactitud, recuperación y puntuación F1. Compare las plantillas y elija la mejor para implementar.
6. Despliegue: en el paso final, la plantilla seleccionada se envía al entorno de producción. Esto incluye integrar el modelo en los sistemas y procesos existentes y ponerlo a disposición de los usuarios finales. También se estableció un plan de seguimiento y mantenimiento para asegurar la efectividad del modelo implementado

2.2.30. Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) capacita a las computadoras para entender el lenguaje humano de manera similar a como lo hacen los humanos. Tanto si se trata de lenguaje hablado como escrito, el PLN emplea la inteligencia artificial para adquirir información del mundo real, procesarla y darle significado de una manera que las computadoras puedan comprender. Así como los humanos tienen diferentes sentidos, como oído y vista, las computadoras cuentan con programas para leer y micrófonos para capturar audio. Y así como los humanos tienen un cerebro para procesar esta información, las computadoras tienen programas para procesar sus propias entradas (Burns & lutkevinch, 2024).

En algún punto del proceso, la entrada se convierte en un código que la computadora puede interpretar.

El procesamiento del lenguaje natural consta de dos fases principales: pre procesamiento de datos y desarrollo de algoritmos:

- El pre procesamiento de datos: implica la preparación y "limpieza" de los datos de texto para que las máquinas puedan analizarlos. Esto implica poner los datos en un formato manejable y resaltar las características del texto con las que un algoritmo puede trabajar. Esto incluye acciones como la tokenización, eliminación de palabras comunes, lematización y derivación, y etiquetado de partes del discurso.
- Una vez que los datos han sido pre procesados, se desarrolla un algoritmo para procesarlos. Existen varios algoritmos diferentes de procesamiento del lenguaje natural, pero los más comunes son los sistemas basados en reglas y los sistemas

basados en aprendizaje automático. Los sistemas basados en reglas emplean reglas lingüísticas diseñadas con cuidado, mientras que los sistemas basados en aprendizaje automático utilizan métodos estadísticos y aprenden a realizar tareas basadas en los datos de entrenamiento que reciben. Con el uso de técnicas como el aprendizaje profundo y las redes neuronales, los algoritmos de PLN mejoran sus capacidades mediante un procesamiento y aprendizaje continuo.

Algunas de las funciones principales realizadas por los algoritmos de procesamiento del lenguaje natural incluyen:

Clasificación de textos**:** implica asignar etiquetas a los textos para categorizarlos. Esta función es útil para el análisis de sentimientos, donde el algoritmo puede determinar las emociones detrás de un texto. Por ejemplo, cuando se menciona la marca A en ciertos textos, el algoritmo puede discernir cuántas menciones fueron positivas y cuántas negativas. También es útil para detectar intenciones, ayudando a predecir las acciones basadas en el texto producido por el hablante o escritor (Burns & lutkevinch, 2024).

Extracción de texto: implica la síntesis automática de texto y la identificación de datos relevantes. Un ejemplo es la extracción de palabras clave, que busca las palabras más importantes del texto, útil para la optimización en motores de búsqueda. Aunque esto requiere cierta programación en el procesamiento del lenguaje natural y no es completamente automatizado, existen herramientas simples que automatizan gran parte del proceso: el usuario solo necesita configurar parámetros dentro del programa. Por ejemplo, una herramienta puede extraer las palabras más frecuentes en el texto. Otra técnica es el reconocimiento de entidades nombradas, que identifica nombres de personas, lugares y otras entidades en el texto (Burns & lutkevinch, 2024).

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

La presente investigación tiene un enfoque mixto (cualitativo y cuantitativo) dando solución al problema planteado.

3.1.1.1. Cuantitativo

Para el enfoque cuantitativo implico la recopilación y análisis de datos de proyectos de investigación a partir de diversas fuentes como: bases de datos y repositorios, seguido por la aplicación de técnicas de machine learning para identificar patrones en diferentes investigaciones ya existentes

3.1.1.2. Cualitativo

El enfoque cualitativo se enfocó en la comprensión y análisis detallado de las experiencias y percepciones de los investigadores sobre la clasificación de proyectos de investigación y cómo perciben la posible integración de técnicas de machine learning en el proceso.

3.1.2. Tipo de Investigación

3.1.2.1. Descriptiva

En esta investigación el objetivo fue describir y comprender el fenómeno de la clasificación automática en su contexto actual. Se podrían utilizar métodos como: deductivo, hipotético-deductivo y análisis de datos secundarios para recopilar información sobre la forma actual en que se clasifican los proyectos de investigación y los factores que influyen en este proceso. Los resultados se presentarían de manera clara y detallada para describir el fenómeno de la clasificación automática y proporcionar información útil para futuros estudios.

3.1.2.2. Investigación – acción

En este tipo de investigación se abordan problemas específicos relacionados con la clasificación de proyectos de investigación mediante un proceso iterativo de acción

y evaluación. El enfoque sería práctico y participativo, involucrando a los interesados directamente en el proceso de investigación. Se llevarían a cabo acciones específicas para facilitar la clasificación de los proyectos de investigación, como la implementación de nuevos métodos o la formación de los actores implicados. Luego, se realizarían ajustes en función de los resultados obtenidos. Este proceso se repetiría hasta alcanzar un nivel óptimo de clasificación automática.

3.2. IDEA A DEFENDER

La integración de algoritmos machine learning en el proceso de clasificación de proyectos de investigación resultará en una correlación más precisa que la clasificación manual.

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

Tabla 9. Operacionalización de variables

Variable	Tipo de Variable	Definición	Dimensión	Indicadores	Técnica	Instrumento
Algoritmos de Machine learning	Independiente	Los algoritmos de Machine Learning se dividen generalmente en tres categorías principales: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo (IBM Cloud Education, 2020).	Precisión Recall F1-Score	Porcentaje de precisión Porcentaje de Recall Valor de F1-Score	Evaluación por validación por lotes.	Matriz de Confusión.
Correlación de proyectos de investigación	Dependiente	La correlación en proyectos de investigación se refiere a la medida de la relación o asociación entre dos o más variables en un estudio y un coeficiente de correlación se utiliza para medir la fuerza y la dirección de la relación entre las variables, y puede variar desde -1 a 1 (Lifeder, 2022).	Fuerza de la correlación.	Coeficiente de correlación.	Coeficiente de correlación de Pearson.	Python

Este tema de investigación tributa al proyecto de investigación SMART DATA LAB

3.4. MÉTODOS UTILIZADOS

3.4.1. Métodos

3.4.1.1. Método inductivo

El método inductivo es un proceso lógico que se utiliza para obtener conclusiones generales a partir de observaciones específicas. Consiste en recolectar información, analizarla y extraer patrones y tendencias para llegar a una conclusión general, a partir de esos patrones comunes, se extrae una conclusión general o una hipótesis que se puede probar a través de la recopilación y el análisis de más datos. (Clarín, 2020).

Aunque el método inductivo puede ser útil para obtener conclusiones generales, es importante tener en cuenta que estas conclusiones pueden no ser ciertas en todos los casos. Por lo tanto, es importante considerar la posibilidad de que haya excepciones a las conclusiones generales que se extraigan a través del método inductivo.

De esta manera te permitirá llegar a conclusiones fundamentadas y contribuir al conocimiento en el campo de los algoritmos de machine learning.

3.4.1.2. Método hipotético-deductivo

Este método combina el enfoque inductivo y el enfoque deductivo. El método científico comienza con una hipótesis general, que es una idea o suposición sobre cómo funciona el mundo. Se hacen predicciones específicas basadas en la hipótesis, y luego se llevan a cabo experimentos o pruebas para probar o refutar esas predicciones. Si los datos son consistentes con la hipótesis, se considera que está respaldada y se puede avanzar en la formulación de teorías más amplias. Si los datos no son consistentes con la hipótesis, se descarta y se busca una nueva explicación (Bastis Consultores , 2021).

Así, se posibilita la investigación y comprensión del comportamiento y desempeño de los algoritmos de machine learning.

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Metodología CRISP-DM

1. Comprensión del negocio:

Tema: "Algoritmos de machine learning para la correlación de proyectos de investigación". Se identifican los siguientes objetivos:

- Comparar la precisión de algoritmos machine learning con técnicas tradicionales de clasificación para determinar su aplicación.
- Evaluar la precisión de diferentes algoritmos machine learning para la correlación entre proyectos de investigación.
- Proponer un algoritmo machine learning para la correlación entre proyectos de investigación.

2. Comprensión de los datos:

Los datos iniciales que se utilizan se los recolecta mediante web scraping o raspado web a repositorios universitarios del Ecuador.

- Naturaleza de los datos: información disponible; título del proyecto, fecha, autores, enlace de referencia

Tabla 10. Web Scraping a Repositorios Universitarios

Universidad	Web Scraping
Universidad Politécnica estatal del Carchi	temas_td = soup.find_all('td', headers='t2') for tema in temas_td: fecha = tema.find_previous('td', headers='t1').text.strip() enlace = "http://repositorio.upec.edu.ec" + tema.find('a')['href'] autor = tema.find_next('td', headers='t3').text.strip() resultados.append((fecha, tema.text.strip(),enlace,autor))
Universidad del Pacifico	temas_td = soup.find_all('td', headers='t2') for tema in temas_td: fecha = tema.find_previous('td', headers='t1').text.strip() enlace = "https://uprepositorio.upacifico.edu.ec/" + tema.find('a')['href'] autor = tema.find_next('td', headers='t3').text.strip() resultados.append((fecha, tema.text.strip(),enlace,autor))
Universidad de Especialidades Espíritu Santo	temas_td_fecha = soup.find_all('td', headers='t1') temas_td_titulo = soup.find_all('td', headers='t2') for fecha, tema in zip(temas_td_fecha, temas_td_titulo): fecha_texto = fecha.text.strip() tema_texto = tema.find('a').text.strip() enlace = "http://repositorio.uees.edu.ec/" + tema.find('a')['href'] autor = tema.find_next('td', headers='t3').text.strip() resultados.append((fecha_texto, tema_texto,enlace,autor))
Universidad Nacional de Loja	temas_td_fecha = soup.find_all('td', headers='t1', class_='evenRowEvenCol', nowrap='nowrap', align='right') temas_td_titulo = soup.find_all('td', headers='t2', class_='evenRowOddCol') for fecha, tema in zip(temas_td_fecha, temas_td_titulo): fecha_texto = fecha.text.strip() tema_texto = tema.find('a').text.strip() enlace = "https://dspace.unl.edu.ec/" + tema.find('a')['href'] autor = tema.find_next('td', headers='t3').text.strip() resultados.append((fecha_texto, tema_texto,enlace,autor))
Universidad de Azuay	temas_td = soup.find_all('td', headers='t2') for tema in temas_td: fecha = tema.find_previous('td', headers='t1').text.strip() enlace = "https://dspace.uazuay.edu.ec/" + tema.find('a')['href']#tema.find('a').get('href') autor = tema.find_next('td', headers='t3').text.strip() resultados.append((fecha, tema.text.strip(),enlace,autor))
Universidad Internacional del Ecuador	temas_td = soup.find_all('td', headers='t2') for tema in temas_td: fecha = tema.find_previous('td', headers='t1').text.strip() enlace = "https://repositorio.uide.edu.ec/" + tema.find('a')['href'] autor = tema.find_next('td', headers='t3').text.strip() resultados.append((fecha, tema.text.strip(),enlace,autor))

3. Preparación de los datos:

Se realizan los siguientes pasos para preparar los datos:

- Limpieza de texto: Se utiliza una función para corregir el texto, convirtiéndolo a minúsculas, eliminando caracteres especiales, URLs, puntuación, números y palabras vacías.

```
def corregir(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

# Aplicar la función corregir a la columna "Resultados"
df["Resultados"] = df["Resultados"].apply(corregir)

# Definir palabras vacías personalizadas
palabras_vacias_personalizadas = ["de", "en", "los", "la", "las", "sobre", "y",
"el", "para", "entre", "del", "su", "(", ")"]

# Eliminación de palabras vacías personalizadas
df["Resultados"] = df["Resultados"].apply(lambda x: ' '.join([word for word
in x.split() if word.lower() not in palabras_vacias_personalizadas]))

df["Resultados"] = df["Resultados"].apply(corregir)
```

- Vectorización de texto: Se aplica CountVectorizer para convertir el texto en características numéricas utilizables por los algoritmos de machine learning.

```
#Dividimos en train y test y aplicamos Countvectorizer.
x = df["Resultados"]
y = df["Resultados"]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25,
random_state=1)
```

```
# Guardamos la función en variable
vectorizer = CountVectorizer()

# Codificamos el train y test de la variable x, es decir, el texto
x_train = vectorizer.fit_transform(x_train)
x_test = vectorizer.transform(x_test)

# Obtenemos el vocabulario como un diccionario
vocabulario = vectorizer.vocabulary_

# Convertimos el diccionario en un DataFrame de Pandas y transponemos
vocabulario_df = pd.DataFrame(vocabulario.items(), columns=['Palabra', 'Indice'])
```

4. Modelamiento:

Para el modelamiento se aplican técnicas de procesamiento de lenguaje natural (PNL). Los pasos son:

- División de los datos en conjuntos de entrenamiento y prueba.

```
x = df["Resultados"]
y = df["Categoria_Num"]
# Dividir los datos en conjuntos de entrenamiento y prueba
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

- Utilización de TF-IDF (Term Frequency-Inverse Document Frequency) para ponderar la importancia de las palabras en los documentos.

```
tfidf = TfidfVectorizer()
x_train_tfidf = tfidf.fit_transform(x_train)
x_test_tfidf = tfidf.transform(x_test)
```

- Entrenamiento del modelo:

Tabla 11. Entrenamiento de modelos

Modelo	Entrenamiento
Logistic Regression	LR = LogisticRegression() LR.fit(x_train_tfidf,y_train)
Decision Tree	DT = DecisionTreeClassifier() DT.fit(x_train_tfidf, y_train)
Random Forest	RF = RandomForestClassifier(random_state=0) RF.fit(x_train_tfidf, y_train)
Gradient Boosting	GB = GradientBoostingClassifier(random_state=0) GB.fit(x_train_tfidf, y_train)
KNN	KNN = KNeighborsClassifier(n_neighbors=5) KNN.fit(x_train_tfidf, y_train)
Red Neuronal Artificial	ANN = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000) ANN.fit(x_train_tfidf, y_train)

5. Evaluación:

Se evalúa el rendimiento de los modelos mediante métricas de evaluación como precisión, recall, F1-score y matriz de confusión. Se compara el rendimiento de los algoritmos de machine learning con las técnicas tradicionales de clasificación.

Logistic Regression

```
pred_lr=LR.predict(x_test_tfidf)

print(classification_report(y_test, pred_lr))
```

	precision	recall	f1-score	support
0	1.00	0.88	0.94	43
1	0.96	0.98	0.97	209
2	0.89	0.98	0.93	146
3	0.99	0.96	0.97	89
4	0.97	0.79	0.87	48
accuracy			0.95	535
macro avg	0.96	0.92	0.94	535
weighted avg	0.95	0.95	0.95	535

Figura 1. Evaluación de Logistic Regression

Matriz de confusión:

```
conf_matrix_lr = confusion_matrix(y_test, pred_lr)

print("Matriz de Confusión:")

print(conf_matrix_lr)
```

```
Matriz de Confusión:
[[102   0   0   0   0]
 [ 16   9   1   0   0]
 [  2   0  16   0   0]
 [  6   0   0   5   0]
 [  6   1   0   0   5]]
```

Figura 2. Matriz de confusión de Logistic Regression

Decision Tree

```
pred_dt = DT.predict(x_test_tfidf)

print(classification_report(y_test, pred_dt))
```

	precision	recall	f1-score	support
0	0.91	0.93	0.92	43
1	0.84	0.84	0.84	209
2	0.79	0.79	0.79	146
3	0.86	0.89	0.87	89
4	0.77	0.71	0.74	48
accuracy			0.83	535
macro avg	0.83	0.83	0.83	535
weighted avg	0.83	0.83	0.83	535

Figura 3. Evaluación de Decision Tree

Matriz de confusión:

```
conf_matrix_lr = confusion_matrix(y_test, pred_dt)

print("Matriz de Confusión:")

print(conf_matrix_lr)
```

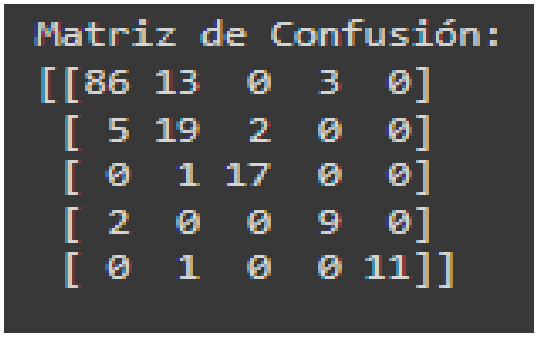
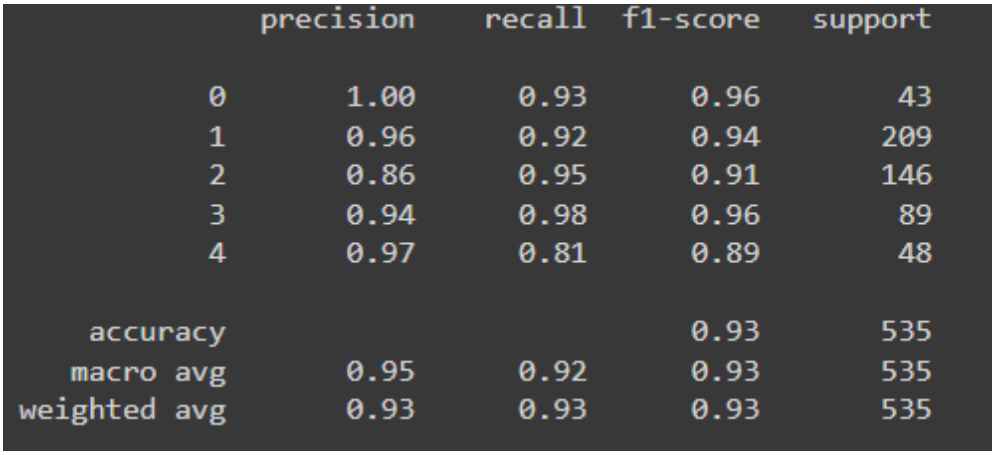


Figura 4. Matriz de confusión de Decision Tree

Random Forest

pred_rf = RF.predict(x_test_tfidf)

print(classification_report(y_test, pred_rf))



	precision	recall	f1-score	support
0	1.00	0.93	0.96	43
1	0.96	0.92	0.94	209
2	0.86	0.95	0.91	146
3	0.94	0.98	0.96	89
4	0.97	0.81	0.89	48
accuracy			0.93	535
macro avg	0.95	0.92	0.93	535
weighted avg	0.93	0.93	0.93	535

Figura 5. Evaluación de Random Forest

Matriz confusión:

conf_matrix_lr = confusion_matrix(y_test, pred_dt)

print("Matriz de Confusión:")

print(conf_matrix_lr)

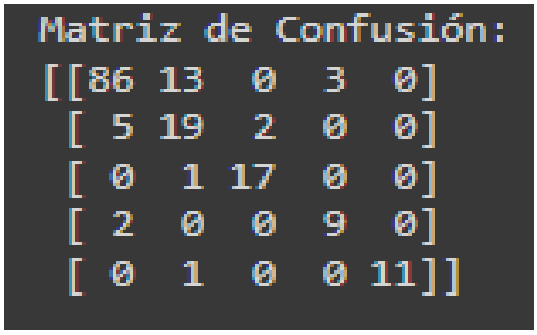


Figura 6. Matriz de confusión de Random Forest

Gradient Boosting

```
pred_gb = GB.predict(x_test_tfidf)

print(classification_report(y_test, pred_gb))
```

	precision	recall	f1-score	support
0	0.95	0.98	0.97	43
1	0.95	0.92	0.93	209
2	0.90	0.95	0.93	146
3	0.97	0.94	0.95	89
4	0.89	0.85	0.87	48
accuracy			0.93	535
macro avg	0.93	0.93	0.93	535
weighted avg	0.93	0.93	0.93	535

Figura 7. Evaluación de Gradient Boosting

Matriz de confusión:

```
conf_matrix_lr = confusion_matrix(y_test, pred_gb)

print("Matriz de Confusión:")

print(conf_matrix_lr)
```

```
Matriz de Confusión:
[[92  6  1  3  0]
 [ 3 21  1  0  1]
 [ 0  0 18  0  0]
 [ 3  0  0  8  0]
 [ 0  0  0  0 12]]
```

Figura 8. Matriz de confusión Gradient Boosting

KNN

```
pred_knn = KNN.predict(x_test_tfidf)

print(classification_report(y_test, pred_knn))
```

	precision	recall	f1-score	support
0	0.79	0.98	0.88	43
1	0.89	0.77	0.83	209
2	0.76	0.82	0.79	146
3	0.85	0.91	0.88	89
4	0.75	0.79	0.77	48
accuracy			0.82	535
macro avg	0.81	0.85	0.83	535
weighted avg	0.83	0.82	0.82	535

Figura 9. Evaluación de KNN

Matriz de confusión:

```
conf_matrix_lr = confusion_matrix(y_test, pred_knn)

print("Matriz de Confusión:")

print(conf_matrix_lr)
```

```
Matriz de Confusión:
[[88  5  3  3  3]
 [ 8 17  1  0  0]
 [ 0  0 18  0  0]
 [ 3  0  0  8  0]
 [ 1  0  0  0 11]]
```

Figura 10. Matriz de confusión KNN

Red Neuronal Artificial

```
pred_ann = ANN.predict(x_test_tfidf)

print(classification_report(y_test, pred_ann))
```

	precision	recall	f1-score	support
0	0.95	0.88	0.92	43
1	0.93	0.88	0.90	209
2	0.84	0.92	0.88	146
3	0.94	0.94	0.94	89
4	0.86	0.88	0.87	48
accuracy			0.90	535
macro avg	0.90	0.90	0.90	535
weighted avg	0.90	0.90	0.90	535

Figura 11. Evaluación de Red Neuronal Artificial

Matriz de confusión:

```
conf_matrix_lr = confusion_matrix(y_test, pred_ann)

print("Matriz de Confusión:")

print(conf_matrix_lr)
```

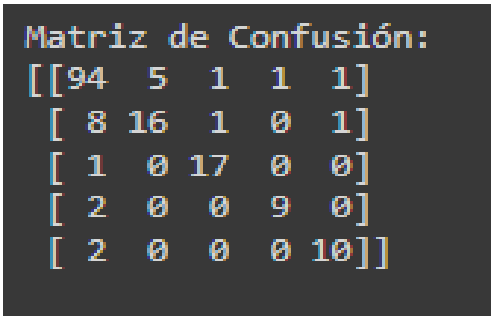


Figura 12. Matriz de confusión de Red Neuronal Artificial

Comparación de la precisión de algoritmos machine learning:

Tabla 12. Comparativa de la precisión de algoritmos machine learning

Modelo	Precisión	Recall	F1-Score	Soporte
Regresión Logística	0.95	0.95	0.95	535
Árbol de Decisión	0.82	0.84	0.83	535
Bosque Aleatorio	0.92	0.93	0.92	535
Gradient Boosting	0.89	0.89	0.89	535
KNN	0.76	0.83	0.78	535
Red Neuronal Artificial	0.85	0.87	0.86	535

A continuación, se describen los resultados en la tabla comparativa de la precisión de algoritmos machine learning.

- Regresión Logística: Precisión = 0.95
  Este modelo tiene la precisión más alta entre todos los evaluados, con un valor del 95%.
- Árbol de Decisión: Precisión = 0.82
  El árbol de decisión muestra una precisión del 82%, lo que lo posiciona en un nivel medio en comparación con otros modelos.
- Bosque Aleatorio: Precisión = 0.92
  El bosque aleatorio logra una precisión del 92%, lo que lo coloca como uno de los modelos con mejor rendimiento en términos de precisión.
- Gradient Boosting: Precisión = 0.89
  Este modelo alcanza una precisión del 89%, mostrando un buen rendimiento en la clasificación de datos.
- KNN: Precisión = 0.76
  El KNN obtiene una precisión del 76%, lo que lo sitúa en un nivel inferior en términos de precisión en comparación con otros modelos.
- Red Neuronal Artificial: Precisión = 0.85

La red neuronal artificial logra una precisión del 85%, mostrando un rendimiento sólido, pero ligeramente inferior al de la Regresión Logística y el Bosque Aleatorio.

Basándonos en la comparativa anterior, es evidente que el algoritmo de regresión logística sobresale como el mejor modelo en el contexto de clasificación. Destaca por su mayor precisión, evidenciada por los valores más altos en las métricas proporcionadas.

6. Despliegue:

Los resultados obtenidos se utilizan para analizar los temas extraídos de los repositorios universitarios, lo que permite clasificarlos de manera más precisa. Esto incrementa las posibilidades de identificar temas relevantes para futuros proyectos de investigación.

Confiriendo de correlación Pearson:

```
resultados = pd.DataFrame({'Real': y_test, 'Prediccion': pred_lr})
resultados = pd.DataFrame({'Real': y_test, 'Prediccion': pred_lr})
# Calcular el coeficiente de correlación de Pearson
correlacion = resultados['Real'].corr(resultados['Prediccion'])
print("Coeficiente de correlación de Pearson:", correlacion)
```

Logistic Regression: 0.9761484322014206

Decision Tree: 0.85525558566921415

Random Forest: 0.650120238073558

Gradient Boosting: 0.7802546987201315

KNN: 0.6987254142434546

Red Neuronal Artificial: 0.7856987420325420

4.1.2. Aplicación web

4.1.2.1. Fase 1: Análisis de requerimientos

En esta fase inicial, se lleva a cabo un estudio exhaustivo del proyecto para comprender a fondo todos los factores que lo componen. Este análisis es crucial para obtener información relevante y pertinente al problema que se busca abordar. A

partir de la información recopilada, se establecen los requisitos fundamentales que el sistema debe cumplir para satisfacer las necesidades planteadas. Finalmente, se examinan los requerimientos generales del sistema para garantizar su correcto funcionamiento y alcance.

Datos preliminares.

La presente investigación se enmarca en las actividades del laboratorio de datos Smart DataLab de la Universidad Politécnica Estatal del Carchi (UPEC), ubicado en el cantón Tulcán, Ecuador. El laboratorio, dedicado al análisis y procesamiento de grandes volúmenes de datos, actualmente enfrenta la necesidad de contar con una herramienta tecnológica que facilite la gestión de sus proyectos de investigación. En la actualidad, el laboratorio Smart DataLab carece de una aplicación web que permita la búsqueda y clasificación eficiente de sus proyectos de investigación. A causa de ello, la gestión de esta información se realiza de forma manual, lo que implica un proceso tedioso, propenso a errores y que limita la productividad del equipo

Requerimientos del aplicativo web.

El sistema propuesto contará con las siguientes funcionalidades principales:

Modulo para el cliente:

- Interfaz de usuario amigable: Se desarrollará una interfaz de usuario web intuitiva y accesible para que los usuarios puedan consultar de manera sencilla la información sobre los diferentes temas de proyectos de investigación
- Búsqueda y filtrado avanzado: La aplicación web permitirá realizar búsquedas y filtrados avanzados de la información sobre proyectos de investigación, utilizando diversos criterios como el tema, la universidad, el año de publicación, enlace de referencia y autores.
- Visualización detallada de proyectos: Se brindará información detallada sobre cada proyecto de investigación, incluyendo el tema, la universidad, el año de publicación, enlace de referencia y autores.
- Agrupación y clasificación de proyectos: La información sobre los proyectos de investigación se organizará de manera eficiente mediante mecanismos de búsqueda, agrupación y clasificación, permitiendo a los usuarios encontrar fácilmente los proyectos de su interés.

Módulo de administración:

- Web scraping o rapado web: El sistema incorporará un módulo de web scraping o rapado web que permitirá extraer automáticamente información sobre proyectos de investigación de sitios web de la UPEC y otras universidades adicionales.
- Subida de información a la base de datos: La información extraída mediante web scraping se procesará y almacenará de forma organizada en una base de datos PostgreSQL, garantizando la integridad y confiabilidad de los datos.
- Gestión de usuarios y permisos: El módulo de administración contará con herramientas para la gestión de usuarios y permisos, permitiendo controlar el acceso a las diferentes funcionalidades del sistema.
- Registro de actividades: Se implementará un sistema de registro de actividades que permitirá rastrear las acciones realizadas por los usuarios dentro del sistema, facilitando la auditoría y el seguimiento de los procesos.

4.1.2.2. Fase 2: Diseño funcional del sistema

Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor representa un modelo de sistema centralizado, en el que las aplicaciones y recursos residen en un servidor central. Desde este punto, la información se distribuye y gestiona, lo que permite a los usuarios acceder a través de la red sin la necesidad de descargar una aplicación. Este enfoque mejora el rendimiento del sistema de información en su conjunto, involucrando dos actores que se complementan entre sí.

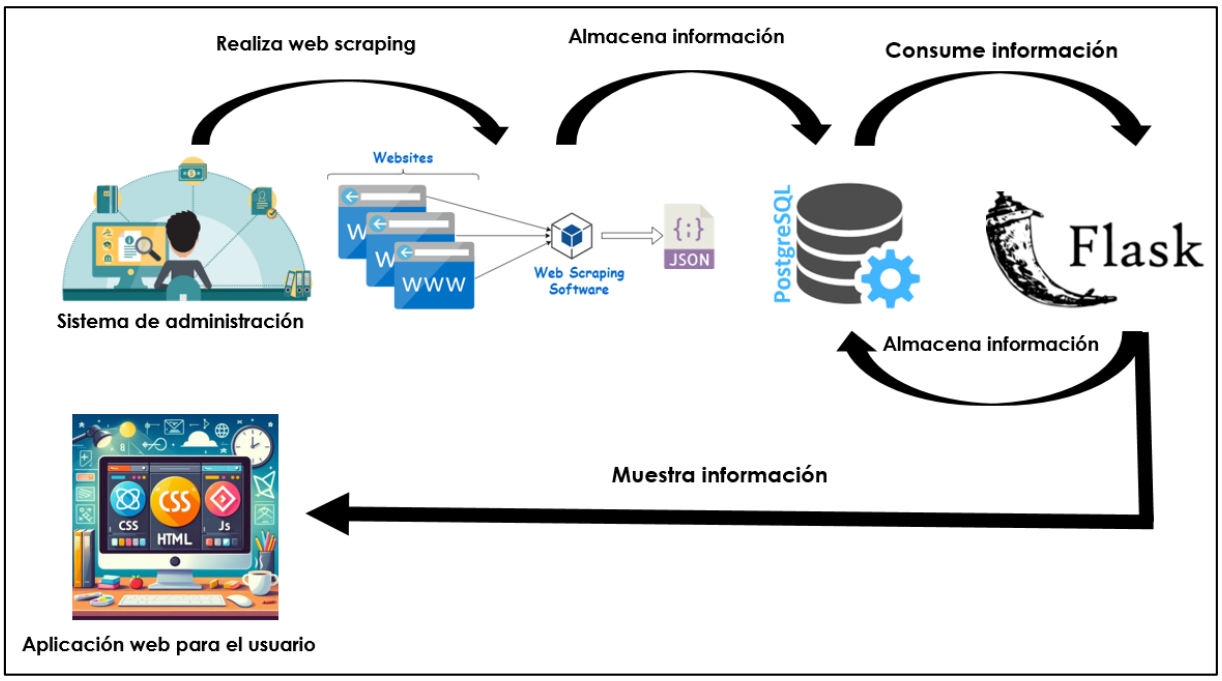


Figura 13. Arquitectura de la aplicación web

Cliente (frontend):

- Realiza las solicitudes de servicios.

Servidor (backend):

- Responde a las solicitudes proporcionando el servicio.

4.1.2.3. Fase 3: Diseño detallado del sistema.

El presente proyecto, una aplicación web desarrollada con Flask y PostgreSQL, se estructura en tres capas esenciales: la capa de presentación, construida con tecnologías web como HTML, CSS y JavaScript, que proporciona la interfaz para la interacción con el usuario; la capa del servidor, donde Flask se encarga de implementar una API REST para la gestión de solicitudes y la ejecución de algoritmos de machine learning destinados a clasificar proyectos de investigación; y la capa del servidor de base de datos, que utiliza PostgreSQL para almacenar y recuperar datos relevantes para el análisis y la correlación de los proyectos.

4.1.2.1. Diagramas de caso de uso.

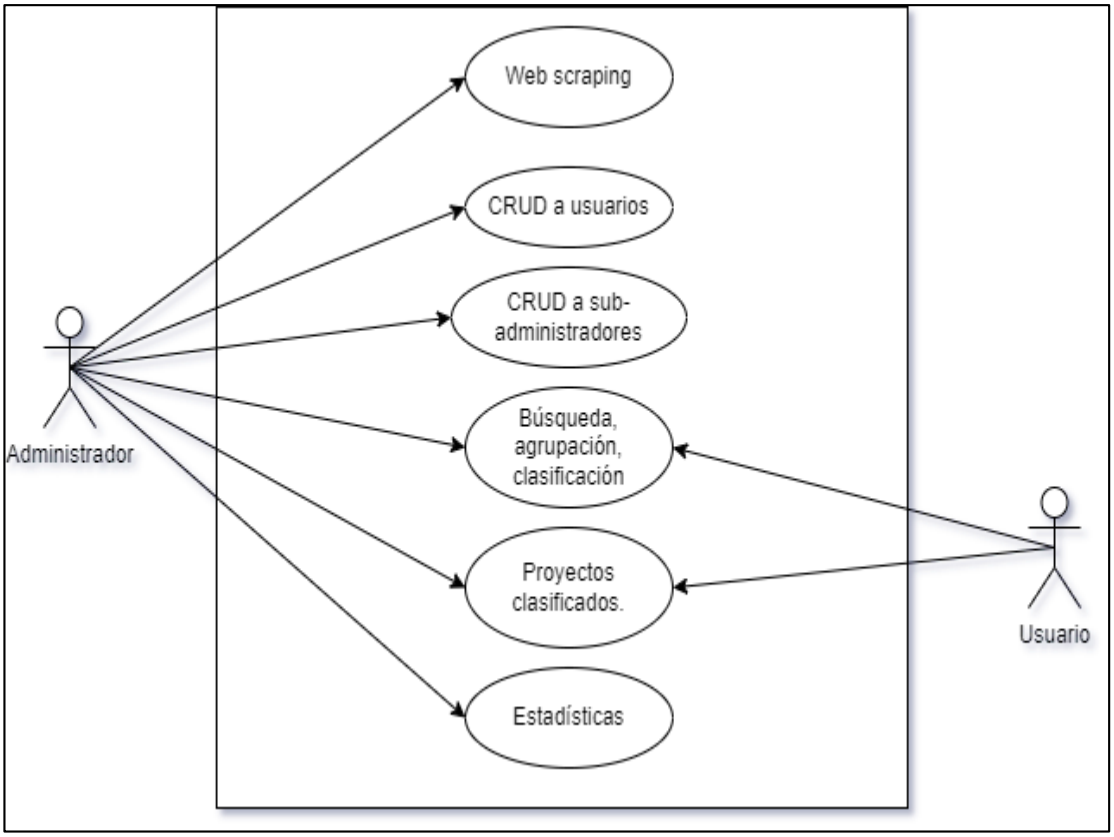


Figura 14. Caso de uso general del sistema

Tabla 13. Caso de uso General del sistema

Caso de uso	Diagrama del sistema general
Actor	Administrador - Usuario
Descripción	Se indica el funcionamiento principal del sistema general
Precondición	Para acceder al sistema, se requiere disponer de un navegador web y estar conectado a internet.
Actividades	• Web scraping • CRUD al usuario • CRUD a los sub-administradores • Búsqueda, agrupación y clasificación • El usuario visualiza los proyectos clasificados • El administrador visualiza las estadísticas

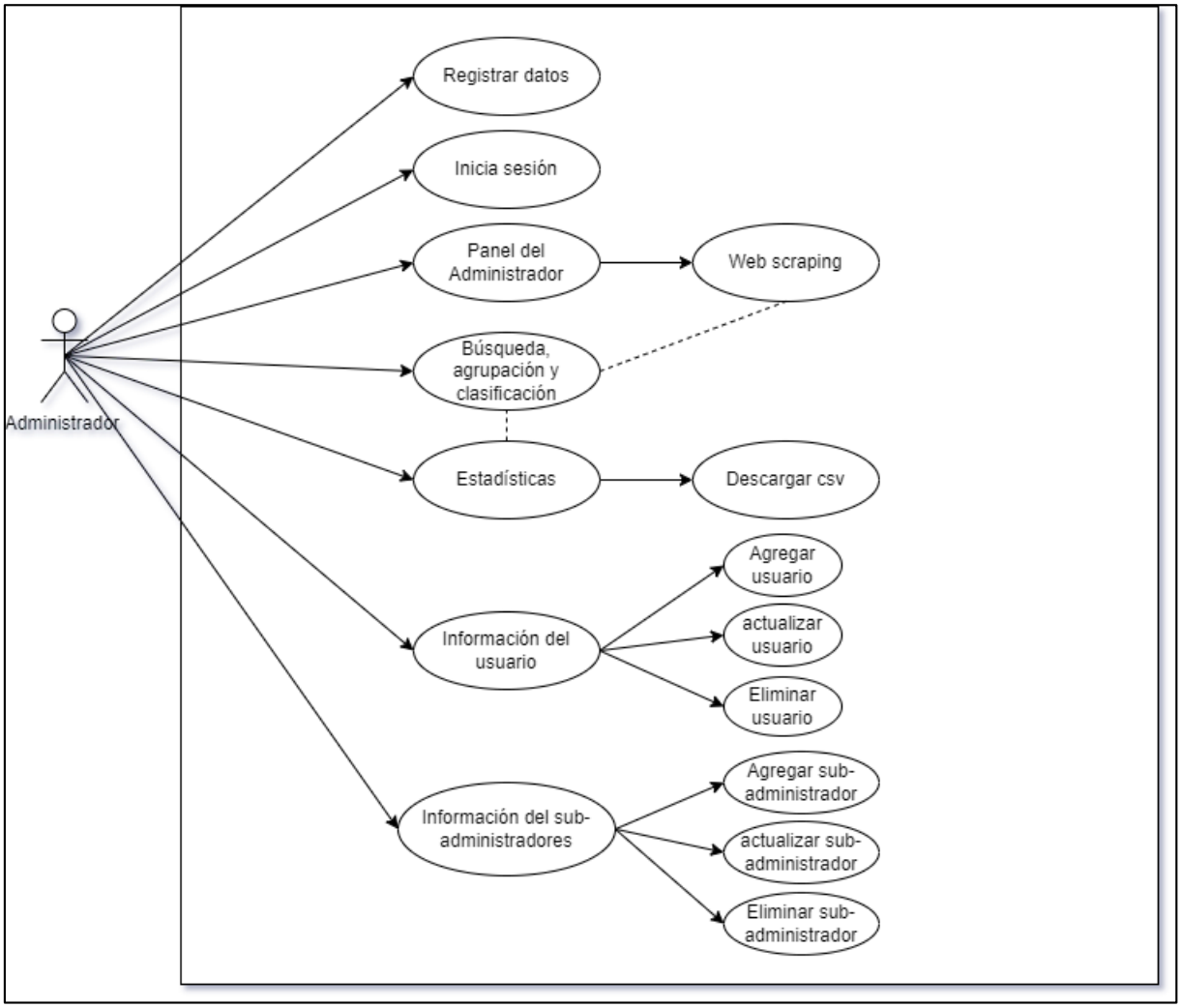


Figura 15. Caso de uso de la aplicación web para el administrador

Tabla 14. Descripción del caso de uso de la aplicación para el administrador

Caso de uso	Diagrama de la aplicación para el administrador
Actor	Usuario
Descripción	El administrador será responsable de supervisar y controlar todas las operaciones del sistema en caso de que surjan dificultades, siendo su responsabilidad principal encontrar soluciones a dichos problemas.
Precondición	Para acceder a este servicio, se requiere disponer de un navegador web y estar conectado a internet.
Actividades	• Registrar datos • Iniciar sesión • Web scraping a repositorios universitarios • Búsqueda, agrupación y clasificación a los proyectos de investigación • Descargar las estadísticas • CRUD al usuario • CRUD a los sub-administradores

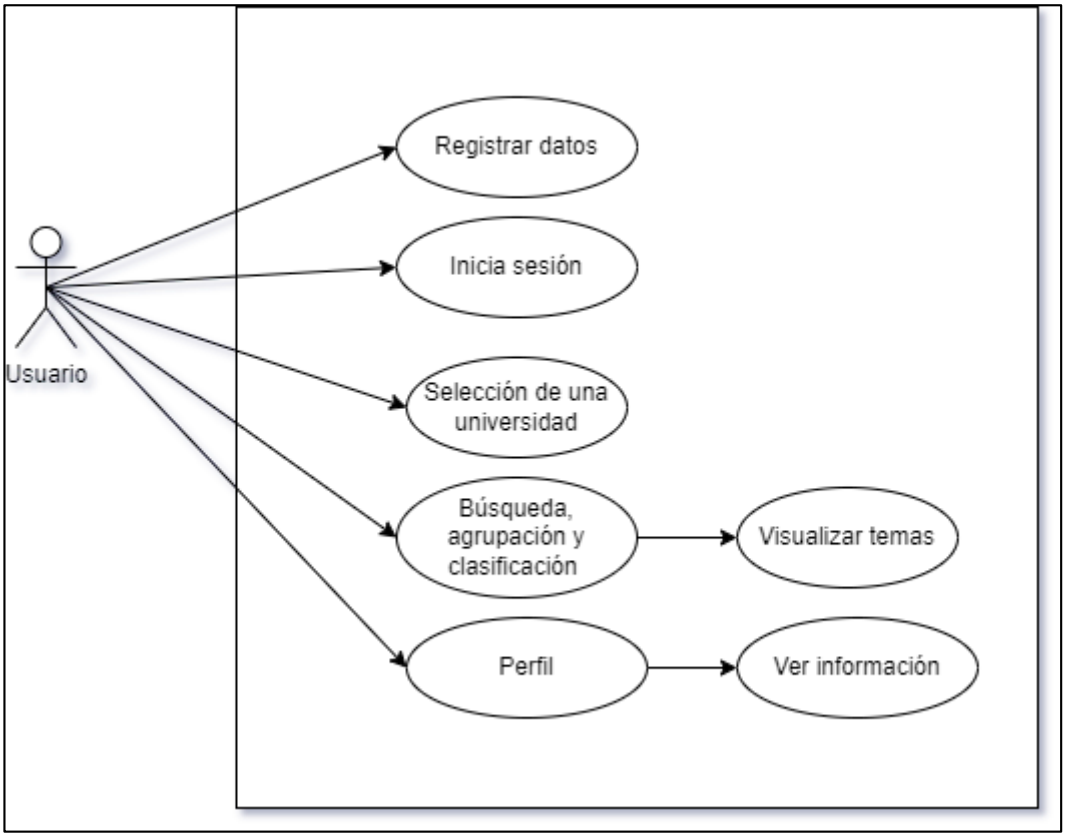


Figura 16. Caso de uso para el usuario

Tabla 15. Caso de uso de la aplicación para los usuarios

Caso de uso	Diagrama de la app para los usuarios
Actor	Usuario
Descripción	Para acceder a la aplicación, el usuario debe completar un proceso de registro inicial. Posteriormente, podrá iniciar sesión y acceder a los temas clasificados disponibles para su visualización.
Precondición	Para acceder a este servicio, se requiere disponer de un navegador web y estar conectado a internet.
Actividades	• Registrar datos • Iniciar sesión • Selección de una universidad • Visualizar temas clasificados • Ver perfil

Sistema para el usuario:

Para iniciar sesión, es necesario que previamente te hayas registrado como usuario. Posteriormente, deberás ingresar tus credenciales, tales como tu nombre de usuario

y contraseña como se muestra en la figura 11. De esta manera, accederás al sistema.

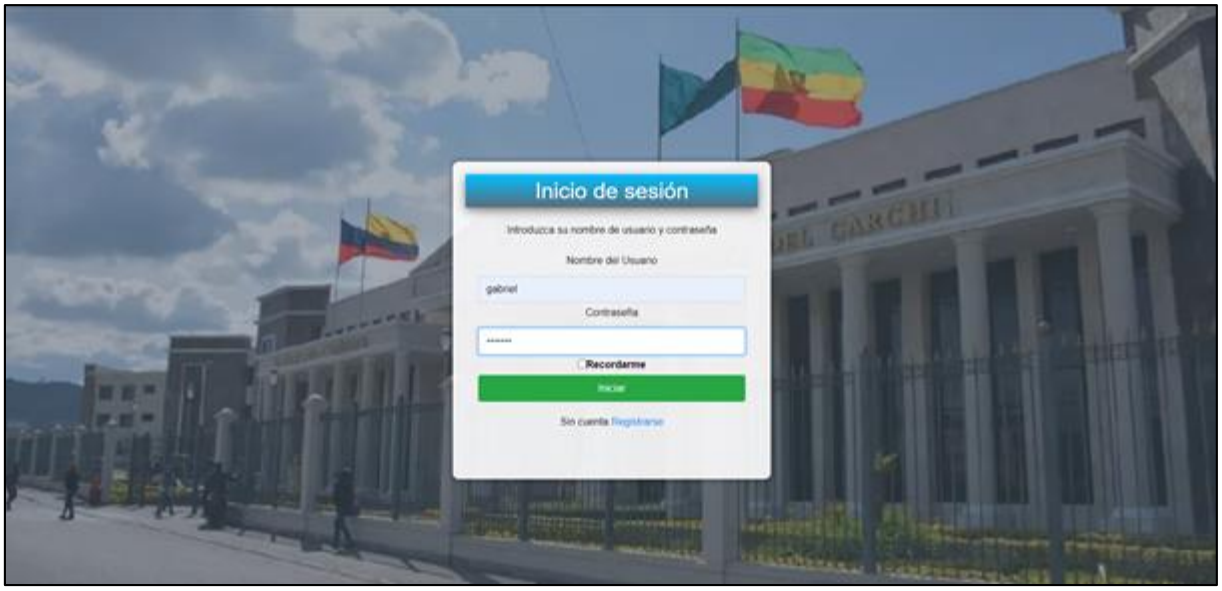


Figura 17. Pantalla de Inicio de sesión del usuario

Para acceder al sistema como usuario, es necesario que te registres previamente como se muestra en la figura 12. Los datos proporcionados durante el registro serán almacenados en la base de datos para su conservación. En futuras ocasiones, simplemente inicia sesión con los mismos datos de usuario y contraseña que registraste anteriormente.

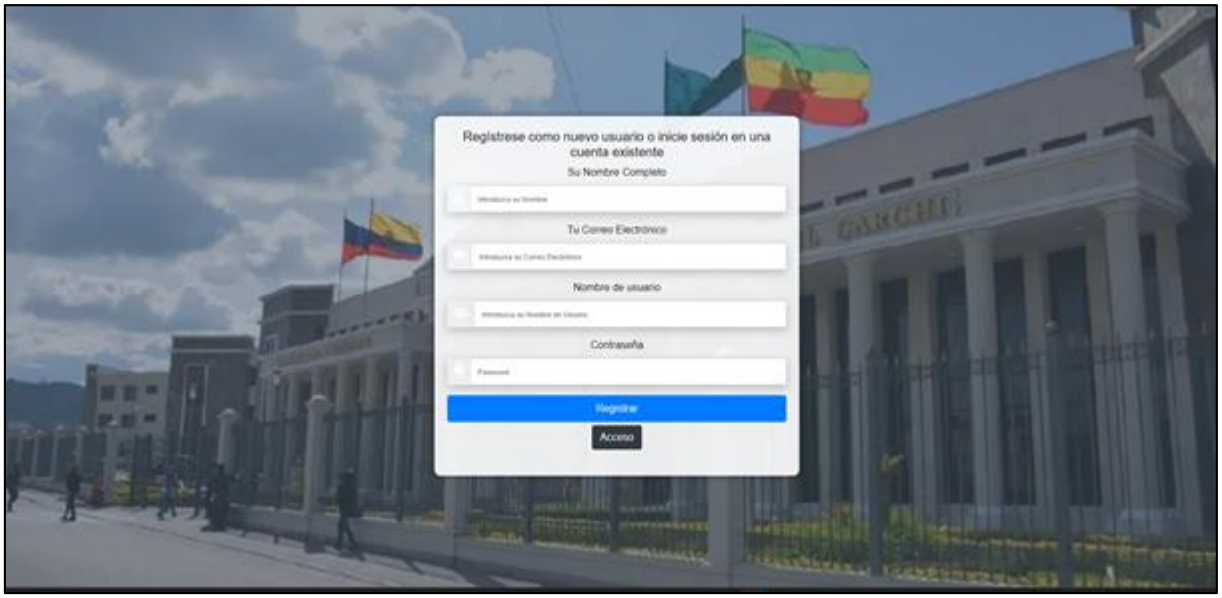


Figura 18. Pantalla de registro del usuario

Una vez que accede al sistema aparece la pantalla como se muestra en la figura 13, donde se puede escoger en la lista desplegable una universidad de las 6 disponibles como se muestra en la figura 14, además de en la barra de búsqueda colocar las palabras a buscar y una vez que damos en buscar se puede visualizar los proyectos de investigación relacionados con las palabras que se buscó como se muestra en la figura 15.

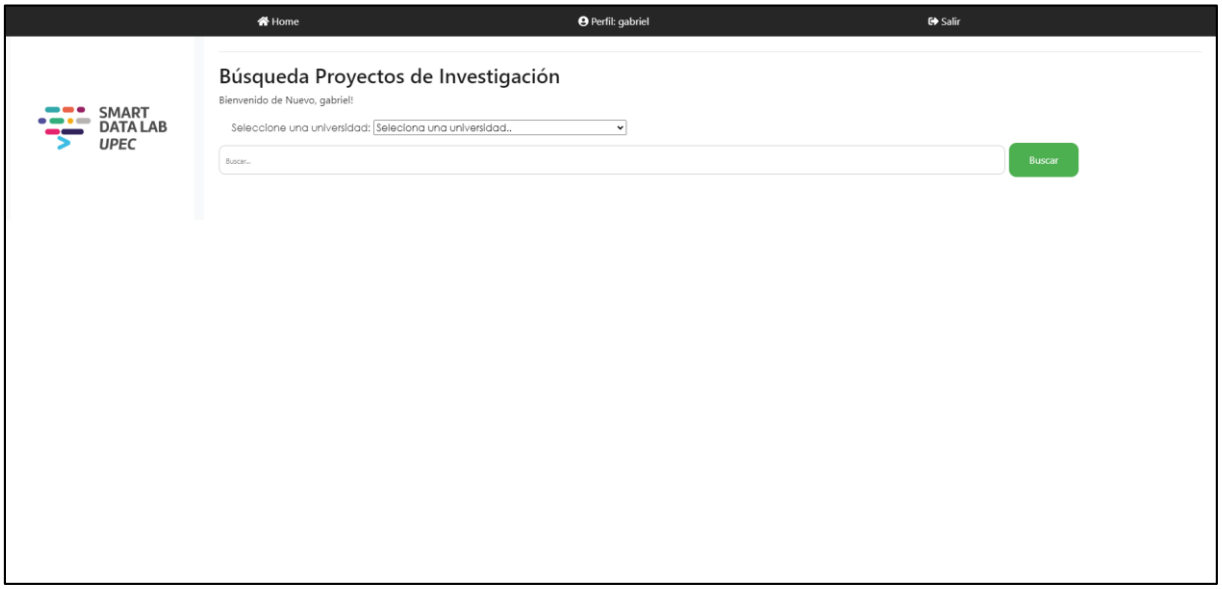


Figura 19. Pantalla de home del usuario

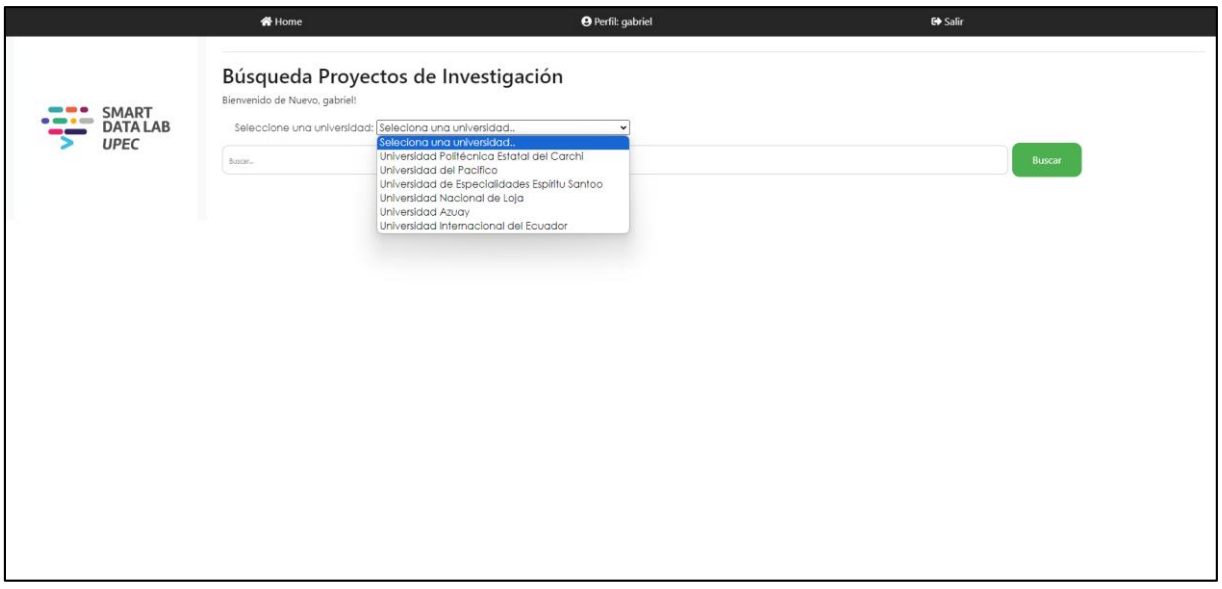


Figura 20. Pantalla Home, lista desplegable de universidades

En la pantalla de inicio del usuario, tras realizar la búsqueda con las palabras clave, se muestran los proyectos relacionados con el tema, número de proyecto, fecha de

publicación, autor y enlace de referencia, tal como se ilustra en la figura 15. Además, hay un área donde se visualizan burbujas de enfoque. Al pasar el mouse sobre ellas, se muestra el tema al que pertenecen; al hacer clic en ellas, se redirige al repositorio de la universidad donde se encuentra almacenado el PDF correspondiente al proyecto.

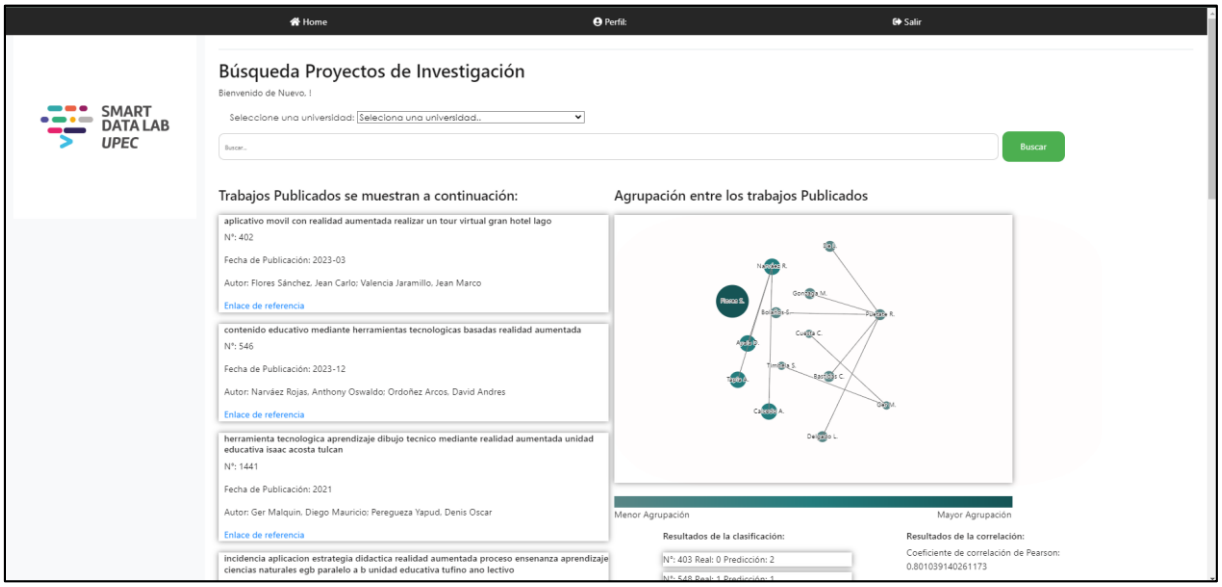


Figura 21. Pantalla donde se muestran los resultados de búsqueda

En la pantalla de perfil correspondiente al usuario, se puede visualizar la información como el nombre completo, nombre de usuario y correo electrónico con la que se registró para entrar al sistema como se muestra la figura 16.

Figura 22. Pantalla de perfil del usuario

La pantalla home correspondiente al administrador, puede realizar web scraping o raspado web a repositorios universitarios para extraer la información de los mismos con la siguiente información: fecha, tema, enlace y autores como se muestra en la figura 18, además hay la opción una vez hecho el web scraping se puede subir esos resultados a nuestro sistema de gestión de bases de datos en este caso es PostgreSQL.

También hay la opción de empezar de nuevo que puede cancelar o iniciar una nueva petición de raspado web.

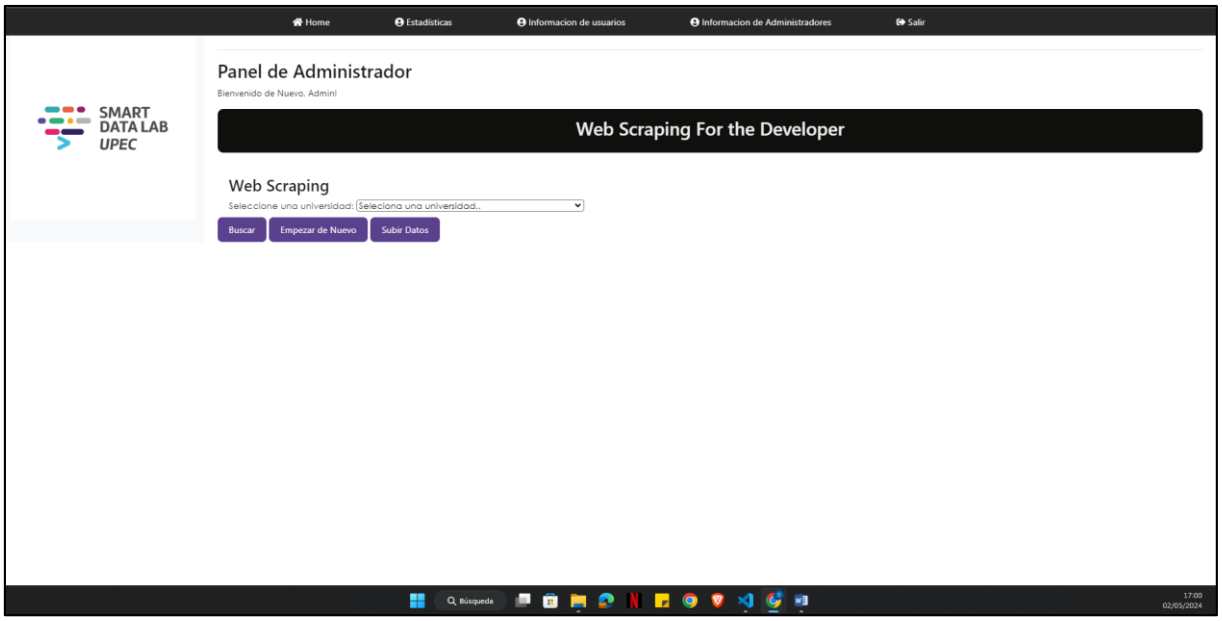
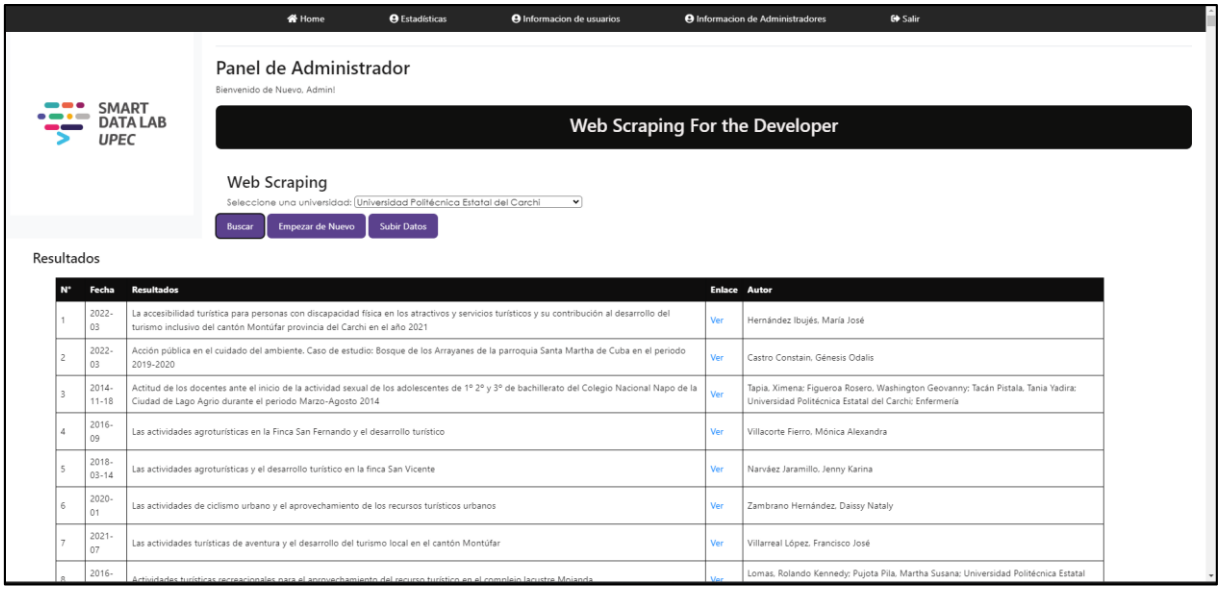


Figura 23. Pantalla del home de administrador



N°	Fecha	Resultados	Enlace	Autor
1	2022-03	La accesibilidad turística para personas con discapacidad física en los atractivos y servicios turísticos y su contribución al desarrollo del turismo inclusivo del cantón Montúfar provincia del Carchi en el año 2021	Ver	Hernández Ibujés, María José
2	2022-03	Acción pública en el cuidado del ambiente. Caso de estudio: Bosque de los Arrayanes de la parroquia Santa Martha de Cuba en el periodo 2019-2020	Ver	Castro Constain, Génesis Odalis
3	2014-11-18	Actitud de los docentes ante el inicio de la actividad sexual de los adolescentes de 1° 2° y 3° de bachillerato del Colegio Nacional Napo de la Ciudad de Lago Agrio durante el periodo Marzo-Agosto 2014	Ver	Tapia, Ximena; Figueroa Rosero, Washington Geovanny; Tacán Pistala, Tania Yadira; Universidad Politécnica Estatal del Carchi; Enfermería
4	2016-09	Las actividades agroturísticas en la Finca San Fernando y el desarrollo turístico	Ver	Villacorte Fierro, Mónica Alexandra
5	2018-03-14	Las actividades agroturísticas y el desarrollo turístico en la finca San Vicente	Ver	Narváez Jaramillo, Jenny Karina
6	2020-01	Las actividades de ciclismo urbano y el aprovechamiento de los recursos turísticos urbanos	Ver	Zambrano Hernández, Daissy Nataly
7	2021-07	Las actividades turísticas de aventura y el desarrollo del turismo local en el cantón Montúfar	Ver	Villarreal López, Francisco José
8	2016-	Actividades turísticas recreacionales para el aprovechamiento del recurso turístico en el complejo lacustre Mojanda	Ver	Lomas, Rolando Kennedy; Pujota Pila, Martha Susana; Universidad Politécnica Estatal

Figura 24. Pantalla Home, web scraping al repositorio de la UPEC

En la sección de estadísticas de la pantalla del administrador, se presenta un resumen detallado de las búsquedas realizadas por los usuarios. Los datos almacenados incluyen la universidad a la que pertenecen, el nombre de usuario, las palabras clave buscadas y la fecha en que se realizó la búsqueda. Además, esta información puede actualizarse y descargarse en formato CSV para su análisis posterior. Asimismo, se proporciona un resumen general de las búsquedas, el cual también puede descargarse en formato CSV. Todos estos elementos están representados en la figura

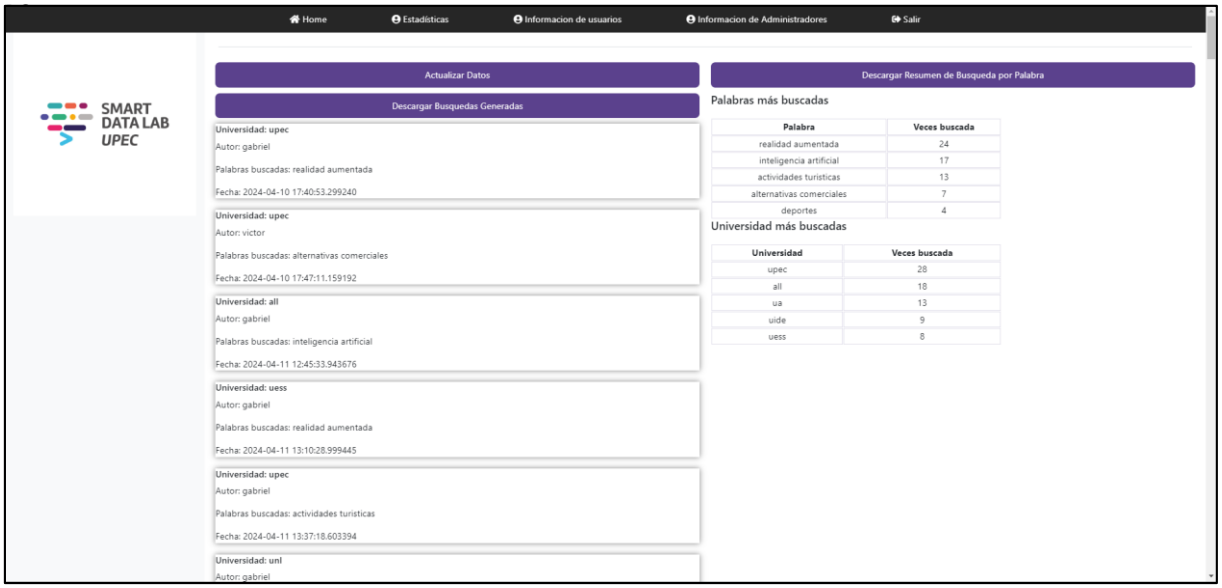


Figura 25. Pantalla de estadísticas del administrador

En la sección de información de usuarios el administrador puede ver, agregar, actualizar y eliminar los usuarios que se registraron como se muestra en la figura 20.

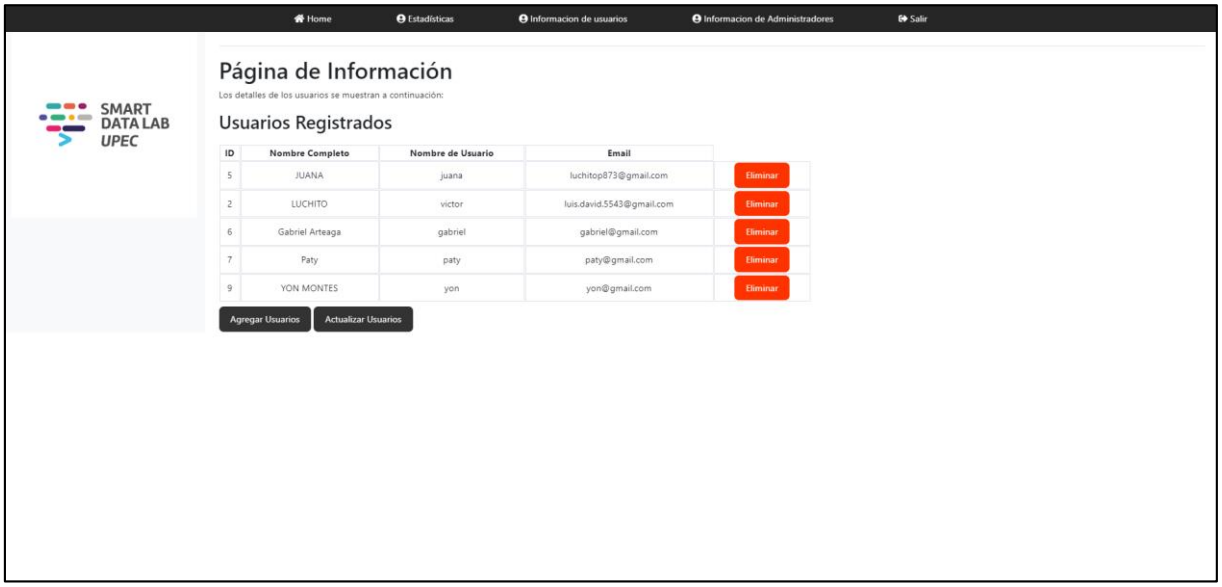


Figura 26. Pantalla de información de usuarios

En la sección de información de sub-administradores el administrador puede ver, agregar, actualizar y eliminar los sub-administradores que se registraron directamente en la base de datos o el administrador los agrega como se muestra en la figura 21.

Figura 27. Pantalla de información de sub-administradores

4.2. DISCUSIÓN

A partir de la investigación realizada, se ha creado una solución tecnológica basada en la web para la búsqueda, agrupación y clasificación de proyectos de investigación, esta solución se presenta mediante una aplicación web. Se da a conocer que, específicamente mediante algoritmos de aprendizaje automático (machine leanring), se puede lograr un alto nivel de precisión en la clasificación de proyectos de investigación. Al evaluar diferentes algoritmos, se obtuvo un alcance del 95%, lo cual coincide con Valero et al. (2022), quienes señalan que un algoritmo con una precisión mayor del 90% es funcional y útil.

La revisión de diferentes algoritmos de aprendizaje automático ha facilitado una comprensión del tema, lo que fue esencial para realizar el trabajo de integración curricular con bases evaluativas, siguiendo pautas y directrices para escoger un algoritmo capaz de clasificar proyectos de investigación. Esto coincide con la investigación de Rojas y Vargas (2022), donde menciona la importancia de parámetros para la evaluación como lo es la precision, recall, f1- score, de igual forma la opinión de Hernández (2022), donde se enfoca en la comparación y evaluación de diversos algoritmos de machine learning para predecir el rendimiento académico, donde mencionan la importancia de la evaluación para identificar un algoritmo que sea preciso, estable y efectivo.

En este contexto, resulta relevante resaltar una aplicación de una metodología para la preparación de datos, lo que respalda la mejora de la compresión y tratamiento de los datos, proporcionando una interpretación más sólida de los resultados y una clasificación más viable, esto concuerda con el estudio de Hernández (2022), donde aplica la metodología CRISP-DM para el procesamiento de datos y así permitir que las computadoras procesen y analicen grandes cantidades de datos.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Los objetivos establecidos se logran al recopilar información de diversos repositorios, documentos y antecedentes con el fin de elaborar un estado del arte. Esta información es utilizada para realizar comparaciones en términos de conceptos, aplicaciones, metodologías, características y el desarrollo del software.
- Aplicando diversos algoritmos de machine learning, como la Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, KNN y red neuronal artificial, cada uno con sus ventajas según la complejidad de los datos y la naturaleza de las relaciones que se analizaron.
- La precisión de los algoritmos de machine learning está influenciada por varios factores clave, incluida la calidad y cantidad de los datos de entrenamiento, la selección adecuada de características relevantes, el tipo de algoritmo utilizado, así como la aplicación de técnicas de regularización y procesamiento de datos.
- Flask es un framework que impulsa el desarrollo web en Python, caracterizado por su ligereza y versatilidad, proporciona una plataforma ágil para la creación de aplicaciones web, permitiendo la libertad de diseñar soluciones dinámicas y escalables. Al aprovechar Python y tecnologías web estándar como HTML, CSS y JavaScript, ofrece un entorno flexible y poderoso para construir aplicaciones web modernas.
- Con la asistencia de una aplicación web en el SMART DATA LAB de la UPEC, se facilita la exploración de nuevos ámbitos de investigación para estudiantes, docentes y cualquier persona interesada en encontrar proyectos de investigación afines a sus intereses temáticos. Esta herramienta ofrece la posibilidad de descubrir y acceder a proyectos de investigación relevantes, permitiendo así un proceso más eficiente en la búsqueda de información académica y científica.

5.2. RECOMENDACIONES

- La investigación es esencial para el desarrollo de cualquier tipo de proyecto. Por ello, es aconsejable utilizar fuentes primarias, revistas científicas y artículos académicos que ayuden a profundizar en el tema en cuestión, asegurando así el logro de los objetivos del proyecto.
- La solución tecnológica web presentada en la investigación curricular ha sido desarrollada utilizando la metodología CRISP-DM. Para obtener un producto final de calidad, se recomienda seguir los pasos establecidos por esta metodología, garantizando así el cumplimiento de los objetivos del proyecto.
- En el desarrollo de proyectos como el presentado en esta investigación, es aconsejable investigar y aprender el lenguaje de programación que se va a utilizar, así como las herramientas que ofrece, como los widgets para la interfaz de usuario.
- Es necesario ampliar la data mediante web scraping a más repositorios universitarios de tal manera que se obtengan más temas de proyectos de investigación para su posterior procesamiento.

VI. REFERENCIAS BIBLIOGRÁFICAS

Bastis Consultores . (8 de noviembre de 2021). *ONLINE-TESIS*. Obtenido de ONLINE-TESIS: https://online-tesis.com/metodo-hipotetico-deductivo/

Redacción APD. (04 de abril de 2019). *apd*. Obtenido de apd: https://www.apd.es/algoritmos-del-machine-learning/

ANEP. (2023). *ANEP*. Obtenido de ANEP: https://www.anep.edu.uy/sites/default/files/images/te-programas/2023/finales/espacios/espacio-tecnico-tecnologico/Ciencias%20de%20la%20Computaci%C3%B3n%20-%20Tramo%205.pdf

Arias, L. M. (8 de noviembre de 2023). *linkedin*. Obtenido de linkedin: https://www.linkedin.com/pulse/metodolog%C3%ADa-crisp-dm-la-gu%C3%ADa-definitiva-para-miner%C3%ADa-de-arias-xyusf/

Benites, L. (24 de marzo de 2022). *statologos*. Obtenido de statologos: https://statologos.com/kendalls-tau-2/

Burns , E., & lutkevinch, B. (febrero de 2024). *computerweekly*. Obtenido de computerweekly: https://www.computerweekly.com/es/definicion/Procesamiento-de-lenguaje-natural-o-NLP

Cabrera Palacios , A. P., & Chiluiza Luna , B. M. (2018). *UNIVERSIDAD POLITÉCNICA SALECIANA SEDE CUENCA*. Obtenido de UNIVERSIDAD POLITÉCNICA SALECIANA SEDE CUENCA: https://dspace.ups.edu.ec/bitstream/123456789/16584/4/UPS-CT008041.pdf

CAF. (2022). *Conceptos fundamentales y uso responsable de la Inteligencia Artificial en el sector público. Informe2.* Colombia: CAF.

Caparrini, F. S. (20 de diciembre de 2020). *Fernando Sancho Caparrini*. Obtenido de Fernando Sancho Caparrini: http://www.cs.us.es/~fsancho/?e=230

Cardona, L. (27 de diciembre de 2021). *Cyberclick*. Obtenido de Cyberclick: https://www.cyberclick.es/que-es/rastreadores-web-crawler-o-aranaClarín. (24 de octubre de 2020). *Clarín*. Obtenido de Clarín: https://www.clarin.com/cultura/que-es-el-metodo-inductivo-significado-pasos-y-ejemplos_0_6AL5shQEw.html

Coca Bergolla, Y., & Llivina Lavigne, M. (noviembre de 2021). *unesco*. Obtenido de unesco:https://en.unesco.org/sites/default/files/l2_la_ia_como_una_ciencia_de_la_computacion_0.pdf

Consejo de Educación Superior . (2024). *Consejo de Educación Superior* . Obtenido de Consejo de Educación Superior : https://www.educacionsuperior.gob.ec/

Consultores Bastis. (21 de mayo de 2020). *ONLINE-TESIS*. Obtenido de ONLINE-TESIS: https://online-tesis.com/datos-secundarios-investigacion/

DataScientest. (2021 de diciembre de 2021). *DataScientest*. Obtenido de DataScientest: https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos

DataScientest. (7 de marzo de 2022). *DataScientest*. Obtenido de DataScientest: https://datascientest.com/es/text-mining-o-mineria-de-textos-definicion-tecnicas-casos-de-uso

DataScientest. (1 de septiembre de 2022). *DataScientest*. Obtenido de DataScientest: https://datascientest.com/es/beautiful-soup-aprender-web-scraping

DATLAS. (28 de junio de 2020). *DATLAS*. Obtenido de DATLAS: https://blogdatlas.wordpress.com/2020/06/28/algoritmos-supervisados-clasificacion-vs-regresion-datlas-research/

ER, S. (26 de abril de 2023). *Analytics Vidhya*. Obtenido de Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

García, L. A. (5 de septiembre de 2020). *KaBel*. Obtenido de KaBel: https://www.kabel.es/aprendizaje-refuerzos/

Gomez, I. (16 de febrero de 2022). *crehana.* Obtenido de crehana: https://www.crehana.com/blog/negocios/text-mining/

González, L. (3 de mayo de 2022). *aprendeia*. Obtenido de aprendeia: https://aprendeia.com/herramientas-de-python-para-machine-learning/

Heras, J. M. (29 de septiembre de 2020). *IArtificial.net*. Obtenido de IArtificial.net: https://www.iartificial.net/clasificacion-o-regresion/

Hernández, D. J. (septiembre de 2020). *Repositorio de la Universidad del BIO-BIO*. Obtenido de Repositorio de la Universidad del BIO-BIO: http://repobib.ubiobio.cl/jspui/bitstream/123456789/3723/1/Reyes_Hern%C3%A1ndez_Diego_Joaqu%C3%ADn.pdf

IBM Cloud Education. (17 de agosto de 2020). *BM Cloud Learn Hub*. Obtenido de BM Cloud Learn Hub: https://www.ibm.com/mx-es/cloud/learn/neural-networks

IBM Cloud Education. (15 de Julio de 2020). *IBM.* Obtenido de IBM: https://www.ibm.com/es-es/cloud/learn/machine-learning

Kaur, S. S. (junio de 2021). *UMH.* Obtenido de UMH: http://dspace.umh.es/bitstream/11000/26781/1/SinghKaur_Sukhwinder.pdf

Lifeder. (21 de octubre de 2022). *Lifeder.* Obtenido de Lifeder: https://www.lifeder.com/investigacion-correlacional/

MERINO, M. (27 de enero de 2019). *xataka*. Obtenido de xataka: https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo

Morales Hernández, M. Á., González Camacho, J. M., Robles Vásquez, H., del Valle Paniagua, D., & Durán Moreno, J. R. (24 de enero - junio de 2022). Algoritmos de aprendizaje automático para la predicción dellogro académico. *Ride*, 28.

Obtenido de Ride: https://www.ride.org.mx/index.php/RIDE/article/view/1180/3517

Ortega, C. (2022). *QuestionPro*. Obtenido de QuestionPro: https://www.questionpro.com/blog/es/coeficiente-de-correlacion-de-pearson/

Parra, A. (2022). *QuestionPro*. Obtenido de QuestionPro: https://www.questionpro.com/blog/es/coeficiente-de-correlacion-de-spearman/

RAMÍREZ, L. (5 de enero de 2023). *IEBS*. Obtenido de IEBS: https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-funciona-big-data/#:~:text=clusters%20K%2Dmeans-%C2%BFQu%C3%A9%20es%20K%2D%20means%20clustering%3F,los%20puntos%20en%20otros%20clusters.

Ray, S. (27 de abril de 2023). *Analytics Vidhya*. Obtenido de Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

RECFACES. (11 de agosto de 2021). *RECFACES*. Obtenido de RECFACES: https://recfaces.com/es/articles/reconocimiento-de-patrones#1

Rico Páez, A., & Gaytán Ramírez, N. D. (2022). Modelos predictivos del rendimiento académico. *REDIECH*, 19.

Rodó, P. (1 de mayo de 2020). *economipedia*. Obtenido de economipedia: https://economipedia.com/definiciones/valor-p.html

Rojas Gonzáles, C. D., & Vargas Gonzáles, Y. M. (2022). *Repositorio Institucional Universidad Católica de Colombia.* Obtenido de Repositorio Institucional Universidad Católica de Colombia: https://repository.ucatolica.edu.co/server/api/core/bitstreams/853a3b3d-bf5f-463f-948b-702c372ccfa6/content

Selenium. (2024). *Selenium*. Obtenido de Selenium: https://www.selenium.dev/

Tokio. (01 de 09 de 2023). *Tokio School.* Obtenido de Tokio School: https://www.tokioschool.com/noticias/c-que-es/

Tokio School. (16 de enero de 2021). *Tokio*. Obtenido de Tokio: https://www.tokioschool.com/noticias/redes-neuronales-machine-mearning/

Tuychiev, B. (27 de diciembre de 2023). *datacamp*. Obtenido de datacamp: https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm

UNIR. (7 de mayo de 2021). *UNIR* . Obtenido de UNIR: https://www.unir.net/ingenieria/revista/arboles-de-decision/

UNIR. (31 de agosto de 2021). *UNIR* . Obtenido de UNIR: https://www.unir.net/ingenieria/revista/postgre-sql/

Universidad Europea. (9 de septiembre de 2022). *Universidad Europea*. Obtenido de Universidad Europea: https://universidadeuropea.com/blog/aprendizaje-supervisado-no-supervisado/

Valero Cajahuanca, J. E., Navarro Raymundo, Á. F., Larios Franco, A. C., & Julca Flores, J. D. (2022). Evaluación de diferentes algoritmos de Machine Learning para su predicción. *Revista de Ciencias Sociales (Ve)*, 12.

Vega, J. B. (2023). *bookdown.org*. Obtenido de bookdown.org: https://bookdown.org/jboscomendoza/r-principiantes4/

Velázquez, A. (2022). *QuestionPro*. Obtenido de QuestionPro: https://www.questionpro.com/blog/es/investigacion-experimental/

Vlad, K., & Leigh, J. (febrero de 2022). *ReserchGate*. Obtenido de ReserchGate: https://www.researchgate.net/publication/352014123_Legality_and_Ethics_of_Web_Scraping

Zamorano, I. V. (24 de agosto de 2020). *linkedin*. Obtenido de linkedin: https://www.linkedin.com/pulse/la-correlaci%C3%B3n-una-herramienta-inicial-para-toma-de-ismael/?originalSubdomain=es

VII. ANEXOS

Anexo 1. Acta de Predefensa.

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

ACTA

DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

ESTUDIANTE:	CISNEROS CAPLOSAMA LUIS DAVID	**CÉDULA DE IDENTIDAD:**	0401918057
PERIODO ACADÉMICO:	2024A		
PRESIDENTE TRIBUNAL	MSC. GEORGINA GUADALUPE ARCOS PONCE	**DOCENTE TUTOR:**	MSC. SAMUEL BENJAMIN LASCANO RIVERA
DOCENTE:	MSC. JEFFERY ALEX NARANJO CEDEÑO		
TEMA DEL TIC:	"Algoritmos de machine learning para la correlación de proyectos de investigación"		

No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	9,17	
2	FUNDAMENTACIÓN TEÓRICA	9,17	
3	METODOLOGÍA	9,17	
4	RESULTADOS	9,17	Revisar interfaz gráfica, acuerdo de confidencialidad, publicar el sistema
5	DISCUSIÓN	9,17	
6	CONCLUSIONES Y RECOMENDACIONES	9,17	revisar las conclusiones y recomendaciones
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	9,17	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	9,00	revisión normas APA

Obteniendo una nota de: 9,12 Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el martes, 9 de julio de 2024

MSC. GEORGINA GUADALUPE ARCOS PONCE
PRESIDENTE TRIBUNAL

MSC. SAMUEL BENJAMIN LASCANO RIVERA
DOCENTE TUTOR

MSC. JEFFERY ALEX NARANJO CEDEÑO
DOCENTE

Anexo 2. Informe de Abstract

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN AND NATIVE LANGUAGE CENTER

Informe sobre el Abstract de Artículo Científico o Investigación.

Autor: Luis David Cisneros Carlosama.
Fecha de recepción del abstract: 15 de julio de 2024
Fecha de entrega del informe: 15 de julio de 2024

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según los rubrics de evaluación de la traducción en Inglés, ésta alcanza un valor de 9, por lo cual se validad dicho trabajo.

Atentamente

Firmado electrónicamente por:
EDISON BOANERGES PENAFIEL ARCOS

Ing. Edison Peñafiel Arcos MSc
Coordinador del CIDEN

Anexo 3. Carta de incorporación

SMART DATA LAB UPEC

Tulcán, 30 de marzo de 2023

Msc. Carlitos Guano

DIRECTOR DE LA CARRERA DE COMPUTACIÓN

Presente. -

De mi consideración:

Por medio del presente se deja constancia de la necesidad que tiene el proyecto de investigación SMART DATA LAB en la creación de "Algoritmos de machine learning para la correlación de proyectos de investigación" el cual debe ser desarrollado en el periodo académico PAO 2023 A, en tal sentido el estudiante Luis David Cisneros Carlosama con número de cedula 0401918057 se ha comprometido a realizar el presente proyecto de tesis, es lo que puedo manifestar en honor a la verdad.

Atentamente,

Msc. Samuel Lascano

Anexo 4. Carta Compromiso

Universidad Politécnica Estatal del Carchi
Educamos para transformar el mundo

Tulcán, 30 de marzo de 2023

Msc. Samuel Lascano Rivera

DIRECTOR DEL PROYECTO

Smart Data Lab

Presente. -

De mi consideración:

Por la presente, agradezco la confianza depositada en mi persona para poder realizar la creación de "Algoritmos de machine learning para la correlación de proyectos de investigación" el cual debe ser desarrollado en el periodo académico PAO 2023 A. a la vez expreso el compromiso de conocer las responsabilidades que implica este proyecto de investigación.

Por lo que me comprometo a:

- *Respetar y cumplir las normativas internas del laboratorio.*
- *No divulgar la información de la institución por ningún tipo o mecanismo.*
- *Guardar el decoro adecuado de mi comportamiento en el interior del laboratorio y durante el tiempo que dure el proyecto de investigación.*
- *Llevar durante el proyecto el adecuado uso de vestimenta acorde con la formalidad que requiere este laboratorio.*
- *Utilizar los recursos de la institución exclusivamente para las tareas encomendadas a mi persona.*
- *Asistir puntalmente a las horas programadas para trabajar en el proyecto.*

Además, dejo constancia que:

- *Comprendo que este proyecto de investigación no involucra ningún compromiso de contrato laborar durante o después de haber culminado el proceso del proyecto a realizar.*
- *Comprendo que este proyecto de investigación, no genera obligaciones patronales de ningún tipo y por lo tanto no puedo reclamarlas de ninguna forma o bajo ningún argumento.*

Atentamente,

Luis David Cisneros Carlosama

C.I. 0401918057

ESTUDIANTE DE LA CARRERA DE CIENCIAS EN LA COMPUTACIÓN

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

Especificación de requerimientos de software

A continuación, se examina la función de implementación del sistema propuesto en esta investigación. Estos requisitos permiten describir las distintas funciones de la aplicación web para satisfacer las necesidades de los usuarios. Las funciones establecidas aquí serán minuciosamente analizadas para garantizar su diseño y funcionamiento óptimo.

Para que el sistema propuesto pueda identificar a los usuarios, debe incluir la función de registro a través de la aplicación web. Este requisito es de gran importancia, ya que permite a los usuarios registrarse y luego identificarse para acceder al sistema

Requerimientos del administrador:

Tabla 16. Requerimiento funcional 01 - página de inicio

RF_01	Página de inicio
DESCRIPCIÓN	El administrador puede iniciar sesión al ejecutar la aplicación por primera vez.
OBJETIVO	Permitir que los administradores inicien sesión en la aplicación.
PRIORIDAD	Alta
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación guarda los datos proporcionados por el administrador y lleva a cabo su verificación.

Tabla 17. Requerimiento funcional 02 - inicio de sesión

RF_02	inicio de sesión
DESCRIPCIÓN	Permite al administrador iniciar sesión ingresando los datos en el formulario, los cuales han sido verificados por PostgresSQL.
OBJETIVO	Permitir el ingreso a la aplicación
PRIORIDAD	Alta
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación verifica los datos y permite el acceso.

Tabla 18. Requerimiento funcional 03 - pantalla principal del administrador (homeadmin)

RF_04	homeadmin
DESCRIPCIÓN	Le permite al administrador realizar web scraping a repositorios universitarios, así como subir la información extraída a la base de datos PostgresSQL.
OBJETIVO	Permitir al administrador realizar un raspado web.
PRIORIDAD	Alta
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación muestra la información extraída y subirla a la base para su posterior uso.

Tabla 19. Requerimiento funcional 04 - añadir cuenta del usuario

RF_05	añadir cuenta del usuario
DESCRIPCIÓN	Permite al administrador crear una cuenta de usuario.
OBJETIVO	Realizar el proceso para la creación de una cuenta
PRIORIDAD	Alta
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	El administrador agrega los detalles relacionados con la cuenta.

Tabla 20. Requerimiento funcional 05 - Borrar cuenta del usuario

RF_06	borrar cuenta del usuario
DESCRIPCIÓN	Permite al administrador borrar una cuenta de usuario.
OBJETIVO	Realizar el proceso de borrar cuenta
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	El administrador puede seleccionar los distintos usuarios según lo requiera.

Tabla 21. Requerimiento funcional 06 - Modificar cuenta del usuario

RF_07	modificar cuenta del usuario
DESCRIPCIÓN	Permite al administrador modificar una cuenta de usuario según sea necesario.
OBJETIVO	Modificar la información de las cuentas.
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación proporciona al administrador la capacidad de efectuar modificaciones en las cuentas según sea necesario.

Tabla 22. Requerimiento funcional 08 - añadir cuenta de sub-administrador

RF_08	añadir cuenta de sub-administrador
DESCRIPCIÓN	Permite al administrador crear una cuenta de sub-administrador
OBJETIVO	Realizar el proceso para la creación de una cuenta
PRIORIDAD	Alta
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	El administrador agrega los detalles relacionados con la cuenta.

Tabla 23. Requerimiento funcional 09 - Borrar cuenta de sub-administrador

RF_09	borrar cuenta de sub-administrador
DESCRIPCIÓN	Permite al administrador borrar una cuenta de sub-administrador
OBJETIVO	Realizar el proceso de borrar cuenta
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	El administrador puede seleccionar los distintos sub-administrador según lo requiera.

Tabla 24. Requerimiento funcional 10 - Modificar cuenta de sub-administrador

RF_10	modificar cuenta de sub-administrador
DESCRIPCIÓN	Permite al administrador modificar una cuenta de sub-administrador según sea necesario.
OBJETIVO	Modificar la información de las cuentas.
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación proporciona al administrador la capacidad de efectuar modificaciones en las cuentas según sea necesario.

Tabla 25. Requerimiento funcional 11 – Pantalla de estadísticas.

RF_11	ESTADÍSTICAS
DESCRIPCIÓN	Esta pantalla permite visualizar las estadísticas generadas en relación a las búsquedas que hizo el usuario.
OBJETIVO	Permitir la visualización de la información de estadísticas.
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación muestra la información en relación a las estadísticas que se generaron a partir de las búsquedas hechas por el usuario, aquí hay las búsquedas generales y detalladas.

Tabla 26. Requerimiento funcional 12 – Descargar estadísticas generales

RF_12	Descargar estadísticas generales
DESCRIPCIÓN	Permite al administrador hacer una descarga general de las búsquedas que hizo el usuario.
OBJETIVO	Descargar la información de búsquedas generales.
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación permite al administrador descargar en un formato csv la información de las búsquedas generales hechas en la parte del usuario, información tal como las palabras que busco, autor, universidad y fecha.

Tabla 27. Requerimiento funcional 13 – Descargar estadísticas detalladas

RF_13	Descargar detalladas
DESCRIPCIÓN	Facilita al administrador hacer una descarga detallada de las búsquedas que hizo los usuarios.
OBJETIVO	Descargar la información de búsquedas detalladas.
PRIORIDAD	Media
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	La aplicación permite al administrador descargar en un formato csv la información de las búsquedas detalladas hechas en la parte del usuario, información tal como las palabras claves que más se buscaron y universidades más buscadas.

Tabla 28. Requerimiento funcional 14 - Finalizar sesión

RF_14	Finalizar sesión
DESCRIPCIÓN	El administrador accede a esta función a través de un botón que finaliza su sesión.
OBJETIVO	Facilitar al administrador la opción de cerrar sesión según sea necesario.
PRIORIDAD	Baja
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	El administrador cierra sesión y sale de la aplicación.

Requerimientos del usuario:

Tabla 29. Requerimiento funcional 15 - página de inicio

RF_15	Página de inicio
DESCRIPCIÓN	Si el usuario no ha utilizado la aplicación anteriormente, se le presentan la opción de iniciar sesión o registrarse.
OBJETIVO	Presentar al usuario información sobre el proceso de registro e inicio de sesión.
PRIORIDAD	Alta
ACOTORES	Usuario
FLUJO DE INFORMACIÓN	La aplicación verifica los datos proporcionados y concede el acceso correspondiente.

Tabla 30. Requerimiento funcional 16 - Registro de usuario.

RF_16	Registro de usuario
DESCRIPCIÓN	Esta pantalla facilita que el usuario se registre en la aplicación proporcionando sus datos en el formulario.
OBJETIVO	Permitir el registro de los usuarios.
PRIORIDAD	Alta
ACOTORES	Administrador
FLUJO DE INFORMACIÓN	Una vez ingresados los datos, PostgresSQL lleva a cabo la verificación y completa el registro..

Tabla 31. Requerimiento funcional 17 – Pantalla principal del usuario(home)

RF_17	home
DESCRIPCIÓN	Esta pantalla facilita que el usuario se hacer una búsqueda de proyectos de investigación.
OBJETIVO	Permitir la búsqueda de temas relevantes
PRIORIDAD	Alta
ACOTORES	Usuario
FLUJO DE INFORMACIÓN	La aplicación muestra la información de la búsqueda, agrupación y clasificación de proyectos de investigación en relación a las palabras que busca el usuario.

Tabla 32. Requerimiento funcional 18 – Pantalla perfil del usuario

RF_18	perfil
DESCRIPCIÓN	Esta pantalla permite ver los datos de usuario
OBJETIVO	Permitir la visualización de la información del usuario.
PRIORIDAD	Media
ACOTORES	Usuario
FLUJO DE INFORMACIÓN	La aplicación muestra la información con la que se registró el usuario.

Tabla 33. Requerimiento funcional 19 - Finalizar sesión

RF_19	Finalizar sesión
DESCRIPCIÓN	El usuario accede a esta función a través de un botón que finaliza su sesión.
OBJETIVO	Facilitar al usuario la opción de cerrar sesión según sea necesario.
PRIORIDAD	Baja
ACOTORES	Usuario
FLUJO DE INFORMACIÓN	El usuario cierra sesión y sale de la aplicación.

Requerimientos no funcionales

A continuación, se detallan los requisitos que sirven como marco para el diseño de funciones según la propuesta planteada.

Tabla 34. Requerimiento no funcional 01 - usabilidad.

RNF_01	USABILIDAD
DESCRIPCIÓN	Asegurar la facilidad de uso de la aplicación web; el usuario debe poder utilizarla de manera sencilla e intuitiva.
OBJETIVO	Proveer un acceso y manejo sencillo para el usuario.
PRIORIDAD	Alta
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	El usuario puede utilizar la aplicación web de manera fácil e intuitiva.

Tabla 35. Requerimiento no funcional 02 – Disponibilidad

RNF_02	Disponibilidad
DESCRIPCIÓN	La aplicación web debe ser funcional en todo momento y no presentar problemas en su funcionamiento.
OBJETIVO	Garantizar la disponibilidad continua de la aplicación.
PRIORIDAD	Alta
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	La aplicación web debe brindar el servicio en todo momento sin presentar problemas en su funcionamiento.

Tabla 36. Requerimiento no funcional 03 – Escalabilidad

RNF_03	Escalabilidad
DESCRIPCIÓN	La conexión de un gran número de usuarios no debe afectar el rendimiento de la aplicación.
OBJETIVO	Gestionar grandes cantidades de usuarios y transacciones sin afectar el rendimiento de la aplicación.
PRIORIDAD	Alta
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	La aplicación no debe presentar problemas al manejar grandes cantidades de usuarios y transacciones.

Tabla 37. Requerimiento no funcional 04 – Seguridad

RNF_04	Seguridad
DESCRIPCIÓN	Es crucial que la información de los usuarios esté protegida contra posibles ataques.
OBJETIVO	Asegurar la protección de la información de los usuarios.
PRIORIDAD	Alta
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	La aplicación debe asegurar las garantías necesarias para prevenir la pérdida de información.

Tabla 38. Requerimiento no funcional 05 – Velocidad de carga

RNF_05	Velocidad de carga
DESCRIPCIÓN	La aplicación web debe ofrecer una respuesta ágil al interactuar con el usuario.
OBJETIVO	Asegurar la protección de la información de los usuarios.
PRIORIDAD	Media
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	La aplicación web debe cargar durante la interacción con el usuario.

Tabla 39. Requerimiento no funcional 06 – Compatibilidad

RNF_06	Compatibilidad
DESCRIPCIÓN	La aplicación web debe ser compatible con los distintos navegadores
OBJETIVO	Garantizar el uso de la aplicación web con diferentes navegadores.
PRIORIDAD	Media
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	La aplicación debe ser accesible en navegadores

Tabla 40. Requerimiento no funcional 07 – Accesibilidad

RNF_07	Accesibilidad
DESCRIPCIÓN	La aplicación web debe ser accesible para usuarios con necesidades especiales, ya sea visuales o auditivas.
OBJETIVO	Proporcionar asistencia que facilite la interacción de los usuarios de manera sencilla.
PRIORIDAD	Media
ACOTORES	Programadores
FLUJO DE INFORMACIÓN	La aplicación debe ser accesible para personas con necesidades especiales.

Modelamiento de la base de datos

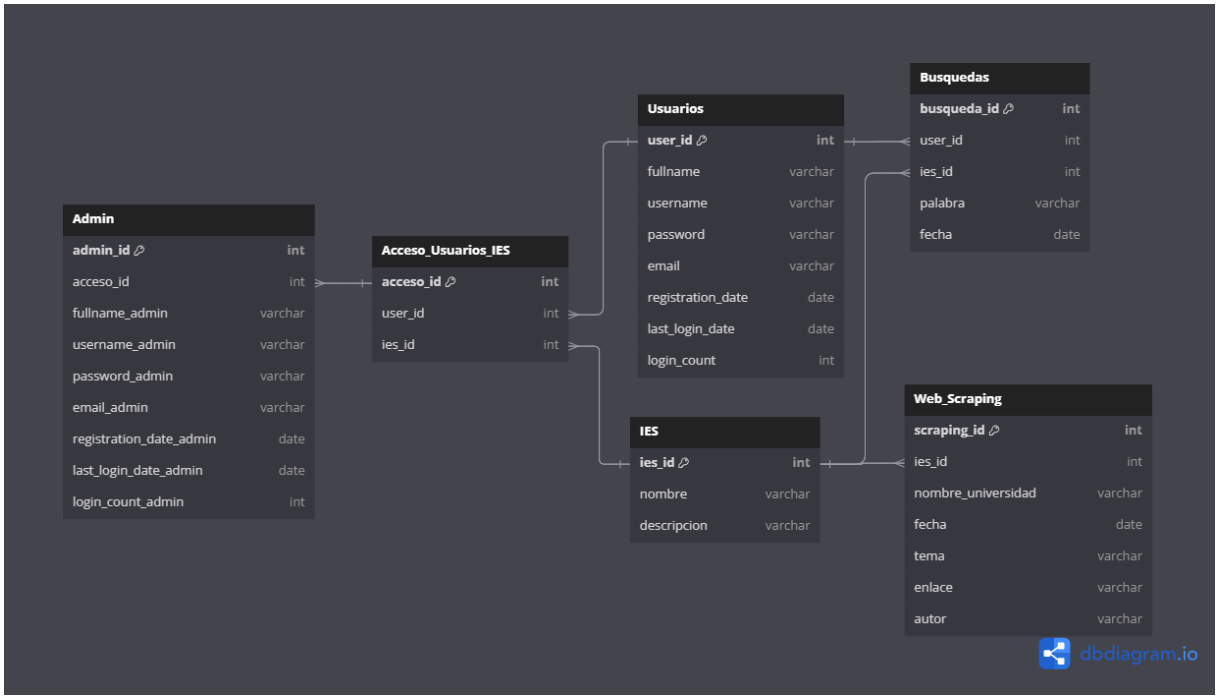


Figura 28. Modelo entidad relación de la base de datos

Manual de usuario

En esta etapa se realizan los ajustes finales a cada una de las pantallas según los requisitos, con el objetivo de asegurar que cada pantalla funcione al 100%. Esta fase se enfoca en la documentación del proyecto y en garantizar el cumplimiento de sus funcionalidades. A continuación, se presenta el manual de usuario elaborado para las tres aplicaciones: administrador y usuario.

Manual para el Administrador:

Página de bienvenida

Primero tenemos la página de bienvenida, donde se da una breve explicación del sistema y al hacer click en el botón de descubre más pasaremos a la página de inicio de sesión.

Figura 29. Página de Bienvenida

Inicio de sesión

Permite al administrador iniciar sesión ingresando los datos en el formulario, los datos serán validados y procesados con PostgresSQL y verificar, si a la cuenta existe pasara a la página principal del sistema para el administrador

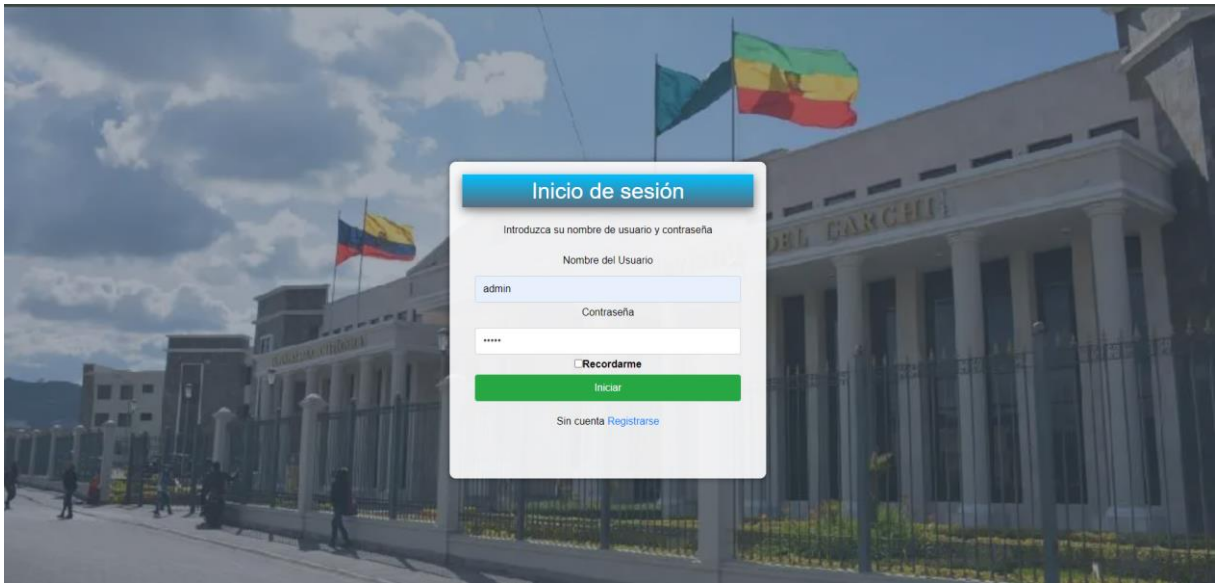


Figura 30. Pantalla de inicio de sesión del administrador

Panel de Administrador

Permite al administrador, realizar web scraping a repositorios universitarios, en los cuales están incluidos las siguientes universidades:

- Universidad Politécnica Estatal del Carchi
- Universidad del Pacifico
- Universidad de Especialidades Espíritu Santo
- Universidad Nacional de Loja
- Universidad de Azuay
- Universidad Internacional del Ecuador

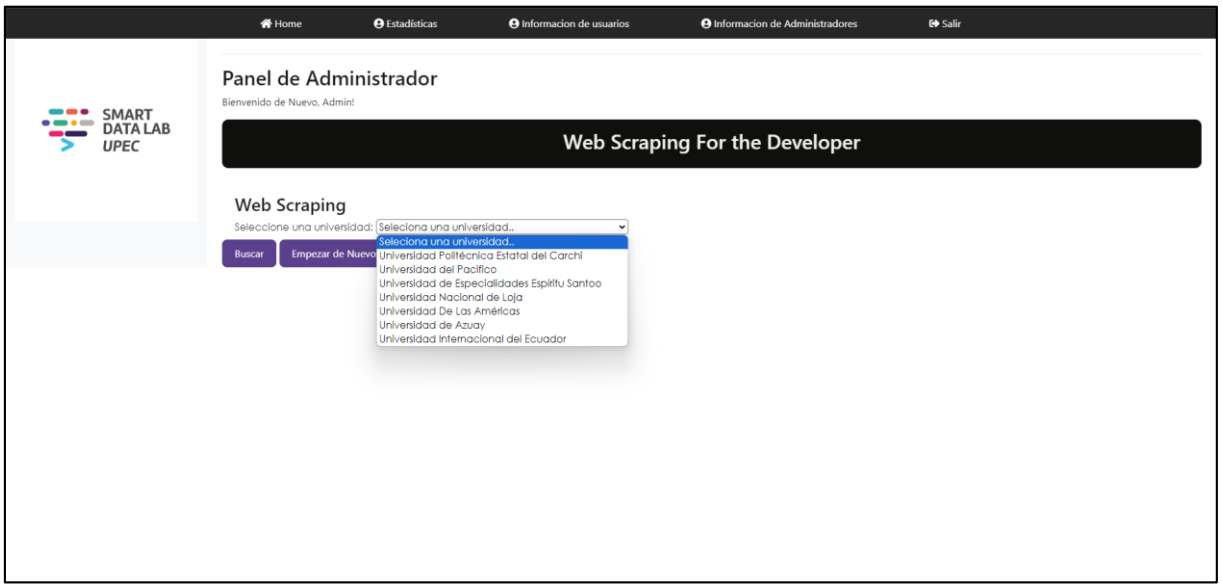


Figura 31. Pantalla Panel del administrador

Luego de escoger la universidad, seleccionamos el botón buscar para hacer el web scraping, también se puede interrumpir la consulta con el botón empezar de nuevo, además tenemos el botón subir datos que su principal función es enviar los datos del web scraping a la base de datos que está en PostgresSQL, para su almacenamiento y posterior uso.

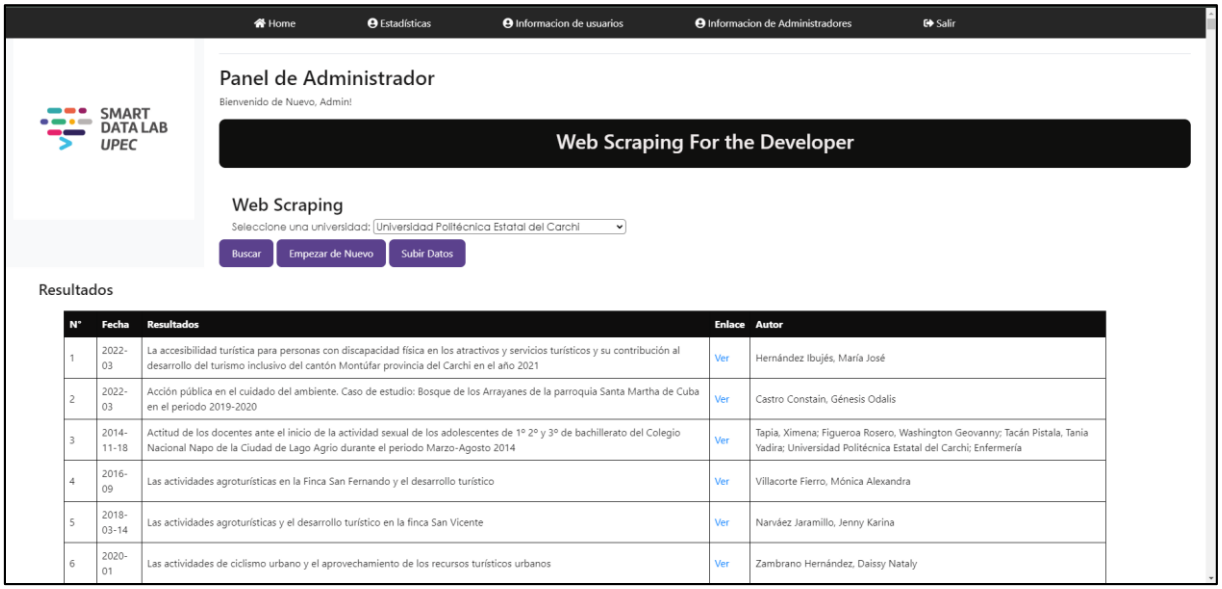


Figura 32. Web scaping a la UPEC

Página de estadísticas:

En este aparatado permite al administrador visualizar y descargar la información de las búsquedas que hizo el usuario, se puede descargar en formato csv las búsquedas generales con la siguiente información:

- Universidad
- Usuario
- Palabras buscadas
- Fecha de búsqueda.

De igual manera se puede descargar en formato csv el resumen de búsquedas con la siguiente información:

- Palabras más buscadas por los usuarios y veces buscada
- Universidad más buscada por los usuarios y veces buscada

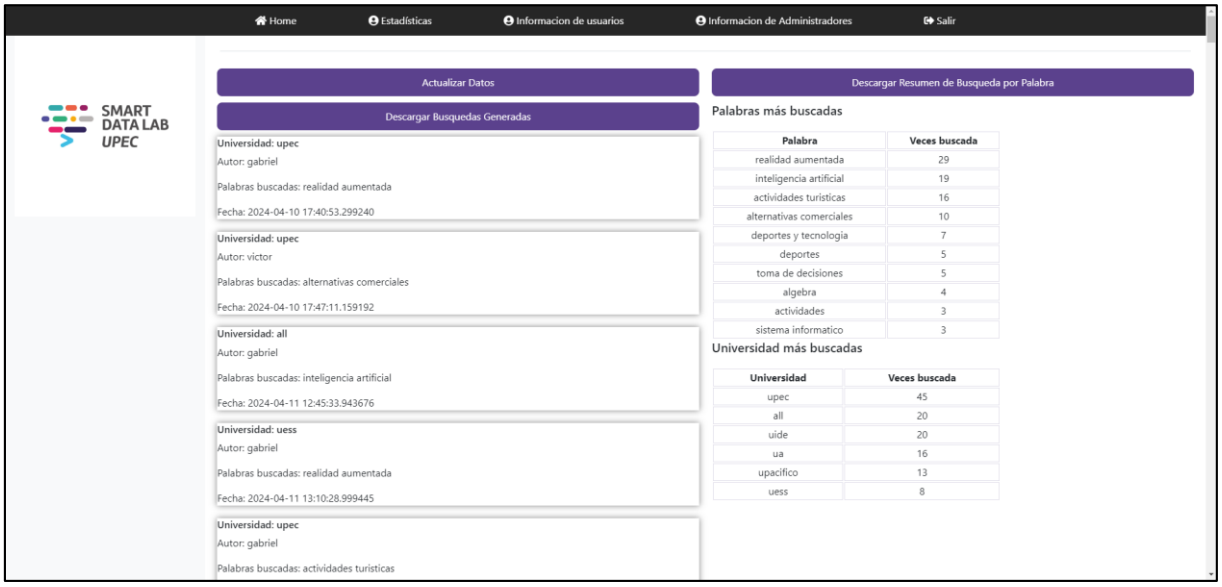


Figura 33. Pantalla de estadísticas del administrador

Página de información detallada de las cuentas de los usuarios

Permite al administrador hacer CRUD (Create, Read, Update, Delete), a las cuentas de los usuarios que se registraron en el sistema

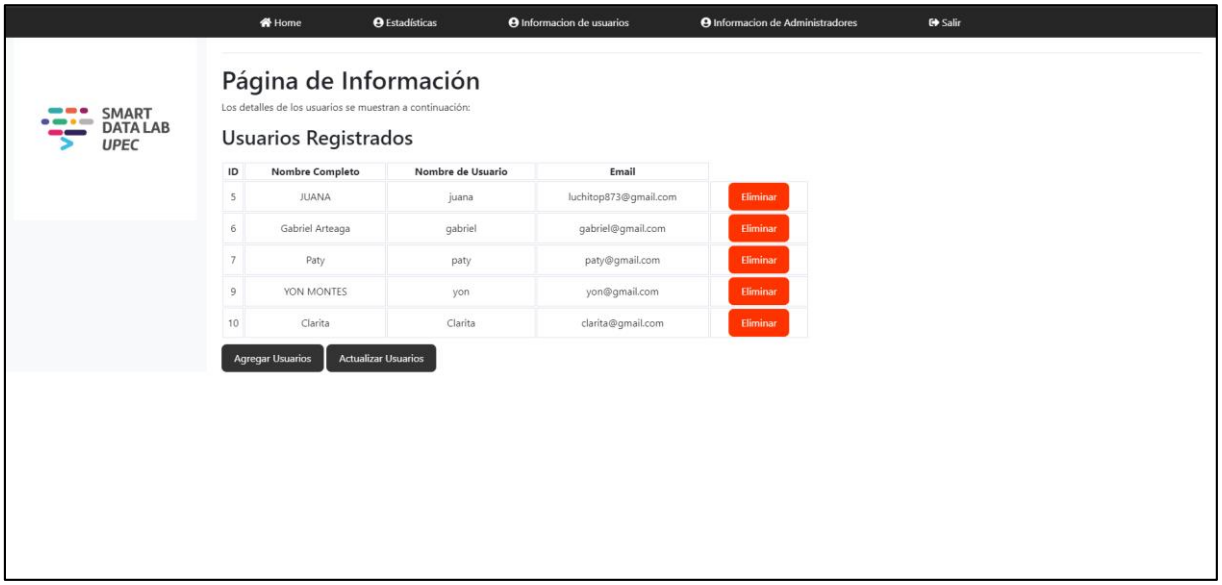


Figura 34. Pantalla de información de usuarios

Pantalla para agregar usuarios:

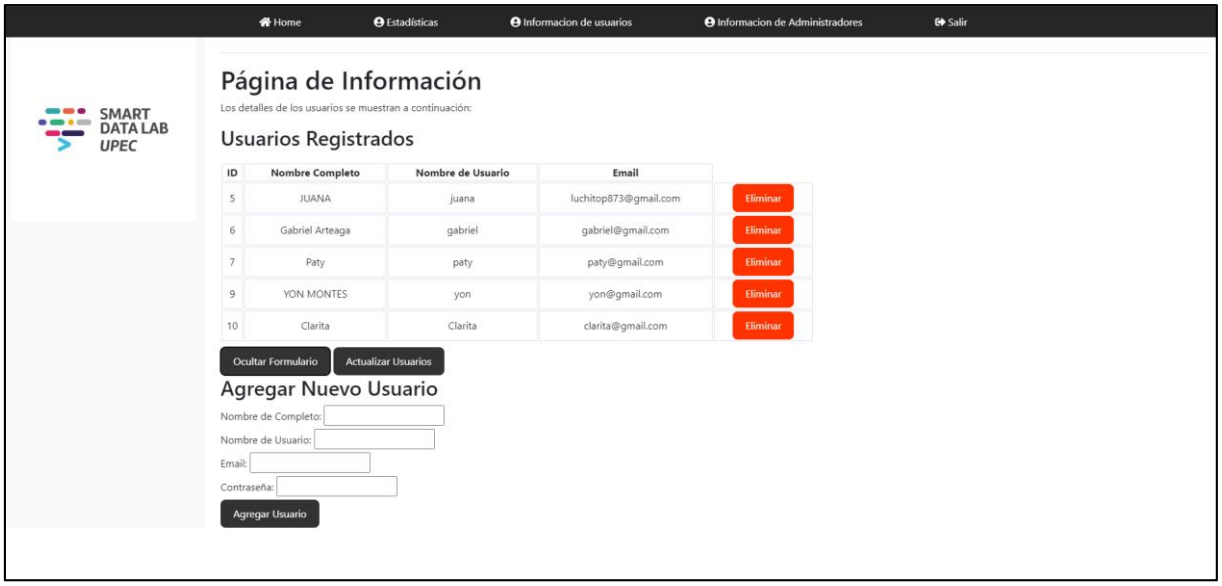


Figura 35. Pantalla para agregar usuarios

Pantalla para actualizar usuarios:

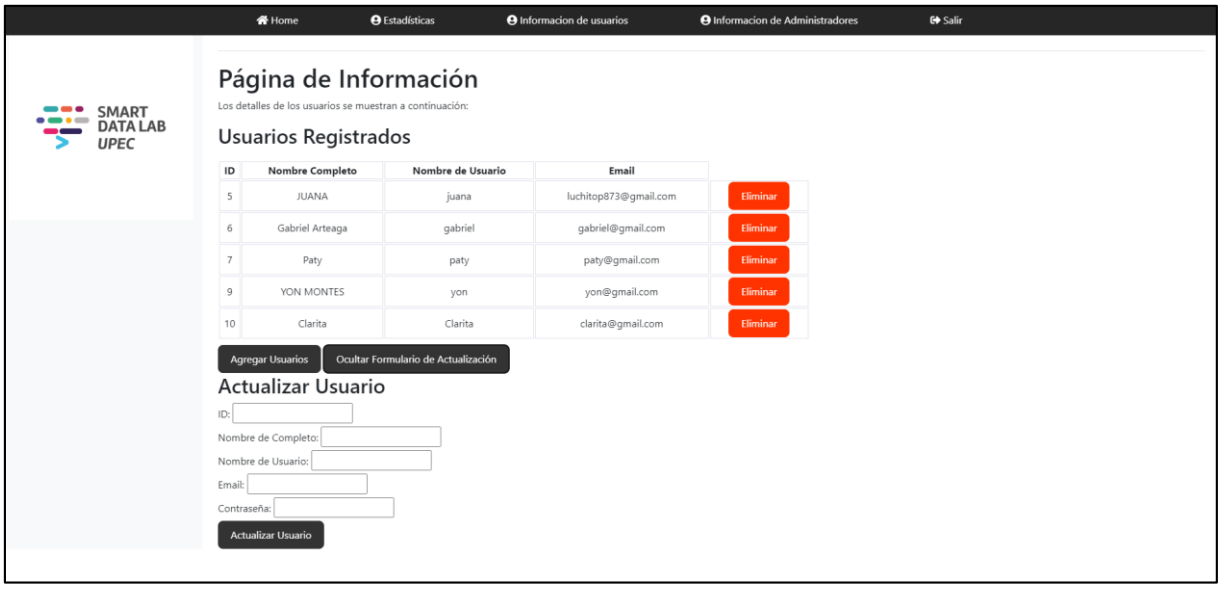


Figura 36. Pantalla para actualizar usuarios

Página de información detallada de las cuentas de los sub-administradores

Permite al administrador hacer CRUD (Create, Read, Update, Delete), a las cuentas de los sub-administradores, que el administrador agrego y delego para seguir de cerca la aplicación.

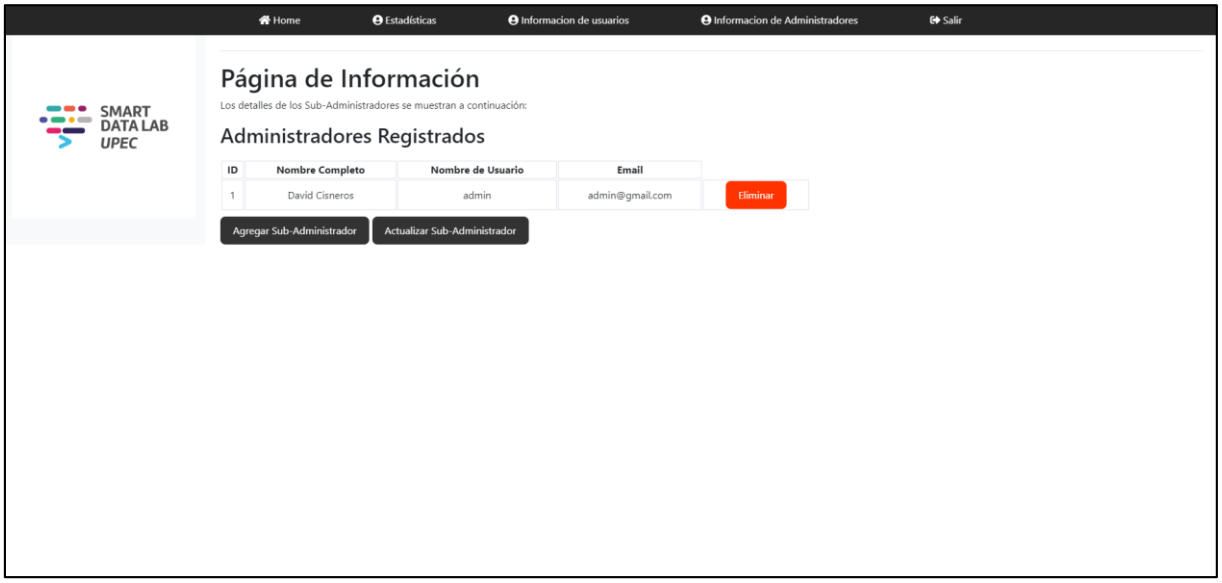


Figura 37. Pantalla de información de sub-administradores

Agregar nuevo sub-administrador:

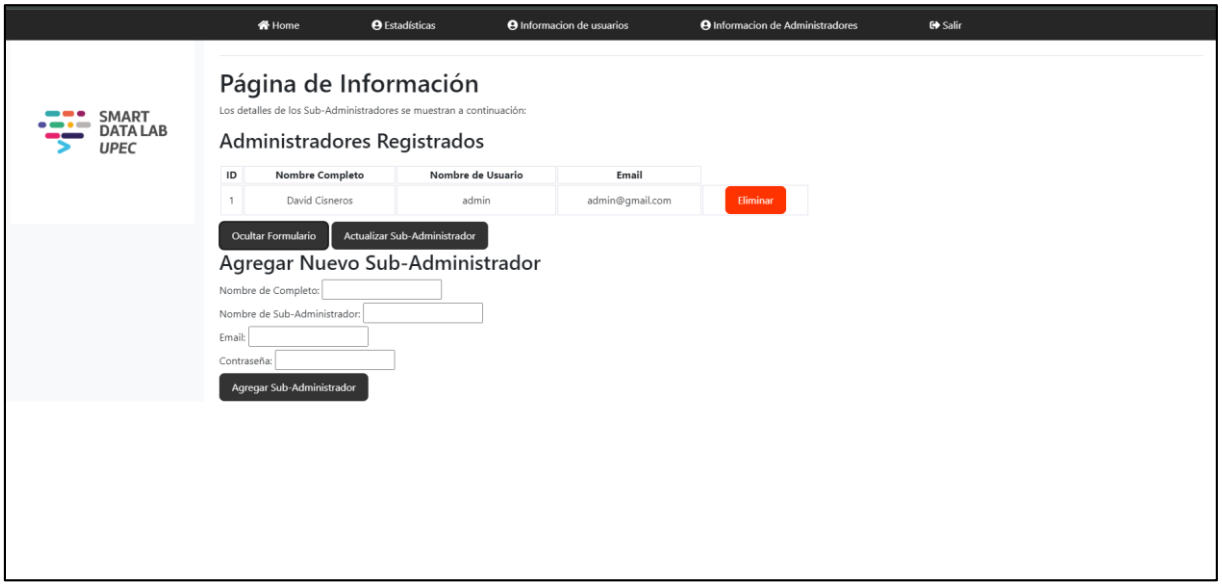


Figura 38. Pantalla Agregar nuevo sub-administrador

Pantalla para actualizar sub-administradores:

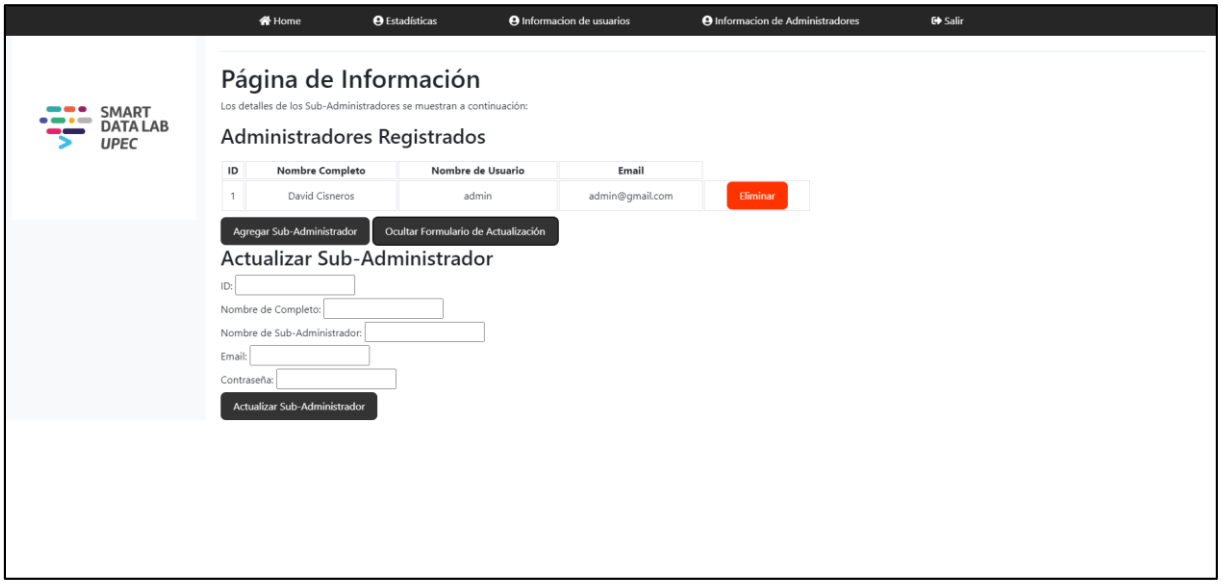


Figura 39. Pantalla para actualizar sub-administradores

Manual para el Usuario:

Página de bienvenida

Primero tenemos la página de bienvenida, donde se da una breve explicación del sistema y al hacer click en el botón de descubre más pasaremos a la página de inicio de sesión.

Figura 40. Página de Bienvenida

Inicio de sesión

Permite al usuario iniciar sesión ingresando los datos en el formulario, los datos serán validados y procesados con PostgresSQL y verificar si a la cuenta existe, de no ser el caso hay la opción para registrarse.

Una vez introducido los datos del usuario registrado previamente pasara a la página principal del usuario home.

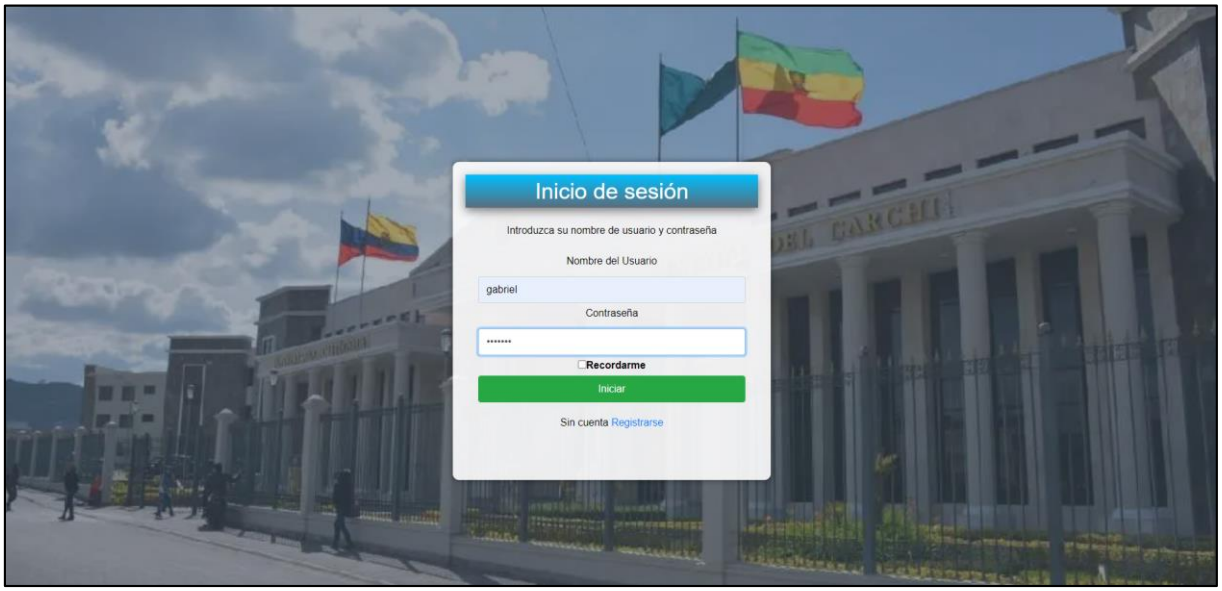


Figura 41. Pantalla de inicio de sesión del usuario

Registrarse

Permite al usuario registrarse y tener acceso al sistema, llenar el formulario con la siguiente información:

- Nombre completo
- Correo electrónico
- Nombre de usuario
- Contraseña

Hacer click en el botón de registrarse, para que la información ingresada en el formulario pase a la base de datos que esta en PostgresSQL, en caso de que la cuenta ya exista aparece un flash diciendo que la cuenta ya existe

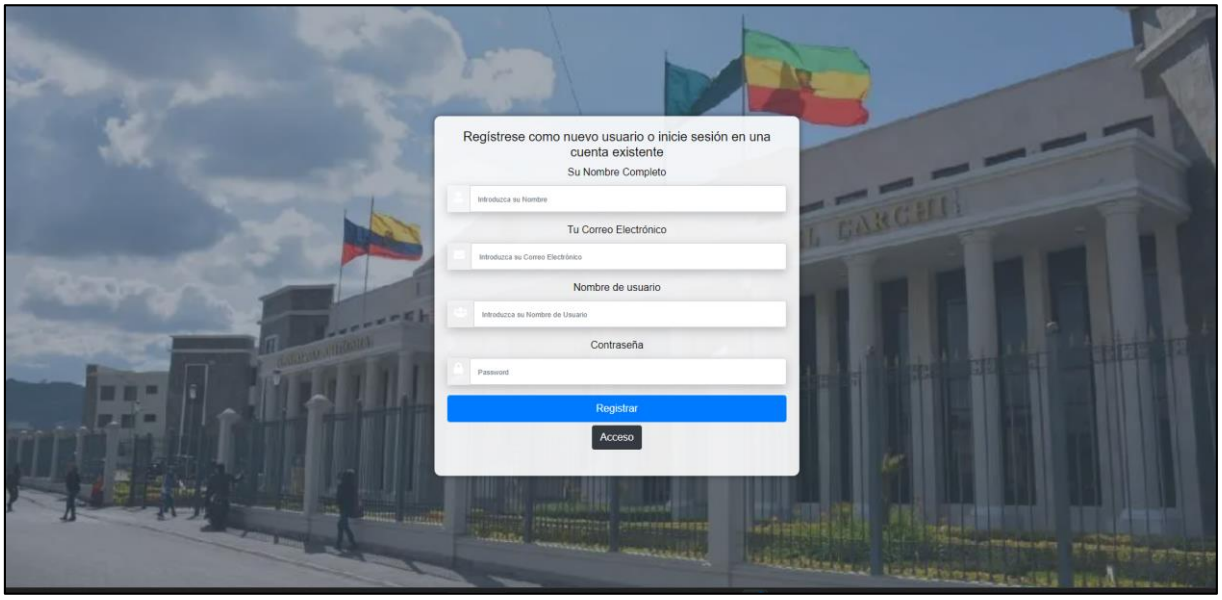


Figura 42. Pantalla de registro

Pantalla Home del usuario

Tenemos la pantalla Home del usuario, en la cual se puede explorar proyectos de investigación conectados en un gráfico visual, donde hay que seleccionar una universidad, en las cuales constan las siguientes:

- Universidad Politécnica Estatal del Carchi
- Universidad del Pacifico
- Universidad de Especialidades Espíritu Santo
- Universidad Nacional de Loja
- Universidad de Azuay
- Universidad Internacional del Ecuador

Una vez seleccionado una universidad, hay que ingresar las palabras claves a buscar.

Se nos muestra la siguiente información:

- Proyectos publicados en relación a las palabras buscadas, información que se muestra es:
    1. Tema del proyecto
    2. Numero de proyecto
    3. Fecha de publicación
    4. Autores
    5. Enlace de referencia

- Grafico visual para la agrupación entre los trabajos publicados
    1. Círculo más grande y obscuro: Mayor importancia
    2. Círculos más pequeños: Menor importancia
- Resultados de clasificación en relación a la predicción que hizo el algoritmo
- Resultados de la correlación, entre más se acerque a 1 es mejor la clasificación que hizo el algoritmo

Figura 43. Pantalla Home

Perfil del usuario

Permite al usuario ver su información con la que creo la cuenta

Figura 44. Pantalla perfil