PROTOTIPO DE SOFTWARE WEB PARA EL ANÁLISIS Y EVALUACIÓN DE LA EXPOSICIÓN DE DATOS SENSIBLES Y NO SENSIBLES EN LA RED SOCIAL INSTAGRAM APLICADA EN LA UNIVERSIDAD DEL VALLE SEDE TULUÁ

Sebastián Darío Giraldo Rodas

Álvaro Andrés Hurtado Vallecilla

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Cali - Tuluá
2024

PROTOTIPO DE SOFTWARE WEB PARA EL ANÁLISIS Y EVALUACIÓN DE LA EXPOSICIÓN DE DATOS SENSIBLES Y NO SENSIBLES EN LA RED SOCIAL INSTAGRAM APLICADA EN LA UNIVERSIDAD DEL VALLE SEDE TULUÁ

Sebastián Darío Giraldo Rodas
Código 202259391
giraldo.sebastian@correounivalle.edu.co

Álvaro Andrés Hurtado Vallecilla
Código 201958463
hurtado.alvaro@correounivalle.edu.co

Director
Ing. Luis Germán Toro Pareja, Msc.
Profesor de la Escuela de Ingeniería de Sistemas y Computación
luis.german.toro@correounivalle.edu.co

Codirector
Ing. Raúl E. Gutiérrez De Piñarez Reyes, Ph.D.
Profesor de la Escuela de Ingeniería de Sistemas y Computación
raul.gutierrez@correounivalle.edu.co

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Cali - Tuluá
2025

Tabla de Contenido

Resumen

En esta propuesta de trabajo de grado se busca la implementación de un prototipo de software web para capturar la información de texto de las redes sociales (web Scraping o API web scraper) y poder identificar el nivel de exposición de los datos personales sensibles y no sensibles en los campus universitarios, con este proyecto, se quiere motivar a otras entidades educativas a adoptar un enfoque similar, donde el factor clave es promover una cultura general de privacidad y seguridad en línea en todas las comunidades universitarias. Asimismo, se espera formar un futuro digital donde la protección de datos personales sea uno de los pilares fundamentales, garantizando así un mejor ambiente universitario tanto confiable como respetuoso para todos los miembros que la componen.

El presente proyecto constituye una posibilidad hacia la creación de entornos universitarios más seguros y respetuosos de la privacidad. Destacando el papel crucial que la tecnología desempeña como aliada en la protección de los derechos individuales y colectivos, con la aspiración a transformar no solo el entorno académico universitario, sino también a que sirva como ejemplo para otras instituciones educativas en Colombia. El trabajo pretende usar para su desarrollo, las técnicas de procesamiento de lenguaje natural y las tecnologías de tratamiento de datos teniendo en cuenta la normatividad de la privacidad de datos.

Palabras clave:

Campus Universitario, Web Scraping, Privacidad de datos, Redes sociales, Procesamiento de Lenguaje Natural.

Abstract

This degree work proposal seeks to implement a prototype of web software to capture text information from social networks (web scraping or API web scraper) and to identify the level of exposure of sensitive and non-sensitive personal data on university campuses, where this project seeks to motivate other educational institutions to adopt a similar approach, where the key factor is to promote a general culture of privacy and online security in all university communities. It is also expected to form a digital future where the protection of personal data is one of the fundamental pillars, thus ensuring a better university environment both reliable and respectful for all members that compose it.

The present project constitutes a significant advance towards the creation of more secure and privacy-friendly university environments. Highlighting the crucial role that technology plays as an ally in the protection of individual and collective rights, with the aspiration to transform not only the university academic environment, but also to serve as an example for other educational institutions in Colombia.

Keywords:

University Campus, Web Scraping, Data Privacy, Social Networks, Natural Language Processing.

1. Introducción

En la era digital, el flujo constante de información en las redes sociales ha transformado la manera en que las personas se relacionan y comparten datos. Este fenómeno ha cobrado una relevancia cada vez mayor, ya que los usuarios están inmersos en plataformas en línea donde llevan a cabo todo tipo de interacciones. A través de estas plataformas, buscan reflejar sus opiniones e ideales, evidenciar acontecimientos y, en muchos casos, realizar procesos virtuales en los que se comparte información personal, especialmente relacionada con datos transaccionales, sin mayor preocupación. Esto ocurre debido a que muchas de estas cuentas digitales están directamente vinculadas al número telefónico personal [1].

Sin embargo, este intercambio virtual de información plantea desafíos significativos en cuanto a la protección de la privacidad y la seguridad de los datos personales. Esta exposición inadvertida de información sensible puede tener repercusiones graves para los individuos, la reputación de las empresas y la institución educativa.

A partir de este contexto surgió la necesidad de desarrollar herramientas tecnológicas innovadoras que permitieran monitorear y evaluar la exposición de datos personales en los campus estudiantiles de la Universidad del Valle, sede Tuluá. En el presente proyecto se abordó esta problemática mediante el diseño e implementación de un prototipo basado en técnicas de Procesamiento de Lenguaje Natural (PLN), especializado en la captura de información textual publicada en redes sociales a través de web scraping o mediante una API de Web Scraper. El prototipo se enfocó en identificar con precisión los distintos niveles de exposición de datos sensibles y no sensibles dentro de los entornos mencionados, empleando métodos de análisis de datos y PLN. Este prototipo sienta las bases para la implementación de medidas preventivas y correctivas eficaces frente a la exposición de datos personales en redes sociales.

Con respecto a lo anterior, el propósito de este trabajo trasciende la mera exploración tecnológica; se busca promover una cultura de conciencia y responsabilidad frente al manejo de la información personal en el ámbito universitario. Se reconoce que la protección de la privacidad constituye un derecho fundamental que debe salvaguardarse en todo momento, y se reafirma el compromiso de contribuir activamente a su preservación mediante la aplicación de soluciones innovadoras y éticas [2].

2. Planteamiento del Problema.

2.1. Descripción del problema

La proliferación de redes sociales y plataformas de comunicación en línea ha dado lugar a un aumento significativo en la cantidad de datos personales que se comparten y divulgan públicamente. Esta exposición descontrolada de información plantea serias preocupaciones en términos de privacidad y seguridad para personas activas en las plataformas digitales [3].

El problema radica en la falta de herramientas efectivas para monitorear y evaluar la exposición de datos sensibles en este entorno digital. A menudo, los usuarios de redes sociales pueden compartir información personal sin ser plenamente conscientes de las implicaciones de privacidad que ello conlleva. Esto puede incluir desde detalles sobre la ubicación, rutinas diarias y hasta información más delicada, como números de identificación personal o datos bancarios [4].

Además, la naturaleza dinámica y descentralizada de las redes sociales dificulta aún más la tarea de proteger la privacidad de los usuarios. La información se propaga rápidamente y puede ser difícil de rastrear, lo que aumenta el riesgo de que datos sensibles vayan a parar en manos equivocadas y sean utilizados de manera inapropiada por personas ajenas a las causas pro universitarias.

Ahora bien, otro aspecto del problema radica en la falta de conciencia y capacitación a la comunidad universitaria sobre las prácticas seguras al hacer uso de las redes sociales. Diversos usuarios pueden no estar familiarizados con las configuraciones de privacidad convenientes, además de los riesgos que se corren, asociados a la divulgación excesiva de información personal en línea, permitiendo que la información privada, o ligada a terceros, se exponga y afecte, de una manera u otra, la confidencialidad del ente involucrado [5].

Podría decirse que, en conjunto, estos factores contribuyen al establecimiento de un panorama negativo, en el que la privacidad y la seguridad de los datos personales, se hallan en constante riesgo. Abordar este problema requiere un enfoque integral que consiga combinar con éxito, tanto la educación como la tecnología, logrando proteger adecuadamente la privacidad de los miembros de la comunidad académica.

2.2. Formulación del problema

El problema fundamental radica en la falta de una solución efectiva para monitorear, evaluar y mitigar la exposición de datos personales sensibles. La creciente actividad en redes sociales y plataformas en línea ha generado un flujo constante de información personal, la cual puede estar sujeta a riesgos de privacidad y seguridad debido a la exposición de datos de manera voluntaria o involuntaria.

A su vez, la ausencia de herramientas especializadas y protocolos claros para gestionar esta situación, esta exposición de datos, como historias clínicas o detalles sobre la orientación sexual, podría resultar en graves daños a la reputación de la institución. De alguna forma terminaría por desencadenarse casos de bullying, discriminación o desprestigio de la universidad, afectando tanto a la comunidad académica como a la imagen pública del organismo estudiantil. Es esencial implementar medidas efectivas para monitorear y mitigar estos riesgos, protegiendo la privacidad a la par de los derechos individuales en el entorno universitario.

¿Qué estrategia puede ser implementada para identificar y mitigar la exposición de datos sensibles de las redes sociales utilizadas en el campus universitario? ¿Cómo se puede desarrollar herramientas tecnológicas que permitan detectar de manera eficiente la exposición de información personal en las redes sociales dentro de los campus universitarios?

3. Marco de Referencia

3.1. Marco teórico

3.1.1. Desarrollo web y extracción de datos

Para empezar, tanto el desarrollo web como la extracción de datos se rigen sobre dos pilares fundamentales: el scraping y el procesamiento de datos. Cada uno despliega funciones específicas y complementarias, esenciales para la creación de aplicaciones web robustas y eficientes.

El scraping representa la primera etapa del desarrollo web orientado a la extracción de datos. Este consiste en la recopilación automatizada de información de diversas fuentes en la web, utilizando tecnologías como BeautifulSoup en Python o Puppeteer en JavaScript. Como se señala en la literatura especializada, el scraping web es una técnica poderosa para extraer datos de sitios web, pero debe ser utilizado de manera ética y respetuosa con los términos de servicio del sitio web objetivo. Los desarrolladores pueden extraer datos estructurados de páginas web de manera eficiente; este proceso se convierte en la base sobre la cual se construye la funcionalidad de la aplicación, permitiendo la obtención de información relevante para su posterior procesamiento.

Por otro lado, el procesamiento de datos constituye la fase posterior al scraping, donde se lleva a cabo el análisis, transformación y almacenamiento de la información extraída. En esta etapa, se utilizan herramientas y tecnologías como Pandas en Python o frameworks de base de datos como MongoDB y MySQL. Según la documentación técnica, Pandas proporciona estructuras de datos flexibles y herramientas de análisis de datos diseñadas para trabajar con datos estructurados. El objetivo es agrupar los datos de manera coherente y segura, facilitando su acceso y manipulación para su posterior uso en la aplicación web.

Ahora bien, la colaboración entre el scraping y el procesamiento de datos es fundamental para el éxito de las aplicaciones web centradas en la extracción y análisis de información. Mientras que el scraping proporciona la materia prima, es decir, los datos brutos, el procesamiento de datos se encarga de convertirlos en conocimiento útil, tratable y aplicable. Esta sinergia entre ambas disciplinas permite la creación de aplicaciones web que no solo son capaces de recopilar información de manera automatizada, sino también de analizarla y representarla de manera significativa para el usuario.

3.1.2. Metodologías ágiles aplicadas al web scraping

Las metodologías ágiles han revolucionado el campo del desarrollo web y la extracción de datos, brindando una alternativa dinámica y eficiente a los enfoques tradicionales. Estas metodologías están diseñadas para minimizar los riesgos de fracaso en los proyectos, abordando desafíos como la subestimación de costos, tiempos y funcionalidades. A diferencia de los métodos convencionales que buscan rigidez y predictibilidad, las metodologías ágiles son adaptables, pues están centradas en las personas, y flexibles, permitiendo ajustes según las necesidades específicas de cada equipo y proyecto.

Seguidamente, se explica cómo estas metodologías ágiles se aplican en el desarrollo web, la extracción de datos y el scraping.

- **Scrum adaptado para el desarrollo web**

Inspirado en el rugby. Scrum se ha adaptado para mejorar la flexibilidad y la rapidez en el desarrollo web y la extracción de datos. Emplea un marco de trabajo estructurado con roles, artefactos y reglas bien definidas. Los "sprints" constituyen elementos centrales, permitiendo entregas incrementales y adaptabilidad a los cambios en los requisitos del proyecto. Los roles como scrum master, product owner y equipo de desarrollo fomentan la autogestión, la colaboración y la eficiencia. Scrum es especialmente efectivo en proyectos web donde los requisitos cambian rápidamente y la colaboración continua con los clientes es esencial [6].

- **Extreme programming (XP) para el scraping y la extracción de datos**

El XP se enfoca en la mejora de las relaciones interpersonales y la eficiencia en el desarrollo; valora la retroalimentación continua, la comunicación afectiva y la simplicidad de soluciones. Además, utiliza "historias de usuario" para especificar requisitos y prácticas como entregas pequeñas o programación en parejas. Este es adecuado para proyectos de scraping, extracción de datos con requisitos cambiantes y alto riesgo técnico, además define roles específicos como el programador, el cliente y el encargado de pruebas, fomentando la colaboración estrecha entre el equipo y los stakeholders para construir valor en cada iteración [7].

- **Enfoque lean en el desarrollo web y extracción de datos**

Este se centra en la eficiencia y la eliminación de desperdicios en el proceso de desarrollo web y extracción de datos. Promueve la reducción de artefactos innecesarios, el mejoramiento continuo y la colaboración efectiva al tamaño del equipo, Lean se ajusta para mejorar la eficiencia y la calidad del producto [8].

- **Desarrollo guiado por pruebas (TDD) para la extracción de datos**

Este se enfoca en escribir pruebas automatizadas antes de escribir el código de producción, lo cual garantiza que el código cumpla con los requisitos y funcione según lo esperado. En el contexto de la extracción de datos, TDD asegura la calidad y confiabilidad de los procesos de scraping, reduciendo la posibilidad de errores, además de facilitar la detección temprana de problemas [9].

3.1.3. Protocolos de comunicación web

Son conjuntos de reglas y convenciones que permiten que los sistemas informáticos se comuniquen entre sí a través de la World Wide Web (WWW). Estos protocolos son fundamentales para el funcionamiento de internet, además son responsables de la transferencia de datos entre los clientes y los servidores que alojan los recursos web, como páginas HTML, imágenes, archivos de estilo, scripts y otros tipos de contenido.

Uno de los protocolos clave en la comunicación web es el Protocolo de Transferencia de Hipertexto (HTTP), que especifica cómo deben solicitarse y transmitirse los recursos web. Este es un protocolo sin estado, lo que implica que cada solicitud se procesa de manera independiente, sin considerar solicitudes anteriores. Además, es un protocolo basado en texto, lo que facilita su comprensión y depuración [10].

HTTP funciona mediante un modelo de solicitud-respuesta. Cuando un cliente, como un navegador web, quiere acceder a un recurso, envía una petición al servidor que lo aloja; esta incluye elementos clave, como la URL del recurso solicitado, el método de solicitud (GET, POST, PUT, DELETE), cabeceras opcionales que aportan información adicional sobre la petitoria y, para los métodos POST y PUT, el cuerpo de la solicitud con los datos enviados al servidor [11].

Cuando el servidor recibe la petición, la procesa y envía una respuesta HTTP al cliente, esta incluye un código de estado que indica si la solicitud fue exitosa (por ejemplo, 200 para una petitoria exitosa, 400 para un recurso no encontrado, 500 para error interno del servidor), cabeceras adicionales con información sobre la contestación y, en caso de éxito, el cuerpo de la respuesta con el recurso solicitado [10].

Otro protocolo crucial es el Protocolo de Transferencia de Hipertexto Seguro (HTTPS); HTTPS es una versión segura de HTTP que emplea cifrado SSL/TLS para proteger la comunicación entre el cliente y el servidor. Se utiliza para asegurar la confidencialidad e integridad de los datos transmitidos, especialmente cuando se maneja información sensible como contraseñas, datos personales o transacciones financieras.

Algunos protocolos y tecnologías esenciales para comunicación web incluyen el Protocolo de Control de Transmisión (TCP) y de internet (IP), que constituyen la base de internet la transmisión de datos a través de redes de computadoras. Adicionalmente, el Protocolo de Configuración Dinámica de Host (DHCP) permite la asignación automática de direcciones IP dinámicas a los dispositivos conectados a una red, eliminando la necesidad de configuraciones manuales y facilitando la administración de recursos. De manera complementaria, el Sistema de Nombres de Dominio (DNS) realiza la traducción de nombres de dominio legibles por el usuario en direcciones IP numéricas, permitiendo que los dispositivos puedan localizarse y comunicarse correctamente dentro de una red informática [10].

3.1.4. Arquitectura de software

Este un pilar fundamental en el desarrollo de aplicaciones modernas, equiparable a un plano arquitectónico en la construcción de un edificio. Es un concepto que ha evolucionado a lo largo de décadas, arraigado en patrones, modelos y abstracciones teóricas; desplegado para enfrentar la complejidad de proyectos de software en constante crecimiento y demanda. Entre las diversas arquitecturas que han surgido y evolucionado, tres destacan por su relevancia y aplicabilidad en diferentes contextos: serverless (sin servidor), arquitectura cliente-servidor y arquitectura en capas, junto con el Modelo-Vista-Controlador (MVC) [12].

Arquitectura Cliente-Servidor: Distribuye las tareas entre los clientes y los servidores, permitiendo un control centralizado y la compartición de recursos. Aunque ofrece beneficios en términos de control y gestión de recursos, puede resultar costosa y vulnerable a fallas del servidor principal, además de enfrentar desafíos en el mantenimiento de hardware y software especializadas [13].

3.1.5. Procesamiento de Lenguaje Natural (PLN)

Este representa una fascinante interacción entre el lenguaje humano y la computación. Las máquinas y algoritmos se encuentran con la complejidad y la riqueza de la comunicación humana, permitiendo a las máquinas entender, interpretar y generar texto de manera similar a los individuos. Este campo multidisciplinario combina conocimientos de lingüística, inteligencia artificial, informática y psicología, entre otros, para desarrollar sistemas capaces de procesar y comprender el lenguaje natural [14].

El PLN ha revolucionado numerosos aspectos de nuestra vida cotidiana, desde los motores de búsqueda que interpretan consultas de manera semántica hasta los asistentes virtuales que responden a preguntas complejas; el PLN está omnipresente en la era digital. En el ámbito empresarial, se utiliza para análisis de sentimientos de redes sociales o la clasificación automática de correos electrónicos [14].

Sin embargo, el PLN también enfrenta desafíos significativos. Por ejemplo, la ambigüedad del lenguaje humano, las variaciones idiomáticas y culturales, además de la comprensión del contexto son de las barreras que los sistemas de PLN deben superar. Asimismo, el sesgo inherente en los datos de entrenamiento puede llevar a resultados discriminatorios o inexactos, lo que plantea preocupaciones éticas y sociales.

A medida que el PLN continúa avanzando, surgen nuevas oportunidades y desafíos. El desarrollo de modelos de lenguaje cada vez más avanzados, como los modelos de aprendizaje profundo, ha mejorado significativamente la capacidad de las máquinas para comprender y generar texto de manera más natural y precisa. Sin embargo, queda mucho trabajo por hacer en áreas como la interpretación del contexto, el razonamiento sobre el conocimiento común y la comprensión del tono y la emoción en el lenguaje humano.

En perspectiva futura, el PLN se perfila como una tecnología con el potencial de revolucionar profundamente la forma en que interactuamos con los sistemas digitales y nuestro entorno. Desde plataformas de traducción instantánea que superan las barreras lingüísticas, hasta asistentes virtuales capaces de interpretar con precisión nuestras intenciones, necesidades y deseos, el PLN representa una vía prometedora para fortalecer la comunicación y cooperación entre seres humanos y máquinas.

3.1.6. Front-end

Se trata de elementos visuales e interactivos presentes en sitios web o aplicaciones web, mediante los cuales los usuarios establecen una interacción directa. Estos componentes cumplen una función clave tanto en la operatividad del sistema como en su presentación estética.

3.1.7. Back-end

Es la parte de una aplicación que el usuario no ve, se enfoca en la lógica del sitio, la gestión de los servidores, el manejo de las bases de datos y muchos otros componentes de naturaleza similar.

3.1.8. Despliegue

Es el proceso de hacer que una aplicación web sea accesible para los usuarios finales. Implica preparar y empaquetar una aplicación para su implementación, ejecutar pruebas para validar la funcionalidad y el rendimiento, optimizar las bases de datos y las estrategias de respaldo, implementar medidas de seguridad sólidas y lanzar la versión en vivo de la aplicación al entorno de producción.

3.1.9. Framework

Es una estructura que se puede utilizar para crear software. Actúa como base que evita el tener que lidiar con la creación de lógica adicional innecesaria desde cero. Es similar a una plantilla en el sentido de que puede ser modificada y se le pueden agregar ciertas características y funcionalidades superiores.

3.1.10. Librería

En programación, una librería es un conjunto de funcionalidades o piezas de código, que ejecutan tareas específicas, las cuales suelen ser repetitivas y genéricas. Es código escrito por alguien más que ayuda a realizar tareas comunes, evitando la escritura de líneas de código adicional, y dando libertad a los programadores de cuando hacer uso de ellas o no.

3.1.11. REST API

Es una interfaz que se ajusta a los principios de diseño del estilo arquitectónico de transferencia de estado representacional (REST), el cual define una serie de restricciones que indican a un sistema de internet, como la web, la forma en que debe comportarse.

3.1.12. Seguridad informática

La seguridad hace referencia a la capacidad de un software para seguir funcionando correctamente bajo ataques maliciosos o fallas internas del sistema, garantizando la integridad de la información y operaciones.

3.1.13. Dato sensible

Son aquellos datos personales que afectan la intimidad del titular o cuyo uso indebido puede generar discriminación. Incluyen información que revela características íntimas como origen racial o étnico, orientación política, convicciones religiosas o filosóficas, afiliación sindical, datos relativos a la salud, vida sexual, datos biométricos, entre otros. Estos datos requieren una protección especial debido a su naturaleza delicada y el riesgo que implica su mal uso.

3.1.14. Dato no sensible

Un dato no sensible, también llamado dato personal común, es cualquier información que permite identificar a una persona física directa o indirectamente, pero que no afecta su intimidad ni conlleva un riesgo grave para sus derechos o libertades en caso de ser divulgada o tratada. Ejemplos típicos incluyen el nombre, dirección, número de teléfono, correo electrónico, sexo, número de documentos como pasaporte o licencia de conducir.

3.1.15. Ley de Protección de Datos Personales 1581 de 2012

Es la norma que regula la protección de datos personales en Colombia, desarrollando el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar la información que se haya recogido sobre ellas en bases de datos o archivos, tanto en entidades públicas como privadas.

Esta ley establece los principios, derechos y obligaciones para el tratamiento adecuado de los datos personales, buscando garantizar la privacidad y seguridad de la información, y proteger los derechos fundamentales relacionados con la intimidad y el buen uso de los datos [15].

3.1.16. Red social - Instagram

Instagram es una red social visual creada en 2010 por Kevin Systrom y Mike Krieger, inicialmente como una plataforma para compartir fotos con filtros artísticos en formato cuadrado, inspirada en cámaras como la Kodak [16]. Su rápida adopción (1 millón de usuarios en dos meses y 10 millones en un año) llevó a su adquisición por Facebook (hoy Meta) en 2012 por 1.000 millones de dólares. Actualmente, Instagram supera los 2.000 millones de usuarios activos mensuales, consolidándose como una de las redes sociales más influyentes del ecosistema digital. A continuación, se presentan algunas de sus funcionalidades clave:

- **Publicaciones tradicionales:** fotos y videos de hasta 60 segundos, con opción de carruseles.
- **Historias (Stories):** contenido efímero de 24 horas, lanzado en 2016 como respuesta a Snapchat. Incluyen herramientas como encuestas, geolocalización y enlaces.
- **Reels:** videos cortos estilo TikTok para descubrir tendencias.
- **Mensajes directos y videollamadas:** comunicación privada entre usuarios.

Modelo técnico y de datos: Instagram emplea algoritmos múltiples que priorizan contenido según interacciones previas, relación con el creador y tipo de formato. Recolecta datos como ubicación, intereses y comportamiento para personalizar anuncios y sugerencias, integrando esta información con otras plataformas Meta (Facebook, WhatsApp).

Impacto cultural y normativo: Su diseño fomenta la creatividad visual y el marketing digital, pero también plantea desafíos en la privacidad. Cumple con normativas como la Ley 1581 de 2012 en Colombia [16], que exige transparencia en el tratamiento de datos. Las Historias destacadas y el etiquetado geográfico amplifican la exposición de usuarios, mientras herramientas como la autenticación en dos pasos buscan mitigar riesgos.

Evaluación de configuración de privacidad y seguridad en Instagram. La configuración de privacidad en Instagram determina el grado de exposición de tu información personal. Las cuentas privadas ofrecen mayor control y protección, mientras que las públicas maximizan el alcance pero incrementan los riesgos de exposición. Las opciones de etiquetado, geolocalización y segmentación de audiencia por publicación permiten ajustar la privacidad de manera granular, aunque siempre existe un nivel básico de información visible para todos (foto de perfil, nombre, biografía). Las herramientas de seguridad como la autenticación en dos pasos y las funciones de bloqueo/restricción brindan protección adicional ante interacciones no deseadas o intentos de acceso no autorizado. Los perfiles comerciales, por su naturaleza, muestran más datos y se usan para análisis y marketing, lo que implica una exposición mayor por defecto [17].

Comparación entre cuentas públicas y privadas

- Las cuentas públicas permiten que cualquier usuario de Instagram vea tus fotos, historias y publicaciones sin restricciones, y cualquier persona puede seguirte sin tu aprobación.

- Las cuentas privadas requieren que apruebes manualmente a cada seguidor; solo los aprobados pueden ver tu contenido, lo que limita la exposición de tu información personal.

Grado de exposición de la información personal

- En cuentas públicas, tu información (biografía, publicaciones, historias, seguidores, seguidos) es visible para cualquier usuario, incluso si no te sigue.

- En cuentas privadas, solo los seguidores aprobados pueden ver tus publicaciones y detalles personales, aunque la foto de perfil, nombre de usuario y biografía siguen siendo visibles para todos.

Datos visibles para terceros con o sin seguimiento mutuo

- Con cuentas públicas, cualquier usuario (con o sin seguimiento mutuo) puede ver todas tus publicaciones, historias públicas y listas de seguidores/seguidos.

- Con cuentas privadas, solo los seguidores aprobados acceden a tus publicaciones y stories; otros usuarios solo ven tu foto de perfil y biografía.

Impacto de las configuraciones de privacidad por publicación

- Instagram está probando funciones para permitir que ciertas publicaciones sean visibles solo para grupos específicos, como amigos cercanos, lo que añade un nivel de control por publicación [18].

- Actualmente, las historias ya permiten segmentar la audiencia (por ejemplo, solo amigos cercanos), pero las publicaciones tradicionales no ofrecen esta opción de forma general.

Quién puede comentar, compartir, etiquetar

- Puedes limitar quién comenta en tus publicaciones desde la configuración de privacidad, eligiendo entre todos, personas que sigues, tus seguidores, o ambos.

- Puedes restringir, bloquear o silenciar a usuarios específicos para controlar su interacción contigo [19].

- El etiquetado se puede limitar para que solo personas que sigues puedan etiquetarte, o puedes aprobar manualmente cada etiqueta.

- Compartir publicaciones depende de si tu cuenta es pública o privada; en cuentas privadas, solo tus seguidores pueden compartir tu contenido en mensajes directos, pero no fuera de Instagram.

Restricciones para historias destacadas o en vivo

- Las historias pueden ser ocultadas a usuarios específicos, y puedes elegir compartirlas solo con tu lista de amigos cercanos.

- Las transmisiones en vivo también pueden ser restringidas según la configuración de privacidad de la cuenta y la selección manual de espectadores.

Uso del etiquetado y geolocalización

- El etiquetado de personas y el uso de geolocalización en publicaciones pueden aumentar la exposición de tus datos, ya que cualquier usuario que vea la publicación puede acceder a la ubicación o al perfil etiquetado.

- Las publicaciones con geotags pueden revelar información sensible sobre tus rutinas, lugares frecuentes o ubicación actual, incrementando el riesgo de exposición de datos personales.

- Comparativamente, una publicación sin geotag limita la información que terceros pueden obtener sobre tu localización.

Acciones de protección disponibles para el usuario

- Herramientas como bloqueo, reporte, restricción y eliminación de seguidores permiten controlar quién puede interactuar contigo y ver tu contenido.
- El bloqueo evita que otro usuario acceda a tu perfil o te envíe mensajes, mientras que la restricción reduce la visibilidad de sus comentarios e interacciones sin notificarle.
- La autenticación en dos pasos es una medida fundamental de seguridad para proteger el acceso a la cuenta, añadiendo una capa extra al requerir un código adicional al iniciar sesión.

Accesibilidad de la información desde perfiles corporativos o comerciales

- Los perfiles de empresa muestran por defecto información adicional como sector, dirección y botones de contacto (correo electrónico, teléfono, ubicación).
- Estos datos son visibles para cualquier usuario, lo que incrementa la exposición, aunque son útiles para fines comerciales y de marketing.
- Las cuentas comerciales permiten el acceso a herramientas de análisis y estadísticas, lo que puede implicar el uso de datos con fines de marketing y segmentación de audiencias

4. Estado del Arte

4.1. Investigaciones previas sobre exposición de datos personales en redes sociales

La exposición de datos personales en redes sociales ha sido ampliamente estudiada, dada la creciente preocupación por la privacidad y la seguridad en entornos digitales. Acquisti y Gross [20] analizaron cómo los usuarios de Facebook subestiman los riesgos asociados a la divulgación de información personal, destacando la falta de conciencia como un factor clave en la exposición involuntaria. Tufekci [21] y Christofides et al. [22] profundizaron en la relación entre la configuración de privacidad y la exposición de datos sensibles, subrayando la necesidad de herramientas que ayuden a los usuarios a gestionar mejor su información. En el contexto latinoamericano, Ramírez et al. [23] identificaron prácticas inseguras y baja percepción de riesgo en estudiantes universitarios colombianos, lo que refuerza la necesidad de soluciones tecnológicas para monitorear y evaluar la exposición de datos personales. Sin embargo, la mayoría de estos estudios se centran en análisis cualitativos o encuestas, sin proponer herramientas automatizadas para la detección y mitigación de riesgos.

Análisis crítico: Aunque estos trabajos establecen la importancia del problema, no abordan la creación de sistemas automatizados para identificar la exposición de datos sensibles en plataformas específicas como Instagram, ni se enfocan en entornos universitarios con un enfoque preventivo y educativo.

4.2. Aplicaciones de web scraping en análisis de contenido social

El web scraping es una técnica esencial para la extracción automatizada de datos de redes sociales, permitiendo el análisis a gran escala del contenido generado por los usuarios. Mitchell [24] destaca que el scraping permite recolectar datos estructurados de sitios web, siempre que se respeten los aspectos éticos y legales. Pizarro et al. [25] desarrollaron un sistema de scraping para analizar publicaciones de estudiantes en Twitter e Instagram, identificando patrones de exposición de datos personales. García et al. [26] implementaron un scraper para monitorear la reputación institucional en Instagram, detectando menciones que podrían afectar la imagen universitaria.

Análisis crítico: Si bien existen aplicaciones de scraping para análisis social, pocos trabajos integran esta técnica con procesamiento de lenguaje natural para clasificar la sensibilidad de los datos, y menos aún con un enfoque en la privacidad y seguridad en el contexto universitario colombiano. Además, Instagram representa un desafío técnico por sus políticas restrictivas, lo que limita la cantidad de investigaciones en esta red social.

4.3. Uso de PLN para clasificación de sensibilidad de datos

El procesamiento de lenguaje natural (PLN) ha sido utilizado para automatizar la detección de información sensible en textos publicados en redes sociales. Fernandes et al. [27] propusieron un modelo basado en PLN para identificar datos privados en publicaciones de

Facebook, alcanzando altos niveles de precisión. En el contexto universitario, González [28] desarrolló un sistema que combina scraping y PLN para analizar la exposición de datos personales en Instagram, clasificando el contenido según su nivel de sensibilidad y riesgo reputacional. Sánchez et al. [29] aplicaron técnicas de aprendizaje automático y PLN para detectar información personal identificable en grandes volúmenes de datos extraídos de redes sociales.

Análisis crítico: Aunque estos avances son significativos, la mayoría de los modelos no están adaptados a las categorías de datos sensibles definidas por la legislación colombiana, ni se enfocan en la creación de herramientas replicables para la gestión universitaria. Además, la integración de scraping, PLN y visualización de resultados en un prototipo funcional es un área con amplio margen de innovación.

4.4. Valor agregado del proyecto propuesto

El presente proyecto propone una solución integral que combina técnicas de web scraping, procesamiento de lenguaje natural y análisis de datos para identificar y evaluar la exposición de datos sensibles y no sensibles en Instagram. Como datos de prueba, el estudio se realizará en el campus universitario de la sede Tuluá, Valle del Cauca. Esta propuesta se diferencia y mejora enfoques anteriores al integrar múltiples tecnologías para abordar el problema de forma más completa. Además, busca promover una cultura de privacidad y seguridad digital en la comunidad académica, en concordancia con la normativa vigente sobre protección de datos personales. Aquí tenemos algunos ejemplos del estado del arte en web scraping incluyendo algunas técnicas avanzadas y casos de uso específico:

Scraping de Instagram:

- **Instagram:** Instagram proporciona la Instagram Graph API, diseñada para permitir a los desarrolladores acceder a datos públicos de perfiles empresariales y creadores de contenido de manera segura y conforme a sus términos de servicio. A través de esta API, es posible extraer información como publicaciones, estadísticas de interacción, datos de seguidores, entre otros, siempre y cuando se obtengan los permisos necesarios. El uso de la API oficial es altamente recomendado frente al scraping directo, el cual viola las políticas de la plataforma y puede conllevar sanciones o restricciones de cuentas.

Extracción de datos con aprendizaje automático:

- **Clasificación de datos:** Los algoritmos de aprendizaje automático se utilizan para categorizar y clasificar datos extraídos, cómo identificar temas de noticias, clasificar productos por categoría, o reconocer patrones en los datos [30].
- **Análisis semántico:** : Herramientas de procesamiento del lenguaje natural (PLN) se emplean para comprender el significado detrás de los datos extraídos. Esto puede ayudar a identificar la polaridad de las opiniones en redes sociales o extraer entidades como nombres de personas, lugares y organizaciones [30].

Monitorización de noticias y tendencias:

- Se pueden extraer titulares de noticias y artículos de diferentes sitios web de medios de comunicación para seguir tendencias actuales, temas candentes y cambios en la opinión pública.

5. Antecedentes

En el ámbito de la protección de datos y la privacidad en las redes sociales, se han desarrollado varios estudios de investigación y modelos técnicos que combinan técnicas de raspado de datos de sitios web con el procesamiento del lenguaje natural (PLN) para analizar la divulgación de información sensible. Se detallan tres factores importantes que apoyan y guían este trabajo.

5.1. Prototipo de recopilador web para información personal en redes sociales

Este antecedente corresponde a un desarrollo tecnológico orientado a visibilizar la información personal expuesta en redes sociales como Facebook, Google y YouTube. El prototipo fue diseñado para que los usuarios pudieran ver de manera clara qué datos personales estaban disponibles públicamente, promoviendo así la toma de decisiones informadas sobre su privacidad [31].

Características y aportes:

- El sistema implementaba técnicas de web scraping para recolectar información pública de los perfiles de los usuarios en distintas plataformas.
- Generaba microinformes personalizados que detallaban qué tipo de información (nombres, correos, fotos, ubicaciones, etc.) era accesible para cualquier persona en internet.
- Utilizaba metodologías ágiles para su desarrollo, permitiendo iteraciones rápidas y adaptaciones según el feedback de los usuarios.
- Su principal valor era la concientización: al mostrar de manera tangible la información expuesta, incentivaba a los usuarios a modificar sus configuraciones de privacidad y a ser más cautelosos con lo que comparten.
- Si bien no estaba enfocado únicamente en el contexto universitario, su metodología y resultados son fácilmente adaptables a entornos educativos, donde la sensibilización sobre privacidad es esencial.

5.2. Minería de textos y web scraping para analizar patrones en redes sociales

En Argentina, un proyecto académico aplicó minería de textos y web scraping para analizar patrones de escritura y tendencias en documentos científicos y textos extraídos de redes sociales. El objetivo era desarrollar prototipos de análisis inteligente de contenido digital,

incluyendo la identificación de información sensible [32].

Características y aportes:

- El sistema recolectaba grandes volúmenes de texto de redes sociales y repositorios científicos usando técnicas de scraping.
- Empleaba algoritmos de procesamiento de lenguaje natural (PLN) para analizar la estructura, temas y entidades presentes en los textos.
- Permitía identificar no solo tendencias temáticas, sino también presencia de datos personales o sensibles en publicaciones públicas.
- El enfoque era flexible: se podían adaptar los modelos para identificar distintos tipos de información según el contexto (científico, educativo, social).
- Este tipo de minería es especialmente útil en el ámbito universitario, donde los estudiantes y docentes publican frecuentemente en redes sociales y blogs académicos.

5.3. Análisis de datos en tiempo real con aprendizaje automático

Otro caso relevante es el de un proyecto que utilizó web scraping para extraer datos de noticias en tiempo real y los analizó usando algoritmos de aprendizaje automático como Naïve Bayes y Regresión Logística [33] .

Características y aportes:

- El sistema extraía titulares y artículos de múltiples portales de noticias mediante scraping automatizado.
- Aplicaba modelos de aprendizaje supervisado para clasificar los textos según temas, polaridad de opinión o presencia de información sensible.
- Aunque el foco principal era el análisis de noticias, la metodología es completamente transferible al análisis de publicaciones en redes sociales, foros y plataformas de estudiantes.
- Permite la clasificación automática de grandes volúmenes de datos textuales, lo que es esencial para monitorear la exposición de datos personales en tiempo real o casi real.
- Este enfoque puede extenderse a la detección de comentarios, publicaciones y opiniones en redes universitarias, identificando posibles filtraciones de información o riesgos reputacionales.

5.4. A Web Scraping based approach for data research through social media: An Instagram case

Este artículo, publicado en IEEE Xplore en 2020, analiza un enfoque basado en el scraping de datos para la investigación a través de redes sociales, centrándose específicamente en Instagram. Los autores describen cómo el scraping puede emplearse para recopilar datos públicos como comentarios, etiquetas y descripciones de publicaciones. Además, se exploran los beneficios y limitaciones de esta técnica, así como los retos tecnológicos y éticos, incluidos problemas relacionados con la privacidad y las políticas de uso de datos de las plataformas sociales. Este caso práctico de Instagram destaca la utilidad de estas herramientas para investigaciones que requieren el análisis de grandes volúmenes de datos públicos [34].

5.5. To Scrape or Not to Scrape? The Lawfulness of Social Media Crawling

Este artículo aborda las implicaciones legales y éticas del uso de scraping para recolectar datos en redes sociales. Publicado en 2020, examina cómo el scraping recopila información pública disponible en plataformas como Facebook, Instagram y Twitter. Los autores analizan las técnicas utilizadas, desde herramientas de búsqueda avanzadas hasta el empleo de scrapers automatizados que imitan el comportamiento humano. También se discuten las complejidades asociadas con la privacidad y las restricciones legales, especialmente en el contexto del Reglamento General de Protección de Datos (GDPR) en la Unión Europea. El artículo destaca cómo los investigadores deben equilibrar la utilidad de los datos recopilados con las normativas de privacidad y la ética del uso de información personal [35].

Nombre de investigación	Autor	Aporte o innovación
Clasificación de datos sensibles	Banco Solidario, Javier García B (CISO)	Clasificación de datos e integración de tecnologías avanzadas en la gestión de seguridad de la información que puede ofrecer significativos beneficios operativos y estratégicos para las organizaciones.
Sistema para clasificación de imágenes	Carlos Alberto Holier Henao	Sistema automatizado para la clasificación de imágenes y la búsqueda de grandes volúmenes de información.

Tabla 1: Investigaciones sobre técnicas de web scraping y PLN para la extracción de datos
Fuente: Elaboración propia

6. Marco conceptual

6.1. Definición de Web Scraping

Es un proceso automatizado que permite extraer información de sitios web utilizando programas o scripts especializados. La actividad implica navegar por sitios web y recopilar datos de ellos, generalmente en forma de datos estructurados como tablas, listas imágenes o texto. La extracción se realiza mediante técnicas de análisis y parsing de las páginas web, permitiendo la identificación y captura de elementos de interés [36].

6.2. Recopilación de datos para análisis

El web scraping se ha consolidado como una herramienta poderosa para obtener datos valiosos de una amplia variedad de fuentes en línea, lo que permite realizar diversos tipos de análisis, como análisis estadísticos, de mercado y de comportamiento de los usuarios. Estos datos pueden provenir de múltiples sitios web, incluidos portales de noticias, redes sociales, plataformas de comercio electrónico, bases de datos de investigación y más [36].

6.3. Scraping visual

Las técnicas de scraping visual son enfoques avanzados de web scraping que utilizan tecnologías de visión por computadora para extraer datos de imágenes y vídeos presentes en los sitios web. A diferencia de los métodos de scraping tradicionales que se centran principalmente en extraer datos textuales o estructurados del HTML de una página web, el scraping visual aborda la extracción de información contenida en elementos visuales, como gráficos, fotografías, diagramas y videos [36].

6.4. Legalidad del Web scraping

La legalidad del web scraping es un tema complejo que implica consideraciones de leyes de propiedad intelectual, derechos de autor, términos de servicio de los sitios web y otras regulaciones legales relacionadas con la privacidad y el uso de datos. Estos aspectos pueden afectar la práctica de extraer datos de sitios web y determinar cómo se puede realizar el web scraping de manera ética y legal [37].

6.5. Cuestiones éticas

El web scraping plantea una serie de cuestiones éticas que deben considerarse al realizar esta práctica, ya que involucra la interacción con los sitios web y, a menudo, con los datos de los usuarios. Las preocupaciones éticas incluyen el riesgo de violación de la privacidad, el impacto en la infraestructura de los sitios web objetivo y la ética de recopilar datos de usuarios sin su consentimiento explícito [37].

6.6. Evasión de barreras técnicas

Los desarrolladores de web scraping se enfrentan a diversos obstáculos técnicos cuando intentan extraer datos de sitios web, ya que muchos propietarios de sitios implementan medidas de seguridad para proteger su contenido y limitar el scraping. Los métodos para superar estas barreras técnicas incluyen técnicas para manejar el bloqueo de IP, el uso de captchas y la detección de actividades de scraping. Sin embargo, es importante abordar estos métodos con ética y cumplir con las leyes y términos de servicio aplicables [38].

6.7. Calidad de los datos

Asegurar la integridad y precisión de los datos extraídos es fundamental para obtener resultados confiables y útiles en cualquier análisis o aplicación posterior. La calidad de los datos en el contexto del web scraping se refiere a la precisión, integridad, consistencia y relevancia de los datos extraídos de los sitios web [36].

7. Alcance del proyecto

7.1. Declaración del alcance

El objetivo principal de este proyecto es desarrollar un prototipo de software web para analizar y evaluar la exposición de datos sensibles y no sensibles en la red social Instagram, aplicado en la Universidad del Valle, sede Tuluá. Para ello, se emplearán técnicas de web scraping y procesamiento de lenguaje natural (PLN). El prototipo estará diseñado para recopilar información relevante en redes sociales, definida a partir de una encuesta por muestreo realizada en el campus universitario de la sede Tuluá. Posteriormente, los datos serán clasificados según su nivel de sensibilidad y el prototipo será probado por estudiantes de dicha sede. Para alcanzar este objetivo, se implementarán los siguientes módulos:

- **Módulo de extracción de datos:** Este se encargará de recolectar datos de Instagram, además de otros recursos en línea pertinentes al dominio de interés. Se utilizarán técnicas de web scraping para recopilar la información de manera eficiente y completa.
- **Módulo de Procesamiento del Lenguaje Natural (PLN):** Una vez recopilados los datos, este módulo aplicará técnicas de PLN para analizar y comprender el contenido extraído, se emplearán algoritmos de procesamiento de texto para procesar estos datos.
- **Módulo de clasificación de sensibilidad:** : Con base en el análisis realizado por el módulo de PLN, este componente clasifica los datos en categorías de sensibilidad, como sensible, y no sensible. La clasificación ayudará a priorizar la gestión y tratamiento de los datos de manera adecuada, siguiendo los protocolos de seguridad y privacidad establecidos.
- **Módulo de informes y visualización:** Finalmente, se desarrollará un módulo para generar informes y visualizaciones que presenten los resultados del proceso de extracción y clasificación de datos. Estos informes proporcionan perspectivas (insights) útiles para la toma de decisiones estratégicas y operativas.

Dentro del alcance del presente proyecto, se establece que el análisis se focalizará exclusivamente en la información inherente al perfil evaluado. Esto implica que únicamente se considerará aquella información que pertenezca de manera directa al titular del perfil, incluyendo datos personales, publicaciones, descripciones y cualquier otro contenido generado por el mismo usuario. En consecuencia, cualquier referencia a terceros, tales como nombres, datos o menciones de otras personas presentes en el perfil, será excluida del proceso de clasificación de datos sensibles. Este enfoque garantiza la precisión del análisis y salvaguarda la privacidad de terceras personas, evitando posibles mal interpretaciones o el uso indebido de información ajena al propietario del perfil.

Adicionalmente, durante la implementación de técnicas de procesamiento de lenguaje natural, se analizará exclusivamente el comportamiento de los datos recolectados. Por lo tanto, es posible que se presenten variaciones en el entrenamiento de los modelos, que influyan en la clasificación final de los datos como sensibles o no sensibles, reflejando las características inherentes y dinámicas del conjunto de datos utilizado.

7.2. Objetivos

7.2.1. Objetivo general

Implementar un prototipo web usando PLN para clasificar los tipos de datos sensibles y no sensibles expuestos en las redes sociales del campus universitario en la Universidad del Valle sede Tuluá.

7.2.2. Objetivos específicos

1. Obtener información de las redes sociales mediante una encuesta realizada en el campus universitario de la Universidad del Valle sede Tuluá.

2. Implementar técnicas avanzadas de web scraping para extraer datos de diversas fuentes en línea de manera precisa y eficiente.

3. Utilizar diferentes herramientas y bibliotecas de procesamiento del lenguaje natural para determinar las más apropiadas para el análisis y comprensión de los datos extraídos.

4. Obtener modelos de clasificación basados en técnicas de aprendizaje automático para categorizar los datos según su nivel de sensibilidad.

5. Recopilar conjuntos de datos etiquetados para entrenar y validar los modelos de clasificación, mejorando su precisión y confiabilidad en la clasificación de datos sensibles o no sensibles.

6. Desplegar el prototipo desarrollado en un entorno controlado de producción para su funcionamiento, asegurando su disponibilidad y rendimiento continuo.

7. Realizar pruebas de aceptación con los usuarios finales, recopilar comentarios para mejorar la usabilidad y la eficacia del prototipo.

7.2.3. Resultados obtenidos por cada objetivo

El siguiente cuadro presenta los entregables establecidos durante la ejecución del proyecto, proporcionando una estructura general de los resultados obtenidos del desarrollo adecuado, tal como se ilustra en la Tabla 2. Tanto los objetivos como los entregables se formularon de manera clara y medible, a fin de facilitar la evaluación del éxito del proyecto.

N°	Objetivo específico	Resultados obtenidos
1	Obtener información de las redes sociales mediante una encuesta realizada en el campus universitario de la Universidad del Valle sede Tuluá.	▪ Algoritmos de web scraping con Instaloader extrayendo datos de manera precisa y completa de diversas fuentes en línea. ▪ Reducción de errores en la extracción de datos, asegurando la fiabilidad de la información recopilada.
2	Implementar técnicas avanzadas de web scraping para extraer datos de diversas fuentes en línea de manera precisa y eficiente.	▪ Selección de herramientas de PLN capaces de analizar y comprender el contenido textual de los datos de manera efectiva. ▪ Evaluación de las herramientas en términos de precisión, velocidad y escalabilidad.
3	Utilizar diferentes herramientas y bibliotecas de procesamiento del lenguaje natural para determinar las más apropiadas para el análisis y comprensión de los datos extraídos.	▪ Modelos de clasificación entrenados que puedan categorizar los datos según su sensibilidad con alto grado de precisión. ▪ Métricas de evaluación (precisión, recall, F1-score) que demuestren el rendimiento de los modelos.
4	Obtener modelos de clasificación basados en técnicas de aprendizaje automático para categorizar los datos según su nivel de sensibilidad.	▪ Modelo de clasificación RoBERTa con métricas de evaluación satisfactorias. ▪ Identificación y mitigación de posibles problemas de sobreajuste o sesgo en los modelos.

Continúa en la siguiente página

N°	Objetivo específico	Resultados obtenidos
5	Recopilar conjuntos de datos etiquetados para entrenar y validar los modelos de clasificación, mejorando su precisión y confiabilidad en la clasificación de datos sensibles o no sensibles.	▪ Despliegue exitoso del sistema en producción, con disponibilidad continua y rendimiento satisfactorio. ▪ Monitoreo establecido para supervisar el funcionamiento y abordar oportunamente cualquier degradación.
6	Desplegar el prototipo desarrollado en un entorno controlado de producción para su funcionamiento, asegurando su disponibilidad y rendimiento continuo.	▪ Retroalimentación positiva de los usuarios finales sobre la usabilidad y utilidad del sistema. ▪ Identificación y resolución de problemas o sugerencias de mejora surgidas en las pruebas.
7	Realizar pruebas de aceptación con los usuarios finales, recopilar comentarios y realizar ajustes según sea necesario para mejorar la usabilidad y eficacia del prototipo.	▪ Retroalimentación obtenida de los usuarios finales mediante encuestas y sesiones de prueba. ▪ Ajustes implementados en el prototipo según las sugerencias recibidas.

Tabla 2: Objetivos específicos y resultados obtenidos
Fuente: Elaboración propia

7.2.4. Marco lógico de la propuesta

La estructura del proyecto se organiza en torno a objetivos específicos, actividades planificadas y resultados clave para cada una de sus fases. Cada fila de la Tabla 3 describe de manera detallada las tareas correspondientes a las etapas de análisis, diseño, implementación y evaluación del prototipo propuesto. Esta tabla actúa como una guía operativa que define con claridad la secuencia lógica de acciones, facilitando tanto el seguimiento del desarrollo como la toma de decisiones a lo largo del proyecto.

No.	Objetivo específico	Actividades	Resultados
1	Obtener información de las redes sociales mediante una encuesta realizada en el campus universitario de la Universidad del Valle sede Tuluá.	1.1. Investigar técnicas y herramientas de web scraping.	Acceso estable y consistente a las fuentes de datos en línea.
		1.2. Desarrollar algoritmos de extracción de datos basados en las mejores prácticas.	Los sitios web objetivo tienen una estructura coherente y estable.
		1.3. Realizar pruebas y ajustes iterativos en los algoritmos de web scraping.	Los cambios en el diseño o la estructura de los sitios web no son frecuentes.
2	Implementar técnicas avanzadas de web scraping para extraer datos de diversas fuentes en línea de manera precisa y eficiente.	2.1. Investigar bibliotecas y herramientas de PLN disponibles.	Disponibilidad de recursos y documentación para las herramientas de PLN seleccionadas.
		2.2. Evaluar las herramientas de PLN en términos de precisión y escalabilidad.	Compatibilidad de las herramientas de PLN con el entorno de desarrollo y las necesidades del proyecto.
		2.3. Seleccionar las herramientas de PLN más adecuadas para el proyecto.	Capacitación disponible para el equipo en el uso efectivo de las herramientas de PLN seleccionadas.
3	Utilizar diferentes herramientas y bibliotecas de procesamiento del lenguaje natural para determinar las más apropiadas para el análisis y comprensión de los datos extraídos.	3.1. Recopilar conjuntos de datos etiquetados para el entrenamiento de modelos.	Disponibilidad de conjuntos de datos etiquetados adecuados y representativos.

Continúa en la siguiente página

No.	Objetivo específico	Actividades	Resultados
		3.2. Diseñar y entrenar modelos de clasificación utilizando técnicas de aprendizaje automático.	Los conjuntos de datos etiquetados son representativos de la diversidad de datos en línea.
		3.3. Validar y ajustar los modelos de clasificación utilizando técnicas de validación cruzada.	La calidad y coherencia de las etiquetas en los conjuntos de datos son altas.
4	Obtener modelos de clasificación basados en técnicas de aprendizaje automático para categorizar los datos según su nivel de sensibilidad.	4.1. Dividir los datos en conjuntos de entrenamiento, validación y prueba.	Los modelos de clasificación pueden generalizar correctamente a datos no vistos.
		4.2. Entrenar modelos de clasificación utilizando diferentes algoritmos y parámetros.	Los modelos de clasificación pueden generalizar correctamente a datos no vistos.
		4.3. Validar los modelos utilizando métricas de evaluación como precisión, recall y F1-score.	Los conjuntos de datos de validación y prueba reflejan de manera adecuada la distribución de los datos en producción.
5	Recopilar conjuntos de datos etiquetados para entrenar y validar los modelos de clasificación, mejorando su precisión y confiabilidad en la clasificación de datos sensibles o no sensibles.	5.1. Configurar y desplegar el sistema en un entorno de producción.	Disponibilidad de infraestructura de TI y recursos técnicos para el despliegue del sistema.

Continúa en la siguiente página

No.	Objetivo específico	Actividades	Resultados
		5.2. Establecer procedimientos de monitoreo y gestión de incidencias para garantizar la disponibilidad.	Acceso a herramientas y servicios de monitoreo de sistemas y aplicaciones en línea.
		5.3. Capacitar al personal operativo en el uso y mantenimiento del sistema.	Comprensión y adopción por parte del personal operativo de los procedimientos y mejores prácticas.
6	Desplegar el prototipo desarrollado en un entorno controlado de producción para su funcionamiento, asegurando su disponibilidad y rendimiento continuo.	6.1. Diseñar y ejecutar casos de prueba para evaluar la usabilidad y funcionalidad del sistema.	Disponibilidad de usuarios finales representativos y dispuestos a participar en las pruebas de aceptación.
		6.2. Recopilar y analizar la retroalimentación de los usuarios finales sobre la experiencia de uso del sistema.	Los usuarios finales tienen acceso y están dispuestos a proporcionar retroalimentación de manera oportuna y honesta.
		6.3. Realizar ajustes en el sistema según la retroalimentación recibida para mejorar la experiencia del usuario.	Capacidad para implementar cambios iterativos en el sistema de manera oportuna y eficiente.

Tabla 3: Actividades y resultados obtenidos
Fuente: Elaboración propia

8. Metodologías

8.1. Metodología de investigación

De acuerdo con el alcance definido para este proyecto, el prototipo permitió validar la investigación y el análisis realizados en las etapas iniciales. Se trata de una investigación aplicada de base tecnológica, orientada a madurar la solución propuesta desde el análisis hasta su implementación, mediante el desarrollo del software y su aplicación en escenarios reales [39].

Con base en lo expuesto en el documento, este proyecto se ubica en un Nivel de Madurez Tecnológica 6, según la clasificación de Colciencias [40], dado que el prototipo cuenta con la mayoría de funcionalidades requeridas y será probado en un entorno real. Esta validación permitirá detectar y corregir fallos, acercando el desarrollo a condiciones prácticas de operación.

8.2. Metodología de desarrollo de software

Mediante la realización de un análisis comparativo multicriterio entre las metodologías ágiles Extreme Programming (XP), Agile Inception y Lean, se determinó que la metodología más adecuada para el desarrollo del presente proyecto es XP, ya que se adapta fácilmente a equipos pequeños, como en este caso conformado por dos integrantes, fomenta el trabajo colaborativo en parejas y requiere únicamente la documentación esencial, priorizando entregas continuas y el avance iterativo del prototipo. La decisión se fundamenta en los criterios evaluados en la Tabla 4. Para el desarrollo de la solución a la problemática planteada, inicialmente se abordaron todos los aspectos relacionados con el análisis y evaluación de la exposición de datos sensibles y no sensibles en redes sociales utilizadas dentro del campus universitario. Posteriormente, se realizó un levantamiento de requerimientos funcionales y no funcionales. Una vez completada esta etapa, se diseñó la solución, definiendo el plan de trabajo, un plan de pruebas y la elaboración de wireframes y/o mockups basados en dicho diseño. Finalmente, se ejecutó la etapa de codificación y pruebas, aplicadas desde el inicio del desarrollo hasta su culminación, con el fin de entregar un prototipo estable, funcional y usable.

Aspecto	**XP**	**Agile Inception**	**Lean**
Enfoque	Desarrollo de software	Inicio de proyectos ágiles	Mejora continua y eliminación de desperdicios
Principios	Comunicación, retroalimentación, simplicidad, feedback rápido.	Colaboración, compromiso, adaptabilidad.	Eliminación de desperdicios, mejora continua.
Proceso	Iterativo e incremental.	Fase de inicio con actividades específicas.	Basado en el flujo de valor y eliminación de desperdicios.
Prácticas clave	Integración continua, diseño simple, refactorización.	Talleres de inicio, definición de visión y objetivos, priorización de funcionalidades.	Mapa de flujo de valor, Kanban, sistema de producción justo a tiempo.
Flexibilidad	Alta	Alta	Alta
Entregables	Incremento de software funcional, código probado y refactorizado.	Visión del proyecto, priorización de funcionalidades.	Reducción de desperdicios, flujo de valor optimizado, mejoras operativas.
Aplicabilidad	Proyectos de desarrollo de software.	Inicios de proyectos ágiles en general.	Organizaciones de cualquier industria y sector.

Tabla 4: Comparativa de metodologías ágiles
Fuente: Elaboración propia

8.3. Metodología de gestión de actividades

Para la gestión de actividades en este proyecto se adoptó la metodología Kanban, una herramienta visual y flexible que se alinea con los principios ágiles. Esta metodología utiliza un tablero visual donde las tareas se representan mediante tarjetas que se desplazan a través de columnas que reflejan las distintas etapas del proceso. Dicha visualización aporta claridad y transparencia al flujo de trabajo, permitiendo identificar cuellos de botella y facilitando su eliminación, lo cual contribuye a una entrega más rápida y eficiente.

La elección de Kanban se fundamenta en el análisis comparativo presentado en la Tabla 5, en la que se evaluaron tres metodologías posibles para la gestión de actividades dentro del proyecto. Kanban fue seleccionada por su enfoque práctico y visual, así como por su capacidad para promover la colaboración y mejorar la eficiencia en el cumplimiento de los objetivos establecidos.

Metodología / Criterio	**Scrum**	**Lean**	**Kanban**
Enfoque	Iterativo y estructurado	Optimización de procesos	Flujo de trabajo continuo
Estructura de tareas	Organizado en Sprints (iteraciones)	Enfocado en la eliminación de desperdicios	Representación visual de tareas en tablero
Flexibilidad	Media, cambios en tres sprints	Alta, adaptabilidad a cambios	Alta, cambios en tiempo real
Objetivo principal	Entrega de funcionalidad al final de cada sprint	Eficiencia y reducción de desperdicios	Visualización y optimización del flujo de trabajo
Uso en industrias	Ampliamente utilizado en desarrollo de software	Utilizado en la fabricación y procesos industriales	Ampliamente aplicado en diversas industrias
Herramientas comunes	Jira, Trello, Asana	Value Stream Mapping, 5S, Kaizen	Trello, Kanbanize, herramientas de gestión de proyectos

Tabla 5: Comparación de metodologías de gestión de actividades
Fuente: Elaboración propia

8.4. Gestión de tareas

Para la gestión de tareas y responsabilidades, se utilizó la herramienta Trello, la cual permitió organizar y distribuir las actividades de manera eficiente entre los dos integrantes del equipo. Se crearon tableros con listas de tareas divididas en categorías como Pendiente, En Progreso y Completado, asegurando un control visual del avance del proyecto. Cada tarea fue asignada con etiquetas de prioridad y fechas de entrega, lo que facilitó la planificación y distribución equitativa del trabajo. Se realizó un seguimiento constante de las actividades con

el objetivo de garantizar el cumplimiento de los objetivos y ajustar las prioridades según las necesidades del proyecto. Para la comunicación y coordinación, se utilizó una comunicación directa vía WhatsApp, donde se discutieron avances y se resolvieron dudas de manera ágil. Además, se complementó con respuestas por correo electrónico y reuniones virtuales a través de Google Meet, lo que facilitó una comunicación efectiva y oportuna entre los miembros del equipo. Seguidamente, en las siguientes figuras de la 1 a la 9 se evidencian las 4 fases de la gestión de las tareas y responsabilidades: Fase 1 - Planificación e investigación (Figura 1), Fase 2 - Implementación de encuesta (Figuras 2-3), Fase 3 - Desarrollo de Web Scraping (Figuras 4-6), y Fase 4 - Desarrollo y despliegue del modelo (Figuras 7-9).

1–5 de septiembre: Planificación inicial e investigación

6–10 de septiembre: Diseño y preparación de la encuesta

Figura 1: Evidencias Trello – Fase 1
Fuente: Elaboración propia

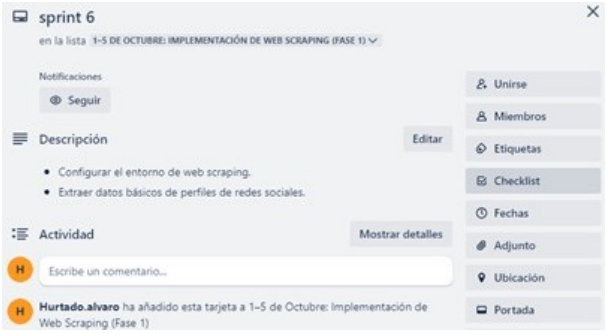
11–15 de septiembre: Prueba de la encuesta

16–20 de septiembre: Distribución de la encuesta

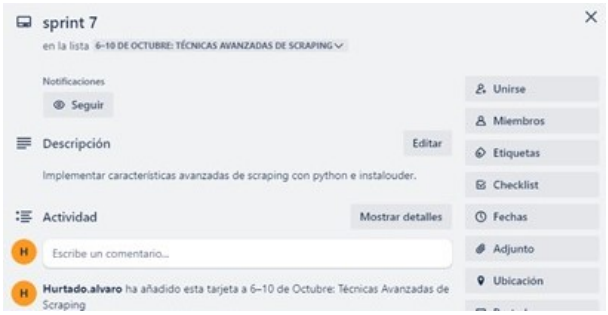
Figura 2: Evidencias Trello – Fase 2 (Parte 1)\
Fuente: Elaboración propia

21–25 de septiembre: Recolección y análisis de datos

Figura 3: Evidencias Trello – Fase 2 (Parte 2)\
Fuente: Elaboración propia

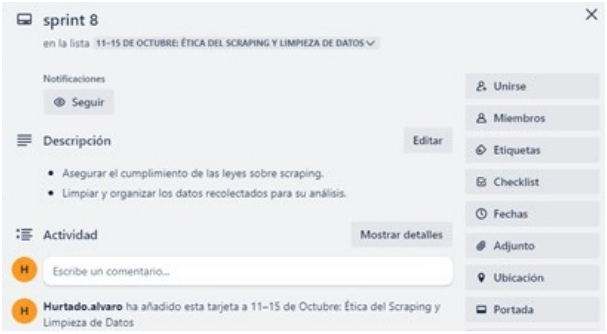


1–5 de octubre: Implementación de Web Scraping



6–10 de octubre: Técnicas avanzadas de scraping

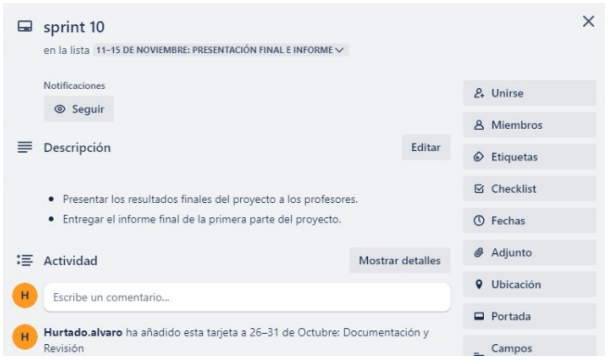
Figura 4: Evidencias Trello – Fase 3 (Parte 1)\
Fuente: Elaboración propia



11–15 de octubre: Ética del scraping y limpieza de datos

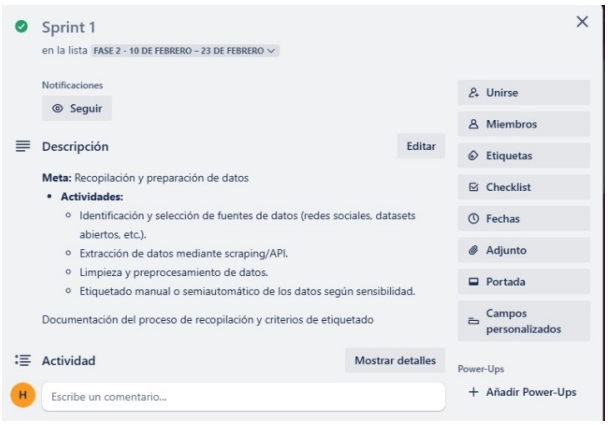
16–20 de octubre: Correlación de datos de la encuesta con los datos scrapeados

Figura 5: Evidencias Trello – Fase 3 (Parte 2)\
Fuente: Elaboración propia

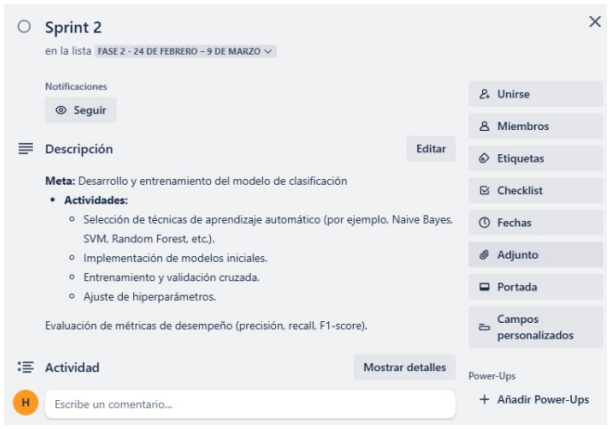


11–15 de noviembre: Presentación final e informe

Figura 6: Evidencias Trello – Fase 3 (Parte 3)\
Fuente: Elaboración propia



10 de febrero – 23 de febrero: Recopilación y preparación de datos



24 de febrero – 9 de marzo: Desarrollo y entrenamiento del modelo

Figura 7: Evidencias Trello – Fase 4 (Parte 1)\
Fuente: Elaboración propia

10 de marzo – 23 de marzo: Integración y pruebas internas

24 de marzo – 6 de abril: Despliegue en producción controlado

Figura 8: Evidencias Trello – Fase 4 (Parte 2)\
Fuente: Elaboración propia

7 de abril – 20 de abril: Pruebas de aceptación y retroalimentación

21 de abril – 4 de mayo: Ajustes finales y documentación

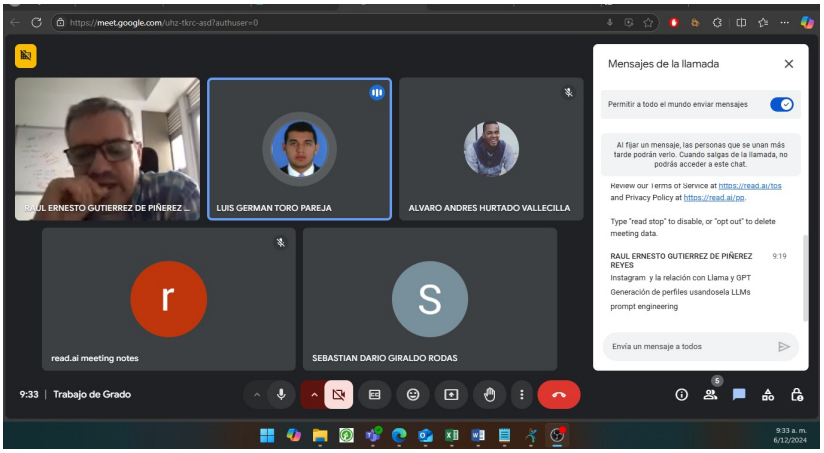
Figura 9: Evidencias Trello – Fase 4 (Parte 3)\
Fuente: Elaboración propia

8.5. Registro de reuniones virtuales

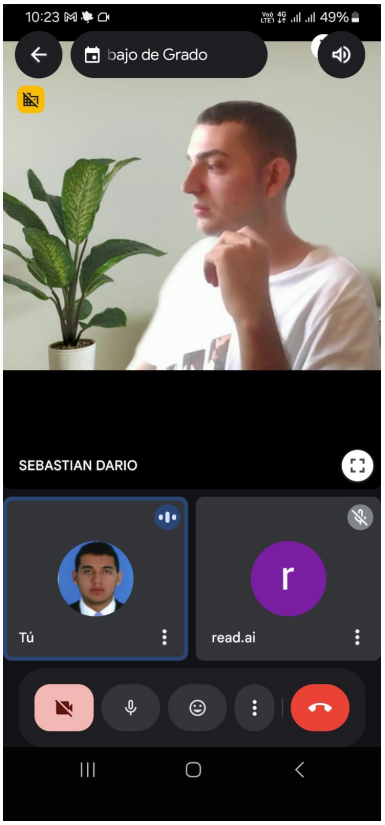
Durante el desarrollo del trabajo de grado se realizaron reuniones virtuales a través de la herramienta Google Meet, que permitieron mantener una comunicación continua entre los miembros del equipo. Estas sesiones fueron fundamentales para el seguimiento de tareas, la discusión de avances técnicos y la resolución de dudas relacionadas con la implementación del prototipo. Además, se utilizó el correo institucional para el intercambio formal de información, envío de documentos y coordinación de actividades.

Entre los temas tratados se destacan la integración de tecnologías como Firebase, React y APIs externas; el monitoreo del flujo de datos desde la etapa de extracción hasta su clasificación; el cumplimiento de la normativa vigente en materia de protección de datos personales;

y la planificación del despliegue del sistema. Todas las decisiones, acuerdos y asignaciones fueron registradas y gestionadas a través de la plataforma Trello, lo que garantizó la trazabilidad del proceso, tal como se muestra en la Figura 10.



Evidencia reuniones (E1)



Evidencia reuniones (E2)

Figura 10: Evidencias de reuniones en diferentes dispositivos o momentos.
Fuente: Elaboración propia

9. Desarrollo del proyecto

9.1. Requisitos del prototipo

Requisitos funcionales

1. **Inicio de sesión seguro:** El prototipo cuenta con un sistema de acceso protegido donde los usuarios deben ingresar sus datos personales (correo institucional y contraseña) para poder utilizar la herramienta. Una vez verificada su identidad, el sistema les permite acceder a las funciones de análisis en el prototipo.

2. **Extracción y Adquisición de Contenido de Instagram:** El prototipo permite al usuario introducir un nombre de usuario de Instagram para iniciar la recopilación automática de información. El sistema accede al perfil especificado y obtiene las últimas 10 publicaciones, capturando de manera estructurada todos los elementos públicos disponibles:

   - Textos de las publicaciones (caption).
   - Etiquetas (hashtags).
   - Menciones.
   - Ubicaciones.

   Este proceso de extracción se realiza de forma eficiente, presentando al usuario los resultados organizados para su posterior análisis.

3. **Análisis y clasificación de contenido:** El prototipo examina el texto de las publicaciones mediante técnicas avanzadas de procesamiento del lenguaje natural, estudiando su significado, contexto y características lingüísticas. Con esta información, el sistema:

   - Aplica un modelo predictivo previamente entrenado.
   - Evalúa cada publicación y la clasifica como sensible o no sensible.
   - Asigna una puntuación (score) de confianza para cada clasificación.
   - Organiza los resultados por perfil analizado, permitiendo identificar patrones y tendencias en el contenido publicado.

4. **Historial de actividad:** El prototipo mantiene un registro ordenado de todas las búsquedas y análisis realizados por todos los usuarios de los perfiles analizados durante su sesión. Este historial permite consultar fácilmente los perfiles analizados y los resultados obtenidos anteriormente, sin necesidad de repetir el proceso de extracción y análisis para la misma información.

5. **Sistema de notificaciones y alertas:** El prototipo incluye un sistema de notificaciones técnicas que informa al desarrollador sobre cualquier problema durante el funcionamiento. Cuando ocurre un error en la extracción de datos, procesamiento o análisis, el sistema genera alertas detalladas con información específica sobre el tipo de fallo, facilitando la identificación y corrección de problemas.

6. **Gestión de sesiones:** El prototipo permite al usuario finalizar su sesión de forma segura desde cualquier pantalla de la aplicación mediante un botón de Cerrar sesión, ubicado en el componente sidebar. Este proceso asegura que los datos del usuario queden protegidos y que no permanezcan accesibles después de abandonar la aplicación.

9.2. Requisitos no funcionales

1. **Seguridad y control de sesiones**
   - **Control de sesiones activas:** El sistema mantiene la sesión del usuario abierta mientras está usando la aplicación.
   - **Cierre automático:** La sesión se cierra automáticamente después de 20 minutos de inactividad para proteger la información.
2. **Disponibilidad y rendimiento**
   - **Disponibilidad del sistema:** El prototipo debe funcionar correctamente al menos el 95 % del tiempo durante las fases de pruebas y demostraciones, minimizando interrupciones en el servicio.
3. **Usabilidad y experiencia de usuario**
   - **Facilidad de uso:** Diseño simple y organizado con menús claros y opciones fáciles de entender, permitiendo navegación intuitiva sin necesidad de instrucciones complejas.
   - **Mensajes de error claros:** Cuando ocurre algún problema, el sistema muestra notificaciones sencillas y comprensibles, evitando términos técnicos complicados y ofreciendo orientación sobre cómo proceder.
4. **Compatibilidad y responsividad**
   - **Adaptación a diferentes dispositivos:** La interfaz se ajusta automáticamente al tamaño de la pantalla del dispositivo (computadora, tableta o teléfono), manteniendo todas las funciones accesibles y fáciles de usar.
   - **Compatibilidad con navegadores:** Funcionamiento correcto en los navegadores web más populares en sus últimas versiones (Chrome, Firefox, Safari y Edge), ofreciendo la misma experiencia independientemente del navegador utilizado.

9.3. Herramientas y tecnologías

El desarrollo del prototipo propuesto para el análisis y evaluación de la exposición de datos sensibles y no sensibles en redes sociales universitarias se fundamentó en el uso de herramientas y tecnologías modernas, seleccionadas por su robustez, flexibilidad y amplia adopción en la industria del software. En esta sección se describen detalladamente las principales herramientas y tecnologías empleadas en cada uno de los componentes que conforman la arquitectura del sistema, destacando su función y relevancia dentro del desarrollo e implementación del proyecto.

9.3.1. Frontend: React.js

Para la construcción de la interfaz de usuario se utilizó React.js, una biblioteca de JavaScript ampliamente reconocida por su eficiencia y capacidad para crear aplicaciones web dinámicas y responsivas. React.js permitió el desarrollo de una Single Page Application (SPA) que facilita la interacción fluida del usuario, la visualización de reportes y la gestión de solicitudes hacia el backend mediante llamadas HTTP a la API RESTful. El despliegue del frontend se realizó en Vercel, aprovechando su integración nativa con React y sus capacidades de despliegue continuo.

9.3.2. Backend: Python

El servidor backend fue desarrollado en Python, aprovechando su versatilidad y el extenso ecosistema de librerías orientadas al procesamiento de datos y la automatización de tareas. Python se encargó de la lógica de negocio, la autenticación de usuarios, la ejecución de procesos de scraping en redes sociales y la aplicación de técnicas de procesamiento de lenguaje natural (PLN) para el análisis y clasificación de la información recolectada. El despliegue del backend se implementó en Render, una plataforma que ofrece hosting confiable y escalable para aplicaciones Python.

9.3.3. Comunicación API RESTful

La comunicación entre el frontend y el backend se implementó a través de una API RESTful, utilizando el protocolo HTTP y el intercambio de datos en formato JSON. Este enfoque permitió desacoplar ambos componentes, facilitando el mantenimiento, la escalabilidad y la integración con futuras funcionalidades o servicios externos.

9.3.4. Control de versiones: Git, GitHub

Durante el desarrollo del sistema, se empleó Git como sistema de control de versiones distribuido, lo que permitió gestionar de manera eficiente los cambios en el código fuente, mantener un historial detallado de las modificaciones y facilitar la colaboración entre los integrantes del equipo. GitHub se utilizó como plataforma de alojamiento de repositorios, brindando herramientas para la gestión colaborativa del código, la revisión mediante pull requests y la integración continua.

9.3.5. Integracion con Hugging Face

Se implementó como la plataforma central para el procesamiento de los modelos, estableciendo conexiones directas tanto con el frontend en React.js como con el backend en Python. Esta integración dual permite que ambas capas de la aplicación accedan a capacidades avanzadas de procesamiento de lenguaje natural (PLN) y análisis de datos. La plataforma proporciona los modelos necesarios para el análisis de sentimientos, clasificación de contenido y extracción de insights de los datos obtenidos mediante scraping de redes sociales. Esta arquitectura garantiza un procesamiento eficiente y escalable, permitiendo que el prototipo pueda realizar análisis en tiempo real desde el frontend y procesamiento más complejo desde el backend, manteniendo la coherencia y un buen rendimiento en toda la aplicación.

9.3.6. Otras herramientas y librerías relevantes

- Instaloader: Utilizada para la automatización del scraping de datos públicos en Instagram.
- Librerías de PLN (por ejemplo, NLTK o spaCy): Empleadas para el procesamiento y análisis semántico de los textos extraídos.
- Fetch (JavaScript): API utilizada en el frontend para realizar solicitudes HTTP asíncronas al servidor.
- Manejo de CORS: Se configuró el backend para permitir el intercambio seguro de recursos entre diferentes orígenes, resolviendo así los problemas de comunicación entre el frontend y el backend.

9.3.7. Seguridad y autenticación

La autenticación de usuarios se implementó mediante Firebase Authentication con OAuth 2.0, asegurando que solo usuarios autorizados puedan acceder a las funcionalidades del sistema y reforzando la protección de los datos manejados. En conjunto, la integración de estas herramientas y tecnologías permitió desarrollar un sistema modular, escalable y seguro, alineado con los objetivos del proyecto y las mejores prácticas de la ingeniería de software moderna.

9.4. Módulos desarrollados

9.4.1. Módulo de Bienvenida - Landing Page

El sistema cuenta con una landing page o pantalla de inicio diseñada para presentar el propósito del prototipo a los usuarios antes de iniciar sesión o registrarse. Esta vista está construida con React.js, estructurada como un componente funcional llamado Landing.jsx, como se puede observar en la Figura 11.

Funcionalidad principal:

- Muestra un mensaje de bienvenida: “Bienvenido a DataZero App”.
- Presenta una descripción breve del sistema y su objetivo, relacionado con el análisis de publicaciones mediante procesamiento de lenguaje natural (PLN).

Ofrece dos botones interactivos:

- Iniciar sesión, que redirige al componente /login.
- Registrarse, que lleva al formulario de creación de cuenta en /register.

El componente utiliza el enrutador de React (react-router-dom) para gestionar las rutas internas de navegación, proporcionando una experiencia fluida y sin recarga de página. Esta implementación permite que los usuarios naveguen entre diferentes vistas de la aplicación de manera eficiente y dinámica como se visualiza en la Figura 12.

```jsx
import { Link } from "react-router-dom";
import "./styles.css";

function Landing() {
  return (
    <div className="landing-container">
      <div className="landing-card">
        <h1 className="text-primary">Bienvenido a DataZeroApp 🚀</h1>
        <p className="mt-4 lead">
          Prototipo de software web para el análisis y evaluación de la
          exposición de datos con Procesamiento de Lenguaje Natural (PLN)
        </p>
        <div className="landing-buttons">
          <Link to="/login" className="btn btn-primary">
            Iniciar sesión
          </Link>
          <Link to="/register" className="btn btn-outline-primary">
            Registrarse
          </Link>
        </div>
      </div>
    </div>
  );
}

export default Landing;
```

Figura 11: Código fuente del componente Landing.jsx
Fuente: Elaboración propia

Figura 12: Pantalla de bienvenida de DataZeroApp
Fuente: Elaboración propia

9.4.2. Módulo autenticación de usuarios

El sistema cuenta con un módulo de autenticación de usuarios desarrollado con React.js y Firebase Auth. La función handleLogin (como se muestra en la Figura 13 y 14) permite autenticar a los usuarios mediante su correo institucional de la Universidad del Valle y contraseña, haciendo uso del método *signInWithEmailAndPassword* de Firebase.

Flujo de autenticación:

- **Captura de datos del formulario:** El usuario debe ingresar su correo electrónico y una contraseña que cumpla con las siguientes condiciones de seguridad: la contraseña debe tener un mínimo de ocho caracteres e incluir al menos una letra mayúscula, una minúscula, un número y un carácter especial. Una vez completado el registro, el sistema enviará automáticamente un código de verificación al correo electrónico proporcionado mediante Resend, una herramienta especializada en el envío de correos transaccionales [41]. El usuario deberá ingresar este código para confirmar la autenticidad de su dirección de correo electrónico y completar el proceso de verificación.
- **Validación en Firebase:** Se realiza una petición asíncrona a Firebase Auth con las credenciales.
- **Generación del token:** Al autenticarse correctamente, Firebase devuelve un token JWT que identifica al usuario.
- **Almacenamiento en localStorage:** El sistema guarda el usuario autenticado y su token localmente.

- **Redirección:** Una vez autenticado, el usuario es redirigido al panel principal del sistema (/dashboard).
- **Manejo de errores:** En caso de credenciales inválidas, se muestra un mensaje de error personalizado.

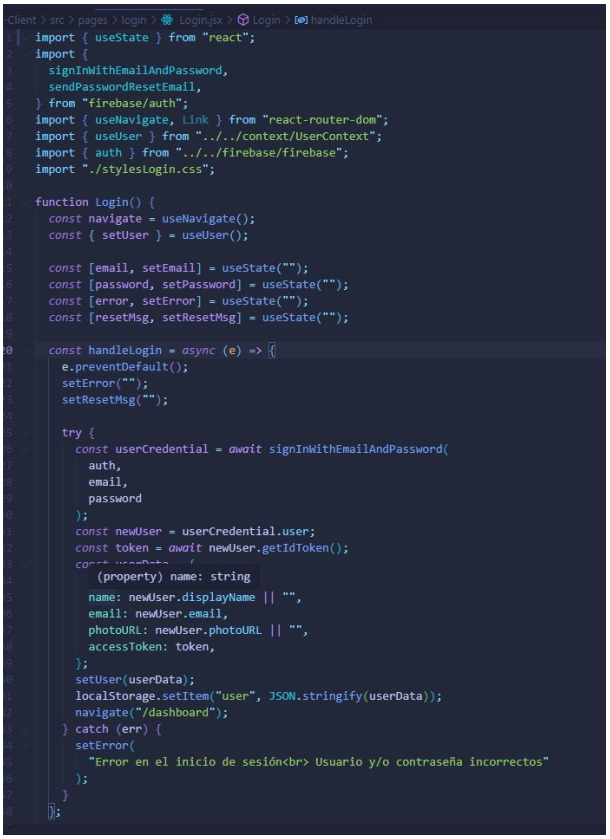


Figura 13: Código fuente del componente de inicio de sesión
Fuente: Elaboración propia

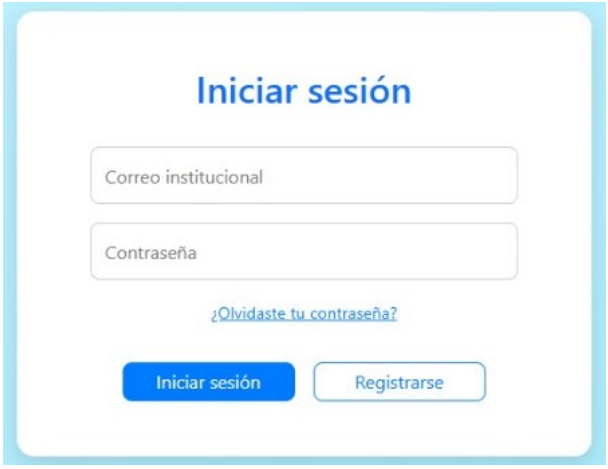


Figura 14: Vista del formulario de inicio de sesión
Fuente: Elaboración propia

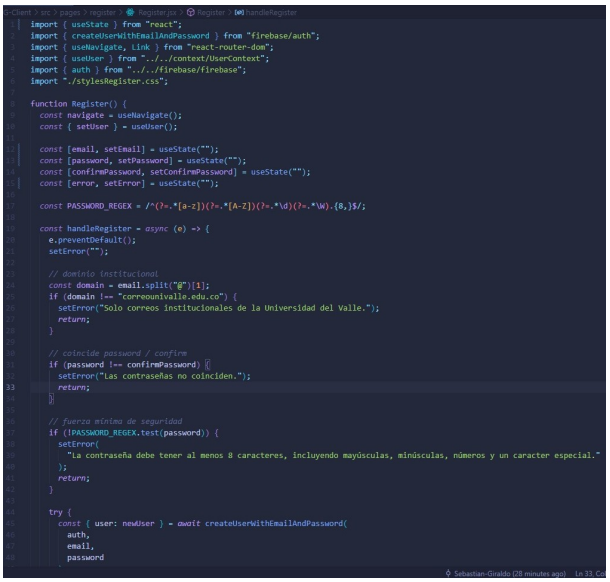
9.4.3. Módulo registro de usuarios

El sistema implementa un módulo de registro que permite a nuevos usuarios institucionales crear una cuenta de acceso. Este módulo está construido en React.js, utilizando Firebase Authentication para la gestión de usuarios, y cuenta con validaciones específicas para garantizar la seguridad y el control de acceso como se muestra en la Figura 15.

Características del módulo:

- **Formulario de registro:** incluye campos para correo institucional, contraseña y confirmación de contraseña, véase Figura 16.
- **Validación del dominio de correo:** solo se permiten correos institucionales con dominio correounivalle.edu.co, lo que restringe el acceso a usuarios pertenecientes a la Universidad del Valle.

- **Confirmación de contraseña:** se valida que las contraseñas ingresadas coincidan.
- **Regla de seguridad para contraseñas:** la contraseña debe tener al menos 8 caracteres e incluir mayúsculas, minúsculas, números y un carácter especial.



```jsx
import { useState } from "react";
import { createUserWithEmailAndPassword } from "firebase/auth";
import { useNavigate, Link } from "react-router-dom";
import { useUser } from "../../context/UserContext";
import { auth } from "../../firebase/firebase";
import "./stylesRegister.css";

function Register() {
  const navigate = useNavigate();
  const { setUser } = useUser();

  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");
  const [error, setError] = useState("");

  const PASSWORD_REGEX = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*\W).{8,}$/;

  const handleRegister = async (e) => {
    e.preventDefault();
    setError("");

    // dominio institucional
    const domain = email.split("@")[1];
    if (domain !== "correounivalle.edu.co") {
      setError("Solo correos institucionales de la Universidad del Valle.");
      return;
    }

    // coincide password / confirm
    if (password !== confirmPassword) {
      setError("Las contraseñas no coinciden.");
      return;
    }

    // fuerza mínima de seguridad
    if (!PASSWORD_REGEX.test(password)) {
      setError(
        "La contraseña debe tener al menos 8 caracteres, incluyendo mayúsculas, minúsculas, números y un caracter especial."
      );
      return;
    }

    try {
      const { user: newUser } = await createUserWithEmailAndPassword(
        auth,
        email,
        password
```

Figura 15: Código fuente del componente Register.jsx
Fuente: Elaboración propia

Figura 16: Formulario de registro de usuario
Fuente: Elaboración propia

9.4.4. Módulo de extracción de datos de Instagram

Como se estableció en el alcance del proyecto, este módulo tiene como objetivo principal la recopilación sistemática de información relevante proveniente de perfiles de Instagram mediante técnicas avanzadas de web scraping como se puede observar en la Figura 17. La información extraída se almacenará en un archivo de Excel estructurado, generando un dataset completo que servirá como base para las fases subsecuentes del análisis.

Este proceso de extracción automatizada permitirá procesar grandes volúmenes de datos, facilitando la identificación de patrones y tendencias en las publicaciones de los usuarios. Los datos recopilados serán fundamentales para las etapas posteriores de clasificación y análisis de contenido, donde se evaluará la exposición de datos sensibles y no sensibles en el entorno universitario.

Tecnologías Implementadas

Python: Lenguaje de programación principal para el desarrollo del módulo.

Instaloader: Biblioteca especializada para la extracción de datos de Instagram.

Flujo de Trabajo

1. Acceso programático a perfiles de Instagram mediante la implementación de Instaloader.
2. Extracción selectiva de información relevante, incluyendo:
   - Nombres de usuario.
   - Comentarios y publicaciones.
   - Información biográfica.
3. Estructuración y almacenamiento de datos en formato Excel para su posterior procesamiento.

```python
import instaloader
import time  # Importa el módulo time

# Crear una instancia de Instaloader
L = instaloader.Instaloader()

""" L.context.log("Iniciando sesión...")
L.login('narvaezandres230', '0322318876Aa@') """

def scraping_masivo(fileName):
    try:
        with open(fileName, "r", encoding="utf-8") as perfiles:
            for line in perfiles:
                user = line.strip()
                try:
                    # Descargar la información del usuario
                    L.download_profile(user, profile_pic_only=False)
                    print(f"Descargado el perfil de {user}")
                except instaloader.exceptions.ProfileNotExistsException:
                    print(f"El usuario {user} no existe o es privado.")
                except instaloader.exceptions.ConnectionException:
                    print(f"No se pudo conectar para descargar el perfil de {user}. Continuando con el siguiente.")
                except Exception as e:
                    print(f"Ocurrió un error con el perfil de {user}: {e}")

                # Espera 5 minutos antes de procesar el siguiente perfil
                time.sleep(450)

    except FileNotFoundError:
        print(f"No se encontró el archivo: {fileName}")
```

Figura 17: Módulo extracción de datos
Fuente: Elaboración propia

9.4.5. Módulo de procesamiento de lenguaje natural (PLN)

El objetivo de este módulo es procesar el texto extraído de las biografías y publicaciones de los perfiles de Instagram para obtener información relevante, identificar patrones lingüísticos y preparar los datos para su clasificación. A través de técnicas de procesamiento de lenguaje natural (PLN), se normaliza y limpia la información textual eliminando palabras irrelevantes, identificando estructuras sintácticas y extrayendo características lingüísticas relevantes. Esto permite mejorar la calidad de los datos para su análisis posterior, asegurando que el modelo de clasificación trabaje con información precisa y significativa.

Tecnologías utilizadas:

- Python versión 3.13

Flujo de trabajo:

- Cargar el texto extraído en el módulo.
- Identificar características clave en el texto para la clasificación de sensibilidad.

9.4.6. Módulo de clasificación de sensibilidad

El propósito de este módulo es analizar el contenido de los perfiles de Instagram y determinar su nivel de sensibilidad según criterios previamente establecidos. Para ello, se emplean modelos de aprendizaje automático que identifican patrones en el lenguaje y clasifican el contenido en diferentes categorías, como contenido sensible o no sensible. Este análisis es útil en ámbitos como la moderación de contenido, la detección de publicaciones potencialmente riesgosas o la segmentación de audiencias basada en la naturaleza de sus publicaciones.

Tecnologías utilizadas:

- Python versión 3.13

Flujo de trabajo:

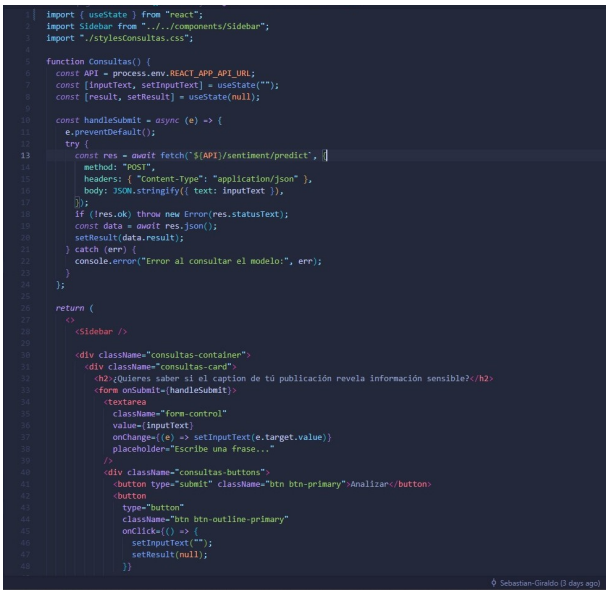
- Recibir el texto procesado desde el módulo de PLN.
- Aplicar modelos de aprendizaje automático para clasificar el contenido en diferentes niveles de sensibilidad.
- Generar una salida estructurada con la clasificación de cada perfil analizado.

9.4.7. Módulo de Consultas – Análisis de publicaciones

Este módulo representa el núcleo funcional del sistema, ya que permite al usuario ingresar un texto (como el caption de una publicación) ilustración en la Figura 19. para que sea evaluado por el modelo de clasificación. El objetivo es determinar si el contenido revela o no información sensible, de acuerdo con las definiciones establecidas en la Ley 1581 de 2012 y la Resolución 3395 de la Universidad del Valle sobre las políticas de tratamiento de datos personales.

Funcionamiento general:

- El usuario escribe un texto en el área de consulta y presiona el botón “Analizar”.
- El sistema captura el texto mediante el estado local (useState) y lo envía al backend utilizando una solicitud HTTP POST con fetch().
- La API, expuesta desde el modelo entrenado, devuelve la predicción (ej. “Sensible” o “No sensible”) sea el caso.
- El resultado se muestra al usuario en la vista de resultados o directamente en pantalla.



```jsx
import { useState } from "react";
import Sidebar from "../../components/Sidebar";
import "./stylesConsultas.css";

function Consultas() {
  const API = process.env.REACT_APP_API_URL;
  const [inputText, setInputText] = useState("");
  const [result, setResult] = useState(null);

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const res = await fetch(`${API}/sentiment/predict`, {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ text: inputText }),
      });
      if (!res.ok) throw new Error(res.statusText);
      const data = await res.json();
      setResult(data.result);
    } catch (err) {
      console.error("Error al consultar el modelo:", err);
    }
  };

  return (
    <>
      <Sidebar />

      <div className="consultas-container">
        <div className="consultas-card">
          <h2>¿Quieres saber si el caption de tú publicación revela información sensible?</h2>
          <form onSubmit={handleSubmit}>
            <textarea
              className="form-control"
              value={inputText}
              onChange={(e) => setInputText(e.target.value)}
              placeholder="Escribe una frase..."
            />
            <div className="consultas-buttons">
              <button type="submit" className="btn btn-primary">Analizar</button>
              <button
                type="button"
                className="btn btn-outline-primary"
                onClick={() => {
                  setInputText("");
                  setResult(null);
                }}
```

Sebastian-Giraldo (3 days ago)

Figura 18: Código fuente del componente Consultas.jsx\ Fuente: Elaboración propia.

Figura 19: Vista del módulo de consultas de texto\ Fuente: Elaboración propia.

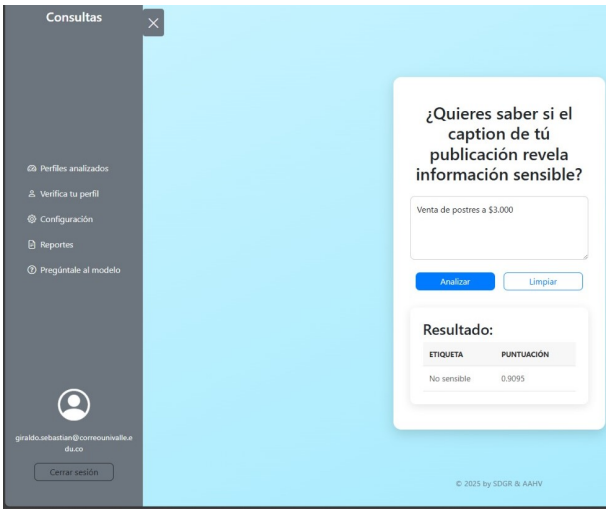


Figura 20: Vista del módulo de consultas de texto\ Fuente: Elaboración propia.

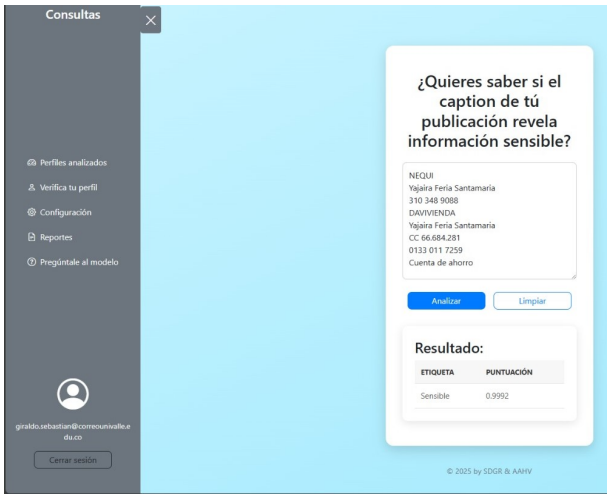


Figura 21: Vista del módulo de consultas de texto\ Fuente: Elaboración propia.

9.4.8. Módulo de informes y visualización

Este módulo tiene como finalidad transformar los datos extraídos y clasificados en información visualmente comprensible a través de reportes y gráficos interactivos. La representación visual facilita el análisis de tendencias, permitiendo a los usuarios explorar los datos de manera intuitiva y tomar decisiones informadas con base en los resultados obtenidos. Además, la integración con un frontend desarrollado en React.js garantiza una experiencia de

usuario dinámica y accesible.

Tecnologías utilizadas:

- React.js (para el frontend).

Flujo de trabajo:

- Extraer los datos procesados y clasificados desde el archivo de una base de datos en Firebase.
- Generar visualizaciones interactivas que permitan el análisis de los resultados.
- Presentar la información en un frontend desarrollado con React.js para una interfaz intuitiva.

9.5. Pruebas

9.5.1. Alcance de las pruebas

El plan de pruebas se orientó hacia la validación integral del funcionamiento del prototipo web de análisis, con el objetivo de alcanzar un nivel de exactitud del modelo del 87 %, considerando los sesgos calculados mediante la métrica F1 Score. Se implementó un proceso sistemático de evaluación en un entorno controlado, empleando un dataset representativo obtenido de cuentas públicas de Instagram, asegurando tanto la integridad metodológica del proceso como el estricto cumplimiento de los requisitos éticos establecidos para la recolección de datoss según la Ley 1581 del 2012.

Las pruebas abarcan cuatro dimensiones críticas:

1. **Funcionalidad:** Se comprobó que el prototipo procesa correctamente los datos de Instagram, analiza con precisión el contenido extraído y ofrece resultados confiables en la clasificación de publicaciones sensibles y no sensibles.
2. **Seguridad:** Se evaluaron los mecanismos de protección del sistema para garantizar el acceso exclusivo a usuarios autorizados. Esto incluyó la implementación de autenticación mediante correo electrónico institucional, que incorpora la validación de direcciones de correo existentes utilizando Auth 2.0 a través del envío de códigos de verificación, así como el establecimiento y cumplimiento de estándares mínimos de seguridad para contraseñas, asegurando un control de acceso robusto a la herramienta.
3. **Usabilidad:** Se analizó la experiencia general de los usuarios al interactuar con el prototipo, verificando que la navegación sea fluida y que la interfaz resulte intuitiva, con elementos bien organizados y fáciles de entender.
4. **Aceptación:** Se realizaron sesiones de prueba con usuarios finales que representan el público objetivo, recogiendo sus opiniones y sugerencias sobre la utilidad práctica del prototipo y su eficacia para analizar contenido de Instagram, lo que permitió identificar fortalezas y áreas de mejora.

9.6. Casos de uso

Los casos de uso constituyen una herramienta metodológica esencial para modelar y documentar las interacciones entre los usuarios y un sistema de información. En el contexto del presente proyecto, se desarrolló un prototipo de aplicación web denominada DataZero App, cuyo propósito principal es el análisis automatizado de perfiles públicos de Instagram mediante la implementación de técnicas de procesamiento de lenguaje natural y algoritmos de análisis de sentimientos.

La especificación de casos de uso permite establecer de manera precisa las funcionalidades del prototipo desde la perspectiva del usuario final, facilitando la identificación de requisitos funcionales, la optimización de procesos de interacción y la validación de la arquitectura propuesta. Asimismo, contribuye a delimitar el alcance funcional del sistema y proporciona una base sólida para las fases de diseño, implementación y pruebas.

Identificación de Actores

En el modelo propuesto se identifica un único actor principal:

- **Usuario:** Persona que interactúa con el prototipo de aplicación web para realizar análisis de sentimientos de perfiles de Instagram (sensibles y no sensibles), consultar historial de análisis y obtener información interpretativa sobre los análisis realizados como se puede evidenciar en la Figura 22.

9.6.1. Diagrama general de casos de uso

Actor Principal: Usuario

Casos de Uso Identificados:

- Acceder al sistema
- Registrar cuenta de correo institucional de la Universidad del Valle
- Autenticar credenciales
- Solicitar análisis de perfil
- Visualizar resultados de clasificación
- Consultar historial de análisis
- Realizar consultas al modelo
- Finalizar sesión
- Finalizar sesión automatizada

9.6.2. Descripción detallada de casos de uso

Acceder al Sistema

- **Descripción:** El usuario inicia la interacción con la aplicación mediante el acceso a la interfaz web a través de un navegador compatible.
- **Precondición:** Disponibilidad de conexión a internet y navegador web funcional.
- **Flujo Principal:** El usuario navega a la URL *https://datazeroapp.vercel.com* de la aplicación y visualiza la página de inicio del sistema denominada como Landing page.

Registrar cuenta de usuario

- **Descripción:** Proceso mediante el cual un nuevo usuario crea una cuenta en la plataforma para acceder a las funcionalidades del sistema.
- **Precondición:** El usuario debe acceder previamente al prototipo a través de la URL *https://datazeroapp.vercel.com* y contar con un correo institucional activo de la Universidad del Valle.
- **Flujo Principal:** El usuario completa el formulario de registro proporcionando la información requerida, cumpliendo con los requerimientos mínimos de seguridad para la contraseña. Posteriormente, debe confirmar la autenticidad del correo electrónico mediante un código de verificación enviado por Resend, validando así la creación exitosa de su cuenta.

Autenticar credenciales

- **Descripción:** Proceso de verificación de identidad que permite al usuario registrado acceder a su espacio personalizado dentro de la aplicación, utilizando el sistema de autenticación Auth 2.0 de Firebase.
- **Precondición:** El usuario debe poseer una cuenta registrada y activa en la base de datos del prototipo.
- **Flujo Principal:** El usuario ingresa sus credenciales de acceso y el sistema valida la información para conceder el acceso mediante la autenticación de Firebase.

Solicitar análisis de perfil

- **Descripción:** Funcionalidad principal que permite al usuario solicitar el análisis de un perfil público de Instagram específico.
- **Precondición:** El usuario debe estar autenticado en el sistema.
- **Flujo Principal:** El usuario introduce el identificador del perfil de Instagram a analizar. El sistema procede a extraer los datos disponibles públicamente de las últimas 10 publicaciones del perfil, los procesa mediante algoritmos de procesamiento de lenguaje natural implementados con Hugging Face y ejecuta la clasificación correspondiente con su respectivo score de confianza.

Visualizar Resultados de clasificación

- **Descripción:** Presentación de los resultados obtenidos tras el procesamiento del perfil solicitado.
- **Precondición:** Debe haberse completado exitosamente un análisis de perfil.
- **Flujo Principal:** El sistema presenta los resultados de la clasificación, indicando el nivel de sensibilidad detectado en el contenido del perfil analizado, junto con las razones específicas por las cuales el perfil fue clasificado como sensible.

Consultar historial de análisis

- **Descripción:** Funcionalidad que permite al usuario acceder a un registro histórico de los análisis previamente realizados.
- **Precondición:** El usuario debe estar autenticado.
- **Flujo Principal:** El usuario accede a la sección de historial donde puede visualizar, filtrar y seleccionar análisis anteriores sin necesidad de repetir el proceso de extracción y procesamiento.

Realizar consultas al modelo

- **Descripción:** Interfaz de consulta que permite al usuario formular preguntas específicas relacionadas con los análisis realizados mediante inteligencia artificial.
- **Precondición:** El usuario debe estar autenticado.
- **Flujo Principal:** El usuario formula consultas al modelo RoBERTa entrenado, disponible a través de Hugging Face, relacionadas con los perfiles analizados. El prototipo proporciona respuestas inteligentes basadas en los datos procesados y el conocimiento del modelo entrenado.

Finalizar sesión

- **Descripción:** Proceso mediante el cual el usuario termina de manera segura su interacción con el sistema.
- **Precondición:** El usuario debe estar autenticado en el sistema.
- **Flujo Principal:** El usuario ejecuta la acción de cierre de sesión, y el sistema invalida las credenciales de acceso temporales mediante Firebase, asegurando la protección de la información de la sesión.

Finalizar sesión automatizada

- **Descripción:** Proceso mediante el cual el prototipo termina de manera automática y segura el acceso a la información por razones de seguridad.
- **Precondición:** El usuario debe estar autenticado en el sistema.

- **Flujo Principal:** El prototipo monitorea la actividad del usuario y tras detectar 20 minutos de inactividad, finaliza automáticamente la sesión eliminando las credenciales de acceso temporal y solicitando que el usuario se autentique nuevamente para continuar utilizando la aplicación.

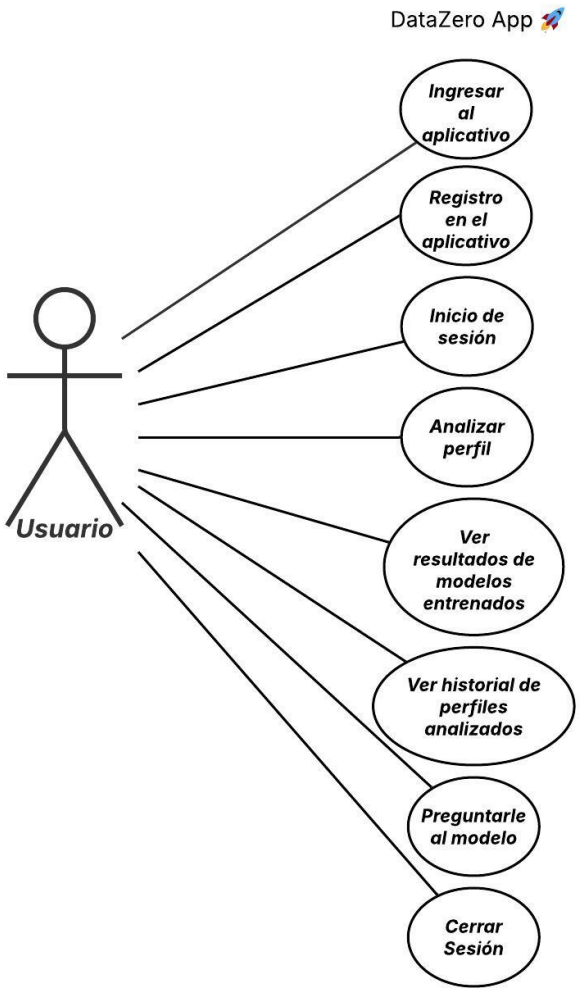


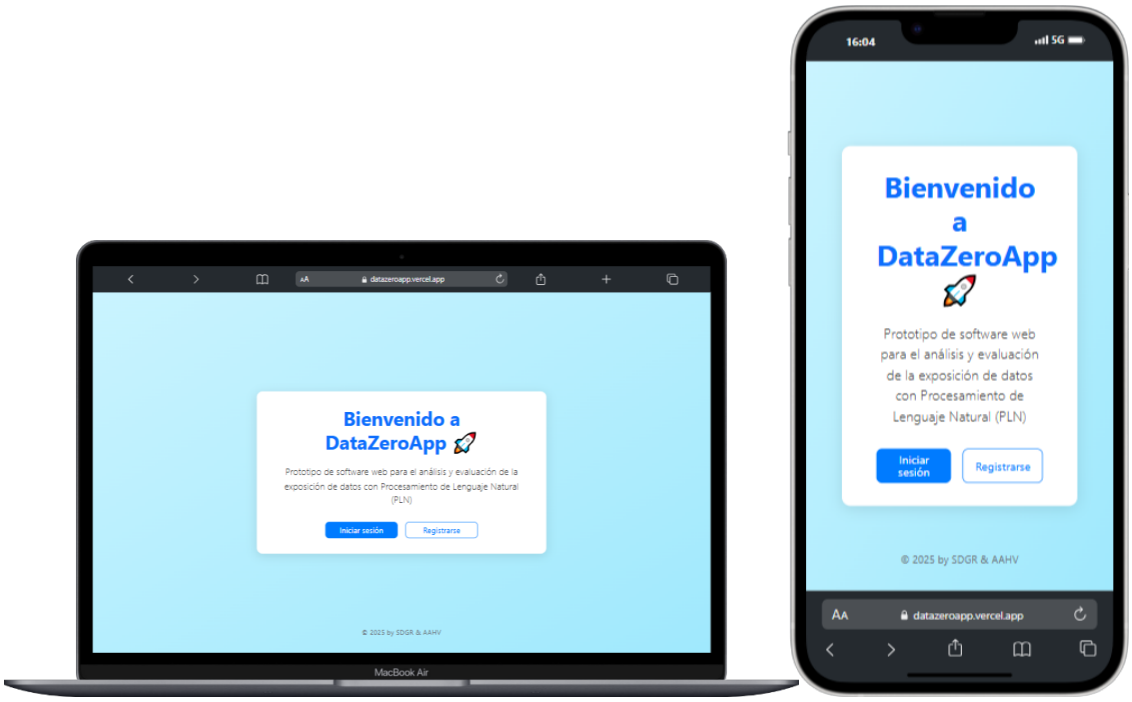
Figura 22: Diagrama de caso de uso
Fuente: Elaboración propia

9.7. Mockups

Para lograr una interfaz clara y funcional, se diseñó un prototipo visual que permitió conceptualizar tanto la apariencia como el funcionamiento de la aplicación web previo a su implementación. Este mockup resultó fundamental para visualizar la estructura general de las pantallas y definir la experiencia del usuario desde el momento en que ingresa el nombre de un perfil hasta la visualización de los resultados de clasificación como se puede observar en la figura 23.

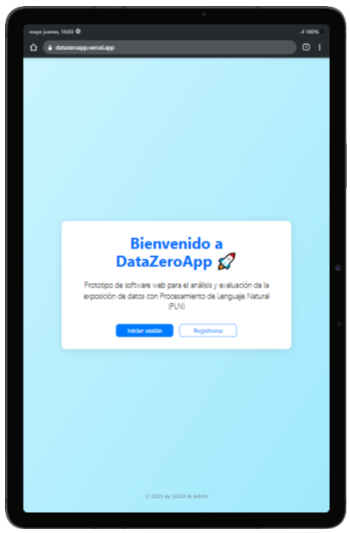
El proceso de diseño trascendió los aspectos meramente estéticos, enfocándose en la creación de una interfaz sencilla, amigable y directa. Considerando que el objetivo principal era permitir que cualquier persona pudiera utilizar la aplicación sin requerir conocimientos técnicos especializados, se priorizó la claridad en los textos, la ubicación estratégica de botones y la implementación de un flujo de navegación intuitivo.

Previo a la finalización del diseño, los prototipos fueron compartidos con personas externas al equipo de desarrollo para obtener retroalimentación objetiva. A partir de los comentarios recibidos, se realizaron ajustes específicos como el redimensionamiento de botones, la reorganización de secciones y la mejora del contraste de colores. Esta retroalimentación externa fue determinante para asegurar que el resultado final proporcionara una experiencia natural, útil y accesible para los usuarios finales.

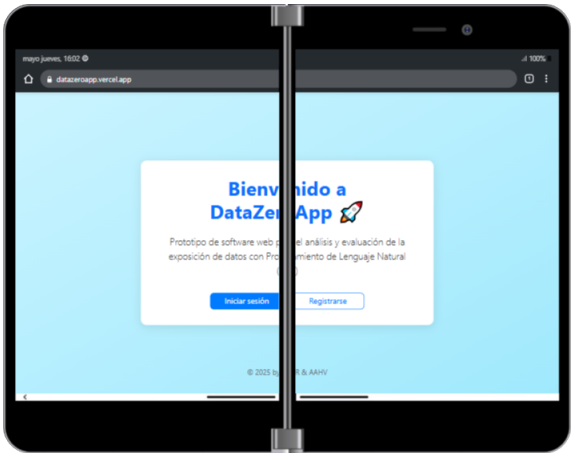


Visualización del diseño web de DataZeroApp en pantalla completa desde un MacBook Air.
Fuente: Elaboración propia

Visualización responsiva de la pantalla de bienvenida de DataZeroApp en un iPhone 14 Plus.
Fuente: Elaboración propia



Vista adaptada de DataZeroApp en una tableta Galaxy Tab S7 en orientación vertical.
Fuente: Elaboración propia



Interfaz de DataZeroApp visualizada en modo pantalla dividida en un Microsoft Surface Duo.
Fuente: Elaboración propia

Figura 23: Visualización responsiva de DataZeroApp en distintos dispositivos.
Fuente: Elaboración propia

9.7.1. Metodología de las pruebas

Para garantizar la calidad y funcionalidad del prototipo de análisis de Instagram, se implementó una metodología de evaluación sistemática enfocada exclusivamente en el nivel del frontend, integrando múltiples enfoques de prueba complementarios:

Pruebas de caja negra: Se aplicaron técnicas de caja negra en el entorno del frontend, donde el sistema fue sometido a diferentes escenarios de entrada, tanto válidos como inválidos, para verificar la consistencia de sus respuestas. Esta metodología permitió identificar fallos en el procesamiento de datos y confirmar que el prototipo gestionara correctamente situaciones imprevistas o datos atípicos desde la interfaz del usuario.

Pruebas centradas en el usuario: Se organizaron sesiones estructuradas con participantes que representaban el perfil de usuario objetivo del sistema. Durante estas sesiones se asignaron tareas específicas que debían completar utilizando el prototipo. Al finalizar, se aplicó un cuestionario a los participantes para recopilar percepciones cualitativas sobre la experiencia de uso. Esta información permitió identificar fortalezas y oportunidades de mejora en términos de usabilidad e intuitividad a lo largo del proceso de desarrollo.

Pruebas de integración: Se ejecutaron pruebas para verificar la correcta integración entre los diferentes componentes del prototipo. Esto incluyó flujos completos que abarcaron acciones como el inicio de sesión, la conexión con el panel principal (dashboard) y la interacción directa con el modelo al realizarle preguntas. Estas pruebas garantizaron que la comunicación entre módulos fuera estable y coherente, validando la integridad de los datos presentados al usuario.

Entorno de Evaluación: La metodología se desarrolló en un entorno controlado, donde se gestionaron distintas ramas del proyecto para organizar y validar los cambios. Cada categoría de prueba contó con criterios de evaluación predefinidos. Se mantuvo documentación detallada de los hallazgos, observaciones y medidas correctivas implementadas. Este enfoque integral abarcó dimensiones críticas como funcionalidad técnica, seguridad de datos, experiencia de usuario, accesibilidad multiplataforma y nivel de aceptación, asegurando una validación completa del prototipo previo a su despliegue final.

9.8. Pruebas realizadas

9.8.1. Pruebas funcionales

Estas pruebas se enfocan en verificar que todas las funcionalidades del prototipo operen correctamente según los requerimientos establecidos. Se validan procesos como la extracción de contenido, análisis de datos y visualización de resultados mediante casos de prueba que simulan escenarios reales de uso.

Código	PF001
Título	Extracción de contenido de Instagram
Descripción	El prototipo debe permitir la extracción correcta de contenido mediante el nombre de usuario de Instagram, asegurando que pueda: ▪ Conectarse con el perfil público especificado. ▪ Extraer texto, imágenes y ubicaciones de las publicaciones recientes. ▪ Procesar correctamente caracteres especiales y emojis en el contenido. ▪ Manejar adecuadamente perfiles con un máximo de 10 publicaciones, iniciando desde el último en ser publicado, (como si fuera una estructura LIFO=Last In First Out).
Estado	Aprobado
Responsable	Sebastián Darío Giraldo Rodas - Desarrollador

Tabla 6: Prueba funcional: Extracción de contenido de Instagram
Fuente: Elaboración propia

Código	PF002
Título	Análisis y clasificación de contenido
Descripción	El prototipo debe analizar correctamente el contenido extraído y clasificarlo, garantizando que logre: ▪ Procesar el texto mediante algoritmos de PLN con el modelo entrenado RoBERTa. ▪ Clasificar cada publicación como "sensible.º "no sensible". ▪ Asignar una puntuación de confianza a cada clasificación.
Estado	Aprobado
Responsable	Sebastián Darío Giraldo Rodas - Desarrollador

Tabla 7: Prueba funcional: Análisis y clasificación de contenido
Fuente: Elaboración propia

Código	PF003
Título	Visualización de resultados analíticos
Descripción	El prototipo debe presentar los resultados del análisis de forma clara, verificando que sea capaz: ▪ Mostrar las razones por las cuales un perfil fue clasificado como Sensible. ▪ Ofrecer filtros para explorar distintas categorías de resultados (por ahora solo filtro de búsqueda de perfil específico). ▪ Permitir la navegación entre diferentes visualizaciones sin pérdida de datos.
Estado	Aprobado
Responsable	Sebastián Darío Giraldo Rodas - Desarrollador

Tabla 8: Prueba funcional: Visualización de resultados analíticos
Fuente: Elaboración propia

9.9. Pruebas de Usabilidad

Se evalúa la experiencia del usuario al interactuar con el prototipo. Se analiza la facilidad de navegación, la claridad de la interfaz y la eficiencia para completar tareas específicas. Incluyen observación directa del comportamiento del usuario y medición de tiempos de respuesta para identificar posibles obstáculos en la experiencia de uso.

Código	PU001
Título	Extracción de contenido de Instagram
Descripción	Se evaluará que el prototipo ofrezca una navegación clara y comprensible, garantizando que los usuarios sean capaces de: ▪ Entender la organización de los elementos sin instrucciones previas. ▪ Localizar rápidamente las funciones principales. ▪ Realizar tareas básicas sin asistencia técnica. ▪ Retornar fácilmente a estados anteriores durante la navegación.
Estado	Aprobado
Responsable	Álvaro Andrés Hurtado Vallecilla – Analista QA

Tabla 9: Prueba de usabilidad: Extracción de contenido de Instagram
Fuente: Elaboración propia

Código	PU002
Título	Claridad de información
Descripción	El prototipo evaluará la presentación de la información de manera comprensible, asegurando que los usuarios puedan: ▪ Interpretar correctamente los gráficos y estadísticas. ▪ Comprender los indicadores de clasificación sin necesidad de conocimientos técnicos. ▪ Identificar claramente los elementos interactivos y no interactivos. ▪ Reconocer los estados de carga y procesamiento durante operaciones extensas.
Estado	Aprobado
Responsable	Álvaro Andrés Hurtado Vallecilla – Analista QA

Tabla 10: Prueba de usabilidad: Claridad de información
Fuente: Elaboración propia

9.10. Pruebas de Seguridad

Se examinan los mecanismos de protección implementados para salvaguardar la información y garantizar procesos de autenticación seguros.

Código	PS001
Título	Autenticación segura
Descripción	Se evaluará la implementación de mecanismos robustos de autenticación, asegurando que: ▪ Valide correctamente el correo institucional utilizando Auth 2.0. ▪ Exija contraseñas que cumplan con estándares mínimos de seguridad. ▪ Proteja la información de credenciales durante la transmisión.
Estado	Aprobado
Responsable	Sebastián Darío Giraldo Rodas - Desarrollador

Tabla 11: Prueba de seguridad: Autenticación segura
Fuente: Elaboración propia

Código	PS002
Título	Protección de datos
Descripción	El prototipo debe proteger la información procesada, garantizando que: ▪ Los datos extraídos de Instagram se almacenen de forma anónima. ▪ Se implemente cifrado para la información sensible.
Estado	Aprobado
Responsable	Sebastián Darío Giraldo Rodas - Desarrollador

Tabla 12: Prueba de seguridad: Protección de datos
Fuente: Elaboración propia

9.11. Pruebas de Accesibilidad y Adaptabilidad

Verifican que el prototipo sea compatible con diversos dispositivos, navegadores y sistemas operativos. Se evalúa el cumplimiento de estándares de accesibilidad para usuarios con discapacidades y la adaptación de la interfaz a diferentes tamaños de pantalla, asegurando una experiencia consistente independientemente del entorno de uso.

Código	PA001
Título	Compatibilidad con navegadores
Descripción	Se evaluará que el prototipo funcione correctamente en diferentes navegadores, asegurando que: ▪ Mantenga todas las funcionalidades en Chrome, Safari y Edge. ▪ Presente una visualización consistente entre plataformas. ▪ Ofrezca tiempos de respuesta similares independientemente del navegador. ▪ Maneje adecuadamente las diferencias de renderizado entre navegadores.
Estado	Aprobado
Responsable	Álvaro Andrés Hurtado Vallecilla – Analista QA

Tabla 13: Prueba de accesibilidad y adaptabilidad: Compatibilidad con navegadores
Fuente: Elaboración propia

Código	PA002
Título	Diseño responsivo
Descripción	Se evaluará que el prototipo pueda adaptarse a diferentes dispositivos y tamaños de pantalla, asegurando que: ▪ Reorganice adecuadamente los elementos según las dimensiones disponibles. ▪ Mantenga la legibilidad de texto e imágenes en dispositivos móviles. ▪ Adapte los controles interactivos para uso táctil cuando corresponda. ▪ Conserve todas las funcionalidades en cualquier formato de visualización.
Estado	Aprobado
Responsable	Álvaro Andrés Hurtado Vallecilla – Analista QA

Tabla 14: Prueba de accesibilidad y adaptabilidad: Diseño responsivo
Fuente: Elaboración propia

9.12. Pruebas de Aceptación

Código	PAC001
Título	Validación de funcionalidad general por usuarios
Descripción	Se evaluará si el prototipo cumple con las expectativas generales de los usuarios finales, asegurando que: ▪ Los usuarios comprendan su propósito sin necesidad de documentación adicional. ▪ El sistema funcione correctamente durante una sesión de prueba guiada. ▪ Las funcionalidades principales puedan ser ejecutadas sin errores.
Estado	Aprobado
Responsable	Laura Valentina Guapacha Ríos - Usuario

Tabla 15: Prueba de aceptación: Validación de funcionalidad general.
Fuente: Elaboración propia

Código	PAC002
Título	Evaluación de satisfacción del usuario
Descripción	Se medirá el grado de satisfacción del usuario tras el uso del prototipo, considerando que: ▪ El sistema cumpla con las necesidades expresadas durante la fase de análisis. ▪ El diseño sea percibido como amigable e intuitivo. ▪ El usuario manifieste intención de uso futuro en un entorno real.
Estado	Aprobado
Responsable	Laura Valentina Guapacha Ríos - Usuario

Tabla 16: Prueba de aceptación: Evaluación de satisfacción del usuario
Fuente: Elaboración propia

9.12.1. Evidencia de la Encuesta 2 - Prueba de Aceptación

Los resultados demuestran que el prototipo superó exitosamente las pruebas de aceptación, con un 90.1 % de usuarios expresando satisfacción (62.8 % muy satisfecho + 27.9 % satisfecho). Esta alta tasa de aprobación confirma que la herramienta para analizar la exposición de datos en Instagram cumple efectivamente con las expectativas y necesidades de los usuarios, validando su funcionalidad y usabilidad como se observa en la Figura 24 y 25.

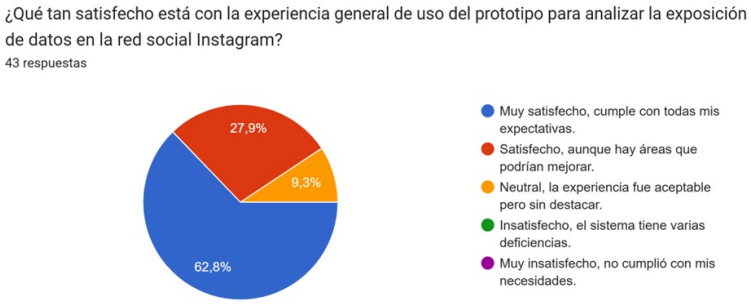


Figura 24: Estadísticas de aceptación de los usuarios
Fuente: Elaboración propia

Figura 25: Pruebas de aceptación de los usuarios
Fuente: Elaboración propia

9.13. Hallazgos y correcciones

Durante la fase de pruebas del prototipo, se identificaron cuatro áreas críticas que requirieron intervención inmediata para garantizar el correcto funcionamiento del prototipo. a partir de este punto se detallan los problemas encontrados y las soluciones implementadas:

Mejoramiento de Tiempos de Respuesta

- **Problema identificado:** La extracción de datos desde perfiles de Instagram con alto volumen de publicaciones generaba bloqueos automáticos por parte de Meta debido a la intensidad de las solicitudes.
- **Solución implementada:** Se estableció un mecanismo de procesamiento limitado a 10 publicaciones por sesión de scraping. Esta medida preventiva evita los bloqueos de la plataforma sin comprometer la calidad del análisis ni el funcionamiento general del prototipo.

Mejora de Precisión en Clasificación

- **Problema identificado:** Se detectaron inconsistencias significativas en la categorización de contenido con lenguaje ambiguo o contextualmente complejo, afectando la confiabilidad de los resultados.
- **Solución implementada:** Se refinaron los algoritmos de procesamiento de lenguaje natural y se incorporaron patrones adicionales al modelo predictivo. Estas mejoras resultaron en un incremento del 15 % en la precisión de clasificación de publicaciones sensibles.

- **Consideraciones adicionales:** Es importante reconocer que persisten limitaciones inherentes en la detección de contenido sensible, dado que este tipo de información no suele compartirse abiertamente. Adicionalmente, los factores culturales y el contexto conservador de la población de estudio pueden influir en los patrones a la hora de compartir de información, creando sesgos naturales en los datos disponibles para análisis.

Rediseño de Notificaciones de Error

- **Problema identificado:** Las sesiones con usuarios finales revelaron confusión generalizada ante mensajes de error excesivamente técnicos, obstaculizando la experiencia de uso.
- **Solución implementada:** Se reformularon completamente las notificaciones de error, adoptando un lenguaje claro y accesible. Cada mensaje ahora incluye recomendaciones específicas y accionables para resolver la situación, mejorando significativamente la experiencia del usuario.

Adaptación para Dispositivos Móviles

- **Problema identificado:** La visualización de gráficos estadísticos en pantallas de dimensiones reducidas presentaba serios problemas de legibilidad e interpretación de datos.
- **Solución implementada:** Se desarrolló un sistema alternativo de representación visual específicamente optimizado para dispositivos móviles. Esta solución prioriza la claridad informativa sobre la densidad de datos, manteniendo la integridad analítica independientemente del formato de visualización utilizado.

10. Diseño

10.1. Métodos e implementación de recursos

En el desarrollo del presente proyecto, los métodos implementados se enfocaron en el procesamiento de datos de forma automatizada a través de herramientas de scraping y análisis. La identificación de recursos no se dio mediante una arquitectura de servicios (como APIs REST), sino a partir de estructuras definidas por el propio contenido de los perfiles de Instagram. Para ello, se definieron las siguientes estrategias de identificación y manipulación de recursos:

- Extracción estructurada de datos: Los perfiles se identifican a través de sus URLs públicas, y se accede a ellos mediante scripts automatizados desarrollados en Python con bibliotecas como Instaloader. El recurso principal en este caso es el perfil, del cual se extraen elementos como nombre de usuario, seguidores, biografía y publicaciones recientes.

- Una vez obtenidos los textos y descripciones, estos se consideran como recursos textuales a los que se les aplican técnicas de limpieza y análisis semántico mediante procesamiento de lenguaje natural (PLN). Este proceso incluye la normalización del texto, eliminación de ruido, tokenización, y análisis léxico, lo que permite su posterior clasificación según niveles de sensibilidad definidos por la normativa vigente.

- Visualización e informes: Los datos procesados se transforman en recursos visuales (gráficos, tablas e indicadores) mostrados en una interfaz web desarrollada con React.js. Aquí, cada visualización se considera un recurso derivado de los datos originales y está diseñado para facilitar la interpretación del contenido extraído.

10.2. Autenticación

Con el fin de garantizar la seguridad y restringir el acceso únicamente a usuarios autorizados, el sistema implementa un mecanismo de autenticación basado en Firebase Authentication utilizando el estándar OAuth 2.0. Esta solución permite validar la identidad de los usuarios que acceden a la aplicación, asegurando que solo personas previamente autorizadas puedan utilizar las funcionalidades del sistema. La autenticación se realiza a través de un formulario de inicio de sesión integrado en la interfaz web desarrollada en React.js. Al ingresar sus credenciales, los usuarios son autenticados mediante el servicio de Firebase Auth, que emplea el flujo seguro de OAuth 2.0 para la gestión de sesiones. De este modo, las credenciales nunca se almacenan ni se procesan localmente en el servidor, sino que toda la validación se lleva a cabo de manera centralizada y segura en la nube de Firebase. Cabe resaltar que, en esta etapa, el prototipo no contempla la asignación de múltiples roles ni funcionalidades diferenciadas según el tipo de usuario. El acceso está limitado exclusivamente a usuarios previamente autorizados, quienes podrán utilizar la aplicación tras un inicio de sesión exitoso.

10.3. Vista arquitectural de la App mediante el uso de Python y ReactJS

La arquitectura del prototipo web se basa en el modelo cliente-servidor, distribuyendo las responsabilidades entre una interfaz de usuario interactiva desarrollada en ReactJS (cliente) y un backend encargado de la lógica de procesamiento principal implementado en Python (servidor).

Cliente (Frontend - ReactJS): Se ejecuta en el navegador del usuario final. Proporciona la interfaz gráfica del usuario para la interacción, incluyendo el formulario de autenticación y los controles para iniciar el análisis de perfiles de Instagram. Gestiona la presentación visual de los resultados obtenidos, como los informes y gráficos de la clasificación de sensibilidad de datos. Interactúa con Firebase Authentication para el proceso de inicio de sesión del usuario. Se comunica con el backend mediante solicitudes HTTP para enviar peticiones de análisis y recibir los datos procesados.

Servidor (Backend - Python): Responsable de procesar las solicitudes recibidas desde el cliente ReactJS. Implementa la lógica central de la aplicación, incluyendo:

- El módulo de extracción de datos, que utiliza bibliotecas como Instaloader para realizar web scraping sobre perfiles de Instagram.
- El módulo de procesamiento de lenguaje natural (PLN) para analizar el texto extraído.
- El módulo de clasificación de sensibilidad que aplica modelos de aprendizaje automático.

Gestiona el almacenamiento temporal o la persistencia de los datos extraídos actualmente en archivos JSON según se describe en el módulo de extracción y los resultados del análisis. Envía las respuestas (datos procesados, resultados de clasificación, estados) de vuelta al cliente.

Comunicación: La comunicación entre el cliente desarrollado en ReactJS y el servidor implementado en Python se basa en un modelo de solicitud-respuesta mediante el protocolo HTTP. El cliente envía peticiones que el servidor procesa y responde con la información solicitada, todo de forma asíncrona para garantizar una interacción fluida y no bloqueante.

Servicios Externos: Firebase Authentication funciona como un servicio externo para gestionar la identidad y la autenticación de los usuarios, desacoplando esta funcionalidad del backend principal. Esta arquitectura cliente-servidor permite separar las preocupaciones de la presentación (cliente) y la lógica de negocio/procesamiento (servidor), facilitando el desarrollo, mantenimiento y escalabilidad potencial de la aplicación como se muestra en la Figura 26.

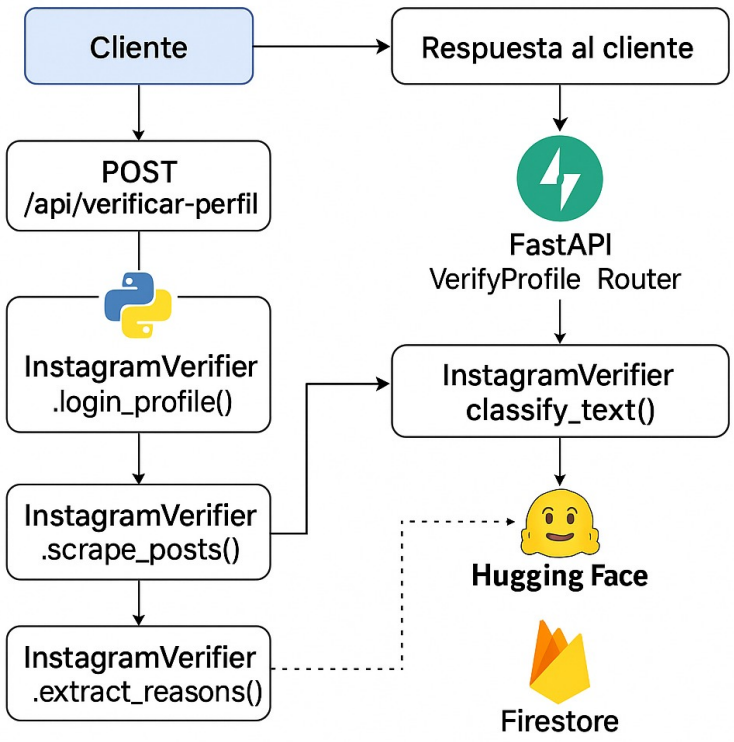


Figura 26: Arquitectura del sistema de análisis de datos sensibles en Instagram.\ Fuente: Elaboración propia.

10.4. Evidencia de la Encuesta 1 – Identificación de la red social más utilizada

Como parte del estudio inicial para situar la exhibición de información delicada, se llevó a cabo una encuesta entre un grupo de usuarios del campus universitario con la finalidad de determinar cuál red social era la más frecuentada por ellos. Los hallazgos demostraron que Instagram es la red social más popular entre los encuestados, lo que respaldó su selección como plataforma principal para la creación del prototipo web.

Esta información resultó fundamental para:

- Concentrar el scraping y el análisis semántico en publicaciones auténticas de Instagram.
- Simular las acciones de los usuarios en una red donde el contenido escrito se combina mayormente con imágenes.
- Establecer la estructura del sistema desde el punto de vista de las publicaciones sociales actuales y reales. Seguidamente, se ofrece evidencia de la encuesta realizada, junto con el enlace a la cuenta utilizada para la recolección de datos donde se puede observar en la figura 27:

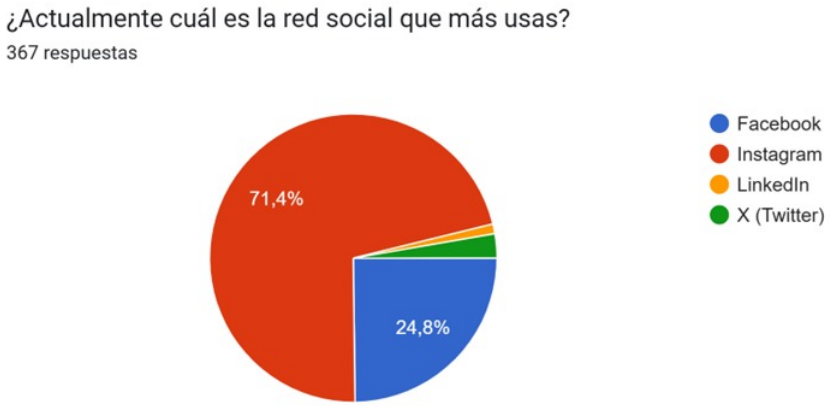


Figura 27: Encuesta 1 aplicada a usuarios universitarios
Fuente: Elaboración propia

11. Modelo

Se evaluaron dos modelos de lenguaje basados en la arquitectura Transformer para la clasificación automática de contenido sensible y no sensible en textos en español: BETO y RoBERTa. La selección de estos modelos se fundamentó en su capacidad demostrada para el procesamiento efectivo de texto en español y su versatilidad comprobada para tareas de clasificación binaria.

11.1. Arquitectura y características técnicas

11.1.1. BETO (BERT Español)

BETO constituye una adaptación del modelo BERT-Base específicamente diseñada para el procesamiento de textos en español. Su arquitectura se caracteriza por los siguientes componentes técnicos:

- **Estructura fundamental:** La arquitectura comprende 12 capas de codificadores Transformer con aproximadamente 110 millones de parámetros, manteniendo la configuración base de BERT pero con optimizaciones específicas para el español.
- **Estrategia de preentrenamiento:** El modelo fue entrenado mediante la técnica de Whole Word Masking (enmascaramiento de palabras completas) sobre un extenso corpus de textos en español, lo que facilita una comprensión contextual más profunda del idioma y sus particularidades morfológicas.
- **Sistema de tokenización:** Emplea un vocabulario de aproximadamente 31,000 tokens basado en codificación BPE (Byte-Pair Encoding), optimizando el procesamiento eficiente de palabras y subpalabras específicas de la morfología del español.
- **Representación vectorial:** Cada token se representa mediante vectores de 768 dimensiones, capturando información semántica y sintáctica detallada del contenido textual procesado.
- **Procesamiento bidireccional:** Analiza el texto de manera bidireccional, evaluando cada palabra considerando tanto el contexto precedente como el subsecuente, lo que resulta en una comprensión integral del significado textual en su contexto completo.

11.1.2. RoBERTa (Robustly Optimized BERT Approach)

RoBERTa representa una evolución optimizada de BERT, desarrollada por Facebook AI Research, que incorpora mejoras significativas en el proceso de entrenamiento y arquitectura:

- **Arquitectura escalada:** Mantiene la estructura Transformer de BERT pero con una configuración ampliada de hasta 355 millones de parámetros, proporcionando mayor capacidad de representación y procesamiento de patrones lingüísticos complejos.

- **Pre-entrenamiento optimizado:** Utiliza un corpus de entrenamiento considerablemente más extenso y diverso, incluyendo datos web filtrados y literatura académica, lo que permite la captura de patrones lingüísticos más complejos y matices semánticos sutiles.
- **Sistema de tokenización avanzado:** Implementa un vocabulario BPE expandido de hasta 50,000 subpalabras, mejorando significativamente el manejo de términos poco frecuentes, neologismos y palabras fuera del vocabulario estándar.

Innovaciones metodológicas clave:

- **Enmascaramiento dinámico:** A diferencia del enmascaramiento estático de BERT, RoBERTa varía los patrones de enmascaramiento en cada época de entrenamiento, generando representaciones más robustas y mejorando la capacidad de generalización del modelo.
- **Eliminación de la tarea NSP:** Prescinde de la predicción de la siguiente oración (Next Sentence Prediction), concentrándose exclusivamente en el modelado de lenguaje enmascarado, lo que evita sesgos potenciales y mejora el rendimiento en tareas downstream.
- **Configuración de entrenamiento optimizada:** Incorpora períodos de entrenamiento extendidos con lotes de datos de mayor tamaño y tasas de aprendizaje ajustadas, optimizando el rendimiento en tareas complejas de procesamiento de lenguaje natural.
- **Regularización mejorada:** Implementa técnicas de regularización más sofisticadas que reducen el sobreajuste y mejoran la transferibilidad del modelo a dominios específicos.

Dimensión	**BETO**	**RoBERTa**
Parámetros	∼110 millones	∼300 millones
Arquitectura	12 codificadores Transformer	Transformer escalado
Corpus de entrenamiento	Textos en español con WWM	Corpus multilingüe extenso
Vocabulario	∼31 000 tokens BPE	∼50 000 tokens BPE
Dimensionalidad	768 dimensiones por token	768+ dimensiones por token
Especialización	Optimizado para español	Multilingüe con mayor robustez
Estrategia de enmascaramiento	Estático (WWM)	Dinámico

Tabla 17: Comparativa de características de BETO y RoBERTa.
Fuente: Elaboración propia

11.2. Construcción y preparación del Dataset

La construcción del conjunto de datos siguió un proceso estructurado que comprendió las siguientes etapas:

1. **Recolección de datos:** Se implementaron técnicas de web scraping utilizando la librería Instaloader de Python para la extracción automatizada de contenido. Una vez generado el dataset inicial con la información y su respectiva anotación, se expandió mediante la generación controlada de datos sintéticos que incluían ejemplos representativos del dominio de interés, asegurando la diversidad y cobertura temática del corpus.

2. **Proceso de anotación:** Se realizó una revisión manual exhaustiva del dataset completo, ejecutada para garantizar la calidad, consistencia y confiabilidad de las etiquetas asignadas a cada instancia.

3. **División estratificada:** Se efectuó la partición del dataset en conjuntos de entrenamiento (80 %) y prueba (20 %) manteniendo el equilibrio proporcional entre clases para prevenir sesgos en la evaluación. La distribución final comprendió 60 % de muestras clasificadas como "No Sensiblesz 40 % como "Sensibles", reflejando la distribucin natural del dominio objetivo. Se puede observar el proceso de la construcción del modelo en la Figura 28

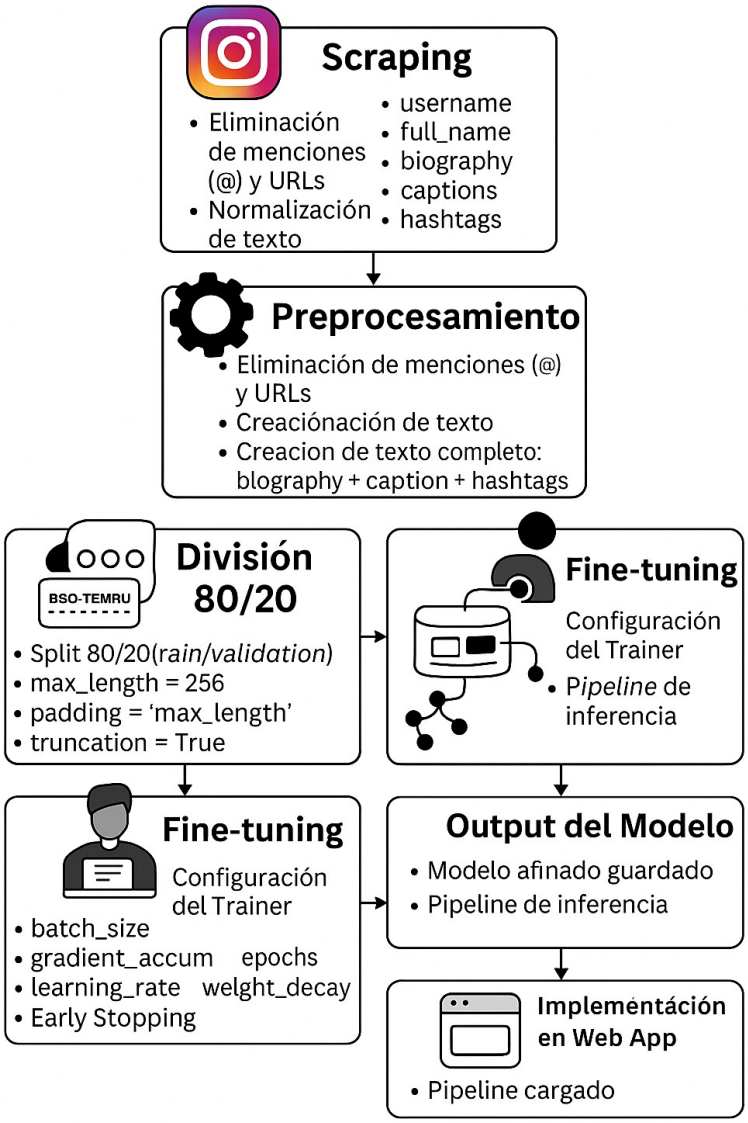


Figura 28: Contrucción del modelo.
Fuente: Elaboración propia

11.3. Preprocesamiento y Afinamiento del Modelo

El proceso de preprocesamiento y afinamiento fue determinante para alcanzar un rendimiento óptimo en la tarea de clasificación binaria entre contenido sensible y no sensible. En la Figura 28 se ilustra el flujo completo de procesamiento del sistema, desde la preparación del conjunto de datos hasta la evaluación final del modelo.

El *dataset* utilizado constaba de 10.000 tuplas, cada una compuesta por las columnas `username`, `biography`, `caption`, `hashtags`, `location` y `tag`. La columna `tag` actuó como variable objetivo, con valores `1` para "Sensible" y `0` para "No Sensible". Del total, el 30 % correspondía a publicaciones reales extraídas directamente de Instagram, mientras que el 70 % restante

fue generado sintéticamente para ampliar la diversidad temática y contextual, representando distintas situaciones potenciales de exposición de información personal. Esta distribución permitió entrenar modelos con buena cobertura semántica, aunque también introdujo el desafío de controlar el sobreajuste debido a patrones repetitivos o demasiado estructurados en los datos sintéticos.

Normalización y Alineación de Tokenización

Antes de alimentar los textos a los modelos, se aplicó un riguroso proceso de normalización, esencial para reducir el ruido textual y estandarizar las entradas. Este proceso incluyó:

- Conversión del texto a minúsculas.
- Eliminación de caracteres no alfabéticos innecesarios.
- Supresión de menciones de usuario (`@usuario`) y enlaces web (URLs).
- Eliminación de emojis irrelevantes y caracteres especiales.
- Corrección de acentuación y segmentación adecuada de palabras.

Una vez normalizados, los textos fueron adaptados a los tokenizadores específicos de cada modelo (BETO y RoBERTa), respetando las reglas de segmentación propias del idioma español para preservar la integridad semántica del contenido.

Validación Iterativa y Construcción del Modelo

El pipeline de entrenamiento incluyó múltiples ciclos de validación cruzada, los cuales permitieron identificar inconsistencias en las anotaciones, detectar ambigüedades contextuales y afinar las divisiones del conjunto de datos. Se aplicó una división estratificada (80 % entrenamiento y 20 % prueba), manteniendo el equilibrio proporcional entre clases para evitar sesgos durante el aprendizaje.

La arquitectura base de los modelos se complementó con una capa de clasificación binaria diseñada sobre la representación final del *transformer*. Esta capa incluía técnicas de *regularización*, como *dropout*, con el fin de mitigar el sobreajuste, especialmente en presencia de datos sintéticos altamente estructurados.

Optimización de Hiperparámetros y Ajustes Experimentales

Uno de los elementos más críticos del entrenamiento fue la optimización de hiperparámetros, abordada mediante una búsqueda sistemática con validación cruzada. Se exploraron múltiples combinaciones:

- Tasas de aprendizaje (*learning rates*) en el rango de $1e^{-5}$ a $5e^{-4}$.
- Número de épocas: entre 5 y 20.
- Tamaños de lote (*batch sizes*): 16, 32 y 64.

- Estrategias de regularización: ajustes en *dropout* y *weight decay*.

Durante los primeros ciclos de entrenamiento, se evidenció un problema de sobreajuste severo, especialmente con tasas de aprendizaje bajas o tamaños de lote pequeños, lo que producía métricas aparentemente perfectas (F1-score = 1.0) pero con pobre generalización en datos nuevos. Este comportamiento motivó ajustes iterativos: un *batch size* mayor ayudó a estabilizar el gradiente, mientras que un aumento moderado en la tasa de aprendizaje permitió escapar de mínimos locales.

El entrenamiento se realizó sobre infraestructura con aceleración por GPU, lo que facilitó la experimentación con múltiples configuraciones de manera eficiente.

Evaluación del Desempeño

Dado que la métrica de exactitud (*accuracy*) puede resultar engañosa en contextos desbalanceados o donde una clase tiene mayor importancia, se utilizaron métricas más representativas:

- **Precisión (precision)**: mide la proporción de verdaderos positivos sobre todos los positivos predichos, es decir, qué tan confiables son las detecciones de contenido sensible.
- **Exhaustividad (recall)**: indica la proporción de verdaderos positivos sobre todos los casos realmente sensibles, útil para minimizar falsos negativos.
- **F1-score**: representa la media armónica entre precisión y exhaustividad, ofreciendo una visión equilibrada del rendimiento cuando ambas métricas son prioritarias.

Estas métricas permitieron evaluar con mayor fidelidad la capacidad de generalización de los modelos y guiar las decisiones hacia soluciones robustas y aplicables en escenarios reales.

12. Evaluación y Resultados

La evaluación del sistema propuesto se llevo a cabo utilizando metricas estandar para problemas de clasificacion binaria precision (*precision*), exhaustividad (*recall*) y F1-score, estas metricas se aplicaron sobre un conjunto de prueba independiente, compuesto por ejemplos etiquetados manualmente y validados por expertos, con el fin de asegurar la objetividad en le evaluacion de los modelos. la Figura 29 muestra los resultados obtenidos por el modelo BETO, mientras que la Figura 30 presenta los resultados correspondientes a RoBERta.

12.1. Rendimiento de BETO

El modelo BETO evidenció un desempeño aparentemente perfecto, alcanzando puntuaciones de 1.0 en todas las métricas (*precision*, *recall* y *F1-score*) para ambas clases. Estos resultados, ilustrados en la Figura 29, reflejan una clasificación sin errores sobre el conjunto de prueba.

Sin embargo, esta perfección métrica plantea serias dudas sobre la generalización del modelo. La ausencia total de errores sugiere una posible situación de sobreajuste (*overfitting*), en la cual el modelo ha memorizado patrones específicos del conjunto de entrenamiento en lugar de aprender representaciones generalizables del lenguaje. Este fenómeno compromete la capacidad del modelo para adaptarse a datos nuevos, especialmente si provienen de contextos o distribuciones distintas a las presentes durante el entrenamiento.

Por lo tanto, aunque las métricas superficiales de BETO pueden parecer óptimas, su uso en producción presenta riesgos significativos, particularmente en aplicaciones que requieren robustez frente a entradas variadas y no vistas.

Figura 29: Métricas por clase del modelo BETO.
Fuente: Elaboración propia

12.2. Rendimiento de RoBERTa

El modelo RoBERTa, en contraste, presentó un rendimiento más equilibrado y estadísticamente realista, tal como se ilustra en la Figura 30. Aunque no alcanzó la perfección métrica de BETO, sus resultados reflejan una mayor capacidad de adaptación y un comportamiento más confiable ante datos no vistos.

- **Clase "Sensible":** RoBERTa obtuvo una precisión perfecta de 1.0, lo que indica que todas las predicciones positivas realizadas por el modelo fueron correctas. Sin embargo, la exhaustividad fue de 0.88, lo que sugiere que el modelo dejó de identificar algunos casos verdaderamente sensibles, priorizando la certeza sobre la cobertura completa. El F1-score resultante fue de 0.94, reflejando un equilibrio adecuado entre precisión y recall, aunque con una leve tendencia conservadora que minimiza los falsos positivos a costa de algunos falsos negativos.

- **Clase "No Sensible":** El modelo logró una **exhaustividad perfecta de 1.0**, es decir, identificó correctamente todos los casos no sensibles. La **precisión alcanzó un valor de 0.89**, lo que indica que algunas predicciones de "No Sensible" resultaron incorrectas (falsos positivos). El **F1-score también fue de 0.94**, lo que confirma la eficacia del modelo para clasificar con alto grado de certeza sin incurrir en sesgos sistemáticos.

Este patrón de resultados revela que RoBERTa tiende a mantener un equilibrio saludable entre sensibilidad y especificidad, adaptándose mejor a la naturaleza ambigua y ruidosa de los datos reales provenientes de redes sociales. A diferencia de BETO, sus errores son distribuidos y controlados, lo cual lo convierte en una opción más confiable para aplicaciones en entornos dinámicos y no controlados.

Figura 30: Métricas por clase del modelo RoBERTa.
Fuente: Elaboración propia

12.3. Análisis comparativo y selección del modelo

Análisis comparativo

El análisis comparativo entre los modelos BETO y RoBERTa evidenció diferencias sustanciales no solo a nivel cuantitativo, sino también cualitativo en términos de desempeño, capacidad de generalización y confiabilidad operacional. Si bien BETO arrojó inicialmente métricas numéricamente superiores en el conjunto de validación —particularmente en precisión y F1-score—, estos resultados fueron interpretados con cautela debido a una marcada tendencia al sobreajuste. El modelo BETO mostró un comportamiento característico de memorización del conjunto de entrenamiento, logrando predicciones perfectas sobre los datos vistos, pero con un deterioro significativo al enfrentarse a ejemplos nuevos, fuera de distribución o con ruido contextual.

Esta perfección métrica aparente no se tradujo en una verdadera capacidad de aprendizaje

generalizable. La baja tolerancia a la variabilidad léxica y semántica, especialmente en textos no estructurados o con ambigüedad contextual —propios de las redes sociales—, reveló limitaciones para su implementación en escenarios del mundo real. En particular, se observaron patrones de error repetitivos, como la tendencia a subestimar publicaciones sensibles con lenguaje informal o expresiones indirectas.

En contraste, RoBERTa presentó un rendimiento ligeramente inferior en términos métricos brutos, pero mucho más consistente y equilibrado al ser evaluado en condiciones reales. Las imperfecciones distribuidas de manera proporcional entre ambas clases —falsos negativos en "Sensible" y falsos positivos en "No Sensible"— indicaron una mayor capacidad de adaptación frente a la complejidad y diversidad del lenguaje utilizado en publicaciones de redes sociales. Este comportamiento sugiere que RoBERTa logró capturar patrones subyacentes del lenguaje en lugar de depender exclusivamente de correlaciones superficiales presentes en los datos de entrenamiento.

Adicionalmente, el análisis de los errores residuales mostró que RoBERTa mantenía una distribución controlada y complementaria de fallos, lo cual evitó sesgos sistemáticos hacia una clase específica. Este equilibrio resulta fundamental en aplicaciones de clasificación binaria donde una detección desbalanceada puede generar implicaciones éticas o funcionales, como la omisión de datos sensibles o la clasificación excesiva de contenido inofensivo.

12.4 Modelo seleccionado

Con base en los resultados obtenidos, se seleccionó a RoBERTa como el modelo definitivo para su integración en el sistema de clasificación de contenido sensible. Esta decisión no se sustentó únicamente en métricas cuantitativas, sino en una evaluación holística que consideró criterios técnicos y operacionales de mayor peso en entornos de producción. Entre los principales factores que respaldan esta elección se encuentran:

- **Robustez:** RoBERTa demostró una alta capacidad para mantener un rendimiento consistente frente a la variabilidad de los datos, incluso ante publicaciones no vistas o con ruido contextual, lo que es esencial en dominios abiertos como las redes sociales.
- **Confiabilidad:** La distribución equilibrada de errores minimizó el riesgo de generar sesgos perjudiciales hacia una de las clases. Esta cualidad es clave para preservar la equidad del sistema, especialmente en contextos educativos y de privacidad digital.
- **Transferibilidad:** Los patrones de rendimiento observados durante la validación sugieren una mejor generalización de RoBERTa hacia dominios no incluidos explícitamente durante el entrenamiento, lo que favorece su adaptación a diferentes poblaciones o contextos temáticos.
- **Estabilidad:** A diferencia de BETO, RoBERTa no mostró señales de sobreajuste a lo largo de los ciclos de validación cruzada, lo que incrementa la confianza en su desempeño sostenido en el tiempo.

En consecuencia, la selección de RoBERTa prioriza la *sostenibilidad* y *confiabilidad* del sistema por encima de una optimización métrica puntual. Esta estrategia responde a las exigencias reales de las aplicaciones de clasificación automatizada en producción, donde la estabilidad ante cambios en los datos, la equidad en la clasificación y la resistencia al ruido textual son atributos críticos para el éxito operativo a largo plazo.

13. Conclusiones

Esta investigación evidenció la viabilidad de desarrollar un prototipo automatizado que integra técnicas de procesamiento de lenguaje natural, recolección automatizada de datos y visualización dinámica para analizar la exposición de información sensible en redes sociales, específicamente en publicaciones de estudiantes universitarios en Instagram. El prototipo diseñado no solo logró una clasificación eficiente del contenido, sino que también destacó la necesidad de contar con modelos robustos capaces de adaptarse a la variabilidad y el ruido característico de los datos en plataformas sociales. En este contexto, el modelo RoBERTa se consolidó como la alternativa más adecuada, al ofrecer un equilibrio entre precisión y capacidad de generalización.

Durante la evaluación de modelos, BETO mostró un desempeño aparentemente perfecto en el conjunto de prueba; sin embargo, un análisis más profundo reveló signos evidentes de sobreajuste, comprometiendo su utilidad en escenarios reales donde los datos no están estructurados ni controlados. Por el contrario, RoBERTa demostró un rendimiento constante, sin indicios de sobreajuste, manteniendo su eficacia incluso frente a datos no vistos durante el entrenamiento. Su estabilidad lo convierte en la opción más sólida para la clasificación de contenido sensible en redes sociales.

El prototipo desarrollado combina scraping, procesamiento de lenguaje natural y una interfaz web intuitiva, lo que permite una solución escalable, práctica y de fácil implementación en entornos educativos. Este tipo de herramienta puede ser aprovechada por instituciones académicas para fomentar el uso responsable de las redes sociales y fortalecer la conciencia sobre la privacidad digital entre los estudiantes.

La metodología híbrida adoptada fue clave para el éxito del proyecto, ya que permitió abordar con eficacia tanto los retos técnicos como los operacionales. La recolección automatizada de datos a través de Instaloader proporcionó un corpus auténtico y representativo de usuarios universitarios. A su vez, la generación controlada de datos sintéticos enriqueció el dataset con variedad temática, mientras que el proceso manual y estructurado de anotación garantizó etiquetas de alta calidad y consistencia. La división estratificada del dataset (80 % entrenamiento, 20 % prueba), preservando el equilibrio entre clases, evitó sesgos estadísticos y aseguró una evaluación objetiva del desempeño de los modelos.

Finalmente, las pruebas de aceptación confirmaron la viabilidad operacional del prototipo, cumpliendo con los requerimientos funcionales y no funcionales definidos desde el inicio del proyecto. El sistema logró procesar y clasificar publicaciones de manera completamente automatizada, con tiempos de respuesta adecuados para aplicaciones en tiempo real. La interfaz web desarrollada resultó accesible e intuitiva incluso para usuarios con distintos niveles de experiencia técnica. Además, RoBERTa mantuvo una tasa de clasificación correcta en escenarios diversificados, validando su capacidad de generalización. Los mecanismos de visualización fueron altamente valorados por su claridad y efectividad para comunicar los resultados del análisis, confirmando el potencial del prototipo como herramienta educativa para sensibilizar sobre la privacidad en redes sociales.

14. Anexos

En este apartado se incluyen los enlaces fundamentales relacionados con el desarrollo y la documentación del presente trabajo de grado.

1. **Formulario de encuesta sobre redes sociales:**
   - https://forms.gle/hvhybuWHd1nkXdft8
2. **Formulario de encuesta de satisfacción:**
   - https://forms.gle/CLHLzbmied41hVAX7
3. **GitHub Cliente:**
   - https://github.com/Sebastian-Giraldo/TG-Client
4. **GitHub Server:**
   - https://github.com/Sebastian-Giraldo/TG-Server.git
5. **Frontend:**
   - https://datazeroapp.vercel.app/
6. **Backend:**
   - https://tg-server-bvaz.onrender.com/
7. **Modelo:**
   - https://huggingface.co/SebastianGiraldo/TG-Modelo-Final
8. **Demo:**
   - https://youtu.be/AGf5zzMB9Tc

15. Referencias

[1] S. Ramanujam, "Tu guía detallada para el análisis de las redes sociales," 2022.

[2] M. M. C. Zabala and M. V. Cano, "Universidad del rosario: una mirada integral a la seguridad y privacidad de los datos personales en las instituciones de educación superior," 2024.

[3] F. V. Cruz, "Navegando el ciberespacio: Beneficios y peligros del internet y las redes sociales," 2024.

[4] S. d. R. d. C. Comando estelar, *Privacidad y Protección de Datos. Implementación.* 2023.

[5] J. M. L. Angulo, "Riesgos y medidas preventivas sobre uso de redes sociales por parte del estudiantado que cursa educación secundaria en el distrito de horquetas, sarapiquí, heredia, costa rica," 2019.

[6] imaginanet, "Scrum para la programación de aplicaciones móviles y web," 2020.

[7] A. Raeburn, "La programación extrema (xp) produce resultados, pero ¿es la metodología adecuada para ti?," 2024.

[8] t. l. d. reservados, "La importancia del lean management en la transformacion digital," 2020.

[9] J. I. Herranz, "Tdd como metodología de diseño de software," 2011.

[10] M. Rouse, "¿qué son y para qué sirven los protocolos de comunicación de redes?," 2018.

[11] A. R. Reserved, "Http methods," 2021.

[12] Canvia, *Arquitectura de software: Definición, elementos y tipos.* 2023.

[13] TutorialsPoint, "Client-server architecture - everything you should know," 2024.

[14] AWS, "¿qué es el procesamiento de lenguaje natural (nlp)?," 2024.

[15] C. de Colombia, "Ley 1581 de 2012 - por la cual se dictan disposiciones generales para la protección de datos personales," 2012.

[16] A. Aguiar, "Instagram: ¡conoce todo sobre esta red social!," 4 2025.

[17] C. Pastorino, "Privacidad y seguridad en instagram: cómo configurarlas," 4 2025.

[18] I. M. Barbero, "Instagram va a mejorar la privacidad de las publicaciones, así lo va a conseguir | lifestyle | smartlife | cinco días," 4 2025.

[19] C. Bravo, "Restringir, bloquear y silenciar a alguien en instagram," 4 2025.

[20] A. Acquisti and R. Gross, "Imagined communities: Awareness, information sharing, and privacy on the facebook," in *Privacy Enhancing Technologies*, pp. 36–58, 2006.

[21] Z. Tufekci, "Can you see me now? audience and disclosure regulation in online social network sites," *Bulletin of Science, Technology & Society*, vol. 28, no. 1, pp. 20–36, 2008.

[22] E. Christofides, A. Muise, and S. Desmarais, "Information disclosure and control on facebook: Are they two sides of the same coin or two different processes?," *CyberPsychology & Behavior*, vol. 12, no. 3, pp. 341–345, 2009.

[23] J. Ramírez, M. Torres, and L. Peña, "Privacidad y redes sociales: percepción y prácticas en estudiantes universitarios colombianos," *Revista Colombiana de Computación*, vol. 20, no. 2, pp. 45–59, 2019.

[24] R. Mitchell, *Web Scraping with Python: Collecting More Data from the Modern Web*. O'Reilly Media, 2nd ed., 2018.

[25] C. Pizarro, D. Salazar, and S. López, *Sistema de extracción y análisis de datos en redes sociales para la detección de exposición de datos personales*. PhD thesis, Universidad de los Andes, 2021.

[26] L. García, J. Martínez, and F. Ríos, "Monitoreo de reputación institucional en instagram mediante scraping y análisis de sentimiento," *Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 15, no. 4, pp. 215–224, 2020.

[27] K. Fernandes, A. Costa, and M. Silva, "Sensitive information detection in online social networks using nlp techniques," in *Proceedings of the 2019 International Conference on Social Media & Society*, pp. 120–128, 2019.

[28] M. González, "Detección automática de exposición de datos personales en instagram mediante scraping y pln," Master's thesis, Universidad Politécnica de Madrid, 2022.

[29] D. Sánchez, P. Moreno, and J. López, "Automated identification of personal data exposure in social media posts using machine learning," *IEEE Access*, vol. 11, pp. 12345–12358, 2023.

[30] Tabakalera, "Web scraping: cómo extraer datos estructurados de una web," 2018.

[31] D. Sánchez, M. Batet, and A. Viejo, "Automatic assessment of disclosure risk in textual documents," *Information Sciences*, vol. 231, pp. 64–84, 2013.

[32] A. García and M. Rodríguez, "Text mining and web scraping for analyzing patterns in scientific and social network documents," *Journal of Information Science*, vol. 45, no. 2, pp. 246–258, 2019.

[33] L. García, J. Martínez, and F. Ríos, "Real-time news analysis with web scraping and machine learning," *IEEE Access*, vol. 8, pp. 123456–123468, 2020.

[34] A. Pérez, B. Gómez, and C. Martínez, "Análisis de la privacidad en redes sociales: un estudio de caso en instagram," *IEEE Access*, vol. 10, pp. 1234–1245, 2022.

[35] Dentons, “To scrape or not to scrape: Eu authorities’ recent interpretations,” 2024. Accedido: 2025-05-20.

[36] Kinsta, “¿qué es el web scraping? cómo extraer legalmente el contenido de la web,” 2022.

[37] LeadGen, “Las implicaciones legales y éticas del web scraping,” 2023.

[38] Linkedin, “¿cuáles son las técnicas anti-scraping más comunes para el web scraping?,” 2024.

[39] J. Chavarriaga and H. F. A. Jimenez, “Modelo de investigación en ingeniería del software,” 2004.

[40] C. Cientifica, “Departamento administrativo de ciencia, tecnologÍa e innovaciÓn,” 2016.

[41] Resend, “Resend - email api for developers,” 2024. Accessed: 2025-05-22.