Pontificia Universidad
JAVERIANA
Cali

Facultad de Ingeniería y Ciencias

Ingeniería Biomédica

Monografía de Trabajo de Grado

DESARROLLO DE UNA HERRAMIENTA QUE PERMITA LA AUTOMATIZACIÓN DE LA EXTRACCIÓN DE LA INFORMACIÓN DE ALERTAS SANITARIAS

Alejandro Córdoba Narváez
Gabriela Quintero Moreno

Director
Dra. Natalia Pedreros

5 de julio de 2025

Santiago de Cali, 5 de julio de 2025

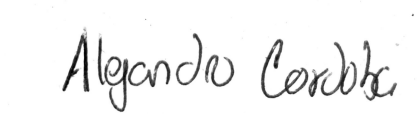
Señores
Pontificia Universidad Javeriana – Cali
Dr. Hernán Camilo Rocha Niño
Decano
Facultad de Ingeniería y Ciencias
Ciudad

Cordial Saludo.

Por medio de la presente nos permitimos presentarle el Trabajo de Grado titulado "DESARROLLO DE UNA HERRAMIENTA QUE PERMITA LA AUTOMATIZACIÓN DE LA EXTRACCIÓN DE LA INFORMACIÓN DE ALERTAS SANITARIAS".

Esperamos que este trabajo reúna todos los requisitos académicos, cumpla el propósito para el cual fue creado y sirva de apoyo para futuros proyectos relacionados con la materia.

Atentamente,



Alejandro Córdoba Narváez

Gabriela Quintero Moreno

Santiago de Cali, 5 de julio de 2025

Señores
Pontificia Universidad Javeriana – Cali
Dr. Hernán Camilo Rocha Niño
Decano
Facultad de Ingeniería y Ciencias
Ciudad

Cordial Saludo.

Certifico que el presente Trabajo de Grado titulado "DESARROLLO DE UNA HERRAMIENTA QUE PERMITA LA AUTOMATIZACIÓN DE LA EXTRACCIÓN DE LA INFORMACIÓN DE ALERTAS SANITARIAS", realizado por Alejandro Córdoba Narváez y Gabriela Quintero Moreno, estudiantes de Ingeniería Biomédica, se encuentra terminado y puede ser presentado para su sustentación.

Atentamente,

Dra. Natalia Pedreros
Director Trabajo de Grado

Agradecimientos

Quisiéramos aprovechar este espacio para agradecer a todas aquellas personas que formaron parte de nuestro proceso formativo, por ustedes somos lo que somos y sabemos lo que sabemos hoy día.

Al Dr. Hernán Vargas Cardona, quien más que un director de carrera intentó y logró ser un amigo que esta siempre presente en momentos donde y cuando sus estudiantes lo necesitan. Además, el conocimiento, no solo de su materia, si no también a nivel personal, es algo que nos ayuda a crecer como personas.

A la profesora Valentina Corchuelo, a quien le debemos gran parte del conocimiento que fue esencial para nosotros en nuestras debidas prácticas profesionales. Además, es un increíble ser humano que nos enseña a que todo en la vida tiene su recompensa.

A Nuestra directora de proyecto de grado, la Ingeniera Natalia Pedreros, por su apoyo en todo el proceso de elaboración del presente proyecto, su acompañamiento fue esencial, siempre a disposición de lo que necesitáramos.

Por último, pero no menos importante, a nuestras familias, que siempre son el pilar y la razón de sacar todo adelante, sin ellos, graduarnos de una de las mejores universidades no estaría dentro de nuestras posibilidades.

En general, nuevamente, muchas gracias a todas las personas, familiares, amigos, profesores, decanos y a todos aquellos que acompañaron el proceso desde el principio y a aquellos que se fueron uniendo al mismo con todas las ganas de vernos crecer.

Glosario

Acrónimos y Abreviaturas

NPL	Natural Processing Language
INVIMA	Instituto Nacional de Vigilancia de Medicamentos y Alimentos
AEMPS	Agencia Española de Medicamentos y Productos Sanitarios
FDA	Food and Drug Administration
SDLC	Software Deveplopment Life Cycle

Resumen

El presente proyecto se centra en el desarrollo de un aplicativo especializado para la extracción automatizada de información de alertas sanitarias, con el objetivo de mejorar la gestión de la información en el ámbito de la salud pública. Se justifica la necesidad de esta herramienta debido a las limitaciones del proceso manual actual, que es propenso a errores y consume mucho tiempo. Para abordar esta problemática, se proponen métodos avanzados de automatización y procesamiento de datos, como el procesamiento del lenguaje natural (NLP), el machine learning y los algoritmos de scraping. Estas tecnologías permitirán no solo extraer datos de manera automática, sino también procesarlos y estructurarlos adecuadamente para su integración en bases de datos.

El proyecto se fundamenta en principios teóricos de NLP, minería de datos y aprendizaje automático, con el objetivo de desarrollar un sistema capaz de interpretar y contextualizar la información de las alertas sanitarias. Esto mejorará la eficiencia y la precisión de la extracción de datos, así como la actualización oportuna de las bases de datos de salud pública. La implementación práctica del aplicativo ofrecerá beneficios significativos, incluyendo una mayor eficiencia operativa, una mayor confiabilidad de la información y una respuesta más rápida a emergencias sanitarias.

El alcance del proyecto incluye el desarrollo de un aplicativo que se enfoque exclusivamente en la extracción de información de alertas sanitarias preexistentes y su integración con las bases de datos existentes. No se abordarán aspectos relacionados con la creación de alertas ni la generación de informes clínicos. El impacto esperado de este aplicativo en la eficiencia operativa de las entidades de salud pública es significativo, ya que facilitará la toma de decisiones informadas y mejorará la capacidad de respuesta ante emergencias sanitarias.

Palabras clave: Alertas sanitarias, automatización, procesamiento del lenguaje natural (NLP), machine learning, gestión de datos, salud pública, eficiencia operativa, extracción de información.

Abstract

The present project focuses on the development of a specialized application for the automated extraction of information from Medical Device Recalls, aiming to enhance information management in the field of public health. The need for this tool is justified by the limitations of the current manual process, which is error-prone and time-consuming. To address this issue, advanced methods of automation and data processing, such as Natural Language Processing (NLP), machine learning, and scraping algorithms, are proposed. These technologies will enable not only the automatic extraction of data but also its processing and proper structuring for integration into databases.

The project is grounded on theoretical principles of NLP, data mining, and machine learning, with the objective of developing a system capable of interpreting and contextualizing Medical Device Recall information. This will improve the efficiency and accuracy of data extraction, as well as the timely updating of public health databases. The practical implementation of the application will offer significant benefits, including greater operational efficiency, increased reliability of information, and faster responses to health emergencies.

The scope of the project includes the development of an application focused exclusively on extracting information from pre-existing Medical Device Recalls and integrating it into existing databases. Aspects related to the creation of recalls or the generation of clinical reports will not be addressed. The expected impact of this application on the operational efficiency of public health entities is significant, as it will facilitate informed decision-making and enhance responsiveness to health emergencies.

Keywords: Medical Device Recalls, automation, Natural Language Processing (NLP), machine learning, data management, public health, operational efficiency, information extraction.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

La gestión eficaz de la información sanitaria es un componente fundamental para salvaguardar la salud pública y respaldar la toma de decisiones informadas. Sin embargo, el proceso actual de recopilación y actualización de información sobre alertas sanitarias suele ser manual, lento y propenso a errores. Esto puede ocasionar retrasos en la respuesta a estas alertas. [1]

Para abordar estas limitaciones, se propone el desarrollo de una herramienta que extraiga de manera automática información de alertas sanitarias. Este proyecto se sustenta en la necesidad de implementar métodos avanzados y robustos de automatización y procesamiento de datos, utilizando tecnologías como el procesamiento del lenguaje natural (NLP), el aprendizaje automático y los algoritmos de scraping. [2]

La pertinencia de este proyecto se deriva de la adopción de enfoques modernos que permitan reducir el tiempo y los recursos humanos necesarios para la gestión de datos sanitarios. Al emplear técnicas de NLP, el sistema puede interpretar y contextualizar la información de las alertas, mejorando significativamente la precisión y confiabilidad de los datos extraídos. [2]

Desde una perspectiva teórica, este proyecto se fundamenta en conceptos clave de la informática y la salud pública, como el procesamiento del lenguaje natural, la minería de datos y el aprendizaje automático. Estas tecnologías permiten a los sistemas entender y procesar el lenguaje humano, identificar patrones en los datos y desarrollar modelos para clasificar y categorizar las alertas sanitarias.[3]

En la práctica, la implementación de este aplicativo ofrece múltiples beneficios, incluida la mejora de la eficiencia operativa de las organizaciones de salud pública, la garantía de la precisión y confiabilidad de la información y la facilitación de la actualización oportuna de las bases de datos de salud pública.[4]

Es decir, desarrollar un aplicativo especializado para la extracción automatizada de información de alertas sanitarias, implementando técnicas avanzadas de procesamiento de lenguaje natural y análisis de datos donde se tendrá un impacto significativo en la mejora de la gestión de la información sanitaria y contribuya a la protección de la salud pública. [5]

Capítulo 2

Planteamiento del Problema

La tecnovigilancia se erige como una herramienta indispensable en el ámbito sanitario colombiano y un pilar fundamental para garantizar la calidad en la atención y la seguridad de los pacientes en los servicios de salud [6]. Este sistema, coordinado por el INVIMA a través del Programa Nacional de Tecnovigilancia, se enfoca en la identificación, evaluación y gestión de eventos adversos asociados con dispositivos médicos, tanto antes como después de su comercialización, que permite prevenir riesgos asociados. [4].

A pesar de contar con un marco normativo sólido, la gestión del programa enfrenta diversos desafíos que obstaculizan un análisis y desarrollo óptimos de los eventos reportados. Entre estos retos se encuentran la subnotificación de eventos adversos por parte del personal sanitario, la falta de recursos y capacitación para una adecuada gestión de la información, y las dificultades para establecer una trazabilidad efectiva de los dispositivos médicos [4].

La tecnovigilancia, piedra angular para la seguridad del paciente en Colombia, enfrenta desafíos que exigen soluciones tecnológicas innovadoras. La alta carga administrativa y el riesgo de errores humanos en la gestión manual de datos son síntomas claros de la necesidad de una herramienta para la automatización de la extracción de información de alertas sanitarias [7].

Los retrasos en la respuesta a incidentes adversos y en la implementación de medidas preventivas, producto de la gestión manual, comprometen directamente la seguridad del paciente y la calidad del servicio sanitario [4]. Inconsistencias en el reporte de eventos, falta de automatización y el desconocimiento del personal sobre las mejores prácticas en tecnovigilancia, son indicadores adicionales que demandan un cambio hacia un sistema más robusto.

Una herramienta para la automatización de la extracción de información de alertas sanitarias permitiría:

- Minimizar errores humanos mediante la automatización de tareas repetitivas y propensas a fallos, como la captura y actualización de datos [6].
- Agilizar la respuesta a incidentes adversos y la implementación de medidas preventivas, gracias a un acceso oportuno y confiable a la información [4].
- Fortalecer la calidad y confiabilidad de los datos al garantizar consistencia en su reporte y gestión [4].

- Promover el conocimiento y la capacitación del personal de salud en tecnovigilancia, mediante herramientas intuitivas y accesibles [4].
- Reducir la carga administrativa y liberar tiempo valioso para que el personal sanitario se concentre en la atención al paciente [4].

La implementación de una herramienta de estas características no solo optimizaría la gestión del programa de tecnovigilancia, sino que también contribuiría a mejorar la seguridad del paciente y la calidad de la atención sanitaria en Colombia [4].

La tecnovigilancia en Colombia se enfrenta a una serie de desafíos que obstaculizan su eficacia [4]. La dependencia de procesos manuales, la falta de capacitación del personal y el cumplimiento normativo inadecuado son algunos de los principales problemas. La complejidad inherente a la tecnovigilancia, con su gran volumen de datos, informes detallados y necesidad de actualizaciones rápidas, agrava aún más la situación.

Para superar estos retos, se requiere un enfoque integral que combine la implementación de herramientas tecnológicas de automatización, el fortalecimiento de la capacitación del personal y la mejora en el cumplimiento de las normativas [8]. Solo así se podrá construir un sistema de tecnovigilancia verdaderamente efectivo en Colombia, capaz de garantizar la seguridad del paciente y la calidad de la atención sanitaria.

Se propone una herramienta innovadora para automatizar la extracción de información de alertas sanitarias, con el objetivo de optimizar la actualización de las bases de datos de manera rápida, precisa y confiable. Esta herramienta busca solucionar los problemas de ineficiencia, errores y retrasos en la toma de decisiones asociados a la gestión manual de datos en la tecnovigilancia de Colombia.

La implementación de este sistema automatizado permitirá reducir la carga administrativa, minimizar errores humanos, agilizar la respuesta ante incidentes adversos, fortalecer la calidad y confiabilidad de los datos, y promover una mejor adherencia a las normativas de tecnovigilancia. En definitiva, esta se perfila como una herramienta fundamental para optimizar la tecnovigilancia en Colombia, contribuyendo así a la seguridad del paciente, la calidad de la atención médica y el cumplimiento de las normativas vigentes.

Capítulo 3

Justificación

La gestión de la información sanitaria es crucial para proteger la salud pública y garantizar la toma de decisiones oportunas. Sin embargo, el proceso actual de recopilación y actualización de información sobre alertas sanitarias es a menudo manual, lento y propenso a errores. Esto puede retrasar la respuesta a brotes de enfermedades y otras amenazas a la salud pública [9].

El desarrollo de una herramienta que permita la extracción de alertas sanitarias se sustenta en la necesidad de implementar métodos avanzados y robustos de automatización y procesamiento de datos. Utilizando tecnologías como el procesamiento del lenguaje natural (NLP), el machine learning y los algoritmos de scraping, es posible construir un sistema que no solo extraiga datos de manera automática, sino que también los procese y estructure adecuadamente para su integración en bases de datos [9].

La pertinencia metodológica radica en la adopción de enfoques modernos que permitan reducir el tiempo y los recursos humanos necesarios para la gestión de datos sanitarios. Al emplear técnicas de NLP, el sistema puede interpretar y contextualizar la información de las alertas, que mejora significativamente la precisión y confiabilidad de los datos extraídos [10].

El desarrollo de una herramienta que automatice la extracción de información de alertas sanitarias es altamente pertinente por las siguientes razones:

- **Aumenta la Confiabilidad de la Información**: La automatización puede ayudar a minimizar los errores humanos y garantizar la precisión de la información de las alertas sanitarias. Esto es esencial para tomar decisiones informadas y proteger la salud pública [11].

- **Facilita la Actualización Oportuna de la Información**: La automatización permite una actualización más rápida de las bases de datos de salud pública, que garantiza que los profesionales de la salud tengan acceso a la información más reciente sobre las amenazas a la salud [6].

- **Disminuye el tiempo dedicado a esta tarea** : La automatización del proceso de extracción de datos puede reducir significativamente el tiempo y el esfuerzo necesarios para actualizar las bases de datos de salud pública. Esto libera a los profesionales de la salud para que se concentren en tareas más críticas [11].

Teóricamente, este proyecto se fundamenta en diversos conceptos clave de la informática y la salud pública. La integración de tecnologías de NLP y machine learning se basa en teorías de inteligencia artificial que permiten a los sistemas entender y procesar el lenguaje humano. Además, el uso de técnicas de scraping de datos se sustenta en principios de ingeniería de Software que facilitan la extracción automatizada de información de fuentes web no estructuradas [12].

Teniendo en cuenta lo anterior, para el desarrollo de la herramienta es importante comprender que el mismo se basa en los siguientes principios teóricos:

- **Procesamiento del Lenguaje Natural (NLP)**: El NLP se utilizará para extraer automáticamente información relevante de textos no estructurados, como informes de noticias, sitios web gubernamentales y publicaciones científicas [10].
- **Minería de Datos**: La minería de datos se utilizará para identificar patrones y tendencias en los datos de las alertas sanitarias, que puede ayudar a mejorar la vigilancia de la salud pública [13].
- **Aprendizaje Automático**: El aprendizaje automático se utilizará para desarrollar modelos que puedan clasificar y categorizar automáticamente las alertas sanitarias, que facilita su gestión y análisis [13].

Por lo anterior, se puede comprender que la pertinencia teórica del proyecto es evidente, ya que aborda desafíos contemporáneos en la intersección de la tecnología y la salud pública. Para la cantidad de alertas sanitarias que se manejan a diario a nivel global, es crucial desarrollar sistemas capaces de manejar grandes volúmenes de datos de manera rápida y precisa. Este enfoque teórico no solo valida la necesidad del proyecto, sino que también proporciona un marco sólido para su implementación [13].

Desde una perspectiva práctica, la implementación de esta herramienta ofrece múltiples beneficios y tiene un impacto significativo en la gestión de información sanitaria. Actualmente, la actualización manual de bases de datos es un proceso tedioso, propenso a errores y consume mucho tiempo. Un sistema automatizado transformaría este proceso, permitiendo actualizaciones rápidas, precisas y confiables [11].

La necesidad del proyecto se manifiesta en su capacidad para mejorar la operatividad de las organizaciones de salud pública, reduciendo el riesgo de errores humanos y mejorando la capacidad de respuesta ante emergencias sanitarias. Además, al garantizar que las bases de datos estén actualizadas en tiempo real, se facilita la toma de decisiones informadas y oportunas, que es crucial para la protección de la salud pública [9].

La herramienta propuesta será un desarrollo innovador para entidades de salud pública, hospitales, laboratorios y cualquier organización que dependa de información actualizada sobre alertas sanitarias. Su utilidad radica en la capacidad de proporcionar datos precisos y actualizados de

manera continua, que a su vez facilita la planificación, prevención y respuesta ante posibles crisis sanitarias [6].

El impacto del proyecto se puede observar en varios niveles:

- **Precisión y Confiabilidad**: Al minimizar el riesgo de errores humanos, la herramienta garantiza que la información en las bases de datos sea más precisa y confiable, que es esencial para la toma de decisiones informadas [9].
- **Respuesta Rápida**: Con datos actualizados en tiempo real, las entidades de salud pública pueden reaccionar más rápidamente ante emergencias sanitarias, mejorando la capacidad de respuesta y mitigando el impacto de posibles brotes o crisis [9].
- **Capacidad Operativa**: La automatización reduce significativamente el tiempo y esfuerzo necesario para la gestión de datos sanitarios, permitiendo que los profesionales de la salud se concentren en actividades más estratégicas y críticas [11].
- **Innovación en Salud Pública**: La implementación de tecnologías avanzadas como NLP y machine learning en la gestión de información sanitaria representa un avance significativo en la modernización de los sistemas de salud pública [6].

Por lo tanto, este puede ser el comienzo del desarrollo de una herramienta para la automatización de la extracción de información de alertas sanitarias es un proyecto altamente pertinente, viable y de gran impacto. El proyecto tiene el potencial de mejorar significativamente la vigilancia de la salud pública, la eficiencia del sistema sanitario y la toma de decisiones.

Además, es importante destacar que el desarrollo de este herramienta debe realizarse en colaboración con expertos en salud pública, informática y procesamiento del lenguaje natural. Además, es necesario establecer mecanismos para garantizar la calidad y confiabilidad de los datos extraídos. Este enfoque colaborativo y riguroso asegurará que el sistema desarrollado sea robusto, preciso y capaz de cumplir con las exigencias del entorno de la salud pública moderna.

Capítulo 4

Objetivos

4.1. Objetivo General

Desarrollar una herramienta que permita la extracción, integración y actualización automática de información de alertas sanitarias emitidas por el Instituto Nacional de Vigilancia de Medicamentos y Alimentos (INVIMA), para facilitar la actualización de las bases de datos utilizadas dentro de las instituciones prestadoras de servicio de salud.

4.2. Objetivos Específicos

- Identificar las necesidades técnicas específicas que tengan los profesionales de la salud, líderes de tecnovigilancia y relacionados, para garantizar que la herramienta cumpla con los requerimientos y facilite la toma de decisiones en el ámbito sanitario.
- Elaborar un algoritmo que permita la automatización completa del proceso de extracción, integración y actualización de la información de alertas sanitarias.
- Evaluar el funcionamiento de la herramienta comparando los resultados obtenidos con los métodos tradicionales de extracción de información manual.

Capítulo 5

Marco de Referencia

5.1. Áreas Temáticas

- Desarrollo de una herrramienta
- Alertas sanitarias
- Seguridad al paciente
- Procesamiento de Lenguaje Natural (NLP)
- Web Scraping
- Machine Learning
- Validación de Datos
- Bases de Datos

5.2. Marco Teórico

5.2.1. Alertas Sanitarias:

Las alertas sanitarias son comunicaciones emitidas por autoridades de salud pública, como el INVIMA a nivel nacional, el AEMPS o el FDA a nivel internacional, para notificar sobre amenazas emergentes a la salud. Estas alertas pueden provenir de organismos internacionales, nacionales o locales y son fundamentales para la gestión de emergencias sanitarias [1].

5.2.2. Instituto Nacional de Vigilancia de Medicamentos y Alimentos (INVIMA):

Instituto técnico, científico del orden nacional, adscrito al Ministerio de Salud y Protección Social, encargado de ejecutar las políticas formuladas por el Gobierno en materia de inspección, vigilancia y control sanitario, basado en la gestión del riesgo de los productos de su competencia, para proteger y promover la salud pública a través de la articulación sectorial e intersectorial y contribuir a la mejora continua del estatus sanitario [14].

5.2.3. Decreto 4725 de 2005:

Regular el sistema de registros sanitarios, los permisos de comercialización y la vigilancia sanitaria relacionados con la producción, procesamiento, envasado, empaque, almacenamiento, venta, uso, importación, exportación, comercialización y mantenimiento de dispositivos médicos para uso humano. Estas regulaciones serán obligatorias para todas las personas físicas o jurídicas que realicen estas actividades en el país [15]. Adicionalmente, este define conceptos importantes como:

- **Tecnovigilancia:** Conjunto de actividades que tienen por objeto la identificación y la cualificación de efectos adversos serios e indeseados producidos por los dispositivos médicos, así como la identificación de los factores de riesgo asociados a estos efectos o características, se rige por el régimen de registros sanitarios, permiso de comercialización y vigilancia sanitaria de los dispositivos médicos para uso humano. Este proceso incluye la notificación, registro y evaluación sistemática de los efectos adversos de los dispositivos médicos con el fin de determinar la frecuencia, gravedad e incidencia de los mismos para prevenir su aparición [15].
- **Dispositivo Médico Activo:** Cualquier dispositivo médico que funcione mediante una fuente de energía eléctrica o cualquier fuente de energía diferente a la generada directamente por el cuerpo humano o la gravedad, y que opera mediante la conversión de esa energía. No se consideran dispositivos médicos activos aquellos productos sanitarios diseñados para transmitir, sin modificar significativamente, energía, sustancias u otros elementos desde un dispositivo médico activo al paciente [15].
- **Equipo Biomédico:** Un dispositivo médico funcional que integra sistemas y subsistemas eléctricos, electrónicos y/o hidráulicos, incluidos los programas informáticos necesarios para su correcto funcionamiento, destinado por el fabricante a ser utilizado en seres humanos con fines de prevención, diagnóstico, tratamiento o rehabilitación. No se consideran equipos biomédicos los dispositivos médicos implantados en el cuerpo humano o aquellos destinados para un solo uso [15].

5.2.4. Seguridad del paciente:

La seguridad del paciente se refiere a la reducción de riesgos y daños asociados con la atención sanitaria. Un sistema de gestión de alertas sanitarias contribuye a la seguridad del paciente al garantizar que los profesionales de la salud tengan acceso a información actualizada y precisa [16].

- **Elementos Clave para la Seguridad del Paciente:**
  - **Identificación de Riesgos:** Detectar y evaluar los riesgos emergentes para la salud [16].
  - **Comunicación Efectiva:** Asegurar que las alertas sanitarias se comuniquen de manera clara y oportuna a los profesionales de la salud y al público [16].
  - **Implementación de Medidas Preventivas:** Basadas en las alertas, implementar medidas para prevenir la propagación de enfermedades y otras amenazas [16].

5.2.5. Desarrollo de una herramienta

El desarrollo de una herramienta es el proceso de diseñar, construir, implementar y mantener aplicaciones informáticas. Este proyecto requiere un enfoque meticuloso para asegurar que el software cumpla con los requisitos específicos de la gestión de alertas sanitarias [17].

1. **Ciclo de Vida del Desarrollo de Software (SDLC):**

El SDLC guía el desarrollo del software a través de varias etapas:

- **Análisis de Requisitos**: Identificar las necesidades del usuario final y especificar las funcionalidades del software. Para este proyecto, esto incluye comprender cómo se generan y distribuyen las alertas sanitarias y cómo se almacenan actualmente en las bases de datos [17].
- **Diseño**: Crear la arquitectura del software, incluyendo el diseño de la interfaz de usuario y la estructura de datos. Este diseño debe soportar la integración de tecnologías como NLP, web scraping y machine learning [17].
- **Implementación**: Programar el software utilizando lenguajes de programación adecuados [17].
- **Pruebas**: Verificar que el software funcione correctamente, identificando y corrigiendo errores. Se deben realizar pruebas unitarias, de integración y de sistema [17].
- **Despliegue y Mantenimiento**: Implementar el software en el entorno de producción y realizar actualizaciones y mejoras continuas basadas en el feedback de los usuarios [17].

5.2.6. Procesamiento natural del lenguaje (NLP):

Área de conocimiento de la Inteligencia Artificial que investiga cómo se comunican las máquinas con los seres humanos. Con el objetivo de desarrollar mecanismos que potencien la comunicación entre las personas y las máquinas[2]

5.2.7. Bases de datos:

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático [18].

5.3. Trabajos Relacionados

- **Investigación de materiales basada en datos habilitada por el procesamiento del lenguaje natural y la extracción de información:** Con el auge de la ciencia de datos y el aprendizaje automático, obtener datos en la ciencia de materiales se ha vuelto crucial. Estos datos son heterogéneos y abarcan muchas órdenes de magnitud, manifestándose como texto numérico o información basada en imágenes que requiere interpretación cuantitativa. La capacidad de consumir y codificar automáticamente la literatura científica mediante técnicas de procesamiento del lenguaje natural tiene un gran potencial para generar los conjuntos de datos necesarios. Esta revisión se centra en los avances en procesamiento del lenguaje natural y minería de textos en la literatura de ciencia de materiales, destacando oportunidades para extraer información de figuras y tablas en artículos. Sin embargo, la implementación de este tipo de proyecto enfrenta limitaciones técnicas y tecnológicas específicas, como la calidad y estandarización de los datos, las limitaciones computacionales, la complejidad del procesamiento del lenguaje natural (PLN), la seguridad y privacidad de los datos, y la necesidad de actualización y mantenimiento continuos. La variabilidad y falta de estandarización de los datos pueden llevar a inconsistencias y resultados no confiables, mientras que las limitaciones en recursos computacionales pueden ralentizar el progreso del proyecto. Además, las técnicas de PLN pueden enfrentar dificultades al interpretar textos científicos complejos, y el manejo de grandes cantidades de datos plantea preocupaciones sobre seguridad y privacidad. Finalmente, la tecnología en este campo está en constante evolución, que requiere un esfuerzo continuo para mantenerse actualizado. Para mitigar estos impactos, se pueden implementar procesos rigurosos de limpieza y preprocesamiento de datos, utilizar infraestructuras de computación en la nube, desarrollar modelos de PLN específicos para la ciencia de materiales, implementar medidas robustas de ciberseguridad, y establecer un plan de actualización y capacitación continua para el equipo de trabajo. Al abordar estas limitaciones de manera proactiva, se puede maximizar el potencial del procesamiento del lenguaje natural y la extracción de información en la investigación de materiales, facilitando la generación de datos precisos y útiles para avanzar en el campo de la ciencia de materiales [19].

- **Automatización de la extracción de datos en revisiones sistemáticas: una revisión sistemática:** La automatización del proceso de extracción de datos en las revisiones sistemáticas podría reducir significativamente el tiempo necesario para completarlas. Sin embargo, el estado de la ciencia en cuanto a la extracción automática de elementos de datos de textos completos no ha sido bien descrito. Tras revisar informes en PubMed, IEEEXplore y ACM Digital Library, encontramos que, de los más de 52 elementos de datos utilizados en revisiones sistemáticas, el 48 % fueron objeto de intentos de extracción automática y el 27 % fueron completamente extraídos, con un máximo de 7 elementos extraídos automáticamente por estudio, obteniendo la mayoría puntuaciones F superiores al 70 %. No se ha encontrado un marco unificado de extracción adaptado al proceso de revisión sistemática, y las técnicas de procesamiento del lenguaje natural en biomedicina no se han utilizado plenamente. Las limitaciones incluyen la calidad y estandarización de los datos, las limitaciones computacionales,

la complejidad del procesamiento del lenguaje natural, la seguridad y privacidad de los datos, y la necesidad de actualización continua. Para mitigar estos impactos, es esencial implementar procesos rigurosos de limpieza de datos, utilizar computación en la nube, desarrollar modelos de PLN específicos, proteger la privacidad con medidas de ciberseguridad y mantener una formación continua del equipo en los últimos avances tecnológicos [5].

- **Procesamiento del lenguaje natural y extracción de información: análisis cualitativo de artículos de noticias financieras:** En la actualidad, los datos financieros cuantitativos se analizan mayoritariamente mediante programas automáticos que emplean técnicas tradicionales o de inteligencia artificial, pero los datos cualitativos aún no se procesan con eficacia, generando una sobrecarga de información para los operadores financieros. El sistema de extracción de información financiera desarrollado en la Universidad de Durham puede identificar tipos específicos de información en artículos fuente, generando plantillas relevantes y reduciendo la sobrecarga de datos cualitativos. No obstante, la implementación enfrenta limitaciones técnicas y tecnológicas como la calidad y estandarización de los datos, las limitaciones computacionales, la complejidad del procesamiento del lenguaje natural (PLN), la seguridad y privacidad de los datos, y la necesidad de actualización continua. Para mitigar estos desafíos, se deben implementar procesos rigurosos de limpieza y preprocesamiento de datos, utilizar computación en la nube, desarrollar modelos de PLN específicos para el dominio financiero, emplear medidas robustas de ciberseguridad, y mantener un plan de actualización y capacitación continua para el equipo. Abordar estas limitaciones permitirá maximizar el potencial del PLN y la extracción de información en el manejo de datos financieros cualitativos, mejorando la toma de decisiones en el sector financiero [20].

- **Extracción de datos clínicos y normalización de registros médicos electrónicos cirílicos mediante procesamiento de lenguaje natural de aprendizaje profundo:** Una gran parte de los datos médicos está no estructurada, que dificulta el avance de la investigación clínica y la mejora del cuidado del paciente, especialmente en idiomas con alfabetos no latinos como el cirílico. Desarrollamos algoritmos de procesamiento del lenguaje natural basados en aprendizaje profundo para extraer automáticamente el estado de biomarcadores de pacientes con cáncer de mama en Bulgaria, logrando puntajes F1 de 0.90 o superiores. Sin embargo, la implementación enfrenta limitaciones técnicas y tecnológicas, como la calidad y estandarización de los datos, las limitaciones computacionales, la complejidad del procesamiento del lenguaje natural, y la seguridad y privacidad de los datos. Para mitigar estos desafíos, se deben implementar procesos rigurosos de limpieza de datos, utilizar computación en la nube, desarrollar modelos específicos de PLN, emplear medidas robustas de ciberseguridad, y mantener un plan de actualización y capacitación continua del equipo. Abordar estas limitaciones permitirá maximizar el potencial del PLN en el manejo de datos médicos no estructurados, mejorando la investigación clínica y el cuidado del paciente [3].

- **Procesamiento del lenguaje natural: algoritmos y herramientas para extraer información computable de HCE y de la literatura biomédica:** Este número especial

de JAMIA destaca la importancia del procesamiento biomédico del lenguaje natural (PNL) en la interpretación de registros médicos electrónicos y la literatura biomédica. Aunque los algoritmos de PNL avanzados y los corpus abiertos mejoran la normalización de enfermedades y la extracción de información, la implementación enfrenta limitaciones técnicas y tecnológicas. La calidad y estandarización de los datos médicos, las limitaciones computacionales, la complejidad del procesamiento del lenguaje natural, y la seguridad y privacidad de los datos son desafíos significativos. Para mitigarlos, es esencial implementar procesos rigurosos de limpieza y normalización de datos, utilizar computación en la nube, desarrollar modelos de PLN específicos, emplear medidas robustas de ciberseguridad y mantener un plan de actualización y capacitación continua. Abordar estas limitaciones permitirá maximizar el potencial del PNL en la interpretación de datos biomédicos, mejorando la investigación clínica y el cuidado del paciente [21].

Capítulo 6

Materiales y Métodos

El presente proyecto tiene como objetivo principal el desarrollo de un software para la extracción de información específica desde documentos en formato PDF expuestos en la página web del INVIMA (https://app.invima.gov.co/alertas/dispositivos-medicos-invima). Se realizaron diferentes etapas que llevaron a completar satisfactoriamente el objetivo propuesto: primero, obtención de URLs, un módulo encargado de realizar scraping en el sitio del INVIMA para identificar enlaces de archivos PDF relevantes; segundo, descarga de archivos PDF; tercero, extracción de metadatos, funciones que extraen información clave como fechas, nombres de dispositivos, registros sanitarios y descripciones desde los PDF; cuarto, machine learning, un módulo que utiliza un modelo entrenado para clasificar dispositivos médicos a partir del texto extraído; y, quinto, actualización de la base de datos.

El diseño de este código se realizó de forma modular, es decir, se definieron funciones a lo largo de este, las cuales cada una tienen un trabajo diferente. Al abordar de esta manera la problemática, realizar pruebas, correcciones y mejoras a partes específicas del código es más sencillo, además, esto permite una implementación más completa y sencilla que es la solución para cada uno de los objetos extraídos de los documentos.

El código fue diseñado, desarrollado e implementado en el lenguaje de programación Python, debido a su amplia disponibilidad de bibliotecas y a la facilidad que ofrece para la escritura de scripts. Además, se utilizó un entorno local de Visual Studio Code (VS Code) para redactar y ejecutar el código. Las bibliotecas utilizadas fueron:

- **Os:** Para la manipulación del sistema de archivos. [22]
- **re:** Para expresiones regulares y procesamiento de texto.[23]
- **Fitz (PyMuPDF):** Es el nombre del módulo principal de la biblioteca PyMuPDF, una herramienta de Python que permite leer, extraer, modificar y analizar archivos PDF de forma eficiente. Aunque el paquete se instala como PyMuPDF, el módulo se importa como fitz.[24]
- **Requests:** Para realizar solicitudes HTTP y descargar archivos.[25]
- **BeautifulSoup (de bs4):** Para la extracción de datos HTML desde páginas web.[26]
- **Pandas:** Para la manipulación y análisis de datos. [27]

- **Joblib:** Para cargar modelos previamente entrenados. [28]
- **PyPDF2 y PDFplumber:** Para tareas avanzadas de extracción de contenido de PDFs.[29][30]
- **Openpyxl:** Para la manipulación de archivos Excel.[31]
- **Tkinter:** Para crear una interfaz gráfica de usuario.[32]
- **Sklearn:** Para dividir datos en entrenamiento/prueba, transformar texto en vectores (TF-IDF), entrenar un modelo de clasificación (RandomForestClassifier) y evaluar su rendimiento.[33]
- **Seaborn:** Hace gráficos estadísticos que proporciona una interfaz de alto nivel a matplotlib y adicionalmente permite integrarse con las estructuras de datos de pandas. [34]

6.1. Actividades Realizadas

OE1:

Con el fin de identificar las necesidades técnicas específicas de los profesionales de la salud, líderes de tecnovigilancia y demás actores relacionados, y así garantizar que la herramienta cumpla con los requerimientos y facilite la toma de decisiones en el ámbito sanitario, se llevó a cabo una indagación con la directora del proyecto, la Ing. Natalia Pedreros. Esta se enfocó en comprender las necesidades puntuales del profesional encargado de actualizar manualmente la base de datos de alertas sanitarias e informes de seguridad.

Para esto, se llevó a cabo una entrevista, donde se hicieron diversas preguntas como:

1. ¿Se desea automatizar el proceso completamente o hay algún dato de los archivos que requieran ingresar manualmente?
2. ¿Cómo se almacenarán y organizarán los datos extraídos?
3. ¿La herramienta va a ser usada desde el código o mejor crear una interfaz?
4. ¿Cúal es la información, proveniente de los archivos PDF, que es necesaria que se registre en la base de datos?

Una vez adquiridas las necesidades que debe cubrir la herramienta, se construyó un diagrama de flujo que guió la arquitectura de la herramienta, que se presenta en la figura 6.1. Al tener un codigo con estructura modular, se construyó paso a paso hasta cumplir con todos los requisitos presentados.

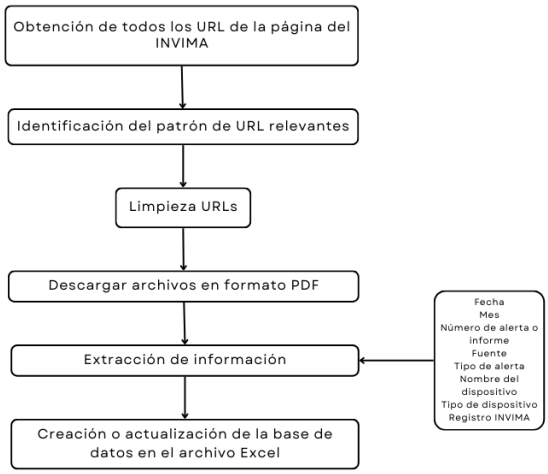


Figura 6.1: Diagrama de construcción del sistema

OE2:

Para el cumplimiento de la elaboración de un algoritmo que permita la automatización completa del proceso de extracción, integración y actualización de la información de alertas sanitarias, primero, se definieron la URL base, headers, rutas de almacenamiento local para los archivos y el archivo excel que será utilizado, como se puede observar en la figura 6.2.

```
base_url = "https://app.invima.gov.co/alertas/dispositivos-medicos-invima"
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'}
PDF_FOLDER = "PDFs_Invima"
EXCEL_FILE = "alertas_invima.xlsx"
```

Figura 6.2: Configuración base

Posteriormente, se creó una función que navega por el sitio web del INVIMA y las múltiples paginas del sitio buscando enlaces que apunten a documentos PDF, para esto, se inicializa un conjunto para almacenar las URLs sin duplicados, se recorren las páginas y se utiliza la función ”requests.get()”obteniendo los enlaces existentes en la pagina web del INVIMA que, posteriormente, utilizando un reconocimiento de patrones identificados en los URL de los archivos pdf de las alertas sanitarias e informes de seguridad, que se muestran en la figura 6.3, se evaluan, dejando así solo los archivos PDF de interés.

```
URL_PATTERNS = [
    re.compile(r"https://app\.invima\.gov\.co/alertas/ckfinder/userfiles/files/ALERTAS%20SANITARIAS/.+\.pdf"),
    re.compile(r"https://app\.invima\.gov\.co/alertas/ckfinder/userfiles/files/INFORMES%20DE%20SEGURIDAD/.+\.pdf")
]
```

Figura 6.3: Patrones URL

Esto da paso a la función de descarga de los archivos PDF, para esto, se hace una limpieza de los URLs obtenidos anteriormente, utilizando la función mostrada en la figura 6.4, los cuales tenían la forma: `https://app.invima.gov.co/alertas/ckfinder/userfiles/files/INFORMES%20DE%20SEGURIDAD/Dispositivos/2025/ENERO/Informe%20de%20seguridad%20No_%20%23007-2025%20.pdf`, dejándolos de la forma: `PDFs_Invima/Informe%20de%20seguridad%20No_%20%23010-2025%20.pdf`, esto ayuda a que, al enviar la solicitud de descarga del archivo, mediante la biblioteca requests, se evite que se produzca un error 500, el cual se presenta cuando se realiza una solicitud invalida a una página web y, además, que se sobrecargue la memoria y el servidor, permitiendo así que se descarguen los archivos PDF en la carpeta destino, la cual fue especificada desde un inicio, y se genere una lista con las rutas locales de los archivos PDF descargados.

```
def limpiar_url(url):
    """Limpia y codifica la URL."""
    url = unquote(url).strip()
    return quote(url, safe=":/")
```

Figura 6.4: Limpieza de URL

Consecuentemente, se definieron cuales eran los objetos de extracción, es decir, la información contenida en los documentos que era requerida en la base de datos, estos son: la fecha del documento, el número de alerta, fuente de la alerta, el tipo de alerta, el nombre del dispositivo, el tipo de dispositivo, el registro INVIMA y la descripción del caso.

Para empezar, con el fin de lograr identificar el tipo de dispositivo que se tenia en cada documento procesado, se construyeron, se entrenaron y se probaron 5 modelos predictivos basados en machine learning: Random Forest Classifier, el cual combina varios árboles de decisión que permite diferentes entradas que llevan a cabo la evaluación de las diferentes posibilidades de clasificación y pertenece al modelo de aprendizaje supervisado; XGboost, un algoritmo de aprendizaje automático basado en árboles de decisión que, mediante el ensamblado secuencial y regularizado de múltiples árboles, optimiza la precisión y el rendimiento en tareas de clasificación; KNN, clasifica una muestra desconocida asignándole la clase más común entre sus k vecinos más cercanos, calculados mediante una métrica de distancia en el espacio de características; SVM, clasificador supervisado que busca el hiperplano que maximiza el margen entre clases, utilizando funciones núcleo para manejar separaciones no lineales cuando es necesario; y MLP; red neuronal feedforward compuesta por una o

más capas ocultas que aprende funciones no lineales mediante el ajuste de pesos con el algoritmo de retropropagación.

Para el entrenamiento y validación de los modelos mencionados, se utilizó una base de datos antigua, proporcionada por la Directora del proyecto, la ingeniera Natalia Pedreros, con esta base de datos, se identificaron los datos importantes como el nombre y el tipo de dispositivo, que se encuentran en la base de datos con la estructura mostrada en la figura 6.5.

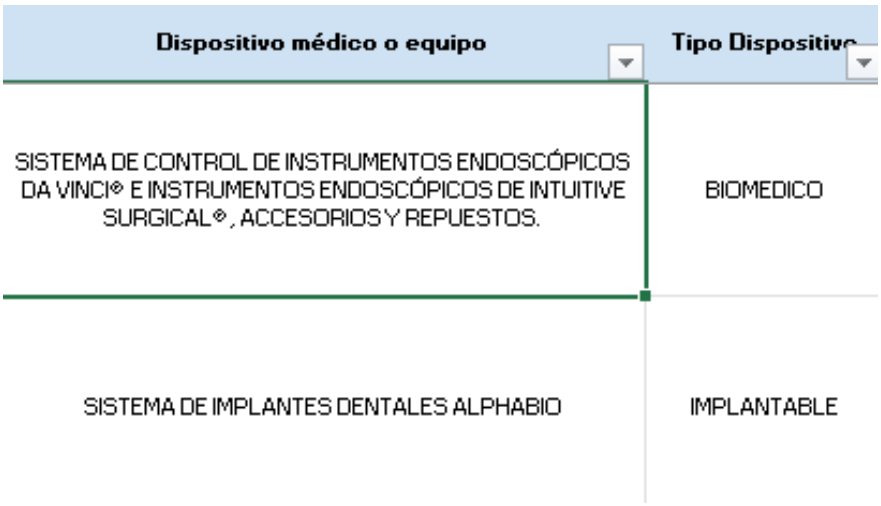


Figura 6.5: Estructura base de datos antigua

Se realizaron ajustes de hiperparametros mediante gridsearch, para Random Forest se ajustaron los parámetros n_estimators (número de árboles), max_depth (profundidad máxima) y min_samples_split (mínimo de muestras para dividir un nodo), buscando un equilibrio entre precisión y sobreajuste. En XGBoost se evaluaron n_estimators y max_depth, controlando la complejidad de los árboles en un enfoque secuencial de mejora. Para KNN se probaron diferentes valores de n_neighbors y métodos de ponderación (uniform y distance) para optimizar la clasificación basada en proximidad. En el modelo SVM se exploraron los hiperparámetros C (regularización) y kernel (tipo de núcleo: lineal o radial) para ajustar la capacidad del modelo de encontrar márgenes óptimos. Finalmente, en el MLP se variaron las arquitecturas de capas ocultas (hidden_layer_sizes) y las funciones de activación (relu, tanh), buscando configuraciones que permitieran una representación no lineal eficiente sin sobreentrenamiento.

Además, se realizó la limpieza del texto, esta se lleva a cabo eliminando caracteres especiales, dígitos y palabras repetitivas o innecesarias que no contribuyen a la identificación del tipo de dispositivo utilizando la libreria nltk (“Stopwords”), también, se realizó la partición de la información para entrenamiento (65 %) y para validación (35 %), y, por último, se exportaron 2 archivos: modelo.pkl, que contiene el clasificador final entrenado para la identificación del tipo de dispositivo,

incluyendo todos los parámetros aprendidos. Lo que permite reutilizar el modelo sin tener que reentrenarlo; Y vectorizador.pkl, que es una herramienta fundamental en el procesamiento de lenguaje natural, ya que convierte texto libre en vectores numéricos comprensibles para los modelos de machine learning. En este caso, se utilizó "TfidfVectorizer", configurado para trabajar con unigramas y bigramas, limitar las características más frecuentes y eliminar aquellas poco representativas. Su función es garantizar que cualquier nuevo texto sea transformado en un vector numérico estandarizado, conservando su significado semántico, y alimentar así los modelos predictivos para que pudieran aprender patrones y clasificar correctamente los dispositivos médicos..

Al comprobar el funcionamiento y la precisión del modelo predictivo y de la extracción, limpieza y descarga de los archivos en formato PDF, se procedió al desarrollo del código perteneciente a la extracción de la información necesitada.

Para la construcción de la herramienta de extracción se utilizó la biblioteca fitz, la cual permitió acceder a los documentos y recorrer sus páginas de manera secuencial mediante la estructura ''with fitz.open(filepath) as pdf", asegurando una gestión eficiente de recursos y el cierre automático de los archivos. Esta herramienta facilitó la apertura de los archivos y la lectura de sus páginas secuencialmente mediante el ciclo ''for page in pdf", donde se aplicó el método get_text("text") para recuperar el texto visible en cada página en el formato existente en cada documento y convertirlo en un texto plano para facilitar la lectura y extracción de la información, esto se muestra en la figura 6.6. Además, la capacidad para conservar la organización lógica del texto lo convirtió en una herramienta robusta frente a documentos con estructuras variadas, permitiendo una integración confiable con expresiones regulares para extraer la información relevante de forma automatizada.

```
try:
    with fitz.open(filepath) as pdf:
        for page in pdf:
            text = page.get_text()
```

Figura 6.6: Uso de Fitz

Teniendo en cuenta lo anterior, se construyó el código paso a paso de la siguiente manera:

1. Se diseñó una función, que se puede observar en la figura 6.7, la cual se encargó de la extracción de la fecha del documento. Para esto se emplea una expresión regular para buscar coincidencias con el patrón existente en los documentos (9 enero 2025), si se encuentra, se realiza la transformación a un formato numérico DD/MM/AAAA y la devuelve.

```python
def extract_complete_date(filepath):
    """Extrae la fecha completa del contenido de un PDF."""
    try:
        with fitz.open(filepath) as pdf:
            for page in pdf:
                text = page.get_text()
                # Busca formatos comunes de fecha
                match = re.search(
                    r'\b(\d{1,2})\s(de\s)?(enero|febrero|marzo|abril|mayo|junio|julio|agosto|septiembre|octubre|noviembre|diciembre)\s(\d{4})\b',
                    text, re.IGNORECASE
                )
                if match:
                    day, _, month, year = match.groups()
                    month_map = {
                        "enero": "01", "febrero": "02", "marzo": "03", "abril": "04", "mayo": "05", "junio": "06",
                        "julio": "07", "agosto": "08", "septiembre": "09", "octubre": "10", "noviembre": "11", "diciembre": "12"
                    }
                    formatted_date = f"{int(day):02d}/{month_map[month.lower()]}/{year}"
                    print(f"Fecha completa encontrada: {formatted_date}")
                    return formatted_date
    except Exception as e:
        print(f"Error al procesar el PDF {filepath}: {e}")
    return "Fecha no encontrada"
```

Figura 6.7: Extracción fecha documento

2. Se realizó una función que permite identificar el tipo de alerta reportada, determinando si se trata de una alerta sanitaria o un informe de seguridad. Se reconoció el patrón con expresiones como “Alerta No. 006-2025” e “Informe de Seguridad No. 002-2025” y se extrajo tanto el tipo de alerta como el número de identificación único de cada una. Esto se puede observar en la figura 6.8.

```python
def extract_alert_info(filepath):
    """Extrae el tipo (Alerta o Informe) y el número asociado del contenido del PDF."""
    try:
        with fitz.open(filepath) as pdf:
            for page in pdf:
                text = page.get_text()
                # Busca los patrones "Alerta No. XXX-XXXX" o "Informe de Seguridad No. XXX-XXXX"
                match = re.search(r'(Alerta|Informe de Seguridad) No\.?\s*(\d{3}-\d{4})', text, re.IGNORECASE)
                if match:
                    alert_type = match.group(1)  # "Alerta" o "Informe de Seguridad"
                    alert_number = match.group(2)  # El número "XXX-XXXX"
                    print(f"Encontrado: {alert_type} - {alert_number}")
                    return alert_type, alert_number
    except Exception as e:
        print(f"Error al procesar el PDF {filepath}: {e}")
    return "Desconocido", "No encontrado"
```

Figura 6.8: Extracción tipo y número de alerta

3. Se construyó una función para sacar el mes del documento, como se puede observar en la figura 6.9, para esto, se tomó la fecha previamente obtenida en formato numérico DD/MM/AAAA, se tomó al “/” como un separador de información y se extrajo solamente la información del

mes (MM), para, posteriormente, transformar el número obtenido en el equivalente en letras, por ejemplo: 03 pasaría a ser marzo.

```python
def extract_month_from_date(date_text):
    """Extrae el mes de una fecha en formato día/mes/año."""
    if date_text == "Fecha no encontrada":
        return "Desconocido"
    try:
        _, month, _ = date_text.split("/")
        month_map = {
            "01": "Enero", "02": "Febrero", "03": "Marzo", "04": "Abril", "05": "Mayo", "06": "Junio",
            "07": "Julio", "08": "Agosto", "09": "Septiembre", "10": "Octubre", "11": "Noviembre", "12": "Diciembre"
        }
        return month_map[month]
    except ValueError:
        return "Desconocido"
```

Figura 6.9: Extracción mes de documento

4. Se hizo la construcción de una función para la extracción del nombre del dispositivo involucrado en la alerta, como se puede ver en la figura 6.10. Para esto, primero, debido a que cada uno tiene un título diferente para el nombre del dispositivo, se identificó si es una alerta sanitaria (nombre del producto) o un informe de seguridad (asunto), y captura el contenido asociado hasta encontrar un nuevo encabezado.

```python
def extract_device_name(filepath):
    """Extrae el nombre del dispositivo médico o equipo del PDF hasta el siguiente título válido."""
    try:
        with fitz.open(filepath) as pdf:
            for page in pdf:
                text = page.get_text("text")  # Extraer texto plano

                # Verificar si es un Informe de Seguridad
                if "Informe de Seguridad" in text:
                    # Buscar "Asunto" y detener en "No. identificación interna del Informe de Seguridad"
                    match = re.search(
                        r'Asunto\s*:\s*(.+?)(?=\n(?:No\. identificación interna del Informe de Seguridad|[A-ZÁÉÍÓÚÑ ]{2,}[:])|$)',
                        text, re.IGNORECASE | re.DOTALL
                    )
                else:
                    # Buscar "Nombre del producto" para alertas
                    match = re.search(
                        r'Nombre del producto\s*:\s*(.+?)(?=\n(?:No\. identificación interna del Informe de Seguridad|[A-ZÁÉÍÓÚÑ ]{2,}[:])|$)',
                        text, re.IGNORECASE | re.DOTALL
                    )

                if match:
                    device_name = match.group(1).strip()  # Capturar solo el texto relevante
                    return device_name

    except Exception as e:
        print(f"Error al procesar el PDF {filepath}: {e}")

    return "No especificado"
```

Figura 6.10: Extracción nombre del dispositivo

5. Se diseñó una función para la predicción del tipo de dispositivo, para esto se cargó el modelo y el vectorizador previamente entrenados y se obtuvo el resultado.

6. Se realizó la construcción de una función, que se muestra en la figura 6.11, encargada de extraer el número de registro sanitario. Mediante una expresión regular, se busca el patrón: "Registro sanitario: INVIMA", seguido de una combinación de letras, números o guiones que representan el código oficial.

```python
def extract_registro_invima(filepath):
    """Extrae el número de registro sanitario del contenido del PDF."""
    try:
        with fitz.open(filepath) as pdf:
            for page in pdf:
                text = page.get_text()
                # Busca "Registro sanitario:" seguido por cualquier formato de código relevante
                match = re.search(r'Registro sanitario\s*:\s*(?:INVIMA\s*)?([A-Z0-9-]+)', text, re.IGNORECASE)
                if match:
                    registro = match.group(1)  # Captura el número del registro
                    return registro
    except Exception as e:
        print(f"Error al procesar el PDF {filepath}: {e}")
    return "No encontrado"
```

Figura 6.11: Extracción registro INVIMA

7. Se realizó una función para extraer la descripción del caso, en esta se debe identificar nuevamente el tipo de documento debido a que para los informes de seguridad se utilizan las librerías pdfplumber y Pypdf2, como se puede observar en la figura 6.12, esto debido a que estas librerías leen y navegan el documento conservando la estructura visual del mismo, pudiendo así acceder a la información necesitada, ignorando el formato del documento. Y, para las alertas sanitarias se utiliza la librería fitz para acceder al documento, como se puede ver en la figura 6.13. Posteriormente, se busca el título descripción del caso y se extrae la información hasta el siguiente encabezado que exista en el documento. En algunos casos, hay un cambio de página, por lo que, debido al formato, sale una fecha entre el texto, que se corrige y se elimina esta fecha y se unen los párrafos separados por el cambio de página.

```python
def extract_case_description(filepath):
    """
    Extrae el contenido de 'Descripción del caso' de un PDF.
    Usa PyPDF2 para informes de seguridad y fitz (PyMuPDF) para alertas.
    """
    try:
        # Leer el PDF completo para buscar "Informe de Seguridad"
        with open(filepath, 'rb') as archivo_pdf:
            lector_pdf = PyPDF2.PdfReader(archivo_pdf)
            texto_completo = ""
            for pagina in lector_pdf.pages:
                texto_completo += pagina.extract_text()

        # Determinar si es un informe de seguridad o una alerta
        if "Informe de Seguridad" in texto_completo:
            with pdfplumber.open(filepath) as pdf:
                texto_completo = ""

                # Leer todas las páginas del PDF
                for pagina in pdf.pages:
                    texto_completo += pagina.extract_text()

            # Buscar la sección "Descripción del caso"
            inicio = texto_completo.find("Descripción del caso")
            if inicio != -1:
                # Encontrar el final del apartado
                final = texto_completo.find("Información para profesionales de la salud", inicio)
                if final == -1:
                    final = len(texto_completo)  # Hasta el final del texto si no hay delimitador claro

                # Extraer solo el contenido después del título
                contenido = texto_completo[inicio + len("Descripción del caso"):final].strip()
                return contenido
            else:
                return "No se encontró la sección 'Descripción del caso'."
```

Figura 6.12: Extracción descripción del caso informes de seguridad

```
249     def extract_case_description(filepath):
else:
    # Lógica para alertas (fitz)
    with fitz.open(filepath) as pdf:
        capturing = False
        description_lines = []

        for page in pdf:
            text = page.get_text("text")  # Extraer texto de la página
            lines = text.splitlines()

            for line in lines:
                line = line.strip()
                # Iniciar captura al encontrar "Descripción del caso"
                if re.match(r'Descripción del caso', line, re.IGNORECASE):
                    capturing = True
                    continue

                # Detener captura al encontrar un nuevo encabezado
                if capturing and re.match(r'(Medidas para|Antecedentes|Acciones tomadas|A los|Nota|Referencia|Registro Sanitario|Enlace Relacionado)', line, re.IGNORECASE):
                    capturing = False
                    break

                # Capturar líneas relevantes
                if capturing:
                    description_lines.append(line)

        # Unir las líneas capturadas en un solo párrafo
        description = " ".join(description_lines)
        # Limpiar fechas del texto
        description = re.sub(
            r'\b\d{1,2}\s(de\s)?(enero|febrero|marzo|abril|mayo|junio|julio|agosto|septiembre|octubre|noviembre|diciembre)\s\d{4}\b',
            '',
            description,
            flags=re.IGNORECASE
        )
        return " ".join(description.split()).strip() if description else "No encontrado"
```

Figura 6.13: Extracción descripción del caso alertas sanitarias

8. Se crea una función para cargar y actualizar o crear un archivo Excel para almacenar los datos extraídos. La información extraída con las funciones descritas anteriormente se organiza en un dataframe y se combina con los datos existentes en el archivo Excel, eliminando duplicados según el número de alerta.

9. Se construye una función donde se integran todas las funciones anteriormente descritas, es decir, se llama a las funciones que se encargan de realizar el procedimiento y se ejecutan.

10. Por último, se crea una interfaz muy simple, mostrada en la figura 6.14, con el fin de realizar todo el proceso desde un solo botón interactivo.

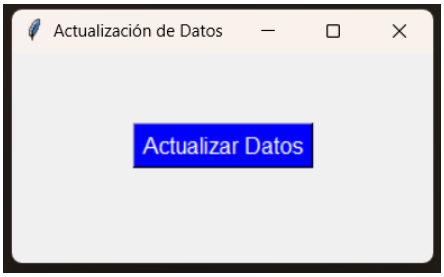


Figura 6.14: Interfaz

OE3:

Se realiza un plan de pruebas con el fin de evaluar el funcionamiento de la herramienta desarrollada, este consiste en las siguientes etapas:

1. Pruebas Comparativas: Se realizaron comparaciones entre la extracción de datos automatizada y el proceso manual tradicional. Esto permitió medir el tiempo aproximado, de forma manual, que se toma con cada método para la realización de la tarea y la precisión de la herramienta.

2. Análisis de Resultados: Los resultados obtenidos se analizaron en términos de precisión, tiempo de procesamiento y consistencia de los datos extraídos. Se identificaron áreas de mejora para optimizar el rendimiento de la herramienta.

6.2. Recursos Técnicos

6.2.1. Sistema

- Aplicación: VScode
- Lenguaje utilizado: Python Versión 3.11.2544.0
- Procesador: Intel(R) Core (TM) i5-9300H CPU @ 2.40GHz 2.40 GHz
- RAM instalada: 8,00 GB
- Tipo de sistema: sistema operativo de 64 bits
- Edición: Windows 11
- Versión: 24H2
- Tarjeta gráfica: NVIDIA GeForce GTX 1650
- Fabricante: Acer

Capítulo 7

Resultados y Discusión

7.1. Resultados Obtenidos

1. Al realizar las preguntas pertinentes, se obtuvo que la herramienta debía permitir:

   - La automatización completa del proceso para evitar la entrada manual de datos.
   - La creación de una base de datos accesible y de fácil manejo (como archivos Excel).
   - La implementación de una interfaz simple que minimice la interacción del usuario con el código, permitiendo su uso por personas con poca experiencia técnica.
   - La precisión en la extracción de datos clave, como el nombre del dispositivo, el registro INVIMA, el tipo de alerta, la fecha, y la descripción del caso.

   Esto ayudó a comprender totalmente lo que se requería para que la herramienta funcionara correctamente.

2. al finalizar la ejecución de la herramienta, se obtuvo lo siguiente:

   - Se genera una carpeta en la memoria local donde se almacenaron los archivos PDF procesados, así como se muestra en la figura 7.1, donde estos se guardan con el nombre dado posterior a la limpieza de los URLs, dejando así una forma mas sencilla de identificar y acceder al documento de interés.

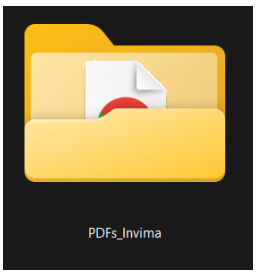


Figura 7.1: Carpeta de PDFs

   - Se obtuvo el resultado del entrenamiento de los diferentes módelos predictivos, obteniendo el porcentaje de precisión y su matriz de confusión correspondiente, donde los items que se observan son: 0 = Biomédico, 1 = Convencionales, 2 = Implantable, 3 = Reactivo.

Además, se obtuvo cuales fueron los hiperparametros que mejor resultado brindaron para cada uno y cual de los modelos es el que debía ser utilizado.

a) Random forest: Despues de realizar el entrenamiento de este utilizando la herramienta de gridsearch, los hiperparametros con los que se obtuvo el mejor resultado fueron: "max_depth": None, "min_samples_split": 5, "n_estimators": 300, dando así un 64 % de precisión, como se puede observar en la figura 7.2, además, se obtuvo la matriz de confusión, que se muestra en la figura 7.3, que deja en evidencia que clasifica correctamente la mayoría de la clase 0 y parte de la clase 3, pero muestra errores en la clase 1 y especialmente en la clase 2, que se confunde con la clase 0 y 1.

	precision	recall	f1-score	support
biomedico	0.64	0.79	0.71	68
convencional	0.52	0.45	0.48	33
implantable	0.60	0.35	0.44	17
reactivo	0.79	0.68	0.73	34
accuracy			0.64	152
macro avg	0.64	0.57	0.59	152
weighted avg	0.64	0.64	0.64	152

Figura 7.2: Resultado del modelo random forest en porcentajes

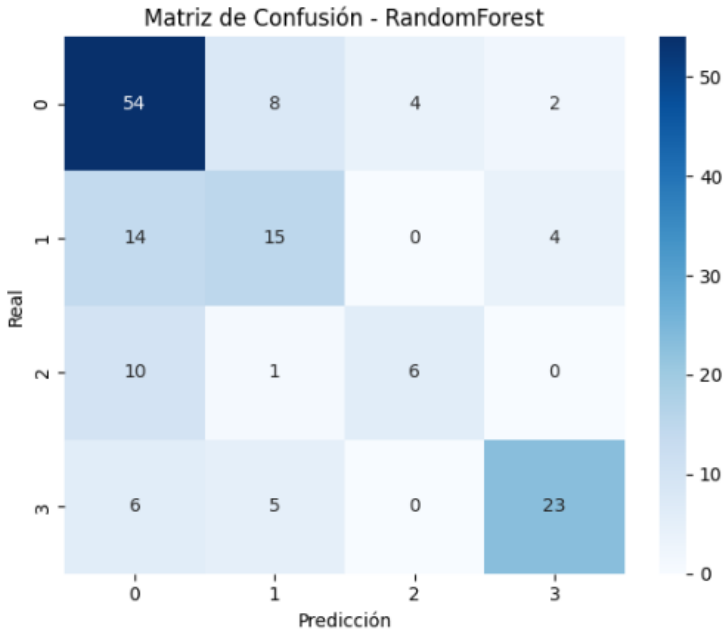


Figura 7.3: Matriz de confusión random forest

b) XGboost: En el caso del modelo XGBoost, tras ejecutar el proceso de ajuste de hiperparámetros mediante GridSearchCV, se identificó que la combinación óptima correspondía a "max_depth": 3, "n_estimators": 300. Esta configuración permitió alcanzar una precisión del 54 %, como se evidencia en la figura 7.4. Asimismo, se generó la matriz de confusión correspondiente, ilustrada en la figura 7.5, que deja en evidencia que tiene buena precisión para la clase 0, pero hay bastante confusión entre la clase 1 y las demás, mostrando dificultades en distinguir correctamente esa categoría.

	precision	recall	f1-score	support
biomedico	0.57	0.75	0.65	68
convencional	0.33	0.27	0.30	33
implantable	0.38	0.18	0.24	17
reactivo	0.70	0.56	0.62	34
accuracy			0.54	152
macro avg	0.49	0.44	0.45	152
weighted avg	0.53	0.54	0.52	152

Figura 7.4: Resultado del modelo XGBoost

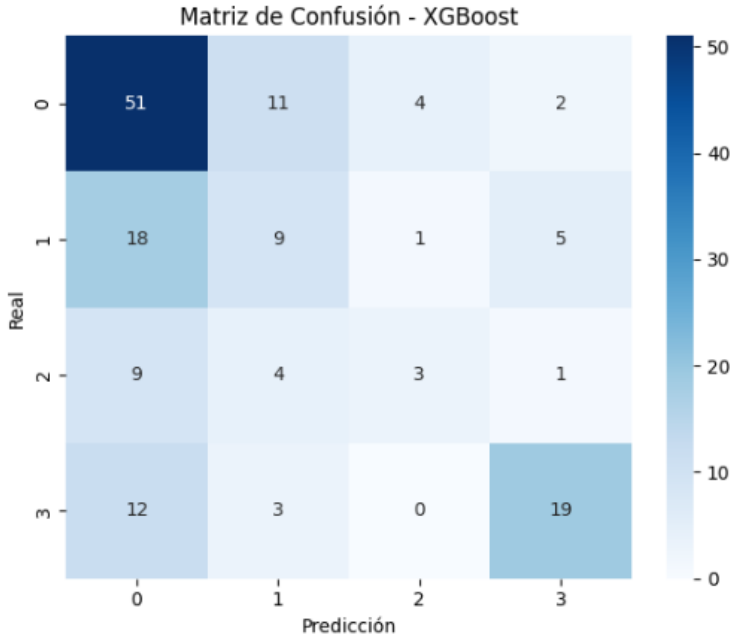


Figura 7.5: Matriz de confusión XGBoost

c) KNN: Para el modelo basado en K-Nearest Neighbors (KNN), los parámetros óptimos seleccionados por GridSearchCV fueron: "n_neighbors": 5 y "weights": 'distance'.

Esta configuración ofreció un desempeño del 55 % de precisión, tal como se muestra en la figura 7.6. La matriz de confusión correspondiente, presentada en la figura 7.7 presenta errores notables en todas las clases, especialmente en la clase 0, que se confunde con las demás, y en la clase 1, que tiene una alta tasa de falsos positivos con la clase 0.

	precision	recall	f1-score	support
biomedico	0.59	0.74	0.65	68
convencional	0.48	0.42	0.45	33
implantable	0.33	0.12	0.17	17
reactivo	0.56	0.53	0.55	34
accuracy			0.55	152
macro avg	0.49	0.45	0.46	152
weighted avg	0.53	0.55	0.53	152

Figura 7.6: Resultado del modelo KNN

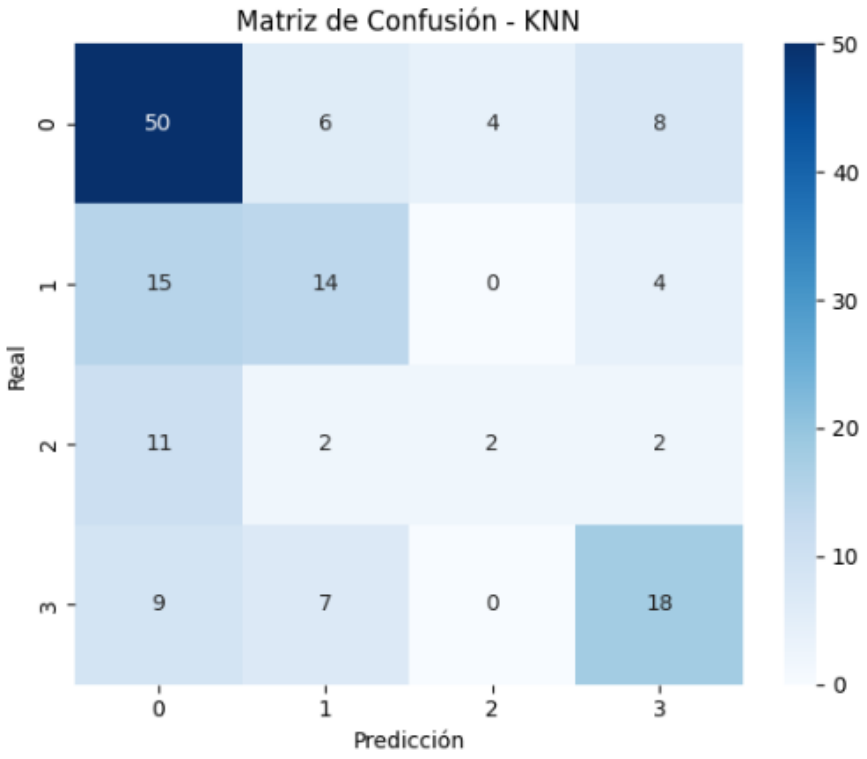


Figura 7.7: Matriz de confusión KNN

d) MLP: En el caso del clasificador basado en redes neuronales MLP, se obtuvo como mejor configuración la estructura de capas ocultas "hidden_layer_sizes": (50, 50) y la función de activación "activation": 'relu'. Con estos parámetros, se alcanzó una precisión del 63 %, visualizada en la figura 7.8. La matriz de confusión que se muestra en la figura 7.9 permite visualizar que presenta confusiones marcadas, especialmente entre las clases 0, 1 y 3. Aunque acierta algunos casos, su rendimiento general es bajo, indicando dificultades para aprender patrones claros con los datos disponibles.

	precision	recall	f1-score	support
biomedico	0.69	0.75	0.72	68
convencional	0.49	0.64	0.55	33
implantable	0.45	0.29	0.36	17
reactivo	0.79	0.56	0.66	34
accuracy			0.63	152
macro avg	0.61	0.56	0.57	152
weighted avg	0.64	0.63	0.63	152

Figura 7.8: Resultado del modelo MLP

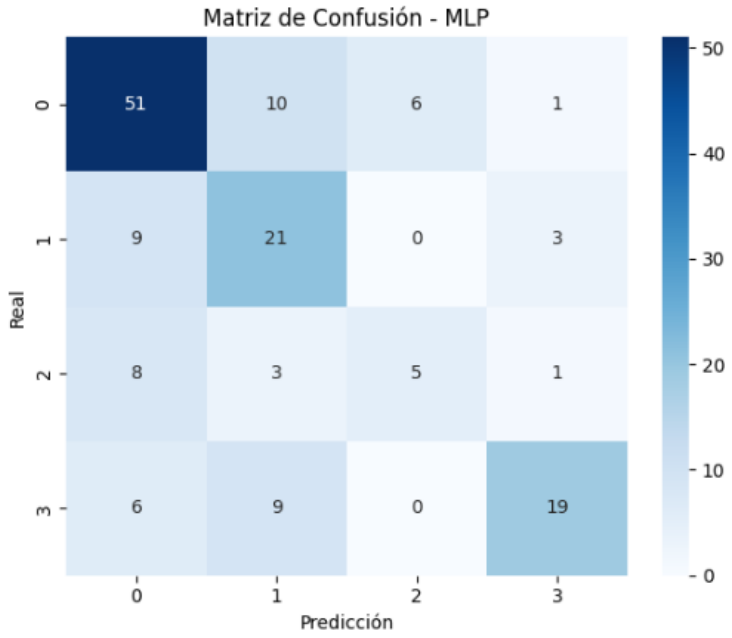


Figura 7.9: Matriz de confusión MLP

e) SVM: el modelo SVM arrojó sus mejores resultados con los hiperparámetros "C": 1 y "kernel": 'rbf', lo que permitió obtener una precisión del 71 %. Este resultado puede verse en la figura 7.10. En la figura 7.11 se presenta la matriz de confusión donde

se observa un buen rendimiento, especialmente en la clase 0 y la clase 3, donde la mayoría de los valores reales fueron correctamente clasificados.

	precision	recall	f1-score	support
biomedico	0.67	0.88	0.76	68
convencional	0.76	0.58	0.66	33
implantable	0.71	0.29	0.42	17
reactivo	0.77	0.71	0.74	34
accuracy			0.71	152
macro avg	0.73	0.61	0.64	152
weighted avg	0.72	0.71	0.70	152

Figura 7.10: Resultado del modelo SVM

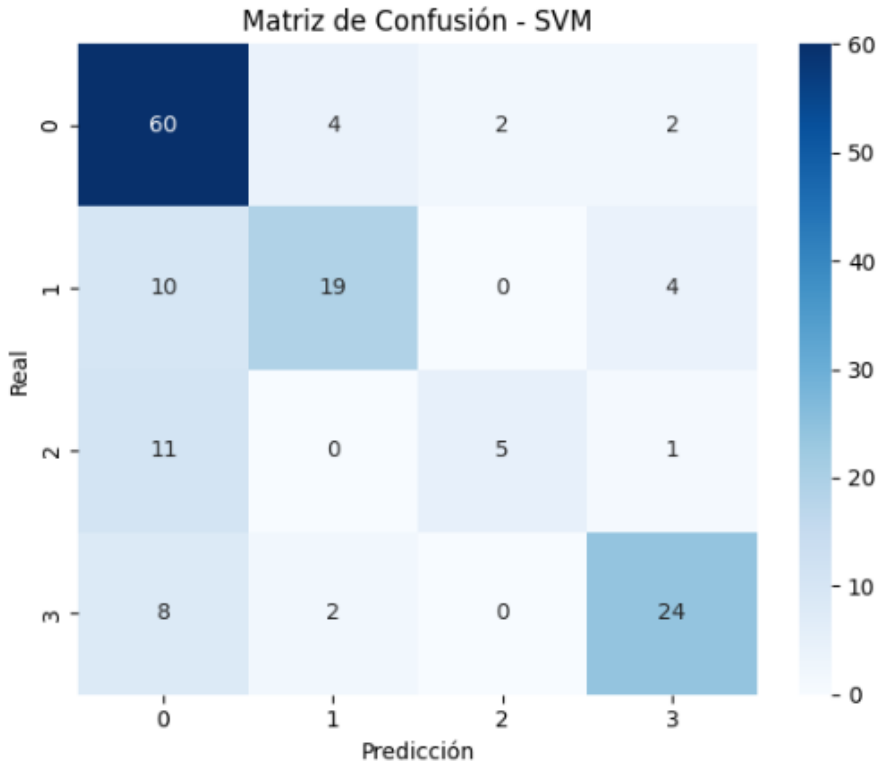
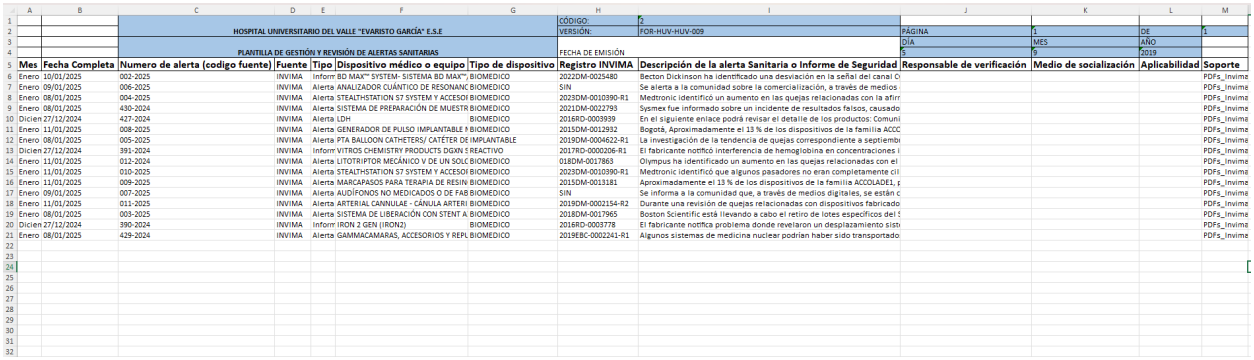


Figura 7.11: Matriz de confusión SVM

Lo anterior dejó en claro que de los modelos predictivos entrenados, el que dió el mejor resultado fue SVM, con un 71 % de precisión, este resultado puede deberse a que este optimiza

el margen entre clases y a la naturaleza del algoritmo, que se adapta especialmente bien a espacios de alta dimensionalidad como los generados por la vectorización TF-IDF, donde cada término del texto representa una dimensión.

3. Al final del proceso, se logró generar un archivo Excel, que se muestra en la figura 7.12, el cual reúne toda la información importante extraída de las alertas sanitarias. En este archivo se organizaron datos como la fecha de emisión, el tipo y número de alerta, el nombre del dispositivo, el tipo de dispositivo (que fue predicho por el modelo), el registro INVIMA y la descripción del caso. Este archivo permite tener toda la información centralizada y más fácil de consultar, lo que ayuda bastante en la revisión de alertas y en la toma de decisiones dentro de los procesos de tecnovigilancia, ya que antes toda esta información estaba dispersa en documentos PDF.



		HOSPITAL UNIVERSITARIO DEL VALLE "EVARISTO GARCÍA" E.S.E				CÓDIGO:	2						
						VERSIÓN:	FOR-HUV-HUV-009	PÁGINA	1	DE	1		
		PLANTILLA DE GESTIÓN Y REVISIÓN DE ALERTAS SANITARIAS				FECHA DE EMISIÓN		DÍA	MES	AÑO			
								5	6	2019			
Mes	Fecha Completa	Numero de alerta (codigo fuente)	Fuente	Tipo	Dispositivo médico o equipo	Tipo de dispositivo	Registro INVIMA	Descripción de la alerta Sanitaria o Informe de Seguridad	Responsable de verificación	Medio de socialización	Aplicabilidad	Soporte	
Enero	10/01/2025	002-2025	INVIMA	Inform	BD MAX™ SYSTEM- SISTEMA BD MAX™)	BIOMEDICO	2022DM-0025480	Becton Dickinson ha identificado una desviación en la señal del canal O				PDFs_Invima	
Enero	09/01/2025	006-2025	INVIMA	Alerta	ANALIZADOR CUÁNTICO DE RESONANC	BIOMEDICO	SIN	Se alerta a la comunidad sobre la comercialización, a través de medios				PDFs_Invima	
Enero	08/01/2025	004-2025	INVIMA	Alerta	STEALTHSTATION S7 SYSTEM Y ACCESOI	BIOMEDICO	2023DM-0010390-R1	Medtronic identificó un aumento en las quejas relacionadas con la afirr				PDFs_Invima	
Enero	08/01/2025	430-2024	INVIMA	Alerta	SISTEMA DE PREPARACIÓN DE MUESTR	BIOMEDICO	2021DM-0022793	Sysmex fue informado sobre un incidente de resultados falsos, causado				PDFs_Invima	
Dicien	27/12/2024	427-2024	INVIMA	Alerta	LDH	BIOMEDICO	2016RD-0003939	En el siguiente enlace podrá revisar el detalle de los productos: Comuni				PDFs_Invima	
Enero	11/01/2025	008-2025	INVIMA	Alerta	GENERADOR DE PULSO IMPLANTABLE I	BIOMEDICO	2015DM-0012932	Bogotá, Aproximadamente el 13 % de los dispositivos de la familia ACCO				PDFs_Invima	
Enero	08/01/2025	005-2025	INVIMA	Alerta	PTA BALLOON CATHETERS/ CATÉTER DE	IMPLANTABLE	2019DM-0004622-R1	La investigación de la tendencia de quejas correspondiente a septiembi				PDFs_Invima	
Dicien	27/12/2024	391-2024	INVIMA	Inform	VITROS CHEMISTRY PRODUCTS DGKN 5	REACTIVO	2017RD-0000206-R1	El fabricante notificó interferencia de hemoglobina en concentraciones i				PDFs_Invima	
Enero	11/01/2025	012-2024	INVIMA	Alerta	LITOTRIPTOR MECÁNICO Y DE UN SOLC	BIOMEDICO	01BDM-0017863	Olympus ha identificado un aumento en las quejas relacionadas con el				PDFs_Invima	
Enero	11/01/2025	010-2025	INVIMA	Alerta	STEALTHSTATION S7 SYSTEM Y ACCESOI	BIOMEDICO	2023DM-0010390-R1	Medtronic identificó que algunos pasadores no eran completamente cil				PDFs_Invima	
Enero	11/01/2025	009-2025	INVIMA	Alerta	MARCAPASOS PARA TERAPIA DE RESIN	BIOMEDICO	2015DM-0013181	Aproximadamente el 13 % de los dispositivos de la familia ACCOLADE1, p				PDFs_Invima	
Enero	09/01/2025	007-2025	INVIMA	Alerta	AUDÍFONOS NO MEDICADOS O DE FAB	BIOMEDICO	SIN	Se informa a la comunidad que, a través de medios digitales, se están c				PDFs_Invima	
Enero	11/01/2025	011-2025	INVIMA	Alerta	ARTERIAL CANNULAE - CÁNULA ARTERI	BIOMEDICO	2019DM-0002154-R2	Durante una revisión de quejas relacionadas con dispositivos fabricado				PDFs_Invima	
Enero	08/01/2025	003-2025	INVIMA	Alerta	SISTEMA DE LIBERACIÓN CON STENT A	BIOMEDICO	2018DM-0017965	Boston Scientific está llevando a cabo el retiro de lotes específicos del S				PDFs_Invima	
Dicien	27/12/2024	390-2024	INVIMA	Inform	IRON 2 GEN (IRON2)	BIOMEDICO	2016RD-0003778	El fabricante notifica problema donde revelaron un desplazamiento sist				PDFs_Invima	
Enero	08/01/2025	429-2024	INVIMA	Alerta	GAMMACAMARAS, ACCESORIOS Y REPL	BIOMEDICO	2019EBC-0002241-R1	Algunos sistemas de medicina nuclear podrían haber sido transportado				PDFs_Invima	

Figura 7.12: Base de datos resultante

4. Como es mostrado en el cuadro 7.1, se evaluó correctamente la herramienta con una prueba comparativa entre la extracción automatizada y la tradicional, donde fue clave la participación de la lider de tecnovigilancia de Imbanaco y de un auxiliar biomédico de la Clínica Colombia, enfocándose en el tiempo de ejecución, que para el caso del método manual se realizó un promedio entre los tiempos de las 2 personas que realizaron el proceso de extracción manual, la precisión y la calidad de los datos procesados. Para su ejecución, se tomaron 20 archivos de prueba, primero dejando en evidencia que el tiempo de procesamiento al utilizar la herramienta se redujo, aproximadamente, en un 98 %, y, que el porcentaje de error, que con el método manual se tenía aproximadamente en el 30 %, usando la herramienta, pasó a ser, aproximadamente, de 1 %.

Método de extracción	**Tiempo aproximado**	**Porcentaje de error por 20 alertas**
Manual	222.4 minutos	30 %
Automatizado	20 segundos	1 %

Cuadro 7.1: Comparación del tiempo y precisión entre métodos de extracción para 20 alertas.

7.1.1. Discusión General

La herramienta desarrollada no solo permite la automatización de los procesos de extracción e integración de alertas sanitarias, sino que también mejora significativamente el tiempo necesario para la actualización de la base de datos, reduce la probabilidad de errores humanos y facilita la toma de decisiones por parte de los actores encargados de la tecnovigilancia. A través de la implementación del modelo de clasificación de dispositivos médicos, se logró una solución que no solo extrae datos relevantes de los documentos en formato PDF, sino que también clasifica los dispositivos en las categorías predefinidas, brindando un valor agregado en la organización y análisis de la información.

Además, se optó por almacenar la base de datos en formato Excel, debido a la facilidad que ofrece esta herramienta para acceder, revisar, actualizar y filtrar los datos según los criterios de interés. Esto permite que el archivo pueda ser utilizado por diferentes perfiles profesionales sin necesidad de conocimientos técnicos avanzados en programación.

Sin embargo, se identificó que, en lo que respecta a la clasificación precisa de la totalidad de los tipos de dispositivos médicos, el modelo aún tiene margen de mejora. Esto se debe, principalmente, al tamaño limitado de la base de datos de entrenamiento, que contenía aproximadamente 400 registros. Con una base de datos más amplia, representativa y balanceada en cuanto a las diferentes clases de dispositivos, es probable que se logre una mejora en el rendimiento del modelo predictivo.

En general, los resultados obtenidos respaldan la viabilidad de la automatización en el área de tecnovigilancia y demuestran que, con ajustes y mejoras, esta herramienta puede convertirse en un apoyo clave para las instituciones encargadas de la supervisión de dispositivos médicos.

Capítulo 8

Conclusiones

Se logró identificar las necesidades técnicas específicas de los profesionales de la salud, líderes de tecnovigilancia y otros actores relevantes. Gracias a las consultas previas, la herramienta fue diseñada con características clave que satisfacen los requerimientos de los usuarios, mejorando la toma de decisiones dentro del ámbito sanitario.

Se elaboró un algoritmo que automatiza completamente el proceso de extracción, integración y actualización de información de alertas sanitarias. Este algoritmo optimiza los tiempos de trabajo y mejora la precisión en la extracción de los datos, eliminando la intervención manual y asegura la integración de información en las bases de datos de las instituciones sanitarias.

Se realizó una comparación entre los métodos tradicionales de extracción manual de información y el proceso automatizado. Los resultados mostraron una reducción significativa en el tiempo necesario para completar la tarea, que valida la hipótesis de que la automatización mejora la eficiencia y la fiabilidad de los datos extraídos.

La implementación de la herramienta contribuye de manera significativa a optimizar los procesos administrativos dentro de las instituciones de salud. Gracias a la reducción de tiempos de trabajo y la mejora en la precisión de los datos, se ha logrado un avance importante en la gestión de alertas sanitarias. Los resultados obtenidos validan la hipótesis planteada al inicio del proyecto: la automatización del proceso reduce el tiempo de trabajo y mejora la fiabilidad de los resultados, que permite una respuesta más rápida ante posibles incidentes o eventos adversos, promoviendo la seguridad del paciente.

Durante el desarrollo del algoritmo, se realizó un análisis exhaustivo de las opciones tecnológicas disponibles para la extracción de datos desde archivos PDF. Se seleccionaron las herramientas que mejor se adaptaban a las necesidades del proyecto, destacándose por su eficacia y capacidad para procesar archivos de manera rápida y precisa. Además, las consultas con los profesionales encargados del procedimiento permitieron identificar los requerimientos esenciales y diseñar un código que cumpliera con las expectativas planteadas, asegurando que la herramienta fuese de utilidad práctica en el ámbito sanitario.

Este código tiene algunas conexiones con NLP (Natural Language Processing) en la prediccion del tipo del dispositivo médico al tener un vectorizador que guarde las caracteristicas principales con las que se clasificaron los dispositivos en sus respectivos tipos.

Se compararon los resultados obtenidos con la herramienta automatizada frente a los métodos tradicionales de extracción manual. Los resultados mostraron una reducción significativa en el tiempo necesario para completar el proceso, que valida la eficiencia de la herramienta desarrollada. Además, se observó una reducción considerable en el margen de error, respaldando la hipótesis de que la automatización mejora la precisión y disminuye los errores humanos en la extracción de datos, garantizando información más confiable para las instituciones de salud.

En conclusión, el desarrollo e implementación de esta herramienta automatizada no solo cumple con los objetivos establecidos al inicio del proyecto, sino que también mejora la eficiencia operativa de las instituciones encargadas de la gestión de alertas sanitarias. Este proyecto demuestra el impacto positivo de la automatización en procesos críticos de salud, contribuyendo a una mejor gestión de riesgos y promoviendo la seguridad del paciente. A futuro, la herramienta podría extenderse a otros sistemas de alerta sanitarios, mejorando aún más la capacidad de respuesta ante riesgos, incidentes o eventos adversos en el ámbito sanitario.

Capítulo 9

Trabajos futuros

Para continuar mejorando la herramienta desarrollada, se proponen los siguientes trabajos futuros:

- **Ampliación de Fuentes de Datos:** Integrar nuevas bases de datos nacionales e internacionales, incluyendo la capacidad de extraer información directamente del Instituto de Salud Pública de Chile (ISPCH) o del FDA, para ampliar la cobertura de alertas sanitarias y mejorar la detección temprana de riesgos.
- **Optimización del Algoritmo de NLP:** Mejorar el procesamiento de lenguaje natural para interpretar textos más complejos y en diferentes idiomas, permitiendo una mayor adaptabilidad de la herramienta.
- **Aplicación de Inteligencia Artificial Predictiva:** Implementar modelos predictivos que identifiquen posibles riesgos sanitarios futuros a partir del análisis de tendencias históricas y datos actuales.

Capítulo 10

Anexos

Anexo – Códigos Utilizados para el desarrollo de una herramienta que permita la automatización de la extracción de información de Alertas Sanitarias

A continuación se presenta el link que pertenece a un repositorio público de github que contiene los códigos utilizados en el proyecto.

- https://github.com/alejoc02x/Herramienta-Extracci-n-de-informaci-n-y-actualizaci-n-base-de-datos..git

Bibliografía

[1] M. de salud y protección social. (2024) Alertas sanitarias invima. [Online]. Available: https://www.saludputumayo.gov.co/index.php/saludpublica/gestion-salud/medicamentos/alertas-sanitarias#:~:text=Que%20es%20una%20Alerta%20Sanitaria,Salud%20P%C3%BAblica%20urgentes%20y%20eficaces

[2] I. School. (2024) ¿qué es el nlp o procesamiento de lenguaje natural? [Online]. Available: https://www.iebschool.com/blog/que-es-nlp-big-data/

[3] B. Zhao, "Clinical data extraction and normalization of cyrillic electronic health records via deep-learning natural language processing," *JCO Clinical Cancer Informatics*, no. 3, pp. 1–9, 2019, pMID: 31577448. [Online]. Available: https://doi.org/10.1200/CCI.19.00057

[4] INVIMA. (2023) Guia para la implementación eficaz del programa de tecnovigilancia. [Online]. Available: https://www.invima.gov.co/sites/default/files/dispositivos-medicos/Vigilancia/Programa-nacional-de-Tecnovigilancia/Documentos-de-interes/GU%C3%8DA%20TECNOVIGILANCIA%202023-final.pdf

[5] M. D. H. Siddhartha R. Jonnalagadda, Pawan Goyal, "Automating data extraction in systematic reviews: a systematic review," *Springer Link*, vol. 4, no. 78, 2015.

[6] J. M. S. Z. P. Del Valle Buitrago, Carolina Ortiz Vásquez, "Tecnovigilancia: complemento del sistema de calidad de la atención en salud, en colombia," *Universidad CES*, 2010.

[7] INVIMA, "Abc de dispositivos médicos," *MINISTERIO DE SALUD Y PROTECCIÓN SOCIAL Instituto Nacional de Vigilancia de Medicamentos y Alimentos - INVIMA*, pp. 13–48, 2013.

[8] M. de la Protección Social. (2008) Resolucion 4816 de 2008. [Online]. Available: https://www.saludcapital.gov.co/DSP/Tecnovigilancia/Resoluci%C3%B3n%204816%20de%202008.pdf

[9] D. K. R. W. L. William A Yasnoff, Patrick W. O'Carroll, "Public health informatics: Improving and transforming public health in the information age," *Public Health Management and Practice*, vol. 7, no. 6, pp. 67–75, 2001.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., ser. Department of computer, science and department of stadistic. USA: Springer, 2006.

[11] B. E. D. J.A. Magnuson, *Public Health Informatics and Information Systems*, 2nd ed. USA: Springer, 2014.

[12] J. M. Daniel Jurafsky, *Speech and Language Processing*, 2nd ed. USA: Pearson, 2008.

[13] R. Mitchell, *Web Scraping with Python: Collecting Data from the Modern Web*, 1st ed. USA: O'Reilly Media, 2015.

[14] INVIMA. (2024) Que hacemos. [Online]. Available: https://www.invima.gov.co/el-instituto/que-hacemos

[15] G. de Colombia, "Decreto nÚmero 4725 de 2005," *MINISTERIO DE LA PROTECCION SOCIAL*, 2005.

[16] M. de salud y protección social. (2023) Seguridad del paciente. [Online]. Available: https://www.minsalud.gov.co/salud/CAS/Paginas/seguridad-del-paciente.aspx

[17] IBM. (2021) ¿qué es el desarrollo de software? [Online]. Available: https://www.ibm.com/es-es/topics/software-development

[18] ORACLE. (2024) ¿qué es una base de datos? [Online]. Available: https://www.oracle.com/co/database/what-is-database/

[19] E. K. O. K. G. C. T. Y.-J. H. A. M. H. Elsa A. Olivetti, Jacqueline M. Cole, "Data-driven materials research enabled by natural language processing and information extraction," *Applied Physics Reviews*, vol. 7, no. 4, 2020.

[20] M. Costantino, R. Morgan, R. Collingham, and R. Carigliano, "Natural language processing and information extraction: qualitative analysis of financial news articles," in *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, 1997, pp. 116–122.

[21] L. Ohno-Machado, P. Nadkarni, and K. Johnson, "Natural language processing: algorithms and tools to extract computable information from EHRs and from the biomedical literature," *Journal of the American Medical Informatics Association*, vol. 20, no. 5, pp. 805–805, 09 2013. [Online]. Available: https://doi.org/10.1136/amiajnl-2013-002214

[22] P. Geeks. (2025) Python os module. [Online]. Available: https://pythongeeks.org/python-os-module/

[23] S. Reddy. (2019) Regular expressions in python. [Online]. Available: https://www.codementor.io/@sadhanareddy/regular-expressions-in-python-rcxs7fq0x

[24] PyMuPDF. (2023) Highlighting text with pymupdf. [Online]. Available: https://medium.com/@pymupdf/highlighting-text-with-pymupdf-08b228044da7

[25] Shubham. (2022) Python http client request - get, post. [Online]. Available: https://www.digitalocean.com/community/tutorials/python-http-client-request-get-post

[26] geeksforgeeks. (2021) Extract json from html using beautifulsoup in python. [Online]. Available: https://www.geeksforgeeks.org/extract-json-from-html-using-beautifulsoup-in-python/

[27] ——. (2024) Pandas dataframe. [Online]. Available: https://www.geeksforgeeks.org/python-pandas-dataframe/

[28] J. developers. (2021) Joblib: running python functions as pipeline jobs. [Online]. Available: https://joblib.readthedocs.io/en/stable/

[29] PyPDF2. (2008) Welcome to pypdf2. [Online]. Available: https://pypdf2.readthedocs.io/en/3.x/

[30] J. B. B. L. e. a. Jacob Fenton, Dan Nguyen. (2025) pdfplumber 0.11.5. [Online]. Available: https://pypi.org/project/pdfplumber/

[31] geeksforgeeks. (2024) Working with excel spreadsheets in python. [Online]. Available: https://www.geeksforgeeks.org/working-with-excel-spreadsheets-in-python/

[32] iamabhishek. (2020) Tkinter windows, widgets and frames. [Online]. Available: https://www.studytonight.com/tkinter/tkinter-windows-widgets-and-frames

[33] J. Hao and T. K. Ho, "Machine learning made easy: a review of scikit-learn package in python programming language," *Journal of Educational and Behavioral Statistics*, vol. 44, no. 3, pp. 348–361, 2019.

[34] M. L. Waskom, "Seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.